Audio Mostly 2007

2nd Conference on Interaction with Sound

Conference Proceedings

September 27 – 28, 2007
Röntgenbau, Ilmenau, Germany

Audio Mostly 2007 - 2nd Conference on Interaction with Sound

Conference Proceedings

ISBN 978-3-00-022823-0

Fraunhofer Institute for Digital Media Technology IDMT

Director
Prof. Dr.-Ing. Karlheinz Brandenburg

Ehrenbergstr. 29
98693 Ilmenau, Germany

Tel.+49 (0) 36 77/69-43 41
Fax+49 (0) 36 77/69-43 99

info@idmt.fraunhofer.de

www.idmt.fraunhofer.de

Audio Mostly Conference Commitee

Committee

Katarina Delsing
Interactive Institute, Sonic Studio, Piteå, Sweden
katarina.delsing@tii.se

Holger Grossmann
Fraunhofer Institute for Digital Media Technology, Ilmenau, Germany
grn@idmt.fraunhofer.de

Stuart Cunningham
University of Wales, Wrexham, UK
s.cunningham@newi.ac.uk

Lilian Johansson
Interactive Institute, Sonic Studio, Piteå, Sweden
lilian.johansson@tii.se

Mats Liljedahl
Interactive Institute Sonic Studio, Piteå, Sweden
mats.liljedahl@tii.se

David Moffat
Glasgow Caledonian University, Glasgow, UK
d.c.moffat@gcal.ac.uk

Nigel Papworth
Interactive Institute, Sonic Studio, Piteå, Sweden
nigel.papworth@tii.se

Niklas Roeber
Otto von Guericke University of Magdeburg, Germany
niklas@x3t.net

Local Organization

Local Arrangements	Yvonne Bäro bro@idmt.fraunhofer.de
Promotion, Proceedings	Henning Köhler koehler@idmt.fraunhofer.de
Registration	Philipp Meyer meyepp@idmt.fraunhofer.de

Contents

Ingerl, Andreas; Döring, Nicola
Visualization of Music – Reception and Semiotics 9

Tuuri, Kai; Mustonen, Manne-Sakari; Pirhonen, Antti
Same sound – Different meanings: A Novel Scheme for Modes of Listening 13

Steele, Daniel; Chon, Song Hui
A Perceptual Study of Sound Annoyance 19

Cunningham, Stuart; Davies, Gareth; Grout, Vic
Visual Stimulus for Aural Pleasure 25

Blaszczak, Dorota
Practical Aspects of Audio Design for the Virtual Environments 31

Raffaseder, Hannes
SoundTableTennis – an interactive soundinstallation 37

Waisman, Nina
The body as transducer: building narratives through gestural and sonic interactivity 40

Raffaseder, Hannes; Hörbst, Kurt
staTdT_kunst – an intermedia performance 45

Seznec, Yann
The Wii Loop Machine 48

Hoeberechts, Maia; Demopoulos, Ryan; Katchabaw, Michael J.
A Flexible Music Composition Engine 52

Dittmar, Christian; Dressler, Karin; Rosenbauer, Katja
A Toolbox for Automatic Transcription of Polyphonic Music 58

Katchabaw, Michael J.; Demopoulos, Ryan
MUSIDO: A Framework for Musical Data Organization to
Support Automatic Music Composition 66

Leitich, Stefan; Toth, Markus
Public Dj – Music selection in public spaces as multiplayer game 72

Farnell, Andy James
Procudural game audio using Pure Data 76

Reynolds, Gordon; Barry, Dan; Burke, Ted; Coyle, Eugene
Towards a Personal Automatic Music Playlist Generation Algorithm:
The Need for Contextual Information 84

Purwins, Hendrik; Martinez Hernandez, Elena; Adiloglu, Kamil; Annies, Robert; Obermayer, Klaus
Representation of Everyday Sounds for Classification of their Perceived Function 90

Gasser, Martin; Pampalk, Elias; Tomitsch, Martin
A content-based user-feedback driven Playlist Generator
and its evaluation in a real-world scenario 96

Kukharchik, P.; Kheidorov, I.; Martynov, D.
Indexing and retrieval scheme for content-based search in audio databases 100

Reiter, Ulrich
TANGA – an Interactive Object-Based Real Time Audio Engine 104

Uhle, Christian; Walther, Andreas; Ivertowski, Michael
Blind One-to-N Upmixing 110

Gräfe, Andreas; Dausel, Martin; Franck, Andreas
Wave Field Synthesis for Games using the OpenAL Interface.. 116

Schimmel, Jiri; Kouril, Jiri
Utilization of Radio-Location Methods for XY-Coordinate Tracking of Audio Source........... 121

Floros, Andreas; Tsakostas, Christos; Deliyiannis, Yiannis
Binaural Rendering for Enhanced 3D Audio Perception 125

Papworth, Nigel; Liljedahl, Mats; Lindberg, Stefan
Effective interactive design through quick and flexible prototyping 129

Henze, Niels; Heuten, Wilko; Boll, Susanne; Klante, Palle
AuditoryPong – Playing PONG in the Dark 134

Berndt, Axel; Hartmann, Knut
Strategies for Narrative and Adaptive Game Scoring.. 148

Ekman, Inger
Sound-based Gaming for Sighted Audiences –
Experiences from a Mobile Multiplayer Location Aware Game 148

Serafin, Stefania; Nordahl, Rolf
Sonic interaction design: case studies from the Medialogy education 154

Hartmann, Katrin; Buggy, Tom
Topical Segmentation of Audio for Tertiary Courseware 160

Röber, Niklas; Huber, Cornelius; Hartmann, Knut; Masuch, Maic Masuch
Evolution of Interactive Audiobooks.. 166

Hug, Daniel
Game Sound Education at HGKZ –
Between Research Laboratory and Experimental Education 168

Holm, Jukka; Aaltonen, Antti
Associating graphical objects with Musical Tempo... 174

Special Presentations within Audio Mostly 2007 180

Visualization of Music – Reception and Semiotics

Andreas Ingerl, Institut für Medien- und Kommunikationswissenschaft, TU Ilmenau, andreas.ingerl@tu-ilmenau.de
Nicola Döring, Institut für Medien- und Kommunikationswissenschaft, TU Ilmenau, nicola.doering@tu-ilmenau.de

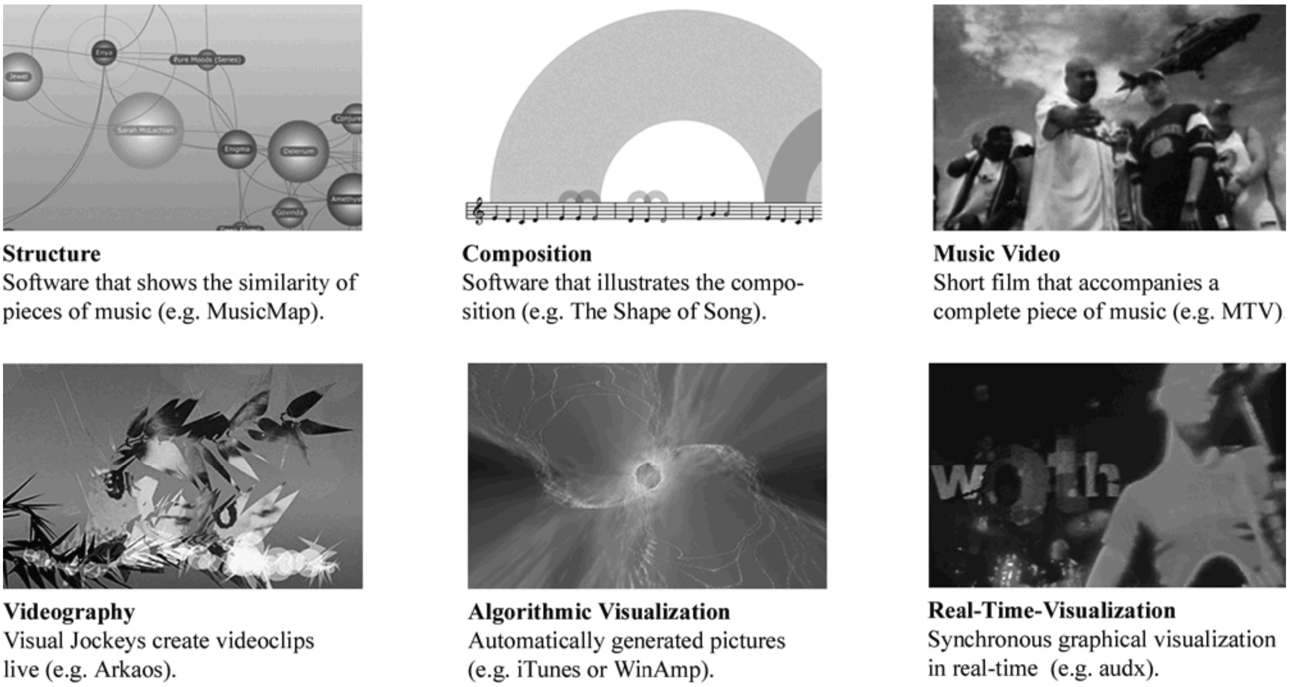
Abstract. Music is primarily auditory but also includes visual elements such as: notation or mental images while hearing music. Music television like MTV or the visualization of music by visual jockeys (VJs) are further examples of combining music with images. However, not much is known about the interaction between auditory and visual representation and perception [1]. How can graphic visualizations intensify or weaken the impression of music concerning entertainment or advancing a better musical comprehension? Are these visuals able to generate a mental model in the viewer's mind, which corresponds to the attributes of the musical source [2]?

1. Introduction

The idea of visualizing music goes back into human history. A relationship between patterns of sounds and patterns of colors has been known since ancient times. For example, Aristoteles picked up this relationship from China and India in his own work. Later, Leonardo da Vinci advanced the work of Aristoteles. Isaac Newton attached the seven-tone scale to the colors of his prism. The composers Franz Liszt and Max Reger created compositions connected with paintings. The best-known creation of "programme music" is Modest Mussorgskij's "Pictures at an Exhibition" from 1874. Alexander Skriabin developed the Color Piano that led to the Color Light Music. Arnold Schönberg also added patterns of color into his compositions [3]. In the early 20th century Oskar Fischinger created several visualizations. He was 'cinema's Kandinsky', an animator who, beginning in the 1920's in Germany, created exquisite "visual music" using geometric patterns and shapes choreographed tightly to classical music and jazz [4].

In 1987 the music television channel MTV went on the air. In those days the music television was the most modern form of music visualization [5]. New possibilities for visualizing music were established around the millennium. During this time of high-potential innovations and Internet hype, many software-products were created for VJs. In addition to classic light- and laser-visualization, computer generated visuals were now available for a mass of people.

In 2001 a method for real-time visualization of music called "audx" was created and patented [6]. This software was the first that basically transformed MIDI (musical instrument digital interface) triggered musical impulses into visual images. With this software it is possible to transform for example the drums of a piece of music directly into visual pictures. This creates a very



Structure
Software that shows the similarity of pieces of music (e.g. MusicMap).

Composition
Software that illustrates the composition (e.g. The Shape of Song).

Music Video
Short film that accompanies a complete piece of music (e.g. MTV)

Videography
Visual Jockeys create videoclips live (e.g. Arkaos).

Algorithmic Visualization
Automatically generated pictures (e.g. iTunes or WinAmp).

Real-Time-Visualization
Synchronous graphical visualization in real-time (e.g. audx).

Figure 1: Different forms of music visualization

intense visual impression of the auditory beat. The effects of audx on the listener/observer remain to be scientifically investigated.

Today a big community of Visual Jockeys (VJs) exists all over the world. Internet communities like vjcentral.com or vjnews.de show the most important and latest trends in this special sub culture of the music scene.

2. Exposition

2.1. Forms of Music Visualization

There are many different forms of music visualization (see figure 1). Firstly, music visualization can be differentiated by form or function. The main forms are: "entertainment" (artistic visualization), "musical similarities" (showing the relation or similarity of several pieces of music) and "visualization of structure" (illustration of the composition) [7]. These forms deal with different content: the form of artistic visualization is the entertaining presentation of changing figures or graphics accompanying a piece of music. Analysing the structure of music connects different pieces of music by genre or style. So music can become comparable by showing the musical similarities. For example the software "MusicMap" possesses such options. For the visualization of structure the software "The Shape of Song" shows the composition of a piece of music. This possibility can be used in music education for example.

On the other hand Rumi Hiraga and Noriyuki Matsuda structure music visualization into: "augmented score" and "performance visualization" [8]. These aspects are quite similar to the entertainment visualization and the visualization of structure of Juan C. Dürsteler. Mainly they wants to understand, analyze, and compare performances and their musical structures using their own prototype system.

After all the best-known form of images accompanying music are music videos, like shown in music television. But a still growing culture of Visual Jockeys or media players like WinAmp or iTunes offer new forms of visualizing music. The entertaining and artistic visualization is the main interest of further research. For understanding this kind of visualization a specific classification is needed: "Videography", for example Visual Jockey (VJ) who create visuals live to the music, "Algorithmic Visualization" like Apple iTunes or by using an oscilloscope and the "Real-Time-Visualization" of audio-events, for example by audx [6]. These three appendages of music visualization are selected for the further work, especially the Real-Time-Visualization.

The visualization system audx uses real-time MIDI-impulses, which represent the character of music. For example, several elements of drums create a specific signal that is converted into a graphical element. Every auditory beat creates a corresponding visual "beat". So the synchronism between music and visualization becomes very important. For example the drums of a band playing live music is triggered by MIDI impulses. Specific parts, like bass drum, snare, or hi-hat, send an impulse every time they are played and start a defined animation. The bass drum creates for example a growing circle that increases its size in a short period of time. So the beat is not auditory any more it also becomes visual. The control by the music is the main input device. The human just creates additional visual elements in an artistic way.

An overview and examples for the different forms of music visualization are shown in figure 1. But after dealing with all that different kinds of music visualization it is important to distinguish between musical graphics and the presentation of musical staves, for example musical notation [9]. Music visualization is an artistic form of illustrating music, as an additional part of perceiving and representing music.

Music visualization technique can be defined by two parameters: The form of visualization and the kind of control. The three forms "Videography", "Real-Time-Visualization", and "Algorithmic Visualization" are the most important for further work. The control can be done directly by the software, by the music input (e.g. MIDI), or by a human (VJ). Like shown in Figure 2 audx is mostly designed to be controlled by the music or a human and visualize music in real-time. Also other input-controls are possible. The software itself is just the managing- and output-tool for the visuals. Other software products have different settings. For example the algorithmic visualization is mainly controlled by the music input and the software, not or less by the human himself.

Form:	**Control:** Software	**Control:** Human	**Control:** Music
Videography	--	possible	possible
Real-Time	--	audx	audx
Algorithms	--	possible	possible

Figure 2: Visualization-Matrix of Control and Form

The shown classification will be verified and enhanced in future scientific research.

2.2. Synesthesia and Semiotics

A phenomenon closely connected with music visualization is the neurologically based phenomenon synesthesia. This means the transmission of sensory pictures or attributes from one sense to another [10]. Some people "see" colors or shapes associated with sound. For them music is not just an auditory experience. Alexander Skriabin described that every key is connected with a special color he could perceive when hearing the notes. However, there is not much scientific knowledge about synesthesia [11]. Many findings are introspections of synesthesists themselves and it's complex to create findings that are not adulterated and comparable. Further, the reception of different synesthesists can differ [12]. Unfortunately it is not established yet how many people actually are able to "see music". Synesthesia seems not to be a universal phenomenon neither can it be applied to everyone. So what must be done to create a universal visual language of music? First steps must be done to analyze if a semiotic of music may exist.

Semiotics is study of signs, sign systems, and processes of signs. There are three important parts of semiotics: The syntax that describes the relation between signs, the semantic that is the relation between the sign und its meaning and the pragmatic that addresses the relation between the sign und its user and situation. First of all the relation between sign and music will be important for music visualization. But also the possible meaning of a sign, in this case the corresponding tone, beat or part of the music. Does the listener/viewer "understand" the sign as a representation of the music? This is also linked to the user himself and the situation the visualization is presented in.

In his theory of signs Rudi Keller describes three different kinds of signs: iconic signs where content and expression is the same, for example a portrait, indicative signs where content and expression possess a natural relation, for example a signature and last symbolic signs where content and expression are arbitrary [13]. Indicative signs seem to be the most important form for visualization of music, but also symbolic signs must be used if users are able to learn the relation between the music and the signs.

The newly established "Bildwissenschaft" ("visual media science") deals with visual information concerning the creation, processing, duplicating and reception of pictures, especially computer graphics and image processing [14].

So one of the main questions is, which part of the music can be visualized with which kind of sign? Which form of relation between tones and signs is already known and which can be learned? Experiments should clarify the mechanisms and rules of tone-sign relations.

2.3. Research Questions and Methods

Today not much is known about the influence of visual perception on auditory perception and information processing [1]. There is much more scientific work on speech and music than on pictures and music [15]. The research field of musical reception is mostly treated by music or cultural studies, by social or psychological studies and by media and communication studies [16]. From a practical and technological point of view media designers are also concerned with the visualization of music. Many diploma theses and class papers are currently prepared at several design universities.

Visual perception doesn't work phenomenal. The viewers construct their own reality [17]. But also reality could be constructed if music and visualization correspond and increase for example the emotional impact of music. A visual stimulus can be added to an auditory stimulus and both will be mixed in reception and interpretation. The utility and reception of music visualization (especially the Real-Time-Visualization) in different user-groups is evaluated socio-scientific in a research project. Hans Neuhoff writes that many media-reception theories also work with music reception research, for example the Agenda-setting theory that describes the relation between mass media and public opinion. Transferred into music visualization research this theory may prove the relation between two pieces of music, with and without music visualization, concerning its publicity. For example today every new song needs to have a music video to be successful. Further the psychological theory of priming could be used. This means that the continuous repetition of specific stimuli increase the effectiveness of reception. Therefore conditioned visual stimuli can perhaps increase the reception of music especially when visualizing important parts of music, for example the beat. Hans Neuhoff also describes the Social Cognitive Theory that purchases with the adoption of new behaviors. This could be a theory to proof the possibility of learning the created forms of music visualization [18].

Music television, media players or perhaps also VJ shows have become part of our society. But are these forms of music visualization well-established and how are they perceived? If the different user-groups are familiar with various forms of visualization or using them in their every day life is examined by oral and written surveys. Basic impact dimensions (e.g. visualization as supporting or disturbing auditory perception) and the effects of single graphical elements are proved in laboratory experiments using the audx-system (e.g. the systematic illustration of tone and pitch in visual forms and colors, like a high tone pitch is connected with bright color and a deep tone pitch with dark color) [19]. Finally, based on the audx-system, the effects of two-dimensional visuals (e.g. computer screen) and three-dimensional visuals (immersive virtual ambiance) are compared.

3. Conclusion

Visualizations of music often aim at generating a mental model in the viewer's mind, which accords to the attributes of the source [2]. In this case the music is the source and the aim is to represent the source using visual and design methods. The specific sources can be: theoretic impulses (Algorithmic Visualization), artificial or artwork impulses (Videography) and music controlled impulses (Real-Time-Visualization).

Different qualities of the visualization of music can be differentiated: Is it expressive, effective and suitable [20]? Expressivity means the direct playback, efficiency the optimal and meaningful usage of visual presentation and suitability the technical effort to generate the elements. When visualizing music these three factors must be analyzed and transformed in real-time without any delay, for example the impulses of drums.

The presentation gives a systematic overview of the latest methods of music visualization and introduces the patented real time visualization with audx. Empirical research questions on the uses and effects of audx are discussed. A first model of a "semiotics of music visualization" is presented and an experimental research design is introduced. The experimental design follows up on the work of Robert Melara. It evaluates the interaction between tone pitch and color. A high tone pitch is connected with bright color and a deep tone pitch with dark color [19].

References

[1] Fischer, H., Entwicklung der visuellen Wahrnehmung, Weinheim, Beltz Psychologie-Verlags-Union (1995)
[2] Robertson P. K., A Methodology for Choosing Data Representations, IEEE Computer Graphics & Application, Vol. 11, (1991)
[3] Dobretzberger F., Geschichtliche Hintergründe der Farbmusik, http://www.planetware.de/colormusic/History.html, (2000)
[4] Fischinger, O., Ten Films, Los Angeles, Center for Visual Music CVM, (2006)
[5] Hausheer C. & Schönholzer A., Visueller Sound, Musikvideos zwischen Avantgarde und Populärkunst. Luzern, Zyklop, (1994)
[6] Ingerl A. & Dringenberg R., Entwicklung von Verfahren und Systemen zur Echtzeitvisualisierung von Audio-Events. Patent-Nr.: DE10214431.1-09, (2001)

[7] Dürsteler J. C.,Visualising Music, http://www.infovis.net/printMag.php?num=161&lang=2, (2005)
[8] Hiraga R., Visualization of Music Performance as an Aid to Listener's Comprehension, Gallipoli, ACM Conference Paper (2004)
[9] Riethmüller A., Stockhausens Diagramm zu Inori. IN: Schmierer, E. et. al., Töne - Farben - Formen: über Musik und die bildenden Künste, Laaber, Laaber-Verlag, (1998).
[10] Marks, L. E., The unity of the sense: Interrelations among the modalities, New York, Academic Press, (1978)
[11] Peacock, K., Synesthetic Perception: Alexander Sriabin's Color Hearing, Music Perception, 2(4), (1985)
[12] Bruhn, H., Musikpsychologie: ein Handbuch, Reinbek, Rowohlt, (1997)
[13] Keller, R., Zeichentheorie: zu einer Theorie semiotischen Wissens, Tübingen, Francke, (1995)
[14] Schreiber, P., Was ist Bildwissenschaft? Versuch einer Standort- und Inhaltsbestimmung, http://www.bildwissenschaft.org/VIB/miStart/gpositionen/bildwissenschaft2.pdf, (2003)
[15] Schmierer, E., Töne - Farben - Formen: über Musik und die bildenden Künste, Laaber, Laaber-Verlag, (1998)
[16] Schramm, H., "Musik und Medien" als Gegenstand medien- und kommunikationswissenschaftlicher Forschung. Eine Einordnung des Forschungsfeldes und der Beiträge des Themenheftes. Medien & Kommunikation, Sonderband "Musik & Medien", Hamburg, Hans-Bedrow-Institut, (2007)
[17] Hoffman, D. D., Visuelle Intelligenz : wie die Welt im Kopf entsteht, Stuttgart, Klett-Cotta (2001)
[18] Neuhoff, H., Zwischen Manipulationsverdacht und Autonomieproposition. Medienbasierte Musikrezeption im Lichte klassischer und moderner Wirkungstheorien, Medien & Kommunikation, Sonderband "Musik & Medien", Hamburg, Hans-Bedrow-Institut, (2007)
[19] Melara R., Dimensional Interaction Between Color und Pitch. Journal of Experimental Psychology, 15(1), Washington, DC, American Psychological Association, (1989)
[20] Schumann H. & Müller W., Visualisierung, Grundlagen und Methoden, Berlin, Springer, (2000)

Same sound – Different meanings: A Novel Scheme for Modes of Listening

Kai Tuuri, Manne-Sakari Mustonen, Antti Pirhonen
Department of Computer Science and Information Systems,
P.o. Box 35, FI-40014 University of Jyväskylä, Finland
{krtuuri, msmuston, pianta}@cc.jyu.fi

Abstract. This paper is grounded on the *multimodal listening* hypothesis, which suggests that listening is a multi-focused process based on multiple distinct, environmentally shaped activating systems and listening strategies. Different modes of listening can operate concurrently complementing each other with different perceptual perspectives. In sound design, the potential of this hypothesis lies in its ability to account for the heterogeneity of different, even contradictory levels of interpretations, meanings and emotions evoked by the same listening experience. In this paper the theoretical basis of listening modes is further analysed and reflected upon in the context of sound design. We propose a comprehensive scheme of eight modes of listening (reflexive, connotative, causal, empathetic, functional, semantic, critical, and reduced) accompanied by examples of their significance in sound design for user-interfaces.

1. Introduction

In contrast to hearing, listening is an active process that provides a means to pick out information for our needs from the auditory environment. It is usually associated with voluntary attention and focusing on something. At present, a reasonable amount of studies exist concerning the area of auditory perception [see 1]. However, psychoacoustic models in which perception is built up from low-level perceptual atoms are inadequate for understanding creation of meanings. In most cases we do not perceive sounds as abstract qualities; rather, we denote sound sources and events taking place in a particular environment (e.g. dog barking) or we concentrate on some other level of information. Apart from a few examples in previous literature, modes of listening have received surprisingly little attention.

Listening is highly multimodal activity in nature. Multimodality of listening means that there are several distinct strategies to listen. It is a distinction of listening strategies and perceived experiences, not a distinction between sounds - although some sounds encourage the use of certain modes more strongly than others. Each mode of listening considers its own source of information in the auditory stimuli either with or without its context. The same sound can essentially be listened to with different kinds of attention and with different outcomes. Despite their separate nature, modes can and often do operate concurrently complementing and influencing each other.

How many ways can we listen to the same single sound? The process of listening to speech provides an example of a variety of perceived meanings. In addition to the conventional (linguistic) meaning of speech we can focus on listening to a speaker as a sound source (e.g. gender, age, dialect, emotional state). Or we can concentrate on the interactive nature of speech in a conversational context (e.g. getting attention, approving, encouraging). We can also pay attention to the qualities of speaker's voice (e.g. timbre, melodic contour, rhythm). Such meanings can even be perceived as contradictory e.g. when non-verbal cues in speech do not match with the verbal content.

The exploration of the multimodality of listening experiences promotes a better understanding of how meanings can be conveyed in effective sound design. This is our primary motivation to study this subject. In the following text previous accounts of the modes of listening are reviewed and then a revised account is proposed and detailed with examples.

2. Previous accounts of listening modes

2.1. Everyday listening

Our everyday listening is not focused on sounds. Instead, we usually hear sound sources: actions and events that cause sounds. We hear footsteps on a sidewalk, a car passing by, breaking of glass etc. We might also try to figure out how far and in what circumstances these events happen as we use listening to outline our environment to support our actions. This source-orientated mode of listening seems to be so effortless that we are not conscious of it. Listening studies by Vanderveer, Ballas and Gaver [2] show evidence that subjects indeed tend to describe a sound by its source or an event that caused it. In the case of ambiguous sounds, confusion and misidentifications are argued to be based on similarities in the mechanical structures of source events (such as "hammering" and "walking"). From the perspective of ecological (gibsonian) perception such confusions relate to shared properties in the affordance structures of various sound events. [3] In such cases, additional contextual information is required to confirm the denotation of the sound source.

The automatic nature of source-orientated listening and the phenomenon of ambiguity is frequently exploited in *Foley-tradition* of sound design [4] by framing non-authentic but believable (i.e. affordable) sounds persuasively into a suitable narrative context. As a part of the craftsmanship of a sound designer is required to identify essential components of sound which can convey a desired narrative effect (e.g. denotation of an event in a fictional environment). The freedom to use non-authentic sound sources gives a designer a much wider range of possibilities to suitably enhance an audience's experience. In contemporary audiovisual narration, even source-visualised on-screen sounds are often produced or treated artificially.

2.2. Reduced and musical listening

Possibly the earliest explicit mention of the modes of listening in previous literature can be found in the work of Pierre Schaeffer [5]. He proposes a distinguished *reduced* mode of listening by which we intentionally divorce the phenomena of sound from any functions as a medium of signification. An objective listening perspective was created to manage and handle sounds as abstract and fixed objects (*objet sonore*) for composition purposes of the *musique concréte* tradition. In

Schaeffer's phenomenology, reduced listening is a mode where the sound (and its qualities) is perceived *per se* resisting any claims about the exterior world. Such a totally abstract and "meaningless" perception of sound is of course a purely theoretical concept, and Schaeffer's work has received criticism as such [6].

Gaver's distinction of *musical listening* as opposed to everyday listening [2] shares essential similarities with the definition of reduced listening. Schaeffer's thoughts highlighted the fact that sounds from the everyday world can be used and listened to musically. The mode of musical listening is not intended to be restricted to sounds defined as music. Gaver thus recognises the "cross-referential" nature of different modes; it is possible to listen everyday auditory environments as music (e.g. attending to pitch contours and rhythm) and conversely possible to listen to musical performances in terms of causality and sound sources (e.g. separating different instruments or stems). Therefore as a term, musical listening can be misleading because music can be listened to in various modes – not just as abstract structures. Reduced listening can be referred to as musical listening only when listening process is concerned with musically determined qualities and structures.

However, the reduced mode of listening was later applied to film sound design by Chion [7]. Here the listening experience was objectified, by voluntarily resisting the natural denotation of a sound source or its meaning. By concentrating on the sound itself, sound designers could "open their ears" to the abstract qualities of sound. In this way, more creative or effective ways to utilise sound in narrative or artistic context became possible. The idea of reduced listening stresses the (analytic) perspective in which it is more important to understand what the sound *sounds* like than how it has been produced (see e.g. Foley tradition). Unlike Schaeffer, Chion puts forward the idea that reduced listening is more of a tool for analytic discovery of sound beyond its evident denotative meaning.

Both Schaeffer's and Chion's views are concerned with *acousmatic* situations [5,6,7] where the actual cause of sound is hidden from listener. This is indeed the case with mechanically reproduced and transmitted sounds e.g. telephone, radio and recordings. Acousmatic sounds thus permit more freedom of imagination for a listener to form a sound-only based perception and allow sounds to be composed artificially often by combining a variety of natural or non-natural sources. Acousmatic situations however do not automatically encourage reduced listening. Chion suggests that they can even intensify the motivation for everyday listening (i.e. causal listening, see 2.3.) when the visual support is removed [7].

2.3. Three listening modes of Chion

In his book Audio-Vision, Michel Chion was the first to introduce a more comprehensive scheme for modes of listening [7]. It consists of *causal*, *semantic* and *reduced* modes (see Figure 1). Two of them, causal listening (i.e. everyday listening) and reduced listening, were discussed above. The third mode, semantic listening, focuses on conventional meanings that the sound might represent by code, language or habit. As causal (everyday) mode of listening refers to ecologically-orientated evident[1] denotations, there was indeed a place for a mode that addresses socio-culturally shaped and learned (symbolic) codes. Spoken language is the most obvious example of cultural convention, but semantic listening can refer to anything which creates a meaning for a sound that is not "literally"[2] there. Examples could range from exact rule-type codes (e.g. Morse code) and natural languages to more passively learned pragmatic habits (e.g. an applause after a performance) or conditioned associations (e.g. an ambulance siren). Many codes are so deep-rooted in the form of dispositions or habits, that they appear almost innate to us.

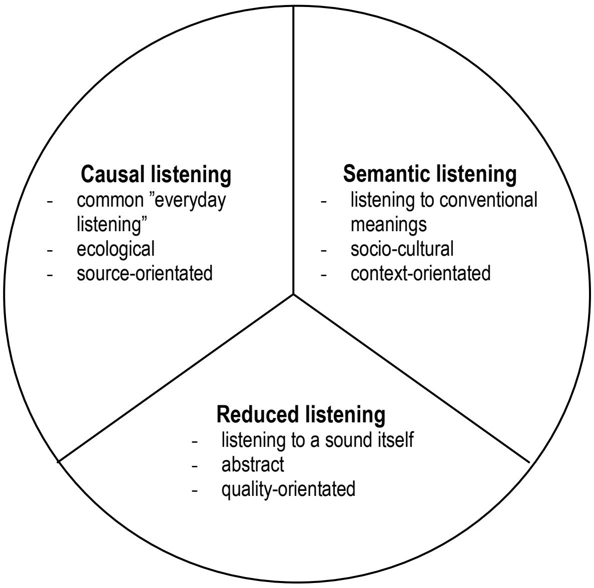


Figure 1. Three listening modes formed on the basis of Chion's classification [7]

Chion's scheme of listening modes is quite well known and it has proven its usefulness for sound designers. It is comprehensive and has broad categories that are easy to understand. However, it fails to capture more refined distinctions between clearly separate modes of listening, which will be covered later in this paper in the form of a revised model. Nevertheless, Chion's scheme forms the basis for our development of a more detailed scheme.

2.4. Activating systems

David Huron has suggested a six-component theory of auditory-evoked emotion. Despite his primary interests in emotions, his theory forms a relevant perspective to the modes of listening. Huron assumes a bio-cultural perspective on emotions as adaptive behaviours expressed within and shaped by an environment. Six *activating systems* are determined (see Table 1). These are evolved to serve specific functions and they all are capable of operating concurrently and evoking various emotional states. [9]

We assume that activating systems are not restricted to evoking emotions only, but in a similar way they can evoke other kind of meanings. Besides the denotative system which is linked to causal listening, the rest of the activating systems concerns novel and complementary perspectives to listening modes.

[1] Evident meanings, as defined by C.S. Peirce, are referred to as *iconic* and *indexical* relations of a sign and the object it refers to [8].

[2] By literal meanings we refer to ecologically inferred denotations, such as "meow" means a cat object.

Table 1. Brief summary of activating systems [9]

Reflexive system	Fast physiological responses
Denotative system	Processes which allow listener to identify sound sources
Connotative system	Processes that allow listener to infer various physical properties and passively learned associations (e.g. from temporal patterns)
Associative system	Arbitrary learned associations
Empathetic system	Allows the listener to perceive cues that signals someone's state of mind (an agent causing the sound)
Critical system	Reflective self-monitoring concerning the verification of perception and the appropriateness of one's responses

2.5. Functional perspective

Every sound which is intentionally used for some purpose has a specific function. When a listener answers a question "what is the purpose of that sound?" she defines the perceived function of the sound that is used in a particular context. We might become aware of the functions e.g. when the sound is perceived as a fire alarm, or when we feel music suitable for relaxing purposes, or when we perceive a sound effect as a transitional cue in audiovisual narration. The functional perspective of sound was explored in previous literature e.g. by Hermann and Ritter [10], and Jørgensen [11] who has examined the functional aspects of game audio. Roman Jakobson's model of communicative functions [12] is also related to this perspective.

Sound as a function can be seen as a pragmatic frame for meaning. The way that sound appears in a functional context affords a certain perspective to the process of interpreting the meaning of sound. The procedural chain of events, actions and causalities in a situation can give an indicative meaning even to a "meaningless" beep.

Although the sound itself can suggest the purpose of its use, perceiving the function of sound requires an awareness of context. The context concerns equally the situational factors as well as the listener's past experience on similar functionalities. The perception of a function is often related to a certain framework of common habits (e.g. habits of non-verbal interaction, conversation, musical performance, audiovisual narration or different user interfaces). The function of sound is particularly important in practices of interactive communication.

Functional semantics of sound can be seen as a distinct level of meaning which indicates and/or promotes a functional purpose of sound. It is indicated by the situational context but can also be indicated by the sound. For an example; let's consider the spoken word "dad", whose "pure" verbal meaning of word is carried by the sound. If we shift our attention to the source of sound we can tell that it is a voice of a child. As we concentrate more on the voice and the way the word was spoken (prosodic qualities of voice), we can guess that a child is demanding attention from her dad. She is definitely not just mentioning the word "dad". The attention-demanding function of this utterance can even be perceived without understanding the verbal content.

3. Hierarchical account of listening modes

Previous accounts of listening modes have been incoherent and limited in their scope Therefore, we have formulated a new scheme. We have applied various relevant perspectives in order to form a more detailed and comprehensive outline of the listening modes. It is intended that this new outline will be utilised by audio designers and tested by audio researchers.

The basis of our pursuit of new categories is Chion's determination of three modes. One of the obvious shortcomings of the original scheme was its inability to consider connotations of sound. For the sake of interactive communication, we believe that perspective of functions of sounds deserves attention. Furthermore, activating systems introduced by Huron [9] offers relevant perspectives for developing a new scheme.

We propose a novel, *hierarchical* scheme of modes of listening (see Table 2) which consists of two pre-conscious modes (*reflexive* and *connotative*), two source-orientated modes (*causal* and *empathetic*), three context-orientated modes (*functional*, *semantic* and *critical*) and a reduced mode. The order of modes implicates their level of cognitive abstraction from low to high.

3.1. Reflexive mode of listening

In reflexive listening the focus is on automatic and fast audio evoked responses. Huron mentions various reflexes [9] including orientating response, startle response, defence reflex and reflexive responses that relates to expectations, habituation, sensory dissonance and attention. These responses resist conscious mediation, so in a strict sense this category cannot be considered as a pure mode of listening as it is impossible to control or focus on automatic reflexes themselves. In any case, reflexive responses represent clearly an important way by which meanings and emotions can be evoked by sounds.

3.2. Connotative mode of listening

In connotative listening, the focus is on early associations that the process of listening pre-consciously evokes. These associations are references made by similarity to past experiences of a listener, prior to any reasoned denotations. A French semiotician Roland Barthes [13] has concluded: "…denotation is not the first meaning, but pretends to be so…". He implies that even though it appears that we can reason a denotative meaning instantly, it is only an illusion because we have already made a number of connotative associations. Denotation can then be defined as a "final", more reasoned connotation. The important thing about connotation to realise is that besides denotative meaning, various connotative meanings can arise from the arsenal of excess associative "building material".

At the most primitive level, connotative processes permits a listener to infer various physical properties of sound. These properties can indicate perceptual information concerning the sound source and environment: size, material, energy, proximity and excitation. Connotations can also be evoked from certain mechanical structures of source events (e.g. temporal patterns that evokes a "galloping-like" meaning) even if the sound source has nothing to do with what it connotates. [9] Besides the physical and ecological environment, associative cues can also relate to arbitrarily learned cultural experiences. For that reason we propose that the connotative mode of listening is not only linked to processes of connotative but also to processes of associative activating system.

Like the reflexive mode, the connotative mode of listening is involuntary in nature. Therefore focussing on connotations can be somewhat established by mentally exploring free associations and voluntarily resisting denotations.

Table 2. Summary of the revised scheme of listening modes with examples

Type:	**Mode:**	**Questions:**	**Example:** *Loud sound of train whistle (in a movie)*
Pre-conscious modes:	**Reflexive**	Did you notice any reflexive responses triggered by sound?	Startle response!! It alarms and grabs an attention.
	Connotative	Can you describe what kind of freely formed associations listening immediately evoked?	Big...strong...lots of power...close proximity ...screeching...air blowing..whistle..scream... ....old steam trains....western movies....
Source-orientated modes:	**Causal**	What could have caused the sound?	It´s a train. *Critical second thought*: the sound comes from the TV.
	Empathetic	Does it feel that sound signals someone's state of mind or intentions?	Whistle sounds feels desolate and angry.
Context-orientated modes:	**Functional**	What was the purpose of the sound? What function does the context indicate?	The driver signals train's departure. *Critical second thought*; sound is used as transitional cue between scenes (just before a visual cut to railway station).
	Semantic	Does the sound seem to represent any symbolic/ conventional meanings?	The whistle represents pain... of a suffering man (by replacing his scream)
	Critical	Was the sound suitable for the situation? Did you understand it correctly?	Ah, no panic. That sound belongs to the movie. It was a cliché but quite effective.
Quality-orientated mode:	**Reduced**	Can you describe the properties of the sound itself as objectively as possible?	Sound is high-pitched and loud. A big contrast against quiet earlier scene.

3.3. Causal mode of listening

In causal listening the focus is on denotation of the source of sound and determination of an event that caused the perceived sound. This mode of listening is derived from the scheme of Chion (see 2.1 and 2.3.). This mode is also directly linked to denotative activating system. Causal listening is often referred to as a mode of common everyday listening.

3.4. Empathetic mode of listening

In empathetic listening the focus is on cues that could signal someone's state of mind. Empathetic mode of listening is thus directly linked to the empathetic activating system [9]. It is closely related to causal listening, in the sense of considering the possibility of a human or animal as a sound source or cause of sound. It is also related to connotative listening in the sense of potential auditory evoked associations (e.g. from intensity or certain rhythmic pattern) which can refer to emotional states, intentions or even communicative functions. For example listener can recognise a sad or nervous voice. On the other hand, listener can perceive e.g. a loud slamming (lots of energy) of a door as a possible expression of anger.

3.5. Functional mode of listening

In functional listening the focus is on the purpose of a sound in its context (see 2.5.). This mode considers the possibility that the sound is used for some specific function, which is pragmatically indicated by a sound in relation to the context. In the domain of non-speech auditory cues, a perceived function can be e.g. attention-demanding, alarming, orientating, approving, prohibiting, marking, prompting, giving a feedback or noticing.

3.6. Semantic mode of listening

In semantic listening the focus is on denoting any conventional meanings that a sound might represent. This is the second mode of listening which is derived from the scheme of Chion (see 2.3.). By semantic listening lower-level meanings are also reorganised for conventional reasoning to take cultural context (habits, codes) into account.

3.7. Critical mode of listening

In critical listening the focus is on the reflective judgement of auditory perception. As a mode of listening, it applies the idea of Huron's [9] critical activating system. Critical listening concerns appropriateness or authenticity of sound in a given context. It also considers the appropriateness of one's responses. That includes judgements of possible misunderstanding, deception, false urgency or generally the need to be concerned with the sound. Additionally, at its highest level, critical judgements can be based on aesthetical dispositions.

3.8. Reduced mode of listening

In reduced listening the focus is on the sound itself and its qualities. This is a third mode of listening which is derived from the scheme of Chion (see 2.2. and 2.3.). The examination of sound phenomena itself requires that a listener is consciously resisting any denotations of a sound source or its meaning. This mode of listening is thus exceptionally voluntary and very likely requires high-level cognitive abstraction.

4. Observations from a case study

For a few empirical observations of modes of listening, we present some examples from a group panel discussions of our earlier case study (see Pirhonen et al. [14] which addresses the development of the design panel methodology). Although the panels were not originally conducted for the study of the listening modes, and during the panel sessions the listening modes were not considered, some examples can be pointed out. In the study, the designing of user interface sounds were studied in a series of group panel discussions. A group of panellists carried out design tasks in an iterative fashion; first for idea generation and then for evaluation of the designed outcome. The target of sound design was the user interface of a physical browsing application [15] used in a bicycle exhibition. Each panel session had different tasks and goals. The examples, relevant to this study, consider a warming-up task from the first session and sound evaluating tasks from the third panel session.

Before the actual design tasks, which was the purpose of our panel sessions, the moderator played some soundscape samples to the panellists and after that panellists discussed what they heard. These soundscape tasks were conducted to "open the ears" of the panellists.

The first soundscape sample was recorded in a bird watch tower during morning hours. The panellists described the environment by the objects causing sounds, such as birds, wind humming in the trees, distant road humming (causal listening) and by connotations and by descriptions such as “trees sound as green as in our summer cottage in the relaxed summer morning, birds sound happy” (connotative and empathetic listening). The sounds of the distant motorway were considered as not suitable to the otherwise relaxed atmosphere (critical listening).

The other listening sample was little walkthrough from an elevator to a silent entrance hall, and from there into a noisy rush hour restaurant. Panellists described the entering from silent room to the noisy restaurant as a defence-reaction evoking event; the sound-mass of the restaurant was described as angry, scary and stressing rush-noise (reflexive, connotative and empathetic listening). The noisy environment was also described as unapproachable and unpleasant (critical listening). The moderator told the panellists before the sample that in the beginning there is a bit noise due to the recording technique. That was actually not true, the recording was clean. We observed that this additional task orientation of considering some inappropriate sounds affected the listening experience of the panellists. On this second task they were generally more critical and analytical listeners and described e.g. the humming and clicking sounds (of the elevator) and considered their appropriateness, whether the sound was original or an error of the recording.

After the ear-opening tasks, the group started the actual design tasks for that panel session. First the panellists familiarised themselves with the application by listening to a use scenario in the form of radio-play narration. Actual events-to-be-sonified (successful activating of a physical link, process of loading, loading ready) were clearly indicated in the story. Secondly, candidate sounds were played alone sequentially for each function and the most promising sounds were voted to go through to the next phase. The third phase of the panel session was that the sounds were played within the radio play, so that the panellists obtained a more holistic experience of the use situation, and heard the nominated UI-sounds connected to the procedure of the use scenario.

When the sounds were played alone, they were mostly judged by the criteria of subjective satisfaction and connotations like “this sound is not good, I do not like it, it sounds too lazy” etc. or “this is good, happy sound, I like it” or “it was good, snappy and attention grabbing sound, suits for the function”. Some sounds were voted to go through to the next phases, others were rejected. One example of the rejected sounds was rattling-like sound which was designed to indicate the loading process. When the sound was judged alone, it was considered as irritating clacking of teeth when feeling cold, and one panellist said that it reminded her of an annoying little boy with a ratchet (connotative, causal, empathetic and critical listening). The sound was rejected as too annoying and not suitable for the context.

In the next phase the panellists heard the radio play again, this time with the selected UI-sounds played within the story. Now, as the panellists were more immersed in the functional context and able to experience the whole situation procedurally, some of the earlier judgements changed. Some sounds that were judged as effective in the second phase were no longer considered suitable for the context, and some rejected sounds were asked to be elevated again. A more important factor than the subjective effect of the sounds was the match with sonic functions and events. During this last phase, the rejected rattling sound was now judged as the top-rated sound as an indicator for the loading process. Now the same sound was described as sound of small cogwheels, indicating the function of something happening, rather than the annoying ratchet or teeth chattering. This example indicates how the functional context provides a crucial part of understanding the sound.

5. Listening modes and the current design paradigms of user interface sounds

In the research field of auditory cues in user interfaces (UI) there has been only little discussion concerning the multi-faceted meaning construction described in this paper. In 1994, the workshop report of CHI’94 discusses that “A more central concern was how to effectively convey information with non-speech auditory cues. Sounds can be interpreted at several levels…Current user interfaces have not yet addressed this in deeper expressive level in their use of sounds.” [16]. Despite the early recognition of the problem, it has not been widely considered within the research paradigm of UI-sounds.

In this paper we proposed a novel scheme for the modes of listening, which comprehensively binds together somewhat scattered discussions of earlier literature that concerns the issue. Listening modes play an important role as a tool for sound design. As we can listen to a sound with various forms of attention, the process of sound design must then concern auditory signs from various perceptual perspectives, in order to ensure consistent support for common communicative goals. The worst case design scenario would be that different listening modes evoke contradictory meanings (e.g. function implies an alarm sound, but the major chord invokes positive or happy connotations), or when the sound is experienced as annoying despite its perceived relevance in the context.

To demonstrate the relevance between sound design and perspectives of different listening modes, let’s first examine two seemingly opposite design paradigms of user interface sounds: *earcons* and *auditory icons*. Earcons are defined “…as abstract, synthetic tones that can be used in structured combinations to create sound messages to represent parts of an interface.” [17]. The symbolic relationship between the sound and its meaning is seen beneficial as sounds do not have to correspond to what they represent [18]. Meanings are thus arbitrarily coded and therefore learning of specific codes is required prior to effective understanding. It seems that the philosophy behind current earcon design is related to information theory [19] with implicit assumption of the role of sound as a carrier of coded information. Current earcon design guidelines consider sound by emphasising psychoacoustic phenomenon on how sound may be masked or how sound stream can be segregated (judgements on timbre, register, rhythm, concurrent sounds etc.) [20]. These channel-orientated perspectives and considerations of channel-noise factors (e.g. masking) further emphasises the information theory based view of communication.

The paradigm of auditory icons, conversely, relies on iconicity and the ecological perspective of auditory perception [21, 3]. This essentially means that when listening we naturally pick up recognisable parts from the auditory stimuli. Relations between sound and its meaning are based on similarities with familiar aspects of our everyday environment. They can be denotations of sound sources or partial indicators that point to some mechanical properties of a sound causing event. In sound design, similarities can also be used in a metaphorical way. The most important difference to the earcon-paradigm is that the design of auditory icons is more focused on how the sound itself, by resemblances, motivates the meaning-creation. Just as in the traditional film sound design, meanings appear to be motivated by the sound.

We can conclude that the earcon-paradigm is concentrated mainly on two modal perspectives: semantic mode (extreme requirement of code) and reduced mode (sound is supposed to be heard as musical parameters). The design of alarm sounds additionally concerns reflexive mode of listening. On the other hand the auditory icon paradigm is determined on perspectives of causal mode (source recognising), connotative mode (physical property indicators of an event) and in some sense also functional mode (meanings of sounds shares iconic similarities with the event it represents in application). We thus find that the distinction between earcons and auditory icons is not intended as a distinction between UI-sounds themselves. In fact, that categorisation seems to be more related to which modes of listening are adopted for the paradigm in question. In light of listening modes, earcons and auditory icons are to be considered as *design paradigms* – not as necessary distinct types of UI-sounds. The cross-related nature of listening modes allows for the consideration of different design paradigms in tandem.

Indeed, an optimally designed earcon can also utilise its expression with e.g. iconic and affective levels of meaning with cues to some familiar qualities or habits of the experienced world - even when an abstract form of expression is chosen.

6. Conclusions

We feel that our own main contribution of this paper is the systematic review and synthesis of listening modes, and within the proposed scheme the inclusion of an explicit functional mode of listening. We argue that the purpose of sound in a functional context is an important factor in *interaction design* within user interfaces. Firstly, every user interface design must address the role of a sound in interaction. Secondly, the perceived function of sound in a situational context represents itself as an important class of meanings. A user can get context-derived indications to suitably interpret even a “meaningless” beep, not to mention sounds that convey some additional semantic support for the appropriate interpretation of meaning. We can find that the two classic design paradigms discussed above (earcons & auditory icons) are deterministically more concerned with the semantic aspects of UI-sound element itself – not the aspects of how the sound is used in the functional context of UI. The complementary perspective we propose is more procedural in nature; in this approach it is more important *how* meaning is created in a given context than *what* the meaning is *per se*. Functional semantics of sound is based on tacit reasoning and pragmatically evoked semantics.

The general process of sound design for a user interface, at least implicitly, should begin with exploring the relevant communicative purposes of sound in UI-interaction. The outcome of that meta-design process will be a list of functions of sounds referring to various events and processes taking place when user tasks are performed. By analysing those functions within those scenarios and situations, a designer can find associative ideas for relevant functional semantics for the actual sound design.

The scheme for modes of listening, which is presented in this paper, is intended to open a discussion concerning the topic. Also this study is to be seen as compiling a review of various aspects concerning the complex scheme of meanings inferred from sound. The new perspective, the functional mode of listening, is the most prominent contribution from the perspective of audio interaction design and research. The observations from our case study support our assumptions. Nevertheless, more empirical evidence should be gathered to validate the appropriateness of modes of listening for user interface design. Sound design cannot afford to overlook the diversity of meanings and the affective responses that the sound evokes in the context of its use. As a conceptual model, the proposed scheme of modes of listening can guide the designer to find answers to that challenge.

Acknowledgements

We thank Professor Tuomas Eerola for his support and encouragement for our work. Many thanks to Emma Murphy for her comments and help in writing English.

This work is funded by Finnish Funding Agency for Technology and Innovation, and the following partners: Nokia Ltd., GE Healthcare Finland Ltd., Sunit Ltd., Suunto Ltd., and Tampere City Council.

References

[1] Bregman, A. Auditory Scene Analysis. Cambridge, MA: MIT Press (1990)

[2] Gaver, W. The SonicFinder: An Interface That Uses Auditory Icons. Human-Computer Interaction 4, 1 (1989), 67-94

[3] Casey, M. Auditory Group Theory: with Applications to Statistical Basis Methods for Structured Audio, Ph.D. Thesis, MIT Media Lab (1998)

[4] Mott, R. Sound Effects: radio, TV, and film. Boston, MA: Focal press (1990)

[5] Schaeffer, P. Traité des objets musicaux. Paris: Editions du Seuil (1968)

[6] Kane, B. L’Objet Sonore Maintenant: Pierre Schaeffer, sound objects and the phenomenological reduction. Organised Sound 12, 1 (2007), 15–24

[7] Chion, M. Audio-Vision: sound on screen. New York, NY: Columbia University press (1994)

[8] Peirce, C.S. What is a sign? In Essential Peirce: Selected philosophical writings vol. 2. Bloomington, IN: Indiana University Press (1998), 4-10

[9] Huron, D. A six-component theory of auditory-evoked emotion. In proceedings of ICMPC7 (2002), 673-676

[10] Hermann, T & Ritter, H. Sound and Meaning in Auditory Data Display. In proceedings of the IEEE, vol. 92, 4 (2004), 730-741

[11] Jørgensen, K. On the Functional Aspects of Computer Game Audio. In proceedings of Audio Mostly 2006, 48-52

[12] Jakobson, R. Closing Statements: Linguistics and Poetics. In Sebeok, T. A. (ed.) Style In Language, Cambridge, MA: MIT Press (1960), 350-377

[13] Barthes, R.. S/Z. New York: Hill & Wang (1974)

[14] Pirhonen A, Tuuri, K., Mustonen, M-S. & Murphy, E. Beyond Clicks and Beeps: In Pursuit of an Effective Sound Design Methodology. In proceedings of HAID2007 (in press)

[15] Välkkynen, P. Hovering: Visualising RFID hyperlinks in a mobile phone. In proceedings of MIRW 2006, 27–29

[16] Arons, B. & Mynatt, E. The future of speech and audio in the interface. In SIGCHI Bulletin 26, 4 (1994), 44-48

[17] Brewster, S, Wright, P. & Edwards, A. A detailed investigation into the effectiveness of earcons. In Kramer, G. (ed.) Auditory display. Reading, MA: Addison-Wesley (1994), 471-498

[18] Blattner, M, Papp, A. & Glinert, E. Sonic Enhancement of Two-Dimensional Graphic Displays. In Kramer, G. (ed.) Auditory display. Reading, MA: Addison-Wesley (1994), 447-470.

[19] Shannon, C. & Weaver, W. The Mathematical Theory of Communication. Urbana, ILL: University of Illinois Press (1949)

[20] Brewster, S., Wright, P & Edwards, A. Experimentally derived guidelines for the creation of earcons. In Adjunct Proceedings of HCI’95, 155-159

[21] Gaver, W. Auditory Icons: Using sound in computer interfaces. Human-Computer Interaction, 2. (1986), 167-177

A Perceptual Study of Sound Annoyance

Daniel L. Steele
CCRMA
Stanford University
Stanford, CA, USA
dlsteele@ccrma.stanford.edu

Song Hui Chon
CCRMA / Electrical Engineering
Stanford University
Stanford, CA, USA
shchon@ccrma.stanford.edu

Abstract. In this paper we study the relative percept of annoyance generated by synthetic auditory stimuli. The factors influencing noise-induced annoyance remain somewhat vaguely defined in perception studies, even though such annoyance is a common and pervasive problem in contemporary life – one that bears significant health and financial/societal burdens. We describe an experiment wherein we rank perceived annoyance using a stimuli set of 24 synthetic, one-second sounds on sixteen subjects. Six types of audio samples were each presented at four different SPLs – 50, 60, 70 and 80 dB. Subjects were presented with pairs of test tones and asked to answer which of the stimuli was more annoying, using a two-interval, two-alternative forced choice (2I-2AFC) protocol. Despite the fact that subjects were not prompted with a definition of annoyance, responses were, in general, consistent. Results predictably showed a correlation between perceived loudness and annoyance in this range of presentation level; however, even though loudness was one of the principal determinants of annoyance, it became clear that there are other, more subtle factors at work, evident in the changes of perceived annoyance ranking at different presentation levels. It is also significant that each of the subjects' responses was either highly consistent with their final result or contributory to the preservation of a no-preference option, even though the decisions were two-forced-choice.

1. Introduction

Annoyance is a common and universal sensation that most people experience everyday. The definition of annoyance may differ from one person to another, but this paper aims at showing common characteristics in sounds that may determine the level of annoyance in a normal-hearing person. The authors propose a novel method for the collection and analysis of annoyance data.

It is generally understood that spectral power is directly correlated to annoyance [12], although the degree of correlation has been under investigation [15]. Previous studies have assumed that noise level is a strict gauge for determining annoyance [10], but it is reasonable to suggest that the influence of the sound level might be more subtle; annoyance is, after all, only a subjective and highly personal characteristic. One study reveals that subjects perceive annoyance differently based on their ability to influence it [9]. Another demonstrates that age plays a significant role in annoyance judgments [1]. In [4], a number of hypotheses were examined with respect to annoyance, but it failed to obtain a conclusive evidence for any. We claim in this paper that loudness defines the general trend for annoyance, but there are many factors that can alter that judgment significantly.

1.1. Addressing Annoyance

Annoyance plays a large role in political, financial, and technological situations. It is a common occurrence for property values to decrease with increasing proximity to airports, highways, and other sources deemed annoying. Physical acousticians have various measurement standards for addressing this, such as dB60 and dB(half-day), where the time integral values of the sound power levels are taken. These measurements average sounds, which is useful for sound fields that tend to have many high peaks [2]. Also, it is often understood that work productivity decreases with increasing distraction. An understanding of this principle led to a workplace revolution on the corporate level with the help of companies like Muzak. The end goal was to input sounds into the workplace that would desensitize workers from the usual office distractions [16]. In fact, the change in work productivity is so apparent that some researchers use time-elapsed measurements under various sound conditions to measure how annoyed some subjects are [3]. A classic example of annoyance perception involved choosing a telephone ringtone. In an interview conducted with Jean-Claude Risset [20], he conveyed a story of some of the early research at Bell Labs; in early tests, subjects were found to be reluctant to answer their telephone because they found the ring quite pleasant and they did not want to interrupt it. As a result, people were missing important communications. Significant effort had to be made to determine that a dissonant bell tone in an on-off-on-off pattern was effective in warning people of an incoming communication.

2. Experiment

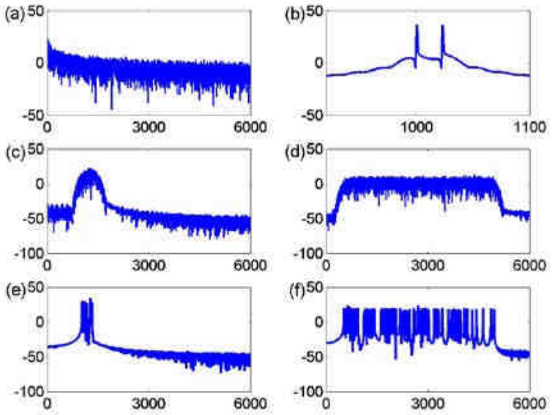


Figure 1: Frequency spectrum representations of the six stimuli at 50 dB SPL. (a) pink noise (PN), (b) two-sinusoids (TS), (c) narrowband noise (NBN), (d) broadband noise (BBN), (e) narrowband tone cluster (NTC), (f) broadband tone cluster (BTC)

All tests were performed in a quiet classroom at the Knoll, home of Stanford's Center for Computer Research in Music and Acoustics (CCRMA). Subjects agreed to respond to a questionnaire and provide feedback, which proved useful in data analysis and for future work considerations.

2.1. Stimuli

The experiment made use of six different kinds of stimuli generated in MATLAB at four intensity levels – 50, 60, 70 and 80 dB SPL, over a frequency range 500 to 5000 Hz.

The six stimulus groups are (1) pink noise (PN), (2) two sinusoids (TS) at 1000 and 1023 Hz with equal amplitudes, (3) narrowband noise (NBN), generated by filtering Gaussian-white noise with a bandpass filter with the passband of 1000 and 1500 Hz, (4) broadband noise (BBN), generated similarly, with the difference of the bandpass filter passing the frequencies between 500 and 5000 Hz, (5) narrowband tone complex (NTC) consisting of ten linearly distributed sinusoids of random amplitudes between 1000 and 1500 Hz and (6) broadband tone complex (BTC), consisting of 100 sinusoids of equal amplitudes whose frequencies are logarithmically distributed (so that there would be an equal number of tones per octave) between 500 and 5000 Hz.

We originally considered two more stimuli – a pure tone of 800 Hz and a pink noise combined with two NBNs (with center frequencies at 1000 and 1023 Hz). After some investigation, those stimuli were removed since there is ample research involving pure tones (e.g. [5][6][14]), and the modified pink noise produced results very similar to those of the BBN. Figure 1 shows the spectrum of six stimuli at 50 dB.

2.2. Procedure

Each subject was asked to fill out a questionnaire concerning potentially relevant distinctions before the experiment. The subjects were asked for their gender, age, ethnic and language background, musical training, and whether they identified as an introvert or extrovert. All 276 unique pairs of stimuli were presented (corresponding to 24 choose 2), each stimulus lasting 1000 msec plus 50 msec of ramp-up and ramp-down time to avoid clicks, with a 500 msec pause between subsequent stimuli. Each subject was asked to select which of the two stimuli was more annoying according to their judgment (the authors did not define the term 'annoyance' to the subjects). The experiment did not allow the subjects to respond "both stimuli are equally annoying", even if it may have applied in some cases.

2.2.1. The Forced-Choice Decision

In our research, we found two methods for measuring annoyance. The first such measurement is to set up a task and see how the subject's productivity changes over varying conditions [3]. This test is very good for getting data that the subject does not influence, because the subject is not answering questions based on their perception. The time changes can be put on an absolute scale and comparing values across tests is easy. Unfortunately, this test might not target annoyance specifically. For example, pleasantness can also slow down the progress of a task and, therefore, sounds that are pleasant as well as sounds that are annoying will produce time changes that lie in the same data range. Separating results like this would be difficult graphically and semantically.

Another method common to annoyance testing is the attribution of a scale. Various scales we have encountered include the measure of annoyance from 0 to 100. Subjects are prompted for an annoyance value and their results are averaged with that of others. Above 72% average is percentage of highly annoyed (%HA), 50, annoyed (%A), 28 a little annoyed (%LA) [13]. The disadvantages of using a scale, however, far outweigh the benefits. First, the response is not done using forced-choice. In other words, the subject is prompted for a subjective result and therefore has no mechanism for ensuring accuracy or minimizing competitiveness. Also, scales that are too fine in detail lose meaning in their inner regions. How does one distinguish something that is 61% annoying from something that is 62% annoying?

In order to explore the options for reducing the risks involved in the methods above, we decided on using a ranking system based on a forced-choice procedure. We should point out that the advantages of a forced-choice ranking system include low-error feedback from subjects and the use of a specific task that is easy to convey. Final results should also converge on accurate values after a significant number of iterations. The downside of this ranking system is that the experiment could be asking the subject to perform a task that does not make sense. If there is actually no preference, then the subject could be giving arbitrary responses. Also, the ranking system does not allow the sounds to be put on an absolute scale; the annoyance of one sound can only be compared to that of another. This downside should not affect the theoretical possibilities of the study, though for practical applications, it might need to be addressed.

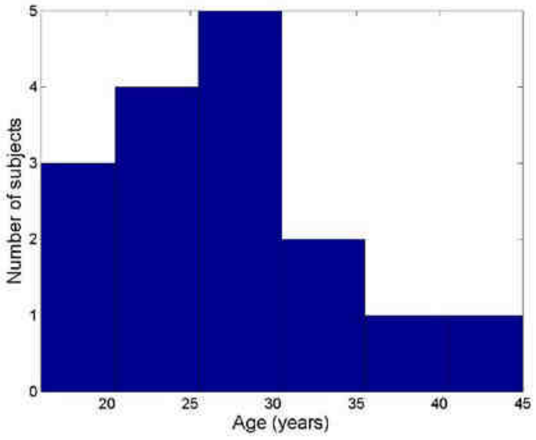


Figure 2: Histogram of sixteen subjects by age.

2.3. Equipment

The experiment was performed monaurally (left ear) on Sony MDR-7506 headphones, coupled with an M-Audio Omni I/O mixer connected to a Linux computer running Matlab, Version 7.3. Before each session, the authors used a sound meter set to measure decibel levels with the A-weighting scheme and calibrate the headphone volume.

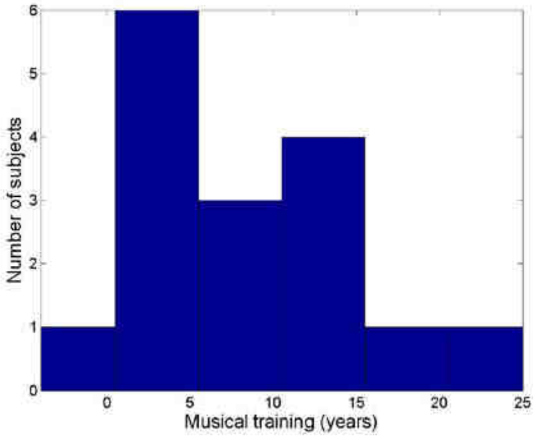


Figure 3: Histogram of sixteen subjects by years of musical training.

2.4. Subjects

Sixteen subjects were recruited from ages 16 to 41; the histogram of age distribution is shown in Figure 2. According to the questionnaire, among the 16 subjects, 7 were male and 9 female; 10 were Caucasian and 6 Asian. Their musical training ranged from no formal training to over 20 years, which is displayed in Figure 3.

2.5. Weighting

There are a few popular weighting schemes used in sound research, such as A-, B-, and C-weighting [17] and ITU-R 468 [18]. The A-weighting is often used for environmental studies, where B- and C-weightings are useful for louder noises such as aircraft turbulence [19]. ITU-R 468 was later proposed to be an appropriate perceptually-based weighting scheme for all types of sound [20], though its use is not widespread in the U.S.A. Since A-weighting has been criticized for being inappropriate for human hearing measurement [7], we were at first concerned about our use of an A-weighted meter instead of ITU-R 468, but the following analysis demonstrates that, at least for this experiment, this difference is irrelevant.

As can be seen from Figure 4, the dB values resulting from using A-weighting and ITU-R 468 can be quite different, especially between 1000 and 10000 Hz. The authors tried to find a sound meter that uses ITU-R 468 weighting, with no success, so we resorted to manually calculating the perceived loudness values using both weightings of all 24 stimuli. The results are shown in Figure 5 and Table 1.

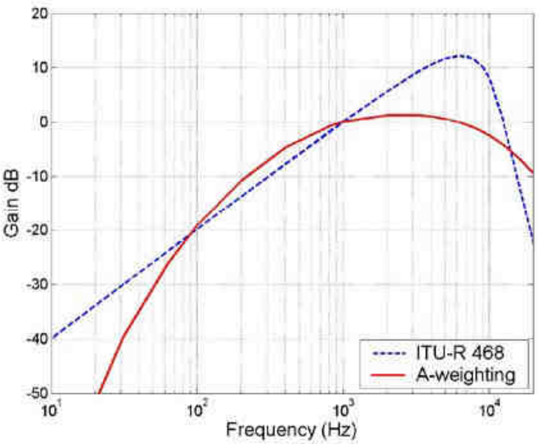


Figure 4: Gain spectrum of two common decibel-weighting schemes.

As we can see from Table 1 and all of parts (a) through (f) of Figure 5, the calculated loudness values can be quite different depending on the weighting methods used. They also show that those differences may come from the spectral content of the stimuli, in other words they are quite small when the bandwidth of the stimuli are narrow (i.e. Figure 5 (b), (c), (e)) and the frequencies lie in the range where the two systems are close in gain, while the differences can be larger with wider-bandwidth stimuli (i.e. Figure 5 (a), (d), (f)).

What are noticeable in Figure 5 are the equal slopes in all cases (dashed and solid lines in each subfigures). The equal slope in each graph (i.e. the relationship between x-axis and y-axis) between successive points is important because this means a 10 dB increase in loudness corresponds to a consistent level of increase in perceived loudness, whether the increase was from 50 to 60 dB or from 70 to 80 dB. Also, the consistent offset between each pair of lines illustrates that even though the theoretically calculated perceived loudness may be different, the difference is consistent and therefore we can ignore this difference (and furthermore, different weighting methods) for the purpose of our study in this paper.

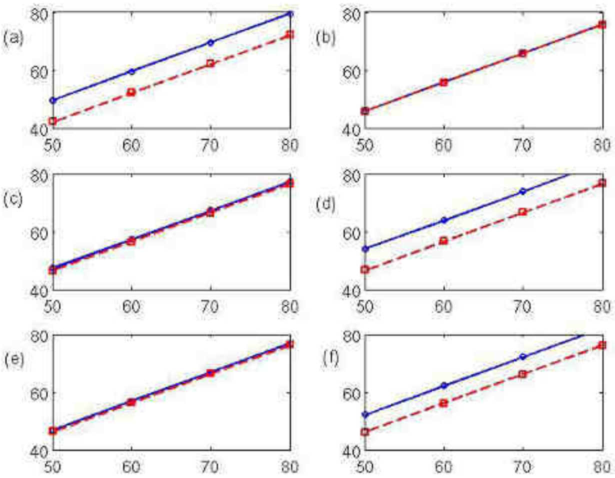


Figure 5: Calculated loudness levels of dB(A) (dashed line) and dB ITU-R 468 (solid line) versus dB SPL per stimulus. (a) PN (b) TS (c) NBN (d) BBN (e) NTC (f) BTC.

Stimulus	Loudness Level (dB)	ITU-R 468 weighted values	A-weighted values
PN	50	49.5450	42.0982
	60	59.5449	52.0983
	70	69.5450	62.0983
	80	79.5450	72.0983
TS	50	45.7669	45.7347
	60	55.7667	55.7346
	70	65.7666	65.7346
	80	75.7666	75.7346
NBN	50	47.3316	46.4027
	60	57.3315	56.4028
	70	67.3315	66.4027
	80	77.3315	76.4027
BBN	50	53.9458	46.5928
	60	63.9460	56.5928
	70	73.9461	66.5929
	80	83.9461	76.5929
NTC	50	46.8315	46.2625
	60	56.8312	56.2623
	70	66.8312	66.2622
	80	76.8312	76.2622
BTC	50	52.1154	46.1958
	60	62.1154	56.1958
	70	72.1154	66.1958
	80	82.1154	76.1958

Table 1: Theoretically calculated perceived loudness values using dB(A) and ITU-R 468 weighting methods for all 24 stimuli

3. Results and Analysis

3.1. Consistency

A primary concern of the experiment is that the question being asked of the subject might not make sense. It is reasonable to assume that if there were some measure of consistency and the subject demonstrated having a high consistency, the question would be a valid one. Fortunately, the nature of the record-keeping in this experiment provides for a built-in consistency measurement. For example, if a subject is asked to rank a pair of stimuli [0, 0], one stimulus will receive one point and one will lose a point, giving an outcome of [-1, 1]. If the same pair is presented at a later time, a consistent subject will make the same decision, thus awarding a second point to the chosen stimulus and subtracting a second from the other, giving an outcome of [-2, 2]. An inconsistent subject will make the opposite decision, thus cancelling the previous decision and giving an outcome of [0, 0]. The sum of the absolute value of each of the elements of the final ranking matrix, called the consistency score, gives an indication of the consistency. In the example above, the consistent subject had a consistency score of 4 while the inconsistent subject had a score of 0. This works for much larger matrices as well.

The Appendix of this paper contains a formal derivation for calculating the maximum consistency score for an arbitrary number of stimuli. Based on this formula, the highest achievable consistency score with 24 stimuli is 288. All of our subjects' scores were between 182 and 274. Because of these high consistency values, we can feel confident in saying that the subjects, in general, understood what was being asked of them and thus, their few conflicting decisions actually contributed to a "no-preference" option. Thus, the test was a quasi-two-interval, three-alternative forced choice (2I-3AFC) one, rather than 2I-2AFC.

3.2. Absolute Annoyance

As it turns out, it is possible to conclude that one type of stimulus is more annoying than another, independent of its loudness. The graph shown in Figure 6 was created by taking the average stimulus ranking over all four of the intensity levels, thus cancelling the loudness effect, and then averaged over all sixteen subjects. The error bars in the graph represent the maximum and minimum of the sixteen subject averages. It is clear that this deviation is rather static, and thus, conclusions can be drawn about the total average annoyance ranking per stimulus.

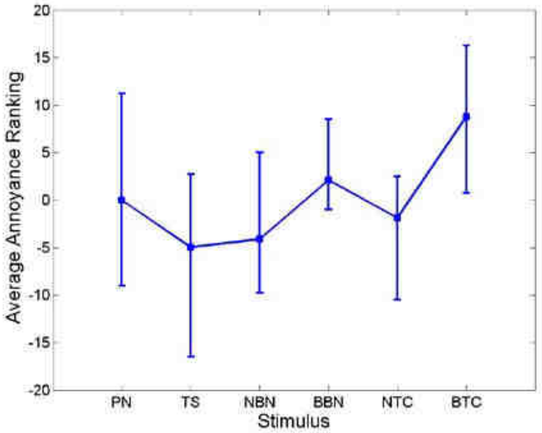


Figure 6: Absolute stimulus annoyance per stimulus type, independent of loudness. Calculated as average ranking of stimulus for all levels over all subjects. Error bars show maximum and minimum average scores.

Note that the magnitudes of the numbers on the y-axis are meaningless; it is rather the relative positions of the stimuli with respect to the others that are relevant. These numbers are dependent on the total number of stimuli presented, and could easily grow or shrink while the overall contour of the graph would remain the same. Based on the location of the averages, it can be concluded that subjects found the two-sinusoid (TS) stimulus least annoying while they found the broadband tone cluster (BTC) the most annoying, in general. Of course, these rankings would change given sufficient loudness discrepancies.

3.3. The Loudness Factor

As expected, the level of the stimulus has a huge influence on the annoyance. Given the experiment's result, one can determine the effect of loudness quite easily. For each stimulus and for each subject, take the difference of the highest ranking and the lowest ranking. The largest of these differences will correspond to the greatest ability of the stimulus level to influence the annoyance. Figure 7 shows the average of this loudness factor value averaged over all sixteen subjects. The error bars show the maximum and minimum differences per stimulus.

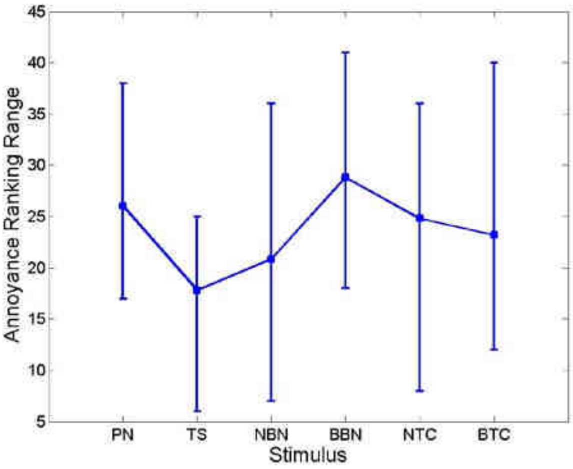


Figure 7: Loudness influence per stimulus type. Influence calculated by taking the highest and lowest ranking per stimulus per subject. Higher influence scores correspond to higher ability of stimulus loudness level to affect annoyance.

Notice that the broadband noise (BBN) has the highest of these difference values, while the two-sinusoids (TS) stimulus has the lowest. This means that the same 10 dB SPL increase on these stimuli will produce a much more drastic increase in annoyance on the BBN than on the TS. Note again that the y-axis values are irrelevant and that only the relative heights of these averages are useful in determining the loudness factor.

4. Discussion

Figure 8 is a graph of the summed results of all sixteen subjects. This graph shows that it is possible for rankings to switch places depending on the level especially, as it appears, at low stimulus levels. It also shows that loudness can have a curious nonlinear effect on stimulus annoyance, most prominently on the NTC, which will be studied in the future.

The authors did not find any conclusive evidence between the duration of musical training and preference toward a particular stimulus. We speculate instead that a certain relationship exists between annoyance and cultural background, but there were not enought subjects involved to make a legitimate claim on this. A previous study looked at this, but failed to acknowledge the large age discrepancy between the subjects of the two cultures in question [8]. With future testing, we aim to converge on a conclusive result on this point.

A subject mentioned in the feedback that familiarity with a certain type of stimulus may play a role in their judgment. In other words, she described that people will readily judge that a beating tone (which may sound like a telephone ring signaling potential communication) is more pleasant than a BBN, which is similar to the noise when the TV broadcast is not in service, which may invoke an unintentional and unpleasant memory from a subject. This effect is unavoidable given the ubiquity of these stimuli.

Subjects also commented frequently on the duration of the annoyance stimuli. Some mentioned that the 1000 msec was not significantly long enough for them to become annoyed. Unfortunately, this was considered during the experiment design phase as a necessary downside considering that annoyance from every stimulus would lead to annoyane and fatigue from the whole test.

The authors were also asked if 500 msec would be long enough a pause between pairs of stimuli. It could be too short for some subjects to forget the sound of the last stimulus, which may affect the perception of the next stimulus; however, much like in choosing the length of the stimulus, the pause was selected to be significantly short to keep the overall length of the test reasonable.

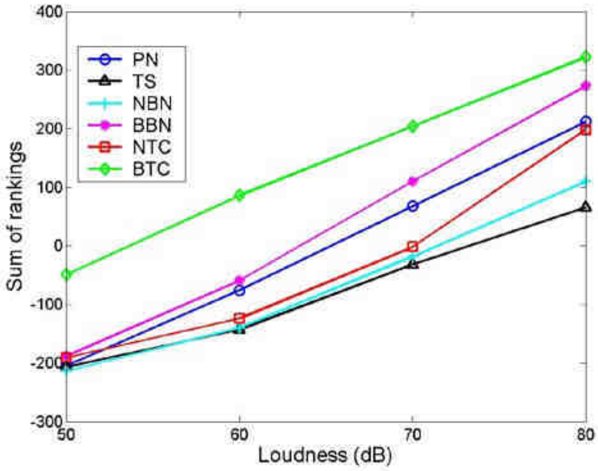


Figure 8: Total sum matrix. Results of all sixteen subjects added together. Shows the influence of loudness and absolute annoyances.

5. Conclusion and Future Work

This paper presented a novel method of judging the annoyance of an arbitrary number of stimuli at different loudness levels. We considered twenty-four stimuli for the experiment with sixteen subjects with various backgrounds. The method proved to be robust under different decibel-weighting schemes. Furthermore, the authors found that different types of stimuli may invoke different levels of annoyance, regardless of the SPL. The relationship between the loudness level and the annoyance rankings were verified in this paper; however, the authors did not yet find any conclusive relationship between the annoyance rankings and more subtle factors such as cultural background, age or the duration of musical training. These will be studied more thoroughly in the future.

The test was also performed monaurally. While reducing the real-world applicability of the results, this decision was made to eliminate inter-aural effects. Since the stimuli would have been identical in both ears and thus, absent from localization cues, the experiment was performed monaurally to avoid having some stimuli, and not others, cause disorientation and thus, more annoyance. The introduction of stereo stimuli will be another consideration for future testing.

The large ranking matrix with forced-choice procedure we proposed in this paper appears to be useful for a task like this that would otherwise be daunting to the subject. Because of the property of the transitive power in mathematics, these results can be obtained in under 30 minutes while producing numbers whose relative values hold a lot of information. This test goes beyond simply ranking annoyances, which would most likely exhaust the subject before completing the task. The test has the capability to preserve no-preference options, give valuable trend information based on various characteristics of the stimulus, and provide an easy task for its human subjects.

Our experiment once more verified the relationship between loudness and annoyance, in general. Also, the result shows that there is a very likely relationship between the bandwidth of a stimulus and the annoyance, i.e. that the wider the bandwidth of the stimulus is, the more annoying it is perceived, on average, at a fixed sound pressure level. This result can be useful in the design of sound signals, such as car alarms, as well as in a basis for consonance/dissonance research.

Acknowledgements

We would like to thank Stanford's CCRMA for the use of their facilities and software. We would especially like to thank professors Chris Chafe and Jonathan Berger for expertise and guidance from the initial phase of this project. Due to the nature of the test, we would like to give special thanks to our subjects for their patience and time as well as their feedback.

Appendix: Consistency Ratio

Assume there are N stimuli, all of which have different (and therefore unique) rankings, $[a_1, a_2, ..., a_k, ..., a_N]$, such that

$$a_1 < a_2 < ... < a_k < ... < a_N.$$

Also assume there is an ideal subject, who is always consistent in their decision making.

Since there are N stimuli, there will be

$$C(N, 2) = N\,(N-1)\,/\,2$$

different pairs, where $C(N, 2)$ is "*N choose 2*".

The algorithm to calculate the consistency of the ideal subject is as follows.

1. Start with $b_1 = b_2 = ... = b_k = ... = b_N = 0$, where b_k is the final ranking of the k-th stimulus, a_k.
2. With a pair of stimuli $[a_i, a_j]$, for $\forall i \neq j$, we award

   $b_i = b_i - 1,\ b_j = b_j + 1$ if $a_i < a_j$ or

   $b_i = b_i + 1,\ b_j = b_j - 1$ if $a_i > a_j$

   At every iteration, +1 and -1 points are awarded, therefore the absolute value of points awarded per iteration is 2. Hence, for the N stimuli test,

   $$A_N = 2 \bullet C(N, 2) = N\,(N-1)$$

   points will be awarded in total.
3. After $C(N, 2)$ iterations, b_k will be

$$b_k = 1 \bullet (\text{number of times } a_i > a_j) \\ - 1 \cdot (\text{number of times } a_i < a_j)$$

$= (k - 1) - (N - k)$
$= 2k - N - 1, \qquad \text{for } k = 1, 2, ..., N$

Due to the symmetry of the setting, there is a relationship of $b_k = b_{N-k}$, and furthermore $b_M = 0$ when $N = 2M - 1$ for some integer M.

4. The total sum of the rankings after all iterations is

$$B_N = b_1 + b_2 + ... + b_N =$$
$$= 0.5N^2 \qquad \text{for } N \text{ even, and}$$
$$= 0.5(N^2 - 1) \qquad \text{for } N \text{ odd}$$

Therefore for the ideal subject, the ratio of the highest consistency score to the total points awarded, or percent consistency, will be

$$B_N / A_N = 0.5N / (N - 1) \qquad \text{for } N \text{ even, and}$$
$$= 0.5\ (N + 1) / N \qquad \text{for } N \text{ odd.}$$

Hence, the perfectly consistent subject will achieve a consistency ratio is 2/3, for $N = 3$ and 4, and it will converge to 1/2 as N grows large.

References

[1] Dick Botteldooren, Andy Verkeyn, *Fuzzy models for Accumulation of reported community noise annoyance from combined sources*, Journal of Acoustic Society of America, 112(4), pp. 1496 – 1508 (2002)

[2] J. Cowan, *Architectural Acoustics Design Guide*, Acentech (2006)

[3] S. Devarajan, D. Levitin, V. Menon, J. Berger, C. Chafe, *Neural dynamics of perceiving event boundaries in music*, Cognitive Neuroscience Society Abstracts, 2006 and Human Brain Mapping Abstracts (2916), 2006

[4] James M. Fields, *Effect of personal and situational variables on noise annoyance inresidential areas*, Journal of Acoustic Society of America, 93(5), pp. 2753 – 2763 (1993)

[5] Rhona P. Hellman, *Growth rate of loudness, annoyance, and noisiness as a function of tone location within the noise spectrum*, Journal of Acoustic Society of America, 75(1), pp. 209 – 218 (1984)

[6] Rhona P. Hellman, Andrzej Miskiewicz, Betram Scharf, *Loudness adaptation and excitation patterns: Effect of frequency and level*, Journal of Acoustic Society of America, 101(4), pp. 2176 – 2185 (1997)

[7] Rhona P. Hellman, Eberhard Zwicker, *Why can a decrease in dB(A) produce an increase in loudness?*, Journal of Acoustic Society of America, 82(5), pp. 1700 – 1705 (1987)

[8] S. Kuwano, S. Namba, H. Fastl, *On the judgment of loudness, noisiness and annoyance with actual and artificial noises*, Journal of Sound and Vibration, 127(3), pp. 457 – 465 (1988)

[9] Eveline Maris, Pieter J. Stallen, Riel Vermunt, Herman Steensma, *Noise within the social context: Annoyance reduction through fair procedures*, Journal of Acoustic Society of America, 121(4), pp. 2000 – 2010 (2007)

[10] Henk M. E. Miedema, *Relationship between exposure to multiple noise sources and noise annoyance*, Journal of Acoustic Society of America, 116(2), pp. 949 – 957 (2004)

[11] Andrew J. Oxenham, Brian J. Fligor, Christine R. Mason, Gerald Kidd, Jr., *Informational masking and musical training*, Journal of Acoustic Society of America, 114(3), pp. 1543 – 1549 (2003)

[12] Ernst H. Rothauser, Günther E. Urbaner, Walter P. Pachl, *Loudness and annoyance of impulsive office noise*, pp. 520 – 522 (1968)

[13] Theodore J. Schultz, *Synthesis of social surveys on noise annoyance*, Journal of Acoustic Society of America, 64(2), pp. 377 – 405 (1978)

[14] Yoshiharu Soeta, Takuya Maruo, Yoichi Ando, *Annoyance of bandpass-filtered noises in relation to the factor extracted from autocorrelation function (L)*, Journal of Acoustic Society of America, 116(6), pp. 3275 – 3278 (2004)

[15] S. M. Taylor, *A comparison of models to predict annoyance reactions to noise from mixed sources*, Journal of Sound and Vibration, 81(1), pp. 123 – 138 (1982)

[16] Emily Thompson, *The Soundscape of Modernity: Architectural Acoustics and the Culture of Listening in America, 1900-1933*, The MIT Press (2004)

[17] *American National Standards Institute (ANSI*) *Standard S1.4-1983(R2006), American Natioal Standard Specification for Sound Level Meters*, 1983 (Revised 2006)

[18] *ITU-R Recommendation 468-4, Measurement of Audio Frequency Noise in Broadcasting, Sound Recording Systems and on Sound Programme Circuits,* Broadcasting Service (Sound), Dubrovnik: International Telecommunication Union, Volume X -Part 1, pp. 285- 291 (1986).

[19] http://en.wikipedia.org/wiki/A-weighting, as of 18 August 2007

[20] Daniel Steele, *Interview with Jean-Claude Risset*, a class project, May 2007.

Visual Stimulus for Aural Pleasure

Gareth Davies, Stuart Cunningham & Vic Grout
Centre for Applied Internet Research (CAIR), University of Wales, NEWI
Plas Coch Campus, Mold Road, Wrexham, LL11 2AW, North Wales, UK
mrgdavies@googlemail.com | s.cunningham@newi.ac.uk | v.grout@newi.ac.uk

Abstract. Can *images* be represented by *music* and if so how? This is one of the principal questions addressed in this paper through the development, implementation and analysis of computer generated music and assessment of the music produced.

Images contain a multitude of *definable information content* such as the colour values and dimensions, which can be analysed in various statistical combinations to produce histograms and other interpretations of the content. This provides music composition algorithms with a variety of data sets and values which can be used to inform parameters used in the music generation functions. However, images also contain *semantic information* which is widely open to interpretation by a person looking at the image. Images, much like music, are able to induce *emotions*, *feelings* and other *soft responses* in a viewer, demonstrating that images too are more than simply the sum of their parts. Therefore, if we are to generate music from image data then a goal of doing so would be to produce music which can begin to invoke similar *humanistic* responses.

The work presented in this paper demonstrates how compositional algorithms can be used to create computer generated music from a set of various images. Several established music composition algorithms are explored as well as new ones, currently being developed by the authors of this paper. We explore different uses and interpretation of the image data as well as different genres of music, which can be produced from the same image, and examine the effect this has. We provide the results of listener tests of these different music generation techniques that provide insight into which algorithms are most successful and which images produce music that is considered to be more popular with listeners.

Finally, we discuss our future work, directions and the application areas where we feel that our research would bring particular benefit. In particular, we seek to incorporate the work presented in this paper with other technologies commonly used to assess visual and psychological responses to images and propose techniques by which this could be harnessed to provide a much more dynamic and accurate musical interpretation of an image. One of the main goals we aim to achieve through further development of this work is to be able to successfully interpret an image into a piece of music or sound, which could be played to a visually impaired or blind listener to allow them to grasp the emotional content and responses which imagery can invoke.

1. Introduction

The reflection and association between imagery and music is a common manifestation in the 21st Century, particularly due to the nature of the rich, multimedia environments and data to which we are exposed everyday. A prime example is the success of the music video and MTV, which has received worldwide recognition and familiarity in a relatively short period of time. The way in which music and images or video are often combined demonstrates the value and semantic link which humans can easily make and the extra information provided when these otherwise independent media are fused [1]. The emotional influence carried by this combination of these media and the imagery and interpretation associated are often thought to be highly powerful and influential in many areas, when interpreted by users into the 'real world' [2, 3, 4].

In this work, we examine the viability and practical success of analysing image data and creating musical compositions influenced by the image data.

The initial outcome of this research is to produce a functioning program that takes a bitmap image as the source file, reads the data and uses that data as the inputs to an algorithm to generate MIDI (Musical Instrument Digital Interface) music files. Users will have a choice of algorithms and musical genre. The choice of genre will affect the tempo and instrumentation in the MIDI file and the image data affects the scales, pitch, time signature, tempo and structure applied. In the longer term, one of our aims is to produce music which is reflective of the semantic content also present in the image. Such techniques could be used, for example, to create representations of images for those who are not fully able to interpret the image due to visual impairments.

We begin by discussing the information and data that is contained within images and how these factors can be employed when attempting to create music. In particular, we draw attention to the fact that images contain both numerical, raw data as well as a harder to define, emotional or humanistic content. We then discuss how image data can be interpreted to produce music and present several compositional algorithms which can be used to automatically generate musical sequences, including an original algorithm developed by the authors of this paper. In the subsequent section, these algorithms are tested with human listeners and the results discussed. Finally, we conclude the work completed to date and discuss areas which are currently under investigation and development to further improve and refine the methods and techniques detailed within this paper.

2. Images and Information

Humans gather a large majority of their information from sight. *A picture is worth a thousand words*[1]. Images allow not only capture and representation of a moment or scene, but also provide a vehicle by which humanistic emotional responses can be triggered. Therefore, we consider images to hold two very distinct values when we interpret visual data and emotion into musical audio data and emotion.

[1] http://en.wikipedia.org/wiki/A_picture_is_worth_a_thousand_words

2.1. Data Content

There are varying types of data that can be extracted from an image. For example, the RGB (Red, Green, Blue), CMY (Cyan, Magenta, Yellow) or HSL (Hue, Saturation, Luminance) values for each individual pixel, histograms, the width and height of the image, and other more meta properties of the properties of the file itself, the name, size on the disk , etc. The content of the image which describes the pixels and colour values which constitute the image are of particular use and these can be analysed through a number of statistical techniques to gain further depth and insight into the particular qualities if the image. Of such techniques, histograms in particular, provide a useful tool which allows us to easily convey large amounts of information about the colour content of an image.

A histogram provides a graphical representation of the frequency of a set of results. In the case of an image it would be the frequency of the occurrence of a particular colour. This allows us to obtain a distribution of the components which form the image. A histogram can be produced from individual values such as all the red values from an image or be made up of a combination of values such as the total RGB values or a combination of the CMY values. This yields a large amount of data from an image and the different permutations of the histogram can provide vastly different values. As an example, consider the greyscale image in Figure 1 and the associated histogram presented in Figure 2.

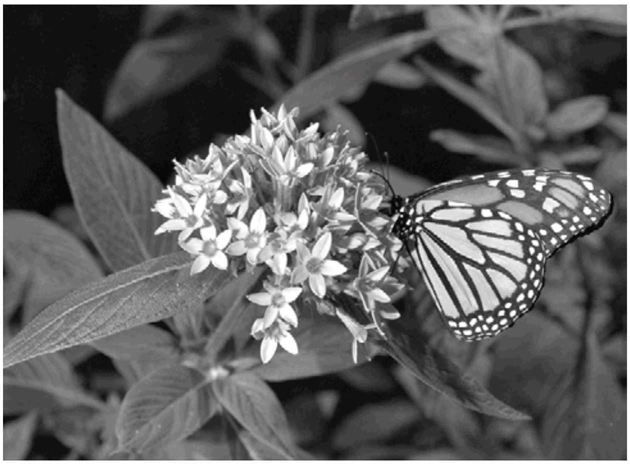


Figure 1 - Sample Greyscale Image

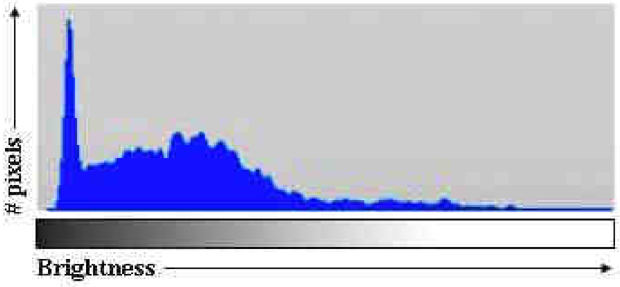


Figure 2 - Histogram of Greyscale Image

Such information about an image gives us initial tools with which to be able to reflect the content of the image in the music we produce. For example, it would be expected that particularly bright images and images with large variations in colour would translate into more diverse and intricate music.

2.2. Semantic Content

When we approach image analysis we must also consider the human and emotional responses which viewers will have to an image. We know that humans frequently exhibit fundamental emotional responses when viewing differing visual imagery [5]. It is worth thinking about the attraction and fascination that humans have with classical works of art such as Leonardo da Vinci’s *Mona Lisa*. Consider also imagery which invokes many types of emotional response such as happiness, sadness, distress and contentment. Images provoke many kinds of emotional and humanistic responses and we must attempt to reflect these responses in the associated music produced from an image.

Such responses are much harder to quantify and predict using purely statistical analysis. For example, consider the expected emotional response from a viewer who observes an image of war, which might depict a bloody battle between two soldiers. In this case we would expect a viewer to have a negative emotional response such as sadness or horror. Now consider an image which might provoke a more positive emotional response, such the image of a man giving a woman a bunch of roses in a grassy meadow. We would expect this to have an emotional response of happiness or compassion. However, if we consider the data present in these images we find many common features. For example, strong reds (blood in the battle image, the roses in the romantic image), flesh tones (the soldiers in the battle, the man and woman in the meadow), light blues (the sky in both images), grassy or earthy backgrounds (the ground in the battle, the meadow in the romantic image). We can surmise therefore, that the histograms extracted from these images would be very similar. This is not an ideal scenario and indicates that we need to consider techniques other than purely the raw data content in order to fully extract emotional content for music from images. We discuss methods of addressing this issue in our ‘Conclusions and Future Work’ section.

3. Creating Music from Images

Music has been composed algorithmically in the past and it can be traced back as far as the ancient Greeks as Grout and Claude stated:

> *"The word music had a much wider meaning to the Greeks than it has to us. In the teachings of Pythagoras and his followers, music was inseparable from numbers, which were thought to be the key to the whole spiritual and physical universe. So the system of musical sounds and rhythms, being ordered by numbers exemplified the harmony of the cosmos and corresponded to it"* [6].

This notion from the Pythagoreans is not necessarily as abstract as it may sound. Autistic people that can do huge numerical calculations without the aid of calculators or computers often see numbers as colours and/or shapes. The fact that they can successfully carry out these calculations quickly indicates that there are associations between these visual objects and the numbers.

More recently, Mozart used a simple form of algorithmic composition in his *Musikalisches Wurfelspiel* or *Dice Music*, where he had written a number of musical parts that he put together with the aid of dice rolls into a complete musical piece. Algorithmic composition with computers began in the 1950’s and 1960’s. Notably a piece called the *Illiac Suite* (1957) was produced in the University of Illinois, USA [7]. These approaches demonstrate how simple tools such as random variables, probabilities, transition tables, and rules can be employed in the generation of musical elements and the structuring of therein. We chose to employ three compositional algorithms in conjunction with the image data.

3.1. Image Selection

The images selected for our investigations play an important part when assessing the results and success of our study. To provide a balanced outlook on the production of music from images, a set of reference images was chosen to be employed in all tests. These sets of images formed four categories: people, city, landscape, and art. This spread allows us to accommodate a large variety of images common in everyday life. This selection should provide the compositional process with differing ranges and distributions of colour values. Samples of images from each of the categories can be seen in Figures 3 to 6.

Figure 3 - People Category

Figure 4 - City Category

Figure 5 - Landscape Category

Figure 6 - Art Category

3.2. Assignment of Musical Properties

A primary issue which had to be dealt with, was to determine the best method to extract properties from the image which can be mapped against the parameters required for a musical composition. To determine which image colour values were best for selecting which MIDI data, a program was produced that generated the charts for the RGB and HSL values averaged over the height. This enables the flow of the song to be analysed given which colour values are used for which musical attribute. The various mapping we experimented with are detailed in Table 1.

Variation	Tempo	Time Sig. Numerator	Time Sig. Denominator
1	Red	Green	Blue
2	Green	Blue	Red
3	Blue	Red	Green
4	Hue	Saturation	Luminance
5	Saturation	Luminance	Hue
6	Luminance	Hue	Saturation
Variation	**Pitch**	**Duration**	**Velocity**
1	Hue	Saturation	Luminance
2	Saturation	Luminance	Hue
3	Luminance	Hue	Saturation
4	Red	Green	Blue
5	Green	Blue	Red
6	Blue	Red	Green

Table 1 - Assignment of Image Attributes to Musical Properties

To complement this, a short excerpt from a song for each algorithm, using the various colour values for the various MIDI data, was produced and a questionnaire to accompany these excerpts was made. This questionnaire was issued to volunteer listeners who listened to the excerpts of the songs and then selected which one they preferred for each algorithm. The results of this test will influence the decision of how to map the image attributes to musical properties.

3.3. Compositional Algorithms

Once various interpretations of image data into musical properties were determined, these could then be used to inform the compositional algorithms. We created small libraries of fixed musical selections, such as various standard musical scales, time signatures, tempos, note duration, and velocity. It was necessary to apply these constraints to the ways in which the image information was interpreted and quantised into a form which was suitable for the algorithms, especially given the relative scale of the investigation. This discussion is detailed and goes beyond the scope of this paper. However, it should be remembered that such constraints were required in order to be able to successfully interpret values from images into musical properties. Roughly speaking the data obtained from images would have a broader numerical range than the musical parameters. Consider the range of 0 to 255 from an image attribute and time signature numerator which is usually much smaller, for example. The algorithms we used as part of our work are as follows.

3.3.1. Algorithm 1 – *1/f*

Due to the self-similar nature of this algorithm and the fact that it has been frequently used previously for musical composition, the *1/f* algorithm proved a logical selection to combine with image data. The *1/f* algorithm has no direct input but has a random number generator which can be seeded with an integer, so this provides an indirect method for the algorithm to be affected by image data.

3.3.2. Algorithm 2 – Euclidian

The Euclidean algorithm takes two number values and returns the greatest common divisor, this accommodates the input and output that is required and was another obvious selection when we consider the amount of image data available. This can be simply demonstrated with the following sample of C code:

```
int GetEuclidian(int x, int y)
        {
        int z;
                while(y!=0)
                {
                  z = y;
                  y = x%y;
                  x = z;
                }
     return x;
}
```

3.3.3. Algorithm 3 – Colour Selection Process

Strictly speaking this is not an algorithm but more of a selection process. It was developed to provide an alternative to the two algorithms already selected.

The process works by using the actual colour values from the image. The theory behind this is that this should provide the most direct link between the image and the musical composition produced. Which colour values are used with which MIDI data will have an effect on the type of composition that is produced.

4. Discussion of Results

4.1. Assignment of Musical Properties

The results of the investigation into how to map image attributes to musical properties is show in Table 2. This shows the percentages of user preference for each colour/music assignment variation, tested against each of the three compositional algorithms.

Variation	*1/f*	Euclidian	Colour Selection	Average
1	8.33%	6.25%	12.50%	9.0%
2	14.58%	14.58%	14.58%	14.6%
3	6.25%	12.50%	10.42%	9.7%
4	29.17%	20.83%	25.00%	25.0%
5	22.92%	27.08%	20.83%	23.6%
6	18.75%	18.75%	16.67%	18.1%

Table 2 - Result of Image Interpretation Variations

The configurations with the greatest number of responses for each of the three algorithms were chosen as follows:

- *1/f* Algorithm – Variation 4
- Euclidian Algorithm – Variation 5
- Colour Selection Algorithm – Variation 4

An interesting point to note is that the majority of responses were selecting the HSL values to represent tempo and time signature and the RGB values to represent pitch, duration and velocity; variation 4, which had the largest overall average of selection across all the tests. Also of note is that variations 2 and 6 produce almost equal results across all of the algorithms.

4.2. Compositional Algorithms

The algorithms currently available give the user a chance to evaluate how a different algorithm with the same image can make a completely different song and can enable them to make a choice about which algorithm makes the best representation of the image with sound. To enhance the ability to see each algorithms performance across the six variations, a series for each algorithm is plotted in Figure 7.

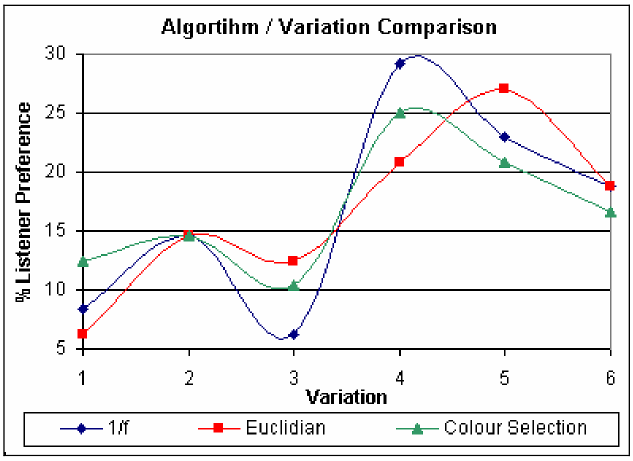


Figure 7 - Compositional Algorithms & Variations

At this stage, we are more focussed upon establishing suitable mappings between the image attributes and the musical properties and the trends of each series in Figure 7 suggest that our assumption and mapping proposed in the previous sub-section is valid.

However, we are also interested in starting to investigate the effect that each of the algorithms may have had upon the user preferences. As we can see in Figure 7 the user selections for some algorithms take the form of a more varied distribution across each of the mapping variations, whilst others are less varied. The data presented in Table 3 also helps to reveal the properties of each of the algorithms across the six variations in terms of the percentage of user preferences.

Algorithm	Min	Max	Variance	Std Dev
1/f	6.25	29.17	76.39	8.74
Euclidian	6.25	27.08	52.08	7.22
Colour Selection	10.42	25.00	29.51	5.43

Table 3 - Data Ranges in Results

If we consider the *1/f* algorithm first, it has the most extreme fluctuations across the six variations. This would suggest that the *1/f* algorithm is more influenced by the mapping of image attributes and therefore the types of image which are used to generate music using the algorithm. Interestingly, the *1/f* method also contains the maximum value and shares the minimum with the Euclidian, in the data sample. Although further testing is required, this further enforces the belief that this technique is sensitive to the image mapping and attributes, but it also indicates that it could be considered to hold the most potential in producing satisfactory musical compositions.

The performance of the Euclidian algorithm tends to vary considerably across the six variations, although for variation 4 and 5 it exhibits almost inverse performance to that of the other two composition methods. However, in terms of the overall variation across all six variations, the results of the Euclidian method are almost as varied as that of the *1/f* algorithm. This, and the fact that the Euclidian algorithm shares the lowest value in the sample data with the *1/f* algorithm, suggests a similar dependency upon careful image attribute mapping and selection.

The trend exhibited by our own Colour Selection process generally follows the same overall trend of the *1/f* technique although it is not scaled to the same extremes. Our method seems to provide the most consistent, flattest results across each of the variations. The Colour Selection technique displays the smallest mathematical variation and standard deviation of all three of the algorithms. Therefore, whilst we consider that the *1/f* algorithm and the Euclidian algorithm are very dependant on image attributes and data, our technique is the most likely to produce satisfactory musical compositions. However, these results suggest that we must consider improvements to this method if we intend to be able to produce excellent musical compositions.

5. Conclusions & Future Work

We recognise the immediate need to provide more detailed assessment of the three algorithms used and the image selections. In this work we were primarily concerned with determining the most suitable mappings of parameters and attributes. However, with the data and resources now available we can launch a more detailed investigation into determining the usefulness or listener ranking for each of the algorithms in terms of the music which is generated. The same also applies to exploring which of the image categories produces the most satisfying music. As in any study, where human perception and interpretation is required, it will always be challenging to ascertain such information as absolute fact, due to the varied nature of perception, and particularly musical preference, across a broad field of listeners. The key to producing substantiated evidence of these facts lies in employing a large range of tests and subjects to ensure as broad a listener base as possible, and to ensure that qualitative data is also produced at these times.

One of the key aspects of this investigation was the representation of images as music. This was achieved from a technical point of view; the software and algorithms will generate music from images and will have a different output for each image that is used. This has been proven in the form of the listening tests performed. It is also true that by using the different image data available for different aspects of the musical composition hugely different musical compositions were produced and the same also applies when using the same image with different compositional algorithms.

An interesting point which leads us to future investigation and development is how well the images are represented by the music produced. Also, will it ever be possible to *truly* represent images with music? Again this is affected by the way that people view images. Different people have vastly different opinions on how an image should be represented by music, for example people who are affected by the condition Synesthesia can hear music while they are looking at images, and these people would therefore likely have a strong opinion on what music would be best represent an image [8]. Other people to consider are those affected by colour blindness they are likely to have a completely different opinion to those who are not.

This leads us into a significant area to further our work to date; to monitor, and subsequently model, the ways in which human viewers look at an image. The ability to use eye-tracking technology provides the research field with the opportunity to further investigate and correlate vision with emotive response [9]. We propose that by employing eye-tracking tools, it will be possible to further determine the most crucial and important parts of an image when viewed by a fully-sighted viewer. Our hypothesis is that the emotional response experienced by the viewer of an image must be directly related to the particular parts of the image which the viewer focuses on as these are the principal stimuli which invoke the emotional response. Work by Jackson *et al.* substantiates this hypothesis [5]. Although their work is more concerned with psychopathology and psychophysiology their work demonstrates distinct correlation between emotional responses to images and the reflection of emotional response in the eyes. This may also help to address the example cited in section '2.2 Semantic Content' where we discuss the issue related to purely statistical analysis of image data in the case of the two contrasting pictures; the scene of a bloody battle and the scene of a man and woman with a rose.

The notion of investigating specific focal points within images to determine the most important areas to a viewer is reinforced by the works of Santella and DeCarlo [10, 11]. Their work is particularly focussed on obtaining and attempting to model the human perception of images, such that which is available in photographs. Although the goal of Santella and DeCarlo's work is to modify and create more abstract interpretations of the images analysed, they focus on being able to ensure that the most significant portions and areas within the image receive the most attention and level of granularity within the new version. This reflects the intentions we propose; to focus music composition algorithms on the most significant areas within an image which hold the greatest value and importance to the viewer.

For example, consider the image shown in Figure 8.

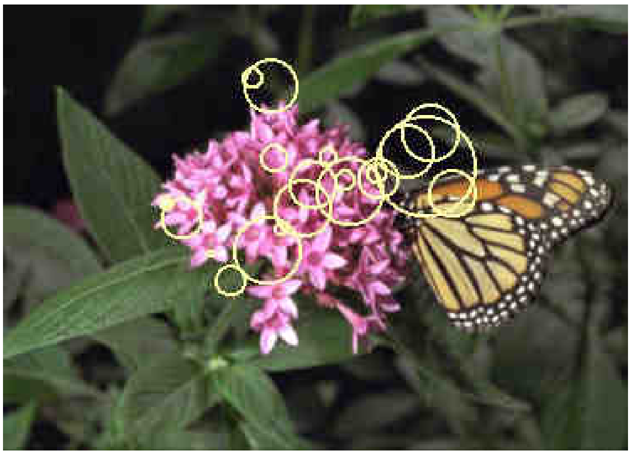


Figure 8 - Image with Eye-Tracking Overlay

This image is taken from data repository of Santella and DeCarlo's work and the eye-tracking data associated with the image has been overlaid [10]. The circles show the areas upon which the viewer has focused. The size of the circle indicates the length of time the viewer spent observing that particular area, the larger the circle, the more time spent viewing that area. This could be considered a weight metric within the image.

We can see that in the case of this example, the viewer focuses primarily around the area of the flower and the butterfly, and only pays small attention to the background setting. We propose that as part of our future work it would be possible to record multiple viewing of images from fully-sighted users and determine the common areas of focus within images and the correlated emotional response. Once a sufficient perception model has been created for an image these additional parameters could act as filters to influence the compositional algorithm. As a simple example, the eye-tracking data could be used to direct the composition techniques at specific parts of the image, rather than analysing the whole image. Through iterative testing and development it is hoped that this process would be refined so as to be able to produce the same emotional experience through a piece of music for a visually-impaired listener as a sighted viewer would have when they see an image. This moves us beyond the question *"What does the Mona Lisa sound like?"* towards attempting to address the question *"What does the Mona Lisa feel like?"*

Of particular interest when attempting to discern and assess suitable emotive responses to images would be to create music using images of *Rorschach inkblots*. The investigation in this case would be to see if the music produced by inkblots also invoked the same emotional functions and responses in listeners as the actual inkblot images produced when viewed. This would provide tangible evidence that the music being created from our compositional methods was indeed resulting in the expected human responses.

Another interesting possibility to allow further tailoring of the music produced to each individual user could be in the use of genetic algorithms. The user could provide the fitness evaluation needed to determine best fit, this way the music composition algorithms could learn an individual's preference and attempt to match their representation of the image to that. However, obtaining suitable fitness metrics for visually impaired users could be unrealistic and therefore a more practical solution would be to base this metric on a suitable distillation, such as the mean, of a more easily obtainable data set, as we suggest previously with the eye-tacking data.

Additional intentions for future testing include integration and evaluation of more compositional algorithms and the integration of an interactive painting window, which would allow playback of images being drawn on screen by the user. This could be further enhanced by experimenting with real-time music generation whilst the user is drawing. This presents another intriguing area for future consideration.

References

[1] Chen, T., Rao, R.R., ***Audio-visual integration in multimodal communication***, Proceedings of the IEEE, Vol. 86-5, pp. 837 - 852 (1998).

[2] Royet, J.P, Zald, D., Versace, R., Costes, N., Lavenne, F., Koenig, O., Gervais, R., *Emotional Responses to Pleasant and Unpleasant Olfactory, Visual, and Auditory Stimuli: a Positron Emission Tomography Study*, Journal of Neuroscience, Vol. 20-20, pp. 7752-7759 (2000).

[3] Sherman, B.L., Dominick, J.R., *Violence and Sex in Music Videos: TV and Rock n' Roll*, Journal of Communication, Vol. 36-1, pp. 79 - 93 (1986).

[4] Hansen, C.H., Hansen, R.D., *How rock music videos can change what is seen when boy meets girl: Priming stereotypic appraisal of social interactions*, Sex Roles, Vol. 19-5/6, pp. 287 - 316 (1988).

[5] Jackson, D.C., Malmstadt, J.R., Larson, C.L., Davidson, R.J., ***Suppression and enhancement of emotional responses to unpleasant pictures,*** Psychophysiology, Vol. 37-04, pp. 515 – 522 (2000).

[6] Grout, D.J., Claude, V.P., *A History of Western Music*, 5th Edition, New York, W. W. Norton & Company (1996).

[7] Maurer IV, J.A., *The History of Algorithmic Composition*, Stanford University, USA (1999). Available at: http://ccrma-www.stanford.edu/~blackrse/algorithm.html [Accessed: 20th August 2007].

[8] Harrison, J., *Synaesthesia: The Strangest Thing*, Oxford University Press, Oxford, UK (2001).

[9] Kiegler, K., Moffat, D.C., ***Investigating the effects of music on emotions in games,*** Proceedings of Audio Mostly Conference, Piteå, Sweden (2006).

[10] Santella, A., DeCarlo, D., *Abstracted Painterly Renderings Using Eye-Tracking Data*, Proceedings of the 2nd international symposium on Non-Photorealistic Animation and Rendering (NPAR), ACM Press, New York, USA (2002).
Data set available at: http://www.cs.rutgers.edu/~decarlo/data/npar02/index.html [Accessed: 21st August 2007]

[11] DeCarlo, D., Santella, A., *Stylization and Abstraction of Photographs*, Proceedings of ACM SIGGRAPH 2002, New York, USA (2002).

Practical Aspects of Audio Design for Virtual Environments

Dorota Blaszczak,
Polish Radio, Warsaw, Poland
Dorota.Blaszczak@polskieradio.pl,
[In the 90's: The Banff Centre for the Arts, Softimage, and Immersence]

Abstract. A virtual environment is created based on the user movement in the space and this creates unique challenges for composing the audio space. This paper attempts to give an overview of many issues concerned with aural aspects of immersive environments: audio system elements, user interface and control signals, audio space composition and interactivity, audio generation, new means of expression, work process, documentation, and exhibition of an interactive work. The paper is based on my experience with sound design of several virtual environment projects that I worked on in the 90's.

1. Introduction

The Banff Centre for the Arts in Canada was engaged in development of several interactive virtual environments in the early 90's, to explore its potential as an aesthetic medium. Char Davies created two immersive pieces - *Osmose* and *Ephemere* - in Montreal in the late 90's. Having been involved in those projects as a sound and interaction designer[1], I have gained practical knowledge of the work process of creating sound for interactive environments (the paper is based on projects that I have worked on). It was not a research but production situation that created specific requirements and additional time pressure to finish the projects on schedule for the opening.

Those interactive, immersive projects made use of a setup of two or more computers and a HMD helmet with headphones, tracking sensors and various additional user interfaces (a "flying" mouse, a data glove, a gripping device, breathing and balance tracking). The specific types of equipment used for the projects are not really relevant to the main topic of the paper that attempts to discuss the sound issues on more general, hardware independent level.

The process of creating sound for virtual environments is really challenging. Unlike in linear media, everything happens in real-time and sound follows the movement of the user and the environment - you need to look for new categories of sound and new means of expression appropriate for the content of the work. Unlike in interactive music, the performer is an "unskilled" user who is a listener at the same time. Unlike in most linear productions, there is no completed version of graphic program to work on the sound for – the post-production process is interactive.

At the beginning of my work on virtual environments, the information I had found about virtual sound was focused on localized sound only (mainly for moving sources and collisions). At the same time there wasn't so advanced technology as today[2] so it pushed me to look for new ways of working with sound that were not possible in linear structure. Through all my work with interactive projects I have tried to focus on responsiveness of the overall sound, using localized sound as a tool for spatial "panning" and not as a goal of sound design.

In this new field of sound design I could make use of my experience and familiarity with music recording, electroacoustic music, film postproduction, theory of hearing and audio perception, rhythmics, sound design for early computer games, and also Roman Ingarden phenomenological aesthetics of art, with music as an intentional object which becomes distinct in each performance and in perception of each listener.

2. Elements of sound system

Audio related issues can be organized according to the elements of sound system, which is a part of the whole virtual environment installation - hardware and software boundaries between elements may vary with technology used for the projects[3].

1. Control signals module
- movement information from sensors on the user body
- graphic space information from the graphic environment
- audio control information from the audio environment

2. Audio space control module
- audio space composition and interactivity
- behaviours of sound objects
- spatial control of localized voices
- overall processing and mixing control

3. Data exchange between audio space control and audio generation modules
- start / stop triggers for audio voices
- selected control signals for specific scenes and voices

4. Audio generation module
- generation algorithms for all the voices
- voice processing control

5. Sound producing module
- samplers, processors, 3D processors, mixers

The paper concentrates on audio space control and composition more than on audio generation itself, in accordance with my role in various projects.

2.1. Control signals module

2.1.1. User interface

The choice of a user interface is very important for the content of the piece since it determines the character of the user

[1] Depending on the project I have worked by myself or together with a composer. In the most complex pieces *Osmose* and *Ephemere,* I was responsible for an audio space (sound objects and interactions) and Rick Bidlack, interactive music composer, worked on generation algorithms of all the voices.

[2] First I could work with two localized sound sources only (one per FocalPoint card), then with eight (Convolvotron and Acoustatron cards). The processing was noisy. And the polyphony of samplers (SampleCell, K2000) was quite limited.

[3] The setups of the projects consisted of a software part (on the Onyx, NeXT, Mac or PC computers) and an audio hardware part (for sound generation, processing, spatialization and mixing) with headphones as an audio output.

movements and interactions with the environment, and it defines the types of control signals available to work within the piece.

First, we had a predefined virtual system – with a sensor on the helmet and a "flying mouse" for pointing the direction of the movement. Other projects introduced additional elements: a dataglove, "gripping" devices for touching and moving the objects, and a microphone in the helmet. In the piece *Inherent Rights, Vision Rights,* a kiosk with a viewer and a joystick was built for the exhibition in the gallery.

Char Davies developed her own more natural interface - a special vest with breathing and balance sensors, based on diving in the water environment.

2.1.2. User movements

Sensors on the user body can track various types of movements – position changes in the real space, leaning of the body, movements of the head, movements of the hands (touching, flying), and breathing.

2.1.3. Control signals

All data from the user sensors are processed in the graphic part of the project. Some selected control signals from the sensors (raw or processed data) and from the graphic environment go to the audio program. This includes the user data mapped to a virtual graphic space: user movement (xyz, speed), breathing, leaning, bending the knees, head movement (azimuth, elevation, and speed); the environment data: light, time (day, season), scene envelope, position and speed of the objects; collision data - distance and angle to the objects.

Those signals are further processed to interpret information and to use for audio space composition, interactions and generation, based on passage of time, scene envelopes, "concentration" and specific movement detection, touching and moving the objects.

2.1.4. Audio-based control signals

Audio signal from a microphone can be used as a control signal derived from the sound envelope and pitch-tracking.

Audio program can also generate control signals for the graphic environment based on pulse or type of sound to make graphics "listen" to the soundscape.

For instance in *Osmose*, leaves movement in the forest scene was controlled by intensity of a wind sound, caused by the user movement. Tempo of scrolling and vanishing of program lines in the code scene was controlled by the rhythm from a music algorithm of a code voice.

Long soundfiles with recorded narrative may need synchronization with graphics to allow the user to listen to the whole file: triggers of the end of the file must be sent to the graphic program to allow for the following changes within the environment.

2.2. Audio space control module

Sonic environment responds to the user movement in virtual space. We need to choose[4] the movements and gestures that would have an influence on audio space to emphasize the meaning and emotions of a given scene. At the same time we can't predict the behaviour of the user who plays an important role in the progress of the piece. Therefore various degree of the user activity must be taken into consideration to achieve the intended aesthetic result and give some feedback on her/his influence on an audio environment created on the fly[5].

2.2.1. Audio program structure

A "score" of an interactive sound is written as an audio program which describes potential behaviour of the soundscape.

The program contains descriptions of individual audio scenes, transitions between the scenes, behaviours of audio objects, and it continuously updates the parameters of an audio environment. It assigns actual voices to audio objects and specifies their responsiveness to the user movements and to the environment. It also controls spatialization of selected sounds, additional audio processing and a final mix.

The audio space control program had two components: one as a part of the graphic program and another one as a separate audio program (written in MAX or a custom language mentioned below).

2.2.2. Transitions between the scenes

There is an important issue that the program must take care of. These are spatial-temporal transitions between audio scenes when sounds can easily get stuck or be interrupted[6]. In most of the projects I had to deal with the transition problem on the audio side of the project since the graphic scenes were quasi-static. It had changed in *Ephemere* where there was a special language used, created by John Harrison[7]. It was based on the time envelope of life of a particular element and it could take care of both audio scene and an individual sound object in case of temporal transition. A transition spread in space required a special hysteresis-like characteristics to switch the scene on and off in slightly different spots in the environment.

2.2.3. Introduction and final scenes

A virtual project is divided into series of scenes – audio-visual spaces arranged within the environment or successive in time. There are two important types of scenes which occur once in the piece.

The project may be entered through the first, special scene. This scene can serve as an introduction to the narrative of the piece (e.g. in the beginning of *Inherent Rights, Vision Rights* the user leaves civilization to walk towards the longhouse with a non-interactive audio introduction) or as an orientation for the user to learn the principles of the movement in the environment. The latter type of introduction was very important for *Osmose* and *Ephemere* which were based on a new interface and were intended for long-term exhibitions in art galleries. There was a grid of Cartesian coordinates in *Osmose* introduction scene used as a good reference for the user movement in the space. A soundscape of that scene was built on straightforward responsiveness to the user movements: going up or down in the space caused a pitch glissando of a base note, head movement up or down built a chord up or down, moving fast added a staccato effect, and finally the user could find a spot of silence with looking straight ahead.

The project may end with a scene that is forced after specific time that is important for exhibiting an interactive work. For instance, there was one ending scene in *Osmose* that was introduced onto any scene the user was in. The ending of *Ephemere* was different for each scene but it had similar character and emotions.

[4] I found it useful to move during the process of designing an interactive audio space – it was like "choreography" for sounds.

[5] The users may not be fully aware of their influence on what they hear especially if they had spent up to 15 minutes in the environment. People with some audio and music experience could notice more details – you need to learn to listen to virtual sound.

[6] Sound interruptions tend to be more noticeable than graphic "jumps". Even in early projects when the graphics could be "switched" immediately from one space to another, the sound from both scenes had to overlap at least with a release of sound.

[7] Some problems from the previous projects specific to temporal character of sound had an influence on the temporal structure of that language.

2.2.4. Layers and types of sound

An overall audio space composition for immersive project can be conceptualized at two levels: designing behaviour of responsive sonic environment and audio engineering including spatialization of selected sounds, various processing, and mixing. Those levels overlap in the process of creating sound. A set of various overlapping classifications below, came from practical experience and was built as an attempt to identify relations between sounds [some examples or explanations are added in the brackets]:

1. Sounds of the user body [a musical drone dependent on the user movement], of graphic objects [a rock], of invisible objects [a fly], or sounds of scenes [sound of the fog].
2. Mono [surface crossing], stereo [body sound], or localized – single point [a fly] or a localized group [stereo pair of a river] - sounds.
3. Static [a rock], moving [a fly], or internal, „carried" by the user [body sound] – sounds.
4. Periodic [required start and stop triggers] or one-shot [required start only] sounds.
5. Sounds of various mixing and processing groups [sounds of an individual scene].
6. Sounds temporarily exempted from processing group [in concentration mode, when only exempted sound was heard].

2.2.5. Sound objects

An interactive soundscape consists of sound objects – various objects, places and situations are represented by software objects which are responsible for creation and potential progress of audio events. The behaviour of the objects is dependant on the user movement and the behaviour of the whole immersive environment.

Sound objects are described by various parameters: coordinates in the space, number of the sound sources and their movement, a voice – audio generative structure – assigned to the sound object, conditions of start and end of the voice, priority flags, control signals for the voice, an algorithm of response to control signals, probability of occurrence of the given sound object, mixing and processing group, type of sound processing.

One sound object may have several different voices assigned as responses to various situations: an object appears, the object is close to the user, the object is on the user side, the user is looking through the object, the user stares at the object for longer time, the user moves the head, the user is in a specific spot, or current scene parameters reach specific values.

The sound object may occupy a defined segment of the space – it is heard in that place only: a line segment, a plane, a sphere in the space. It can be attached to a graphic object – static or moving, or to a long graphic object with a sound sliding along the closest point to the user.

A special type of sound object can be attached to the graphic object – that sound object doesn't generate any sound itself but it changes parameters of processing of other sounds existing at the moment, when the user look towards that graphic object (e.g. the sun and the moon "changed" reverberation time of the whole scene in *Ephemera*).

Another special type of sound object was used in *Placeholder*. It could record the user voice and than play it back when the user touched the graphic object it was attached to.

2.2.6. Processing and mixing

Selected voices were localized in the space. They were binaurally processed according to control signals of azimuth, elevation and distance of a sound source in reference to the user head.

Localized sound can be used for sound sources attached to the graphic objects or invisible objects, simulation of extensive sound sources (a line, a plane), movement of the sound sources in the space or around the user head.

Localized sound adds an audio feedback to the user movement in the space. On the other hand localized sound is very interesting itself through its potential of movement and creating different acoustical spaces.

An additional audio processing (like chorus, delays, and reverberation) was also used at the level of individual sounds or groups of sounds. In *Placeholder* piece live voice from a microphone was processed when the user had changed to a critter – there was a delay introduced by processing that made people talk in different way than normally.

All the layers of sound are mixed for each user in real time what creates a unique, often unforeseen soundscape.

2.3. Data exchange between audio space control and audio generation module

Several start/stop triggers and continuous control signals must be sent to the generation module. There was a simple MIDI-like protocol[8] used in most of the projects. But it could cause sounds to be interrupted because of lack of information exchange. In *Ephemere* a special hand-shake protocol was used. It allowed keeping track of realization of audio commands and preventing an attack and a release of sound from cutting and wrong processing.

2.4. Audio generation module

The latter module of audio space control creates control signals for an actual audio output generation for predefined voices. The voices are triggered on and off and modified according to specific parameters. As a result the module sends a stream of data (MIDI or program variables) to samplers, processors, and mixers to produce sound output.

The MAX or ISPW MAX languages were used for the generation module in most of the projects. In *Ephemere*, custom software written by the composer Rick Bidlack was used. He also created the compositional algorithms for each voice we hear in *Ephemere* and *Osmose*.

2.4.1. Sound material

In many projects sound material consisted of pre-recorded samples (e.g. environmental noises, animals, female and male voices, viola sounds) and synthesized sound (mostly based on sine tones and noise). I consider those types of material as the most suitable for the interactive soundscape.

Sometimes also long soundfiles were used for narrative pieces like *Archeology of a Mother Tongue*. But long soundfiles give an impression of repetition and they are not as responsive as short samples.

The user voice was also introduced into some pieces. In *Placeholder* piece the users' voices could be recorded into special "voiceholders" and then re-used as an additional sound material through playing back by other users.

However there is no control over the user voice what may cause undesirable effects.

2.4.2. Sound composition methods

Each voice is represented by a program routine that is invoked with a number of parameters in response to the user or the environment behaviour. The voice program routine consists of some algorithms for sound transforming on an "instrument" level (selecting a region of a sample file, changing volume and pitch envelopes, changing parameters of sound synthesis algorithms, changing parameters of various processing) and

[8] A serial protocol with high bit set for commands and high bit set to zero for data.

some algorithms for sound composition on a "score" level (playing sequences of notes).

All the algorithms should allow for a fast change of the sound output in response to parameters change to achieve responsiveness of the soundscape. It is very important that the algorithms take an absence of control signals into consideration when the user doesn't move.

The composition algorithms apply to all sound components, not only to music and ambience layers but also to short sounds, as a way to achieve enough variation of a voice, each time it is played in the environment.

When I worked on audio generation I mainly used a method of adding some random or pattern-based jitter to various parameters of sound algorithms (within a specific percentage). Much more sophisticated compositional methods were used by Rick Bidlack in *Osmose* and *Ephemere* – he has used chaotic systems as generators of note events and other variations in sound parameters both for music and all sound voices.

3. A virtual audio space

It is possible to use a pre-recorded piece of music to accompany an interactive experience to enhance emotions of a graphic environment. But much more interesting is to look for new ways of working with virtual audio space, to go beyond the means of expression used for linear sound composition.

3.1. The impact of related fields

There are several fields related to sound and space that have a potential to contribute to the process of creating virtual audio space: phenomenology with the body movement as a main source of our knowledge of the environment (in virtual space the user movement "creates" the visual and audio space), architecture and urban design with their intended influence on people behaviour in space and in time (one person experience in virtual space allows to concentrate more on our own perception of the space; speed of the user movement changes what can be seen and heard in the space), psychology of hearing with principles of perceptual organization and the role of attention (user gesture of attention in virtual space may change the soundscape according to principles of perception, what normally happens in our head), phenomenological aesthetics with music as an intentional object (each fly-through the environment - unique, strongly dependant on the user behaviour – can illustrate the process of active listening), rhythmics and dance with strong relationship between music and movement (virtual audio space may become an instrument for a dancer).

3.1.1. Music and space

In the 50's and 60's space gradually became an element of music composition, and not only the performance attribute. There were a lot of new ideas about space in the composition and about new role of the listener: a spatial separation of music layers by musicians placement, space intervals, sound objects, wall of sounds, an immersion of the listeners "inside" the music, an active role of the listener through the movement in the space (various aural perspectives) and on perception level (shifts of attention between the sound layers). Those ideas could be now recreated in the virtual audio space.

3.2. An attempt to specify new means of expression

Behaviours of sound objects were thought up in close relation to artistic concept of the pieces, to support its content through a specific sound response to the user and the environment. New means of expression for virtual audio space can be defined through exploring that sound response. That issue will be considered in connection with several examples of audio space composition and interactions.

3.2.1. An attention

Ephemere: if you move fast you hear fewer of localized environmental voices and more of your body sound (you can hear more if you pay attention to the environment). That response rewards the user for the attention.

3.2.2. A localized sound

Ephemere: there is a rock object that calls you if you are passing by but not looking at it (you can hear invisible sounds that drag your attention and you look around).

"Placeholder": there is a localized sound of another scene coming from a portal you can use for a transition.

These are attempts to influence the user behaviour.

3.2.3. A viewpoint

Ephemere: you can hear a continuous sound of a group of flying objects ("a flock") when you see it, but if you don't see it the sound changes (the behaviour of a creature may change if you do not look at it).

3.2.4. New rules of perception, audio sub-spaces

Crossing the border: if you look at the border you are in its sonic space, if you turn around to face the suburb - without changing the place - you hear the suburb only (it is like a gesture of closing your eyes, turning away when you don't want to see - those audio sub-spaces go against the rules of perception). This is an attempt to create an emotional soundscape to express emotions in that near-border zone where people want to forget about the border and have normal lives.

3.2.5. A concentration

Ephemere: if you get closer to a special rock object with a localized voice playing, and spend some time listening and watching it, the rock will change and start to play its internal sound, while other sounds (and images) disappear, and then you are surrounded by the rock image with the sound changed from localized to stereo sound that you carry with you for a while - any fast movement can break the process (you need to stop and spend more time to get familiar with an object or a person; if you start to concentrate on some sound or a thought, all the other sounds form a background and you stop hearing them).

That response to the user concentration emphasize that you can discover more being gentle in the environment. It also tries to show the changes of perception when the user is immersed in thoughts.

3.2.6. Memory

Ephemere: if you get onto the river it may carry you to the next scene, each time you move to another scene with the river, you carry one layer of sound from the previous scene and you can hear sound of two scenes together [you can collect sounds from different scenes]. This kind of transition tries to explore memories of past events.

3.2.7. Light

Ephemere: when you look towards the sun or the moon all the sounds you hear become more reverberant. It is a way to emphasize the perception of the light of those objects.

3.2.8. Processing

Placeholder: you can talk but your voice got changed when you had become one of the creatures, you can also hear another person with changed voice within the environment. This change tries to influence the user behaviour.

3.2.9. A surface

Osmose: if you go below the earth surface you suddenly stop hearing the sounds from above. It is an attempt to "show "the surface in sound since there is no solid graphic surface and you still see what's above.

You can explore the capability of externalizing our perception, work with the difference in aural and visual perception, or change the rules of perception for a particular project. There are possibilities to work with responsive audio space based on changes of acoustic properties of space, foreground and background sounds, sense of distance. The audio space may have some layers or sound objects to be discovered, making the experience of the space more personal. Using mono, stereo and localized sound on the headphones gives variety of placing sounds in the space.

4. Project workflow

An interactive project involves collaboration of a team of people who have different skills. Usually there were more people working on the graphics than on audio part of the project. Char Davies decided to have two people on audio and two on visual side what was an important change.

It was a production situation. We had to make things happened[9] so we needed to use some intuition first, and then analyze the projects later on. This was a new area and production practices had to be improvised. And I think it may stay this way even now since the artists don't want to work within pre-programmed templates.

4.1. Preliminary project of an interactive audio space

In most cases a preliminary audio project had to be done without any working version of the visual space. Many interactions between audio and visual spaces were discussed on the paper and through the real movement.

4.2. Collecting an original sound material

Sound material must be collected based on the preliminary project since it takes much time to do sound recordings, to search pre-recorded sounds and synthesis algorithms, and to organize the material within the sampler programs or a database.

4.3. Programming

Each of the sound objects or sound layer must be programmed - it always takes time to present an audio idea to an artist (similar situation applies to visual part). Artists often are not aware what is possible or what is hard in terms of programming and the time required for a specific task can be underestimated.

4.3.1. A rehearsal mode

This is a very important part of a program for the working process (but there is often no time to program it)[10]. It is necessary to be able to start the piece from various scenes (in space and in time) not from the beginning only, and to force the time forward in time dependent scenes. It is important to work with one sound objects at a time in the environment, to rehearse its behaviours before putting the scene together (various variables and the position of invisible sound objects should be printed in the space with the object, not just on the monitor). The rehearsal mode should include a method to initialize all the sound routines ("a reset button") and an option of simulation of control signals (from the sensors and the environment), for an off-line work on audio space without the whole visual system available.

4.3.2. An interactive "post-production" process

Production of audio part of the project overlaps with visual production. Ideally the graphics should be almost finished before audio postproduction but some interactions require working on both parts together. Also after adding sound objects, the graphics might require some timing or space corrections[11].

Sound turned out to be a debugging method for the whole program. A graphic object with sound attached could be traced in a scene and transition by listening to it - all abrupt cuts in the sound meant a bug.

The final version of the project should be well tested - it requires long-term test as a preparation for many days of running the piece in the gallery.

4.3.3. Program backup

There is often not enough time to finish everything before the opening and some work is done in the gallery while setting up the piece. So the final backup of all data must be done over there. It should be possible to restore all the software, data and connections in case of failure of the equipment, based on that backup.

5. An exhibition in an art gallery

The immersive projects were shown in various places: in a project studio, in a big conference space, even in a shopping mall, but mainly they were exhibited in art galleries.

It takes few days for an installation of a complex project including all the equipment setup and an acoustic adaptation of the space (e.g. a carpet and curtains on the walls[12]). The latter is very important since art galleries were not built for audio art (there are very reverberant rooms with high ceilings, there is a noise of ventilation and noises of audience steps on the floor, there is not enough acoustic isolation between rooms with different sound installations). It is also worth checking some audio details like good labelling (left/right) for audience headphones (sometimes the label should be visible in a relatively dark room) and an information for audience that what they hear is a real-time generated sound in response to a person in the helmet (most people assumed a pre-recorded "soundtrack").

It is useful to keep a similar setup for more projects if you plan to exhibit them together. This way *Osmose* and *Ephemere* were exhibited together on the alternate days.

It is necessary to make training about the setup (including sound check routines) and emergency procedures (in case of a crash) for museum technicians and attendants (who help the users). The procedures of reloading the system should be as simple as possible – detailed instructions must be written for the gallery (it is hard to find time for that).

An art gallery needs to book time slots for people for an individual experience of the virtual space – this is a main reason to introduce a 15 minutes time limit per user which affects the overall time structure of the piece[13].

6. An interactive art lifetime

For complex projects, it might be difficult to gather again all the equipment after the first exhibition. This happened to big installations (*Archeology of a Mother Tongue*, *Placeholder*) that required great amount of equipment and was shown just once in the Banff Centre for several days only. *Inherent Rights, Vision*

[9] When a proper hand-shake connection between visual and audio programs hasn't been done on time, John Harrison and I had to prepare overnight a simple MIDI-like serial protocol to start working with the artists on scheduled projects.

[10] The most advanced rehearsal module was written by John Harrison for *Ephemere*.

[11] MAX language I used for sound programming allows watching variables and making changes while running the piece.

[12] These are the elements from both *Osmose* and *Ephemere* exhibitions – people entering that subspace of the gallery could hear the difference.

[13] In case of earlier projects there was no limit on duration of the project so an attendant had to inform the user about the end.

Rights was prepared for an exhibition by building a special "kiosk" with a viewer, and Bar *Code Hotel* got ported to more compact setup. This is an important issue to consider while deciding about the equipment, to be able to exhibit the work[14], especially as interactive art involves fast changing technology that disappears.

6.1. Archiving of an interactive project

There was no method of archiving interactive projects other way then on a video tape. At present most of the projects exists as memory and as a video documentation. I've been also recording sound of several flies-through the pieces to save those ephemeral soundscapes.

Unfortunately no tape will grasp the relation between the users and space what makes it difficult to evaluate the piece.

People, who make a documentary based on a videotape of the piece, should be aware of a connection between audio and visuals in real-time creation process, and just leave it in sync. I've seen some documentaries with "enhanced" and re-edited audio that had nothing to do with the actual piece. So the piece might be only debated …

7. Conclusions

The virtual environment can be considered as a metaphor for perceiving, discovering and affecting real environment. The response of the virtual environment to the user behaviour can demonstrate the way we perceive the world and its aural space. It gives a lot of possibilities to use virtual sound as an aesthetic medium and as a means to explore our perception, aural illusions, current and historical soundscapes.

The perception of virtual sound must be learned in the same way people have learned cinematic sound, to comprehend behaviours of spatial audio. In case of exhibition, an audience should be informed about real-time virtual sound to bring its specificity to people attention.

The last project I worked on, *Ephemere,* is a very important piece for me. For the first time we have managed to realize many of my sonic ideas for virtual environments. *Ephemere* contains some layers of interactive sound that require time to be discovered and maybe they are still waiting to be uncovered.

Acknowledgment

My experience comes from the work I've done in the Banff Centre for the Arts and from the work for Char Davies in Montreal (supported by Softimage and Immersence). There were a lot of people in the Centre who I worked with so in symbolic way I would like to thank to Kevin Elliott who has assigned me to a strange sounding *Sense8* project which turned out to be the beginning of my adventure with virtual audio spaces. I wish to thank to John Harrison, the virtual environments programmer I worked with in all projects, who helped me a lot and was always open to new ideas, and to Rick Bidlack, the composer and programmer, who introduced me to computer music and was very supportive. Finally I would like to thank to Char Davies for inviting me to work on her two complex pieces when I had a chance to learn her aesthetics and use it in my work.

Selected projects

[1] Char Davies - *Osmose* (1995), *Ephemere* (1998/2001)
www.immersence.com
John Harrison - custom VR software,
Georges Mauro - computer graphics,
Rick Bidlack - sound composition/programming,
Dorota Blaszczak - sonic architecture/programming.
[2] Perry Hoberman - *Bar Code Hotel* (1993),
[3] Toni Dove, Michael Mackenzie - *Archeology of a Mother Tongue* (1993),
[4] Brenda Laurel, Rachel Strickland - *Placeholder* (1993),
[5] Lawrence Paul Yuxweluptun - *Inherent Rights, Vision Rights* (1992),
[6] Pauline Cummins – *Crossing the Border* (1991)
[7] Tim Brock - *Kandinsky* (1991)
[8] Roger Doyle - *Babel* (1991)

References

[1] Schaffer, R.M. T*uning of the world,* New York, Knopf (1977)
[2] Blauert, J. *Spatial Hearing,* Cambridge, The MIT Press (1983)
[3] Moore, B.C.J., *An Introduction to the Psychology of Hearing,* London, Academic Press (1988)
[4] Moore, F.Richard, *Elements of Computer Music,* Englewood Cliffs, Prentice Hall (1990)
[5] *Virtual Seminar on the Bioapparatus,* The Banff Centre for the Arts (1991)
[6] Bidlack, R, *Chaotic Systems as Simple (but Complex) Compositional Algorithms,* CMJ, Vol.16, 33-47 (1992)
[7] Bidlack, R., Blaszczak, D., Kendall, G., *An Implementation of a 3D Binaural Audio System within an Integrated Virtual Reality Environment*, Tokio, ICMC Proc. (1993)
[8] Rowe, R., *Interactive Music Systems,* Cambridge, The MIT Press (1993)
[9] Harley, M., *Space and Spatialization in Contemporary Music: History and Analysis, Ideas and Implementation,* Ph.D. diss., Montreal, McGill University (1994)
[10] Moser, M. A., MacLeod D. (Eds.), *Immersed in Technology: Art and Virtual Environments,* Cambridge, The MIT Press (1996)

Exhibition catalogues

[11] *Land Spirit Power,* Ottawa, National Gallery of Canada (1992)
[12] *Char Davies, Osmose,* Montreal, Musee d'Art Contemporain de Montreal (1995)
[13] *Electra 96,* Hovikodden, Henie Onstad Kunstsenter (1996)
Serious Games, Art Interaction Technology, Newcastle-upon-Tyne, Laing Art Gallery (1996); London, Barbican Art Gallery (1997)
[14] *Char Davies, Ephemere,* Ottawa, National Gallery of Canada (1998)
[15] *010101: Art in Technological Times,* San Francisco, Museum of Modern Art (2001)

[14] *Osmose* and *Ephemere* have been exhibited again this summer – visual part of the project has been ported to a new computer, there are two sets of equipment and some spare parts. *Bar Code Hotel* is now in Media Museum of ZKM.

SoundTableTennis – an interactive sound installation

Prof. (FH) Hannes Raffaseder,
St. Poelten University of Applied Sciences
Matthias Corvinus-Straße 15,
3100 St. Poelten, Austria
hannes.raffaseder@fh-stpoelten.ac.at

Abstract. This paper discusses some of the major problems involved in the design of interactive sound installations: interface-jesign, realisation with limited budgets, reliability and stability, parameter extraction, mapping and sounddesing. As one possible solution to these problems the approach of SoundTableTennis is presented both form an artistic and a technological point of view.

1. Introduction

SoundTableTennis is an easy-to-use interactive sound installation which was built with little money and without any sophisiticated technology, which was originally commisioned by the so called Klangturm[1] (engl. sound tower) in Sankt Poelten, Austria. As a museum for sonic arts and interactive media in Lower Austria the Klangturm welcomes up to 350 visitors a day. The installation was on show for six months between the end of April and the beginning of November 2005 amd more than 25.000 people had fun playing SoundTableTennis.

Figure 1: Klangturm St. Poelten, Austria

2. Problem Definition

Interactive media installations play an important role in current art practice and are becoming more important for advertising, entertainment and business events and shows. But apart from some outstanding festivals of media arts visited by specialists from all over the world, the majority of users face similar problems again and again.

2.1. Interface Design and Usability

Usability and User Interaction are amongst the most important issues. Especially in museums and art galleries people are used to the sign “Please don´t touch!”. Often the interface is far too complex or the reactions of the system is to sophisticated. For these reasons interacting with a media installation often takes an effort to overcome one´s inhibition and there is a need for easy-to-use interfaces. But at the same time a certain amount of complexity is important for the interface to be exciting and make people curious to explore and to play with the installation. Solving this conflict between simplicity and complexity is propably one of the most important steps on the way to succesful media-installations.

2.2. Limited Budget

Designing an interactive media installation, artists usually face another problem as well: In the arts you can do whatever you want, as long as you do not need any money!“
Usually artists have to realize their ideas with low or even no budgets. Of course this is a crucial problem for the realisation of interactive media installations, as not only several people – artists and designers, computer scientists and technicians – are involved in such projects, but often expensive technology is necessary as well.

2.3. Reliability and Stability

“Sorry, this installation is out of order!”

Interactive media installations have to work without interruption over a longer period of time. But as everybody knows from his or her own experience, computers crash and software has bugs. That is why it is a major challenge to construct reliable and stable interactive installations. To find a solution to this problem seems to be even more difficult within the field of media arts, as only brand-new technology, software and sensors are what the nerds (the experts, the gallery owners, the journalists,…) call “cool”.

3. Basic concept

The basic concept for solving these problems was to use a well-known and easy to handle game as interface for an interactive media installation. Starting with SoundTableTennis, users have to do no more than play “Ping Pong”. Thus no sophisticated explanations are necessary for interacting with the installation and usually people really like to play the game.

Besides the usability, the number of parameters for interaction is defined by the game as well. This fact guarantees that on the one hand the system has a sufficient degree of complexity to ensure

[1] http://www.klangturm.at (last visit 24.08.07)

a wide-spread variation in the artistic output. But on the other hand, it is still rather easy to handle.

For SoundTableTennis the common rules for table tennis are changed slightly: rather than compete with each other, users must collaborate to play the ball back and forth as regularly and to keep the installation alive for as long as possible.

4. Technical realisation:

SoundTableTennis is realized with very simple means to fit the often limited budgets for artistic media installations and to work in a stable and durable fashion.

4.1. Hardware

Six contact microphones, so-called drum triggers, like for instance the Yamaha DT-20, are used as sensors mounted on the back of a ping-pong table.

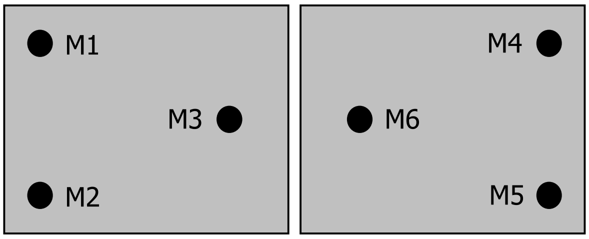


Figure 2: Position of six microphones

To analyze the signals of these microphones they are connected to a standard PC (P4 / 2,4GHz / 512 MB) via a soundcard with at least six inputs and two outputs, like the M-Audio Delta 1010LT PCI soundcard. For the audio output of the interactive system two active loudspeakers are needed.

4.2. Software

The software was developed using Native Instrument’s Reaktor[2], a modular programming environment optimized for digital sound processing. Distinguishing between the so-called “Panel Window” for the User Interface and the “Structure Window” for the development of a user application, NI Reaktor offers a very good compromise between complexity and operability. In the structure you choose and connect so-called “modules” from a huge variety of possibilities, like very simple mathematical operations as well as more complex filters, samplers, oscillators.

Equipment		**Price** (in Euro)
HW	Ping-Pong Table	200.-
	6 Trigger Microphones (Yamaha DT-20)	140.-
	PC (P4 / 2,4GHz / 512 MB)	600.-
	Soundcard M-Audio Delta 1010LT	195.-
	2 Genelec 8020	550,--
SW	Native Instruments Reaktor 5.0	380.-
	Total	**2065.-**

Table 1: Hard- and Software used for SoundTableTennis

Some other modular software environments for interactive processing of digital sound and media, like Max/Msp[3] from Cyling 74, the open source software PD[4] (Pure Date) or Meso´s VVVV[5] (beta version for free download) offer even more possibilities and more flexibility compared to Native Instrument’s Reaktor. But more detailed knowledge in digital signal processing is needed for these tools. Also the design of the user interface of the specific application is easier with NI Reaktor in most cases.

5. Parameter Extraction

The six contact microphones are arranged on the back of the ping-pong table as shown in Figure 2. The envelope of each of these signals is detected.

The signal levels of these microphones are compared by means of quite simple mathematical and logical operations to extract the following parameters for interaction:

- Hit
- Tempo
- Position
- Number of Hits
- Start
- Stop
- Next

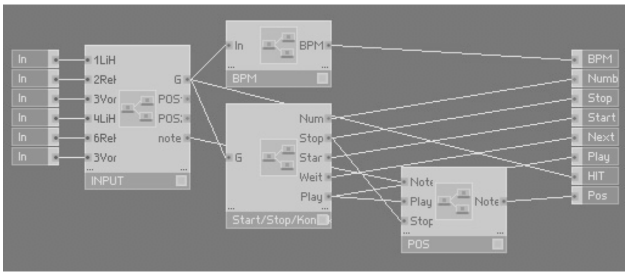


Figure 3: Parameter Extraction

5.1. Hit

If the average level of all microphones rises above a certain threshold, the ball hits the table and a Gate-Impuls is generated.

5.2. Tempo

The time between two hits is measured and converted into BPM (beats per minute) to control musical tempo.

5.3. Position

Comparing the levels of the six microphones, it is possible to detect eight different positions of the ball hitting the table, which seems to be sufficient for this application.

If for instance the level of Microphon 1 is bigger as the level of Microphon 2, the ball hits the table on the left side. If the average level of Microphon 1 and Microphon 2 is bigger as Microphon 3, the ball hits the table on the back side.

5.4. Total Number of Hits

The total number of hits is counted as well as the number of hits on each position.

[2] http://www.native-instruments.com/index.php?id=reaktor5 (last visit 24.08.07)

[3] http://www.cycling74.com/products/maxmsp (last visit 24.08.07)

[4] http://puredata.info/ (last visit 24.08.07)

[5] http://www.vvvv.org/ (last visit 24.08.07)

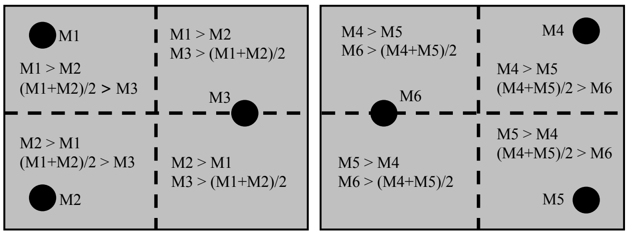


Figure 3: Eight Positions comparing
the levels of six microphones

5.5. Start

As people should play to getter and the shouldn´t be a sound by mistake, the sound-installation starts only after two hits.

5.6. Stop

If there is not Hit for a certain amount of time, SoundTableTennis is muted.

5.7. Next

This parameter indicates a certain amount of hits. For instance "Next" can indicate every eight hits and enforce the use of other sound-samples to ensure enough variety in the sounding output of the installation.

6. Mapping

When developing an interactive installation, decisions must be made regarding which interaction parameters force which reactions of the system. This process is usually called "Mapping".

6.1. Compexity vs Clarity, Variety vs. Prediction

On the one hand, sufficient complexity is needed to ensure variety in the artistic output of the system and to keep the installation exciting for users over a long period of time. On the other hand, the allocation of the user's action to the system's reaction has to be clear and easy to understand. Users have to be sure that they can influence the system's output. Otherwise they will not play with and explore the tool. If the mapping is too complex, users soon get the feeling that the system's output is changing just by chance. For this reason the system's reaction to a certain user action has to be predictable at least in some respects. Dealing with these tradeoffs between complexity and clarity and between variety and prediction is a difficult task.

6.2. "Not the instrument, the music is interesting!"

Media artists should always be aware that the output of an interactive installation produced by certain actions of the user, not the installation itself has to be considered as most important. If you take, for instance, a piano, which of course can be considered as an interactive tool, it is absolutely obvious that people want to listen to the music played on that instrument. But within the field of (interactive) media arts it is far too often only the newest technology used in an installation which surprises and attracts the attention of the users. In most of these cases the output turns out to be rather boring and people lose their interest very quickly. In contrast to the piano, users do not practise the use of interactive devices for several years. That is why the artist has to ensure that the output of an interactive installation is interesting, whatever action is taken by the user. Again this problem has to be solved within the process of mapping.

6.3. Mapping for SoundTableTennis

As everybody suspects, SoundTableTennis triggers a Sound Sample every time the ball hits the table. To guarantee enough variety, each position triggers another sound and the position-specific sound is changed after it has been played eight times. As the sound is produced by the Ping-Pong Game, the Samples are designed to be percussive.

In addition to this kind of sound material, more rhythmical loops are used. It is obvious that the tempo of the ping-pong controls the tempo of the drum loop. The selection of a specific loop is also affected by the tempo: Slow play is accompanied by atmospheric sounds, more rapid play causes faster drum beats in techno style. After the ball has hit the table eight times, the loop must be changed to ensure variety. The longer people can play, the more sophisticated the music becomes.

Parameter of Interaction	**Reaction of the System**
Hit	Triggers a short percussive sound sample
Tempo	Determines tempo and mood (atmoshperic sound or techno loop) of the drum loop played in the background
Position	Determines the specific sound sample
Total Number of hits (Next)	Changes the loop in the background
Number of hits on one position (Next)	Changes the position-specific percussion sound

Table 2: Mapping

7. Evaluation

SoundTableTennis was developed between February and April 2005 for the Klangturm in St. Pölten. Within six months there were no crashes or any other technical problems. As the installation was meant to be built by rather simple means, the overall budget was only around 2000 Euros. As some equipment (loudspeakers, NI Reaktor) was already available at the Klangturm, even less money was needed.

The interactive sound installation was used by up to 350 persons a day between 29th of April and 1st of November 2005. As most of the visitors of the Klangturm had great fun playing with the installation, we can conclude that the use of a well-known game as an interface together with well considered and comprehensive mapping seems to be a good choice for an interactive sound installation.

The body as transducer: building narratives through gestural and sonic interactivity

Nina Waisman 168 West Glaucus Street, Encinitas, CA, USA, 92024 waisman2@cox.net
http://www.ninawaisman.net

Abstract. The interactive sound-and-sculpture installations I've made over the past few years highlight the negotiation of bodies though sensor-mined sonic environments, exploring the impact these negotiations have on the generation of meaning. The work underscores the interaction between physical, sonic and logical choreographies, and the ways these choreographies are altered by the presence of "others". These "others" can be social, physical and/or technological. We humans have always been transducers of the energies presented to our bodies. What limitations and additions to the grammar of transduction do today's interactive spaces produce?

1. Transducer

transducer, n. a device that is actuated by power from one system and supplies power usually in another form to a second system. [1]

2. Quinine

(Video: http://www.ninawaisman.net/quinine)

Quinine Waisman, 2004 photo © 2004 Steven A. Heller

In *Quinine*, five interactive sound-and-sculpture stations pair gesture with language and the sonic distortion of that language in order to raise questions about the dominance of language in the face of other forms of power transduced by a body. I will analyze a couple of these interactions closely here, to explore how a body might perform as a transducer.

In *Quinine*, you enter a room with a slightly haunted, but not unpleasant soundscape. Your entry draws a sound out of the murk and begins to bring it in focus. At some point, you might realize the sound is a word. As you continue to walk, your movement into the room alters the sound's pitch and rhythm. You can mold the sound or be attacked by it; the sound is distorted by your presence. You might eventually climb a small staircase, swivel in a chair, etc., and, in each case, your movement alters the delivery of a different spoken word. The sound you hear when you are not interacting is simply the five central words with pitch extended so they sound like wind. (Note: The windy sound is unfortunately distorted in the video; it is much subter in the actual installation.)

Quinine explores the slippery relationship between language and experience, experimenting in particular with the performative aspect of communication. Language is presented as a sensory experience; experience/activity is offered as the source of occasionally articulated words and ideas. There are grammatical phrases to be made from the five words driving the piece, but most visitors succumb to experiential combinations of behavior and sound built into the interactions.

The first interactive "station" in *Quinine* is quite transparent in its presentation. In it, a proximity sensor, the sort used to control sliding doors, is sitting plainly on a plinth. This station uses the most elemental method of displaying a sculptural object: put art-object on plinth so art-audience can approach, circle, and consider said object. But here, rather than sitting mutely for a visitor's contemplation, the sculptural object considers the visitor, audibly acknowledging his or her style of approach, imposing a new channel for manipulation of a visitor's physical behavior and attention. The sensor-plinth sculpture is set at the end of a 20 foot long, subtly blue-lit area. The lighting creates an approach path to the sensor (this is hard to discern in the photo-documentation). Those who walk the lit path control that particular sound space. As you enter the path, you hear some of the windy sound gradually increase in speed and pitch until a rhythmic pattern becomes clear. As you get closer to the sensor, that rhythmic pattern slowly resolves into a normal pronunciation of the word "is". If you get very close to the sensor, that "is" increases in speed and volume so that it seems to quickly and aggressively punch out intonations of "is" aimed at your chest. This creates a physical sensation. I bury a large Bose speaker in the shredded newspaper right behind the sensor so the quick bursts of "is" are felt as subtle compressed punches of air by those who come close.

I wanted this first station to address someone while they were doing something quite natural - i.e. walking into the room and approaching the first object in the space. I saw this as a kind of announcement that our simplest acts can be fodder for digital manipulation, for unexpected acts of transduction. This manipulation is a little creepy, but curious enough to make most people willing to participate as they learn what control they might have over the experience. A different form of manipulation molds the interaction at each station in the piece. I bury a single large speaker in the newspaper next to each sculpture-station. In this way, you, the visitor, become more aware of the relationship between your particular movements and the sound generated next to you. However, you are not alone in this installation and the sound these powerful speakers make draws the attention of other visitors towards you. Your private gestural interaction become public, as others turn towards the sound. This makes participation in the piece pointedly social and performative, in a way it isn't always in artworks.

How then is the body functioning here as a transducer? What powers are being translated by a visitor's body, and what power does that body then provide to the global system? The most obvious force powering a body at this station is sonic. As the sound waves enter the body in response to its motions, the body responds by avoiding or courting these waves. The sonic waves might be transduced by a body into the force of curiosity, driving a visitor further into the piece. Or perhaps it is the desire for control that is actuated by the comprehension that the physical waves penetrating the body can be manipulated. Or maybe there is a physical attraction actuated when a body understands the sound to be unthreatening, particularly as that sound resolves itself into rhythm. As a visitor triggers this sound, others in the room often turn to consider the sound source, i.e. the behavior of the person triggering the sound. The transducer in this case suddenly becomes a watched performer and this role functions as a stimulant for some or a discomfort for others. The sonic energy at this point is transduced by a body into social dynamics centered around exhibitionist energies and bodily confidence. Symbolic/logical transduction also enters into the equation with grammatical recognition of the word pronounced at each station, which may in turn be translated into symbolic or metaphorical powers of understanding.

One of the key questions in *Quinine* swirls around control. Who is in control? Whose power is being actuated? Globally, it might seem I am in control - I have programmed the electronic transductions of visitor presence into sonic response. I have created a world of objects placed deliberately to encourage particular behaviors and hopefully evoke consideration of particular issues. But people are not robots and they don't stick to a script. They find ways to play with the system, to hack it, and therefore develop their own meanings. Their bodies and minds are not passive transducers, but actuators that supply feedback to an evolving system. If they find a sound or a gesture-sound combination interesting they will play with it. Others, if attentive, notice this play and respond to it by playing with the sound at their particular station. The system shifts then, from one that might be read to one that can be played. To be fair, the possibility of play is to some degree programmed into the system, but it is not confined. Meaning in the piece derives from a visitor's gestural and sonic experiences, which generate individual contributions to an overall performance. Narratives are constructed through each visitor's particular melding of the linguistic and experiential elements present in the room at a given moment. Thus the question of what it means to have the word "is" punched at you in the middle of considering an artwork becomes understood not primarily through a grammatical consideration of the word, but rather through feel and social interaction; by how the body transduces the sound, molding and reacting to it rhythmically and physically in the midst of the larger landscape of the group performing *Quinine* at that time.

The words attached to each gesture in *Quinine* are deliberately quite open and imprecise. The first "is" is followed by the word “this”, which plays at the staircase. The staircase is an almost iconic stepping platform in the mode of some minimalist sculpture from the 70's. The form evokes a pure idea of a stair. But it is made not of the minimalists' reductive materials. Instead it is clad in a material that announces its cultural inflection – shiny, diamondplate steel which is fetishistically used for truck detailing in the US. Visual cues are used throughout the piece in this way to undermine ideas of "pure" forms, just as ideas of "pure" linguistic meaning are consistently called into question. And, of course, this staircase doesn't sit quietly on view, but tempts you to consider walking on it. When you do, it responds by playing a range of synthesized ditties made from a voice speaking "this” – a different one is played in response to each step you take. The idea of a particular "this" dissolves into multiple "this-es" that can be reordered or heard simultaneously by moving on the steps in atypical patterns. Again, meaning is made not by tuning an audience's attention to the grammatical sense of the word, but rather to the simultaneous transduction of the sound of the word, its meaning, and the movement and position of the body. Meaning is not simply linguistically determined, but highly impacted by the body's engagement with the pronouncement of words, and with a body's play with a particular sound-gesture combination within a multi-user performative sound space.

A starting point for *Quinine* was my own experience of ideas and language surfacing seemingly unrelated to physical actions I performed as I moved through a given environment. Where did the ideas come from? Was it the social construct of language imposing these thoughts, or, more fantastically, could environments and inanimate objects somehow be talking through me? Was it my movements that shaped the ideas? Or were my ideas more likely a hybrid soup based on the ever changing mix of symbolic and concrete inputs passing through me? These questions arise even without introducing technology into our environments. But now that we have spaces that read and respond to us, pointedly introducing "appropriate" stimulus (advertisements, lighting, etc), the thought-making soup becomes more complex: on the one hand more surveillant, on the other hand, new modes of communication & expression are afforded.

Brian Massumi recently introduced a model of the body as transducer in his book *Parables For The Virtual* [2], describing the motors for players movements during a soccer game. Massumi sketches a scene populated not by bodies, but by "part-objects" of the soccer game:

> The body figures not as an object…but as a part-object, a conversion channel, a transducer...[3]

In Massumi's soccer match, every gesture of an attentive player is charged by the game’s goal. Each body is addressed not as a whole, but as body parts that might or might not be able to achieve the goal. Each “part-object” senses its global role in the field of attractors and responds by transducing the game's global tension into an action that is an expression of the game, not of the individual. A foot in motion (and thus the body connected to the foot) is driven by all the particular details of a given moment in a game on a given field - and all those environmental details are charged by the game's goal.[4] The transducing human lives in and through its bodily conversion of energy “inputs” - the goals and stimuli of a given moment. The human is defined at and through exchange between input and output – it is neither a purely inner-mind space nor a purely bodily-material existence. Human existence unfolds in a non-site of exchange. *Quinine* explores this reality as it begins to be populated by new digital players.

3. Around

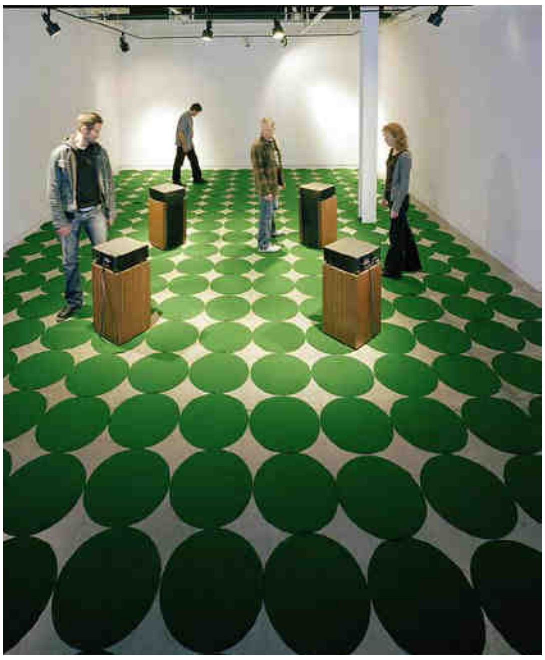
(Video: http://www.ninawaisman.net/around)

Around Waisman, 2004 photo © 2004 Steven A. Heller

In *Around*, I have built an overall choreography that is meant to parallel the spoken words heard in the piece. While in *Quinine*, *individual* gestures and positions are highlighted as transducers/generators of social and narrative meanings, in *Around*, it is the complete time-based choreography performed by the visitor that embodies the linguistic sense of the piece. Linguistic sense is amplified by this physical doubling, but may also be undermined as visitors play the system against a single narrative grain.

Around addresses surveillance through an interactive sound-and-sculpture installation, tuned to make the viewer conscious of her physical and mental entanglement with systems that map meaning onto experience. Searching for meaning implies a hunt for a system through which to organize experience; but if we find a system, are we not mapped or watched by it, to the degree that we allow it to sort our experiences?

Visitors to *Around* move at first in unprogrammed ways, interrupted by words and out-of-sync bursts of room-tone emitted in the space. Some people listen, consider, and exit unconcerned as to why sound plays; they experience *Around* as a fixed piece. Others, sensing that their movements are triggering the words heard in the room, walk carefully on the green felt dots, sure that will trigger more words and meaning. But seeking mastery of *Around* lures visitors into moving about four speakers in hypnotic, seemingly choreographed gestures. The expectation that this mastery will be rewarded with meaning is turned upon itself as the words played reveal the phrase: *it’s around here somewhere*



Like *Quinine*, this installation explores questions of control, but here the choreography is much more tightly woven into the narrative logic guiding the piece. How a person interacts with *Around* dictates the nature of their comprehension. Visitors who move freely about are free of the system and will likely not have their experience controlled by its linguistic narrative. In this situation, words are rarely heard. Instead, the sonic focus is on the long loop of recorded room tone playing continuously against the real room tone, creating a subtle sound-time-space warp. This is hard to pick up in the video, but adds a dizzying quality to the experience of the space. The piece is ismply a suggestion of technology's ability to displace time and space for those players who remain lightly engaged. Those who get more involved in *Around* often move carefully about on the green dots in hopes of triggering sound. Their motions become more constrained by the piece's narrative concern with meaning generation. They pursue the visual cartesian logic of the green dots, moving their bodies like markers along the point-studded x/y axis in an organized attempt at finding meaning. They trigger words on occasion, but generally not enough to put together a phrase. Their experience is more of a hide-and-seek game whose linguistic sense remains more hidden than found. But they generally become self-conscious of both their physical and intellectual hunts for meaning, and perhaps of the relationship between the two.

Another type of response comes from those people who cross paths with the large speakers – they trigger words each time they do this. The triggered text might erupt in the speaker in front of them or in one to the side or behind. If the sound erupts in front of the visitor, they typically approach it. This changes the intonation of the word played from a whisper to more desperate call to an angry declaration, etc. If the sound erupts behind the visitor, they typically turn and move towards it. Arriving at the sounding speaker will trigger another word to erupt from a different speaker. A puzzle-solving (goal-oriented) visitor will chase after the words bouncing through the speakers, and will, as the men in this video do, finish by moving about the speakers in circle. Their walking patterns parallel the circular logic of the piece. I wanted the body to perform the logic of the piece, to move freely when not encumbered with meaning, and to move in an increasingly controlled fashion as meaning takes hold of the experience. The phrase *it's around here somewhere* eventually forms, offering itself somewhat ironically as a reward for hide-and-seek, a reflexive declaration of the search itself.

Balanced against these hermetic underpinnings is an interest in Deleuze & Guattari's proposal in *A Thousand Plateaus* that sound play - chromatics, as they call it - might eventually push language towards a "beyond of language”. [5] As visitors move closer to speakers in *Around*, sound and intonation become quieter; moving further away increases volume & anxiety in the intoned words. With certain motions, visitors can "scratch" words. Some of the sensors trigger recorded sound incidents present in the real room tone. Visitors play with these variables, turning temporal, spatial and linguistic logic into malleable games. Additionally, explorations of the space shift the ground-covering of green-felt sound-absorbent dots about, subtly altering the quality of the sound played in the room as the newly arranged felt absorbs sound differently. These entropic elements suggest freedom in the midst of the programmed experience.

4. Dictation

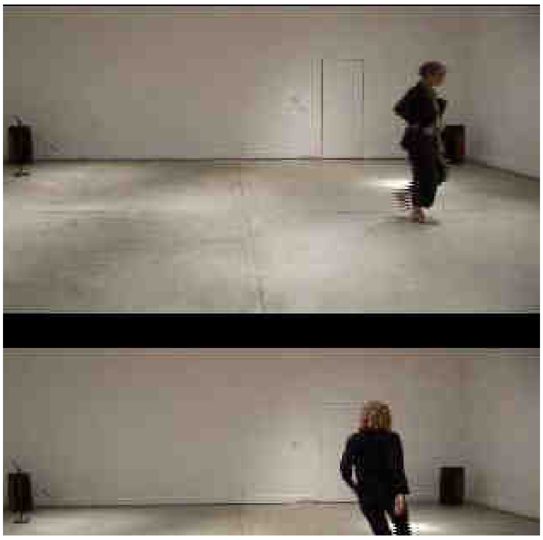
(Video: http://www.ninawaisman.net/walks0107/index.html)

Dictation 4 Waisman, 2006-ongoing

What kinds of control and/or creativity are afforded when something as essential as a walking rhythm is reprogrammed? *Dictation* presents a body's transduction of another body's sound in order to consider the deeper internalization of both control and understanding made possible with common technologies.

Technology increasingly splices disparate effects and minders into our lives, targeting a body's various intake channels for a range of satisfactions. People walk the city paced by individual soundtracks shuffled from commercially produced material, making lives more hip, more metered, more efficient. Our paces are set by virtual rhythms and tracking systems to which we happily subscribe. What new forms of bodily targeting and splicing will be generated as our environments gain intelligence? Your footsteps might be swapped out for a gait and pace meant to adjust your mood or style. You might walk in the steps of your idol of the moment, coming closer bodily to the one you wish you could be. You might download tracks to help you learn the productive movement rhythms of successful figures in your field. Your body subtly gives way, disappears, as you puppet the moves deemed more desirable. Visitors to a mall might be induced to slow down as they pass by consumables by a beat altered to repeat just a hair slower than their natural rhythms. This idea is not new to our culture. Musak has, of course, spent years researching and producing the perfect 8-hour shifts in musical speed to produce a more profitable work/consumer force. But how much better it could be done if gait could be tracked and responded to on the fly!

On the other hand, there is a possibility for a new kind of attunement here. Language, both written and spoken, images, and external observations of physical gestures and qualities are the mediums through which we generally read others. What might we understand of someone through a more direct alignment/attunement with this person's physical state? What if we could inhabit some unique aspect of their bodily rhythms? Will understanding be achieved, or is this just another high-tech way to consume others?



Consider the video *Training 2* [6]. A woman is walking in a large empty space, with a speaker set into each corner. She is barefoot, but she generates the footfall of someone wearing hard-soled shoes, the sound echoing loudly off of the concrete. This continues for a few moments. The effect is briefly, viscerally disconcerting. Soon enough, the viewer realizes this is just an effect. Though the walker's steps appear aligned with the sonic footfalls, a stiffness in her gait early in the piece reveals her work aligning her feet to sound she is not emitting. An occasional glitch in the alignment of her step and the sound then make it clear she is walking to someone else's gait. It's comic perhaps, or simply odd. Her walked patterns respond to the highly regular pace, cutting mostly straight lines through the space as if she is marching or walking a beat. You, the viewer change your impressions of this character over time. You might think of a prisoner in a cell, a cultish follower, a cop on a beat, or a meditative stroller. The space the walker moves in is not small, but the insistent beat of the sonic step makes it feel a bit claustrophobic; the regularity of the pace binds the space and time, generating tension.

As I walk in these videos I become the transducer for the sound of another body – in this case, the recorded footsteps of a Hollywood trope, the typical male authoritative walker from a 40's film. I am tracked by these sonic footsteps played in the room in an odd way. Consider how the footsteps constrain me and make me adjust my rhythms. I must slow down to match the pace of the walker; in so doing I can't help but adjust internally. My mind and body tend to speed along, but both are slowed by this pace, as I adjust to its cadence. Later, as I gain a bodily mastery of the rhythm, my mind is free to move at its own pace, but not entirely. The artificiality of the gait leads me to imagine the other walker. I think of him as a man, with longer legs than mine, who takes more time than I would to move from A to B. One leg is shorter than the other, one receives more weight that the other. I spend much of my time moving between my own thoughts and imaginations of and identification with the body and mind of the unseen walker. I am that man in some way, though at the same time I can't ever be. I am trying to be or to understand someone else, but the method I'm using is quite invasive. The sound waves physically enter and propel my body. Even when I don't think of the walker, my motor timing is altered and this alters my mental state, as it is given its form in part by the sonic and choreographic pacemaking built into my concrete present.

Marcel Mauss in his 1934 essay "Techniques of the Body" [7] made a very strong case, through globally wide-ranging anecdotes, that all physical behavior is culturally inflected. Stance, gestures, posture are sculpted, at least in part, through social imprinting that teaches some behaviors only to prevent others. Hillel Schwartz's Foucaultian essay "Torque: The New Kinaesthetic of the Twentieth Century" surveys the mid-19th through mid-20th western belief that centered, unified, mind-body rhythms acted as the determinants and perfectors of efficient action. That era's analysis and attempted modulations of these rhythms, in the service of increasing mental and physical efficiency, re-formed the fields and forms of dance, industrial design, education, handwriting analysis, etc. [8] Jonathan Crary in his 1999 book *Suspensions of Perception* looks at the late 19th and 20th century obsession with attention "to demonstrate that vision is only one layer of a body that could be captured, shaped, or controlled by a range of external techniques. [9] In brief, gesture and sound have long been employed as mediums of social control, mediums through which a body could be made to transduce dominant ideas of efficiency, centrality, and predictability. But while the visual and linguistic lend themselves well to the past century's techniques of deconstruction in the interest of revealing social control, the indiscrete nature of sonic and gestural impingements on bodies leave this area open to more open-ended, multi-modal considerations. As an artist, I find this very compelling.

My videos do not pointedly deconstruct power, but seek to draw a viewer's attention to a somewhat strange place of attention (the footstep) in order to consider what kinds of communication, control and escape might be possible through this particular medium. It is perhaps easier to consider power through a strange visor rather than a more familiar one. After making the videos of walking using archival footsteps, I made a series of videos in which I walked in the footsteps of friends I had filmed walking. I didn't look at these walkers' videos, I simply walked to their recorded sounds, wondering if I might break some of my own bodily conventions prompted by the rhythms of others' bodies. By focusing my attention on the sound of their steps, I thought I might be able to find a line of flight from conventions embedded in my body, or at least expose these conventions.

Consider the video *"Dictation 4"* [10]. Veronika, the woman in the top frame, paces as she chooses, following my request to walk for 10 minutes any way she likes. In the frame below her, I try to align myself with her walk. Through much of the video my concern with finding her in space is apparent. It is hard to do, and prevents my arriving at a clean temporal alignment with her footstep. It feels something like a hide-and-seek game as I do it. My focus is more on locating her than on matching her rhythm and escaping my own body. But when I do fall into her steps, her rhythm changes mine, and it's an odd feeling. I feel like I might be able to think the thoughts paced out in her shifting composition. When I fall out of step, I'm made instantly aware of the huge distance between us, but when I find her rhythm again I sense something of what she might be living. Not her actual thoughts, but the rhythm of them. A pause registers more contemplation, then an uptick in the pace suggests something coming at a quicker pace in her mind. I move back and forth between this alignment and the cat-and-mouse game of trying to find her. In my experience of the moment, I discover there is also something oddly reassuring about being directed by another's rhythm. Someone else is telling me what to do and at what pace for a while. I give up my agency, but my existence becomes easier in some ways. I am relieved of spending the energy necessary to decide for myself what do to or how to move. Conversely, an impression of stalking simultaneously comes up in this video. In my struggle to align with someone, I seem to be chasing them, at times cutting into a private space, at times feeding off of someone else's agency. My initial empathy takes on a different tone. My effort to understand someone looks like I want to catch them or become them.

As technology gains access to previously unarticulated territory – the passages between poses, the body's rhythms, the time and space between and around events – it is worth considering what is at stake. Will our need to navigate this evermore minutely mapped world lead to increased expressivity, increased constraint, both, or something in between? Every "advance" in the ability to capture information increases the turf over which battles between enclosure and expression will be fought. Whose rhythms will guide our bodies' understandings of these increasingly data-mined spaces?

References

[1] *Webster's New Collegiate Dictionary*, Springfield, Mass, G. & C. Merriam Company (1977)
[2] Massumi, Brian, *Parables For The Virtual: Movement, Affect, Sensation,* Durham & London, Duke University Press (2002)
[3] Massumi, 75
[4] Massumi, 75-77
[5] Gilles Deleuze and Felix Guattari, *A Thousand Plateaus: Capitalism and Schizophrenia*, Minneapolis and London, University of Minnesota Press, 1987), 99 (1980)
[6] http://www.ninawaisman.net/walks0107/dictation2.mov
[7] Marcel Mauss, "Techniques of the Body" [1934], in *Incorporations*, ed. Jonathan Carey and Sanford Kwinter, New York, Zone Books, 454-77 (1992)
[8] Schwartz, Hillel. "Torque: The New Kinaesthetic of the Twentieth Century", in *Incorporations*, eds. Jonathan Crary and Sanford Kwinter, New York, Zone Books, 71-126 (1992)
[9] Jonathan Crary, *Suspensions of Perception: Attention, Spectacle and Modern Culture*, Cambridge, MA and London, The MIT Press, 3 (1999)
[10] http://www.ninawaisman.net/walks0107/dictation4.mov

staTdT_kunst - an intermedia performance

Prof. (FH) Hannes Raffaseder,
St. Poelten University of Applied Sciences
Matthias Corvinus-Straße 15,
3100 St. Poelten
hannes.raffaseder@fh-stpoelten.ac.at

Abstract. Recording, storage, editing and reproduction of audio signals have compensated some principles of sound perception and changed the human habits of listening. The project staTdT_kunst makes strong use of these possibilities. Introducing the idea of transformation and re-mixing of sound in real-time the intermedia performance reflects on these changes. Rather then reproducing the recorded sounds and visuals as close to the "original" as possible, their time structure is deconstructed, movment patterns are varied and the timbre is varied. The paper presents the basic concept of the intermedia performance staTdT_kunst, discusses the changes of some basic principles of sound perception due to possibilities of recording, storing and editing sounds and points out the idea of transformation and re-mixing in realtime.

1. Introduction

staTdT_kunst[1] is an ongoing project from Kurt Hörbst[2] (Visuals) and Hannes Raffaseder[3] (Sound) which started in Linz in 2002 and was performed in cities like Braunschweig, Vienna, Belfast, Edinburgh, Zurich and Amsterdam.
The title has two different meanings: 'statt Kunst' (instead of art) deals with the role and significance of (contemporary) art in our society, while 'stadt kunst' (city art) uses field recordings of specific sounds and voices, photos and videos sourced in the city or documentary-style interviews with people passing by as basic material for an audiovisual live-performance. Filtered through the artists subjective point of view, this footage is deconstructed, categorised, transformed, and then re-assembled into a site specific intermedia live performance.

Since their performance *Edinburgh_220306* at the Dialogues festival for new music and new media in Edinburgh in March 2006 the artists collected the audio and video material only on the course of one single day . After a short editing process of about an hour, they performed in the evening of the same day re-mixing the videos and transforming the sounds live during the concert. Using state-of-the-art techniques like live-coding, real-time editing, analyse-resynthesis, granular-synthesis, mutlichannel sound-projection and VJing Raffaseder and Hörbst create a site- and time-specific audio-visual performance.

Figure 1: Hannes Raffaseder and Kurt Hörbst

2. Artistic Concept

Today the used possibilities of recording, storing, editing and reproduction of audio and video signals are absolutly common in our society. But usually we do not care about the fact, that these possibilities have compensated some basic principles of the perception of sound and little by little changed human habits of listening. For this reason this staTdT_kunst reflects on this changes from an artists point of view. Thus, it is rather a summary of subjective observations and artistic conclusions and not a report on provable scientific research.

2.1. Some basic principles of sound perception

2.1.1. Sound as a transient medium

As pointed out in various books on musical acoustics like [1], [2] and [3] sound is definitely a transient medium. Every sonic event only lasts for a certain amount of time. Even if the duration may vary between a tenth of a second for percussive sounds to several hours for a Wagner opera, it fades out afterwords. Sound had to be perceived "now or never". It was impossible to record, store, edit and reproduce audio signals until Thomas A. Edinson (1847-1931) invented the Phonograph in the year 1877. [4] Excitation, propagation and perception had to happen simultaneously. For this reason, one had to focus his or her attention to the specific sound. Active participation and personal experience were common attributes for the listening process. Probably the direct and strong emotional impact of music, is closely related to this fact.
It was impossible to repeat a certain sound identically. Even with the same sources excited by similar means, you get at least slightly different audio signals to be perceived. If one hits on a table and tries to reproduce this sound again, than he or she will get a similar sound but definitely not the same. If a musician interprets the same composition again and again, than of course the result will be similar in some aspects, but still will differ in many others. It was impossible to listen to a musical performance again an again. Thus, sound used to be something unique and sonic events, no matter if it was a speech, a lecture, music or ambient sound, were of value to be listened to.

2.1.2. The interrelation of cause and effect

A specific sound was strongly connected to its excitation, to the sounding objects and to the space in which the soundwave propagated. Thus the listener could perceive several attributes concerning the source, the space and the excitation.
Sound had to be stimulated by a dynamic process. Only a static environment is totally silent. Without Movement and variation no sound could be heared, and hardly any movement could be

[1] http://www.stadtkunst.com (last visit: 24.08.2007)
[2] http://www.hoerbst.com (last visit: 24.08.2007)
[3] http://www.raffaseder.com (last visit: 24.08.2007)

observed without hearing a matched sound. There was an obvious interrelation between the cause and the effect as an irreversible principle. [3]

2.1.3. Static objects vs. dynamic processes

In opposite to the sonic medium, optical impressions tend not to vary over time, at least if one observes the same object in an unchanged environment. For example, one still has the possibility to admire the „Mona Lisa" painted by Leonardo da Vinci in the year 1505. While the ear catches nothing but silence in a static surrounding, the eye in this case delivers precise information full of details describing the shapes and surfaces of the environment. But Movements and dynamic processes, which causes are observed and analyzed with the ears, can only be improperly perceived with the eyes. Even a few single screens in a second evoke the perception of continous movement, while fast movement gives the impression of blured contoures.

2.1.4. Music as a time-based art

The transience of sonic energy, the impossiblity of identical repetition, the irreversible principle of cause and effect and the need for dynamic processes make sonic events to be considered phenonemons of time. There is no doubt, that music is (or at least was) a time-based artform.

According to [5] one has to point out that humans do not have a specific sense for the perception of time. Thus, time is experienced individually, rather than percieved in an objective way. To measure time individuals correlate the inner rhythms of their body (pulse, breathing stepfrequency,…) and their current personal mood (stress, boredom, fear,…) to external rhythms (the clock, the change of night and day, the cycle of the seasons) and environmental influences (frequency and content of perceived events). Especially in the western tradition, form and structure as time-based concepts, play a very important role in music. Attributes like tempo, rhythms, melodic and harmonic structure fit into a concept of dramaturgy, which at least in many cases aimes to affect the listeners experience of time.

To perceive and analyse a certain melodic theme, one first has to presume it as a joint unit. Although this part has a certain duration, it should be observed like a fixed-image. As in some aspects time is frozen while listening to the specific formal unit, it could be considered as a representation of the musical presence. [6]

2.2. The influence of recording and reproduction of sound on human listening habist

Thomas A. Edison´s invention of the Phonograph about 120 years ago marks the beginning of rapid development of technology for recording, storing, editing and reproducing sound signals. Today almost everybody is used to high-quality digital audio in multichannel surround to be transmitted and stored at rather low bitrates. Of course these possibilities have change the way of sound perception as well as our habits of listening to music, speech and ambient sound.

Once a sound was recorded it could be easily repeated. Using Loudspeakers dynamic processes are no precondition for sound production anymore. At least if one neglects the almost invisible movement of the speakers membrane.

The transience of sonic energy, the impossiblity of identical repetition, the former irreversible principle of cause and effect and the need for dynamic processes are not that important anymore or have even lost their validity. The question arises whether sonic events have still to be considered phenonemons of time.

Thinking about listening habits, CD player, walkman and I-Pod have displaced the concert as important event. Nowadays, music is available via the world wide web like water in the pipe. Many people have almost unlimited access to sound and music at any time and any place. Sound is omnipresent and easily reproduced. There is no need to listen "now or never" and specific sounds are in many cases not unique anymore.

Listeners don´t care about the beginning and the end, the formal structure, the dramaturgy of a certain piece of music anymore. That is probably the main reason why one of the main tasks of a DJ is to hide the transition from one song to another. As time tends to looses importance, it seems as sound is getting more and more a phenomenon of space. Examples for this development are not restricted to sound-installations in the field of sonic arts or ambient music as proposed by Brian Eno in the late 1970s installations. More and more often sound and music are used to design the atmosphere of private as well as public rooms like supermarkets, restaurants or pub. The specific image of club depends strongly on the music played.

While using headphones and listening to music in public, people shut themselves away from their (sonic) environment and create their own private area.

Figure 2: staTdT_kunst: Transforming and Re-Mixing in Real-Time

2.3. Soundtransformation and Re-Mixing in Real-Time

2.3.1. Overview

The possibilities of audio technology together with the changes of listening habits call for new artistic concepts dealing with the sonic medium, especially when recording, editing and reproduction techniques are used. Sound Transformation and Re-Mixing in Real-Time should be considered as an attempt in this direction. While in most cases sound recording aims to reproduce the „original" sonic environment as close to the reality as possible, sound transformation and re-mixing in real time makes use of recording, editing and reproduction techniques to alter the sonic event itself even during its recording. This means that sound is digitally recorded, edited, transformed and reproduced using multi-channel speaker system, if possible at exactly the same time or – if this fits better in the overall artistic concept of the specific project – right after the recording.

Of course the recording itself already compensates the former linear structure of time, as even simple repetition causes a cyclic structure. Current music production usually doesn´t reflect on this important change in sound perception and most listeners don´t notice or don´t care about it. Thus, for re-mixing in real sound effects and transformations causing strong impacts on the input signals are needed, so that it is more or less impossible to recognize the originating sound, one just listend to. But on the other hand some very subtle transformations have to be used as well, to enable the listener to establish links to the recorded sound sources.

This concept of Soundtransformation and Remixing in Real-Time has been developed by Martin Parker[4] and Hannes Raffaseder[5] since 2003 within their digital sound duo snail.

3. Software

The Software used by the Author for the task of sound transformation and re-mixing in real time within the intermedia performance staTdT_kunst is based on Native Instruments Reaktor[6]]. Several so called Macros and Instruments for recording, analysis, transformation, control and spatialisation purposis were designed. The different instruments can be freely accessed with the help of a matrix-mixer during a live performance. Reaktor has less programming features as other similar audio programming environments, as for instance Max/Msp[7] or PD[8]. But Reaktor takes advantage from its stability even under very high CPU load, a very good soundquality because of intern 32 Bit floating point operation and an easy interface, which enables rapid prototyping.

4. Sound-Design

As mentioned above, the sound is based on field recordings made on the day of the performance: cars, trams, foodsteps, voices, doors, atmos and so on. These sounds are transformed and remixed in real-time. The sound-design concept focuses on three basic principles: Deconstruction of the time-structure, variation of movement patterns and the transformation of timbre.

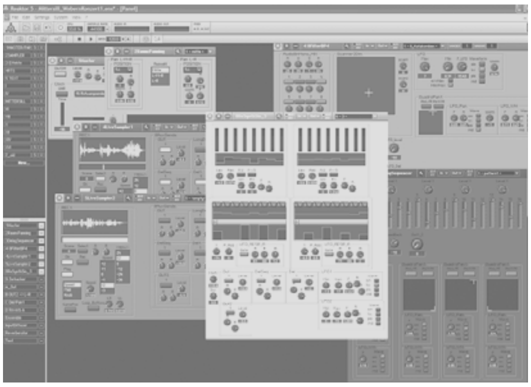


Figure 3. NI Reaktor Ensemble
for sound transformation in real time

4.1. Deconstructing the time-structure

As argued earlier the recording itself already compensates the linear time-structure. In addition any segments with any duration and from any time position within the recorded material could be defined and several segments can be played back in every single moment. Thus, the consecutive sequence of different sonic events is transformed to a more parallel structure in time. The musician is free to control the duration of the segments to given values or within given intervals. Alternatively the duration can be controlled with random processes or Low Frequency Oscillators. The Repetitions of the sequences can be triggered with a MIDI-keyboard as well as with a pattern sequencer or from random processes.

4.2. Variation of movement patterns

With the use of resampling and granular synthesis each sequence of the recording can be played back at different speeds. Both extreme acceleration and slowdown are used to transform the sound. Especially within this variation of movement patterns the interrelation of time and space becomes apparent. Thus, spatialisation using delays, reverb and a mulit-channel speaker systems is important in addition to the change of the speed. Detached from their former time structure specific patterns of movement to be triggered by the musician and controlled with LFOs and random processes can be attached to the sounds.

Figure 4. staTdT_kunst : Videotsill

4.3. Transformation of Timbre

To alter the timbre of a sound segment common filters are used as well as a Reaktor instrument designed for simple analysis-resynthesis purposes. For the analysis a bank of 16 bandpass-filters is used. The lowest cutoff frequency and distance between two filters can be controlled by the musician. The output of one specific filter controls the level of a sinusoid oscillator, which frequency initially matches the cutoff frequency of the filter. In addition the frequency of each oszillator can be detuned manually or with LFOs and random processes. Furthermore the output of the 16 filters can be stored in a table at any time. In this way, the timbre of this specific moment is frozen.

[4] http://www.tinpark.com (last visit 24.08.2007)
[5] http://www.raffaseder.com (last visit 24.08.2007)
[6] http://www.nativeinstruments.de (last visit 24.08.2007)
[7] http://www.cycling74.com (last visited: 24.08.2007)
[8] http://puredata.info (last visited: 24.08.2007)

References

[1] Hall, D. *Musikalische Akustik. Ein Handbuch.,* Schott, Mainz, 1997

[2] Roederer J. G., *Physikalische und Psychoakustische Grundlagen der Musik.* Springer, Berlin, 2000

[3] Raffaseder, H., *Audiodesign.* Hanser Fachbuchverlag, Leipzig, 2002

[4] Schubert, H. *Historie der Schallaufzeichnung.* Deutsches Rundfunkarchiv. Frankfurt am Main, 1998/2003 Quelle: http://www.dra.de/rundfunkgeschichte/schallaufzeichnung/index.html, Stand: 01.07.2006

[5] Schneider, N.J. *Zeit – Rhythmus – Zahl.* Schott Musik International, Mainz 2003

[6] Schneider, N.J. *Komponieren für Film und Fernsehen. Ein Handbuch.* Schott Musik International, Mainz 1997

The Wii Loop Machine
Musical Software Development for the Nintendo Wii Remote

Yann Seznec, Edinburgh, Scotland. yannseznec@gmail.com; +44.7867.588.685

Abstract. This paper will discuss the Wii Loop Machine, a standalone application built in March 2007 in Max/MSP that enables anyone to generate and manipulate music wirelessly using the Nintendo Wii Remote. While not the first "software hack" to use the Wiimote to play with music, I would argue that it is perhaps the most successful. I will explain the conception and creation of this project, as well as it's reception. I will examine why the Wii Loop Machine was so successful, and share some ideas concerning it's potential as a commercial product.

1. Introduction

The Nintendo Wii console was released towards the end of 2006 in North America, Japan, and Europe. By all accounts the reception took virtually everyone by surprise – Amazon.co.uk's entire pre-order stock was sold out in just seven minutes, making it the fastest selling product in the site's history.[1] Since then the console has unexpectedly outsold the Sony Playstation 3, has even outpaced the Xbox 360 to become (according to some reports) the highest selling seventh generation gaming console.[2]

Why is the Wii console such a success, breaking all sales predictions and perhaps even reinventing the console gaming market? Many have remarked on it's relatively poor graphics and sound capabilities, at least compared to the rival Xbox and PS3 devices. However this is only a consequence of consumer-oriented, rather than spec-oriented design. Much thought has been put into making the Wii console fun, classy, and endearing to as many people as possible; this is a breath of fresh air in a market where consoles are typically designed only for speed and power, in order to play blindingly fast and realistic games.

Perhaps even more crucial to the Wii's success has been the innovative wireless Wii Remote. By using motion sensing technology the controller has captured the imagination of millions of people who imagine swordfighting, driving, or playing an instrument simply by moving around. Gestural control is certainly a romantic notion that appeals to gamers and non-gamers alike, although it has ironically become one of the major critiques of the console as well; when not completely accurate it can be infuriating.

Parallel to this success a growing "hacking" community has flourished online, generally centered around using the innovative wireless Wii Remote as anything from a replacement mouse to a Vjiing tool. Thanks to the tools developed by this loosely affiliated group of independent programmers I was able to create a piece of music software called the Wii Loop Machine, which uses the Wiimote to launch and manipulate audio in real time. This software was unexpectedly successful, in part thanks to a general enthusiasm for all things Wii, but mostly because it was the first standalone application for creating music with the Wiimote. I believe that the success of the Wii Loop Machine exposes a major untapped style of commercial music software.

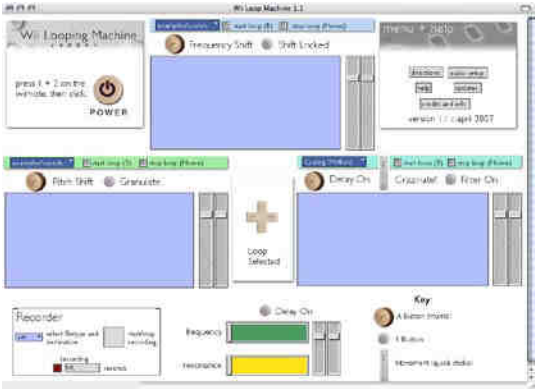


Figure 1: Wii Loop Machine Screenshot

2. Development of the Wii Loop Machine

2.1. The Wiimote and the "hacker" community

The Wiimote was very quickly picked up by the so-called "software hacking" community, with sites like Wiili.org sharing information about how to access and manipulate the Wiimote data. Part of the allure was the controller's use of the Bluetooth wireless standard to communicate with the console. This theoretically meant that it could also be used with any Bluetooth enabled computer, making it the most easily appropriated major console controller on the market. In addition, the Wiimote can be bought separately from the console for US$40, which is an extremely attractive price considering it has three accelerometers, 11 buttons, and an infrared sensor. Building a comparable controller yourself would be significantly more expensive (and probably not as solid).

Much of the hacking community has focused on using the Wiimote as a gestural interface for music and video. The use of movements to manipulate media is particularly appealing to electronic musicians and video artists sick of performing with a computer mouse and keyboard.

Three of the major successes to date, in terms of independent Wiimote development, have been GlovePIE, Darwiin Remote, and aka.wiiremote.

GlovePIE[3], by Carl Kenner, is a Windows-only text-based coding application (similar to Java and BASIC) that creates an emulation of a joystick or a keyboard from a video game controller input. This has been used very successfully with the Wiimote to control a number of different types of software. The

[1] *Wii Crushes Amazon Pre-Order Records,* computerandvideogames.com

[2] *Nintendo Wii is market leader in home console business*, vgchartz.com

[3] http://carl.kenner.googlepages.com/glovepie

interface is not particularly user friendly, however, and someone without at least some coding knowledge would be unable to do anything complex.

Darwiin Remote[4], by Hiroaki, is a nicely designed software interface for the Wiimote on Mac OS X. It accesses the Wiimote automatically and allows you to use it as a mouse replacement, or to control other software. While very simple to use and well designed, Darwiin is not a very powerful tool.

Aka.wiiremote,[5] however, is an extremely powerful object for Max/MSP that facilitates access to the Wii Remote data. Created by Masayuki Akamatsu, it has been continuously updated since it's release, and now boasts full data capture from the Wiimote as well as many of the peripheral controllers (nunchuck, classic, etc). It is well designed and easy to use, with a very nice help file. The only downsides are that you must be quite familiar with Max/MSP to take advantage of its features, and it is only available for Mac OS X. Similar Max objects have appeared for Windows, although nothing quite as powerful.

Although many pieces of computer software have been released in the past few months that use the Wiimote to control music, video, games, and much more, GlovePIE, Darwiin, and aka.wiiremote have been perhaps more important, in that they are tools that allow for the relatively easy development of software to use the Wiimote. Nearly all independent computer-based Wiimote software is indebted to at least one of these three tools.

2.2. The concept

In March of 2007 I decided to use the aka.wiiremote object in Max/MSP to create a piece of standalone Wiimote-enabled music software as part of my Sound Design Masters course at the University of Edinburgh. I wanted it to be loop-based, much like Ableton Live, and at every point I wanted to make it simple and straight forward, easy enough for anyone to use. I had found then, and I still find to a certain extent, that nearly every software tool for using the Wiimote was unnecessarily difficult to use, and accessible only to coders and programmers. In addition, I generally find that most „pro-sumer“ music creation software (Reason, Live, etc) has a relatively steep learning curve, shutting out a vast quantity of people who want to make music but do not have the time or prior knowledge needed in order to do so. These two concerns, I felt, could be addressed and combined with the general enthusiasm for the Wiimote by gamers and non-gamers alike.

2.3. Building the Machine

Although linking the controller to the computer can be a bit confusing at first, as a general rule the aka.wiiremote object is quite simple to use and allows very easy access to the data from the Wiimote. However I soon learned, as have most independent Wiimote developers, that this data is quite difficult to use and manipulate.

While using movement to control various musical parameters from filter frequency to grain location is an exciting concept, accelerometers are not necessarily a very efficient way to achieve those goals. The three accelerometers in the Wiimote can very accurately measure rotation, which means that slow turning motions can be used effectively, but this is a rather boring method of manipulating music, and is not what people expect or want to do with a Wii controller. Faster or more drastic movements are much more difficult to track, as the accelerometers only give useful data whilst the Wiimote is moving. It is very difficult to accurately describe mathematically where someone has moved the controller, or how far. This in turn makes designing musical applications for the accelerometer data quite problematic (particularly for a musician with no mathematical training).

Unfortunately, the first thing most people do when picking up a Wiimote is swing it about like a sword – a far cry from the delicate movements that are easy to track and map to musical parameters. Therefore I decided to create some effects that would take advantage of these extreme movements, if not entirely accurately.

The infrared sensor in the Wiimote is, of course, far more precise than the accelerometers, and is used by pointing towards the screen. This sort of tool has huge potential in terms of musical interface, but requires a "sensor bar" to triangulate the infrared signal and turn it into useful data. The Wii console comes with a sensor bar, and they are not terribly difficult to build (regular LED's work fine, as do two candles). However when starting the Wii Loop Machine I decided not to force people to locate or build another peripheral in order to use my software. I wanted to have as much control over the final product as possible, and including an infrared system would have been just another variable to control.

2.4. Making the musical side of things

My goal for the Wii Loop Machine (WLM) was to essentially make a very simple version of Ableton Live, in the sense of a tool for controlling and manipulating loops of audio. However I wanted to make my software as accessible as possible to non-musicians and non-programmers. I thus settled on dividing the interface into four "modules", corresponding to the four directional arrows on the Wiimote. Three of these are looping audio players, and the fourth is a very simple subtractive synthesizer.

The audio modules each have different effects that are activated by pressing one of the buttons, and manipulated by moving the controller. Any number of audio files can be loaded into each module (by dragging a folder over the module), and the loops are selected and launched with the controller.

I was not sure how much quantization to build into the WLM. On the one hand a system that always plays the loops in time will always sound "right", but it also takes away a good deal of control and feedback. Having no quantization, on the other hand, forces the user to listen carefully and work on their timing. This can be frustrating, but more importantly it can be rewarding. A truly musical interface must be designed with the "dexterity/musical result curve"[6] in mind; it must be set up to give the user a sense of accomplishment in order to encourage them to do more. A lack of quantization, I decided, could create that challenge.

I did, however, create a very simple time stretching system to make sure that all loops were played back at the same length. The top audio module is the "master" loop, and the other two loops are automatically re-pitched and re-sized correspondingly. Thus when the user launches the loops they will all be the same length.

Each module has a different set of effects. These are a granulation/pitch shifting system, a frequency shift/hold system, a filter, and a "crazinator" (a randomly variable delay effect). Each of these can be manipulated by moving the controller.

The Wii Loop Machine is far from perfect. Only one module can be selected or launched at a time, for example, and ideally

[4] http://www.wiili.org/index.php/Darwiin

[5] http://www.iamas.ac.jp/~aka/max/

[6] Behrman ‚Designing Interactive Computer-Based Music installations' (p. 140)

the master quantization and tempo settings should be changeable. Future builds will begin to address these and many other issues.

Figure 2: Frame from the Wii Loop Machine Demo Video

2.5. Release and reception

The Wii Loop Machine was originally made as a project for my Sound Design Masters course, and as such had a very fixed deadline. I finished the software the night before the deadline, and decided to make a demo video (Figure 2) showing it in action. I shot and edited the video at 5 am and submitted the software, video, and a small writeup a few hours later. I was not sure whether what I had done would be of any interest to anyone at all.

The next day I posted the software on my blog (theamazingrolo.blogspot.com) and on the electronic music forum em411. By the end of the day I had received some very positive feedback, and within the next few days my project was featured on Engadget, Make, Amazon, and more (by way of Create Digital Music). I received phone calls from the Wall Street Journal and the Times, and hundreds of emails from all over the world. To call it a surprise would be a gross understatement – I had not imagined more than a few hundred people being interested in my project, but within a week I was getting several thousands hits and downloads a day. A very conservative estimate would put the number of Wii Loop Machine software downloads in the hundreds of thousands at least. That a Mac-only prototype software would garner so much attention is remarkable, and shows the potential for this type of application.

I was not the first person to use the Wiimote to make music on a computer, however I was the first to create a standalone software that was easily downloadable and usable by anyone with no programming or musical skills. While that can explain a certain amount of it's popularity, the continuing attention paid to the software shows that there are other underlying reasons for the success.

It would be naïve to overlook the importance of the demo video I made in the early morning hours of my submission. The video was featured on virtually every blog entry about the software, and much of the praise (and dismissals) was actually directed at my performance rather than the software itself. More than anything, though, the video contributed to the success of the software by showing the Wiimote in action making music.

Of course, the main attraction of the WLM is the appropriation of existing hardware. The vast majority of people who have used the software, I imagine, are owners of Wii consoles who are familiar with the Wiimote and are either disappointed with the games available on the console or are simply curious to see what else can be done.

The simplicity of the software appeals to most people as well. While many people have heard that it is possible to pair the Wiimote with a computer, it is certainly difficult for anyone to do without some programming skill and a fair amount of free time. My software, thanks to the aka.wiiremote object (and reasonably compliant Mac Bluetooth drivers), is quite simple for anyone to use. This applies as well to the musical side of things. I included some default audio loops, so the user can begin making music right away. The interface is fairly rigid and unchangeable, which may not appeal to hardcore musicians, but it is powerful enough for a user to feel that the music they are generating is their own.

Music production software is a big business, as the growing number of pro-sumer quality digital audio workstations can attest. Faster computers and better quality software has made the production of music accessible to more people than ever before, but there is still a large untapped market of people who want to make their own music but are unable to do so due to the learning curve and time outlay required for existing software. What the WLM proposes and delivers is a system for creating highly personalized and creative music by using familiar pre-existing tools – essentially a game where the only goal is original music. The success of the WLM shows just how universal the desire to make original music is, and it provides the tools to do so.

2.6. Criticism

Like any music software, the WLM was somewhat polarizing, and was the target of heavy critiques, some quite damning and others constructive and helpful.

The biggest complaint was that the software is currently unavailable for Windows. While not a complaint as such (actually it is quite flattering that people want a Windows version so much), it is definitely a major concern. The only reason I have not yet done so is because the aka.wiiremote object has not been successfully compiled for Windows. While this sounds simple, the true culprit is easily pairing the Wiimote with Windows Bluetooth drivers. For the moment the easiest way to do so is by using BlueSoleil, a third party Bluetooth stack, and from there either use GlovePIE or a Windows Wii Max/MSP object like tk.wii, but these methods are not yet satisfactory. I hope to find a solution and compile a Windows version within the next few months.

A more damning critique had to do with the sound and the manipulations of the audio in my demo video. The effects in the WLM, particularly the granulation and "crazinator" functions, were designed with a very specific glitchy electronic sound in mind. This sonic aesthetic, I felt, was a good match for the movements I imagined people would make with the software. While some commenters (on sites like Joystiq or Engadget) appreciated that sound, saying "That sounds like an Aphex Twin song", others said "the more he manipulated the beat the worse it sounded".[7] I am certain, however, that the people who left the most negative feedback did not bother to download the software and try it out for themselves, as they would have learned that my performance was only a demo showing one possible way of using the WLM.

A more confusing critique was that the software was "pointless". While that in of itself can be explained as someone who is not interested in making music, it was often followed a statement along the lines of "this could be done much better with ____". The blank was generally the name of some commercial hardware or software, from the Kaoss Pad to Fruity Loops.[8] I often found that these commenters were simply trying to show off their own knowledge of consumer level music production, and were ignorant of the possibilities that a system like mine offered. Certainly, traditional musical production tools will always offer better quality sound and far more precise

[7] *Drop mad beats with the amazing Wii Loop Machine*, Joystiq.com

[8] *Wii Loop Machine*, The Amazing Rolo

manipulations of the music. However moving and dancing around with the Wii controller will almost certainly be more fun than twisting a knob or clicking a mouse, and it is definitely more fun to watch.

I have received many emails from people asking me to make either a Wiimote Theremin or a Wiimote Drum Kit. A version of latter has since been released as the Wiinstrument, and numerous attempts at emulating a theremin have been made,[9] but I have very little interest in those types of projects. Rather than use the Wiimote to create poor imitations of existing tools, I would much rather try and make a system that takes full advantage of the innovative controller to make something that could not have existed before.

Despite these critiques, however, the response to the WLM was overwhelmingly positive, with a vast majority of encouraging comments. I was pleasantly surprised by the number of people who used the Wii Loop Machine, even posting videos on YouTube of their performances. Perhaps even more rewarding personally were people like Soressa Gardner, a musician from Vancouver, Canada, who modified the WLM for use in her degree in applied music.

2.7. Continued development

As mentioned earlier, the Wii Loop Machine is still essentially a prototype, a proof-of-concept that the Wiimote can be used to make music, and that this is a type of software that could develop a very strong and devoted fanbase. As an initial prototype the WLM was been a resounding success, and now much thought must be put into what lies ahead.

The most pressing step, as mentioned earlier, is to develop a Windows version. I hope to do so in the coming weeks, along with a Version 2.0, which will feature more stable architecture and much more flexible quantization and effects systems, all still built in Max/MSP. In addition, I hope to implement some MIDI and/or ReWire support, to enable users to easily integrate the Wii Loop Machine into their existing setup. With those new features, as well as a general redesign, the software should reach it's full potential as a prototype.

From there, my thoughts turn towards making a commercial product. This would involve rebuilding the software from the ground up in C++, using the initial Max/MSP prototypes as guides. Ideally it would be cross-platform and support multiple players and network jamming, and have several different "modes" (loops, synths, customizable effects, etc). The system would have to be expandable, flexible, simple, customizable, and of course sound good and be fun to play. Eventually I would aim for a release on the Wii console.

However it would always be important to differentiate the Wii Loop Machine from karaoke-style video games like *Guitar Hero*. The WLM should always be aimed towards people who want to generate and perform their own music, rather than play along with pre-existing tracks. The user must always feel a certain ownership and relationship with the music they are creating. In this sense the WLM could be most closely associated with Toshio Iwai's *Electroplankton* for Nintendo DS, which sets up innovative and intuitive graphical systems for making music. Much in the same vein, I do not want to make a game, but rather a tool that happens to be fun to play.

Of course, game consoles do not have very long lifespans. However the gestural control interface pioneered by the Wii will almost certainly continue in future console systems by Nintendo and others. I would therefore like to develop the Wii Loop Machine as a general framework for generating music which could be applied to any gaming system with a movement-based controller.

3. Conclusion

The Wii Loop Machine has proved to be a popular system with great potential. It has shown that there is a major untapped market for music generation software, particularly using existing video game hardware. Until now electronic music production has been the domain of computer musicians or programmers. Very little fills the gap between karaoke-style video games that let you play along with pre-existing tracks and powerful music production tools with steep learning curves. The Wii Loop Machine has the ability to fill that gap and create a game that takes advantage of the exciting possibilities inherent in the Wiimote, allowing anyone to intuitively create and manipulate music with their movements.

References

[1] Carl Kenner, *GlovePIE*, http://carl.kenner.googlepages.com/glovepie

[2] Christopher Grant, *Drop mad beats with the amazing Wii Loop Machine* (blog comments), http://www.joystiq.com/2007/03/22/drop-mad-beats-with-the-amazing-wii-loop-machine/ (March 22, 2007)

[3] *DarwiinRemote* http://www.wiili.org/index.php/Darwiin

[4] David Behrman, 'Designing interactive computer-based music installation', Contemporary Music Review, 6:1 (1991) 139-142

[5] Masayuki Akamatsu, *aka.objects* http://www.iamas.ac.jp/~aka/max/

[6] *Nintendo Wii is market leader in home console business*, http://www.vgchartz.com/news/news.php?id=508 (August 22, 2007)

[7] *Nintendo Wiimote Theremin with Moog Little Phatty,* http://uk.youtube.com/watch?v=8N-LpbXF33g

[8] *Wii Crushes Amazon Pre-Order Records,* http://www.computerandvideogames.com/article.php?id=149694 (November 22, 2006)

[9] Yann Seznec, *Wii Loop Machine* (blog comments), http://theamazingrolo.blogspot.com/2007/03/wii-loop-machine.html (March 19, 2007)

[9] *Nintendo Wiimote Theremin with Moog Little Phatty*, youtube.com

A Flexible Music Composition Engine

Maia Hoeberechts, Ryan J. Demopoulos and Michael Katchabaw

Department of Computer Science
Middlesex College
University of Western Ontario
London, Canada N6A 5B7
hoebere, rdemopo2, katchab@csd.uwo.ca

Abstract. There is increasing interest and demand for adaptive music composition systems, which can change the character of generated music on the fly, for use in diverse areas such as video game music generation, film score composition, and development of interactive composition tools. This paper describes AMEE^{TM} (Algorithmic Music Evolution Engine), a prototype system for dynamic music generation. The features which distinguish AMEE^{TM} from other composition systems are the use of a *pipelined architecture* in the generation process to allow a structured, yet flexible approach to composition, the inclusion of *pattern libraries* for storing and accessing musical information, and the addition of an *emotion mapper* which allows music to be altered according to emotional characteristics.

1. Introduction

Traditionally, the music one finds in video games consists of a static set of compositions packaged with the game. Creating game music is often an expensive proposition, requiring either the licensing of existing pieces from artists, or hiring professionals for custom compositions. Furthermore, having a static collection of music can become repetitive, and fixed selections cannot be altered as part of the user's interactive experience during game play. Considering these limitations, we are working towards an alternative: music generated on demand during game play, which can be influenced by game events and change its character on the fly.

This paper describes AMEE^{TM} (Algorithmic Music Evolution Engine), a prototype system for dynamic music generation. AMEE^{TM} was designed with several principal goals in mind:

(a) Permit maximal flexibility in the composition process. The engine can either generate music without any restrictions, or a human composer can guide musical choices.
(b) Provide an extensible architecture that can be easily integrated with other software.
(c) Incorporate a multi-level application programming interface which makes AMEE^{TM} functionality accessible to users with varying levels of musical and/or programming knowledge.
(d) Reuse elements, such as note sequences and harmonic structure, both from existing human composed pieces or computer generated material.
(e) Allow music to be altered based on emotional characteristics such as happiness, sadness, anxiety, liveliness etc.

AMEE^{TM} is an object-oriented system written in J# and Java. It includes high-level classes such as Musician and Instrument that model real-world entities involved in music composition. The features which distinguish AMEE^{TM} from other composition systems are the use of a *pipelined architecture* in the generation process to allow a structured, yet flexible approach to composition, the inclusion of *pattern libraries* for storing and accessing musical information, and the addition of an *emotion mapper* which allows music to be altered according to emotional characteristics. A more detailed description of these features can be found in Section 3.

There are two main applications areas we envision for AMEE^{TM} at the present time. First, AMEE^{TM} could be embedded within application software to facilitate online, dynamic composition of music for immediate use within that application. For example, using the engine in a video game would allow endless variety in the game music, and since composition is done dynamically, the generated music could be tuned to reflect emotional context during game play. Alterations to the music could be initiated from within the game, by the game player, or both. Second, we foresee using AMEE^{TM} as a basis for stand-alone composition tools. For example, consider a system which would permit collaboration among human composers who could exchange parts of pieces created with the engine, or share virtual musicians, instruments, and musical elements. The AMEE^{TM} architecture can also support the creation of virtual bands and jam sessions.

AMEE^{TM} is currently in a prototype stage. The software is fully functional, but there are many planned extensions and improvements that we are working on. This paper describes the features, design, implementation, and future development of the AMEE^{TM} prototype.

1.1. Related Work

AMEE^{TM} is an example of an automatic music composition system. These systems can be broadly classified into five categories: stochastic (random) methods, genetic/evolutionary approaches, recombination approaches, grammar/automata based methods, and interactive variants. An overview of some of these systems can be found in [3] and an historical account of some automatic music composition systems can be found in [1]. Concerning the above classification, AMEE^{TM} is a flexible system which uses a combination of stochastic methods, recombination and interaction.

Recently, the terms *interactive music* or *adaptive music* have been adopted to describe music which changes its character according to context in a film, video game or real-life scenario. A comprehensive survey of these systems is beyond the scope of this paper, but the following are few examples of interactive music systems. MAgentA (**M**usical **Agent A**rchitecture) supports the generation of mood-appropriate background music by dynamically choosing composition algorithms which were previously associated with particular emotional states [2]. In [6], the authors describe a system for altering music to produce emotional variation based on *structural rules* (for example, affecting tempo) and *performance rules* (for example affecting accenting). Informal empirical testing showed some sucess of the system in perceived emotional content of the generated music. Scorebot [7] is a low-level API to manipulate music to be used in a film score based on scene information such as emotional content, timing and events. It enables the storage of musical themes which can be sent to manipulation modules that change the characteristics of the theme. Dynamic Object Music Engine (DOME) appears to be a system with similar goals to AMEE^{TM}, although not much information is publically available on its implementation details [9].

2. Overview of Design and Implementation

The prototype was developed in J# and Java. The source code of the version described in this paper was assembled and compiled under Microsoft Visual Studio .NET. We believe that the software can easily be ported to standard Java and with some modifications to J2ME, although we have not attempted these conversions at this time.

The software is divided into several main groups of classes:

- **The pipeline.** These classes control the flow of the music generation process, and are responsible for calling methods on the generating classes.
- **The producers.** These classes produce high level musical elements from which the composed piece is comprised (sections, blocks and musical lines).
- **The generators.** These classes create the lower level musical elements (harmonic patterns, motif patterns, modes, meters) from which the products are assembled. Generators can have library-based or pseudo-random implementations.
- **High-level classes.** Musician, Instrument, Performer, Piece Characteristics, Style, Mode, Meter, and Mood. These classes implement the real-world entities modelled in the engine.

For the prototype of AMEE^{TM} described in this paper, we have provided rudimentary implementations of all the necessary classes in order for the engine to produce music. The classes are named using the convention “StandardClassName,” for example the StandardPipeline and StandardMotifGenerator. The purpose of writing these classes was as a proof-of-concept for the method of music generation used by AMEE^{TM}.

2.1. Process of Music Generation

In order to use the engine, producers and generators must be created and loaded in the pipeline.

The first step is to use a GeneratorFactory to create the generators. Five types of generator are necessary for the engine: a StructureGenerator which creates the overall sectional structure of the piece (e.g. ABA form), a HarmonicGenerator which creates a sequence of chords for each section (e.g. I-IV-V-I), a MotifGenerator which creates short sequences of notes (e.g. a four note ascending scale of sixteenth notes beginning on the tonic), a ModeGenerator which returns modes for the piece (e.g. start in F+, progress to C+, divert to D- and return to F+), and a MeterGenerator (e.g. 4/4 time). Every generator contains at least one pseudo-random number component which is used for decisions it needs to make.

The generator might also contain a PatternLibrary which provides the generator with musical elements it can use directly, or as starting points for musical element generation. PatternLibraries are created in advance, and are intended to embody musical knowledge. For example, for the purposes of the prototype, we have created a “Bach” MotifPatternLibrary containing motifs from Bach’s Invention No. 8 and a HarmonicPatternLibrary based on the same piece. Eventually, more extensive PatternLibraries should be created by musicians for distribution with the engine, and end users of the music generation package will also have the ability to add to the libraries.

Once the generators have been created, then the producers must be created using a ProducerFactory. There are four producers necessary. The SectionProducer uses the StructureGenerator to produce Sections, where a Section is a chunk of the piece with an associated length in seconds. Each Section contains a number of “blocks” which are short segments of the piece (for example, 4 bars) composed of a musical line played by each musician. The BlockProducer is responsible for deciding how to coordinate the musical lines using a HarmonicPattern created by a HarmonicGenerator associated with the BlockProducer. The musical lines themselves are composed by the LineProducer which uses a MotifGenerator to create the actual note sequences in the musical line played by each musician. Finally, an OutputProducer must be initialized which will convert the generated piece to MIDI and output it as audio.

When the producers have been initialized, they are loaded into a Pipeline. The Pipeline oversees the generation process and calls on each producer in turn to assemble the piece. An example of the process is as follows: the SectionProducer is called on to get a new section of the piece, then the BlockProducer returns the first block of that section, which in turn is passed to the LineProducer and filled in with notes, and lastly the completed block is sent to the OutputProducer for sound output. At each step in the pipeline, if desired the products can be sent to the EmotionMapper for Mood dependent adjustments. Music generation continues until the duration for the piece, specified by the user or determined by AMEE^{TM}, is completed.

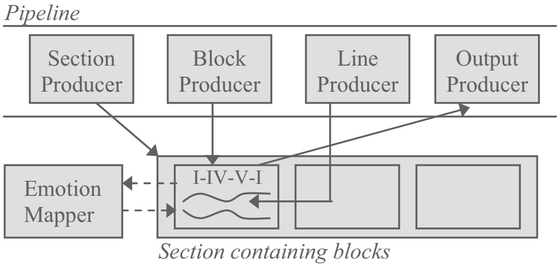


Figure 1: Illustration of pipelined generation

3. High-level Features of AMEE^{TM}

This section will describe the unique features of AMEE^{TM}, and highlight some of the design choices which were made.

3.1. Realistic Modelling

A key consideration in the design phase was that the entities involved in music creation reflect their real-life counterparts. At the highest level, the engine deals with the following classes: Musician, Instrument, Performer, PieceCharacteristics, Style, Mode, Meter, and Mood.

A Musician plays an Instrument and has a Mood. Also, a Musician has an Ability, and knows a number of Styles. Our intention was to model a real musician, who can play one instrument at a time, but has the ability to play that instrument in different styles with varying ability. Consider a pianist who is classically trained, but also plays some improvisational jazz. We would model this pianist as a Musician who knows (at least) three styles: classical, jazz and her own style – a personal repertoire of improvisational riffs.

The purpose of storing an ability for the musician is that eventually we would like to be able to model "good" and "bad" musicians. Some aspects we have considered are: ability to play an instrument in tune; ability to follow a beat; ability to contain one's mood and play in the manner desired by the conductor. The styles known by the musician will also reflect ability. On the other hand, we might want to model a musician with limited ability – perhaps in a video game role (imagine a bar scene where a bad band is playing), or perhaps eventually we might create an application where one has to "train" musicians, or one might want "bad" musicians simply for the entertainment value of hearing them play. Although our prototype does not currently make use of a Musician's Ability, all the hooks are in place to make this possible.

Our motivation for modelling musicians in this way is that we envisioned applications where users could eventually develop and trade musicians with one another. When considering an end-user of the software, we pictured users having collections of musicians each with their own skill set. The Musicians could be shared, traded, combined into groups, and perhaps even marketed.

3.2. Pattern Libraries

The purpose of the PatternLibraries is to allow new music to be generated by reusing compositional elements: either elements from existing pieces or elements which have been previously generated. Existing pieces used to extract musical elements will be in the public domain to ensure that no copyright issues are encountered. A PatternLibrary can be thought of as a repository of musical ideas. The four libraries which are currently used in AMEE^{TM} are the HarmonicPatternLibrary, the MotifPatternLibrary, the ModePatternLibrary and the MeterPatternLibrary. Our motivation in creating the PatternLibraries was twofold: to give the system the ability to compose in different styles, and to provide a mechanism for exchanging and storing musical ideas. Each of these goals will be discussed in turn.

To answer the question, "What style is this piece of music?" you would listen for clues among the musical elements of the piece to determine its classification. The instruments being played are often an initial hint (the "sound" of the piece). Further to that, you would attend to the rhythmic structure, the harmony, the melody, the mode and the meter, and transitions between elements. Consider the knowledge of a jazz trumpet player, for instance Louis Armstrong. He knows typical harmonic progressions that will be used in a piece, and has many "riffs" in mind that he can use and improvise on. In our system, this knowledge would be captured in a HarmonicLibrary and MotifLibrary respectively.

How the libraries are used is determined by the implementation of the generators. A generator implementation could use the library as its only compositional resource, that is, all the musical elements returned by the generator are those taken from the library, or it could use the library patterns as starting points which are then modified, or it could return a mixture of library patterns and generated patterns (and naturally, it could ignore the library contents entirely and return only generated patterns). Thus the libraries are rich musical resources which can be flexibly used depending on the desired musical outcome.

Regarding the exchange of musical ideas, consider a situation where you meet someone using AMEE^{TM} who has a very dynamic guitar player (let us call him Jesse Cook). The knowledge of the guitar player is contained in the libraries the guitar player is using for music generation. You could allow your Louis Armstrong player to learn how to make his trumpet sound like a flamenco guitar by incorporating the riffs from Jesse Cook's MotifLibrary into Louis Armstrong's existing library. A different possibility is that you could add the Jesse Cook player to your band. What would it sound like if the two musicians jammed together? Or, you could ask Jesse Cook and Louis Armstrong to collaborate in an Ensemble. Could they influence each other while playing? All of these functions are directly supported by the idea of PatternLibraries as embodiments of musical knowledge.

3.3. Pipelined Architecture

The pipelined architecture described in Section 2.1 has several significant advantages over other potential architectures. The generation process can be pictured as an assembly line for constructing musical blocks. Each of the producers along the pipeline fills in the elements its generators create, until the finished block is eventually passed to the OutputProducer for playback. The producers all work independently, which means that there is potential to parallelize the generation process. Furthermore, to dynamically alter a composition, a different generator can be substituted in a producer without affecting the rest of the pipeline.

3.4. Emotion Mapper: Mood Based Variations to Music

A key feature of AMEE^{TM} is the incorporation of Mood as a factor which can affect music generation. Mood is considered in two contexts: individual Musicians have a Mood which can be adjusted independently of other Musicians, and a piece can have a Mood as well. Imagine an orchestra with 27 members. Suppose that the bassoon player is depressed because she just learned that she can no longer afford her car payments on her musician's salary. Suppose that the orchestra conductor is trying to achieve a "happy" sound at the end of the piece currently being played. Depending on how professional the bassoon player is, she will play in a way which reflects her own "sad" mood, and the desired "happy" mood to varying degrees. These are the two contexts in which we have considered musical mood.

The StandardEmotionMapper is a simple implementation of a class which makes adjustments to musical elements based only on the emotions Happy and Sad. The StandardEmotionMapper has methods which adjust the Mode, MotifPattern (pitch) and tempo. The logic behind the emotion-based changes is based on research in music psychology. A summary of some papers and ideas which was done for this project can be found in [5]. The adjective descriptors used in the mood class are those defined by Kate Hevner as her well-known Adjective Circle [4].

Presently, all mood adjustments are made based on the Mood characteristics of the first musician added to the group. All other musicians have Mood as well, but it is ignored at this time. The changes necessary to use Moods from all musicians are minimal, however there are a few questions which need to be resolved regarding interactions between global (piece based) mood and local (musician based) mood. For example, Mood can affect tempo of the piece. Should tempo be constant for all musicians? What would happen if musicians could play at different tempos depending on their moods? Our current implementation shows that interesting results can be achieved by altering mood, but we have envisioned many other possibilities which have yet to be fully explored.

4. Implementation Features

This section provides some details about the implementation of AMEE^{TM}. Extensive planning was done in the design phase such that the engine is easily extensible, and in order to allow additional features to be added. Hence, the current capabilities of the prototype only reflect a proof-of-concept of the engine's functionality, and by no means define its limitations.

4.1. The MotifGenerator

The MotifGenerator, although it is only one small component in AMEE^{TM}, contains the code which one would normally think of as the main element of a music generation system: it generates sequences of notes and rests with associated timing. One very important design decision differentiates AMEE^{TM} from most other music generation systems: the notes (Motifs) that are generated are *independent of both the mode and the harmony*. This is best illustrated by example.

Consider the opening phrase of Mozart's Sonata in A+, K.331:

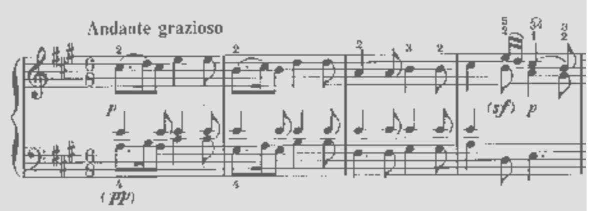


Figure 2: Mozart Sonata in A+, K.331

The right hand melody in the first bar begins on C#, the third of the scale, and is played over the tonic chord in root position in the bass (harmony I). In the second bar, in the left hand part, we see the exact same melodic pattern, this time starting on G#, played over the dominant chord in first inversion (harmony V_6).

In a motif, we would encode this musical idea as

Pitches:	2	3	2	4	4
Locations:	0.0	1.5	2.0	3.0	5.0
Durations:	1.5	0.5	1.0	2.0	1.0

where pitches are positions in the mode relative to the root of the harmonic chord (with the root as 0), the locations indicate an offset from the beginning of the bar (location 0.0), and the duration specifies the length of the note. Locations and durations are expressed in terms of number of beats (in 6/8 time, an eighth note gets one beat).

The purpose of encoding motifs in this way is to capture the musical pattern associated with the sequence of notes, without restricting ourselves to a specific mode or harmonic chord. The approach we have chosen for motif storage allows composed pieces and musical lines to be transposed and reinterpreted in any mode. Moreover, as illustrated in the above example, a particular pattern of notes often appears more than once during a piece, but serving a different function depending on the underlying harmony.

The StandardMotifGenerator in the prototype operates in both the library-based and pseudo-random manner. When a motif is requested, it first checks whether a library has been loaded. If it has, it attempts to retrieve a motif of the specified length (number of beats), and the desired type (end pattern, which is one that could be found at the end of a section, or regular pattern, or either). If any suitable patterns are found, they are returned. Otherwise, a new pattern will be generated.

Two pseudo-random generators are used in motif creation: a number generator and a pitch generator. A loop continues generating notes/rests as long as the desired number of bars has not been filled. The motifs are generated according to musically plausible rules and probabilities.

As previously mentioned, the motif is encoded in a mode-independent and harmony-independent manner. Thus, the "pitches" that are generated and stored are actually relative pitches to the tonic note in the mode. Consider the following concrete example: Suppose the motif contains the pitches values 0 – 1 – 0. If that motif is eventually resolved in a position where it appears as part of the I chord in C+, would be resolved to the actual MIDI pitches for C – D – C. However, if that same motif were used in a position where the harmony required the V chord in C+, the motif would now be resolved to G – A – G. Suppose now that the motif contains the pitch values 0 – 0.5 – 1 – 0. The value "0.5" indicates that a note between the first and second tone should be sounded, if it exists (this will produce a dissonance). Thus, in C+ for chord I, 0 – 0.5 – 1 – 0 would be resolved as C – C# – D – C. If an attempt is made to resolve a dissonant note where one does not exist (for example, between E and F in C+), one of the neighbouring notes is selected instead.

4.2. Collaboration Between Musicians

The purpose of generating a harmonic structure for a piece is to permit groups of musicians to perform a piece together which will sound musically coordinated. The LineProducer is the entity in the pipeline which is responsible for resolving motifs into notes that fit into chords within a mode. When multiple Musicians are playing together in an Ensemble, the harmonic structure of the block being generated is determined (a HarmonicPattern is chosen), and then each Musician's line is generated *based on this same HarmonicPattern*. The result is that even though each musician is playing a different line from the others (and possibly different instruments, each with its own range), at any given time each musician will be playing a motif which fits into the current harmonic chord and mode for the piece. Of course, this does not imply that every note each musician plays will be consonant with all other notes sounding at that time – that would be musically uninteresting. Musicians might be playing passing tones, ornaments, dissonant notes, and so on, but the harmonic analysis of each of their musical lines will be the same.

4.3. Choice of Mode

Music can be generated in AMEE^{TM} in *any* mode which can be supported by the underlying MIDI technology. This is a very flexible implementation which allows music played to be played in any major or minor key, or using a whole-tone scale, chromatic scale, blues scale, Japanese scale etc. The restrictions imposed by MIDI on the scale are that the period of repetition for the scale is an octave, and that the smallest distance between two notes is a semi-tone. Thus, an octave is divided into 12 semi-tones, designated by consecutive whole numbers in the MIDI format (i.e. 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11). Note that these are "normal" restrictions for Western music. The pattern of the scale is stored using offsets from the starting note of the scale. These offsets specify the number of semi-tones between steps in the scale. For example, every major scale has the following step pattern: 0, 2, 4, 5, 7, 9, 11. To specify which major scale is desired, we also store the starting note of the scale, which is a whole number between 0 and 11 which corresponds to an offset from MIDI pitch 0, which represents C.

Motifs can be resolved into any mode, regardless of the mode in which they were originally specified. An unavoidable consequence of this flexible implementation is that the motifs might sound strange if the mode into which they are resolved contains less tones than the mode in which they were designed. The resolution method which is used in this case is that the notes are "wrapped around" using a modulus-style computation (i.e., the sixth note in a five note mode becomes the first note one octave higher).

4.4. Flexibility of Implementation

A key consideration in the design and implementation of AMEE^{TM} was to provide maximal flexibility for additions, improvements and extensions. We achieved this by implementing all the key classes as Abstract classes which can be subclassed according to the goals of the programmer. As an example, consider the Abstract class MotifGenerator. We have provided a simple concrete implementation of the MotifGenerator in the class StandardMotifGenerator. Now suppose that a developer wishes to have a MotifGenerator which is designed to create jazz style motifs. This would be accomplished by writing a class, JazzMotifGenerator, which extends MotifGenerator, and provides implementations of all the abstract methods. Once this was done, the rest of AMEE^{TM} can be used without any changes.

All the other principal classes in AMEE^{TM} follow this same pattern, and thus the whole system is easily extensible and enhanced.

A further area in which we have attempted to maximize flexibility is in the data structures for the musical elements. As previously mentioned, the mode can be anything which is MIDI supported. The piece length can be user defined or determined by AMEE^{TM}. Any MIDI available instruments can be used. Motifs can be as long or as short as the user desires. Any number of PatternLibraries can be developed and used. Any harmonic chords can be defined, which include any number of notes. Musical styles are completely user defined. And so on...

5. Future Development and Open Questions

5.1. Checkpointing

All the pseudo-random number generators are completely deterministic. This means that if generators are initialized with the same seed values during two different runs, the music produced will be exactly the same. We plan to exploit this characteristic in order to be able to checkpoint the pipeline during music generation so that musical elements can saved and repeated.

5.2. Dynamic Alterations to Music

Eventually, we would like to be able to alter the music which is playing "on the fly." This would permit us, for example, to allow the music generated in a video game to change depending on the player's interactions within the game. Alterations to the music could occur because of a change in mood, addition or removal of a musician, or direct request from a user.

We designed AMEE^{TM} such that one block (a small number of bars) is generated at a time and then output to MIDI. It is therefore possible to make changes between blocks to the music being produced. The difficulty in doing this is that music generation occurs much more quickly than playback, and thus, all the blocks are generated while the first one is playing. To support dynamic alterations, we need a means of keeping track of which block is currently being played and we need to be able to alter or replace subsequent blocks when a change is made.

Furthermore, we might want alterations to occur gradually rather than abruptly, or a mixture of the two. For a gradual change, we would need to know the parameters for starting and end points of the transition, and generate blocks in between accordingly. This gives rise to an additional checkpointing problem since we need to keep track of present and future parameters.

5.3. Better Generator Implementations

To improve and vary the music produced by AMEE^{TM}, it will be necessary to implement better generators. Our current prototype only represents a proof of concept; in the next steps of development we are looking to collaborate with musicians to produce different generators for various styles, and to extend the functionality which currently exists.

A few examples of improvements we have planned are the following. (This is by no means an exhaustive list!)

(a) For harmonic generators, we would like to consider transitions between blocks of music – if the previous block ended with an imperfect cadence to chord V, what should come next?
(b) Develop MotifGenerators for various styles of music
(c) Allow generated motifs to be stored in the pattern libraries to facilitate repetition and variations
(d) Some additional musical aspects which we would like to consider are: staccato vs. legato playing; better motifs for endings; motifs typical on different instruments (piano vs. violin vs. guitar vs. electronic etc.)

5.4. Extensions to Mood Implementation

Presently, although our implementation of the Mood class contains 66 emotional descriptors, the StandardEmotionMapper only alters music based on the emotions Happy and Sad. We need to do more research to determine which emotions we want to be able to vary, and how those emotions will be translated into changes to musical elements. Over the past 70 years, there have been many publications concerning emotional expression in music which we can draw on, but we might also need to do some empirical research in order to properly implement a full EmotionMapper. There is evidence that people only perceive certain emotions in music, but not others [8]. Maybe we will find that some terms collapse into others (perhaps Merry and Joyous are exactly the same in terms of musical expression?) All of these questions require more theoretical investigation.

5.5. Extensions to PatternLibraries

At present, only one of each type of pattern library can be used by the generators. We would like to make it possible for more than one library to be loaded into a generator, especially in the case of the MotifPatternLibrary. This would facilitate combining resources from more than one style in the same piece. Also, it would allow the user to try library combinations without having to make any changes to the libraries themselves. There would be some decisions to be made concerning how patterns are chosen if there are multiple libraries – are any patterns of the desired length returned, or only from one library? Are all the libraries accessed equally often?

We would also like all musicians to be able to use their own libraries. The main difficulty in using a different library for every musician is that it will necessitate storing and copying large amounts of data every time the musician is changed. On a PC this is not a problem, but in future it might be a concern if we try to implement AMEE^{TM} on cell phones or PDAs.

In addition, we would like to be able to add to pattern libraries during composition. This would be simple to implement, and would facilitate repetition in a piece, and also reuse of motifs in future compositions. The only difficulty would be in deciding which motifs to store – for a 3 minute song with 4 musicians, a typical number of generated motifs would be around 200, some of which we might not want to save. Also, if the same library were used many times, it would start accumulating an unreasonably large number of patterns.

5.6. Jam Sessions and Ensembles

We designed AMEE^{TM} such that musicians could play together in two ways: as a coordinated ensemble, which performs a piece with a harmonic structure followed by all the musicians, and as musicians jamming, in which case there is no decided upon harmonic structure, and all the musicians rely on their own musical knowledge to decide what to play. Currently, only the ensembles are implemented, although it is possible to generate a piece with no harmonic structure.

5.7. User Interface

For testing and demonstration purposes, we will be adding a graphical user interface which will allow the functionality of the software to be easily accessed.

6. Potential Applications

As mentioned in the Introduction, the versatile functionality in AMEE^{TM} allows it to be used as an embedded component in a larger software product such as a video game, or to form the basis for a stand-alone music composition application. In this section we mention two other potential applications.

6.1. Emotional Equalizer

A Mood is a collection of emotional descriptors, each present to a different degree. Imagine an “emotional equalizer” which would allow a listener to alter the music’s mood as it is playing. This could be either a hardware or a software device which would operate exactly like a normal stereo equalizer, except that the sliders would be marked with emotional descriptors rather than frequency ranges. So, while listening, rather than turning up the bass in the song, you could turn up the “happy.”

6.2. Long Distance Musical Collaboration

An application for AMEE^{TM} which is particularly relevant in Canada where people are spread over long distances, would Internet based musical collaboration. Picture three users, one in Iqaluit, one in Vancouver and one in Halifax, each of whom has a collection of Musicians with different qualities and abilities. Those Musicians could perform together and the results could be heard by all the users.

7. Conclusions

The AMEE^{TM} prototype described in this paper is a promising first step toward a dynamic music composition system. We are continuing development based on the extension ideas presented above, and we anticipate using AMEE^{TM} in several exciting application areas in the near future.

Acknowledgments

This research was funded in part by CITO/OCE and by Condition30 Inc.

References

[1] Charles Ames, *Automated composition in Retrospect*, Leonardo, 20(2), 169-185 (1987)

[2] Pietro Casella and Ana Paiva, *MAgentA: an architecture for real time automatic composition of background music*, Intelligent Virtual Agents, LNCS 2190, 224-232 (2001)

[3] Ryan J. Demopoulos, *Towards an Integrated Automatic Music Composition Framework*, MSc Thesis, Department of Computer Science, University of Western Ontario (2007)

[4] Kate Hevner, *Experimental Studies of the Elements of Expression in Music,* The American Journal of Psychology, 48(2), 246-268 (1936)

[5] Maia Hoeberechts, *API Design: Summary of Progress to Date*, Internal progress report, University of Western Ontario (2005)

[6] Steven R. Livingstone, Ralf Mühlberger, Andrew R. Brown and Andrew Loch, *Controlling musical emotionality: an affective computational architecture for influencing musical emotions*, Digital Creativity, 18(1), 43-53 (2007)

[7] Steven M. Pierce, *Experimental Frameworks for Algorithmic Film Scores*, MA Thesis, Dartmouth College (2004)

[8] Mark Meerum Terwogt and Flora Van Grinsven, *Musical Expressions of Moodstates,* Psychology of Music, 19, 99-109 (1991)

[9] Dynamic Object Music Engine (DOME), http://www.dometechnics.com, accessed August 23, 2007.

A Toolbox for Automatic Transcription of Polyphonic Music

Christian Dittmar, Karin Dressler, Katja Rosenbauer
Fraunhofer Institute for Digital Media Technology, Ilmenau, Germany
dmr@idmt.fraunhofer.de, dresslkn@idmt.fraunhofer.de, rosenbka@idmt.fraunhofer.de

Abstract. This publication introduces a software toolbox that encapsulates different algorithmic solutions directed towards the automatic extraction of symbolic note information from digitized music excerpts. This process, often referred to as automatic music transcription is still confronted with many issues such as mimicking the human perception or making a decision between ambiguous note candidates for symbolic representation. Therefore, the current publication describes algorithmic procedures dedicated to the detection and classification of drum notes, bass notes, main melody notes and chord structure. The focus on four different domains of automatic transcription allows utilization of specialized analysis procedures for almost every aspect of music. This paper provides insight into the single transcription methods and their performance. Additionally, various application scenarios for the transcription based interaction with music and audio are sketched with regard to the required technologies.

1. Introduction

Automatic note transcription refers to the extraction of a parametric representation (e.g. MIDI) of the notes played within a polyphonic music excerpt. In the field of Music Information Retrieval (MIR), this process is widely acknowledged as one of the most ambitious tasks in machine listening. Seemingly a research object of purely scientific interest, the automatic transcription of music potentially provides various fruitful application scenarios. The most obvious advantage of automatic transcription is the possibility to enrich large-scale music catalogues (e.g. online music shops) with high-level content descriptions. These advanced metadata in turn enable novel search and browsing functionalities. Listeners might for example be interested to retrieve solely songs in the key of G-minor or songs based on a characteristic Eastern European scale. In addition, a database of transcribed melodies in conjunction with specialized melody search strategies enables the well-known Query-by-Singing or Humming (QbSH) scenario. QbSH allows the user to retrieve a link to a song by just singing or humming the catchy tune in case he wants to find out the corresponding artist and title. At the same instant the user can be pointed to cover-versions or songs that feature a similar melodic theme. Creative communities enjoying music performance and production can also profit from automatic music transcription technologies. They allow aficionados to quickly practice their favorite song without being experienced with manual transcription. Indeed, first applications especially directed to generating guitar tabs from audio files are recently becoming commercially available[1]. Leisure games as well as educational software evolving around music, singing and dancing can be quickly enriched with content and user generated assets using automatic transcription technologies.

1.1. Problem definition

A symbolic transcription representation is an organized sequence of consecutive notes and rests. A note is characterized by its pitch (note name), starting time (onset) and ending time (offset). This notation has to be automatically extracted from digitized excerpts of real world music, where usually a complex mixture of harmonic sustained (e.g. melody) instruments as well as the percussive un-pitched (e.g. drum) instruments is present. Automatic transcription in its original meaning also implies the need for the extraction of the musical context, e.g., the tempo, the bar measure, repetitions etc. However, the MIR community has adopted the term transcription for the process of detecting and classifying notes. For the special case of drum instrument classification this term will also be used because drum patterns are usually notated on the different staves of a score sheet.

1.2. Related work

Automatic music transcription is a very challenging task, both from the signal-processing as well as the musicology point of view. There still exist no definitive solutions for problems like mimicking the human perception or coming to a decision between ambiguous candidates for symbolic representation. The challenge with transcription of percussive instruments resides in the fact, that a great variety of sounds can be generated using a single instrument. An overview of publications concerned with drum detection and classification can be found in [1], where a unique template adaptation and matching technique in conjunction with harmonic structure suppression for detection and identification of common drum instruments is described. Many papers have focused on sound source separation methods decomposing a music spectrogram into multiple spectrograms of source instruments. These classes of algorithms are realized by higher order statistical methods such as Prior Subspace Analysis [2], Non-Negative Independent Component Analysis [3] or Non-Negative Spectrogram Factorization [4]. An alternative approach is described in [5], where a noise sub-space projection method is proposed. An adaptive feature modeling method has been introduced in [6].

The transcription of bass-lines in real-world music has actually been underrepresented in the MIR literature. The most well known approaches are described in [7] and [8]. To the current knowledge of the authors, the only newer publication is [9], which proposes a multiple fundamental frequency (F0) estimator as a front-end followed by acoustic and musicological models. The method described in the present paper outlines and extends the methodology given in [10].

Melody transcription is a very active research topic. The conventional approach is to estimate the F0-trajectory of the melody within polyphonic music, such as in [11], [7], [12], [13]. An alternative class of transcription algorithms directly detects discrete notes as a representation of the melody [14], [15].

Detection of chords in a piece of music is closely related to the well-investigated research field of key estimation [16], [17], but has attracted less attention in the past years. Examples are [18]

[1] Examples can be found on http://www.sienzo.com and http://www.daccordmusic.com

that use HMMs to model chord transitions, and [19], [20] that introduce chord detection in order to analyze and segment pieces of music. The latter also shows the interlacement of key finding and chord detection in their approach. The current publication summarizes the method described in [21].

The remainder of this paper will be organized as follows: Section 2 provides a brief overview of the four transcription algorithms dedicated to the different musical domains as well as the overall system architecture. Section 3 describes the evaluation procedures for every algorithm and depicts the achievable results accordingly. Section 4 sketches application scenarios with special focus on interaction with sound. Section 5 concludes this paper and summarizes the most important insights.

2. System overview

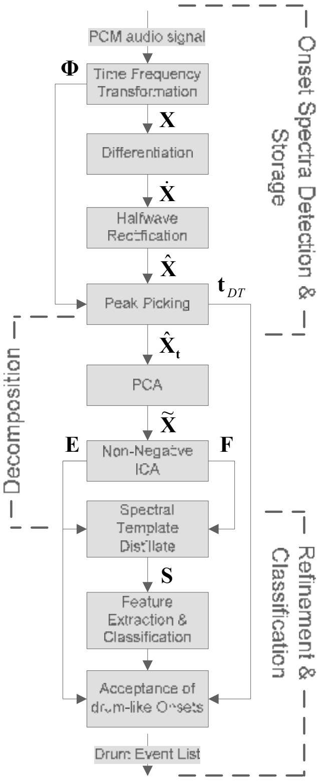
The transcription toolbox itself is just a graphical user interface (GUI) encapsulating the four different transcription technologies as well as appropriate methods to display the results (i.e. notes in a piano-roll view). It also features a very basic synthesizer for rendering the detected notes and drum instruments in sync with the original music excerpt. Thus, the toolbox is suited to quickly assess the performance of the single transcription technologies and test new implementations. The GUI enables the user to load an audio file (WAV or MP3), to select an excerpt that shall be analyzed, to run all four transcription methods, to view and listen to the results and also to save all results as a MIDI file. Per default all audio data is internally converted to 44.1 kHz sampling rate and 16 bit per sample on loading.

2.1. Drum Transcription

This publication focuses on the vast field of popular music where only a limited set of percussive un-pitched instruments is presumed to be present. There are mainly two instrument classes in scope: membranophones and idiophones (as well as their electronic counterparts). The drum transcription algorithm described in this paper is able to identify up to 17 distinct drum and percussion instruments in real-world music and generate a MIDI notation of their onsets. An overview of the drum transcription algorithm is presented in figure 1. It shows that the signal processing chain can be subdivided into three main stages that will be described in the following sections.

2.1.1. Onset Spectra Detection & Storage

A time-frequency representation of the audio excerpt is computed using a conventional Short Time Fourier Transformation (STFT). Thereby a relatively large block-size in conjunction with 10 ms hop-size is applied. The unwrapped phase-information $\mathbf{\Phi}$ and the absolute spectrogram values $\mathbf{X}$ are taken into further consideration. The time-variant slopes of each spectral bin are differentiated over all frames in order to decimate the influence of sustained sounds and to simplify the subsequent detection of transients. Using half-wave rectification, a non-negative difference-spectrogram $\hat{\mathbf{X}}$ is computed for the further processing. The detection of multiple local maxima positions $\mathbf{t}_{DT}$ associated with transient onset events in the musical signal is conducted by means of peak picking in a detection function derived from summing up all bins of $\hat{\mathbf{X}}$ and smoothing the resulting vector. Two verification instances aim at avoiding the acceptance of small ripples for onsets. First, a tolerance interval of 68 ms is defined which must at least occur between two consecutive onsets. Second, the unwrapped phase information of the original spectrogram serves as reliability function in this context. It can be observed that a significant positive phase jump must occur near the hypothetic onset-times. The main concept of the further process is the storage of one spectrum frame of $\hat{\mathbf{X}}$ at the time of the onset. From the manifold of collected onset spectra the significant spectral profiles related to the involved instruments will be gathered in the next stages.



PCM audio signal
Φ
Time Frequency Transformation
X
Differentiation
Ẋ
Halfwave Rectification
X̂
Peak Picking
t_DT
X̂_t
PCA
X̃
E
Non-Negative ICA
F
Spectral Template Distillate
S
Feature Extraction & Classification
Acceptance of drum-like Onsets
Drum Event List
Onset Spectra Detection & Storage
Decomposition
Refinement & Classification

Figure 1: Drum transcription process

2.1.2. Decomposition

From the steps described in the preceding section the onset times $\mathbf{t}_{DT}$ as well as the corresponding onset spectra $\hat{\mathbf{X}}_t$ is deduced. With regard to the goal of finding only a few significant subspaces, Principal Component Analysis (PCA) is applied to $\hat{\mathbf{X}}_t$. Using this well known technique it is possible to decimate the number of collected spectra to a limited set of d de-correlated and variance normalized (whitened) principal components $\tilde{\mathbf{X}}$. The whitened components $\tilde{\mathbf{X}}$ are subsequently used as input for a Non-Negative Independent Component Analysis (NN-ICA) in order to acquire the original source components. NN-ICA utilizes the intuitive concept of optimizing a cost function describing the non-negativity of the components [22]. This cost function is related to the reconstruction error introduced by axis pair rotations of two or more variables in the positive quadrant of the joint probability density function (PDF). The assumptions for this model imply that the original source signals are positive and well grounded, i.e. they exhibit a non-zero PDF at zero, and they are to some extent linearly independent. The first concept is always fulfilled because the vectors subjected to NN-ICA originate from $\hat{\mathbf{X}}$, which does not contain any values below zero, but certainly some values at zero. The second constraint is taken into account when the spectra collected at onset times are regarded as the linear combinations of a small set of original source-spectra characterizing the involved instruments. This seems, of course, to be a rather coarse approximation, but it holds up well in the majority of the cases. The onset-spectra of real-world drum instruments do not exhibit invariant patterns, but are more or less subjected to changes in their spectral composition. Nevertheless, however, it may safely be assumed that there are some characteristic properties inherent to spectral profiles of drum-sounds that allow us to separate the whitened components $\tilde{\mathbf{X}}$ into their potential sources $\mathbf{F}$ by computing $\mathbf{F} = \mathbf{A} \cdot \tilde{\mathbf{X}}$. Here, $\mathbf{A}$ denotes the $d \times d$ un-mixing matrix estimated by the NN-ICA. The sources $\mathbf{F}$ will be named spectral profiles since they represent the onset spectra of the source signals. The spectral profiles are used to extract the spectrograms amplitude basis, hereafter referred to as amplitude envelopes, by computing $\mathbf{E} = \mathbf{F} \cdot \mathbf{X}$. The extracted amplitude envelopes present relatively salient detection functions with sharp peaks, sometimes accompanied by smaller peaks and plateaus stemming from crosstalk effects. Drum-like onsets are detected in the amplitude envelopes using conventional peak picking methods. Solely peaks near the original times $\mathbf{t}_{DT}$ are accepted as candidates (68 ms tolerance). The value of the amplitude envelope's magnitude is assigned to every onset candidate at its position. If this value does not exceed a certain dynamic threshold then the onset is not removed.

2.1.3. Refinement & Classification

Using the information about the probable onset times, a spectrogram template is extracted for every valid component. This refinement step is comparable to the methodology used in [1]. It can be achieved by peering through excerpts of the original spectrogram $\mathbf{X}$ near the times corresponding to the detection function's highest local maxima. The original spectrogram is taken into account to obtain multiple observations of the instruments in the time-frequency domain, from which a distillate of the actual instruments spectrogram can be derived. Multiple instances of such templates are extracted at different onset times, ordered descending by the magnitude of the peaks in the corresponding amplitude envelope. This procedure is motivated by the assumption, that the original instrument is probably present at those points with high amplitudes. By virtually stacking all preliminary templates on top of each other and performing element-wise minimum computation, it is possible to capture the final spectrogram template $\mathbf{S}$ per instrument. It turned out that a relatively small number of template observations is sufficient to distillate $\mathbf{S}$. The template represents the detected drum instruments spectrogram and exhibits minimized interference of other instruments. Since it tends to smooth out spectral variation caused by slight playing variations of the drums it is suited for feature extraction and classification of the drums sounds. Classification itself is based on a linear combination of multiple MPEG-7 compliant audio features directly computed from the mean template in the sense of linear discriminant analysis (LDA). The LDA transformation coefficients are computed on features extracted from a comprehensive collection of isolated drum sounds with corresponding class labels. Categorization of the detected drum instruments to the predefined classes is finally achieved by simple nearest neighbor classification using Euclidean distance measure in the LDA space.

2.2. Bass Transcription

The bass transcription is the fastest extraction procedure of the toolbox, because it only concentrates on the low-frequency musical events and thus allows for a rather strong sub-sampling of the audio signal, discarding irrelevant high-frequency content. The bass transcription processing chain can also be subdivided into three main stages as depicted in figure 2 and described in the following sections.

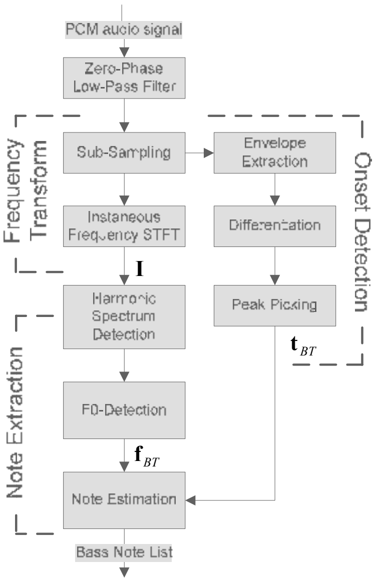


Figure 2: Bass transcription process

2.2.1. Frequency Transform

As already stated, the audio signal to be analyzed for bass notes is sub-sampled by factor 32, thus reducing the Nyquist frequency to approximately 700 Hz. It can safely be assumed, that the majority of bass notes and their most important harmonic overtones are captured in that frequency range. It is obvious that this rather strong sub-sampling would normally lead to aliasing effects which would strongly interfere with the wanted signal. To prevent these effects, a zero-phase low-pass filter is applied before the sub-sampling. This well-known filter method virtually doubles the filter order and has precisely zero phase distortion. The sub-sequent frequency transform is given by an STFT with additional estimation of the instantaneous frequency (IF) spectrogram $\mathbf{I}$ estimated from successive phase spectra via the well-known phase vocoder method [23]. This method computes the evolution of the instantaneous frequency per spectrogram bin, and is thus much more suitable to analyze the low-frequency content of a music signal, where slight frequency deviations may cause erroneous note estimation.

2.2.2. Onset Detection

In parallel to the frequency transform, a time-domain based onset detection is conducted on the low-pass filtered and sub-sampled signal. The detection of candidate note $\mathbf{t}_{BT}$ onsets is realized as conventional envelope extraction based on two-wave rectification and low-pass smoothing. It is followed by differentiation of the envelopes and detection of rising slopes exceeding a dynamic threshold criterion.

2.2.3. Note Extraction

A detailed spectral analysis is deployed in this stage to overcome the common problem that drum instruments overlap with the bass notes in the low-frequency range of music signals. The main issue is that drum instruments sounding simultaneously to a bass note can have a negative influence on the F0-estimation of this note. To prevent this behaviour, a frame-wise criterion for the reliability of the low-frequency spectrum is computed as a combination of the spectral flatness measurement (SFM) and the peak energy of the spectrum's autocorrelation function (ACF). The motivation for the latter measurement is based on the observation that drum spectra will lead to a less spiky shape of the ACF than harmonic spectra, whose overtones cause salient periodicities along the frequency axis. This property is exploited to retrieve the part of a bass note, where the harmonic spectrum is most salient and least superposed with drum spectra. Actually, the attack part of the bass notes is ignored in most cases when a drum instrument is present at the same onset. Instead the sustain phase or even the release phase are considered most reliable for the fundamental frequency estimation. In this context it is also an advantage, that in the majority of cases, the bass notes tend to be longer than the competing drums sounds, which can be considered as one-shot events. The fundamental frequency of the bass notes is estimated in the time intervals that are marked as reliable via a method similar to [13]. This pitch estimation also includes the frequency of identifiable overtones in order to refine F0. The reason for that procedure is that physical bass instruments usually have slightly detuned overtones (e.g. due to string thickness) that influence the overall pitch impression and must be accounted for. As a concluding step the estimated fundamentals $\mathbf{f}_{BT}$ are mapped to MIDI notes. The corresponding note offsets are defined as the points where the reliability measure decreases below a certain percentage of the maximum that had been observed for the particular note object.

2.3. Melody Transcription

The melody transcription algorithm is tailored towards identifying the notes of preferably monophonic instrument phrases in polyphonic and multi-timbral music excerpts. Figure 3 shows the overview over the complete processing chain, the single stages will be described in the following sections.

2.3.1. Spectral Analysis

A multi resolution spectrogram representation $\mathbf{M}$ is obtained from the audio signal by calculating an STFT with varying length of zero padding using a Hann window. Thereby, a Multi Resolution-FFT (MRFFT), an efficient technique used to compute STFT spectra in different time-frequency resolutions, is utilized [24]. For all spectral resolutions the resulting STFT frame size and the hop size of the analysis window are 2048 and 256 samples, respectively. This processing step is followed by the computation of the magnitude and phase spectra and the IF spectrogram $\dot{\mathbf{M}}$.

2.3.2. Pitch Estimation

Sinusoidal components of the audio signal contain the most relevant information about the melody. Yet, it is a challenge to reliably identify sinusoidal partials in polyphonic music. Of course a consistent and moderate change in magnitude and frequency of the examined spectral peaks is a good criterion for the identification of sinusoids. However, this requires a continuous tracking of the partials with time, a demand which cannot be implemented easily for polyphonic audio signals. For this reason a psychoacoustic model is applied in order to exclude non-audible peaks from further processing. A simplified implementation of simultaneous and temporary masking is used, which by far does not reach the complexity of models used in modern lossy audio coders. However, this way many spurious peaks can be erased from the spectrum in order to speed up the further processing. The magnitude and instantaneous frequency of the audible peaks $\mathbf{p}_{MT}$ are extracted by a pitch estimation method, as the frequency of the strongest harmonic may not be the perceived pitch of a periodic complex tone. At first, the pitch estimator performs a magnitude weighting and then it analyzes the harmonic structure of the polyphonic signal. The algorithm covers four octaves – computing pitch frequencies and an approximate prediction of the pitch salience in a frequency range between 80 Hz and 1280 Hz.

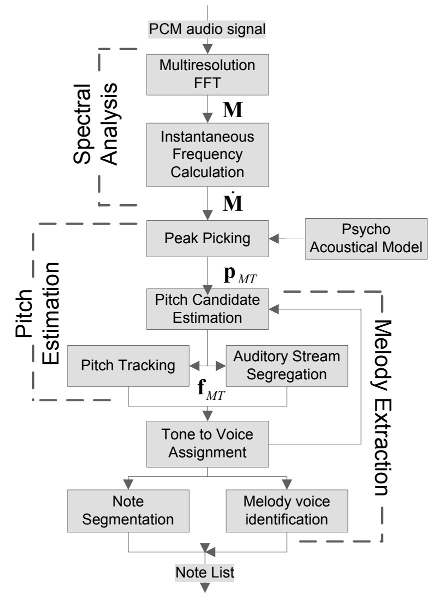


Figure 3: Melody transcription process

2.3.3. Melody Extraction

A variable number of pitch candidates $\mathbf{f}_{MT}$ at each frame (about five pitches on average) is used to track tone objects. At the same time the frame-wise estimated pitch candidates are processed to build acoustic streams. As only the magnitude and the frequency of the pitch candidates are taken into account for the auditory stream segregation, the method does not mimic the performance of human listeners. The implemented method solely subdivides the full four octave frequency range into smaller voice regions. Tones which have a sufficient magnitude and are located in the adequate frequency range are assigned to the corresponding voice. Every voice may possess only one active tone at any time. In competitive situations the active tone is chosen with the help of a rating method that evaluates the tone magnitude and the frequency difference between the pitch of the tone and the actual voice frequency. Conversely, a tone is exclusively linked to only one voice. That means solely tones from the most salient voice are considered to be melody tones. Therefore, the correct identification of the melody stream is very important for the success of the method. For each tone in a voice, the onset and offset is detected by evaluating the magnitude and frequency deviation of the tone. Moreover, a stable pitch frequency is assigned with the help of a rating scheme. In order to determine a discrete note name, the short-term tuning frequency is estimated for each voice as proposed in [25]. Finally, the melody voice must be chosen. In general the most salient voice is identified as the melody. Of course it may happen that two or more voices have about the same magnitude and thus no clear decision can be made. In this case, the stream magnitudes are weighted according to their frequency. Streams from the bass region receive a lower weight than streams from the mid and high frequency regions. If no clear melody stream emerges during a short time span, the most salient weighted stream is chosen.

2.4. Chord Transcription

The extraction of the harmonic structure comprises the detection of as many chords as possible in a piece. That includes the characterization of chords with a key and type as well as a chronological sequence with onset and duration of the chords. The current implementation is able to identify the five most common triads (maj, min, dim, aug, sus4) as well as 10 more exotic quads (that augment the triads with sevenths and ninths). Figure 4 shows the system overview of the harmony transcription algorithm. The proposed method focuses on deriving the utmost degree of information from the signal and only needs basic musical knowledge, but works without training or a priori knowledge about the pieces to be processed.

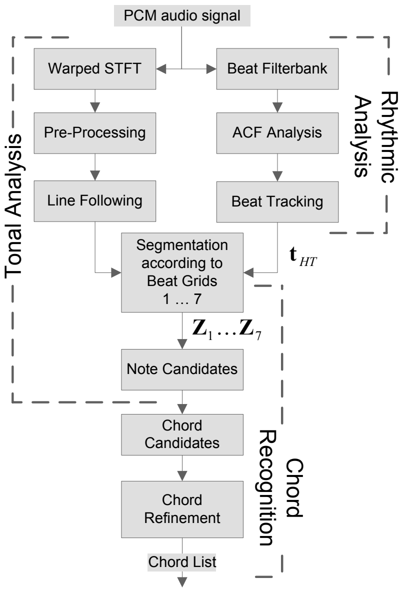


Figure 4: Chord transcription process

2.4.1. Tonal Analysis

The chord detection is based on the analysis of the most salient pitches in a spectrogram representation of the audio material. As front-end for the spectral transformation, a combination of a warped FFT [26] for the lower frequency range up to 1.5kHz and conventional FFT for the upper frequency range up to 8kHz is used. As a result, a spectrogram with a frequency resolution of two to six bins per half-tone for the whole frequency range from 100Hz to 8kHz is obtained with a hop-size of 8ms. Furthermore, the spectrogram is logarithmized and aurally compensated to derive amplitudes representing perceived loudness. To find the most salient pitches in each FFT-frame, a procedure inspired by [7] is used. For each frequency bin in the desired frequency range, the amplitudes of all hypothetic harmonics are summed up. To obtain the most salient pitches, the eight strongest local maxima in the resulting sum spectrum are chosen as note candidates. In this reduced sum spectrogram, continuous F0-trajecories are sought. From this set of note candidates, any note whose frequency is an integer multiple of a note sounding simultaneously is removed to avoid erroneous interpretation of higher harmonics as fundamental frequencies. Furthermore, notes with a total duration shorter than 80ms are removed, as they do not form stationary tonal events and thus do not carry relevant pitch information.

2.4.2. Rhyhtmic Analysis

The rhythmic analysis is intended to extract the beat grid, i.e., the times where main rhythmic accents are situated (e.g. the instances that a human individual would tap along while listening to the song). This analysis is carried out as a parallel processing branch, where the audio signal is first decomposed into six band signals using a filter-bank adjusted to the frequency ranges of the most common percussion instruments. Amplitude envelopes are derived from the single band signals by means of two-wave rectification and low-pass smoothing. Multiple ACFs of overlapping envelope excerpts are computed throughout the whole song in order to find the most salient rhythmic periodicities contained in the music signal. The periodicities are represented by the salient peaks of the ACF functions. Their lag positions can be directly converted to BPM values. One of them represents the actual tempo inherent to the song. Unfortunately it does seldom correspond to the absolute maximum. Several tempo (resp. beat) candidates are tracked throughout the song and a final decision about the tempo and the beat points is deduced when the complete audio signal has been processed. This way, the final beat grid $\mathbf{t}_{HT}$ is available as information source for the chord recognition stage described in the following section.

2.4.3. Chord Recognition

Utilizing the beat grid $\mathbf{t}_{HT}$, multiple hypotheses for corresponding rhythmic segmentation are generated. To prevent propagation of eventually occurring octave-errors in the beat grid, different kinds of segmentations are set up according to following procedure. The step-width from one beat point to the next is multiplied using the integer factors $\mathbf{a} = \{1,2,3,4\}$ (comparable to conventional sub-sampling). Due to the fact, that no information on the actual on-beat position is available, additional shifted segmentations are generated for all factors greater than one. Using this scheme it can safely be assumed that one of the segmentations corresponds to the actual beat-structure of the analyzed song. For each segmentation $\mathbf{Z}$, every beat interval is analyzed for its tonal content separately. Therefore, all F0-trajectories starting, crossing or ending in that interval are subjected to the note candidate extraction. These candidates are estimated using a statistical stabilization of the trajectories apparent in the interval, followed by an assignment to the nearest musical note on a chromatic scale. In a subsequent step, octave information of the notes is discarded by mapping them to pitch classes analogous to those proposed in [27]. The entries in the chroma-vector representing one beat are given by accumulating the note candidates energy sum normalized by the ratio of the note duration to the interval duration. A rough key estimation is carried out from these pitch classes, using the enhanced probe tone algorithm according to [16]. In contrast to the somehow similar approach in [28], all probable chords are formed from the chroma-vector entries that exceed a dynamic threshold criterion. The most plausible chord sequence is derived in the chord refinement stage with respect to an energy related confidence measure as well as the fitness to the previously estimated key. The confidence measure is also used to decide, whether there are any audible chords or rests. Furthermore, it also provides the cue which sequence of chords is the most stable and which chord changes are certain. This way, one of the segmentation hypotheses and its corresponding chord sequence is picked as the final result and transformed into a MIDI notation.

3. Evaluation

The assessment of MIR algorithms is a crucial and often discussed task. During recent years the international competition called Music Information Retrieval eXchange (MIREX) has established widely acknowledged test criteria, and even more important, has provided a platform for exchange of ideas in the MIR community. Thus, MIREX-results or parts thereof will be given in this paper were possible. All evaluation methods found in this paper will make use of the well-known information retrieval measures precision, recall and F-measure, only the melody evaluation uses different measures, that will be described below.

3.1. Drum Transcription Evaluation

The performance assessment of the presented drum transcription algorithm is given by an excerpt of the results measured during the so-called 'Audio Drum Detection' evaluation task conducted during the MIREX2005[2].

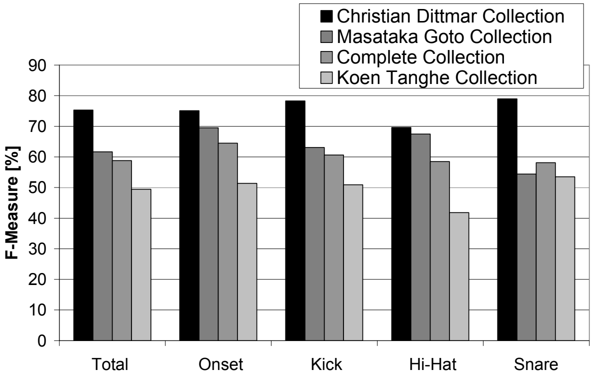


Figure 5: Drum transcription test results given by descending F-measure per test target

3.1.1. Drum Transcription Test Data

Within the MIREX2005 contest, approximately 50 audio files with corresponding drum annotations were used as test-bed for the evaluation. Many genres were encompassed and various degrees of drum density (with regard to instrumentation as well as intensity) were contained in the files. Although the drum transcription method presented here does support up to 17 different types of drum and percussion instruments, the conditions for the MIREX task only required the discrimination of three drum types (kick, snare, hi-hat). Three test collections provided by the participants were used, with a minimum duration of 30 seconds up to complete songs. To quantify the performance of each algorithm the participants agreed to use the F-measure for each of the three drum types as well as the onsets, resulting in four F-measure scores and their average score.

3.1.2. Drum Transcription Test Results

All participating algorithms were evaluated against music from each individual audio file collection, and then the three collection scores were averaged to produce a composite score, in which the method presented in this document positioned itself at rank five. It was outperformed by the winning algorithm of Kazuyoshi Yoshii and three algorithmic variations submitted by Koen Tanghe. Figure 5 shows the results achieved by the proposed method. It can clearly be seen that the numbers vary quite drastically for the different test-sets.

3.2. Bass Transcription Evaluation

The performance assessment of the presented bass transcription algorithm has been described in detail in [10].

3.2.1. Bass Transcription Test Data

A test collection of 37 music excerpts from a broad range of genres has been assembled for testing purposes. The majority of files feature a length of approximately 30 seconds. The whole collection amounts to a total of 15 minutes audio material. The test criterion was again the F-measure. The recall was computed in three versions, the first only taking exact note matches as valid, the second tolerating octave errors and the third tolerating fifth (i.e. 7 half-tones deviation) errors.

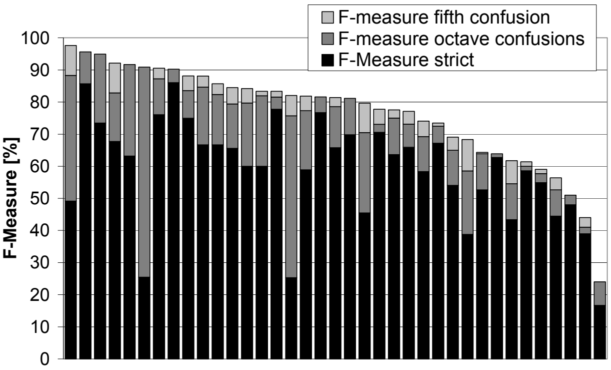


Figure 6: Bass transcription test results given by descending total F-measure per test item

3.2.2. Bass Transcription Test Results

Figure 6 shows the results achievable with the proposed bass transcription method. It can clearly be seen, that a rather high number of octave errors occur, which in turn are acceptable to a certain extent, since the bass notes will still sound consonant with the original audio excerpt. These erroneous detections will only sound disturbing when they occur amidst a staccato line of equal notes.

3.3. Melody Transcription Evaluation

The performance assessment of the presented melody transcription algorithm is given based on an excerpt of the results documented in the task 'Audio Melody Extraction' of MIREX2006[3]. The aim of the task was to extract melodic content from polyphonic audio. The transcription problem was divided into two subtasks, namely the melody pitch estimation and the distinction of melody and non-melody parts (voiced/unvoiced detection). It was possible to give a pitch estimate even for those parts that have been declared unvoiced.

3.3.1. Melody Transcription Test Data

The algorithm was tested on two manually annotated datasets, the MIREX2005 dataset, consisting of 25 audio files, and the ADC2004 dataset, consisting of 20 audio files, both across different music styles. Moreover, each of these datasets were split into files in which the predominant melody is sung and files in which the predominant melody is non-vocal. The test criteria were actually not bound to note extraction but instead to F0-detection, voicing detection.

3.3.2. Melody Transcription Test Results

The MIREX2006 evaluation results show that the presented algorithm performs best in pitch detection and melody extraction. As indicated by the excellent runtime of the algorithm (fastest among the participants), the implemented methods allow a very efficient computation of the melody pitch contour. The following measures are given in diagram 7: Voiced Recall is the probability that a frame which is truly voiced is labeled as voiced. Voiced False Alarm is the probability that a frame which is not actually voiced was mis-labeled as voiced. Raw Pitch Accuracy is the probability of a correct pitch value (within a tolerance interval of ± ¼ tone) given that the frame is indeed pitched. This includes the pitch guesses for frames that were judged unvoiced. Overall Accuracy combines both the voicing detection and the pitch estimation to give the proportion of frames that were correctly labeled with both pitch and voicing.

[2] Further details at http://www.music-ir.org/mirex2005/

[3] Further details at http://www.music-ir.org/mirex2006/

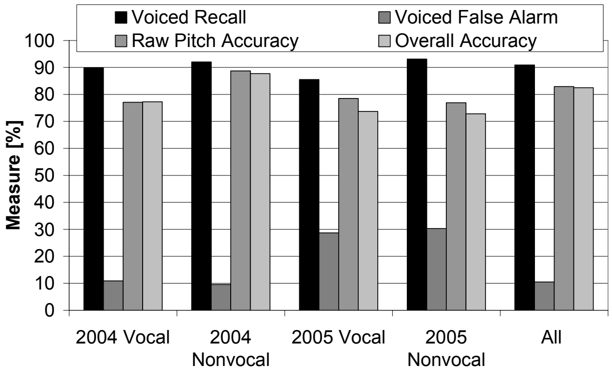


Figure 7: Melody transcription test results

3.4. Chord Transcription Evaluation

The performance assessment of the presented chord transcription algorithm has been described in detail in [21]. Here, only the most important findings will be given in compact form.

3.4.1. Chord Transcription Test Data

The chord test-set consists of three re-synthesized MIDI files for which notes and chord transcriptions are available and 28 one minute excerpts of pop/rock songs for which note and chord transcriptions where taken from guitar and keyboard magazines as well as commercially available song books. The utilized ground truth comprises all detectable chords in the pieces with their tonics, types and temporal positions (onset and duration). The recall and precision were computed on a frame basis.

3.4.2. Chord Transcription Test Results

The results obtained with the proposed algorithm are presented in diagram 8. For better readability, the values are sorted descending over all test items in accordance to the total F-measure achievable with tolerance to enharmonic and triad-quad confusion. It can be observed, that for a number of songs, there are significant confusions between annotated triads and detected quads or vice versa. When checking the results via re-synthesized notes played simultaneously to the original music excerpt, these errors are tolerable since the synthetic rendition still sounds consonant to the audio material.

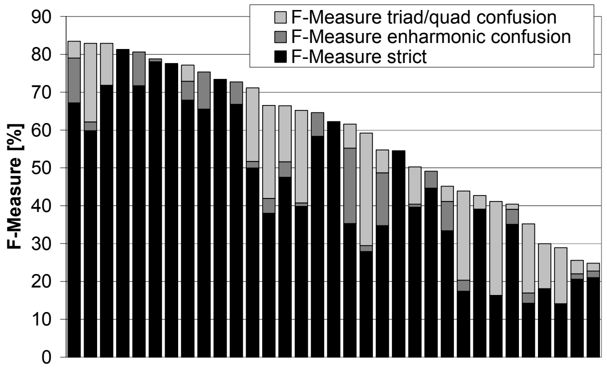


Figure 8: Harmony transcription test results given by descending total F-measure per test item

4. Interactive application scenarios

This section will sketch various interactive application scenarios for which the existence of automated transcription methods for music is of crucial importance. At present, not all of these scenarios are feasible but constant evolution of automatic transcription methods will enable them in the future.

4.1. Music Education

Many music education books are sold in combination with instructional records[4]. The obvious advantage of such additional material is the possibility to convey a higher degree of insight into particularities of the targeted musical performance, in addition to the information that can be found in the score sheets. In this respect, it is only consequent to also record the music student's rendition of the piece and compare his skills to the instructional audio. In this context, automatic transcription methods can point out mistakes in the student's playing or special intonation details that he has to practice more. Additionally the student can derive note transcriptions from any song he likes. Even if the transcription is not perfect on the first hand it is a good starting point for generating his own practice sheet and can support aural training.

4.2. Remote Composition

These days, thousands of amateur composers and musicians possess a bulk of unfinished or sketchy content like music backing tracks or single instrument tracks on their personal hard drives. Often they are faced with very ordinary problems like a general lack of inspiration, instrumental skills, technical devices or even a partner to discuss creative ideas with. Since their creativity is mostly not bound to commercial interests, they sacrifice their spare time for the hobby and can not afford to pay for professional instrumentalists or expensive sampling CDs. Automatic transcription methods can facilitate the composition by enhanced search for content or users, closing the creative gap that prevents hobby artists from realizing their ideas. An innovative search interface would for example be a world map showing participating users based on their location and transforming that view with respect to melodic similarity of their shared music items.

4.3. Computer aided Remixing

During the last years so-called Mash-Ups have become more and more popular. With the help of suitable audio sequencer programs almost every user can produce Mash-Ups by mixing excerpts of two or more popular songs using pitch-shifting and time-stretching to synchronize the music with regard to rhythm and melody. With the help of automatic transcription technologies the experimental guess-work necessary for creating a Mash-Up can be facilitated by automatically finding song segments suited for mixing and proposing the necessary temporal and tonal alignment. This concept can even be pushed further to an automatic DJ player, which plays a never ending stream of music from the user's collection with beat matching and a varying degree of manipulation. In contrast to dance mixes, bio-feedback controlled relaxation music can also be generated. For example, a steady stream of gently interweaved songs from the user's collection can be played in conjunction with synthesized melodies and chords, while the music selection and control of parameters is derived from e.g., the listeners pulse rate. The automatic transcription is thereby needed to retrieve matching music and to adopt the synthetic motives to the harmonic structure and the tempo of the original songs. This is indispensable in order not to interrupt the relaxation effect with dissonant tones.

4.4. Singing and Dancing Games

There exist a number of very popular music related games, which evolve around singing, dancing, drumming or similar

[4] An outstanding example is the series 'Play Guitar with the Ventures', the first and only instructional albums that entered the American album charts.

activities using specialized controllers. Normally the producers of these titles publish two or more titles per year that usually contain new content instead of new game features. Using automatic transcription, alternative games can be thought of where the user is able to choose the music he wants to play with. Dancing steps, singing references and drum patterns to play along can be quickly generated from the music itself.

4.5. Personalized Game Soundtracks

With modern day computer games, a lot of effort is put into composing and placing soundtrack music. Through careful synchronization to in-game events a lot of additional immersion can be achieved. With the increasing demand for individualization of game contents (e.g. avatar design) it would be only consequent to also personalize the soundtrack by using items from the user's private collection. Automatic transcription technologies are perfectly suited for that scenario. With their help, the mood and tension contained in songs can be estimated and the corresponding in-game situations can be enhanced with surprising musical layers.

5. Conclusions

In this publication, a set of transcription methods for the important music aspects drums, bass, melody and chords has been described. The dedicated algorithms are encapsulated by a comprehensive software framework allowing quick inspection of the achievable transcription results. Evaluation procedures and results for every single method have been described and discussed. Innovative concepts for interaction with sound and music through the help of automatic transcription have been sketched to show that automatic music transcription is not solely an academic playground.

References

[1] K. Yoshii, M. Goto, H.G. Okuno, *Drum Sound Recognition for Polyphonic Audio Signals by Adaptation and Matching of Spectrogram Templates With Harmonic Structure Suppression*, in IEEE Transactions on Speech and Audio Processing, Volume: 15, Issue: 1, pp: 333-345, (2007)

[2] D. Fitzgerald, B. Lawlor, E. Coyle, *Drum transcription in the presence of pitched instruments using prior subspace analysis*, in Proc. Irish Signals Syst. Conf. (ISSC), pp. 202–206, (2003)

[3] C. Dittmar, C. Uhle, *Further Steps towards Drum Transcription of Polyphonic Music*, in Proc. Audio Engineering Society 116th Convention (AES), (2004)

[4] J. Paulus, A. Klapuri, *Drum Transcription with Non-Negative Spectrogram Factorisation*, in Proc. European Signal Processing Conf. (EUSIPCO), (2005)

[5] O. Gillet, G. Richard, *Drum track transcription of polyphonic music using noise subspace projection*, in Proc. Int. Conf. Music Information Retrieval (ISMIR), (2005)

[6] V. Sandvold, F. Gouyon, P. Herrera, *Percussion classification in polyphonic audio recordings using localized sound models*, in Proc. Int. Conf. Music Information Retrieval (ISMIR), (2004)

[7] M. Goto, *A real-time music-scene-description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals*, in Speech Communication, vol. 43, no. 4, pp. 311–329, (2004)

[8] S.W. Hainsworth, M.D. Macleod, *Automatic bass line transcription from polyphonic music*, In Proc. International Computer Music Conference (ICMC), (2001)

[9] M. P. Ryynänen, A. Klapuri, *Automatic bass line transcription from streaming polyphonic audio*, In Proc. IEEE Int. Conf. Acoustics,Speech and Signal Processing (ICASSP), (2007)

[10] T. Korn, *Untersuchung verschiedener Verfahren zur Transkription von Basslinien aus dem Audiosignal*, Diploma Thesis Technical University Ilmenau, (2005)

[11] J. Eggink, G. J. Brown, *Extracting melody lines from complex audio*, in Proc. Int. Conf. Music Information Retrieval (ISMIR), (2004)

[12] M. Marolt, *Audio melody extraction based on timbral similarity of melodic fragments*, in Proc. Int. Conf. "Computer as a tool" (EUROCON), (2005)

[13] K. Dressler, *Extraction of the melody pitch contour from polyphonic audio*, in Proc. 6th Int. Conf. Music Information Retrieval (ISMIR), (2005)

[14] G. E. Poliner, D. P.W. Ellis, *A classification approach to melody transcription*, in Proc. 6th Int. Conf. Music Information Retrieval (ISMIR), pp. 161–166, (2005)

[15] R. P. Paiva, T. Mendes, A. Cardoso, *On the detection of melody notes in polyphonic audio*, in Proc. 6th Int. Conf. Music Information Retrieval (ISMIR), pp. 175–182, (2005)

[16] D. Temperley, *The Cognition of Basic Musical Structures*, The MIT Press, (2001)

[17] O. Izmirli, *Template based key finding from audio*, In Proc. International Computer Music Conference (ICMC), pp. 211–214, (2005)

[18] A. Sheh, D. P. W. Ellis, *Chord segmentation and recognition using EM-trained hidden markov models*, In Proc. 4th Int. Conf. Music Information Retrieval (ISMIR), pp. 183–189, (2003)

[19] N. C. Maddage, C. Xu, M. S. Kankanhalli, X. Shao, *Content-based music structure analysis with applications to music semantics understanding*, In Proc. ACM Multimedia, pp. 112–119, (2004)

[20] A. Shenoy, Y.Wang, *Key, chord, and rhythm tracking of popular music recordings*, Computer Music Journal 29(3), 75–86, (2005)

[21] K. Rosenbauer, *Entwicklung eines Verfahrens zur Analyse der Harmoniestruktur von Musikstücken unter Einbeziehung von Rhythmus- und Taktmerkmalen*, Diploma Thesis Technical University Ilmenau, (2006)

[22] M. Plumbley, *Algorithms for Non-Negative Independent Component Analysis*, in IEEE Transactions on Neural Networks, 14 (3), (2003)

[23] J. L. Flanagan, R. M. Golden, *Phase vocoder*, Bell System Technical Journal, pp. 1493–1509, (1966)

[24] K. Dressler, *Sinusoidal extraction using an efficient implementation of a multi-resolution FFT*, in Proc. Int. Conf. Digital Audio Effects (DAFx-06), pp. 247–252, (2006)

[25] K. Dressler, S. Streich, *Tuning frequency estimation using circular statistics*, in Proc. 8th Int. Conf. Music Information Retrieval (ISMIR), (2007)

[26] Härmä, A., M. Karjalaunen, L. Savioja, V. Välimäki, U. K. Lane, J. Huopainiemi, *Frequency-warped signal processing for audio applications*, Journal of the Audio Engineering Society, 48(11), 19–22, (2000)

[27] T. Fujishima, *Realtime chord recognition of musical sound: a system using common lisp music*, In Proc. International Computer Music Conference (ICMC), pp. 464–467, (1999)

[28] C. A. Harte, M. B. Sandler, *Automatic chord identification using a quantised chromagram*, In Proc. Audio Engineering Society 118th Convention (AES), (2005)

MUSIDO: A Framework for Musical Data Organization to Support Automatic Music Composition

Ryan Demopoulos and Michael Katchabaw
Department of Computer Science
The University of Western Ontario
London, Ontario, Canada

Abstract. It is becoming increasingly prevalent in research towards automatic music composition to make use of musical information extracted and retrieved from existing music compositions. Unfortunately, this can be an unnecessarily complex and tedious process, given the incompatibilities in music modeling, organization, and representation between extraction and composition algorithms. This paper introduces the Musical Data Organization platform and framework (MUSIDO), which is aimed at resolving this problem by providing middleware to facilitate access to musical information in a simple and straightforward fashion. In doing so, MUSIDO provides an effective method to support automatic music composition based on existing music as source data. This paper discusses MUSIDO's design and implementation, and presents our experiences with using MUSIDO to date.

1. Introduction

Automatic music composition is the process of writing music withdrawn from human intervention. Many recent efforts to improve this process focus on learning from music written by humans, which involves extracting musical features and feeding these data directly into composition algorithms. The capabilities of these approaches are strongly dependent on the information gathering mechanisms employed; fortunately, algorithms that can automate this process are becoming increasingly sophisticated [4].

In practice, unfortunately, it is difficult for composition algorithms to take advantage of multiple extraction algorithms due to differences in how these algorithms model musical data. Most composition algorithms require musical data to be carefully formatted for a specific purpose and so researchers typically prefer to format their source data in a fashion that specifically serves the needs of their own approach. Extraction algorithms tend to collect and present information in an ad hoc fashion as well [5], and the information gathered is seldom meant for use in automatic composition systems, leading to issues in interoperability.

This is particularly troublesome since most extraction algorithms tend to focus on one aspect of music for recognition and extraction, and most composition systems typically require multiple sources of information to use for composition. Integration is therefore a significant problem, because it is not just one source to interface with, but rather many. To address this issue, a unified way of communicating musical information between extraction approaches and automatic composition algorithms is needed.

Our current work introduces the Musical Data Organization platform and framework, collectively referred to as MUSIDO. The purpose of MUSIDO is to facilitate the development of automatic music composition systems that rely on existing music as source data. This is accomplished by introducing a middleware entity to transfer musical information between extraction and composition algorithms in an easy, straightforward, and flexible fashion. This middleware is comprised of two important aspects: a clearly-defined data model specifically designed to organize musical data and metadata, and a standard and supportive application programming interface (API) to provide input/output access to a platform conforming to the data model.

While other platforms and frameworks already exist for working with musical data [1,2,7,9,11,12,14,15], these approaches either lack elements useful for automated extraction or composition, or were simply not designed specifically for these tasks. As a result, they fall short in terms of their ability to specify, represent, manipulate, store, and query musical information or in terms of their programming interface for constructing extraction or composition systems. In some cases, no programming interface or ability is provided at all, meaning that a considerable amount of development effort is required to make use of them.

This paper presents the findings of our current work, examining the design and development of MUSIDO in detail and discussing our experiences in using it in developing an automatic music composition system that makes use of music extraction algorithms. We have found that MUSIDO greatly facilitates the development of such systems, and does so in a way that is a considerable improvement over existing work, demonstrating significant potential for the future.

2. Overview of MUSIDO

Traditionally, constructing an automated music composition system that uses existing music as source data encounters a compatibility issue in which the music information extraction and retrieval and algorithms and composition algorithms were not initially designed or intended to work with one another. This is shown in the left side of Figure 1, in which an incompatible Automatic Music Information Retrieval (AMIR) algorithm attempts to interface with an Automatic Music Composition (AMC) algorithm.

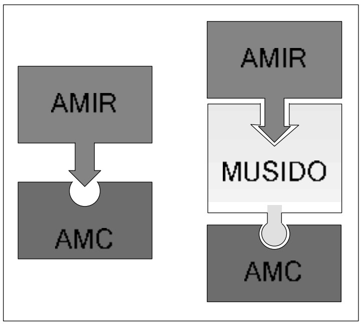


Figure 1: Traditional (left) Versus MUSIDO (right) Approaches

The main goal of MUSIDO is to resolve this issue, by providing a framework to facilitate such algorithms working together, as shown in the right side of Figure 1. This is discussed in the next section in detail.

2.1. The MUSIDO Framework

In designing the MUSIDO framework, we set out to achieve the following:

- Provisioning of a middleware entity, the MUSIDO platform, whose purpose is to communicate musical information between music extraction and music composition systems.
- Reduction of integration effort required through the support of multiple types and formats of data.
- Architectural simplicity, lowering the barrier to adoption.

This led to the development of the MUSIDO framework as depicted in Figure 2, below.

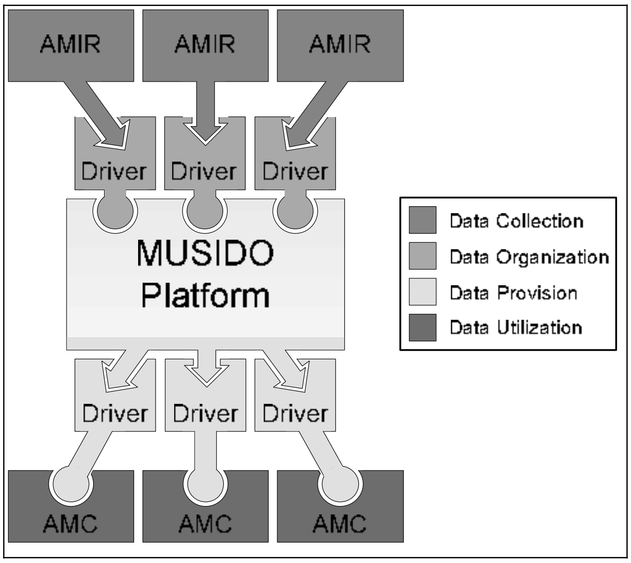


Figure 2: The MUSIDO Framework

The framework shown in Figure 2 is comprised of three types of entities, specifically algorithms, drivers, and a central platform. This contrasts existing frameworks for automatic composition, such as those involving a human component, or those resembling the left side of Figure 1, where composition algorithms make direct use of an extraction algorithm for the collection of musical data. The framework can also be viewed as a workflow with each possible entity participating in one of four separate stages:

- *Data collection*, comprised of one or more extraction and retrieval algorithms, gathering musical data from potentially multiple sources of music.
- *Data organization*, comprised of two entities: input drivers, and the MUSIDO platform, which are collectively responsible for the conversion and management of data.
- *Data provision*, comprised of the MUSIDO platform as well as drivers that serve composition algorithms.
- *Data utilization*, comprised of one or more composition algorithms for generating music.

The input to this workflow is pre-existing musical data; while we consider human compositions as the primary source, our framework does not exclude the use of synthetic data as well. Systems conforming to the framework take these data and apply each of the four stages in turn, eventually outputting newly-composed music. It is important to recognize that our framework does not provide guarantees as to the quality of music produced; this would be impossible, since quality is based on the particular entities participating (most importantly, the composition algorithm) rather than the framework itself. Instead, we strive to support automatic composition systems in order to provide the best opportunity for quality to be evaluated and reliably achieved.

2.1.1. MUSIDO Platform

The central and most significant component of our framework is the MUSIDO platform. Unlike the framework itself which exists as a conceptual entity, the platform has been implemented and exists as usable software. It is comprised of two important aspects: a flexible data model, and a software API. The platform acts as a broker of musical information, providing automated composition algorithms with musical data in a clearly-stated manner. To achieve this level of interoperability, our platform is not designed around any one particular digital music format. Rather, our data model is specifically designed to allow different types of musical formatting to co-exist, with one important caveat: only those musical features pertinent to the majority of compositions systems are directly supported in our work. This helps to achieve a concisely focused API; we leave extensions to this data model up to the individual systems that require them. Thus, improvements can be easily made in rare cases when they are needed; indeed it is not possible for any data model to represent all types of musical data and metadata, as there are a vast number of ways to analyze music itself.

Any composition system that conforms to the MUSIDO framework is one that uses the MUSIDO platform to store and organize passages of music that serve as input to the composition process. Due to its emphasis and complexity, we separately discuss the data model and API of this platform in Sections 2.2 and 2.3 respectively.

2.1.2. Drivers

The MUSIDO platform was introduced to allow music extraction and composition algorithms to work together harmoniously; however, the platform itself is comprised of a single input/output API. Newly designed algorithms will not have difficulty working within our framework, since the API for storing and retrieving data through the platform entity is well known ahead of time. Unfortunately, this is not the case for existing algorithms that have not been designed to cooperate with the platform; thus, the platform alone does not solve the problem of non-interoperability between these entities.

In order for previously created algorithms to communicate with each other through the platform, we need to introduce another entity into our framework, namely *drivers*. The responsibility of a driver is to facilitate the exchange of information between an extraction or composition algorithm and the MUSIDO platform; specifically, drivers translate information between the platform's API and the musical format expected by these algorithms. In simple terms, drivers extend existing algorithms so that they are compatible with the input/output API of our platform; in cases where these algorithms have been explicitly designed to work with MUSIDO, drivers may be unnecessary—in reality, they have already been built into the algorithm itself.

The very requirement for driver entities raises an important question: how does the MUSIDO framework differ from a traditional framework, considering that drivers could be written to allow direct communication between a music extraction process and a composition process? It is true that current composition systems which make use of existing musical data already make use of some form of driver; however, these drivers are almost always integrated directly into the composition algorithm rather than existing as a separate entity, thus demonstrating the improved modularity of our approach. More importantly, even if these drivers were extracted into their own module, the MUSIDO framework would have two distinct advantages. The first advantage of our framework is that its platform is capable of converting many types of musical data automatically, whereas extraction and composition algorithms traditionally are not. As a result, the process of creating a driver to sit between two of these algorithms would often be significantly more complex, since the drivers themselves would require complex conversion logic.

A second and even more important advantage that our framework provides is reusability of algorithms. While it may not be unreasonable for a single driver to be written within a traditional framework, such a driver would only be useful to the specific extraction/composition algorithm combination that it bridges. A problem would occur, say, if the composition algorithm designers wished to make use of a different or new extraction algorithm for gathering data. In such a case, an entirely new driver (with a different set of potentially complex conversion logic) would be needed. Thus, combining the MUSIDO platform with supporting drivers allows one algorithm/driver combination to serve a potentially limitless set of other algorithms interacting with the platform.

Mathematically, suppose we have m extraction algorithms and n composition algorithms. To interface each extraction algorithm with each composition algorithm directly would require $m*n$ points of integration. To interface each extraction algorithm with each composition algorithm using MUSIDO, however, would only require $m+n$ points of integration, to produce the driver for each algorithm in question to interface with the MUSIDO platform. Furthermore, in this scenario, if a new extraction algorithm was created, n points of integration would be required to directly interface it with each composition algorithm, and if a new composition algorithm was created, m points of integration would be required to directly interface it with each extraction algorithm. Using MUSDIO, however, only one integration step would be required with each new algorithm, to develop a driver to enable interactions with MUSIDO.

2.2. Platform Data Model

Prior to developing a software implementation of the MUSIDO platform, our work focused on the design of a data model that would satisfy the key goals of our middleware, most of which came from our review of existing algorithms as outlined in [3]. Currently we offer support for many features found in common Western civilization music; our model does not *fully* support other musical styles from different cultures, however we have preserved flexibility throughout the design process, and in some cases we have explicitly included facilities that allow alternative music forms to be expressed.

Our data model serves to organize two types of musical information. First, many elements that have some corresponding representation on a musical score (such as notes, phrases, and bars) are modeled in a hierarchical fashion. The second type of data is concerned with descriptive non-score elements (including metadata), which serves to represent musical information in a variety of capacities. In each of these cases, the data model is only meant to encompass those data types that are immediately applicable to automatic music composition, to avoid adding unnecessary complexity to the platform when compared to improved utility.

For brevity, the subsections below only provide an overview of various elements from our data model. For further details on the entire data model, the reader is urged to consult [3].

2.2.1 High-level Structure

The highest abstraction of our data model is concerned with how musical information collected by musical information extraction algorithms can be grouped and stored as individual entities, shown in Figure 3. Broadly speaking, all musical information within our platform is stored within a Repository. Repositories are analogous to database instantiations such that each Repository is capable of storing any number of Record objects; the purpose of this is to allow a way to group common Records into one conceptual set.

Record objects exist to track a single passage of music; usually this will be a song. Each can be described using the various types of data in our model, though very few data are required and in many cases fields can be left unspecified. Furthermore, Records stored in the same Repository may contain different degrees of information; there is no requirement that these Records must be similar in structure, allowing Repositories to be very flexible in organization. For example, suppose that a Mozart Repository is created, and two extraction algorithms are used to store Records in the Repository. One of these algorithms may be responsible for identifying chord progressions that occur within a set of pieces, while another algorithm may be responsible for creating a Record to supply statistics on the distribution of Mozart's use of pitch intervals. These data can be stored in separate Records, all within the same Repository, even though each Record is responsible for a different type of data than the others. In this case, Records would not be representing songs, but rather aspects of music.

In general, Records are meant to contain only those data needed for any particular application. Each of the possible data types that can be stored directly into a Record object is explained in the subsections that follow:

- A list of Sections that occur within the Record, such as verses and choruses.
- A list of musical Parts.
- A set of directives, which provide instructions on how the passage should be performed.
- A set of properties (or details), such as a title for the record or the name of a composer.

Records are flexible for storing musical information, being able to contain data from merely a few musical statistics on a piece, to entire pieces themselves. Ultimately, extraction algorithms decide what musical information should be extracted from existing music and stored. In some cases, these algorithms may extract and store important musical features along with the entire piece from which those features are derived, all within the same Record. This is possible due to the flagging system that our data model provides where themes, repeating sequences, melodic lines, and other passage types can be identified as a subset within a Record.

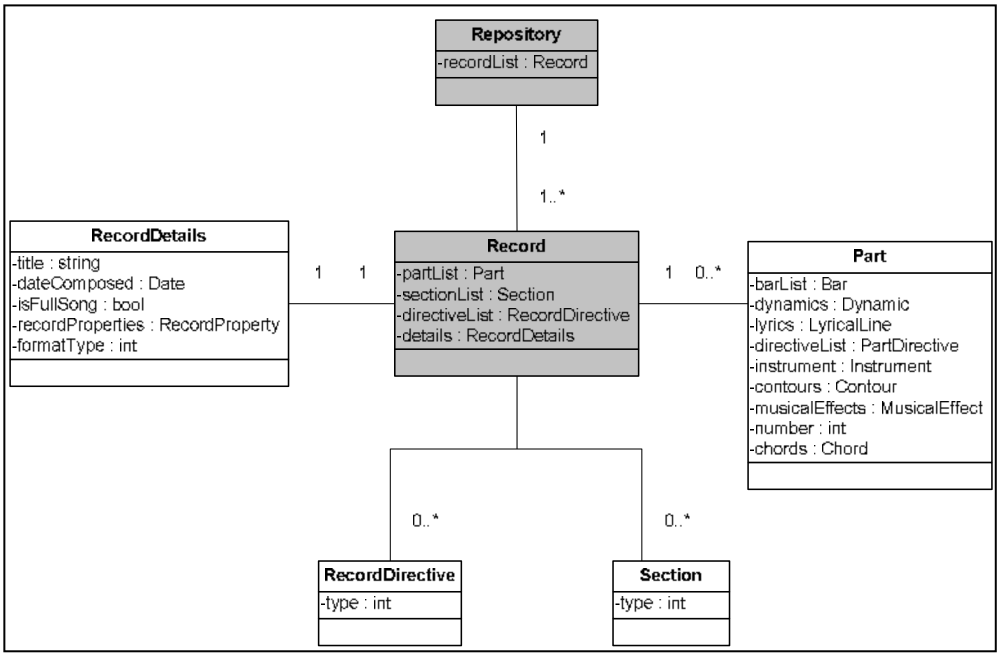


Figure 3: High-level Structure of the MUSIDO Data Model

2.2.2. Sections

We define musical structure as a set of abstract conceptual divisions within a piece, using a collection of Section elements in our data model. Each Section is a segment of music bound by a starting bar and beat and an ending bar and beat. This allows several different types of Sections to be specified for a single passage of music, with each potentially overlapping as well. As an example, a Record that represents a folk song may have one Chorus Section and five Verse Sections. This is useful for automatic composition systems to apply analysis on different parts of a piece, or to determine how the structure of newly-formed music should occur if attempting to mimic existing work.

Our model defines five types of Sections by default: Chorus, Verse, Bridge, Climax, and Alternate Ending. The model, as mentioned earlier, is highly extensible, allowing other Section types to be defined as necessary.

2.2.3. Parts

One of the most important aspects of music is the actual notes that are played. In our platform, musical Notes and Bars are assembled into Parts. This follows common conventions: most prominent digital music formats organize data this way as well, including GUIDO [8], MuseData [9], MusicXML [6], and Lilypond [13], since printed musical score is organized in this manner. Platform Records contain a set of Parts, each of which contains the following: a unique integer identifier number, an Instrument, a set of directives, a set of Musical Effects, a set of Lyrics, a set of Dynamics, a set of Bars (containing Notes in Note Positions), a set of Contours, and a set of Chords (serving as a Chord progression).

Some of these associations may be somewhat obvious in design. It is clear that Bars should be associated with Parts, according to the common layout of music. The integer assigned to a Part serves as a unique ID value within a Record; this allows data associated with the part to be retrieved directly, or through the data projection system supplied by the platform. Also, the association of an Instrument to a Part merely reflects the same common association that exists in real-world music; both written score and digital formats such as MIDI already assign single Instruments to Parts, even though it is possible that two instruments could potentially have notes on the same musical line, as is sometimes done in choral music. In such cases, our platform's 1:1 mapping of Part to Instrument requires that polyphonic multi-voiced parts be separated into separate lines of music. As a practical aide to most existing algorithms that deal with MIDI, our platform implementation includes a subclass of Instrument specifically designed to represent a MIDI instrument, including a field for a MIDI patch number.

In contrast to the more obvious design decisions above, it may be less apparent as to why the remaining elements are associated with Parts, rather than some other entity. For example, many types of musical contours can exist at a finer granularity, such as pitch contours within Bar objects. It would seem advantageous to associate these Contours with Bars rather than with Parts; in the case where a Contour is needed that spans an entire Part, one could be created from the individual Contours of each Bar contained within, although some method of determining Contour symbols between Bars themselves would be needed. While this is true, the MUSIDO platform (and Records in particular) is designed to hold partial information. Consider a music extraction algorithm that operates on MIDI data by extracting a pitch contour from the notes that occur in a melodic line. If our platform required that the contour be broken down on a per-Bar granularity, then some understanding of the Bar divisions of the piece would be required. Unfortunately, MIDI data does not include any notion of Bar lines or divisions; thus, the algorithm would not be able to specify the contour within the platform unless it first applied some additional extraction approach to Bar induction; certainly a considerable problem, and even more so in other cases such as Chord progressions, where the challenge of extracting Chords is already very difficult. Thus, our approach allows the full specification of a Contour within a Part, regardless of the presence of musical Bars.

In some cases it may make sense to track a type of data on a Bar-per-Bar basis, and yet this same type of data may also cross bar line boundaries. For example, many Dynamics can easily be attributed to a specific note within a specific Bar. Despite this, Dynamics may also span across several Bars; for example, a crescendo may occur over a long sequence of notes. To store these data, two solutions exist: track Dynamics within Bars with complex analysis required for those Dynamics that cross bar lines, or simply track Dynamics according to Parts, requiring that a start time and possibly a finish time be set. The latter option maintains simplicity, particularly for developing a platform API, whereas the former requires complex and expensive algorithms to coordinate dynamics participating in several Bars. Thus, we take the latter approach, applying this to Lyrics, Chord progressions, and Musical Effects also.

2.2.4. Directives

The purpose of a directive is to provide general musical instructions for the entities that it is associated with. In our platform, we make use of two types of directives.

The Record Directive type may seem to serve a similar purpose to the Record Details type (discussed in next section); while both provide data that affects or describes the entire Record as a whole, they are different in two regards. First, Record Directives describe musical features while Record Details are responsible for musical metadata. In particular, Record Directives describe how music should *sound*; our current platform includes a core set of directives, while extensions to these may be included by other specific systems. The second difference between these entities is that Record Directives take place at a specific point within a piece, rather than applying to the piece universally. This is reflected in the hierarchy of the class; each directive has a distinct start and end point as defined by the four temporal variables tracking bar and beat timing.

Part Directives closely resemble Record Directives; their purpose is very similar, only limited to the scope of the particular Part that they are associated with. A wider range of directive types exist for Parts; in particular, data concerning how a piece is traversed and what play style should be used are included, again leaving the option for additional directives to be defined by individual systems using our framework.

2.2.5. Record Details

Aside from musical features, Records also contain metadata describing this information. There is potentially a large amount of data that could exist on a musical piece or passage; this was learned early in the data modeling process of the Record class, where the number of variables and functions involved quickly became unreasonable to track in a concise manner. Thus, we have chosen to separate all metadata described for musical Records in a separate object called Record Details. Record Details can include a wide variety of attributes, including title, composer, composition date, genre, status, quality ranking, data format, and so on.

There are two purposes for storing Record metadata in general. First, these data may be useful in describing some aspect of music to be considered by a composition system when generating new compositions; for example, a composition system may use the title of an existing piece to give some indication of what a newly-composed piece should be named. The number of applications in this regard are probably few, but still potentially useful. The second and likely more common way to use these metadata is for the purpose of Record organization. For example, a composition system using a large Repository of Records may wish to isolate only Jazz Records, or perhaps only those musical passages written by a particular composer. This descriptive information lends well for categorizing Records into utility-specific groups.

2.3. Application Programming Interface

The API supported by the MUSIDO function consists of three categories of functionality: input, output, and processing. Input and output functions are fairly straightforward: input functions take musical data into the platform, while output functions allow musical data to be retrieved, either at a low-level in a more direct fashion, or at a high-level, in which the data can be organized to make common information-gathering tasks simpler to accomplish.

Processing functions are provided to enable computational offloading. In some cases, the data stored in MUSIDO may require analysis or manipulation that goes beyond mere reorganization. We have therefore included some functions that act as composition services, processing common tasks that many composition systems require based on our observations as discussed earlier in this paper. These include functions for format conversion, contour conversion, data projection, transposition, statistical calculations, and so on.

Further details on this API can be found in [3].

3. Prototype Implementation

The primary goal of the MUSIDO platform is to allow a wide range of music extraction and composition algorithms to communicate information across the platform. In an effort to satisfy this goal, we have chosen to implement this middleware in two different languages: Microsoft's J♯ .NET, and Sun's Java. These languages are designed to operate over a diverse range of heterogeneous operating systems; while Java currently offers greater portability, an increasing number of software developers are making use of the .NET platform, and we felt it was important to provide application interoperability in this environment as well. Our selection of J♯, rather than another .NET language, was motivated by the potential for simple code migration. J♯ uses a syntax identical to that of Java, and nearly all of J♯'s API is a subset of Java's; precisely, J♯ is equivalent to J2SE v1.1.4. As a result, our code was written in J♯ knowing that migration to Java would be trivial. Also, due to the .NET framework's internal language compatibility, our platform can be compiled as a .NET dynamic link library (DLL) and used in other more common .NET languages, such as C♯, C++, and Visual Basic.

Unfortunately, the J2SE v1.1.4 API, and thus J♯'s API, is not sufficient to allow many automatic composition systems to be written in J♯. The source of this problem is a lack of support for MIDI data; specifically, the *javax.sound.midi* package was not added to J2SE until revision 1.2, and so this package is not available when using J♯. As previously mentioned, compositions systems frequently use MIDI data to specify newly-written music, and so the absence of complex MIDI-creation logic serves as a major deterrent to the use of J♯ for a system conforming to the MUSIDO framework. Currently, the best way to avoid this problem is to refrain from using J♯ as a potential language for developing a music composition or extraction system. The popular language C♯ does include MIDI data support, and the MUSIDO platform can still be used in this language when imported as a DLL package.

For developers desiring to use J♯ as a basis for their systems, an alternative does exist; in fact, the AMEE™ composition system [10], with which most of our validation was done, is written in J♯. AMEE™ uses a customized J♯ version of the MIDI package from Java. This package was originally taken from the GNU Classpath project, and was modified to work in the .NET environment by the developers of AMEE™. This solution not only allows some composition systems to work with MIDI in J♯, but it also preserves the code migration benefits of working with Java/J♯ syntax. Thus, while this customized package is not part of our platform's design, we consider it an important extension for systems using MUSIDO in J♯.

4. Experiences

To validate MUSIDO's use in supporting automatic music composition, we conducted a variety of tests and experiments. This section provides a brief overview of our experiences; the reader is urged to consult [3] for additional details as necessary.

Initial testing involved ensuring that MUSIDO could effectively communicate musical information between music extraction and composition algorithms. Through the use of simple algorithms, drivers were constructed to interface them with MUSIDO, and they were able to use MUSIDO easily and efficiently through its API to store, organize, and retrieve various musical elements.

More substantial evaluation of MUSIDO came in the form of testing with the automatic music composition system known as the Algorithmic Music Evolution Engine (AMEE™), as discussed earlier. Four simple music extraction algorithms were developed to gather themes, chords, probabilistic values, and contours from existing pieces of MIDI music. This data was fed into MUSIDO, and accessed by AMEE™ through the use of appropriate MUSIDO driver modules.

In doing this integration, we found that AMEE™ was able to produce music quite effectively using the musical information retrieved from MUSIDO, and was able to produce music using MUSIDO that was identical to a direct integration with the aforementioned extraction algorithms. The music composed in both cases was identical, both in the time, onset, and rhythmic feature domains. This helps to confirm the correctness of the MUSIDO platform itself; no loss of information on pitch, rhythm, or onset time is experienced when the platform is used. In the process, we did find, however, that MUSIDO greatly facilitated integration efforts, allowing AMEE™ to access musical information far more easily than with a traditional, direct integration with the algorithms.

In the end, our initial experiences indicated that MUSIDO both allows for an efficient and effective exchange of musical information between music extraction and composition systems, and enables this exchange far more readily than direct integration. This demonstrates that MUSIDO is quite promising for supporting automatic music composition efforts in the future.

5. Conclusions and Future Work

Using existing music as source data in automatic music composition systems is an approach attracting more attention from the research community. Our current work in MUSIDO is aimed at supporting this work by providing a framework to allow music extraction and music compositions to exchange musical information to support composition processes. Early experiences with MUSIDO have been quite successful, demonstrating great promise for continued work in the future.

There are several potential research directions to be explored in the future. We would like to continue experimentation with MUSIDO, in particular through user evaluation of MUSIDO and its results. Support for additional music formats and representation could be added to MUSIDO, allowing the platform to work with music in GUIDO notation, MusicXML, and sampled formats. We would also like to study the extension of MUSIDO to include musical concepts that are currently unsupported, including musical elements from other cultures.

References

[1] C. Agon, G. Assayag, M. Laurson, and C. Rueda. Computer Assisted Composition at Ircam: PatchWork & OpenMusic. *Computer Music Journal*, 23(5), 1999.

[2] G. Assayag, M. Castellengo, and C. Malherbe. Functional Integration of Complex Instrumental Sounds in Music Writing. *In Proceedings of the 1998 International Computer Music Conference*, San Francisco: International Computer Music Association, 1985.

[3] R. Demopoulos. Towards an Integrated Automatic Music Composition Framework. *Masters Thesis*, Department of Computer Science, The University of Western Ontario, London, Canada, May 2007.

[4] R. Demopoulos and M. Katchabaw. Music Information Retrieval: A Survey of Issues and Approaches. *Technical Report #677,* Department of Computer Science, The University of Western Ontario, London, Canada, January 2007.

[5] S. Doraisamy and S. Rüger. A Comparative and Fault-tolerance Study of the Use of *n*-grams with Polyphonic Music. *In 3rd International Symposium on Music Information Retrieval*, Paris, France, October 2002.

[6] M. Good. MusicXML for Notation and Analysis. *In The Virtual Score: Representation, Retrieval, Restoration*, W. Hewlett and E. Selfridge-Field, eds., MIT Press, Cambridge, MA, 2001.

[7] M. Henz, S. Lauer, and D. Zimmermann. COMPOzE: Intention-based Music Composition through Constraint Programming. *In Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, Toulouse, France, November 1996.

[8] H. Hoos, K. Hamel, K. Renz, and J. Kilian. The GUIDO Notation Format – A Novel Approach for Adequately Representing Score-Level Music. *In Proceedings of the International Computer Music Conference*, 1998.

[9] W. Hewlett. MuseData: Multipurpose Representation. In *Beyond Midi: the Handbook of Musical Codes*, MIT Press, Cambridge, 1997.

[10] M. Hoeberechts, R. Demopoulos, and M. Katchabaw. A Flexible Music Composition Engine. *Proceedings of Audio Mostly 2007: The Second Conference on Interaction with Sound.* Ilmenau, Germany, September 2007.

[11] D. Huron. The Humdrum Toolkit: Reference Manual. *Center for Computer Assisted Research in the Humanities*, Menlo Park, California, ISBN 0-936943-10-6, 1995.

[12] M. Laurson and J. Duthen. PatchWork, a Graphical Language in PreForm. *In Proceedings of the International Computer Music Conference*, San Francisco, CA, 1989.

[13] H. Nienhuys and J. Nieuwenhuizen. Lilypond, a System for Automated Music Engraving. *In XIV Colloquium on Musical Informatics*, Firenze, Italy, May 2003.

[14] D. Psenicka. FOMUS: A Computer Music Notation Tool, *FOMUS Project Documentation*. February, 2007.

[15] A. Sorensen and A. Brown. Introducing jMusic. *In InterFACES: Proceedings of The Australasian Computer Music Conference*. Brisbane, Australia. 2000.

PublicDJ - Music selection in public spaces as multiplayer game

Stefan Leitich, Markus Toth
University of Vienna
Institute for Distributed and Multimedia Systems
Liebiggasse 4/3-4, 1010 Vienna, Austria
stefan.leitich@univie.ac.at, a0025690@univie.ac.at

Abstract. Music is an important tool to achieve a certain atmosphere for social gatherings in public spaces (e.g., Cafés, Pubs, Clubs) . The selection of proper music tracks is often either done by a staff member of such a location or a professional music selector – a disc-jockey (DJ).

The mobility of personal media libraries is increasing. A portable music player has become an everyday item and people are used to have their favorite music with them at any time. Selecting the preferred music for their own makes them at least as professional for themselves, as a "true" professional music selector performing in public. In a usual setting a single person determines what a whole audience is listening to, degrading the listeners to passive consumers. Creating the possibility to allow every person in the auditory getting involved in the music selection process, by using their portable music library, would result in a totally new kind of interactive listening experience in public spaces.

We present our concept for music selection in public spaces as a multiplayer game – PublicDJ – and its prototypical implementation. The concept is based upon a round based multiplayer game, where each player can submit music tracks to a server. The server analyses submitted tracks and selects the best matching track, based on a former announced criteria for playback. Selection criterias can range from high-level manually annotated audio metadata to low-level audio metadata extracted from audio using music information retrieval techniques. This allows the implementation of tasks like "Submit songs of the same genre!" or "Submit songs of the same artist!", which the users have to fulfill and can be used as a steering instrument for the music played.

Our prototype for collaborative music selection in public spaces as a multiplayer game increases interaction and involvement of listeners by providing the possibility of active participation in a previously completely passive experienced procedure.

1 Introduction

"Last night a DJ saved my life" is a famous chorus of a song released in the early 80's, describing the situation of a boring night out, until the disc-jockey (DJ) played a special tune. Thereby, she changed the mood of the people in the club and animated the audience. Maybe not the most profound example, but a good one describing the power of music in influencing the emotional state of human beings – as told by the song's lyrics, as well as by the song itself in clubs all over the world every saturday night. Furthermore, it displays the concentration of this power in the decision of a single person – the DJ – for the whole audience. The desire to be a DJ with this "magic energy-giving power" themselfes may exist in many persons. It's not everyone's favour to spend a lot of money to collect music, stand up in front of an audience, and select or mix music. Nevertheless, everyone who is into music is a music expert in her preferred music domains. Listeners often have a certain idea which song to listen to in which emotional state, through having heard her most liked music over and over again. People are their personal DJ's, as soon as they select the music they want to listen to intentionally, either in company or on their own.

Personal music libraries gain in mobility through new generations of multimedia enabled playback devices, creating the possibility for everyone to carry around her very own music library or at least excerpts of it. These devices (e.g., MP3 players, mobile phones, etc.) become more and more wireless network enabled and often provide a runtime environment for custom applications. Those technologies allow the execution of applications with low computational load and the exchange of a certain amount of data within reasonable time.

These are the basic conditions to create the possiblity for everyone to play an active role in the music selection process in a public space, while bluring the boundary between the domain expert (i.e., DJ) and the consumers (i.e., audience).

In the reminder of this paper related work is presented (cf. 2) and the concept of PublicDJ (cf. 3) is introduced with the game principle (cf. 3.1) and song selection criterias (cf. 3.3) implemented so far. A short

MPEG-1 Audio Layer 3

description of the prototype follows (cf. 4, and the paper finishes with intended future work (cf. 5) and conclusions (cf. 6).

2 Related Work

Approaches in the domain of automatic playlist generation are manifold, but mainly aimed on satisfying a single listener's needs by automatically choosing the right sequence of songs. Those approaches share in common the description of songs by low- and/or high-level metadata, which is furthermore used by various algorithms (e.g., graph algorithms, recommender algorithms, etc.) to figure out the right sequence of songs, fitting certain criterias.

The pure audio data, necessary for playback, is analyzed to extract content based features (e.g., [7]). Another approach to describe the audio is manually annotated or automatically generated high-level metadata [2]. Automatically annotated metadata often concerns tracking listening habits [1], observing skipping behavior [11] or simple feedback loops [8]. These techniques are of scientific nature and already being used in real market products (e.g., `last.fm`).

Publications concering music listening as social experience are, for example, the *MUSICtable* [12] introduced by Stavness et al. They use a landscape metaphor to visualize a manually arranged music collection on a table-mounted display, accessible from all sides. Buttons arranged around the display can be pressed by the users involved, to affect the “wind” on the virtual landscape and move the cursor position in a direction, thus controlling the next song to be played. A democratic approach for music choice in public spaces is proposed by O'Hara et al. with the *Jukola* system [10]. The system provides a voting mechanism through various input devices (PDA, Touch-Screen) to enable visitors of a public place (e.g., a pub) to select music democratically, while still allowing control of the music pool selected from. The paper observes the social impact of such a setting and underlines the interest of people taking part in the music selection process.

The following publications possess a definitive aim at a club scenario. These approaches tackle music selection by the audience, or even try to create and manipulate music, through monitoring feedback by multimodal sensors [3, 4, 13, 6].

3 Concept

PublicDJ 's concept is to allow every interested visitor of a public space, to bring her own music collection along on a network enabled device, and take part in the selection process of music that is going to be played at this place.

The DJ is responsible for selecting and mixing songs in order to create a certain kind of atmosphere in a public space. Making this decision process a complete democratic one without any kind of control may lead to inadequate abrupt changes in music style and therefore circumventing the creation of an atmosphere at all. The music style could also be developing into a direction undesired for a certain place or time [5]. These requirements have to be considered for the successful design of a system, that strives for the goal of involving diverse people, while simultaneausly creating a certain kind of atmosphere.

3.1 Game Principle

The game principle of PublicDJ is a round based multiplayer game. With the start of each new round users are requested to submit songs to a server that they want to hear, and in their opinion fit a given criteria. The length of a round is determined by the current song playing. After analyzing the received submissions the server determines the best matching song, with respect to the given criteria and the actual song played. The best matching song will be started as soon as a new round is initiated.

Basically, the criterias that have to be fulfilled by submissons are extendable in any way a feature extraction technology exists. The boundaries for low-level features are the computational load for extraction, which has to be performed within a round's duration (i.e., time left of the actual song played, after transmission to the server). For high-level features the requirement is their presence, which is often not given.

3.2 Game Round

As depicted in Fig. 1 a round can be divided into phases. Songs are submitted and analyzed afterwards. The extracted feature attributes are normalized, distances are calculated, and the respective best matching song is selected. Those phases are not visible to users. They can start submitting their selection anytime during the round. If the submission can not be finished during the round, it is canceled and the user is suggested to re-submit her selected song in the next round. Same is valid for the analysis. If a song can not be analyzed totally till the end of the round, the analysis is canceled and the submission is not considered in the selection process. This explains the demand for

`http://last.fm` - A social music platform, recommending music for individual users by tracking it's whole community listening behavior and relevance feedback to suggested songs.

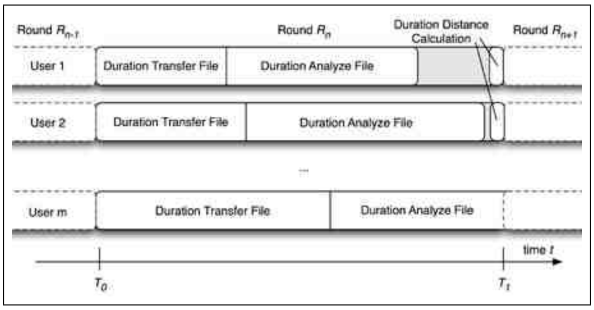


Figure 1: Game round R_n with submitted songs by user 1 to m runnning from T_0 till T_1. Submission m would be ignored, as the analysis is not finished before round end.

estimated time amount calculations on the server for each and every transmission and analysis.

3.3 Song selection criterias

The song selection criterias we tackled for this proof-of-concept, are (1) a simple enqueue mechanism, (2) low-level-, and (3) high-level metadata criterias. In case of the simple enqueue mode (1) all songs submitted by clients are enqueued at the server for playback without any processing. The extracted low-level metadata (2) allows an estimation of perceived similarity between songs. Similiarity in context of the features used can be defined as timbre and rhythm related. High-level metadata (3) allows the comparison of songs regarding, for example, artist, year of creation, or even artwork, if the corresponding feature extraction is able to process images.

Low-level metadata For low-level metadata based selection criterias we used features described in [7]. Those features are established and well performing descriptors for song similarity, as evaluated in the Music Information Retrieval (MIR) community in recent years.

Rhythm Patterns are a time invariant descriptor, which contains the amplitude of modulation for the 24 critical frequency bands, with respect to modulation frequencies. Basically it is extracted by transforming the audio signal with a Fast Fourier Transformation (FFT) into the frequency domain and process the result in psychoacoustically motivated stages (e.g., frequency grouping, loudness leveling, loudness sensation per band). Finally applying a second FFT stage results in the time invariant modulation amplitude information.

The *Statistical Spectrum Descriptor* is a feature vector consisting of seven statistical moments (mean, median, variance, skewness, kurtosis, min- and max-value) for the 24 critical bands. The descriptor captures the frequency characteristics in terms of statistical information about the audio signal, after transforming it into the frequency domain by a FFT and applying frequency grouping into the Bark scale, resulting in a feature vector of 168 elements.

The *Rhythm Histogram* features gather information about the distribution of magnitudes of modulation frequencies in 60 bins summed up for all 24 critical bands. Modulation frequencies are captured in a range from 0 to 10 Hz, segments processed are of 6 sec. duration. This algorithm produces a feature vector with 60 elements, describing “rhythmic energy” per modulation frequency bin.

High-level metadata As we assumed MP3 as the most common audio format for personal music libraries, a metadata extractor for ID3 tags [9] was implemented. ID3 tags are embedded in the MP3 file format, and, if present, can contain annotated song metadata, for example artist name, track title, year of creation, or artwork in terms of an image. The main issue with high-level metadata is, that there is no guarantee in a real world setting for the presence of the information required for song comparison. In case of missing information the metadata extractor has to consider the submitted song as dissimilar.

4 Prototype

The implementation of PublicDJ was done in Java, because it is a platform supported not only by personal computers, but by mobile devices (e.g., mobile phones, PDAs) as well. Communication between the components of PublicDJ is realized using Java’s Remote Method Invocation (RMI) interface. The client implementation conforms to IBMs J9 Java Virtual Machine (JVM) to run on a variety of PDAs.

4.1 Architecture

The main component of the system is a central *server*, responsible for controlling the game logic, audio playback, estimating remaining time amounts for transfers and analyses, storing received analysis values, rudimentary display of game stats, and communication between the different parts of the systems. Login and connection status of clients are monitored by the server. Incoming song submissions are received and delegated, in respect to load balancing, to the *analysis server*’s connected. Externalising the feature extraction in arbitrarily instantiations of *analysis servers* ensures scalability. Those extract, in dependency of selected criterias, the regarding metadata. The *administration* and *client* applications are basically the same and allow login and song submission to the system. The *administration* application furthermore allows to adjust server settings remotely regarding the condition, that song submissions have to try to fulfill to be selected as the best matching one for the succeeding round.

5 Future Work

We understand this first prototypical implementation of PublicDJ as a proof-of-concept. The most important aspects we want to implement in the next development steps are the following tasks. (1) Adapting the client interface to an applet or web application, to remove the barrier of requiring the setup of a software component on the client devices. (2) Extend the server with automatic playlist generation capabilities, thereby turning it into an competitor comparable with human users. Depending on this playlist generation algorithm, the server can also gain the possibility to affect the development of the sequence songs are played over time, and provide a certain amount of control to avoid possibly undesired music styles. (3) Finally, we would like to perform a field test of PublicDJ and observe users interacting and using the system, to conclude further improvements and ideas.

6 Conclusion

In this paper we introduced our collaborative music selecting approach – PublicDJ – for public spaces. It allows to submit songs from network enabled audio playback devices for automated relevance evaluated music selection. By making use of various ways of audio metadata extraction, best matching songs for given criterias are determined and selected for playback. The application depicts an interactive system, turning passive listeners into active music selecting protagonists in public spaces.

References

[1] Andreja Andric and Goffredo Haus. Automatic playlist generation based on tracking user's listening habits. *Multimedia Tools Appl.*, 29(2):127–151, 2006.

[2] J.J. Aucouturier and F. Pachet. Scaling up music playlist generation. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*. Lausanne, Switzerland, August 2002.

[3] Dave Cliff. hpdj: An automated dj with floorshow feedback. Technical Report HPL-2005-88, Digital Media Systems Laboratories, HP Laboratories, Bristol, UK, May 2005.

[4] Mark Feldmeier and Joseph A. Paradiso. An interactive music environment for large groups with giveaway wireless motion sensors. *Computer Music Journal*, 31(1):50–67, April 2007.

[5] Carrie Gates, Sriram Subramanian, and Carl Gutwin. Djs' perspectives on interaction and awareness in nightclubs. In *DIS '06: Proceedings of the 6th ACM conference on Designing Interactive systems*, pages 70–79, New York, NY, USA, 2006. ACM Press.

[6] Dennis Hromin, Michael Chladil, Natalie Vanatta, Susanne Wetzel, Farooq Anjum, and Ravi Jain. Codeblue: a bluetooth interactive dance club system. In *Proceedings of the IEEE Global Telecommunications Conference GLOBECOM*, pages 2814–2818, December 2003.

[7] Thomas Lidy and Andreas Rauber. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In *ISMIR 2005, 6th International Conference on Music Information*, pages 34–41, 2005.

[8] Beth Logan. Content-based playlist generation: Exploratory experiments. In *ISMIR 2002, 3rd International Conference on Music Information*, 2002.

[9] M. Nilsson. Id3 tag version 2.3.0. http://id3.org/d3v2.3.0, February 1999.

[10] Kenton O'Hara, Matthew Lipson, Marcel Jansen, Axel Unger, Huw Jeffries, and Peter Macer. Jukola: democratic music choice in a public space. In *DIS '04: Proceedings of the 2004 conference on Designing interactive systems*, pages 145–154, New York, NY, USA, 2004. ACM Press.

[11] Elias Pampalk, Tim Pohle, and Gerhard Widmer. Dynamic playlist generation based on skipping behavior. In *ISMIR 2005, 6th International Conference on Music Information*, pages 634–637, 2005.

[12] Ian Stavness, Jennifer Gluck, Leah Vilhan, and Sidney Fels. The musictable: A map-based ubiquitous system for social interaction with a digital music collection. In *Entertainment Computing - ICEC 2005, 4th International Conference, Sanda, Japan, September 19-21, 2005, Proceedings*, volume 3711 of *Lecture Notes in Computer Science*, pages 291–302. Springer, 2005.

[13] Ryan Ulyate and David Bianciardi. The interactive dance club: Avoiding chaos in a multiparticipant environment. *Computer Music Journal*, 26(3):40–49, March 2002.

Synthetic game audio with Puredata

Andy Farnell (andy@obiwannabe.co.uk)

August 24th, 2007

Abstract. This document looks generally at the application of synthetic audio to video games and presents some experimental results and analysis from research in generating efficient synthetic sound for virtual worlds. Dynamic sound effects for real-time use in game worlds require far more careful thought to design than brute force methods employed where sound is computed prior to delivery, not just in terms of static computational cost, but in how they are structured and behave when parametrised by world events. To make synthetic audio a possibility for the next generation of game audio compromises must be found that satisfy several conflicting variables. Examples are presented that demonstrate cost effective approximations based on key physical parameters and psychoacoustic cues, what can be called ”Practical synthetic sound design”.

1 Why the move towards synthetic sound?

Some may argue that synthetic audio has always been a solution looking for a problem, that synthesisers were just fashionable musical toys in the 1980s. Now a real problem has arrived. That problem is how to provide the colossal amounts of content required to populate virtual worlds for modern video games. While graphics technology has received an enormous investment of money and attention sound, despite claims in defence of the games industry [2], is still in the dark ages in terms of technology. The reasons for a reluctance to embrace progressive technologies deserves an entire study by itself, which I will address elsewhere.

1.1 Generating content

Anecdotally, it seems as of Autumn 2007 many directors and project leaders in game development are realising it is challenging to generate sufficient content to populate worlds even with a full-time team of sound designers. The size of games worlds and the number of objects will continue growing. One solution for massively multi-player worlds such as ”Second Life” is to rely on user generated content. Disadvantages to this, are unreliability, inconsistency and lack of appropriate control. In any case it is desirable to have default sounds for most classes that are overridden by user content. Massively single-player models such as ”Spore” rely on procedurally generated content. There are important differences between procedural content and synthetic content which are subtle, and will be discussed later, but essentially synthetic audio is an extension of the ”massively single-player” approach applied to the general case of virtual world sounds.

Growth The mapping between objects and their visual manifestation is one to one. Although many instances of a particular object may be created, for each instance there is one mesh and one set of textures that defines its appearance. Regardless of the distance from which it is viewed or the lighting conditions the same data is used to represent an object. By contrast, sound is an interactive phenomenon, so there is not a fixed number of sounds belonging to one object. If you like, the number of ”object views” is greater than for visual appearance.

Relationships If we take the entity relationship ”collides” to be unique per object pair then the growth is factorial. Of course this is overkill, for the majority of relationships such as `hammer->hits->bell` or `anvil->hits->bell` are indistinguishable. But the number of realistic couplings that lead to distinguishable excitation methods and consequent sound patterns is still large. An object may by struck in a glancing fashion or impacted by a body that remains in contact, it can be scraped to cause frictional excitation, blown to excite an air cavity or shaken by a nearby coupled source into resonance. These things can happen in air or underwater affecting the speed of sound in the medium, or while the object, observer or propagation medium is in motion causing Doppler effects. Additionally, any sound object may be parametrised by a set of continuous physical variables far more extensive than the small number of vectors defining appearance.

Combinations Consider for example the case of footsteps in traditional game sound design. Conservatively we might have 20 surface textures such as wood, metal, gravel, sand, concrete and so forth. We must multiply this by the number of characters that have unique footwear, perhaps 3 or 4, and again by several distinct speeds of movement since a footstep sound is different when the player creeps, walks or runs. We might also like to factor in the weight of the player according to personal inventory, and maybe adapt the footsteps to reflect the work done ascending or walking on a level surface. Taken as discrete states this matrix of possible sounds quickly grows to the point where it is difficult to satisfy using sample data. Difficulties arise with consistency of level and quality as the data set grows. We get to the point where processing is better deferred until performance (play) where continuous rather than discrete control is possible.

Access A further problem arises with the length of play and arrangement of the audio data pool. Some adventure games may extend beyond 40 hours of play. "Heavenly Sword", a recent PS3 title boasts over 10GB of audio data. 10GB of data requires careful memory management, placing data at reasonable load points and forcing the flow of play to be over-structured. This moves us towards a linear narrative more like a film than a truly random-access "anything can happen" virtual world. From many points of view we are approaching a situation where it may not be possible to continue extending game audio content within the data model, even with advanced storage like Blu-Ray or HDVD. The storage requirements are dwarfed by data management requirements.

1.2 Parametrising content

Some variations of sound, traditionally handled by blending loops of given states, are hard to recreate with the data model. Composite objects such as an vehicle engine or other machines may have many sounds that are ascribed to the same source depending on continuously variable parameters rather than events or states. The sounds of an engine starting, accelerating, stalling or misfiring are all different, yet come from the same source. The sound of an engine close to the cylinder block, inside the vehicle, at the rear close to the exhaust, or the sound of a distant vehicle speeding away are all different manifestations of the same object. To maintain a continuous relationship between all these positions and states while keeping the components synchronous for blending is difficult. Techniques developed for manipulating sample loops [10] are very poor in comparison to synthetic methods and since this has been a recurring complaint from players and developers alike we are keen to encourage developers to look at embedding code for vehicle engine synthesis.

Dynamic and varied content "Dynamic" in this case does not refer to the range of loudness but to the ability to reconfigure the DSP during performance, for example to dynamically alter a filter as an object moves around. Sample data intrinsically lacks dynamics, it is a fixed recording of a sound. Research has been conducted and articles written [15] on avoiding too much similarity when using sample data. Many of these amount to introducing random playback points, replay speeds and filter settings. These can really only be seen as temporary remedies or tricks. A major strength of synthetic sound is introducing variations that mean no two sounds from the same event need ever be exactly the same. This is an inbuilt feature of subtractive methods that use noise as the basis waveforms. No two examples of the same impact using a noisy excitation and material formant approach will ever have exactly the same time domain waveform. Although the differences may be extremely subtle, or even immeasurable by spectral analysis, the brain has a peculiar ability to recognise that these sources have some "live" quality compared to exactly the same waveform replayed over and over using sample data. Although I have no hard experimental evidence to support the claim it seems they are less fatiguing. Sometimes it is desirable to introduce subtle variation into sound events that are very similar but distinguished by a continuous variable. An example is a dropped ball which sounds very slightly different on each bounce as it loses kinetic energy to entropy. Some objects like bells and tubes create different sounds depending on their state. A tube that is already vibrating and is then struck in a different location incorporates the new excitation into its vibration pattern to make a different sound than if it were struck while at rest. This is mentioned later where we consider the advantages of waveguide models, though I would like to note here that it is not possible with sample data because superposition cannot reproduce it.

2 Synthesis Methods

Traditional approaches to synthesis tend to focus on one method and attempt to use it universally. Here I am concerned with efficient results rather than an attempt to explore the theoretic limits of any particular method. Theoretically the range of timbres obtainable from each method is equivalent, but in reality there are boundary conditions that limit each to a practical range. For example, additive and subtractive methods are complementary, each containing a special case that is extremely expensive in one method while trivial in the other. The practical approach is to use DSP built from hybrid methods, to use all the tools in the box. That means not being limited to spectral, physical, or time domain waveform modelling, but rather to use each as it is appropriate. Physical and psychoacoustic interpretations are also very important. They represent the end points in a synthetic facsimile, the cause or production mechanism is rooted in physics, while the effect is psychological and perceptual. Many solutions in practical synthesis involve some kind of "trick" or efficiency. Some of these are undocumented, but the majority are taken and used creatively from literature on psychoacoustics [12].

2.1 Synthesis Implementation

Our synthesis methods are developed in Puredata. While dataflow systems cannot express all idioms required to make any kind of DSP, for example it lacks recursion and effective sample accuracy to implement IIR filters, it is extremely productive, easy to to work with and covers 90 percent of requirements for synthetic sound design. Because games developers have never adopted such a thing as a real "sound engine" (meaning a system capable of executing arbitrary DSP graphs), it is currently necessary to rewrite code using Faust [17], STK [4] or hand craft the prototypes in C/C++. Hopefully tool-chain improvements will soon make it possible to write synthetic sound objects directly in dataflow for execution on game consoles. Nevertheless, current design approaches must look ahead as if such a path were already available, so many programming techniques are used in anticipation of real-time client-side execution. At all stages efficiency is paramount. One constraint is to avoid the use of memory operations as much as possible. Wherever possible an efficient computational solution is preferred to one involving a lookup table. Most algorithms start with one or more phasor blocks, an accumulator from which other periodic functions can be derived. Sinusoidal and cosinusoidal functions are efficiently calculated using a truncated Taylor series approximation, or a reflected 4 term parabolic approximation. Noise is obtained by linear congruential pseudo-random generation. Waveshaping operations such as Chebyshev tables can in many cases be substituted by polynomial expansions limited in range to the areas we are interested in. The reason for avoiding memory table structures is to facilitate future execution with parallel threads across multiple processors where it is easier to manage numerical code expressed in a purely procedural way without reference to shared resources. Simple FM and AM methods that do not require "musical accuracy" are quite effective for sound effect design. As given by Arfib and LeBrun there is an equivalence between FM and other non-linear methods. Hybrid non-linear synthesis methods that can be interpreted partly as waveshaping and partly as modulation are attractive because they can yield natural spectral evolution with great efficiency. The disadvantage of these is that they must often be designed specifically for a narrow class of sound objects, such as pipes, metal sheets, strings and so forth, and are not easily programmable as general purpose sources.

2.2 Physical Modelling

The disadvantage of waveguide methods [5] is their requirement for memory in the form of circular buffers for propagation delays and resonators. Mass-spring-damper networks are efficient in terms of element cost but require an extremely large number of elements to approximate realistic sounds. Again, a practical compromise is to combine waveguide or MSD methods with non-linear ones to construct hybrid structures which take the best of both approaches while offering efficiency. A waveguide tapped at critical points or MSD oscillator comprising a small number of elements may be substituted as the primary oscillator in a hybrid system to add "statefullness ", by which the sound of an object depends on its previous level of excitement. This works well for struck tubes and bells and "dropped" sounds where objects undergo a large number of collisions at different points on the body within a short time interval during bouncing or rolling.

2.3 Granular approximation

Granular synthesis is appropriate for a class of sounds that are heterogeneous extents. Examples are water, waves on the seashore, rain and crowds of people. In the case of crowds, one example investigated is applause. Individual clapping is easily synthesised by carefully filtered noise bursts. Difficulties arise when combining a large number of these. Statistical analysis of density [14] can help with getting correct sounding results but it is hard in practice to achieve a consistent and realistic composition from summing individual claps efficiently. It was found that beyond a certain density, about 20 claps per second, blending in granular noise with the correct spectrum in order to blur the composite effect gave great results at low cost.

2.4 Dynamic DSP graph construction

An interesting advantage of the dataflow system given by Pd is that instead of sending entire synthesisers as sound objects to the client we can build new ones at a low background cost by sending instructions to insert and connect new atomic objects. This is a big advantage over the previously proposed MPEG4.SA mechanism for delivering synthetic sound to a remote client. Instead of sending unit generators or entire synthesisers we send procedural "code to build code". Unfortunately it isn't yet possible to do this on a 'live' process. Ideally we would like to be able to construct DSP chains dynamically, adding in unit generators or new abstractions and crossfading these into the running sound once stable. Unfortunately a current limitation of Pd, which is built on a doubly linked list datastructure, is its requirement to rebuild the entire DSP graph when new units are inserted which causes clicks and audio dropouts. Some thought is presently being given to this problem by Pd developers.

2.5 Voice management

The ability to abstract DSP operations and instantiate parallel or cascade forms in a single operation is powerful. Supercollider and Fausts DSP algebra are both able to do this but neither replaces Puredata for efficient real-time execution, the latter being an intermediate stage for producing compilable C++ code. While it currently has some shortcomings Barknechts [nqpoly4] unit is a powerful device for achieving simple parallel voicing for granular synthesis. Techniques borrowed from music technology are quite appropriate for most sound effects, which are either sustained or one-shot decaying patterns. DSP code may be turned off and deallocated once a sound has decayed beyond a threshold value. Polyphony of can be handled by voice stealing from least active or quietest instances, or simply allocated round-robin from a fixed pool.

2.6 Dynamic level of detail

We wish for sounds to grow or diminish in detail with proximity, atmospheric damping and occlusion as the player or objects move around. In traditional game audio technology such as the FMOD engine, simple attenuation or low pass filtering is applied as a function of distance. Synthetic sound wins out over data driven sound as we approach the condition of large scale sound scenes, between 100 and 1000 concurrent sources. Sampled data retains a fixed cost per source, but the density of a typical scene has a limit, both in the practicalities of accurate summation and the psychoacoutics of superposition. At around 100 signal sources the effect of contributory sampled signals for their cost diminishes. On the other hand a synthetic solution can offer variable cost per source. An example of this is breaking glass where we model each fragment as it falls onto the ground. A synthetic method can produce very realistic results and still retain a high degree of audio visual correlation by replacing the peripheral fragments with single sine or noise granules while providing a high level of detail for fragments that fall close to the player.

3 Examples

I have scores of examples of efficient real-time executable sound effects at various stages of development. Some of the more advanced synthetic sound objects in development include animals such as birds (from [16], [13], [11], [7]) and mammals, weather features such as rain and thunder (from [9]), or particular machines such as cars or aeroplanes. Most of these have specific features which are too detailed to discuss here and I am constantly exploring innovative new ways to parametrise them according to more detailed physical interpretations. The five examples below are given because they illustrate one or more general features that are common to other models or general principles that have emerged during research.

3.1 Material and structure

Once class of sounds accounting for the vast majority of incidental audio in any scene is excitations of fixed structures. Dropped objects, bullet impacts, doors closing and so forth. Noisy impulses applied to networks of parallel and cascade filters may be used to approximate the static formants of almost all common materials. In thinking about how to synthesise materials generally it is necessary to consider the propagation of energy within it. Ideally large MSD networks would be used, but these are too expensive so we need to employ less expensive filters with data from spectral analysis and some heuristics. There is some correlation between bulk modulus, density and damping. The exact damping of a material is not easy to predict from it's chemical composition, it also requires knowledge of the microstructural composition. For example, wood and cardboard are both cellulose fibres but have different densities and arrangements of material. Textbooks of physical and material constants are helpful in picking numbers, but occasionally it is necessary to just experiment with real materials and a spectrograph. Sound file [1] gives metal plate, card, wood, glass and metal formant modelling examples.

Propagation and damping Energy is reflected more in elastic materials such as glass than in wood, with metals lying somewhere between. Data has been taken from uniform excitation of plastic, cardboard, wood, china, metal and glass with equal geometry in order to create cheap filters that approximate these materials. Of course it is not possible to separate geometry, and eigenmodes of propagation, from the absolute frequencies found in any sample. What matters is the ratios of each frequency and the ratios of decay which are

[1] http://www.obiwannabe.co.uk/sounds/effect-pmimpacts.mp3

set by filter resonance. To a certain degree these simple formant filters can then be scaled to account for larger bodies of the material. Formant filters of this kind may be used as coupling materials to construct composite objects. An common example from musical instrument literature is to insert a virtual bridge between the string and sounding board of a violin, but this technique can be used generally to create many sound effects by considering the physical construction of an object.

Telephone The example given is a telephone bell[2]. What makes this sound effect work more than anything else is the crude approximation to "bakelite", a thermosetting resin used to construct old style telephones. The bells alone, which are additive models, sound reasonably good, but the sound comes alive when we add a short noise burst to approximate the solenoid armature coupled to the body of the telephone. Composing sound by thinking about the physical coupling of materials leads to coherence. The second example is a creaky door[3] which comprises two parts. The first part is a squeaky hinge and the second part is the slam of the door in its frame. In reality the hinge is attached to the door, so it makes a sound that is strongly affected by the properties of the door not just the hinge. By making the squeaky hinge resonate through the same filter as used for the door we bind these sounds into something that is recognised as the same object.

3.2 Fire

Fire is great example used to demonstrate dynamic level of detail[4]. The full model itself comprises a component analysis of the physical processes during combustion of wood. Different models are used for gas and liquid fires, but the "bonfire" example covers all of the properties we are interested in. It is an example of a heterogeneous or composite sound, comprising many different sources that together create the desired effect. These, (with the common English word used for the sound in parentheses) are, fragmentation of fuel (crackling), outgassing of water vapour and other gaseous phase compounds (hissing), aerial conflagration of particles (fizzing), turbulence at the gas combustion front (roaring), settling of fuel (clattering), quasi-periodic relaxation oscillation during outgassing (whining), and low frequency air induction (woofing). In addition we may consider some more processes such as stress caused by thermal expansion of fuel and nearby objects (groaning) and boiling of liquid phase components (sizzling, bubbling). Each of these components may be approximated with its own unit generator, the details of which are too complex to consider here, but the essential feature is that they are causally linked and blend in a particular way according to the intensity of the fire and the distance of the observer from it. The example shows a simplified model of fire that would be used at a medium level of detail. It contains the hissing, roaring and crackling elements.

3.3 Wind

Wind is an example of implicit production. Wind makes no sound at all, it is other objects in the path of moving air flow that make the sound of wind. Therefore every object in a scene has the potential to make a sound caused by turbulent flow around it. This would be impractical in a real scenario and raises the interesting question of how to limit objects that have universal implicit production methods. The boolean qualifier "Can" might be applied to objects that `Can_Burn`, or `Can_Fragment`. They represents a threshold qualifier that is evaluated before investigating any degree of combustibility, fragmentation or ability to howl in the wind. Objects that `Can_Howl` might be telephone poles and wires, rough surfaces of brickwork or large rocks. So wind is a distributed extent that emits from many objects in the location of the player. The key to a good wind sound is to model this parallelism, often with just 4 or 5 sources within a certain radius of the player. Wind sounds themselves [5] are extremely simple, band-passed noise with a fair degree of resonance and a center frequency that maps to the air velocity. Narrow objects like wires tend to emit higher and more resonant sounds, they whistle. Large irregular objects like rocks form the unfocused rushing sound of the wind. The psychoacoustic trick to really nice wind is to recognise that although each emitter moves in parallel there is a non-trivial propagation time as the wind moves across the scene. If the wind direction is left to right then a slight delay in the rise and fall of sources on the right side relative to the ones on the left produces the correct effect. This is achieved by simply low pass filtering the control signal.

3.4 Water

Flowing water is a homogeneous extent, being composed of many asynchronous sources that approximate to a rising sine wave distributed across a surface. These

[2] http://www.obiwannabe.co.uk/sounds/effect-oldphonebell.mp3

[3] http://www.obiwannabe.co.uk/sounds/effect-doorcreak1.mp3

[4] http://www.obiwannabe.co.uk/sounds/effect-fire.mp3

[5] http://www.obiwannabe.co.uk/sounds/effect-wind2.mp3

are the result of air resonating in small cavities produced as the surface is disturbed. The body of water itself imparts some resonance to the sound and indicates depth. The density of the sine wave components is a good indicator of speed of flow. We can approximate most bodies of flowing water with three variables, depth, speed of flow and impedance. The last of these is a measure of how much obstruction the liquid encounters causing irregular movement and cavities. There are several interesting ways of synthesising water such as subverting noise reduction algorithms, taking the FFT of noise and thresholding out all but a few of the mid range components before resynthesising with an inverse FFT. A less expensive way is to employ an additive/granular method which composits many small bursts containing a rising sine wave. The example given [6] is an efficient algorithm developed by trial and error that takes the positive part of the first difference of low pass filtered noise and uses it as a frequency modulator for a sine oscillator.

3.5 Footsteps

This is an example which demonstrates the range of real-time parametrisation possible [7]. It is based on an analysis of bipedal movement. Actor speed and weight are combined with a footprint pattern to obtain a ground response force (GRF) which approximates the pressure at each stage of a step. There are three parts for each step, one for the ball of the foot, one for the heel, and one for the outstep. On each step the weight is transferred from the heel to the ball in a pattern that depends on whether the actor is accelerating or slowing down, moving uphill or on the level, because different amounts of work are done in different parts of the foot [3], [6]. As the player moves between different speeds the phase relationship of six parts, three on each foot, changes. Running involves a quite different sequence than creeping. The former maximises locomotion, the latter, which is a feature of predators, minimises the change in GRF for stealth. Walking is a compromise that trades off efficient locomotion against energy expenditure [8], [1]. These control cycles have been modeled using polynomial approximations of the GRF curves and fed to a simple granular synthesiser that gives reasonable dirt and gravel surface textures. With a set of synthesisers appropriate for different surface textures this arrangement can replace many megabytes of sample data with a few kilobytes of code while giving subjectively superior results.

3.6 Machines

Some machine sounds are ambient keypoints, from the players point of view they are always running and always have been running. Noises such as fans, ventilation blowers, refrigerators and generators are part of the furniture in a world level to add a background ambiance. In the simplest interactive case a machine, weapon or tool always has at least one event hook to ”activate”. This is usually a boolean toggle and implies deactivate. This immediately adds three other considerations, a switch sound to activate/deactivate, and start and stop versions of the continuous running sound. It implies a transition between the two states, one for starting up and one for stopping. Machines that have higher degrees of control may require several continuous variables, and of course these are unique to the object in question and can’t be discussed here generally. At least one will usually be some kind of speed/intensity control. When speed or intensity of a variable change there is often a secondary effect, for example a machine acts differently under changing load while the motor spins up than it does at a constant work rate, so we often derive $|dV/dt|$ as an automatic continuous variable. The more interesting class of machines are mechanical, because we can consider real physics. Electronic machines are entirely from the realm of fantasy and can have arbitrary sounds. It’s worth noting that all machine sounds are a result of inefficiency and friction. Perfect machines would make no noise at all in theory.

Propeller Fans and propellers are modelled by a variable slope pulse wave modulating a resonant filter and short delay [8]. Sound emits from the blades as turbulence noise and is modulated in frequency by Doppler effect from the fast moving edges. Where the listener is perpendicular to the plane of rotation a more even sound is heard, but as the blade edge reaches a rotational velocity of 340m/s a chopping/snapping sound is heard where the listener is in the plane of rotation. The apparent position of the listener can be changed by altering the attack slope of the pulse wave. This is good for helicopter sounds or aircraft that move past the listener.

Electric motors Motors are modelled by considering their design, generally the brush and commutator type which give a high pitched whining sound [9]. Some sparking, which is modelled by chirp impulses can be

[6] http://www.obiwannabe.co.uk/sounds/effect-water1.mp3

[7] http://www.obiwannabe.co.uk/sounds/effect-footsteps-gravel.mp3

[8] http://www.obiwannabe.co.uk/sounds/effect-propellers2.mp3

[9] http://www.obiwannabe.co.uk/sounds/effect-robotmotors.mp3

added for good effect. One of the strongest characters that affects a motor sound is the resonance of the housing and how it is mounted to a support. If a motor is in a metal housing we use formant filters of short waveguides to simulate the body resonance. A particular subversion of FM is very helpful here, by modulating the carrier and keeping the modulator fixed in a simple FM block we can simulate the effect where the body resonates at certain speeds. This is very effective for starting and stopping sounds.

Switches Levers, switches, solenoids and relays are all impact or friction excitations that can be broken down into a simple series of clicks and resonances. The fourth power of a line segment is usually good enough as an exciter pulse to an allpass network or formant filter built according to the materials involved. Ratchets and cogs are simulated by repeating the excitation in a regular way. Again, the greatest influence on the sound is not that of the switch components themselves, but the surface it is mounted on. A lever mounted on a wooden board makes a clunk that is dominated by the resonance of the wooden panel. The example given demonstrates an alarm clock built from a synchronous arrangement of contributory clicks that have a common resonant body [10].

Mechanical assemblies Other components are modelled on a case by case basis. Virtual gears, belt drives, and reciprocating or pendulum mechanisms have all been created as the components of man made mechanical devices. The trick is to design each as a synchronous component so that they can be plugged together to create more complex machine sounds. It is easy to give a synchronous component its own clock source to make it asynchronous, but not the other way about. The example given is from the "Machine-Machine" [11], an early version of a general purpose motorised device. It features a fan, noise source, gearbox and switching sounds as well as an overdriven pipe resonance model to simulate petrol engine exhaust.

4 Conclusions

Efficient synthetic methods for game audio have been demonstrated. No single synthetic method solves all required cases and a mixture of approaches is required. Issues like dynamic level of detail and cost management have been examined and some approaches offered. Much of the groundwork for specific elements has already been done by researchers working on audio synthesis. It has been built upon here and will continue to be improved by the next generation of synthetic sound designers. Like the field of computer graphics the realism of synthesised audio will improve dramatically once a serious investment is made in the technology. With the appearance of multi-core processors the deployment of synthetic audio in games is no longer limited by resources. Programming problems such as making good use of parallelism, managing a highly dynamic DSP graph in an environment with variable resources, and automatically generating filter coefficients from geometry eigenvalues still exist. Larger scale software engineering issues allowing producers and designers to work on sound objects "in world" with appropriate tools should be a priority for practical use. A reasonable and measured approach to commercial deployment is to incrementally introduce the technology alongside existing methods.

4.1 Future directions

I believe this is a unique project which attempts to bring together the work of many researchers and add new insights. The aim is to provide the full range of synthetic sounds necessary for a complete video game and control strategies to manage them. This will take interactive sound far beyond the possibilities of currently used sample data. More work must be done to make effective demonstrations, starting with small games projects where the audio can be complemented with synthesis. The final goal of hearing a real-time generated fully synthetic game audio soundtrack using a proper "sound engine" is some way off. The remainder is advocacy and biding time for the opportunity to assemble a team capable of attacking this task. The next stage is to take this work in-house and evaluate the ability of sound designers to work with the concept of automatic generation, and using dataflow tools to construct sound objects with dynamic methods.

References

[1] S. H. Adamczyk, P. G. Collins and A. D. Kuo. The advantages of a rolling foot in human walking. *J. Exp. Biol. Vol 209*, pages 3953–3963, 2006.

[2] Rob Bridgett. Updating the state of critical writing in game sound. *Gamasutra, 31st August 2006*, 2006.

[3] Evie Vereecke K D'Aout L Van Elsacker D De Clercq. Functional analysis of the gibbon foot during terrestrial bipedal walking. *American Journal of PhysicalL Anthropology*, 2005.

[10] http://www.obiwannabe.co.uk/sounds/effect-alarmclock.mp3

[11] http://www.obiwannabe.co.uk/sounds/effect-machines6.mp3

[4] P. Cook and G. Scavone. The synthesis toolkit (stk. *Proceedings of the International Computer Music Conference, Beijing (1999).*, 1999.

[5] S. Van Duyne and J.O. Smith. Physical modeling with the 2-d digital waveguide mesh. *Proc. Int. Computer Music Conf. (ICMC'93), pages 40–47, Tokyo,Japan, Sept. 1993.*, 1993.

[6] P Aerts EE Vereecke K D'Aout. Speed modulation in hylobatid bipedalism: A kinematic analysis. *Journal of Human Evolution*, 2006.

[7] Seppo Fagerlund. Acoustics and physical models of bird sounds. *Laboratory of Acoustics and Signal Processing, HUT, Finland.*, 2003.

[8] Willems P.A. Cavagna G.A. and Heglund N.C. External, internal and total work in human locomotion. *J. Exp. Biol. Vol 198*, pages 379–393, 1995.

[9] Dipankar Roy Herbert S. Ribner. Acoustics of thunder, a quasilinear model for tortuous lightning. *J. Acoust. Soc. Am. 72*, page 6, 1982.

[10] Steve Kutay. Bigger than big: The game audio explosion. *gamedev.net, March 2007*, 2007.

[11] Frederico Avanzini Mark Kahrs. Computer synthesis of bird songs and calls. *Proc. COSTG-6 Conference on Digital Audio Effects (DAFX-01), Limerick, Ireland.*, 2001.

[12] S. McAdams and E. Bigand. Thinking in sound. the cognitive psychology of human audition. *Oxford Science Publications*, 1993.

[13] Hans Mikelson. Bird calls. *Csound Magazine*, Winter:2, 2000.

[14] Leevi Peltola. Analysis, parametric synthesis and control of hand clapping sounds. *Masters Thesis. Laboratory of Acoustics and Signal Processing, Helsinki University of Technology.*, 2004.

[15] Scott Selfon. Techniques for fighting repetition in game audio. *GDC Archives, March 2005*, 2005.

[16] T. Smyth and J.O.Smith III. The sounds of the avian syrinx – are the really flute-like? *Proceedings of DAFX 2002, International Conference on Digital Audio Effects, Hamburg, Germany, September 2002*, 2002.

[17] Stephane Letz Yann Orlarey, Dominique Fober. Faust: Functional programming for signal processing.

Towards a Personal Automatic Music Playlist Generation Algorithm: The Need for Contextual Information

Gordon Reynolds, Dan Barry, Ted Burke and Eugene Coyle

The Audio Research Group,
School of Electrical Engineering Systems,
Dublin Institute of Technology, Kevin St, D8, Ireland

{gordon.reynolds | dan.barry | ted.burke | eugene.coyle} @dit.ie

Abstract. Large music collections afford the listener flexibility in the form of choice, which enables the listener to choose the appropriate piece of music to enhance or complement their listening scenario on-demand. However, bundled with such a large music collection is the daunting task of manually searching through each entry in the collection to find the appropriate song required by the listener. This often leaves the listener frustrated when trying to select songs from a large music collection. In this paper, an overview of existing methods for automatically generating a playlist is discussed. This discussion outlines advantages and disadvantages associated with such implementations.

The paper then highlights the need for contextual and environmental information, which ultimately defines the listener's listening scenario. Environmental features, such as location, activity, temperature, lighting and weather have great potential as meta-data. Here, the key processes of a basic system are outlined, in which the extracted music features and captured contextual data are analysed to create a personalised automatic playlist generator for large music collections.

1. Introduction

With the aid of cost affordable storage and greater device inter-connectivity, a listener's personal music collection is capable of growing at an extraordinary rate. When faced with such large music collections, listeners can often become frustrated when trying to select their music. Hence, it becomes increasingly difficult for a listener to find music suited for a particular occasion.

To further the problem of music selection, today's culture of mobile technology enables the listener to transport an entire music collection in the pocket. Mobile music players now boast of song storage of up to 40,000 songs. As a result, many listeners will plan and prepare playlists for mobile activity that corresponds to a specific activity or mood, such as travelling and exercising. However, according to Suchman [1], plans alone do not dictate actions but only provide a framework that individuals can use to organise action. This implies that the listener attempts to execute previously prepared plans while continuously adapting their actions to the environment [2]. This scenario has led to a study of context-aware music devices [2] and the examination of the role of emotion in music selection.

This paper discusses a design proposal to further the research area of context-aware and emotion-aware music devices. In particular, how environmental data may be used to infer a listener's mood and how such information may integrate into the process of automatically generating a music playlist.

2. Overview of Existing Playlist Methodologies

This section provides a definition of a playlist and presents playlist attributes asscociated with such a definition. The automatic playlist generation process is then discused with an overview of its major themes.

2.1. Defining a Playlist

A playlist may be defined as a finite sequence of songs which is played as a complete set. Based upon this definition there are three important attributes associated with a playlist. These attributes are: 1) the individual songs contained within the playlist, 2) the order in which these songs are played and 3) the number of songs in the playlist.

The Individual Songs in the playlist are the very reason for generating such a playlist. It is therefore essential that each song contained within the playlist satisfies the expectations of the listener. These expectations are formed based upon the listener's mood, which in turn is influenced by the environment.

The Order in which the songs are played provides the playlist with a sense of balance which a randomly generated playlist can not produce. In addition to balance, an ordered playlist can provide a sense of progression such as, a playlist progressing from slow to fast or a playlist progressing from loud to soft.

The Number of Songs in a playlist determines the time duration of the playlist. An understanding of the length of a playlist is important, as song ordering and song balancing of the playlist is unachievable otherwise.

2.2. Playlist Implementations

As catagorised by Vossen [3], the current status of research involving automatic playlist generation is portrayed under two major types of impletmentations. These implementations are 1) Recommender Based Playlists and 2) Constraint Based Playlists.

2.2.1. Recommender-Based Playlists

A Recommender-Based System estimates the user's music preference from a localised music collection and then generates a set of songs based on these estimates from a wider music collection. There are two common approaches to implement a Recommender-Based System, these are 1) Content-Based Learning and 2) Collaborative Filtering.

Content-Based Learning analyses each song in the music collection and then matches songs which have musically similar attributes, such as tempo, instrumentation or genre. If a listener likes a particular song, usually indicated by the user listening to the entire song, then the Content-Based System will recommend songs that are similar to that song. Figure 1 outlines the Content-Based Playlist generation procedure.

As shown in Figure 1, the user is required to specify a *seed song* and the number of songs required in the playlist. The seed song represents the type of music that the listener wants to listen to. The system then filters the music collection based on similarity to the seed song. A similarity song space is hence created from which a playlist is generated.

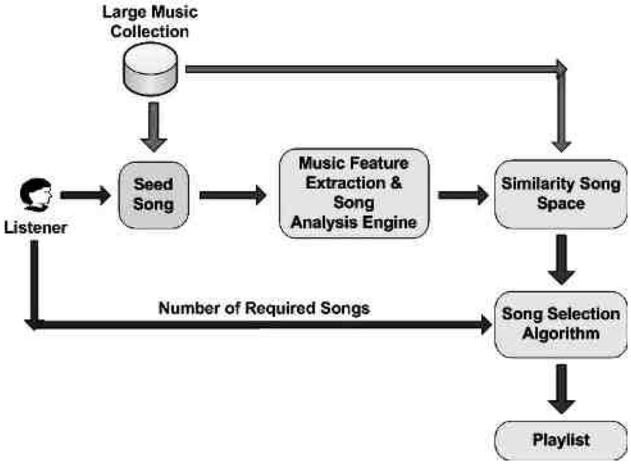


Figure 1: The Content-Based Playlist Generation Procedure.

In a Content-Based System, a significant disadvantage is that song order has no meaning since all the songs are similar. This may also suggest that the playlist may seem dull due the lack of song variation. However, such a system may be useful in circumstances where a themed playlist is required.

Collaborative-Filtering is a community process, as it employs a multi-user approach that uses explicit preference to match songs to a specific user. The system then expands this set of songs by finding another user with a similar taste in music. The system then recommends songs from this user back to the original user [4].

Figure 2 outlines the basic principle of Collaborative-Filtering in a Venn diagram. With Collaborative-Filtering, song order is not taken into account. However, Collaborative-Filtering does provide a varied playlist which may be more interesting to listen to when compared to a Content-Based Learning system.

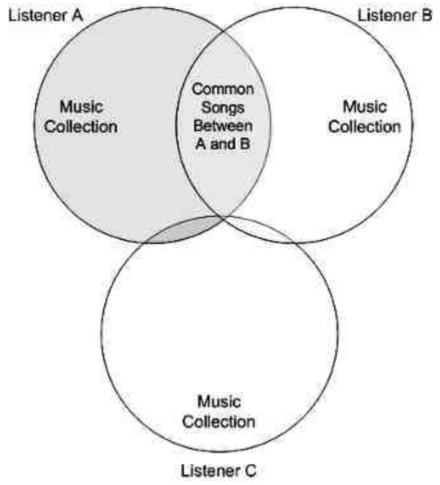


Figure 2: A Venn Diagram Indicating Collaborative-Filtering.

From Figure 2, Listener A has a lot in common with Listener B in terms of their music collections, compared to Listener C. As a result, the Collaborative-Filtering System will recommend songs from Listeners B collection to listener A and recommend songs from Listeners A collection to listener B. Nothing is recommended from Listeners C collection to either Listener A or Listener B. The system assumes that since Listener A and Listener B have so much music in common that their preferences must be the same, i.e. they have the same musical taste. Hence, Listener A would enjoy Listener B's music collection and vice-versa.

2.2.2. Constraint Based Playlist

In the Constraint-Based approach, song order in the playlist becomes a primary focus and hence to date, the only systems that consider the three requirements for a playlist, these are 1) Songs, 2) Order and 3) Length. This is achieved by forming a rule set which defines the song order in a playlist. An overview of such a procedure is given in Figure 3.

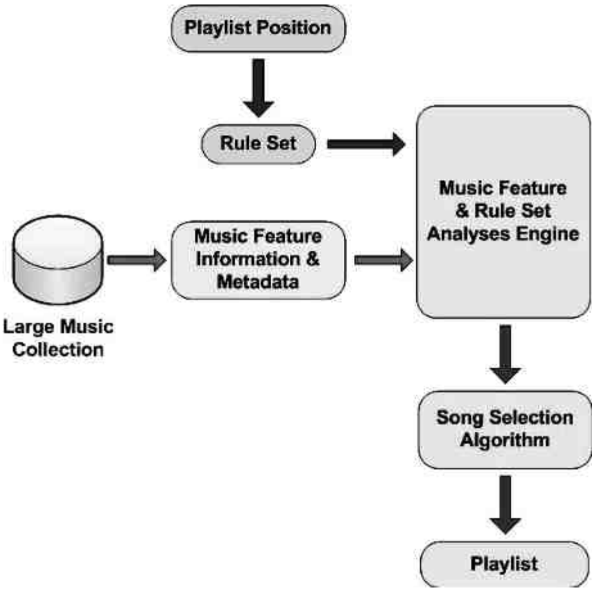


Figure 3: Overview of the Constraint-Based Process.

The rule set provides a set of definitions to which a song must adhere to before being selected. An example of a rule set, as implemented by Vossen in [3], defines a global rule by the requirement that the tempo for each song in the playlist must be above a desired value. Other examples of constraint rules include, that no two adjacent songs in the playlist can be from the same artist or same album. Once the appropriate song is found in the music collection it is inserted into the suitable playlist location. The system then searches for a song to fit the rule set of the next playlist location.

Based on the previously presented playlist implementations, this project is currently investigating the compatability of a Constraint-Based approach in its implementation. The Constraint-Based approach provides the most flexibility yet strict framework for creating an algorithm for automaticlly generating a music playlist.

3. The Need for Contextual/Environment Data

The selection process is dominently ruled by the emotional state and attitude of the individual. Individuals are a function of mood [5] and music selection is no different. Therefore, to provide a listener with a meaningful personalised automatic playlist generation system, it is ideal for the system to consider the listeners mood. Measuring such a parameter directly from the listener borders on impossibility. However, with the establishment of attitude theory in the 1930's, strong links have been forged between an individuals environment and attitude, which in turn defines mood and behaviour [6]. The experience of an individual in the outside world reflects how they feel on the inside [5].

With such strong defined theoritical links between an individuals environemnt and their behaviour, it may be possible to reduce the need to infer mood from a listener in order to create an automatic playlist of songs to suit that mood. Such an approach may circumvent problems raised by *Tolos et al.* in [7], such as defining a set of moods that is relatively unabigous, widely accepted and useful for the average user.

It is proposed to design a system that will monitor a listeners environment and observe their choices in music selection. Analogous to a basic input/output black box system, given the inputs (environmental features) and the outputs (the selected songs) one is required to reconstruct the transfer process, i.e the listeners mood or behaviour, Figure 4.

.

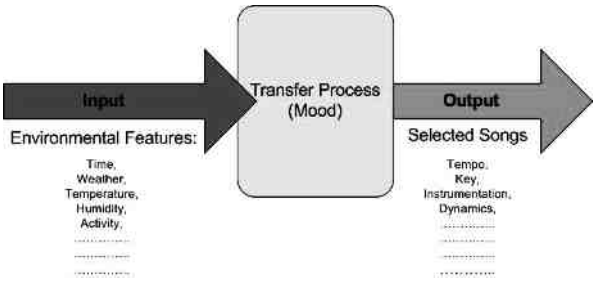


Figure 4: A black box approach, given the inputs and the outputs, one is required to reconstruct the transfer process.

It should be noted that this process is not equivelent to extracting the musical mood or the musical emotional state of a composition as implemented by *Liu et al.* [8]. With the aid of music theory, *Liu et al.* showed with the implementation of an algorithm, that a composition may occupy a particular emotional space, musically speaking.

However, each individual listener subjectively interperates the emotional or mood space of a composition based on their experience. Hence, the same composition is capable of causing a diverse array of emotions within the listening community. Such an example of this is a composition by *Carl Orff* titled *Carmina Burana*. A classical enthusiast and non-horror movie viewer may recognise and perceive the piece in the appropriate classical context it was written in. However, the non-classical lover and horror movie fanatic will recognise the piece as the theme tune to the horror movie '*The Omen*'. In this case, hearing the music piece out of context may induce a sense of fear, uneasiness and terror for the listener. This is because the listener only associates this theme with horror.

The long term consistancy and reliability of using envionmental data in the selection process of automatic playlist generation is founded upon the habitual qualities of human nature. Covey explains, that an individual's action and re-action is pre-conditioned by their environment [5]. Also, as outlined by Ostrom in [6], an individuals attitude, which is a description of their behaviour and formed through experience in their environemnt, operates to make the individuals world predictable and orderly.

3.1. Choosing Appropriate Environmental Features

It is required to identify and categorise environmental features that may affect a listener's mood or music selection process. As an example of how environmental features can affect mood, an investigation into to the effect of lighting on office workers [9] discovered that natural lighting reduces stress and promotes a general sense of *good being* compared to artificial lighting. It is also suggested that the type of lighting combined with the intensity of the artificial light may determine the level of negative effects experienced. In addition, it has also been documented that the weather appears to influence mood and productivity [10].

To commence, it is proposed to consider seven environemnetal features. These features are 1) Time and Date, 2) Weather, 3) Lighting Conditions, 4) Humidity Conditions, 5) Temperature Conditions, 6) Noise Conditions and 7) the Listener's Activity.

3.2. Capturing Environment Data

A brief outline on how each of the environmental features may be captured is given in this chapter.

3.2.1. Time and Date

It is possible to capture time and date using the system clock of the proposed music player, which is a PC based device. With the availability of time and date, it is possible to expand the analysis to include day of the week (Monday, Tuesday, ...), month of the year (Janruary, February, ...), time of the day (morning, afternoon, ...) and season (Winter, Summer, Autumn and Spring).

3.2.2. Weather

It is proposed to obtain weather data through an available METAR service online due to its strict and compact data format. METAR data is available from all airports and it is regularly updated on a thirty minute schedule.

3.2.3. Lighting, Temperature, Humidity and Noise Conditions

With the use of an appropriate sensing device, lighting, temperature, humidity and noise conditions can be monitered and captured. An array of hardware devices exist to capture such parameters. Hardware considerations are discussed further in Chapter 5.

3.2.4. Activity

It is proposed to determine a listeners activity in two forms, these are 1) social scheduling and 2) using an accelerometer. Social scheduling is based upon calender events which involves taking advantage of predictable behavour such as working schedules, travelling schedules, excercising schedules and relaxing schedules. Further information on a listener's activity may be captured electronically with the use of an accelerometer, such as the E-LIS3L02AS4 from STMICROELECTRONICS. An accelerometer is capable of measuring a listener's physical movement, such as walking, running and jumping.

To summerise, environmental parameters have a significant affect on mood and hence influences the music selection process. Therefore, environmental features have great potential as meta-data to allow the listener greater flexibility when searching or accessing a music collection. In addition, environmental features may provide a valuable source of information for an automatic playlist generation algorithm in the generation of playlists to suit a listener's mood.

4. Integrating Contextual/Environment Data into an Automatic Playlist Generation System

Textual meta-data such as artist's name, song title and music genre were initially the only mechanism a listener had for indexing their music collection. However, in recent years, musicians and technologists established the research field of Music Information Retrieval (MIR). One of the principle achievements of MIR was to extend the available meta-data to include musical features extracted directly from acoustic signals. These musical features include tempo, key and timbre. These features allow the user to express music selection and indexing based on actual acoustic information rather than tagged textual information.

It is proposed to develop a system that will extend the range of meta-data further. It proposes to use environmental information to represent the listeners listening scene and mood. With the consideration of such environmental features, Figure 5 outlines three unique feature spaces which may be used to represent a music collection, namely textual descriptions, musical features and environmental features.

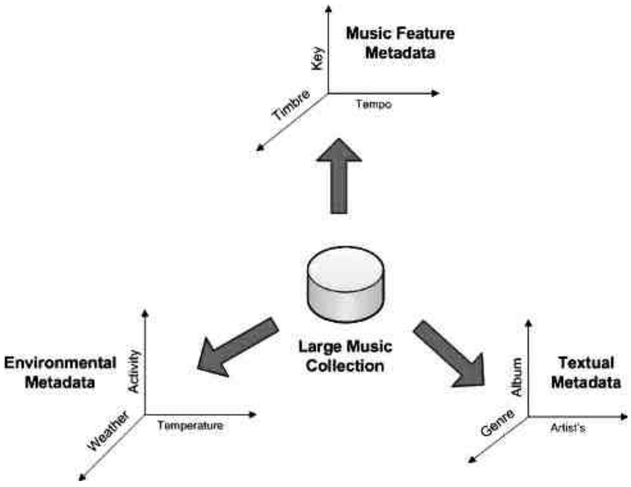


Figure 5: Outlines three unique feature spaces in which a listener's music collection may be indexed.

4.1. Existing Meta-Data

Meta-data is described as everything that is not essence, that is, it is data about data [11]. In audio terms, it usually means data that describes, relates to, or structures essence. To date, the use meta-data is the most dominant means that allows a listener to search a music collection. The search criteria can be specified by artist's name, album name, album genre and release date – only to name a few.

Given the vast range and availability of meta-data, the proposed audio system will integrate such information into its music selection process in combination with environmental features and music features.

4.2. Music Feature Extraction

The proposed system will initially consider three music features. These features are 1) Global Timbre – generally used to describe the similarity of two compositions, 2) Tempo – which describes the speed of a compoition and 3) Musical Key – which describes the relative pitches between notes.

However, as the system develops additional music features will be examined, for example music dynamics, temporal features and spectral features.

4.2.1. Global Timbre

Timbre is a percepual audio characteristic which allows listeners to perceive or distinguish between two sounds with the same pitch and same intensity [12]. The term *Global Timbre* refers to the timbre description covering the full duration of a composition and not just at a particular instant in time nor particualr instrument.

In [13], *Aucoutrier* implements a music similarity technique that employs the use of Mel Frequency Cepstrum Coefficients with Gausian Mixture Models. Based on subject evaluation, Aucoutrier found 80% of the songs suggested by the system as being similar was also identified as being similar by the test users. Logan has also used this type of method to indicate music similarity with similar positive results.

4.2.2. Tempo

Tempo is defined as the speed at which a musical composition is played at [12]. Experiments concerning musical tempo have conveyed its' effects on people in the areas of performance of track athletes [14] and on general spatial awareness, arousal and mood [15].

Leue uses Spectral Energy Flux in combination with a Comb Kernal Filter Bank to deduce tempo from a composition [16]. With such a system, *Leue* acheived accuracies of 80% for pop music and 63% for classical music. Pop music generally produces the more accurate result with tempo detection due to the heavey percussive nature of pop music compared to classical music. *Alonso et al.* has also investigated tempo tracking and extraction using a similar method and has obtained accuracies of up to 89.7%.

4.2.3. Musical Key

Musical key may be defined as the relative pitches or notes contained within a composition [12]. Determining the key of a composition has several applications including mood induction. They mode of the key is deemed to provide a specific emotional connatation [17].

Using a Chroma-based estimation technique, Peeters uses Harmonic Peak Subtraction with a Hidden Markov Model to extract a key from a composition [18]. Confined to the following classical categories, keyboard, chamber and orchrastra an accuracy of 90.3%, 94% and 85.3% was obtained respectively. Pauws also introduced a key extraction algorithm based on chromagrams with an accuracy of 75.1% for classical music [19].

4.3. Using Intelligence

The heart of the proposed system requires the implementation of an intelligent engine capable of learning and applying the learned information to intelligently make a decision. Intelligent systems currently being investigated are Artificial Neural Networks and Hidden Markov Models.

However, due to the complexity of the system it is expected that several intelligent systems are required. For example, in music feature extraction, Gaussian Mixture Models are reported to provide the most efficient performance in calculating song similarity were as, Hidden Markov Models are noted for their performance in segmentation [20].

In the context of an automatic playlist generation algorithm, several different intelligent models have been used. *Jin* describes a process using Hidden Markov Models and experiences an 83% improvement in retrieval time when compared to a forward searching algorithm [21].

4.4. System Process Overview

This section provides a high-level overview of the processes required within the proposed system. This overview is described in two parts, 1) the Learning Process and 2) the Operational Process.

4.4.1. The Learning Process

Figure 6 overviews the learning process of the proposed playlist generation system with integrated environmental meta-data. As the listener selects the required music manually, the system monitors all events in the background which involves the capturing of environmental features. These environmental features are then added to the chosen songs as new meta-data. As different songs are selected, the system identifies and quantifies how each selected song is similar and how they differ. MIR algorithms then use this information to find other songs deemed musically similar to the chosen songs within the bounds of a similarity threshold. Once identified, these songs are also tagged with the same environmental meta-data. The identification process is based upon existing meta-data and music features extracted from each song as previously described in Section 4.2. All results are then catalogued within the system for future reference.

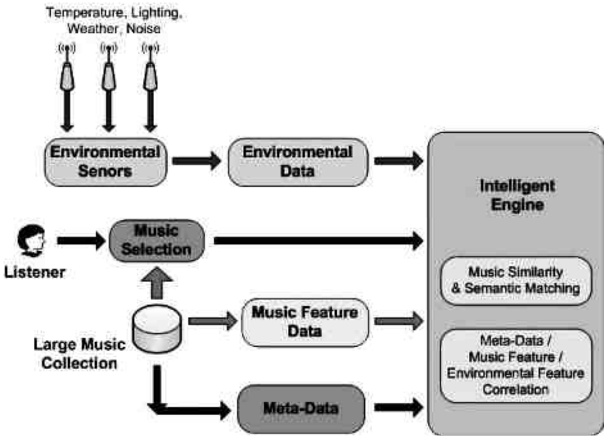

Temperature, Lighting, Weather, Noise
Environmental Senors
Environmental Data
Intelligent Engine
Music Selection
Listener
Music Similarity & Semantic Matching
Large Music Collection
Music Feature Data
Meta-Data / Music Feature / Environmental Feature Correlation
Meta-Data

Figure 6: Learning Process of the Proposed Playlist Generator.

Once the system has gained enough experience of the listener's selection process, the system is then capable of automatically generating a meaningful music playlist to suit the listeners listening environment and hence their listening needs. Such a process is proposed in Figure 7.

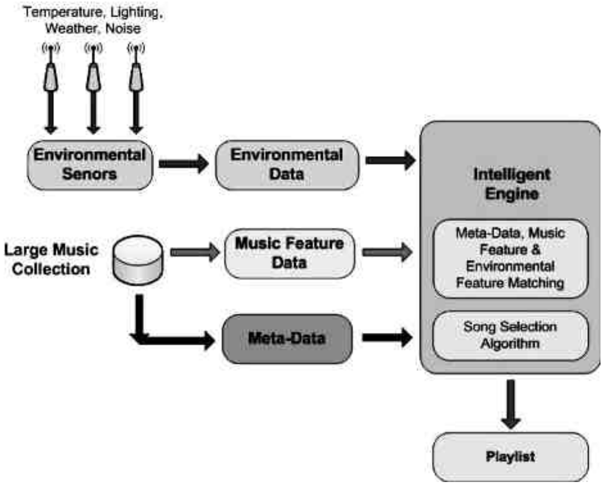


Figure 7: Operational Process of the Playlist Generator.

4.4.2. Operational Process

The trained system, Figure 7, operates without the need for the listener's interaction, since the intelligent engine functions under the same set of parameters and definitions in which it was trained. When the listener requires to listen to music, the trained system will analyse the current listening environment through a sensor array. Based upon the systems previous training, the captured environmental data is matched and assigned to appropriate music features. The music collection is then filtered according to the required features and a song selection algorithm generates the appropriate playlist.

5. Hardware Considerations

It is important that the development and test hardware platform for the proposed music player is mobile, unobtrusive and does not contribute to the listener's mood. In addition, the music device must have appropriate storage capacities and processing power. As a result, a small form factor PC such as the Samsung Q1b – Ultra Mobile PC is currently being used as a testbed. This device is portable and highly inter-connectable with the availability of onboard LAN, WLAN, Bluetooth and USB services. The system uses a 7" touch screen and operates under Windows XP Tablet Edition or Windows Vista rather than a scaled version of Windows such as PocketPC.

A hardware consideration for sensors to allow the capture of environmental data includes the HOBO U12 data logger which is currently being used. This device is compact, portable and self-contains temperature, light and humidity sensors. The unit also has an available external data channel which allows the connection of an external noise level sensor. The data logger is accessible through a standard USB connection and is compatible with the Keyspan USB Server allowing access via Ethernet or WiFi.

To detect and monitor a listener's movement, an Olimex accelerometer (MOD-MMA7260Q) is currently being used. This is a 3-axis device and is pre-mounted on a development board which includes the appropriate support ICs. A mini-USB connection is required to interface with the accelerometer.

6. Conclusions

The concept of an automatic music playlist generation system is presented in this paper. An overview of existing playlist generation techniques is discussed were advantages and disadvantages are outlined.

A Constraint-Based approach was identified as an appropriate playlist generation method. This is due to the fact that it completely encapsulates the entire definition of a playlist and it provides a flexible yet strict framework to work within.

In addition, the key processes of a proposed system were outlined, in which meta-data, the extracted music features and captured environmental data are analysed to create a personalised automatic playlist generator for large music collections.

To further research in this area, a survey has been created to gather appropriate information. This survey can be found on the Audio Research Group's website at www.audioresearchgroup.com/survey. All participation is welcomed.

To conclude, this paper has shown that mood determines the listeners music selection process. Also in reverse, it was shown how music may induce mood in a listener. But more importantly, this paper has discussed and demonstrated that an individual's environment strongly influences mood and hence the listeners music selection process. Based upon these strong influences, it is concluded that environmental features pertaining to a listeners environment has significant potential as meta-data and may provide a valuable resource in the automatic generation of music playlists.

References

[1] L. Suchman. *Plans and Situated Actions, The Problem of Human-Machine Communication.* Cambridge University Press (1987).

[2] S. Tamminen, A. Oulasvirta, K. Toiskallio and A. Kankainen. *Understanding Mobile Contexts*. Proc. Mobile HCI, 17-31 (2003).

[3] M.P. Vossen. *Local Search for Automatic Playlist Generation.* Masters Thesis, Technische Universiteit Eindhoven (2006).

[4] N. Kravtsova, G. Hollemans, T.J.J. Denteneer, and J. Engel. *Improvements in the Collaborative Filtering Algorithms for a Recommender System*. Technical Note NL-TN-2001/542, Philips Research, Eindhoven, (2002).

[5] S. R. Covey. *The 7 Habits of Highly Effective People.* Simon&Schuster UK LTD, (2004)

[6] A.G. Greenwald, T.C. Brock, T.M. Ostrum. *Psychological Foundatins of Attidue.* Academic Press, New York. (1973).

[7] M. Tollos, R. Tato, T. Kemp. *Mood-Basd Navigation Through Large Collection of Musical Data.*. The 5th International Conference on Music Information Retrieval, Barcelona (Spain), ISMIR (2004).

[8] D. Liu, L. Lu and H-J. Zhang. *Automatic Mood Detection from Acoustic Data*. John Hopkins University, (2003).

[9] L. Edwards and T. Torcellini. *A Literature Review of the Effects of Natural Light on Building Occupants*. Technical Note NREL/TP-550-30769, National Renewable Energy Laboratory, Colorado, (2002).

[10] A.G. Barnston. The *Effect of Weather on Mood, Productivity, and Frequency of Emotional Crisis in a Temperate Continental Climate*. International Journal of Biometeorology. Vol. 32, No. 2, (1998).

[11] AES, *Demystifying Audio Metadata*, Journal of the Audio Engineering Society, Vo. 51, No.7/8, 744-751 (2003).

[12] Orio. *Music Retrieval: A Tutorial and Review.* Foundations and Trends in Information Retrieval. Vol. 1, No. 1, 1-90 (2006).

[13] J-J. Arcourtier and F. Pachet. *Music Similarity Measures: What's the Use?* The 3rd International Conference on Music Information Retrieval, Paris (France), ISMIR (2002).

[14] J.R. Brown. *The Effects of Stressed Tempo Music on Performance Times of Track Atheletes*. Florida State University. Florida, (2005).

[15] G. Husain, W. F. Thompson and E. G. Schellenberg. *Effects of Musical Tempo and Mode on Arousal, Mood and Spatial Abilities. Music Perception*, Vol. 20, No. 2, 151-171, (2002).

[16] I. Leue and O. Izmirli. *Tempo Tracking with a Periodicity Comb Kernel.* The 7th International Conference on Music Information Retrieval, Victoria, BC (Canada), ISMIR (2006).

[17] Kastner. *Perception of Major / Minor Distinction: OIV Emotional Connections in Young Children.* Music Perception, Vol. 8, No.2, 189-202.

[18] G. Peeters. *Chroma-Based Estimation of Musical Key from Audio-Signal Analysis.* The 7th International Conference on Music Information Retrieval, Victoria, BC (Canada), ISMIR (2006).

[19] S. Pauws. *Musical Key Extraction from Audio.* The 5th International Conference on Music Information Retrieval, Barcelona (Spain), ISMIR (2004).

[20] J.C. Platt, C.J.C. Burges, S. Swenson, C. Weare, and A. Zheng. *Learning a Gaussian Process Prior for Automatically generating music playlists.* In proc. Of the 14th Conference on Advances in Neural Information Processing Systems, Volume 14,(2001)

[21] H. Jin and H.V. Jagadish. *Indexing Hidden Markov Models for Music Retrieval*. IRCAM-Centre Pompidon, Michigan, (2002).

Perceptual Representations for Classification of Everyday Sounds

Elena Martinez Hernandez[1,2], Kamil Adiloglu[1], Robert Annies[1],
Hendrik Purwins[1,2], Klaus Obermayer[1]
,[1] Neural Information Processing Group, Berlin University of Technology,
{kamil,robokopp,oby}@cs.tu-berlin.de
[2] Music Technology Group, Pompeu Fabra University,
{emartinez,hpurwins}@iua.upf.edu

Abstract. In the recognition and classification of sounds, extracting perceptually and biologically relevant features yields much better results than the standard low-level methods (e.g zero-crossings, roll-off, centroid, energy, etc.). Gamma-tone filters are biologically relevant, as they simulate the motion of the basilar membrane. The representation techniques that we propose in this paper make use of the gamma-tone filters, combined with the Hilbert transform or hair cell models, to represent everyday sounds. Different combinations of these methods have been evaluated and compared in perceptual classification tasks to classify everyday sounds like doors and footsteps by using support vector classification. After calculating the features a feature integration technique is applied, in order to reduce the high dimensionality of the features. The everyday sounds are obtained from the commercial sound database "Sound Ideas". However, perceptual labels assigned by human listeners are considered rather than the labels delivered by the actual sound source. These perceptual classification tasks are performed to classify the everyday sounds according to their function, like classifying the door sounds as "opening" and "closing" doors. In this paper, among the gamma-tone-based representation techniques, other spectral and psycho-acoustical representation techniques are also evaluated. The experiments show that the gamma-tone-based representation techniques are superior for perceptual classification tasks of everyday sounds. The gamma-tone filters combined with a inner hair cell model and with the Hilbert transform yield the most accurate results in classifying everyday sounds.

1 Introduction

Unlike music and speech, the audio analysis and recognition of everyday sounds have not yet received so much attention in the literature. Nonetheless, everyday sounds play a significant role in communication, localisation and interaction. In this paper, we focus on the ability of machine learning algorithms to identify and classify sounds by example to gain insight what aspects of – mostly complex – sounds lead to a certain categorisation by humans.

Everyday sounds in urban environments are emitted from machines, human interaction with mechanical devices, and natural phenoma. In a natural environment, sounds get modulated by the acoustical properties of the environment itself and get mixed with different sources of sounds. This shows the complexity of everyday sounds. However, for analysis purpose, it is necessary to reduce the complexity of the representation to meaningful features.

The question is whether and how it is possible to detect properties of the sound generating process and/or listeners perception, i.e. what is the sound source, what materials are involved, and what impressions are received by a listener. We investigate this question in a classification framework. Several classification studies have been made to distinguish different kinds of everyday sounds from other types of sounds such as music, speech, etc. However, in our approach we classify only everyday sounds based on the function they fulfil, their material or shape, or the objects they interact with. Obviously, this task, classifying everyday sounds is much more difficult then distinguishing everyday sounds from other sound classes, which have totally different characteristics.

We would like to thank Mathieu Pellerin, Guillaume Lemaitre, and Olivier Houix from IRCAM for their support.

This work is supported by the European CLOSED project (FP6-NEST-PATH "measuring the impossible" project no. 29085).

2 Representation

Low level spectral features, like zero-crossings, centroid, roll-off etc. of a given sound give useful information about the sound. However this information is not sufficient to understand sound perception. In order to be able to understand how perception works, psychoacoustical facts should be considered, and psychoacoustical features should be utilised to define efficient representation schemes.

Mel Frequency Cepstrum Coefficients (MFCC's) [7] are well established representation scheme, which dominate applications in speech recognition and music processing. They have rarely been applied to environmental sounds. MFCC's are short-term spectral based features, based on the Mel scale. The Mel scale is a mapping between the actual frequency values and the perceived pitch. This mapping is linear for the low-frequency values. As the frequency increases, the mapping becomes logarithmic. Generally the first 13 MFCC features are used as descriptors to represent a given sound in different tasks, because as it has already been shown, these first coefficients concentrate most of the signal energy [17].

However, there are other psychoacoustically or biologically motivated methods, which take the critical bands into account, where inputs whose frequency difference is smaller than the critical bandwidth cause the so-called beats. Another methods aim to simulate the cochlea in the inner-ear. In our representation method, we introduce the gamma-tone filters, which are considered to be biologically motivated as well.

2.1 Gamma-tone Filterbank

A gamma-tone auditory filterbank [11, 12] incorporates two insights into auditory physiology: 1) the higher frequency resolution for low frequencies, 2) the higher temporal resolution for high frequencies. With increasing centre frequency the spacing of the gamma-tone filters increases and the length of the filter decreases. Mimicking the basilar membrane, spacing and bandwidth of the filter is based on the equivalent rectangular bandwidth (ERB) [4]. Roughly, the centre frequencies are spaced linearly for low frequencies and logarithmically for high frequencies. ERBs are similar to the Bark or the Mel scale. Due to its properties, the ERB scale is a highly biologically plausible representation.

As a pre-processing of the everyday sounds, we use the gamma-tone filter implementation in Malcolm Slaney's Auditory Toolbox [13, 14]. Starting with the lowest centre frequency of 3 Hz, we use 18 gamma-tone filters in total. Therefore, for each given sound, we obtain 18 filter responses from the gamma-tone filter bank.

The gamma-tone filters can be combined with other representations, in order to obtain a more complete representation scheme.

2.2 Hilbert Transform

The first method, which we can combine the gamma-tone filters with is the Hilbert Transform. The Hilbert Transform is mainly used to explain the relationships between the real and imaginary parts of a signal. The Hilbert transform of a signal is nothing but the convolution of the time domain signal with $\frac{1}{\pi t}$. Combining the Hilbert transformed signal with the original signal, we obtain the analytic signal. This process deletes the negative components of the signal in the frequency domain, and doubles the amplitudes on the positive side. Furthermore the analytic signal is a base band signal.

In our representation scheme, the Hilbert Transform is applied to each gamma-tone filter response. After applying the gamma-tone filterbank the Hilbert transform is calculated for each filter response. Then the power spectral density is calculated for each of the Hilbert transformed filter responses [15]. The power spectral density is the Fourier Transform of the autocorrelation of the signal. The periodogram method is generally used to calculate the power spectral density. In the standard case, the power per unit frequency is calculated, where the results have the unit $\frac{power}{frequency}$. However, we calculate the mean-squared spectrum. The mean-squared spectrum is calculated for each frequency value depending on the sampling rate. Therefore, the unit of the mean-squared spectrum values are $power$. Figure 1 shows an example mean-squared spectrum gamma-tone filter. In this figure, a closing door sound has been taken. After applying gamma-tone filters onto the sound, we obtained 18 filter responses, one for each filter in the filterbank. Then a single filter response was taken, in order to apply the Hilbert Transform onto the filter response. As the last step, the mean-squared spectrum was calculated for the Hilbert transformed filter response.

After these steps, we obtain the mean-squared spectrum for each Hilbert transformed filter response which can be considered as a matrix. However, we should reduce the dimensionality of this matrix, in order to be able to use them as a representation of the sound. Therefore we sum up these values within each of four groups of adjacent centre frequencies [1]. We take the average of the calculated values for each group. These groups are the DC values, the frequency interval 3-15 Hz, 20-150 Hz, and the rest. The interval 3-15 Hz emphasises the speech syllabic rates. The interval 20-150 Hz is the perceptual roughness [18]. After this step, we obtain four values for each filter response.

2.3 Inner Hair Cell Model

Another method, which can be combined with the gamma-tone filters is the inner hair cell model of Med-

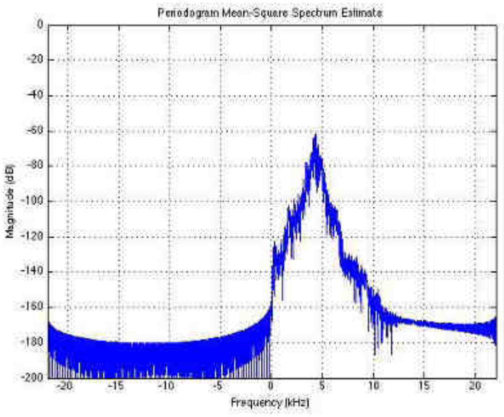


Figure 1: A single gamma-tone filter output transformed by Hilbert transform. The mean-squared spectrum of the signal is shown.

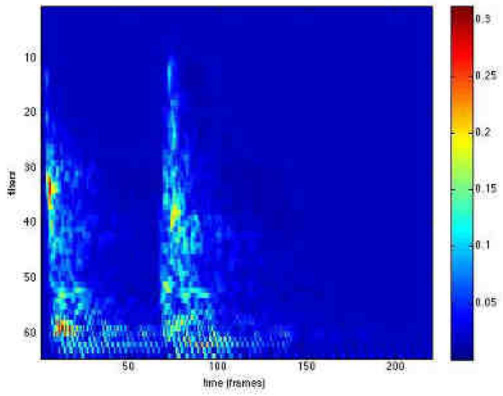


Figure 2: An example representation of gamma-tone filters combined with the inner hair cell model is shown. The sound of a closing door is analysed.

dis [8] [9] [10]. In this model, the firing rate of the inner hair cells, connected to the basilar membrane, is modelled. Hence, the gamma-tone filters and the inner hair cell model complement each other [16]. The inner hair cells fire, when a stimulus arrives and the basilar membrane is deflected at a point of a resonance frequency where the hair cell sits. This firing is simulated by the dynamics of production and flow of transmitter substance. A certain amount of transmitter substance is released into the synaptic cleft between the hair cell and another neuron, depending on the strength of the stimulus. For each arriving stimulus, the Meddis inner hair cell model calculates these amounts iteratively. In our representation, we use the rate of transmitted part of the transmitter substance. Figure 2 shows a closing door sound represented by gamma-tone filters combined with the inner hair cell model.

3 Feature Integration for SVM

3.1 Mean, Variance, Derivatives

The gamma-tone filter bank yields a multi-channel response for a single sound. Besides, the filter bank does not decrease the length of a given sound. Summing up the filter outputs reduces the dimensionality of the representation from 18 to 4 bands. The details of this method are given in the corresponding representation section. However, we compress the outputs of the bands across the entire length of the sound into a single representation vector. In order to integrate the responses of individual filters, we take the mean and variance of each of the values in the four bands. In order to be able to track the change in those values between different filters, we also calculate the mean and the variance of the first derivative of these filter responses [2]. After these calculations we obtain a 16 dimensional feature vectors for each sound.

3.2 SVM Settings

In these experiments, we used support vector machines [5], which are known to be one of the best classification tools, based on the maximisation of the margin of the classification boundary between two classes, c-SVM, which is implemented in the libsvm software library [3]. The c-SVM has two parameters (c and g), which should be determined beforehand. In order to find the optimal parameter values for these two parameters, we performed a grid search, where we changed these two parameters slightly to find optimal parameter settings. As grid values we use all combinations of $c = 2^8, 2^9, 2^{10}, \ldots, 2^{12}, g = 2^{-16}, 2^{-15}, 2^{-14}, \ldots, 2^{-4}$. For the final experiment in Table 5 we use a wider step size for the g parameter: $g = 2^{-16}, 2^{-14}, 2^{-12}, \ldots, 2^{-4}$ In the results, we will present the best results, which have been obtained by the optimal values of these two parameters.

4 Data Sets

Recordings are taken from the sound collection "Sound Ideas" [6]. For the experiments, recordings of footsteps and opening/closing doors are selected. The door sounds have a complex temporal structure whereas the steps are short and only consist of a few onsets. In both cases a mixture of temporal patterns and spectral properties can reveal information about material or function of the sound.

The doors dataset includes a series of recordings of wooden doors being closed or opened. A single sample contains either an opening or a closing door sound, not both one after another. 74 closing and 38 opening doors in total are used as input for the experiments.

The footstep dataset consists of recordings of different kind of shoes (heels, boots, barefoot) on various grounds (concrete, gravel, wood, dirt). They are taken from CD 16 and 17 from [6]. The recordings are cut, when necessary, such that exactly one step in contained in one sample. The limit of steps taken from the same recording is 10 to avoid overfitting effects. This leads to a dataset with 125 footsteps per class.

The datasets that we use in our experiments are listed out in the following:

- doors
  - opening - closing
- footsteps
  - high heels - boots
  - high heels - non high heels
  - barefoot - sneakers
  - on wood - on dirt

Besides performing binary classification experiments on these data sets, we evaluate our representation schemes on a multi-class classification experiment as well. In this experiment, we use five different types of footstep sounds (barefoot, sneakers, leather, heels, boots), all on concrete or marble.

The labels of these datasets are all psycoacoustically validated. We did not perform detailed psycoacoustical experiments, but we checked the sounds by listening them by ourselves. We discard sounds whose class cannot be identified while listening to them.

5 Experiments and Results

In order to evaluate the results of our experiments in a reliable way we use cross validation to calculate the average accuracy. In particular, we use the leave-one-out cross validation method. In leave-one-out cross validation experiments, all sounds but one are put into the training set. Then the trained algorithm is tested on the remaining single sample, that had been previously excluded from the training set. This procedure is repeated for all possible partitions into training/test set. The total accuracy of the system is the average accuracy of all tests.

The experiments that we perform are all binary classification experiments. The door sounds are classified as opening and closing sounds. Table 1 shows the results of SVM classification using four different representations: 1) gamma-tone filters combined with the Hilbert transform (GT Hil), 2) gamma-tone filters combined with the inner hair cell model (GT Med), 3) MFCC's, and 4) MFCC's combined with the low-level spectral features (SLL), which contain zero-crossings, roll-off and centroid.

GT Hil	GT Med	MFCC	SLL
84.0%	63.2%	60.4%	67.1%

Table 1: Classification of opening/closing door sounds by SVMs using various representations (cf. text).

The footstep sounds were classified based on the sole types (high-heels vs. boots, high-heels vs. not high-heels), and the floor (concrete vs. gravel, wood vs. dirt). Furthermore, we also designed a multi-class experiment based on the sole types as well. In the latter, we use five different sole type classes (barefoot, sneakers, leathers, heels, combat boots) on concrete floor.

In Table 2 the results of the binary classification experiment heels vs. combat boots are shown. All representation methods work almost perfect for this classification task. Only the MFCC's do not yield 100% accuracy for the task heels vs. combat boots, but 98.3% accuracy is still good. Based on these results, this task can be considered as the conceptual proof that our representation methods work at least as well as traditional methods. In order to make this task a little bit more complicated, so that we can observe differences between the accuracy rates of different representations, we perform another experiments, where we classify heels vs. non-heels. The results of this second experiment are shown in Table 3. Here, the gamma-tone based methods outperform the MFCC based methods.

GT Hil	GT Med	MFCC	SLL
100.0%	**100.0**%	98.3%	**100.0**%

Table 2: Classification results of the footstep sounds heels vs. boots are shown.

GT Hil	GT Med	MFCC	SLL
91.6%	**100.0**%	80.9%	89.2%

Table 3: Classification results of the footstep sounds heels vs. not heels are shown.

We also perform binary classification experiments by using different sole types on two different floor types, namely wood floor vs. dirt, leaves, sand and gravel. The second class of sounds contain four different floor types. However, all these floor types are perceived almost equal to the human listener. Therefore, creating a single class out of these sounds does not any harm to the experiment from a psychoacoustical point of view. Table 4 shows the classification results of this experiment. Again the accuracy is 100% for all different representation methods. Hence, these results prove that the representation methods we propose work well for general binary classification tasks.

GT Hil	GT Med	MFCC	SLL
100.0%	**100.0%**	**100.0%**	**100.0%**

Table 4: Classification results of the footstep sounds wood floor vs. dirt are shown.

Beside these binary classification tasks, we also perform a multi-class experiment. In this experiment, we use five different sole types, and classified each of those classes against the others. Each class consists of 40 instances. We perform five separate binary classification runs. In each, one class is classified against the rest, the other four classes. As a result, we obtain five binary classification results. In order to calculate the overall accuracy of the experiment, we take the average of these five separate classifications. Table 5 shows not only the overall average of the separate classification experiments, but also the results of the separate binary classification experiments themselves. In contrast to Table 3, the classification heels vs. non-heels in Table 5 uses a different set of samples. In the latter only steps on concrete are used.

	GT Hil	GT Med	MFCC	SLL
Barefoot	94.4%	**100.0%**	94.4%	94.4%
Sneakers	**94.4%**	83.3%	76.6%	82.1%
Leather	86.3%	**100.0%**	**100.0%**	**100.0%**
Heels	85.7%	92.8%	98.3%	**100.0%**
Boots	85.0%	**85.7%**	78.7%	79.3%
Average	92.2%	**94.2%**	88.8%	89.3%

Table 5: Classification accuracy of sole types in step sounds. Altogether, the gamma-tone based representations outperform the MFCC based methods.

6 Conclusion

We have investigated the potential of four representations for classifying everyday sounds, in particular opening/closing doors and the sole and floor material in footsteps. In each experiment, the best method performs with at least 84% accuracy, in several instances even with 100%.

In contrast to low-level descriptors (such as zero-crossing rate, spectral centroid, roll-off), sound can be pre-processed by models physiologically inspired by the basilar membrane. MFCC's and gamma-tone filter banks are prominent examples, both modelling the decreasing frequency resolution for high frequencies. In addition, gamma-tone filterbanks also model the low temporal resolution for low frequency signals and their impulse response closely resembles physiological measurements in the basilar membrane.

The experiments show that in general more physiologically relevant models, gamma-tone based representations, outperform other pre-processing methods. We combined the gamma-tone filters with the Hilbert transform and with Meddis' inner hair cell model. In order to reduce the dimensionality firstly we summarised the filter responses in four different frequency bands, and then applied a feature integration method to these representations. In the end, we obtained feature vectors, which can be used to perform classification experiments with support vector machines. Beside these two representation schemes, we performed classification experiments with MFCC's, and MFCC's combined with low-level spectral features. For the simple classification tasks they classified the sounds perfectly. These simple tasks are considered to be the proof of concept in general for these representation schemes.

However, we performed more complicated experiments, in order to observe the classification accuracy of the gamma-tone based representation compared to the MFCC based methods. Comparison of these experiments showed that, in general, gamma-tone based representation methods outperformed MFCC based representation methods. Although there are several special cases, where the MFCC's performed better than the gamma-tone based methods, gamma-tone representations yielded better results.

Interestingly, on the most complex data set, the door sounds, the gamma-tone / Hilbert transform method performed significantly better then the other methods (17% better than the second best method). On the other hand, the inner hair cell combination yielded slightly better results for the footstep sounds than the Hilbert transform.

The representations used here are basically stationary and include only little information about the temporal dynamics of the sounds. The MFCC's used here include variance, discrete first derivation, and the variance of the latter. Thereby, MFCC's momentarily capture some temporal behaviour. The gamma-tone filterbank gives better temporal resolution for higher frequencies than for lower frequencies. The Meddis hair cell model essentially emphasises on- and offsets. This feature may be the reason why a gammatone filterbank with subsequent Meddis hair cell model sometimes performs better than the Hilbert transform. Combining the Hilbert transform based representation with delta coefficients like the MFCC's or an onset detection feature like the Meddis hair cell could improve this representation. For a more complex consideration of the time course of the events a higher level analysis would be useful. Promising approaches include dynamic time warping, hidden Markov models, some sort of analysis of the rhythm (regularity, acceleration, deceleration) generated by the onsets of the signal.

References

[1] J. Breebaart and M. McKinney. Features for audio and music classification. In *International Conference on Music Information Retrieval*, 2003.

[2] J.J. Burred and A. Lerch. A hierarchical approach to automatic musical genre classification. In *International Conference on Digital Audio Effects*, 2003.

[3] C.C. Chang C.W. Su and C.J. Lin. *A Practical Guide to Support Vector Classification*. Department of Computer Science, National Taiwan University, 2007.

[4] B.R. Glasberg and B.C.J. Moore. Derivation of auditory filter shapes from notched-noise data. *Hearing Research*, 47:103–138, 1990.

[5] S. Haykin. *Neural Networks*. Prentice Hall, London, 2 edition, 1999.

[6] Sound ideas sound database, http://www.sound-ideas.com.

[7] B. Logan. Mel frequency cepstral coefficients for music modeling. In *International Symposium on Music Information Retrieval*, 2000.

[8] R. Meddis. Simulation of mechanical to neural transduction in the auditory receptor. *Journal of the Acoustical Society of America*, 79-3:702–711, 1986.

[9] R. Meddis. Simulation of auditory-neural transduction: Further studies. *Journal of the Acoustical Society of America*, 83-3:1056–1063, 1988.

[10] P.B. Ostergaard. Implementation details of a computation model of the inner hair-cell/auditory-nerve synapse. *Journal of the Acoustical Society of America*, 87-4:1813–1816, 1990.

[11] R.D. Patterson and J. Holdsworth. A functional model of neural activity patterns and auditory images. *Advances in Speech, Hearing and Language Processing*, 3:547–563, 1996.

[12] K. Robenson R.D. Patterson and J. Holdsworth. Complex sounds and auditory images. *In Proceesings of Auditory Physiology and Perception*, 9:429–446, 2004.

[13] M. Slaney. *An Eifficient Implementation of the Patterson-Holdsworth Auditory Filter Bank*. Apple Computer, 1993.

[14] M. Slaney. *A matlab toolbox for auditory modeling work*. Interval Research Corporation, 1998.

[15] J. O. Smith. *Spectral Audio Signal Processing*. W3K, Stanford, march 2006 draft edition, 2006.

[16] C. Spevak and R. Polfreman. Analysing auditory representations for sound classification with self-organizing neural networks. In *International Conference on Digital Audio Effects*, 2000.

[17] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transaction on speech and audio processing*, 10-5, 2002.

[18] E. Zwicker and H. Fastl. *Psychoacoustics: Facts and Models*. Springer-Verlag, Berlin Heidelberg NewYork London, 22 edition, 1990.

A Content-Based User-Feedback Driven Playlist Generator and its Evaluation in a Real-World Scenario

Martin Gasser[1], Elias Pampalk[2], and Martin Tomitsch[3]

[1]Austrian Research Institute for Artificial Intelligence (OFAI)
Freyung 6/6, A-1010 Vienna, Austria

[2] National Institute of Advanced Science and Technology (AIST)
IT, AIST, 1-1-1 Umezone Tsukuba, Ibaraki 305-8568, Japan

[3] Research Industrial Software Engineering (RISE)
Concorde Business Park 2A/F, Technology Center Schwechat, Austria

Abstract. The Simple Playlist Generator (SPG) is a software audio player with integrated playlist generation capabilities. It combines purely content-based playlist generation with an iterative playlist refinement process based on user-feedback. In this paper we briefly describe the system, and we present a systematic user evaluation of the software. Our results indicate that content-based playlist generation bears a lot of potential in the domain of personal music collections. Users very quickly adapted to the idea of playlists based on music similarity and were satisfied with the quality of automatically generated playlists.

1 Introduction

Web radios like last.fm or pandora.com are becoming increasingly popular because of their ability to automatically generate personalized playlists based on a small set of seed songs and user ratings. Those services use collaborative filtering approaches and manual annotations to calculate song-to-song similarities. Since personal music collection management applications often cannot rely on large metadata repositories, content-based music similarity is an alternative in this domain. Recently, many approaches to content-based playlist generation have been proposed (see e.g. [1, 2, 5]).

Simple Playlist Generator (SPG) integrates a conventional audio player with content-based playlist generation functionality. The basic idea is to facilitate user feedback to improve the quality of automatically generated playlists. Users can rate songs/artists positively or negatively and regenerate the playlist. The system keeps track of the ratings and adapts itself to the user's notion of what music should be contained in the playlist.

The software performed satisfactory on the authors' personal limited music collections; however we could not generalize the results of these tests, as the collections were already quite specialized to certain types of music, which eases the job of an automatic playlist generation system considerably. Therefore we also conducted an evaluation of the software with respect to usability and user acceptance in real-world scenarios.

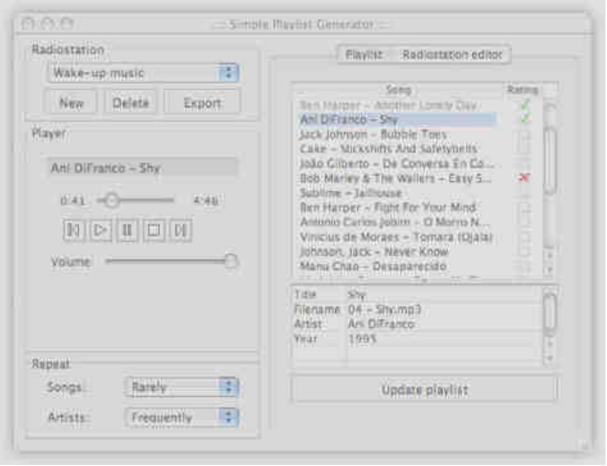
The study was carried out in three stages: (1) an online survey to define the potential target audience for playlist generation software, (2) face-to-face interviews and usability tests, and (3) a qualitative user evaluation of generated playlists.

2 Basic functionality and GUI

SPG combines content-based audio similarity, simple heuristics, and user feedback for playlist generation. The heuristics search for songs similar to positively rated songs/artists and avoid songs similar to negatively rated (banned) songs/artists. Thus, it is possible for users to "teach" the system what kind of music they want to listen to.

Similar to pandora.com, it is possible to define *radio stations* in SPG. A radio station is a named collection of song or artist ratings which is associated with a certain situational context (e.g. music for "Candle light dinner", "Workout",...) by the user. Users can store, reload, and edit radio stations at any point in time.

SPG was implemented in Java (figure 1 shows a screenshot of the application running on MacOS X). The list on the right side shows the current playlist which is generated by the selected radio station. The user can rate songs by clicking on the icons to the right of each song (this only has an impact on the current radio station). The playlist is updated by clicking on "update playlist". On the left side (top to bottom) are the radio station selector, audio player controls, and controls to adjust the variance of the playlist. The export button underneath the playlist selector allows the



Radiostation: Wake-up music

Player: Ani DiFranco – Shy

0:41 4:46

Song	Rating
Ben Harper – Another Lonely Day	✓
Ani DiFranco – Shy	✓
Jack Johnson – Bubble Toes	
Cake – Stickshifts And Safetybelts	
João Gilberto – De Conversa En Co...	
Bob Marley & The Wailers – Easy S...	✗
Sublime – Jailhouse	
Ben Harper – Fight For Your Mind	
Antonio Carlos Jobim – O Morro N...	
Vinicius de Moraes – Tomara (Ojalá)	
Johnson, Jack – Never Know	
Manu Chao – Desaparecido	

Title: Shy
Filename: 04 – Shy.mp3
Artist: Ani DiFranco
Year: 1995

Repeat
Songs: Rarely
Artists: Frequently

Figure 1: Screenshot of the SPG user interface

users to export the current playlist to an M3U file (and thus allows them to use their preferred audio player).

The controls to adjust the variance allow the user to set the number of times artists and songs are repeated to either “rarely”, “sometimes”, or “frequently”. If, for example, the user chooses to repeat songs rarely, then SPG would rather play a not so similar song instead of playing a previously played song again. If the user chooses to repeat artists frequently, then SPG (if appropriate) might play a number of songs by the same artist within an hour. The user can only set the repetition frequency of artists to the level that was set for songs or below (e.g. repeating songs frequently, but artists rarely is not a valid setting).

Not shown is the edit radio station panel where the users can view their ratings and change the name of the radio station. One list is shown for all rated songs, and one list for all rated artists. New items can be added using a simple search function. Ratings in the lists can be changed in the same way they are changed in the playlist.

3 Music similarity measure

Our audio-based music similarity measure is based on distances between statistical distributions of low-level audio features for each pair of songs in the music collection (see figure 2 for a sketch of how the components of the system interact). For each song, we (1) calculate Mel Frequency Cepstral Coefficients and train a single Gaussian with full covariance to the sequence of MFCC frames [6] and (2) extract a set of fluctuation patterns [5] which capture periodicities in subbands of the audio signal. Finally, a similarity measure is calculated by combining a symmetric version of the Kullback-Leibler divergence between Gaussians and the Euclidean distance between fluctuation patterns.

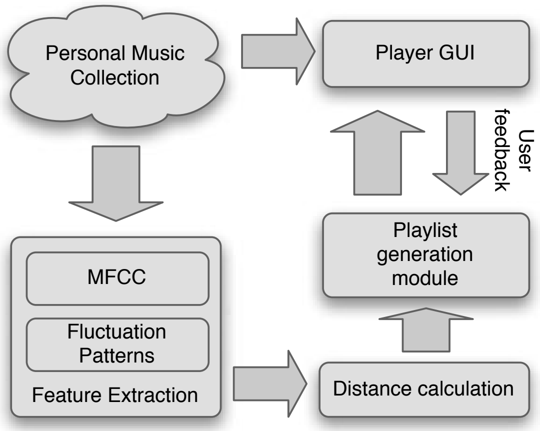


Figure 2: Components in the SPG system.

Currently, the music similarity calculation is implemented as a Matlab module and has to be done offline. However, we are working on a high performance implementation that can be deployed in consumer level applications.

4 Playlist generation

To create playlists, we use a variant of heuristic D described in [4]. Basically, songs which are close to any of the positively rated songs, and far away from negatively rated songs are recommended. If the user dislikes or likes an artist, then all songs from this artist (unless they are rated individually) are treated as favorite or banned songs.

Algorithm 1 sketches one step of the playlist generation heuristic in pseudo–code. First, we give an explanation of the symbols used in the code:

S_{pos}/S_{neg} : Set of positively/negatively rated songs

A_{pos}/A_{neg} : Set of positively/negatively rated artists

S_{cand} : Set of candidate songs

$songs(A)$: Finds all songs of artist A

$\Delta(s_1, s_2)$: Distance from song s_1 to song s_2

$closest\_song(S, a)$: Finds $s \in S$ with minimal $\Delta(s, a)$

The system always keeps track of which songs have been played in the past. This knowledge and the user’s variance preferences in the SPG user interface are used to select a subset of candidate songs from the entire music collection. The set of candidate songs is then used as input to the heuristic, which returns the next song in the playlist.

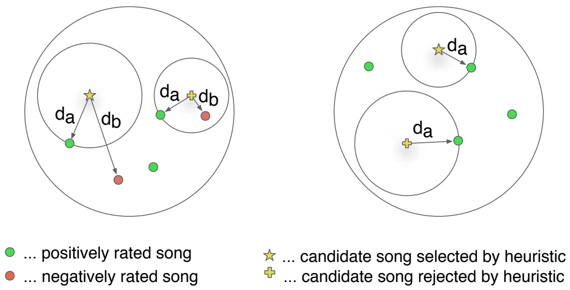

da
db
da
db
da
da
● ... positively rated song
● ... negatively rated song
☆ ... candidate song selected by heuristic
✚ ... candidate song rejected by heuristic

Figure 3: Playlist generation heuristic.

Figure 3 illustrates lines 4–10 and 11–15 of the algorithm. On the left, we can see that only candidate songs which satisfy the condition $d_a \leq d_b$ remain in the candidate set. On the right side, the song with the smallest d_a distance is selected.

Algorithm 1 Playlist generation algorithm

```
1:  $A \leftarrow S_{pos} \cup (songs(A_{pos}) - S_{neg})$
2:  $B \leftarrow S_{neg} \cup (songs(A_{neg}) - S_{pos})$
3:  $C \leftarrow S_{cand} - B$
4:  for all s in C do
5:      $d_a(s) \leftarrow \Delta(s, closest\_song(A, s))$
6:      $d_b(s) \leftarrow \Delta(s, closest\_song(B, s))$
7:      if $d_b(s) < d_a(s)$ then
8:          $C \leftarrow C - s$
9:      end if
10: end for
11: if empty(C) then
12:     return $argmin(d_a(s)/d_b(s))$
13: else
14:     return $argmin(d_a(s))$
15: end if
```

5 Evaluation

The evaluation of SPG was done in three steps which will be described below.

5.1 Target audience

The first step of the evaluation was the definition of the target audience for systems like SPG. In contrast to the study presented in [3] we first explored the characteristics of the target group by carrying out an anonymous web questionnaire in which we asked users about demographical data, their music listening habits, and if they would like to participate in a user study. We presumed that users who satisfy the following requirements are likely to be interested in automatic playlist generation software: the user (1) owns a large music collection (more than 1.000 files), (2) owns different styles of music, and (3) does not want to spend too much time on creating playlists manually.

From 53 participants (23 female, 30 male, average age 27, SD=3.7), 35 showed interest in using our tool on their music collection, whereas only 19 indicated willingness to participate in a user study. From the 35 interested participants, 7 stated that they use song rating capabilities in music players. Nobody from the group of uninterested participants used rating capabilities. Interested participants owned on average 20.000 music files, whereas uninterested people had on average 4.000 files. Twenty-seven of the interested participants declared that they use random shuffle mode at least rarely, 14 of the uninterested participants use random shuffle at least rarely.

We asked interested subjects to explain why they are interested; most people answered that they would use playlist generation whenever they do not have enough time or no chance to select songs manually (during parties, during work, fitness training, cleaning their flat). Some also stated that they would use it for finding new songs or finding other songs that match their current mood.

We also asked uninterested participants to give reasons for their lack of interest. From 18 uninterested participants, 3 preferred to listen to the radio, 3 had not enough MP3s, 3 preferred random shuffle, and 3 had a too homogenous music collection. Two stated that automatic playlist generation does not make sense for the type of music they listen to (i.e., classical music), and 1 did not listen to music on the computer at all. Two subjects also declared that they prefer to select the songs in their playlists individually.

5.2 Usability considerations

In the second stage, we conducted interviews with potential users of our system. In a series of interviews, we collected data about the socio-cultural background, and the music listening habits of 11 subjects (average age 28, STD=4.6). Subsequently, the subjects were asked to complete 10 predetermined tasks (e.g. creating a specific radio station, rating songs, excluding an artist, etc.). The interviewers measured the elapsed time upon completion of the tasks, and took notes about if the user completed the tasks without help. Based on the notes from the interviews, we decided whether subjects fall into the target group that has been defined in phase 1. All subjects had a decent background in using computers. On average, their musical taste can be described as “alternative mainstream”.

All subjects experienced difficulties with the first task, in which they had to define a radio station by specifying a seed song. This task involved most parts of the user interface. After successful completion users therefore were able to finish all the remaining tasks without or with very little assistance.

In general, users understood the radio station based concept of playlist generation.All subjects stated that they would probably use such an application, but that they would like to use it with their own music collection to be able to better judge the quality of the playlist. We observed that most users showed considerably more interest in using SPG after finishing the usability test.

5.3 Playlist quality

In the last phase of the evaluation, we invited 5 interested users who matched the target audience definition to use our software on a subset of their music collection. They were asked to define several distinct radio stations by providing seed songs/artists, excluding songs/artists they disliked, and altering the variance of the playlist (i.e., the repetition frequency of songs/artists). After giving the system some feedback, we asked the users to judge the quality of the generated playlists.

The overall user feedback was very positive. In general, users not familiar with the current state of the art in MIR research were surprised at how well content-based methods work. However, one user complained about the playlists *"being boring because the songs either sound all the same or are too arbitrary"*. Another user was actively addressing this issue, thinking about *"meta-playlists"*, that is, a specification of which styles of music can be played in what order, and how long sub-playlists should be.

Currently, SPG allows the user to control the variance of playlists by setting the repetition frequency of artists or songs to one of the predefined levels *rarely*, *sometimes*, or *frequently*. Most users seemed to understand this concept and how to use it to adapt radio stations immediately, however, one user would have preferred a more fine-grained scale to control the variance.

A common observation was that a decent amount of songs from different styles is required for SPG to provide meaningful results. One user requested to be informed by the system as to how many pieces of the collection are covered by the current radio station settings.

All users noted that the quality of the playlists depended very much on the seed songs they selected. One of the reasons for this was that in some cases there were only few songs in the collection that would have been suitable for the playlist defined by the seed song. Although the overall quality was quite good in average, any further improvements are very desirable.

6 Conclusions and future work

In this paper, we presented an implementation of a hybrid (content- and user feedback-based) approach to playlist generation, and we also carried out a user evaluation of the system. For the evaluation, we followed a systematic approach that involved an online survey to define the target audience of the software, face-to-face interviews to select test subjects based on the previously defined target group, and an evaluation of the software in a real-world scenario. Although the current user interface is far from perfect, the user acceptance was quite high.

For the future, we are considering an implementation of SPG technology as a plugin for a popular media player, based on the findings of the here presented study.

Acknowledgements: This work was partly funded by the WWTF project Interfaces to Music (I2M).

References

[1] T. Pohle, E. Pampalk, and G. Widmer, "Generating Similarity-Based Playlists Using Traveling Salesman Algorithms," in *Proc. of the Intl. Conf. on Digital Audio Effects*, 2005.

[2] M. Goto and T. Goto, "Musicream: New Music Playback Interface for Streaming, Sticking, Sorting, and Recalling Musical Pieces," in *Proc. of the ISMIR Intl. Conf. on Music Information Retrieval*, 2005.

[3] F. Vignoli, "Digital Music Interaction Concepts: A User Study.," in *Proc. of the ISMIR Intl. Conf. on Music Information Retrieval*, 2004.

[4] E. Pampalk, T. Pohle, and G. Widmer, "Dynamic Playlist Generation Based on Skipping Behaviour," in *Proc. of the ISMIR Intl. Conf. on Music Information Retrieval*, 2005.

[5] E. Pampalk, "Computational Models of Music Similarity and their Application in Music Information Retrieval," Ph.D. dissertation, Vienna University of Technology, 2006.

[6] M. Mandel and D. Ellis, "Song-Level Features and Support Vector Machines for Music Classification," in *Proc. of the ISMIR Intl. Conf. on Music Information Retrieval*, 2005.

[7] E. Pampalk and M. Gasser, "An Implementation of a Simple Playlist Generator
Based on Audio Similarity Measures and User Feedback," in *Proc. of the ISMIR Intl. Conf. on Music Information Retrieval*, 2006.

Indexing and retrieval scheme for content-based search in audio databases

P. Kukharchik, D. Martynov, I. Kheidorov
Belarusian State University
dimamrt@tut.by, ikheidorov@sacrament.com.

Abstract. Rapid increase in the amount of the digital audio collections demands a generic framework for robust and efficient indexing and retrieval based on the aural content. In this paper we focus our efforts on developing a generic and robust audio-based multimedia indexing and retrieval framework. First an overview for the audio indexing and retrieval schemes with the major limitations and drawbacks are presented. Then the basic innovative properties of the proposed method are justified accordingly. Finally the experimental results and conclusive remarks about the proposed scheme are reported.

1. Introduction

Rapid increase in the amount of the digital audio collections presenting various formats, types, durations, and other parameters that the digital multimedia world refers, demands a generic framework for robust and efficient indexing and retrieval based on the aural content. From the content-based multimedia retrieval point of view the audio information is mostly unique and significantly stable within the entire duration of the content and therefore, the audio can be a promising part for the content-based management for those multimedia collections accompanied with an audio track. Traditional key-word based search engines usually cannot provide successful audio retrievals since they usually require manual annotations that are obviously unpractical for large multimedia collections.

The usual approach for indexing is to map database primitives into some high dimensional vector space that is so called feature domain. Among many variations, careful selection of the feature sets allows capturing the semantics of the database items. The number of features extracted from the raw data is often kept large due to the naive expectation that it helps to capture the semantics better. Content-based similarity between two database items can then be assumed to correspond to the (dis-) similarity distance of their feature vectors. Henceforth, the retrieval of a similar audio items with respect to a given query (item) can be transformed into the problem of finding such audeo items that gives feature vectors, which are close to the query feature vector. This is QBE. QBE is costly and CPU intensive especially for large-scale multimedia databases since the number of similarity distance calculations is proportional with the database size. This fact brought a need for indexing techniques, which will organize the database structure in such a way that the query time and I/O access amount could be reduced [1, 2].

There are some techniques to speed up QBE but all they may not provide efficient retrieval scheme from the user’s point of view due to their strict parameter dependency. All QBE alternatives have some common drawbacks. First of all, the user has to wait until all of the similarity distances are calculated and the searched database items are ranked accordingly. This might take a significant time if the database size is large. In order to speed up the query process, it is a common application design procedure to hold all features of database items into the system memory first and then perform the calculations. Therefore, the growth in the size of the database and the set of features will not only increase the query time but it might also increase the minimum system memory requirements such as memory capacity and CPU power [3].

All the systems based on these techniques achieved a certain performance; however present some more limitations and drawbacks. All techniques are designed to work in pre-fixed audio parameters. It is a fact that the aural content is totally independent from such parameters. And, they are mostly designed either for short sound files bearing a unique content or manually selected (short) sections. However, in a multimedia database, each clip can contain multiple content types, which are temporally (and also spatially) mixed with indefinite durations. Even the same content type (i.e. speech or music) may be produced by different sources (people, instruments, etc.) and should therefore, be analyzed accordingly.

For the past three decades, researchers proposed several indexing techniques that are formed mostly in a hierarchical tree structure that is used to cluster (or partition) the feature space. Initial attempts such as KD-Trees [4] and R-tree [5] are the first examples of Spatial Access Methods (SAMs). But, especially for content-based indexing and retrieval in large-scale multimedia databases, SAMs have several drawbacks and significant limitations. By definition an SAM-based indexing scheme partitions and works over a single feature space. However, a multimedia database can have several feature types, each of which might also have multiple feature subsets. In order to provide a more general approach to similarity indexing for multimedia databases, several efficient Metric Access Methods (MAMs) are proposed. The generality of MAMs comes from the fact that any MAM employs the indexing process by assuming only the availability of a similarity distance function, which satisfies three trivial rules: symmetry, non-negativity and triangular inequality. Therefore, a multimedia database might have several feature types along with various numbers of feature sub-sets all of which are in different multi-dimensional feature spaces. The MAMs so far addressed present several shortcomings. Contrary to SAMs, these metric trees are designed only to reduce the number of similarity distance computations, paying no attention to I/O costs (disk page accesses). They are also intrinsically static methods in the sense that the tree structure is built once and new insertions are not supported. Furthermore, all of them build the indexing structure from top to bottom and hence the resulting tree is not guaranteed to be balanced.

As a summary, the indexing structures so far addressed are all designed to speed up any QBE process by using some multidimensional index structure. However, all of them have significant drawbacks for the indexing of large-scale multimedia databases.

2. New approaches

2.1. Indexing scheme: cellular tree

To improve the retrieval feasibility and efficiency audio databases need to be indexed in some way and traditional methods are no longer adequate. It is clear that the nature of the search mechanism is influenced heavily by the underlying architecture and indexing system employed by the database. Therefore, we present a novel indexing technique, *Cellular Tree (CT)*, which is designed to bring an effective solution especially for indexing multimedia databases.

CT is a dynamic, cell–based and hierarchically structured indexing method, which is purposefully designed for query operations and advanced browsing capabilities within large-scale multimedia databases. It is mainly a hierarchical clustering method where the items are partitioned depending on their relative distances and stored within cells on the basis of their similarity proximity. The similarity distance function implementation is a *black-box* for the *CT*. Furthermore, *CT* is a self-organized tree, which is implemented by genetic programming principles. This basically means that the operations are not externally controlled; instead each operation such as item insertion, removal, mitosis, etc. are carried out according to some internal rules within a certain level and their outcomes may uncontrollably initiate some other operations on the other levels. Yet all such "reactions" are bound to end up in a limited time, that is, for any action (i.e. an item insertion), its consequent reactions cannot last indefinitely due to the fact that each of them can occur only in a higher level and any *CT* body has naturally limited number of levels.

2.1.1. Cell Structure

A cell is the basic container structure in which similar database items are stored. Ground level cells contain the entire database items. Each cell further carries a Minimum Spanning Tree (MST) where the items are spanned via its nodes. This internal MST stores the minimum (dis-)similarity distance of each individual item to the rest of the items in the cell. All cell items are used as vantage points for any (other) cell item. These item-cell distance statistics are mainly used to extract the cell compactness feature. In this way we can have a better idea about the similarity proximity of any item instead of comparing it only with a single item (i.e. the cell nucleus) and hence a better compactness feature. The compactness feature calculation is also a black-box implementation and we use a regularization function obtained from the statistical analysis using the MST and some cell data. This dynamic feature can then be used to decide whether or not to perform mitosis within a cell at any instant. If permission for mitosis is granted, the MST is again used to decide where the partition should occur and the longest branch is a natural choice. Thus an optimum decision can be made to enhance the overall compactness of the cell with no additional computational cost. Furthermore, the MST is used to find out an optimum nucleus item after any operation is completed within the cell. In *CT*, the cell size is kept flexible, which means there is no fixed cell size that cannot be exceeded. However, there is a maturity concept for the cells in order to prevent a mitosis operation before the cell reaches a certain level of maturity. Therefore, using a similar argument for the organic cells, a maturity cell size (e.g. 6) is set for all the cells in *CT* body (level independent)

2.1.2. Level structure

HCT body is hierarchically partitioned among one or more levels, as one sample example shown in Figure 1. Apart from the top level, each level contains various numbers of cells that are created by mitosis operations occurring at that level. The top level contains a single cell and when this cell splits, and then a new cell is created at the level above. The nucleus item of each cell in a particular level is represented on the higher level.

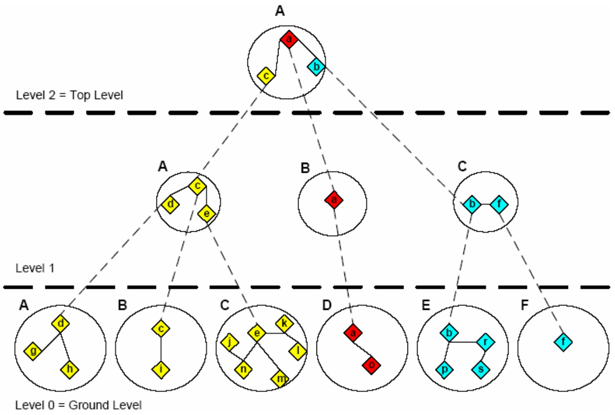


Figure 1. A Sample 3-level CT body

Each level dynamically tries to maximize the compactness of their cells although this is not a straightforward process to do since the incoming items may not show a similarity to the items present in the cells and therefore, such dissimilar item insertions will cause a temporary degradation on the overall (average) compactness of the level. So each level, while analyzing the effects of the (recent) incoming items on the overall level compactness, should employ necessary management steps towards improving compactness in due time (i.e. with future insertions). Within a period of time (i.e. during a number of insertions or after some number of mitosis), each level updates its compactness threshold according to the compactness feature statistics of mature cells, into which an item was inserted.

2.1.3. CT Operations

There are mainly three *CT* operations: cell mitosis, item insertion and removal. Cell mitosis can only happen after any of the other two *CT* operations occurs. Both item insertion and removal are generic *HCT* operations that are identical for any level. Insertions should be performed one item at a time. However, item removals can be performed on a cell-based, i.e., any number of items in the same cell can be removed simultaneously.

By means of the proposed dynamic insertion technique, the MST is initially used and updated only whenever necessary. A sample dynamic insertion operation is illustrated in Figure 2.

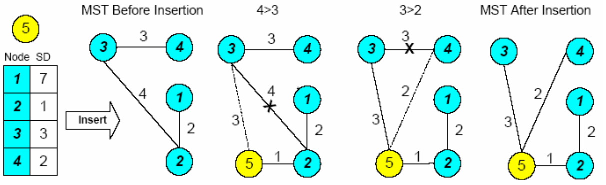


Figure 2. A sample dynamic item (5) insertion into a 4-node MST

Due to the presence of MST within each cell, mitosis has no computational cost in terms of similarity distance calculations. The cell is simply split by breaking the longest branch in MST and each of the newborn child cells is formed using one of the MST partitions.

2.2. Query

In order to eliminate drawbacks mentioned above and provide a faster query scheme, a novel retrieval scheme, *Progressive Query*, was developed. It is a retrieval (via QBE) technique, which can be performed over the databases with or without the presence of an indexing structure. Scheme provides intermediate query results during the query process. The user may browse these results and may stop the ongoing query in case the results obtained so far are satisfactory and hence no further time should unnecessarily be wasted, so it may perform the overall query process faster (within a shorter total query time).

The principal idea behind the new design is to partition the database items into some subsets within which individual (sub-)queries can be performed. Therefore, a sub-query is a fractional query process that is performed over any sub-set of database items. Once a sub-query is completed over a particular sub-set, the incremental retrieval results (belonging only to that sub-set) should be fused (merged) with the last overall retrieval result to obtain a new overall retrieval result, which belongs to the items where query operation so far covers from the beginning of the operation. The order of the database items processed is a matter of the indexing structure of the database. If the database is not indexed at all, simply a sequential or random order can be chosen. In case the database has an indexing structure, a *query path* can be formed in order to retrieve the most relevant items at the beginning during a query operation.

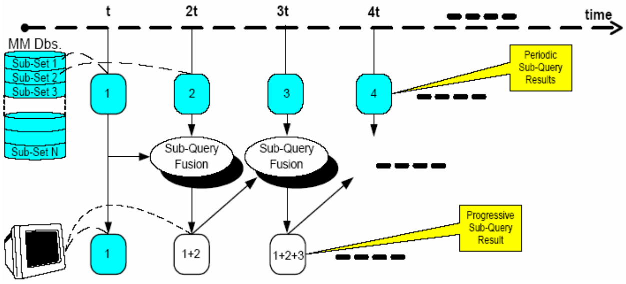


Figure 3. Progressive Query Overview

Obviously, *query path* is nothing but a special sequence of the database items, and when the database lacks an indexing structure, it can be formed in any convenient way such as sequentially or randomly. Otherwise, the most advantageous way to perform query is to use the indexing information so that the most relevant items can be retrieved in earlier sub-query steps.

Query operation over *CT* is executed synchronously over two parallel processes: *CT tracer* and a generic process for sub-query formation using the latest *query path* segment. *CT tracer* is a recursive algorithm, which traces among the *CT* levels in order to form a *query path* (segment) for the next *sub-query* update.

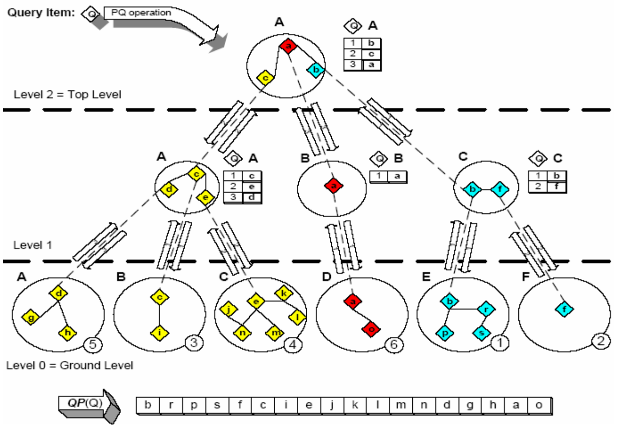


Figure 4. Q*uery path* formation on a sample *HCT* body

3. Experiments

Like it was mentioned before the similarity distance function, all decision rules are “black boxes” for our indexing scheme. So it is possible to work with any type of multimedia data. In our work we made an indexing scheme for dataset that contains audio records with music.

3.1. Feature extraction

Feature vectors were formed on the basis of wavelets. The idea to build feature vector on wavelets for audio classification was previously reported by Li et al [9] and Tzanetakis et al in [10]. These authors used discrete wavelet transform (DWT) coefficients for their method of feature extraction for content-based audio classification. Compared to DWT, continuous wavelet transform allows much more flexibility due to its arbitrary time-scale resolution. Experimental results show that features based on continuous wavelet and wavelet-like transforms can serve as one of the basic features for music classification.

A window of constant dyadic length is applied to input audio data picking out an audio segment. Then continuous wavelet or pseudo-wavelet transform is applied to this audio segment. The result of this transform is time-frequency representation of the given signal. Despite there are many ways to extract information from CWT representation of a signal, at the beginning of our study of CWT-based features we decided to limit ourselves by a simplest way of feature extraction from CWT of a signal.

We used reduction of wavelet information by averaging of neighbouring wavelet coefficients on time-frequency plane.

The whole time-frequency plane is cut into subbands along the scale axis and subsegments along the time axis. The width of subbands and subsegments is equal and this results in uniform tiling of CWT time-frequency plane. Then the procedure averages all the coefficients in every tile resulting in one mean value for every tile. These mean values form a feature vector (fig. 5).

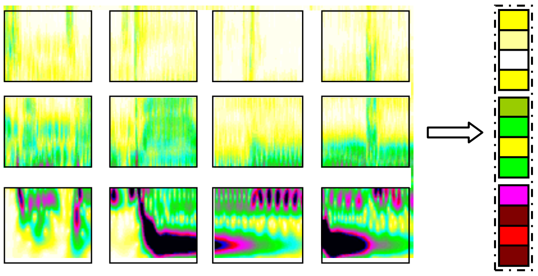


Figure 5. Feature vector building scheme

3.2. Experimental results

The main point of our experimental researches was to compare new Progressive Query with normal Query by Example (NQ) when first one is working over Cellular Tree indexing scheme. All experiments are carried out on an Athlon64 2800+computer with 1024 MB memory.

The first difference is that if *NQ* is chosen, then the user has to wait till the whole process is completed but if *PQ* is chosen then the retrieval results will be updated periodically (with the user-defined period value) each time a new sub-query is accomplished.

PQ and *NQ* eventually converge to the same retrieval result at the end. Also in the abovementioned scenarios they are both designed to perform exhaustive search over the entire database. However *PQ* has several advantages over *NQ* in the following aspects:

System Memory Requirement: The memory requirement is proportional to the database size and the number of features present in a *NQ* operation. Due to the partitioning of the database into sub-sets, *PQ* will reduce the memory requirement by the number of *sub-query* operations performed.

"Earlier and Better" Retrieval Results: Along with the ongoing process *PQ* allows intermediate query results (*PSQ* steps), which might sometimes show equal or 'even better' performance than the final (overall) retrieval result. This is obviously an advantage for *PQ* since it proceeds within sub-queries performed in (smaller) sub-sets whereas *NQ* always has to proceed through the entire database.

Query Accessibility: This is the major advantage that *PQ* provides. Stopping an ongoing query operation is an important capability in the user point of view. The user can stop it any time (i.e. when the results are so far satisfactory).

Overall Retrieval Time (Query Speed): The overall query time is the time elapsed from the beginning of the query to the end of the operation. For *PQ,* since the retrieval is a continuous process with *PSQ* series, the *overall retrieval* means that *PQ* proceeds over the entire database and its process is finally completed. As mentioned earlier, at this point both *PQ* and *NQ* should generate identical retrieval results for a particular queried item. If *PQ* is completed with only one sub-query, then it basically performs a *NQ* operation. As experimentally verified, *PQ's* overall retrieval time is 0-20% faster than *NQ* retrievals (depending on the number of sub-query series) if *NQ* memory requirement does not exceed the system memory.

4. Conclusions

In this paper we focused our efforts on developing a generic and robust audio-based multimedia indexing and retrieval scheme. Indexing structure – Cellular Tree – is a dynamic, parameter independent and flexible cell (node) sized indexing structure, which is optimized to achieve as many focused cells as possible using aural descriptors with limited discrimination factors. Retrieval scheme – Progressive Query – an efficient retrieval technique (via QBE), which works with both indexed and non-indexed databases, it is the unique query method which may provide "faster" retrievals and provides "Browsing" capability between instances (sub-queries) of the ongoing query. *CT* is particularly designed to work with *PQ* in order to provide the earliest possible retrievals of relevant items. Since the ultimate measure of any system performance is the satisfaction of the system user, the most important property achieved is therefore its continuous user interaction, which provides a solid control and enhanced relevance feedback mechanism along with the ongoing query operation. The experiments demonstrate the superior performance achieved by *PQ* over *CT* in terms of speed, minimum system requirements, user interaction and possibility of better retrievals as compared with the traditional query scheme.

References

[1] E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based Classification, Search, and Retrieval of Audio", *IEEE Multimedia Magazine,* pp. 27-36, Fall 1996.

[2] J. T. Foote, "Content-Based Retrieval of Music and Audio", *in Proc. of SPIE, vol. 3229*, pp. 138-147, 1997.

[3] F.A. Cheikh, B. Cramariuc, M. Gabbouj, "Relevance feedback for shape query refinement", *In Proc. of IEEE International Conference on Image Processing, ICIP 2003*, Barcelona, Spain, 14-17 September 2003.

[4] J. L. Bentley, "Multidimensional binary search trees used for associative searching", *In Proc. of Communications of the ACM*, v.18 n.9, pp.509-517, September 1975.

[5] A. Guttman, "R-trees: a dynamic index structure for spatial searching", *In Proc. Of ACM SIGMOD*, pp. 47-57, 1984.

[6] R.L. Graham and O. Hell, "On the history of the minimum spanning tree problem," *Annual Hist. Comput.* 7, pp. 43-57. 1985.

[7] J. K. Uhlmann, "Satisfying General Proximity/Similarity Queries with Metric Trees", *Information Processing Letters*, vol. 40, pp. 175-179, 1991.

[8] T. Zhang and C.–C. J. Kuo, "Hierarchical Classification of Audio Data for Archiving and Retrieving", *In Proc. of IEEE Int. Conf. on Acoustics, Speech, Signal Proc.*, pp. 3001-3004, Phoenix. March 1999.

[9] T. Li, M. Oginara and Q. Li, "A comparative study on content-based music genre classification," in Proc. of the 26th annual int. ACM SIGIR conf.on Research and development in information retrieval, pp. 282–289. ACM, ACM Press, July 2003.

[10] G. Tzanetakis, and P. Cook "Musical genre classification of audio signals", IEEE Trans. on Speech and Audio Processing, vol. 10, no. 5, July 2002, pp. 293 – 302.

TANGA - an Interactive Object-Based Real Time Audio Engine

Ulrich Reiter, Inst. Media Technology, TU Ilmenau, D-98693 Ilmenau, ulrich.reiter@tu-ilmenau.de

Abstract. This paper describes the TANGA interactive modular real time audio engine originally developed for use in the I3D MPEG-4 player software. It details the engine's Component concept, presents the room acoustic simulation methods currently implemented, and introduces its multi threading capabilities. Furthermore, it briefly outlines mechanisms that can be used to provide interactivity in audiovisual scenes.

1 Introduction

At the Institute of Media Technology at Technische Universität Ilmenau, we have designed an interactive MPEG-4 player ('I3D') capable of rendering audiovisual scenes. The system can display three-dimensional virtual scenes with embedded 2D video streams of arbitrary shape. Both 2D (CRT monitors, TFTs, projectors, etc.) as well as 3D (shutter glasses, autostereoscopic) displays can be driven. The I3D allows for interaction with the user and it provides a modular audio engine called TANGA for real time rendering of (room) acoustic simulations and effects. This range of capabilities makes it unique among the MPEG-4 players available today.

TANGA is the most advanced audio rendering engine available in an MPEG-4 player. It has been developed to allow a coherent presentation of interactive audiovisual content. The modular nature of the engine allows adding new functionality by simply interconnecting already existing audio signal processing plugins (so-called Components) in a new way or by adding (programming in C++) new Components and incorporating them into the existing signal processing structure. This system of Components is highly flexible and very well suited for interactive, object-based applications in general. Section 2 details the Component Concept implemented in the TANGA.

TANGA can provide the user with an acoustic impression correlating to a 3D scenery presented visually. Therefore, a number of reverberation algorithms have already been implemented. These can be performed and influenced in real time to make interactive applications possible. Section 3 introduces the currently available options.

TANGA is also the first audio engine to be explicitly multi-thread and multi-core capable. It provides means of optimally distributing the computational load caused by the audio signal processing by applying newly developed heuristics. It can handle an arbitrary number of cores/processors and is therefore well suited for future CPU/PC developments. Some details are presented in section 4.

Parameters of the signal processing Components can be controlled from within a 3D audiovisual scene when the TANGA engine is used as part of the I3D MPEG-4 player. The audiovisual scene in turn can be influenced interactively via MIDI or via so-called events. These can have a time-related, object-related or Boolean logic driven origin. This offers a wealth of interactivity options. Section 5 very briefly introduces the use of the I3D as part of an evaluation system for subjective assessments of perceived audiovisual quality.

2 The Component Concept

TANGA is a tongue-in-cheek acronym for *The Advanced Next Generation Audio.* It is an object-oriented, platform independent, and modular software framework used to model real time audio signal processes in C++. TANGA is designed to allow an abstract representation of these processes. It encapsulates the underlying signal processing, thus providing a uniform, relatively simple to use interface to its functionality. A major incentive for using a software framework usually consists in the straightforward reusability of software components. This concept is consistently applied in the TANGA.

By means of using interfaces, additional functionality can be integrated, and new signal processing algorithms can be realized easily. In the TANGA system, the processing of audio signals is done through so-called TANGA Components. Each TANGA Component constitutes a signal processing unit with a given number of input and output channels. The audio signal arriving at the input is transformed by the signal processing logic implemented in the Component. It is then available (e.g. for further processing in other Components) at the output of the Component. So-called *ComponentConnectors* provide signal connections between Components. *MessageDispatchers* provide control message connections between Components. The

TANGA System can be expanded to virtually any audio functionality by means of writing a new 'plug-in' (read: Component). Because of this, TANGA is a powerful yet flexible software system.

In the TANGA system, the hardware (sound card) related parts are strictly separated from those related to the functionality (signal processing algorithm). Therefore it is divided into four entities: The Hardware, the Host, the Engine and the Signal Processing modules.

2.1 TANGA Hardware Module

The hardware module uses the sound card drivers registered in the system for audio input and output to any multichannel sound card. As it is linked to the host, the host has access to the hardware via the drivers. Functionality such as opening a driver and obtaining information on the current hardware setup is provided. The TANGA host module supports various drivers and is cross platform compatible. The performance of a single sound card can vary for different drivers, depending on the quality of the driver implementation. The drivers supported are ASIO, WMM and DirectSound on Windows and ALSA on Linux systems.

Upon the first start of the I3D, the hardware is instructed to describe the capabilities of the sound card (number of input and output channels available, drivers available in the system, and available sample rates). The user must choose from and set these parameters, which are then registered with the system.

2.2 TANGA Host Module

The host module implements the bridge between the hardware and the TANGA Engine. It consists of the PortAudio Application Programming Interface (API) [1]. Communication between hardware and host is realized via the driver.

One of the most important requirements for the audio API used in the TANGA System is that it should provide a DAC output time stamp. This should be the time when the samples being buffered are actually played at the audio output of the sound card. This feature is essential for synchronization purposes. PortAudio was chosen because it provides such a time stamp and has in general a very good support for real time operations. Whereas PortAudio also provides audio streams in blocking read / write mode, this feature is not useful for the TANGA System. TANGA relies on the non-blocking audio streams which use a callback method for filling the output buffers. The callback function invoked by PortAudio is used to control the TANGA Engine. PortAudio ensures that this function is always called in time such that the hardware output buffers are filled as needed by the sound card to continuously output the correct audio signal.

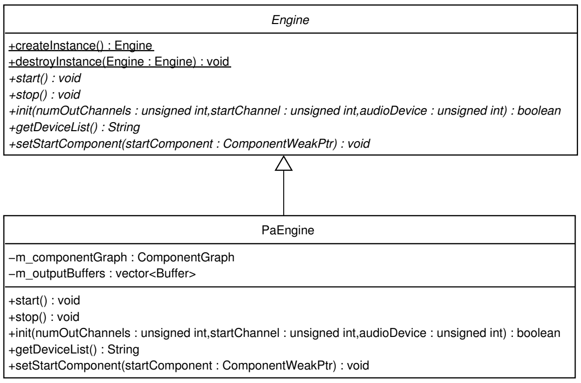

Engine
+createInstance() : Engine
+destroyInstance(Engine : Engine) : void
+start() : void
+stop() : void
+init(numOutChannels : unsigned int,startChannel : unsigned int,audioDevice : unsigned int) : boolean
+getDeviceList() : String
+setStartComponent(startComponent : ComponentWeakPtr) : void
PaEngine
−m_componentGraph : ComponentGraph
−m_outputBuffers : vector<Buffer>
+start() : void
+stop() : void
+init(numOutChannels : unsigned int,startChannel : unsigned int,audioDevice : unsigned int) : boolean
+getDeviceList() : String
+setStartComponent(startComponent : ComponentWeakPtr) : void

Figure 1: UML class diagrams of abstract interface class Tanga::Engine and implemented class Tanga::PaEngine.

Before signal processing starts in the signal processing module, the hardware is instructed by the host to open the driver specified in the registry. Then the callback function is activated and called whenever the output buffer of the sound card needs to be filled. The callback function in turn calls a 'perform' method to do the actual audio processing.

2.3 TANGA Engine Module

The engine module is responsible for controlling the signal processing module. At the same time it has to pass on the computed audio samples to the host module for audio output on the sound card or to a sound file. Thus the engine module has to cope with different use cases, a fact that suggests implementing the engine as an abstract interface class.

Fig. 1 shows the Unified Modeling Language (UML) class diagram of the classes Tanga::Engine and Tanga::PaEngine. Tanga::PaEngine is an implementation of the Tanga::Engine interface class. It uses the API specified in the host module, here the PortAudio API. Alternative APIs can easily be used when Tanga::PaEngine is modified accordingly. Tanga::PaEngine has a member of the class Tanga::ComponentGraph (m_componentGraph), in which a ComponentConnector registers the signal paths that exist between signal processing Components. m_outputBuffers contains the buffers that the final component in the signal processing chain is writing the samples into.

2.4 TANGA Signal Processing Module

The signal processing module contains the actual audio processing functionality of the TANGA system. It is connected to the engine module via the Tanga::ComponentGraph class. Tanga::ComponentGraph is one of the most important elements of the TANGA framework. It abstracts the connections between Components that make up the Signal Flow Chart (SFC) of the signal processing algorithm to form a directed graph, the so-called Component Graph. The Component Graph consists of nodes (the Components) and edges (the signal connections between the Components). The edges of the Component Graph are directed against the signal flow direction, and only ever one edge exists between two nodes, independently from the actual number of signal connections (number of channels) between the respective Components.

3 Room Acoustic Simulation

In interactive audiovisual scenes, the time and investment necessary to develop completely accurate auditory and visual models is as much of a limiting factor for how much detail will be rendered in many systems as is the computational power alone. Not least because of this the audio simulation in these scenes does not try to be as exact as possible, but merely as credible as necessary to convey a convincing overall impression.

3.1 Implementation of Perceptual Approach

MPEG-4 Audio provides the *Perceptual Approach* as an alternative to room acoustic rendering of audio based on the geometry and material characteristics of the virtual scene. Here a number of parameters (the so-called *perceptual parameters*) is provided with which the characteristics of a filter network representing an artificial room impulse response can be influenced (amplitude-, frequency-, and time-wise). MPEG-4 Advanced Audio BIFS specifies these parameters, and the standard provides detailed information which the implementation in TANGA is based on. Fig. 2 shows an intermediate transformation step of the schematic view of the model as described in the standard.

3.2 Implementation of Physical Approach

For geometry-based room acoustic simulation, MPEG-4 does not provide any guidelines. Any method based on the physical appearance of the room could be used that is able to do the necessary calculi in real time. So far, TANGA uses a simplified image source method for the computation of the early reflections part of the room impulse response. This method should only be applied to convex rooms because no visibility check is performed on the image sources. Elevated sources are projected onto the horizontal plane using the cosine law to modify their gain according to the elevated distance from that plane.

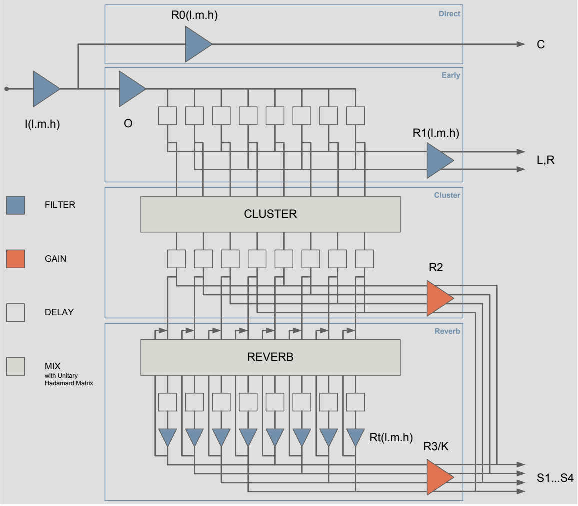


Figure 2: *Schematic view of the MPEG-4 Perceptual Approach signal processing algorithm.*

Late reverberation is created using a nested all pass filter network as described by Gardner [2]. He suggests three different networks (differing in computational complexity) depending on the reverberation time to be generated - small room, medium room and large room. TANGA selects these networks according to the scene description and integrates the associated Component(s) automatically into the Signal Flow Chart. For each loudspeaker channel to be used for reproduction, one such network needs to be computed in order to have largely uncorrelated signals at the loudspeakers.

Therefore the overall computational complexity of the geometry-based simulation depends on three factors:

1. The number of image sources.
2. The number of acoustically relevant objects (walls) present in the scene description.
3. The number of loudspeakers used for the reproduction.

Generally, the Perceptual Approach is slightly less demanding for shoe-box shaped rooms reproduced via four loudspeakers, compared to a simplified image source method based computation of order two. Unfortunately, the Perceptual Approach can, under certain circumstances, develop an unnatural, metallic coloration of sound. Also, its parameters are rather difficult to handle for purposeful manipulation of sound character.

4 Multi-Thread Processing

Computing power available on consumer PCs has been continuously increasing as predicted by Moore's Law for the last decades. Whereas in the past this increase has expressed itself in ever higher CPU clock speeds, the last two years have clearly shown that future increases in computing power will only be possible by integrating a higher number of processor cores into one CPU.

In the area of real time audio rendering, this insight needs to be translated into a systematic approach for distributing and parallelizing the steps of a signal processing chain in order to benefit from future increases in computing power. Whereas this process is rather simple for traditional real time algorithms and complexities, it is all but trivial for scene graph / object based applications as defined e.g. in the ISO/IEC 14496-11 standard (MPEG-4 BIFS) [3].

MPEG-4 describes the audio signal processing chain by means of a so-called scene graph contained in the scene description, a tree structure embracing all audio objects and their dependencies. MPEG-4 players need to interpret the scene description dynamically to render the contained audio in real time, from simple playback of audio files to complex effects resulting in e.g. room acoustic impressions of a 3-dimensional audio visual scene.

Basically, the I3D interprets the MPEG-4 scene description and builds a so-called Component Graph which contains all audio signal processing units (Components) necessary for the rendering of audio. As these Component Graphs can become arbitrarily complex, e.g. for room acoustic simulations as described in section 3, special care has to be taken to make sure that the correct audio is present at the outputs at all time. This can become critical when the processing is distributed into more than one thread to make the computations multi core capable. Basically, this is an optimization problem, but a number of constraints need to be observed. Two solutions have been implemented and compared in the TANGA.

4.1 Dynamic Component Parallel Rendering

One of the methods, the Dynamic Component Parallel Rendering (DCPR), is relatively simple to implement. It is based on a topological sorting generated with a modified breadth first search within a reverse edge Component Graph[1], starting from the root Component. In this graph traversal process, each Component may be visited several times. A Component gets pushed to the beginning of a list during its inspection in the traversal process. If the Component already exists in the list, the existing entry is deleted before adding the Component again. For the Component Graph in fig. 3, the method described delivers $TS_1 = S_1, D_1, ..., D_6, F_1, ...F_6, M_1$. It is obvious that $D_1, ..., D_6$ as well as $F_1, ..., F_6$ can be executed in parallel, because their input does not depend on each other.

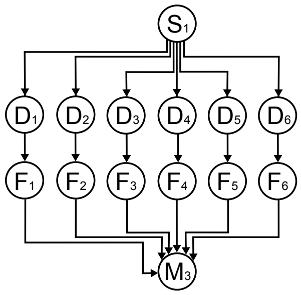


Figure 3: Example for a Component Graph with different types of Components. TANGA Components used: *! *6; 5- %23 □#! #.3+? □%d %23.8□ (! (2>

For multi-threading purposes, the generated topological sorting might be written to a global list. Because the computing of digital audio signals usually is performed block-wise, every signal processing Component has an output buffer which is filled in each render pass. For every render pass, each so-called worker thread[2] takes a Component from the list and performs it. This is done until the list is empty. For a new render pass the list is refilled and the worker threads are started again. To avoid conflicts in the render sequence, the execution of a Component can only be started if the output buffer of the preceding Component(s) have been correctly filled. The worker thread itself verifies that all parent Components have been completely performed before starting. If rendering has not completed, the thread will wait until all parent Components have been performed.

Especially for Component Graphs of higher complexity as e.g. in fig. 4 this can become disadvantageous. Another major disadvantage of this approach

[1] In reverse edge Component Graphs the direction of edges is reversed.

[2] The threads reserved for actually performing the computations defined in a signal processing Component are called 'worker threads'.

is its fine granularity, which generates a lot of synchronization overhead. This is roughly the same for all Components, but for computationally less intensive Components the relative weight of the synchronization overhead is much higher compared to more complex signal processing Components. The advantages of the DCPR are its simplicity and the dynamic adjustment to the graph's structure and to different numbers of threads.

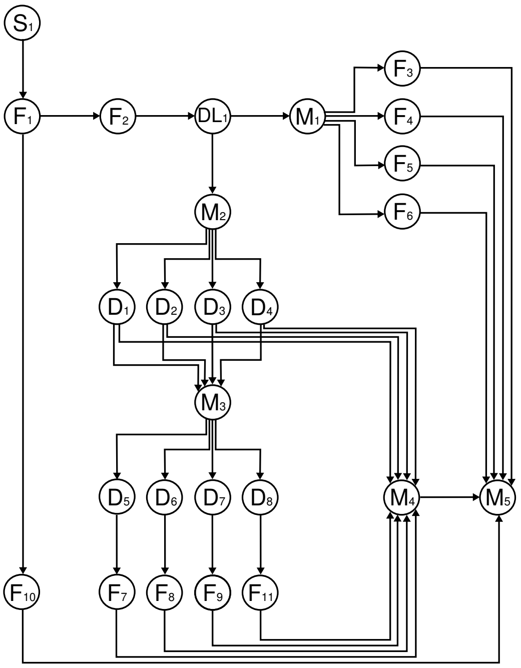


Figure 4: The Component Graph for the) . 8, . 7:; +3 " 7786+, 1 with four internal channels. TANGA Components used: *! *6; 5- %23 □ #! #. 3+? □#' ! #. 3+?' 25. □%! %23. 8 □(! (2>

4.2 Dynamic Component Cluster Rendering

The other method, the Dynamic Component Cluster Rendering (DCCR), avoids the granularity problem by clustering Components, thus forming larger groups of signal processing units to be rendered in parallel threads. The basic idea is to have worker threads executing whole sections of the Component Graph and not individual Components. Therefore a graph has to be divided into sub-graphs. For this, information about the computational complexity of each Component and the number of threads executing the graph is necessary.

Each TANGA Component has an immanent measure describing its average processing time, the Processing Time Index (PTI): this measure varies from Component to Component. A complex filter algorithm will have a different PTI than a simple signal mixing algorithm. This measure needs to be determined a priori and becomes an attribute of the signal processing Component itself.

Clustering of a graph is a so-called NP[3]-complete decision problem. This means that there is no algorithm which yields the exact right solution for the optimal clustering of a graph in polynomial time. To find a solution for this kind of problem, heuristics (also known as *rules of thumb*) must be used. These heuristics render satisfactory results in a very large number of cases.

The clustering method developed for the TANGA engine is called Color Clustering and consists of three steps:

1. Structural segmentation of a graph or sub-graph.
2. Optimization by clustering sub-graphs.
3. Estimation of the clustering efficiency based on the PTI. From this, the *Graph Processing Time Index* (GPTI) is derived and compared to find the optimum solution.

The method is described in more detail in [4]. The advantage of the DCCR is a higher performance compared to the DCPR. Its downside is a slightly more complex (code-wise) implementation. A detailed comparison of the performance of the two methods can also be found in [4].

Unlike in other implementations, the TANGA multi threading approaches can handle an arbitrary number of cores or CPUs for optimal distribution of Components to threads. Both methods are therefore very well primed for future developments of multi core CPUs. In analogy to the TANGA Component Graph, the audio signal processing tree is conceived as a graph upon which the optimization is performed. The result is an optimum distribution of computing processes to available cores or threads.

5 Interactive Scenes

Figs. 5 and 6 show example screenshots of two interactive audiovisual scenes rendered by the I3D. At TU Ilmenau, the I3D is mainly used as an experimental platform for evaluating the effects of human audiovisual perception. A number of experiments regarding the perceived overall quality, also under varying degrees of interaction, have been performed. In these scenes, different levels of details of room acoustic simulation as well as different simulation methods have been compared. More details can be found in e.g. [5, 6].

In addition to the standard MPEG-4 input sensors (key sensor and mouse sensor), a MIDI sensor has been

[3] Non-deterministic Polynomial time

implemented in the Windows version of the I3D. It is based on the MIDI wrapper library by Leslie Sanford [7]. This wrapper library encapsulates the MIDI functions of the Windows Multimedia API in an object oriented way and thus provides easy to use send and receive mechanisms for all types of MIDI messages for the I3D. These can be routed and transformed using the ECMA script language [8] to influence any MPEG-4 node field specified as *exposedField*. Message routing and transformation is part of the scene description itself, which results in very high flexibility.

Messages can also be triggered by events which in turn can have a wealth of sources. This way, both simple and complex interaction schemes can be devised.

As the I3D can also render 2D scenes or combined 2D/3D scenes containing 2D videos of arbitrary shape, a large number of interactive applications can be thought of. These might be interactive games, individualized news broadcasts, or edutainment applications, to name a few.

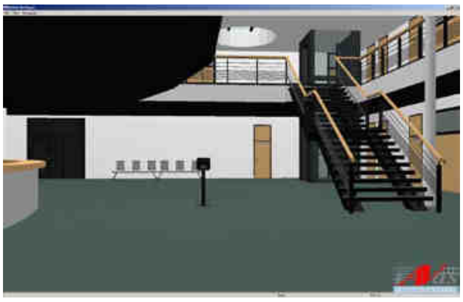


Figure 5: Screenshot of an interactive scene representing the entrance hall of the main lecturing building at TU Ilmenau.

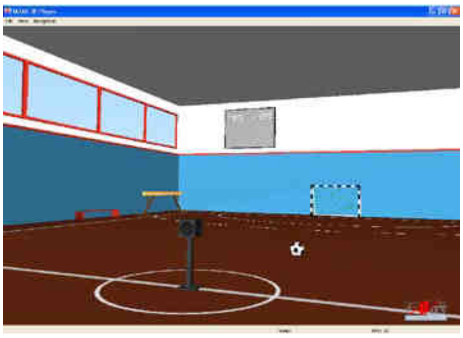


Figure 6: Screenshot of an interactive scene representing a sports gym. Users can e.g. interact with the football.

6 Conclusion and Outlook

This paper has presented the most outstanding features of the TANGA real time audio engine. TANGA can be used stand-alone or as part of the I3D MPEG-4 player. The latter provides elegant ways to interactively influence audio rendering mechanisms as well as audiovisual appearance and content of scenes in general.

Future additions to the TANGA will include the implementation of a beam tracing method for rendering of audio in 3D scenes consisting of connected spaces.

7 Acknowledgment

This work is supported by the EC within FP6 under Grant 511568 with the acronym ’3DTV’.

References

[1] Bencina, Ross; Burk, Phil: *PortAudio - an Open Source Cross Platform Audio API*, Proc. 2001 Int. Computer Music Conference, Havana, Cuba, Sept. 2001, pp 263-266.

[2] Gardner, W.G.: *A Realtime Multichannel Room Simulator*, J. Acoust. Soc. Am., 92(4), pp 2395, and presented at the 124th Meeting of the Acoustical Society of America, New Orleans, USA, Nov. 1992.

[3] Int. Std. (IS) ISO/IEC 14496-11:2004, *Information technology - Coding of audio-visual objects - Part 11: Scene description and Application engine*, Geneva, Switzerland, 2004.

[4] Reiter, U.; Partzsch, A.: *Multi Core / Multi Thread Processing in Object Based Real Time Audio Rendering: Approaches and Solutions for an Optimization Problem*, Proc. 122nd AES Convention, Vienna, Austria, May 5-8, 2007.

[5] Reiter, U.; Jumisko-Pyykkö, S.: *Watch, Press, and Catch - Impact of Divided Attention on Requirements of Audiovisual Quality*, HCI Int. Conf. on Human Computer Interaction, Beijing, PR China, July 22-27, 2007.

[6] Reiter, U.; Weitzel, M.; Cao, S.: *Influence of Interaction on Perceived Quality in Audio Visual Applications: Subjective Assessment with n-Back Working Memory Task*, Proc. AES 30th Int. Conf. Saariselk, Finland, March 15-17, 2007.

[7] Wrapper Library for Windows MIDI API, by Leslie Sanford, http://www.codeproject.com/audio/MIDIWrapper.asp

[8] Int. Std. (IS) ISO/IEC 16262:1999, *Information Technology - ECMAScript language specification*, Geneva, Switzerland, 1999.

Blind One-to-N Upmixing

Christian Uhle, Andreas Walther, and Michael Ivertowski
Fraunhofer IIS, Erlangen, Germany
{uhle, wth, ivertoml}@iis.fraunhofer.de

Abstract. Recently, a novel method for the extraction of an ambience signal using Non-negative Matrix Factorization for ambience-based blind upmixing of one-channel audio signals has been proposed. This paper investigates how to improve the overall subjective quality of the surround presentation by applying suitable post-processing techniques. The basic requirements for the upmixed multi-channel signal are formulated and appropriate audio signal processing methods are proposed. This includes signal adaptive equalization, transient suppression and signal decorrelation. Listening tests confirm the listeners' preference of the post-processed upmix compared to unprocessed audio and previous upmix algorithms when played back on a 5.0 system.

1 Introduction

Multi-channel surround sound reproduction enables a more realistic reconstruction of a sound field than two-channel stereophony. The increasing availability of surround sound systems (e.g. home theatre and multimedia computer setups) evokes the consumers' desire to exploit their advantages for the reproduction of legacy content. The mismatch between the surround sound setup and the legacy content format (mono or stereo) creates the need for content format conversion which is addressed by this paper.

The term m-to-n upmixing describes the conversion of an m-channel audio signal to an audio signal with n channels, where $m < n$. Two concepts of upmixing are widely known: Upmixing with additional information guiding the upmix process and unguided ("blind") upmixing without use of any side information.

Among the upmix methods are ambience-based techniques [2, 1, 3]. Their core component is the extraction of an ambient signal which is fed into the rear channels of a multi-channel surround sound signal. Ambient sounds are those forming an impression of a (virtual) listening environment, including room reverberation, audience sounds (e.g. applause), environmental sounds (e.g. rain), artistically intended effect sounds (e.g. vinyl crackling) and background noise. The reproduction of ambience using the rear channels evokes an impression of envelopment ("immersed in sound") by the listener.

The work described here focuses on the unguided ambience-based upmix of mono signals. Recently, a method for the extraction of the ambience signal from mono recordings using Non-negative Matrix Factorization (NMF) [8] and post-processing techniques for the extracted ambience signal [12] have been described. This publication investigates how these techniques and others can be combined to optimize the overall subjective quality of the surround sound signal. The gain over previous methods is quantified by means of listening test results.

The paper is organized as follows: Section 2 provides an overview of the ambience extraction method. The post-processing techniques are described in Section 3. The listening test procedure and results are illustrated in Section 4. Conclusions are drawn in Section 5.

2 Ambience-based blind upmixing

This section provides an overview of the NMF, of the ambience extraction method and of the upmix process applied here.

2.1 Non-negative Matrix Factorization

NMF aims at approximating the matrix $V \in \mathbb{R}^{n\times m}$ by the product of two matrices $W \in \mathbb{R}^{n\times r}$ and $H \in \mathbb{R}^{r\times m}$, with $v_{ik}, w_{ik}, h_{ik} \geq 0$, $\forall i, k$ and factorization rank r [4].

$$v_{ik} \approx (WH)_{ik} = \sum_{a=1}^{r} w_{ia} h_{ak} \tag{1}$$

The factors W and H are computed by solving the optimization problem of minimizing a cost function $c = f(V, WH)$ measuring the error of the approximation. Appropriate cost functions are e.g. the generalized Kullback-Leibler divergence and the Frobenius norm [4].

The product WH is a data compressed representation of V if the factorization rank r fulfils the condition $(n + m)r < nm$. If the parameters r, n and m are chosen such that a perfect reconstruction of V is not achieved, the NMF aims at the representation of both

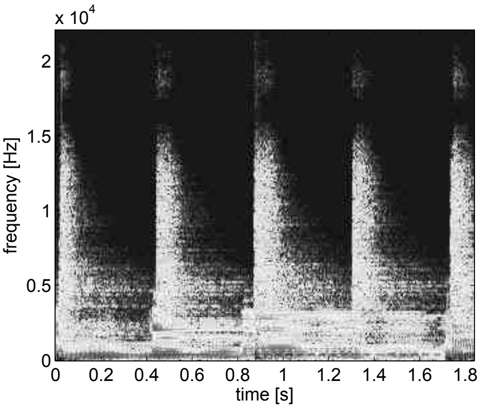


Figure 1: Magnitude spectrogram of the unprocessed audio signal.

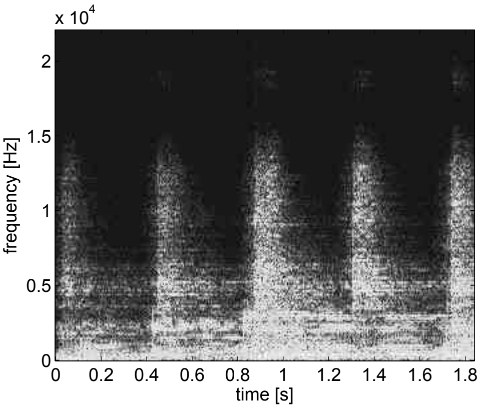


Figure 2: Magnitude spectrogram of the extracted ambient signal.

repetitive structure and regions with concentrated energy content in the data matrix, i.e. a parts-based representation.

2.2 Ambience extraction

The input signal $x\,[k]$ is assumed to be an additive mixture of a direct signal $d\,[k]$ and an ambient signal $a\,[k]$. The spectrogram $X(\tau,\omega)$ of the audio signal $x\,[k]$ is computed by means of the Short-term Fourier Transform (STFT), with the frame index τ and the index of the frequency bin ω.

In comparison to the direct signal, the ambient signal is rather widely spread over the spectrogram. Consequently, an NMF of the magnitude spectrogram $|X(\tau,\omega)|$ provides a better approximation of the magnitude spectrogram $|D(\tau,\omega)|$ of the direct signal than of the magnitude spectrogram $|A(\tau,\omega)|$ of the ambient signal.

An estimate of the magnitude spectrogram $|\hat{A}(\tau,\omega)|$ of the ambient signal is derived by

$$|\hat{A}(\tau,\omega)| = |X(\tau,\omega)| - |\hat{D}(\tau,\omega)| \qquad (2)$$

whereas an estimate of the magnitude spectrogram $|\hat{D}(\tau,\omega)|$ of the direct signal is computed as an NMF of $|X(\tau,\omega)|$. Figures 1 and 2 show the magnitude spectrogram of the input signal and the corresponding estimate of the magnitude spectrogram of the ambient signal, respectively.

Finally, an estimate of the ambient time signal $\hat{a}\,[k]$ is resynthesized from $\hat{A}(\tau,\omega) = |\hat{A}(\tau,\omega)|e^{j\angle X(\tau,\omega)}$ using the inverse STFT, where $\angle X(\tau,\omega)$ denotes the phase spectrogram of the input signal $x\,[k]$. An estimate of the direct signal $\hat{d}\,[k]$ is resynthesized in a similar way using $|\hat{D}(\tau,\omega)|$ and $\angle X(\tau,\omega)$.

2.3 Assembly of the surround sound signal

Subsequently to the ambience extraction step, a 5.0 surround signal is obtained by feeding the rear channels with the ambient signal. A few milliseconds delay is introduced into the rear channel signals to improve the stability of the front image by making use of the precedence effect.

The center channel is used to enlarge the sweet spot and plays back the estimate of direct signal.

Additional post-processing as described in the following section optimizes the overall subjective quality of the surround sound signal.

3 Post-processing for one-to-n upmixing

3.1 Requirements for upmixed surround sound

Although there is no fixed ground truth how the result of a blind upmix should sound, some basic requirements guide the development of the ambience post-processing techniques:

- The processed audio signal should sound natural.
- A stable front image should be preserved.
- The surround sound should evoke the listening impression of being immersed in sound (envelopment).

These issues are addressed by applying suitable post-processing techniques, namely signal adaptive equalization, transient suppression and decorrelation.

3.2 Signal adaptive equalization

To minimize the timbral coloration of the surround sound signal, the ambience signal is equalized to adapt its long-term power spectral density (PSD) to the input signal. This is carried out in a two-stage process.

The PSD of both, the input signal $x[k]$ and the ambience signal $a[k]$ are estimated using the Welch method, yielding $I_{xx}^{W}(\omega)$ and $I_{aa}^{W}(\omega)$, respectively. The frequency bins of $|\hat{A}(\tau,\omega)|$ are weighted prior to the resynthesis using the factors

$$H(\omega) = \sqrt{\frac{I_{xx}^{W}(\omega)}{I_{aa}^{W}(\omega)}} \tag{3}$$

The signal adaptive equalization is motivated by the observation that the extracted ambience signal tends to feature a smaller spectral tilt than the input signal, i.e. the ambient signal may sound brighter than the input signal. In many recordings, the ambient sounds are mainly produced by room reverberations. Since many rooms used for recordings have smaller reverberation time for higher frequencies than for lower frequencies, it is reasonable to equalize the ambient signal accordingly. However, informal listening tests have shown that the equalization to the long-term PSD of the input signal turns out to be a valid approach.

3.3 Transient Suppression

The introduction of a time delay into the rear channel signals (see Section 2.3) evokes the perception of two separate sounds (similar to an echo) if transient signal components are present [11]. This echo is attenuated by suppressing the transients in the surround sound signal. Additional stabilization of the front image is achieved since the appearance of localizable point sources in the rear channels is reduced.

Considering that ideal enveloping ambient sounds are smoothly varying over time, a suitable transient suppression method reduces transient components without affecting the continuous character of the ambience signal. One method that fulfils this requirement has been described in [12] and is applied here.

First, time instances where transients occur are detected. Subsequently, the magnitude spectrum belonging to a detected transient region is replaced by an extrapolation of the signal portion preceding the onset of the transient.

Therefore all values $|X(\tau_t,\omega)|$ [1] exceeding the running mean $\mu(\omega)$ by more than a defined maximum deviation are replaced by a random variation of $\mu(\omega)$ within a defined variation interval.

To assure smooth transitions between modified and unmodified parts, the extrapolated values are cross-faded with the original values.

Figure 3 and 4 show the spectrogram of a signal containing tonal parts and transient components before and after the transient suppression.

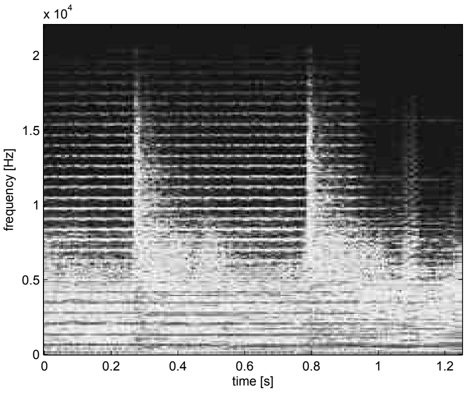


Figure 3: Spectrogram of an audio signal before transient suppression

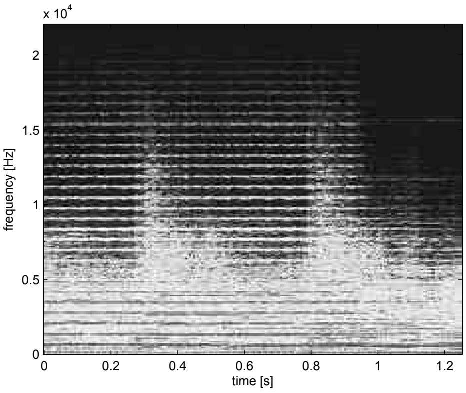


Figure 4: Spectrogram of an audio signal after transient suppression

3.4 Decorrelation

The correlation between the two signals arriving at the left and right ear influences the perceived width of a sound source and the ambience impression. To improve the spaciousness of the impression, the inter-channel correlation between the front channel signals and between the rear channel signals is decreased by applying the principle of Adaptive Spectral Panoramization [10] with slight modifications. This process is described in the following for the front channel signals.

Two mutually decorrelated versions $x_L[k]$ and $x_R[k]$ of the input signal $x[k]$ are computed by applying complementary time-varying weights $G_L(\tau,m)$ and $G_R(\tau,m)$ to the sub-band signals $X_s(\tau,m)$ of the input signal, where $m = 1\dots M$ denotes the frequency band index and $G_L^2(\tau,m) + G_R^2(\tau,m) = 1$.

The sub-band signals $X_s(\tau,m)$ are derived by com-

[1] subscript t indicates frames belonging to a transient region

puting the STFT of $x[k]$ and subsequently grouping the frequency bins into band signals according to the Bark scale. The weights are computed using low-level feature extraction in analysis bands $X_a(\tau,m)$, whereas the adjacent bands for the m-th band of $X_s(\tau,m)$ are subsumed to the m-th analysis band $X_a(\tau,m)$ (except for $m \in \{1, M\}$ where only one adjacent sub-band exists and is used) (see Equation 4).

$$X_a(\tau,m) = \sum_{n=m-1}^{m+1} X_s(\tau,n) \tag{4}$$

The spectral centroid $\Theta(\tau,m)$ is calculated for each sub-band m of the sub-bands $X_a(\tau,m)$ of the input signal and normalized to $\Theta \in [0,1]$, whereas $\Theta = 0$ corresponds to the lowest frequency and $\Theta = 1$ corresponds to the highest frequency of the sub-band. A panning position ϕ is computed from the centroid Θ using the arbitrarily chosen function in Equation 5, whereas the parameter c determines the maximum width of the panning and the parameter w_i depends on the sub-band index.

$$\phi = c \cdot (-1)^m cos\left(2\Theta w_i \pi - \pi\right) \tag{5}$$

The weights are derived as a function of ϕ according to

$$G_L(\tau,m) = \frac{\sqrt{2}}{2}\left(\cos\phi(\tau,m) + \sin\phi(\tau,m)\right) \tag{6}$$

$$G_R(\tau,m) = \frac{\sqrt{2}}{2}\left(\cos\phi(\tau,m) - \sin\phi(\tau,m)\right) \tag{7}$$

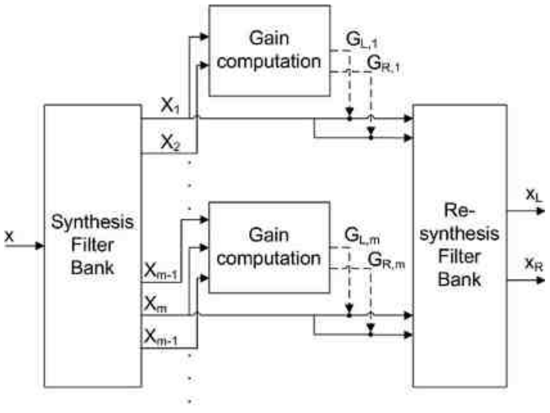


Figure 5: Block diagram of the decorrelation process using ASP for an input signal x.

This process is illustrated in Figure 5, where it can be seen that there is a tight relation to a prominent class of signal enhancement techniques known as short-term spectral attenuation or spectral weighting (comprising e.g. spectral subtraction, Wiener filtering and the Ephraim-Malah algorithm for noise reduction), see e. g. [5]. The rear channel signals are decorrelated using the same gain coefficients as applied to the front channel signals.

4 Listening Tests

4.1 Description of the procedure

Subjective listening tests according to the scheme proposed in [7] were carried out to confirm the listeners' preference of the post-processed upmix compared to unprocessed audio and previous upmix algorithms.

The test was performed in a non standard listening room with a reverberation time of approximately 0.36s. The listening position was the sweet spot of a 5.0 loudspeaker setup consisting of *Genelec 1031A* two-way active monitors arranged according to ITU-R BS.775 [6] with a radius of 2.0 meters. Eight items representing complete musical phrases from a variety of musical genres with a length between 7 and 12 seconds each were chosen as test items.

Phantom-mono reproduction (in the following referred to as *two-ch*) was defined as the condition relative to which the listeners rated the other conditions as *worse*, *equal* or *better* and simultaneously ranked all conditions in comparison to each other. Other test conditions used for comparison are

- *one-ch*: the mono signal played back by the center-speaker exclusively
- *trivial*: the mono signal played back by all 5 loudspeakers
- *trivial-post*: the *trivial* condition with additional processing of the surround channels by low-pass filtering and transient suppression
- *NMF*: the original mono upmix system described in [8]

The newly proposed test condition is referred to as *NMF-post.*

The levels of all conditions were adjusted to assure that no volume differences between the different conditions influence the rating. Additionally, the perceived loudness of the rear channel signals of all conditions were equalized prior to the adjustment of the overall level.

4.2 Results

Figure 6 shows the combined ratings of all subjects as notched box plots. The notches indicate the 95% confidence interval about the median. If the notches

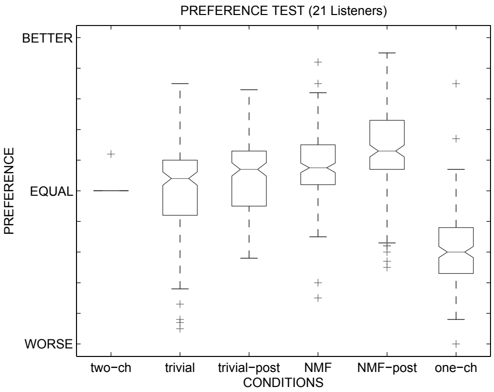


Figure 6: Results of the preference test for all listeners.

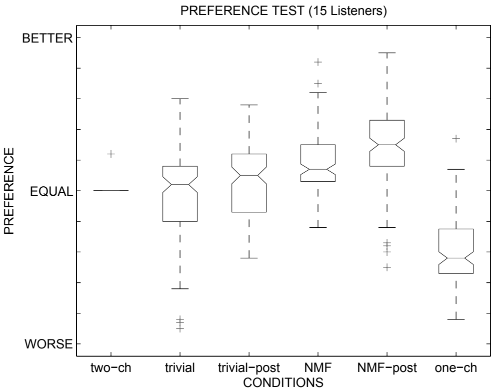


Figure 7: Results of the preference test for the significant listeners.

of two boxes do not overlap, we can be confident at a 95% level that the two medians are different [9]. Each box plot in Figure 6 is calculated from a total of 168 ratings (21 listeners times 8 items).

The variance of the responses of each listener was analyzed using a Kruskal-Wallis test. Six listeners were discarded because their ratings did not indicate that they perceived at least one statistically significant difference ($p \geq 0.05$) between the multi-channel versions *trivial*, *trivial-post*, *NMF* and *NMF-post*.

Figure 7 shows box plots of the combined results of the remaining 15 listeners. The results show that the one channel reproduction *one-ch* is the only one which is clearly rated worse than the phantom mono reproduction *two-ch*. The trivial solution is rated as being equally good compared to *two-ch*. One reason for the gain in preference compared to previous tests [8] could be the described rear-level adjustment. *Trivial-post* is rated as being preferred in a statistically significant sense compared to *two-ch* and shows a slight though not statistically significant preference increase over the *trivial* condition. The original mono upmix *NMF* is rated with a statistically significant higher preference compared to *two-ch* and *trivial*. Compared to *trivial-post*, *NMF* is rated slightly better but not in a statistically significant sense. Condition *NMF-post* is rated as being statistically significant preferred compared to all other conditions.

5 Conclusions

Various suitable post-processing techniques for upmixing and their application to unguided ambience-based upmixing of one-channel audio signals have been presented.

The applied ambience extraction method is based on the NMF of the magnitude spectrogram of the input signal. The post-processing methods were motivated by the formulated requirements for upmixed surround sound and included signal adaptive equalization, transient suppression and signal decorrelation.

A substantial improvement compared to previous methods and un-processed audio has been shown in listening tests, where listeners were asked for their preference.

Acknowledgements

The authors wish to express their thanks to Jürgen Herre and Fabian Küch for valuable discussions and proof-reading of the paper.

References

[1] C. Avendano and J. M. Jot. Ambience extraction and synthesis from stereo signals for multi-channel audio up-mix. In *Proc. of the ICASSP*, 2002.

[2] D. Griesinger. Multichannel matrix decoders for two-eared listeners. In *Proc. of the AES 101st Conv.*, 1996.

[3] R. Irwan and R. Aarts. Two-to-five channel sound processing. *J. Audio Eng. Soc.*, 50, 2002.

[4] D. Lee and H. Seung. Algorithms for non-negative matrix factorization. In *Proc. of NIPS*, 2001.

[5] G. Schmidt. Single-channel noise suppression based on spectral weighting - an overview. *EURASIP Newsletter*, 2004.

[6] ITU Radiocommunication Sector. Multichannel stereophonic sound system with and without accompanying picture. Recommendation ITU-R BS.775-2, 2006. Geneva, Switzerland.

[7] T. Sporer, A. Walther, J. Liebetrau, S. Bube, C. Fabris, T. Hoheberger, and A. Köhler. Perceptual evaluation of algorithms for blind up-mix. In *Proc. of the AES 121st Conv.*, 2006.

[8] C. Uhle, A. Walther, O. Hellmuth, and J. Herre. Ambience separation from mono recordings using non-negative matrix factorization. In *Proc. of the AES 30th Conference*, 2007.

[9] P.F. Velleman and D.C. Hoaglin. Applications, basics, and computing of exploratory data analysis. Previously published by Duxbury Press, Boston. Republished by "The Internet-First University Press", http://dspace.library.cornell.edu/handle/1813/62.

[10] V. Verfaille, U. Zoelzer, and D. Arfib. Adaptive digital audio effects (a-dafx): A new class of sound transformations. *IEEE Transactions on Audio, Speech, and Language Processing*, 2006.

[11] H. Wallach, E.B. Newman, and M.R. Rosenzweig. The precedence effect in sound localization. *J. Audio Eng. Soc.*, 21:817–826, 1973.

[12] A. Walther, C. Uhle, and S. Disch. Using transient suppression in blind multi-channel upmix algorithms. In *Proc. of the AES 122nd Conv.*, 2007.

Wave Field Synthesis for Games using the OpenAL interface

Andreas Gräfe
IOSONO GmbH
andreas.graefe@iosono-sound.com

Martin Dausel
IOSONO GmbH
martin.dausel@iosono-sound.com

Andreas Franck
Fraunhofer IDMT
fnk@idmt.fraunhofer.de

Abstract. High-quality audio becomes a more and more important aspect of interactive applications. Especially in video games, the efforts towards a more realistic audio experience have been raised during the last years. Spatial reproduction techniques have gained widespread use and contribute greatly to the immersion of the user.

Wave field synthesis (WFS) is a reproduction concept that offers superior localization quality. In contrast to most existing audio reproduction techniques, the area of correct reproduction is not limited to a rather small sweet spot, but gained in a large reproduction area.

Until now, the application of wave field synthesis in games and other interactive applications is limited due to the lack of standardized interfaces.

In this article, we describe an implementation of the Open Audio Library (OpenAL) interface that controls a WFS reproduction system. OpenAL is a free application programmer interface for 3D positional audio that is used by numerous games and other applications. Its features include among others several distance attenuation models, doppler shift and sound source directivity. A great advantage of OpenAL lies in its portability, since there exist implementations for many platforms and audio interfaces.

OpenAL is well-suited for interfacing a WFS reproduction system because the object-based description of the auditory scene is similar to the model used in wave field synthesis. Despite this, some minor translations are necessary in order to fit the two models to one another. In this article, we describe the necessary steps.

Based on an implementation of the system, we present several applications using the WFS OpenAL interface. It is shown that auditory scenes described using the OpenAL interface can be reproduced by a Wave Field Synthesis system. Many applications, in particular games, benefit from the properties of wave field synthesis, such as the good localization properties. So, using WFS for audio reproduction can improve interactivity and immersion of interactive applications.

1 Introduction

Interactive virtual reality applications not only demand a convincing visual representation of the virtual environment, but also high quality spatial audio reproduction. In video games, the audio part is no longer limited to background music, but plays an important role in creating a realistic environment with natural sounds. In some games, the sound effects are an essential part of the game since they provide clues about the position and distance of an enemy or the noise created by the user's avatar. The Thief$^{\mathrm{TM}}$[7] and Splinter Cell$^{\mathrm{TM}}$[6] series serve as good examples for such "stealthy" games.

At the same time, potential application areas for the Wave Field Synthesis concept are discussed, including home entertainment (see [11] and [12]). The improved spatial sound reproduction and localization features of WFS are very likely to enhance the immersion of the user significantly. At this moment, there are only a few WFS applications available since there is no standardized software interface for audio applications.

In this paper, we want to examine the feasibility of such an interface by developing a concept for an interface between OpenAL and WFS. The concept shall be proven by implementing a prototype that allows any OpenAL compliant application to control a WFS system.

In the first section we give a short introduction into the principle of Wave Field Synthesis and describe a reproduction system which uses this technique. Secondly, the OpenAL interface will be introduced. After that, a concept for an implementation of OpenAL which controls a WFS system is presented. Finally, we draw a conclusion about the results and name some possible options for further development.

2 Wave Field Synthesis

Wave Field Synthesis (WFS) is a relatively new sound reproduction concept which has been developed in the 90's at the TU Delft in Netherlands (see [8] and [14]).

In contradiction to traditional channel-oriented reproduction concepts, WFS attempts to reproduce the actual wave field emitted by one or more physical sound sources. By doing this, the area of spatial sound reproduction can be extended from a small sweet spot to a wide listening area.

2.1 Background

The theoretical base of WFS is the Huygens principle, named after Christiaan Huygens. This principle states that any point of a propagating wave front can be seen as the source of a so called elementary wave (see figure 1). In 3D space, this elementary waves are spherial waves. Therefore, an arbitrary wave front can be generated (or *synthesized*) by superposition of a number of elementary waves.

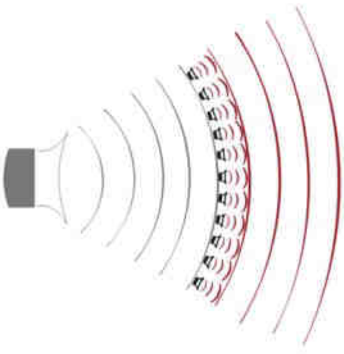


Figure 1: Huygens principle

Since the wave front is traveling in space, the elementary sources would also have to. This can be avoided by using a number of fixed position loudspeakers (often called loudspeaker arrays), whose driving signals are scaled and delayed according to the wave front of the source to synthesize (see figure 2). These speakers are called *secondary sources*. Consequently, the actual synthesized source is called *primary source*. The Huygens principle can be described analytically with the Kirchhoff-Helmholtz integral representation. By applying this integral to equally-spaced loudspeaker distributions, the driving signals for the secondary sources can be derived and used to synthesize the wave field of a primary source [14].

In most practical applications, however, the use of two-dimensional loudspeaker distributions is too complex. Therefore, the theory has been simplified towards linear loudspeaker arrays. This step also reduces the valid reproduction area for a primary source to a plane (see [14]).

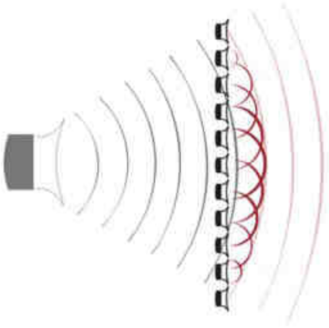


Figure 2: WFS principle

Note that in WFS context the operation of calculating the loudspeaker signals is often denoted as *rendering* according to computer graphics terminology.

2.2 Implementation of a WFS system

There are two different approaches for the calculation of the driving signals of the secondary sources: data based rendering and model based rendering [13].

In *data based rendering*, the driving signals of the secondary sources are generated by convolving the source signal with precalculated impulse responses. This method is very flexible, because it permits arbitrary source- and loudspeaker-dependent filtering operations. On the downside, sources are limited to a fixed grid of locations, requiring either large data for storage imulse responses or computational resources for on-line computation of the filters. Furthermore, the fixed positions of the impulse responses complicates the synthesis of moving sound sources.

In reproduction systems based on *model based rendering*, the secondary source signals are calculated using so-called *synthesis operators* [14]. This operators calculate individual scale and delay values for each combination of virtual sources and loudspeakers based on geometric properties of the virtual source and the loudspeaker setup. This allows real-time rendering of sources at arbitrary positions or moving sound sources.

The IOSONO wave field synthesis system, which has been developed at the Fraunhofer IDMT, uses the model based rendering approach. Since the interface described in this paper was designed to work with the IOSONO system, all further explanations refer to this system. The configuration of a typical IOSONO rendering system is depicted in figure 3.

The core of the system is the Renderer. Using the WFS operators derived in [14], it calculates the loudspeaker signals out of the audio data and the source-wise scene description information. After the render-

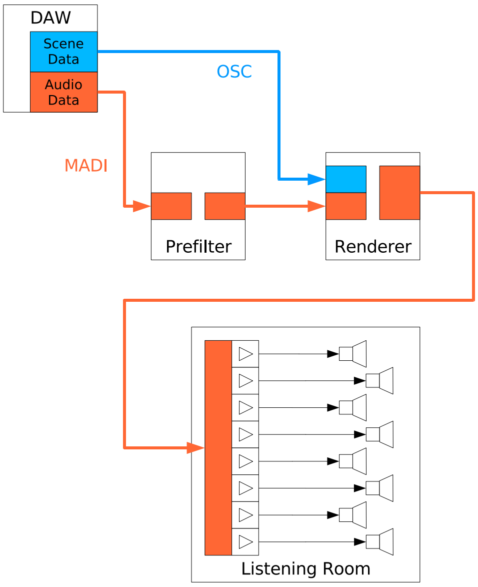


Figure 3: System concept

ing process, the loudspeaker signals are amplified and played back by the speaker array, thus creating the wave field. In most practical systems, the source audio data has to be filtered in advance to compensate the effects of the reproduction room. This is done by the Prefilter unit. The audio data has to be fed source-wise into the system, using a multichannel audio interface (typically MADI). The scene description data (see section 2.3) is provided via ethernet using the Open Sound Control (OSC) protocol [5]. Thus any external application (here for instance a Digital Audio Workstation) is able to control the IOSONO system if it implements the scene protocol.

2.3 WFS Scene Description

In order to create an auditory scene by wave field synthesis, the rendering system needs the audio data to be played by the sound sources a well as information about the scene itself. The IOSONO system uses an object-oriented scene description, which allows the assignment of several parameters to each sound source. These parameters are updated at regular intervals and can be controlled by an external application by using the OSC protocol. The following parameters can be controlled:

- **Source position** in the reproduction plane
- **Angle** with respect to the speaker array, only relevant for plane waves
- **Volume** of the source
- A flag **PlaneWave** which indicates whether the source is a point source or a plane wave
- A flag **DDL** which triggers the distance dependent loudness calculation
- A flag **DDD** which triggers the distance dependent delay calculation

By controlling these parameters, several application scenarios are possible, e.g. a virtual 5.1 setup without sweet spot [9] or synthesis of moving sound sources by simply varying the source position with respect to time.

3 The Open Audio Library

3.1 Concept

Open Audio Library (OpenAL) is an open source, platform independent 3D audio API. It was mainly designed to be used within games, but is not limited to this area of application (see [2]). The API is knowingly designed similar to its graphics pendant OpenGL in order to simplify the development of cross-platform audio/video applications. Also, OpenAL offers the opportunity to extend the API with custom functionality. For example, one of this *extensions* permits the programmer to use hardware accelerated room simulation effects by using the Creative EAX^{TM}API (see [1]).

OpenAL implementations are available for many platforms, including Windows, Linux, MacOS and Microsoft Xbox. Also, a growing number of games and audio applications use the OpenAL interface (see [2]) despite that the rival Microsoft DirectSound3D is more widely-used.

3.2 Scene Description

OpenAL uses an object-oriented scene description to specify a 3D auditory scene. In OpenAL terms, the scene is called *context*. Within a context, three types of objects are used to describe the scene:

- **Source objects** representing a single sound source in 3D space
- A singleton **Listener object** which represents the user
- **Buffer objects** containing the audio data to be played by one or more sources

Each of this objects has a number of parameters, such as position, velocity or gain that can be controlled by the application. The parameters are used to synthesize certain acoustic effects which contribute to the authenticity of the auditory scene. In particular these effects are:

- Distance dependent attenuation of a sound source

- Doppler effect (velocity dependent, distance changes are ignored)
- Source directivity

Each of this effects may be parametrized in order to allow the greatest possible flexibility in modeling auditory scenes, even if the effects may sound ”unnatural”.

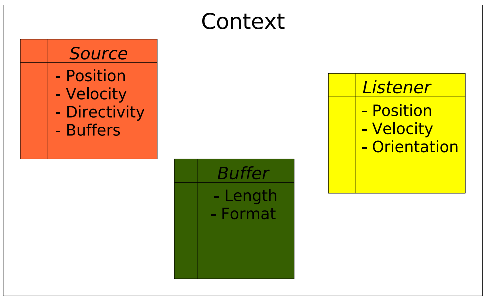


Figure 4: OpenAL Scene Model

3.3 Implementation

An implementation of OpenAL provides the API on a particular platform (i.e. an operating system or a games console). The main task of such an implementation is to interpret the parameters of the objects within the context and calculate the output audio data accordingly. For example, if the audio hardware supports 5.1 playback the implementation has to ensure an appropriate panning of the distinct output channels depending on the source position relative to the listener. Furthermore, the acoustic effects mentioned in section 3.2 must be calculated and applied to the audio data in case the audio hardware doesn’t support any DSP effects.

Since OpenAL is hardware independent, an implementation on a particular platform has to use system specific audio backends (e.g. DirectSound on Windows, ALSA on Linux etc.) in order to access the audio hardware. Some vendors, however, provide native OpenAL drivers for their products.

4 Porting OpenAL to WFS

4.1 General Concept

Comparing the scene descriptions of OpenAL and WFS, the similarities become obvious. Both systems describe a number of sound source objects and use their parameters (basically the position) to apply signal processing operations to audio streams in order to create the impression of spatial sound. An interface between both would have the main task to transform the parameters and effects used in OpenAL into an adequate WFS scene description. Also, the audio data played back by the OpenAL application has to be fed into the WFS system.

Since the OpenAL API is portable, it seems suitable to design the WFS interface also to be as hardware independent as possible. For this purpose, the PortAudio interface [3] was used as an adaption layer between OpenAL and the audio hardware. Concerning the scene description, a portable OSC implementation called OSCPack was used to assemble the packets containing the source parameters and send them via UDP to the Renderer. Figure 4.1 shows the abstract design of the WFS to OpenAL interface. A more detailed description can be found in [10].

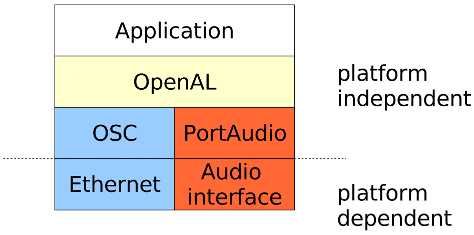


Figure 5: Interface Concept

Based on the preceding considerations, a prototype implementation for Windows XP was set up using PortAudio (with the ASIO backend) and OSCPack.

4.2 Problems and Trade-offs

Despite the similarities of the scene description, some problems had to be solved in order to match the two scene descriptions to one another. First, the OpenAL description does not incorporate a length measure, while the WFS scene does. To work around this, the sound velocity has to be used to calculate the positions of the sources in absolute coordinates. Secondly, the WFS scene does not contain the notion of a listener object since there is no sweet spot. Thus the OpenAL listener has to be placed implicitly at a fixed position within the listening room (usually the point of origin) and the source positions have to be transformed corresponding to their position relative to the listener.

Furthermore, the parametrization of the Doppler effect and the distance dependent sound pressure decay are not fully supported by the WFS system. Therefore these effects have to be calculated in software in order to keep up with the specification.

5 Applications

5.1 Games

The following games were tested with the prototype:

- Unreal 2
- Pariah
- Cold War
- Jedi Knight: Jedi Academy
- Soldier of Fortune
- Quake 4

All titles except Quake 4 worked well with the prototype. The localization of sound sources within the games was possible and the impression of spatiality could be established. Since WFS is able to synthesize a sound source within the listening room, the impression of a sound source just next to the user or flying around the head could be created. It is also possible for the user to move freely in the listening room without losing this impression.

5.2 Other

Like mentioned in chapter 3, OpenAL is also used by several game engines and a 3D modeling tool called Blender 3D [4]. Using such applications with audio and video modeling capabilities, it is possible to create a virtual environment with spatial distributed sound sources. Using the WFS reproduction system, these sources can be synthesized at their actual positions without the need for a 5.1 downmix. There is also the opportunity of tracking the user position and updating the auditory scene with respect to movements of the user. Therefore, a highly interactive environment can be created in order to perform e.g. perception tests.

6 Conclusion

The main goal of this work was to prove the concept of an application controlling a WFS system using the OpenAL interface. The descriptions for auditory scenes in OpenAL and the WFS system were compared and their similarities pointed out. A concept for an easy portable implementation of OpenAL driving a WFS system was derived and a prototype for the Windows platform implemented. For the future, more evidence for the enhancement of the audiovisual experience by using a WFS system has to be given. A comparative listening test with a 5.1 system is suggested, but there are yet no reliable parameters to measure the “immersion” of the user. Furthermore, the prototype is not yet able to perform a room simulation. Since in many games the EAX^{TM}API is used for reverberation synthesis, further development is necessary to make this feature available for WFS systems.

References

[1] Creative labs developer relations, `developer.creative.com`, 2006.

[2] Open audio library homepage, `www.openal.org`, 2006.

[3] Portaudio homepage, `www.portaudio.com`, 2006.

[4] Blender homepage, `www.blender.org`, 2007.

[5] Open sound control homepage, `www.cnmat.berkeley.edu/OpenSoundControl`, 2007.

[6] Splinter cell homepage, `splintercell.de.ubi.com`, 2007.

[7] Thief 3 homepage, `www.eidos.de/gss/legacy/thiefdeadlyshadows`, 2007.

[8] A. J. Berkhout. A holographic approach to acoustic control. In *AES Journal Vol. 36*, pages 977–995, 1988.

[9] Marinus Boone, Ulrich Horbach, and Werner de Bruijn. Virtual surround speakers with wave field synthesis. In *106th AES Convention*, pages 9905–9917, 1999.

[10] Andreas Graefe. Studienarbeit, 2006.

[11] Beate Klehs and Thomas Sporer. Wave field synthesis in the real world: Part 1 - in the living room. In *AES 114th Convention Paper*, 2003.

[12] Thomas Sporer. Wave field synthesis - generation and reproduction of natural sound environments. In *Proceedings of the* 7^{th} *International Conference on Digital Audio Effects (DAFx'04)*, Naples, Italy, October 2004.

[13] Sascha Spors, Achim Kuntz, and Rudolf Rabenstein. An approach to listening room compensation with wave field synthesis. In *Proceedings of the AES 24th International Conference*, pages 70–82, Banff, Alberta, Canada, Juny 2003.

[14] Edward Nico Gerard Verheijen. *Sound reproduction by wave field synthesis.* PhD thesis, TU Delft Netherlands, 1997.

Utilization of Radio-Location Methods for XY-Coordinate Tracking of Audio Source

Jiri Schimmel, Jiri Kouril
Dep. of Telecommunications FEEC
Brno University of Technology
Brno, Czech Republic
e-mail: schimmel@feec.vutbr.cz

Abstract. This paper describes a project that deals with a practical examination of a system for the tracking multiple audio sources, mainly speakers or singers, in a 2D space that uses radio-location methods. Acquired and stored XY coordinates can be used to control algorithms of multi-channel audio processing, e.g. surround panning in the playback of the performance in the home-theatre environment. The basic requirements are low price of the system, easy integration into the current equipment of music halls, theatres and other PA systems, and minimum inconvenience to the performer.

1. Introduction

There are numerous tracking systems based on the ultra-sound, sensor, and other principles, which however require the tracked object to be equipped with a special device. The project is based on the fact that in most cases every performer on the stage is equipped with a UHF transmitter for wireless transmission of the audio signal from microphone and each of those transmitters is set to a different carrier frequency. The performer does not need to be equipped with additional device when this transmitter is used for tracking.

2. Tracking Method

During design of the system of an audio signal source tracking, several potential tracking principles were compared as well as a suitable representation of the result. The triangular or the hyperbolic method is mostly used for radio transmitter location. The triangular method was chosen because the hyperbolic method requires time stamps to be transmitted from the transmitter. System is able to track audio signal source in 2D space and acquires and stores position that is represented by XY coordinates.

2.1. Modified Triangular Method

The triangular method tracks the location according to the radio signal strength scanned at three points. A direction, in which a transmitting object lies, is found at these points. Position of the object is given by intersection of the measured angles (azimuths) (see Figure 1.) [1]

This method fails if the object lies in the direction of the location base. Third receiver station is added to eliminate this limitation and to improve accuracy of the position detection.

In the project, modified triangular method was used: four receivers are used instead of three ones to allow the utilization of more complex tracking methods. The system tracks the location according to the radio signal strength scanned at these four points. Relative strength of the signal in comparison with maximum strength of the signal represents the relative distance from the transmitter to the receiver. This distance is considered to be radius of a circle with centre at the transmitter position. The actual position where the wireless source of the signal is located is determined by the intersection of at least three circles.

The maximum strength if the signal is determined during the calibration process: the transmitter is placed in the corner of the scanned room (i.e. beside the receiver) and measured strength of the signal is considered to be the maximum. The relative strength is in the range of 0 to 1.

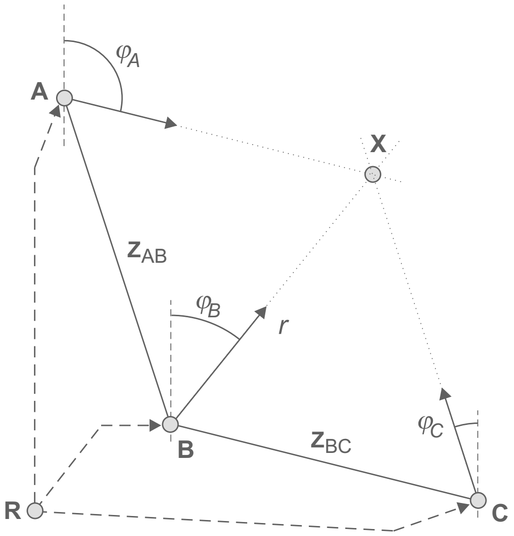


Figure 1: Position tracking using triangular method

Receivers are located in the corners of the square room itself (Figure 2) and three most high values from measured ones are found prior to computation of the position. This selection is performed mainly for decreasing of the coordinate computation error. For example, in the case of a speaker with a microphone located in the sacral area, the radio-wave shade of the body causes a situation, when one receiver measures less signal intensity than in the free field.

2.2. 2D-Coordinates Representation

Following example shows advantage of the 2D-coordinates representation: there is another position tracking system that gives information about the angle and radius of the tracked position that referred to a specific listener position. When the listener position changes, it is necessary to evaluate both values again.

For that reason it is optimal to have information about X and Y position that could be is easily recalculated to any other position or transferred to another system of coordinates according to the used algorithm. Recalculation or conversion has to be performed by the subsequent processing algorithm itself because type of this algorithm is not known to the tracking method described below.

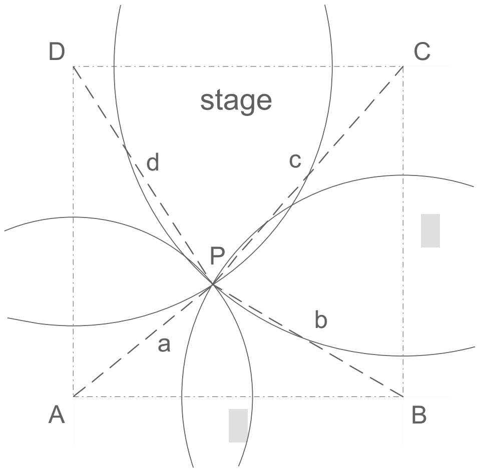


Figure 2: Location of measuring points

3. Realization

The tracking equipment designed consists of a master computing unit and four slave units that measure the strength of transmitter signal strength. These units are equipped with a receiving antenna each, placed in the corners of the area in which the tracked object is about to travel (e.g. the stage). This solution assumes that the tracked object will move in the rectangular or square areas. If the sound should reach beyond the speakers, a reverberation algorithm has to be used [5].

The interconnection of slave units with the master unit is realized using the RS485 interface. The basic part of the framework is the ATmega2560 microcontroller [2], which performs all communication and computational procedures, and the TH71221 tuned UHF receiver/transmitter equipped with the RSSI (Receive Signal Strength Indication) output (see Figure 3).

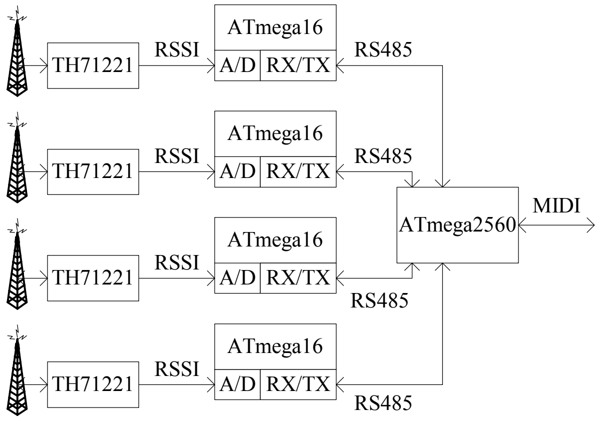


Figure 3: Tracking system architecture.

The master unit sends request to all slave units in sequence with intervals of 1.2 µs via the RS485 interface. The request is one-byte message with most significant bit set. Other bits are used for request identification (currently only one type).

After the request is received, each slave unit reads the actual relative strength value (10 bits) and stores it using two data bytes into TX register.

3.1. TH71221 Transceiver

The TH71221 is a single chip FSK/FM/AM transceiver integrated circuit. It is designed to operate in low-power multi-channel programmable or single-channel stand-alone, half-duplex data transmission systems. It can be used for applications operating in the frequency range of 300 MHz to 930 MHz [3]. Since the wireless microphone systems use the FM signal in the range of 500 MHz to 800 MHz, the TH71221 is suitable for most of them.

The RF front-end of the receiver part of the TH71221 is a super heterodyne configuration that converts the input radio frequency into an intermediate frequency signal. The most commonly used intermediate frequency is 10.7 MHz but intermediate frequencies in the range of 0.4 to 22 MHz can be also used. The front-end consists of a low-noise amplifier, mixer, and an intermediate frequency limiting amplifier with received signal strength indicator [3]. The local oscillator signal for the mixer is generated by the PLL frequency synthesizer.

The transceiver can operate in two different user modes. It can be used either as a 3wire-bus-controlled programmable or as a stand-alone fixed-frequency device. After power up, the transceiver is set to Standalone User Mode. Using the logic level change at control pin, the transceiver enters into Programmable User Mode. In this mode, the user can set any PLL frequency or mode of operation by the serial control interface [3]. There are four registers available to set the voltage-controlled oscillator frequencies in receive and transmit mode. The channel frequency depends on the external crystal frequency. The feedback divider ratio in RX operation mode can be set using 17 NR bits in C-word of the control register.

3.2. Calculation of Coordinates

After the master unit receives relative strength values of the signal from all slave units, it performs calculation of the position using the triangular method. Current algorithm selects three highest values and computes the *x* and *y* coordinates using following simple equations

$$\begin{aligned} x &= \frac{1023^2 + a^2 - b^2}{2046}, \\ y &= 1023 - \sqrt{\left|c^2 - (x - 1023)^2\right|} \end{aligned} \qquad (1)$$

where *a*, *b*, and *c* are highest values of relative strength of received signal (see Figure 2). After the testing of the system in the real environment, more complex algorithm will be developed based on an analysis of the acquired input data.

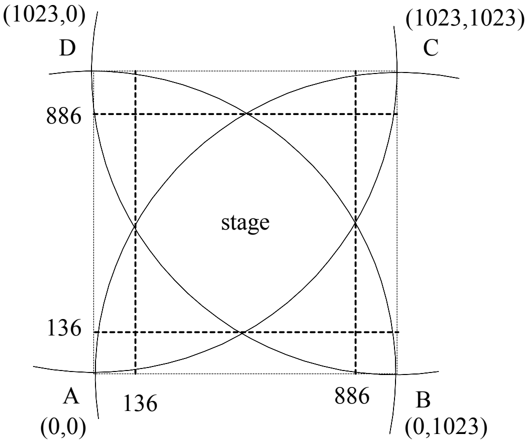


Figure 4: Ideal borders of transmitter movement.

Figure 4 shows coordinates of position of the receivers in a square area. As can be seen, there are four areas where all relative strength values are zero so the transmitter cannot be tracked in these areas. Figure 4 also shows the ideal borders of transmitter movement, which lies on the relative strength values of 136 and 886.

3.3. Master Interface

The resulting tracked position of the transmitter, and thus the performer's position as well, is sent into the target system (e.g. personal computer) using the MIDI (Musical Instrument Digital Interface) protocol [4] in the form of XY coordinates.

For wide compatibility with the target MIDI equipment, the system can be configured to send one of three types of MIDI messages: pan controllers number 9 and 10, non-registered parameter numbers, and universal system exclusive data. The resolution of relative strength is 10 bits so two MIDI data bytes have to be used (see Figure 5) because only seven bits can be used id the MIDI data byte [4].

LSB	0	v	v	v	v	v	v	v
MSB	0	r	r	r	r	v	v	v

Figure 5: Coding of XY coordinates into MIDI data bytes (v – value bits, r – reserved bits)

3.3.1. Pan Controller Mode

In the MIDI specification [4], the controller number 9 is undefined and controller number 10 is used for left-right panning. Those controller numbers are the MSB controller numbers. When higher resolution is required, the LSB controller numbers 41 and 42 (MSB controller number + 32) are used in addition. The system uses the controller numbers 9/41 for front-back panning (Y coordinate) and controller numbers 10/42 is used for left-right panning (X coordinate). Figure 6 shows the data sequence.

BnH	09H	f-b msb	29H	f-b lsb	0AH	l-r msb	2AH	l-r lsb

Figure 6: MIDI message for pan-controller mode (n – MIDI channel number, f-b – front-back panning, l-r – left-right panning).

For the MIDI voice and mode messages the running status can be used. When status byte #Bn is received and processed, the receiver will remain in that status until a different status byte is received Therefore, if the same status byte would be repeated, it can optionally be omitted so that only the data bytes need to be sent [4].

3.3.2. Non-Registered Parameter Number Controller Mode

Non-Registered Parameter Numbers (NRPN, MIDI controller numbers 62H/63H) are used to represent sound or performance parameters and may be assigned as needed by individual manufacturers. The basic procedure for altering a parameter value is to first send the Non-Registered Parameter Number corresponding to the parameter to be modified, followed by the Data Entry (controller number 6H/26H), Data Increment (controller number 60H), or Data Decrement (controller number 61H) value to be applied to the parameter [4].

After the reception on Non-Registered Parameter Numbers has been enabled, the receiver should wait until it receives both the LSB (62H) and MSB (62H) for a parameter number to ensure that it is operating on the correct parameter. Once a new Parameter Number is chosen that parameter retains its old value until a new Data Entry, Data Increment, or Data Decrement is received. The designed tracking system uses the Data Entry controller number only. The parameter numbers for the front-back and left-right panning are user-configurable. Figure 7 shows the data sequence.

BnH	63H	p1 msb	64H	p1 lsb	06H	f-b msb	26H	f-b lsb
	63H	p2 msb	64H	p2 lsb	06H	l-r msb	26H	l-r lsb

Figure 7: MIDI message for NRPN-controller mode (n – MIDI channel number, p1 – parameter 1 number, p2 – parameter 2 number, f-b – front-back panning, l-r – left-right panning).

3.3.3. System Exclusive Mode

System Exclusive messages starts with SysEx status byte (F0H), can contain any number of data bytes, and can be terminated either by an End of Exclusive (EOX, F7H) or nay other status byte.

System Exclusive messages have their own special format according to an assigned manufacturer's identifier. To avoid conflict with non-compatible Exclusive messages, a specific ID number is granted to manufacturers of MIDI instruments by the MMA or JMSC organizations. Special identifier 7DH is reserved for non-commercial use (e.g. research, etc.) [4]. Figure 8 shows the data sequence of System Exclusive message of the designed system. To avoid conflict with another device with the same manufacturer identifier (7DH), the messages contains device identifier, which is user-configurable. The fourth data byte (01H) is the message identifier.

F0H	7DH	ID	f-b msb	f-b lsb	l-r msb	l-r lsb	F7H

Figure 8: MIDI message for System Exclusive mode (ID – device identifier, f-b – front-back panning, l-r – left-right panning).

3.4. System Configuration

The system configuration uses MIDI System Exclusive messages described above. There are two control messages: Mode Settings and Frequency Settings, which are differentiated via message identifier. Figure 9 shows the Mode Settings message.

F0H	7DH	ID	02H	mode	p1 msb	p1 lsb	p2 msb	p2 lsb	F7H

mode	0	r	m	m	n	n	n	n

Figure 9: MIDI message for Mode Settings (ID – device identifier, m – data mode, n – MIDI channel, r – reserved, p1/p2 – parameter 1/2 number for NRPN mode).

The fourth data byte (02H) is the message identifier. If the data mode is set to NRPN type, four bytes with NRPN follows the mode data byte.

Figure 10 shows the Frequency Settings message. The fourth data byte (03H) is the message identifier followed by the number of channels used. In the current system, only one channel is allowed. However, the future work will be focused to the multi-channel tracking using TDM tracking methods. Up to 127 channel frequencies can be sent using the Frequency Settings message. However, we assume that the number of scanned channels will be considerable lower in practice.

F0H	7DH	ID	03H	n	f1.1	f1.2	f1.3	...	F7H

Figure 10: MIDI message for Frequency Settings (ID – device identifier, n – number of channels).

Each channel frequency is transferred as 17 NR bits of the control register of the TH71221 transceiver (see above) using thee MIDI data bytes (7+7+3 bits).

4. Conclusion

The first development period of the project has been finished recently and the system is prepared for the testing phase in the real environment. This phase should prove if any application of the system is possible at all, in how wide areas and with what sensitivity and accuracy the system can work. This phase should supply input data for subsequent development of tracking algorithms. The turn of the performer’s body, obstructions in the area under tracking, etc. will have an effect on the strength of received signal.

In comparison with other tracking position systems currently used in multimedia, such as ultra-sound systems, sensor systems installed inside the stage floor, systems using IR cameras, and other principles, the designed system has advantages of low-price construction, easy integration into the current equipment of music halls, theatres and other PA systems, and minimum inconvenience to the performer.

Acknowledgement

This paper is supported by project No. 102/07/P505 of the Grant Agency of the Czech Republic and project No. FT-TA3/010 of the Ministry of Industry and Trade of the Czech Republic.

References

[1] J. Kouril, *Audio Signal Source Tracking for Audio Effect Control* (in Czech), Diploma Thesis, Brno University of Technology, Brno, 2007.

[2] *ATmega640/1280/1281/2560/2561 Datasheet*, document version 2549KS-01¬/07, Atmel Corporation, 2007.

[3] *TH1221 27 to 930 MHz FSK/FM/ASK Transceiver Datasheet*, Rev. 005. Melexis Microelectronic Integrated Systems, 2007.

[4] *The Complete MIDI 1.0 Detailed Specification*, document version 96.1, MIDI Manufacturers Association, Japan MIDI Standard Committee, 1997.

[5] E. Zwicker, *Psychoacoustics – Facts and Models*, Second Edition. Springer, Berlin, 1999.

Binaural Rendering for Enhanced 3D Audio Perception

Christos Tsakostas[1], Andreas Floros[2] and Yiannis Deliyiannis[2]

[1]HOLISTIKS Engineering Systems, Digeni Akrita 29, 122 43 Athens, Greece
tsakostas@holistiks.com

[2]Dept. of Audiovisual Arts, Ionian University, 49 100 Corfu, Greece
{floros, yiannis}@ionio.gr

Abstract. Despite the recent advantages on multichannel audio coding and playback technologies, 3D audio positioning is frequently required to be performed over legacy stereo loudspeaker setups, due to certain application limitations and restrictions mainly presented in portable and low cost systems. In this work, a 3D audio platform is introduced which allows the real-time conversion of any stereo audio signal to a high-quality immersive audio stream using binaural technology. Due to a number of advanced binaural processing algorithms and features, the proposed conversion platform demonstrates perceptually improved 3D audio spatialization, rendering it suitable for implementing high-quality immersive audio applications.

1. Introduction

Two-channel stereophony has been the basic representative of audio reproduction systems for more than half a century. Today, the advent of high-resolution storage formats (such as Super Audio CD [1], DVD-Video/Audio BluRay and HD-DVD) has significantly boosted the proliferation of many surround sound coding schemes used for audio-only, movies, home theatre, virtual reality and gaming applications. However, despite the advantages in multichannel sound, a number of parameters have to be considered for developing high-quality spatial audio representation: a) the volume of the existing stereo recordings and stereo reproduction systems, including TV sets, CD players, laptop computers and mobile phones b) the large number of loudspeakers with specific placement specifications that are usually required for multichannel reproduction (e.g. for 5.1, ambisonics or wavefield synthesis setups) c) the frequently complicated loudspeakers' cable interconnections and d) the decreased application portability, due to the receiver position limitations imposed by the multichannel coding schemes.

Today, the advent of many portable devices (including small audio/video players, Personal Digital Assistants – PDAs and mobile phones and computers) usually equipped with ordinary stereo loudspeakers necessitates the development of low-cost, low-power algorithms for 3D audio immersion using legacy stereo setups. Towards this perspective, binaural technology [2] represents a very attractive format, as it allows accurate 3D audio spatialization by synthesizing a two-channel audio signal using the well-known Head Related Transfer Functions (HRTFs) between the sound source and each listener's human ear [3], [4]. Hence, only two loudspeakers or headphones are required for binaural audio playback.

Binaural technology was recently employed for parametric MPEG surround coding [5], where the spatial information is extracted from multichannel audio and a downmix is produced and transmitted together with the low-rate spatial side information, allowing backward compatible representation of high quality audio at bitrates comparable to those currently used for representing stereo (or even mono) audio signals. In this work, a novel real-time 3D audio enhancement platform is presented which converts any stereo audio material to a high-quality immersive audio stream using binaural rendering. The proposed platform incorporates novel HRTF equalization methods that significantly improve the spatial position perception of the active sound sources compared to previously reported methods [6]. Efficient crosstalk cancellation algorithms are also incorporated for stereo loudspeaker support, with a large number of FIR coefficients (2048 at 44.1kHz sampling frequency). Moreover, Sound Field Models are supported, allowing the definition of fully customizable virtual auditory environments or the selection of predefined virtual world templates. As it will be described later, an unlimited number of virtual sound sources can dynamically be linked to the channels of the stereo input, allowing accurate and fully parameterized 3D audio positioning.

The rest of this paper is organized as following: In Section 2, an overview of the binaural technology is provided, followed by the brief description of the proposed algorithm for 3D enhancement of stereo audio signals presented in Section 3. In Section 4, a typical implementation of the 3D audio enhancement technique is provided. Finally, Section 5 concludes this work.

2. Binaural technology overview

It is well known that using binaural technology the accurate spatial placement of any virtual sound source is achieved by filtering monaural recorded (or synthesized) sound with appropriately selected Head Related Transfer Function (HRTFs) [7]. In general, the latter functions describe the paths between a sound source and each ear of a human listener in terms of a) the interaural time difference (ITD) imposed by the different propagation times of the sound wave to the two (left and right) human ears and b) the interaural level difference (ILD) introduced by the different propagation path lengths, as well as the shadowing effect of the human head. Both ITD and ILD sound localization cues result into two different sound waveforms arriving to the human ears, allowing the perception of the direction of any active sound source.

When using binaural technology, the above basic localization cues are incorporated into HRTFs, which represent directional-dependent transfer functions between the human listener's ear canal and the specific sound source placement [8]. Hence, convolving the mono sound source wave with the appropriately selected pair of HRTFs produces the sound waves that correspond to each of the listener's ears. This process is called binaural synthesis. Binaural synthesis can be also combined with existing sound field models producing binaural room simulations and modelling. This method facilitates listening into spaces that only exist in the form of computer models. In more detail, the sound field models can output the exact spatial-temporal characteristics of the reflections in a space. In this case, the summation of binaural synthesis applied to each reflection produces the Binaural Room Impulse Response. Finally, the binaural left and right signals are reproduced using headphones or a pair of conventional stereo loudspeakers. In the latter case, the additional undesired crosstalk paths that transit the head from each speaker to the opposite ear must be cancelled using crosstalk cancellation techniques [9].

3. 3D Audio spatialization using binaural rendering

By employing the above binaural synthesis approach, one can create virtual sound sources placed around a listener within an open or closed space. This is the well-known concept of binaural rendering, which allows the creation of 3D virtual sound environments. More specifically, as explained previously, the convolution of the original audio signal with a pair of HRTFs for each sound source and the final mixing of the resulting binaural signals produces the desired 3D audio perception.

In order to achieve 3D audio enhancement of typical stereo material, the stereo audio input must be mapped to a number of virtual sound sources appropriately placed into the desired virtual world. Figure 1 illustrates this mapping procedure. Prior to mapping, the input stereo signal is pre-processed in order to produce the necessary audio streams that will be mapped to the selected virtual sound sources. Apart from the profound audio streams of the left and right channels, more audio streams can be acquired such as the summation or the difference of them.

Each audio stream is then mapped to an arbitrary number of virtual sound sources. The benefit of this approach is that different aspects of the audio signal can be freely manipulated, while any spatial effect can be theoretically achieved. For example a low passed version of the (L+R)/2 audio stream can be mapped to a virtual sound source mimicking a low frequency playback unit (subwoofer).

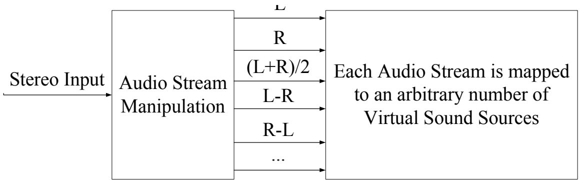

Stereo Input
Audio Stream Manipulation
L
R
(L+R)/2
L-R
R-L
...
Each Audio Stream is mapped to an arbitrary number of Virtual Sound Sources

Figure 1: Stereo audio signal to virtual sound sources mapping

After audio stream to virtual sound source mapping, the binaural impulse responses are calculated for both left and right ears, and the final binaural signal is derived by convolving them with the derived audio streams as described in the previous Section. Obviously, in the case of a closed virtual world with specific geometry and material properties, the binaural room impulse response are calculated and employed for deriving the final binaural signal.

4. Implementation and performance evaluation

Using the proposed 3D audio enhancement platform, all the required binaural processing of the original stereo material is performed using the Amphiotik Technology framework developed by the authors and presented in detail in [10]. The benefit of this approach is that the Amphiotik Technology core (namely the Amphiotik 3D Audio Engine) offers the capability of rapid 3D-Audio applications development, yet preserving a carefully designed balance between authenticity and real-time operations / calculations. More specifically, the Amphiotik 3D Audio Engine incorporates state-of-the-art binaural processing algorithms, such as a novel algorithm for HRTFs equalization, cross-talk cancellation techniques and room acoustics modeling for accurate acoustic representation of virtual environments. The Amphiotik 3D Audio Engine state-machine is also responsible for the signal routing that must be performed in order to perform all calculations required for producing the binaural signal within the defined virtual world.

The communication of the 3D audio enhancement platform with the Amphiotik 3D audio engine is performed using the Amphiotik API, which provides easy to use software methods for defining the binaural model and the virtual world parameters in real-time. Figure 2 illustrates the Amphiotik Technology architecture employed. The Amphiotik API provides the necessary functions for the definition of the overall virtual auditory environment, that is: (a) the geometry and materials of the virtual world, (b) the sound field model and (c) the virtual sound sources and receivers characteristics and instantaneous position. In addition, it provides functions that interact directly with the 3D-Audio engine to parameterize various aspects of the engine such as the HRTF data set to be used, the cross-talk cancellation algorithm activation, the headphones equalization as well as user-defined parametric frequency equalization.

The shape of the virtual world can be arbitrary, but for the shake of real-time processing in moderate power computers, “shoebox” like spaces are better supported. For this case, the API provides simple functions for defining the dimensions of the room (length, width and height) and the materials (absorption coefficients) of each surface. An internal materials database is utilized for the re-use of the above materials parameters.

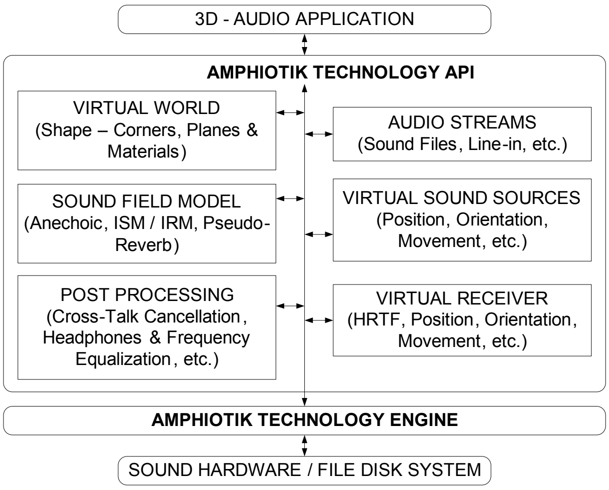

3D - AUDIO APPLICATION
AMPHIOTIK TECHNOLOGY API
VIRTUAL WORLD
(Shape – Corners, Planes & Materials)
SOUND FIELD MODEL
(Anechoic, ISM / IRM, Pseudo-Reverb)
POST PROCESSING
(Cross-Talk Cancellation, Headphones & Frequency Equalization, etc.)
AUDIO STREAMS
(Sound Files, Line-in, etc.)
VIRTUAL SOUND SOURCES
(Position, Orientation, Movement, etc.)
VIRTUAL RECEIVER
(HRTF, Position, Orientation, Movement, etc.)
AMPHIOTIK TECHNOLOGY ENGINE
SOUND HARDWARE / FILE DISK SYSTEM

Figure 2: Architecture of the Amphiotik Technology framework

The API also allows the definition of one virtual binaural receiver, while there are no limitations on the number of the defined sound sources. Both the virtual receiver and sound sources can be placed arbitrarily and in real-time in the 3D virtual auditory environment, concerning both their position and orientation. An arbitrary number of audio streams can be defined, which are linked to the virtual sound sources. An audio stream can be associated with more than one virtual sound sources. Audio streams are usually sound files on the local disk system but they can also originate from the soundcard’s line-in input or even an Internet media file link.

Using the Amphiotik 3D audio engine, the acoustical environment modeling can be performed using one of the following sound field models: (a) anechoic, (b) early part, and (c) early part & pseudo-reverb. For the anechoic case reflections are not considered, consequently the geometry of the room is ignored. On the other hand, the early part is simulated by means of the "Image Source Method" and "Image Receiver Method" [11]. The order of the reflections can be altered in real-time and its maximum value is limited to five. Early part & pseudo-reverb uses a hybrid algorithm in which the early part is estimated as described earlier and the reverberation part (i.e. the late part) of the room impulse response is estimated with digital audio signal processing algorithms. Specifically, two reverberation algorithms are currently supported: (a) Schroeder [12] and (b) Moorer [13]. According to the Schroeder algorithm the late part is calculated by the means of comb-filters and all-pass filters, whilst for the case of the Moorer technique the late part is approximated with an exponentially decaying white noise. The reverberation time is calculated using the Sabine equation [14]. A proprietary algorithm has been employed in order to combine the binaural early part with the monaural late part.

Figure 3 depicts a general overview of the Amphiotik 3D-Audio Engine. For each pair of virtual receiver and sound source, a binaural IR is calculated, taking under consideration their instantaneous positions, orientation and room geometry and materials as well, if the early part option is enabled. Real-time convolution is accomplished by the means of un-partitioned and partitioned overlap-and-add algorithms. Each convolution produces two channels: Left (L) and Right (R). All the L and R channels, produced for each virtual sound source, are summed up producing finally only two channels.

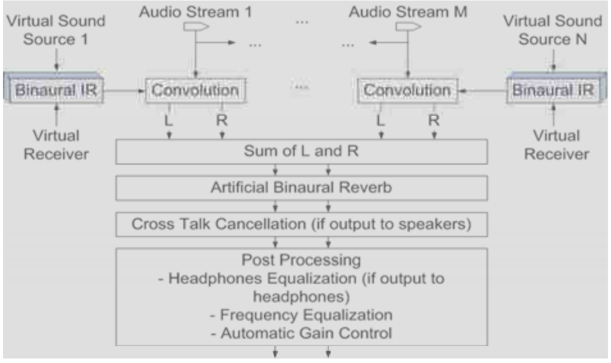


Figure 3: Amphiotik 3D audio engine general structure

Crosstalk cancellation (CTC) is applied if the audio playback is performed over stereo loudspeakers. CTC filters are built, in real-time, with HRTFs. The Amphiotik API gives the capability to select a different set of HRTFs for the crosstalk cancellation processing than the one used for spatialization. In general, CTC filters built with HRTFs impose the problem that the low-power low frequencies are excessively amplified and vice-versa. Two strategies have been used in order to overcome this problem: (a) the employment of band-limited CTC and (b) a special equalization method called "Transaural Equalization". Band-limited CTC simply partially overcome the above problem by not using the very low and the very high frequencies. The outcome is that musicality becomes significantly better, and at the same time the loss of the very low and the very high frequencies is not particularly perceptible. Transaural equalization on the other hand, is based on post-filtering of the transaural audio channels, in order to approximate the magnitude that they would have if listening over headphones was selected. In addition, the software gives the capability to apply the crosstalk cancellation filters directly to the stereo input without prior processing through the 3D-Audio engine. It is also important to note that the Amphiotik API gives the capability to select non-symmetric loudspeakers positions (e.g. for loudspeakers in cars).

Additionally, as it shown in Figure 3, post-equalization is applied to the synthesized binaural signal, which may optionally include headphones equalization, user-defined frequency equalization and Automatic Gain Control (AGC). Pre-equalization is also supported for the stereo audio signals before they are spatialized.

Concerning the sound motion simulation, a time-varying filtering method is employed that minimizes the need of additional computations for accurate moving sound-sources representation. This mechanism additionally takes into account psychoacoustic criteria and cues for perceptually optimizing the 3D audio representation performance [15].

Finally, for effective real-time operation and interaction with the user, the Amphiotik engine checks for any possible change of the parameters in time frames, which are defined by the block length used (typically 512 - 8192 samples at a sampling rate equal to 44.1 KHz) and re-initializes all the appropriate modules.

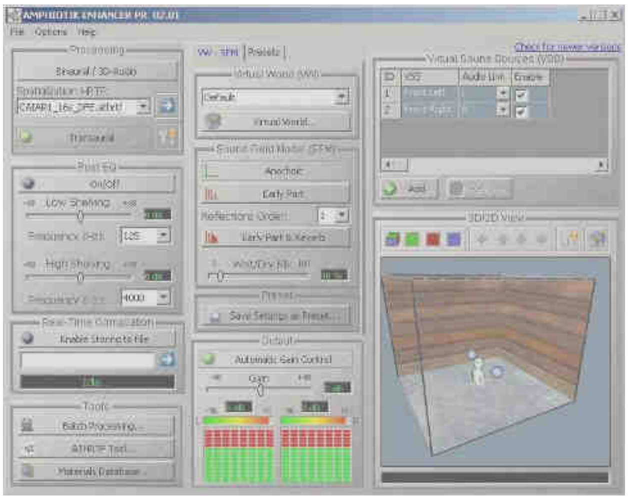
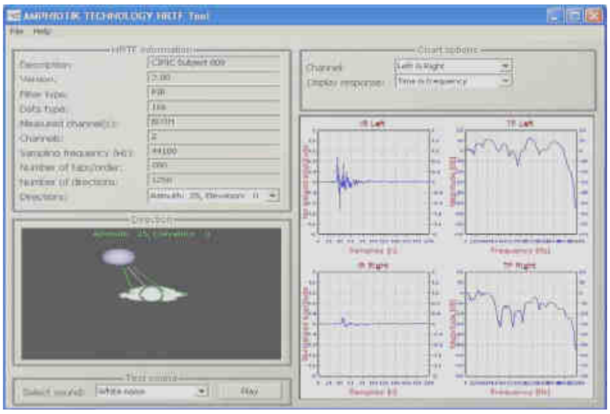


Figure 4: The Amphiotik Enhancer module

In order to demonstrate the capabilities and the performance of the 3D audio enhancement platform, the Amphiotik Enhancer plug-in was developed during this work, which uses the Amphiotik API and engine (Figure 3). As mentioned previously, its main purpose is to enhance the standard stereo audio signals, offering a much more pleasant 3D acoustical experience. The Amphiotik Enhancer module is compatible and can be plugged in several popular audio hosts (like Winamp, Windows Media Player, VST Hosts, and DirectX).

The Amphiotik Enhancer incorporates all the API features that were described previously, through a graphical user interface (GUI). Typical user interaction procedures include manipulation and definition of the audio streams, definition and placement of the virtual sound sources and the virtual binaural receiver, the analytic description of the desired sound field models, as well as the selection of the HRTF library (as shown in Figure 5). Cross-talk cancellation or headphones playback options are also available while frequency equalization can be additionally selected. It is important to note that the GUI supports 3-Dimensional view of the virtual world, for better user-perception of the intended simulations and the final binaural playback.



Description: CIPIC Subject 009
Version: 2.00
Filter type: FIR
Data type: 16b
Measured channel(s): BOTH
Channels: 2
Sampling frequency (Hz): 44100
Number of taps/order: 200
Number of directions: 1250
Directions: Azimuth: 25, Elevation: 0
Channel: Left & Right
Display response: Time & frequency
Azimuth: 25, Elevation: 0
Select sound: White noise

Figure 5: HRTF library selection GUI

The perceptual performance of the proposed 3D audio enhancement platform was evaluated through a number of subjective audio tests. During these tests, the Amphiotik Enhancer module was used for producing enhanced 3D audio material from typical stereo (CD-quality) recordings. The responses of the experienced audience that participated into these tests have shown a significant improvement of the perceived audio immersive impression. Moreover, from these tests it was found that for real-time operation and for sound field models activated, up to 8 virtual sources in average can be defined, allowing the high-quality and flexible stereo to virtual sound source mapping.

5. Conclusions

The advent of many multichannel audio coding and playback formats and schemes is nowadays leading towards the surround sound prospect. However, due to the large volume of existing legacy stereo audio material as well as the usage of electronic / consumer playback devices equipped with stereo loudspeakers, the employment of appropriate signal processing algorithms for 3D enhancement of stereo audio signals is now more demanding.

In this work, a novel 3D audio enhancement platform is presented which converts any stereo audio material to a high-quality immersive audio stream. The conversion is performed using binaural rendering, which allows the creation and accurate representation of various forms of virtual auditory worlds. The proposed platform may be employed for the development of both software plug-ins as well as for hardware-based applications, using an Application Protocol Interface (API) available. Here, it is shown that, in terms of 3D audio perception, the proposed platform achieves high degree of authenticity, due to a number of optimized algorithms, such as a novel HRTF equalization scheme. Moreover, the 3D enhancement platform is optimized for real-time operation, allowing its employment to consumer devices, game engines etc.

Future research work will consider in more detail moving virtual sound sources, taking into account additional psychoacoustic parameters and cues that are required for authentic sound reproduction, such as the Doppler effect.

References

[1] E. Janssen and D. Reefman, *“Super-Audio CD: an introduction”,* IEEE Signal Processing Mag., Vol. 20, No. 4, pp. 83–90 (2003)

[2] H. Møller, “Fundamentals of Binaural Technology”, Appl. Acoustics, Vol. 36, pp. 171 – 218, (1992).

[3] H. Møller, M. F. Sørensen, D. Hammershøi, C. B. Jensen, *“Head-related transfer functions of human subjects”,* J. Audio Eng. Soc., Vol. 43, No. 5, pp. 300-321 (1995)

[4] E. Wenzel, M. Arruda, D. Kistler and F. Wightman, *“Localization using nonindividualized head-related transfer-functions”,* J. Acoust. Soc. Am., Vol. 94, pp. 111-123 (1993)

[5] J Breebaart, J. Herre, L. Villemoes, C. Jin, K. Kjorling, J. Plogsties and J. Koppens, “Multichannel Goes Mobile: MPEG Surround Binaural Rendering”, AES 29th International Conference, Seoul, Korea, (2006)

[6] S. Olive, *“Evaluation of five commercial stereo enhancement 3D audio software plug-ins”*, Presented at the 110th Convention of the Audio Engineering Society, Amsterdam, The Netherlands, Preprint 5386 (2001)

[7] J. Blauert, *“Spatial Hearing: The psychophysics of human sound localization”,* Revised edition, Cambridge, Massachusetts, The MIT Press, (1997)

[8] V. Pulkki, *“Virtual Sound Source Positioning Using Vector Base Amplitude Panning”,* J. Audio Eng. Soc., Vol. 45, No. 6, pp. 456 – 466 (1997)

[9] A. B. Ward, G. W. Elko, *“Effect of loudspeaker position on the robustness of acoustic crosstalk cancellation”*, IEEE Signal Processing Letters, Vol. 6, No. 5, pp. 106-108 (1999)

[10] Ch. Tsakostas and A. Floros, “Optimized Binaural Modeling for Immersive Audio Applications”, Presented at the 122nd Convention of the Audio Engineering Society, Vienna, Preprint 7100 (2007)

[11] C. Tsakostas, *“Image Receiver Model: An efficient variation of the Image Source Model for the case of multiple sound sources and a single receiver”* presented at the HELINA Conference, Thessaloniki Greece (2004)

[12] M. R. Schroeder, *“Natural Sounding Artificial Reverberation”*, J. Audio Engineering Society, Vol. 10, No 3, (1962)

[13] J. A. Moorer, *“About This Reverberation Business”*, Computer Music Journal, Vol. 3, No 2. (1979)

[14] W. C. Sabine, *"Reverberation"* originally published in 1900. Reprinted in Acoustics: Historical and Philosophical Development, edited by R. B. Lindsay. Dowden, Hutchinson, and Ross, Stroudsburg, PA, (1972)

[15] Ch. Tsakostas and A. Floros, “Real-time Spatial Representation of Moving Sound Sources”, to be presented at the Audio Engineering Society 123rd Convention, New York, (2007)

Effective interactive design through quick and flexible prototyping.

Nigel Papworth, Mats Liljedahl, & Stefan Lindberg
nigel@tii.se mats.liljedahl@tii.se stefan.lindberg@tii.se

Sonic Studio, Interactive Institute, Sweden

Abstract:
One of the problems with designing new and untried applications, is the sheer effort needed to produce a testable prototype. Many projects get bogged down with production problems and hardware/software issues before they can even be tested for their basic functionality. This creates delays and frustrations within the project structure as well as creating unnecessary pressures on the project structure itself.
This is equally true for the two main types of prototype/application created in design research:
1. a potential new product
2. a viable test bed to test an hypothosis.

This paper investigates approaches, both practical and philosophical, to design processes and construction techniques that endeavour to bypass the most common pitfalls and stumbling blocks to a working prototype.
In the main, the paper focuses on designing with 'Audio' as an effective tool for quick, effective and emotive prototyping.
Practical examples, both successful and catastrophic, are used to illustrate these aspects. In specific we will be comparing two audio based games, Journey and Beowulf, looking at their respective weaknesses and strengths.
These prototyping techniques are realised, in the main, through easily available tools and recording techniques, based on standard audio-visual techniques. This provides a neat fast-track to developing ideas into viable applications.

However while we can utilize a vast array of existing technologies to enable fast proto-typing, the most crucial element in successfully creating testbeds and bringing them before a test group is the intellectual approach to experiment design. Understanding the fundamental processes and isolating the essential questions the application is intended to answer has a direct bearing on our ability to channel available creative and productive energies with the maximum focus. Audio provides us with emotive tools highly suited to performing these kinds of functions.

The advantage of taking this approach is a quick, flexible and, not least, economic route to getting vital early answers to practical research problems in IT user-based projects.

1. Introduction

There is a huge amount of difficulty involved in initiating design research projects (though this can as easily apply to behavioural studies where some practical test bed is needed).
Often, a project is faced with two choices, to create a clear specification from pre-determined project goals or develop one or a series of test-beds and/or prototypes designed to lead the project towards a satisfactory conclusion.
While specifications can initially seem a clearer and more concrete route to take, this is often less effective and more costly than creating a prototype[3], especially when the goal of a project is research rather than pure product development.

For these kinds of projects to be truly viable, some kind of appropriate prototype must be created to allow for flexible testing and assessment. The irony of this is that the prototype itself, if over ambitious or poorly conceived, can demand the lion's share of the effort and resources that are available to the research team; It can even be argued that these kinds of over ambitious prototypes fall into the category of a specification based pre-product[3]. Those responsible for the initial design-decisions in the team must therefore be very sure that:

A. The finished prototype will work as expected.
B. The finished prototype is capable of answering the research questions.
C. The development requirements of the prototype are in balance with the overall project plan.

One trap that research teams can easily fall into, is the idea that new research must mean innovative and groundbreaking elements at all levels of the project. The tendency to pour creative resources into trying to meet this ambition can, in many cases, prove highly counter-productive.
This paper is an attempt at looking at alternative methods that can provide a better resource balance and allow for a faster pipeline to a project's desired result.

2. Comparison: 'Journey-Beowulf'

In order to be as clear as possible and to illustrate the pitfalls and common mistakes involved in these kinds of projects we will be comparing two actual projects, one a total disaster and one a qualified success (according to the reaction the team has received from their peers).

3. Project A. 'Journey'

The first project 'Journey' was designed to investigate the role of sound, and especially music, in delivering, controlling and

relating gaming content to the user. It was designed to control the actual rendering of the music using the game state information generated by the player's actions within the game itself. This in turn would be converted to trigger-logic, which would steer and to a large extent create the music and audio effects content of the game on the fly.
The game was designed as a high end application for mobile phones and the project was intended to produce a working prototype within six months as part of it's deliverables package. This was considered imperative to the success of the project. The intention was to lift the audio content from its traditional supporting role, and allow it to share the role of principle driver for the gameplay with the graphic presentation.

Because of the nature of the deliverable a huge potential problem had been created, both practically and tactically.
The deliverable mixed what was essentially a working software prototype with a research test bed designed to address a number of research questions. This was to have dire consequences.

The game was envisaged as an open landscape of both visual and audio stimuli combining to make a rich interactive tapestry of sound and vision. There were various themes; city, seascape, pastoral, etc. These were intended to reflect and represent various genres in contemporary music.

It should be also noted at the point that the project was a combination of two skill resources and centres;
Group A, with experience in advanced audio for mobile telephones, and a Group B, with experience in audio applied to games and gameplay design and graphics generation. These two groups were often separated geographically and although they met on a regular basis this did directly affect the level of communication.
The knowledge and experience of both these teams, in regard to both music generation and the construction and function of computer games applications, was extremely high.
The project management was focused in group A, while the group B provided most of the technical resources and were responsible for the actual generation of the software, both audio and graphics.
The project was managed through a shared design document and very clear goals and milestones were set for each stage of the project.

The project management stated that the graphics were to be rendered in a full 'real-time' 3D environment. The audio rendering was based on technology that had been used successfully in two previous projects by Group B.

The project started well enough with some initial interactive 'sketches' being produced with the help of children's plastic toys and the film camera in a mobile phone. This was used to establish some of the criteria for combining sound to in-game events.

However, the project soon got bogged down in an endless stream of detail and documentation.

After six months of intense work, the conclusion had to be drawn that, while much useful work had been generated and a host of valuable lessons had been learnt, the project had failed miserably and that the deliverables had not been met.

The project had produced a nearly completed analysis and mapping of the theoretical music structure. This covering the relationships between tempo, harmony, instrumentation and phrasing, and while in itself was a comprehensive and original piece of work it was highly theoretical and needed the prototype to show if it was valid or not. This phase of the project had also demanded some 60% of the available resources.

There was also a test bed for the graphics. This used primitives such as cubes and triangles to represent objects and characters in the game. The test bed was intended to see what the result of triggering sound from game events was and assess the programming demands with a view to the end delivery system, a mobile phone. This test bed was bogged down by huge technical problems from the beginning and never really worked on the authoring PC let alone on a mobile simulator or actual phone, a minefield that affects most attempts at placing high-end, gaming content onto mobile phones[4].

Most of the difficulties with utilizing audio are to be found in the interpretive/artistic areas rather than the technical. The eye is quick to pick up on visual deficiencies (a missing shadow under a tree) whereas audio is normally more forgiving.
Audio is also capable of creating far more realistic scenes with far less resources[2]. This is a huge advantage when using audio to create prototypes. The problems Journey encountered had very much to do with the interaction between audio and the visual graphic rendering technology.

This was hardly an auspicious result for the effort put into the project, as we have stated…it was a substantial failure.

Why was this?

With the benefit of hindsight, it is possible to identify a number of pitfalls that from day one had doomed the project to its eventual fate:

1. By mixing a practical prototype with a research test bed we had put the cart alongside the horse. The prototype needed the answers that the test bed was designed to give before it could, itself, be defined.

2. There was a very clear intention to produce the prototype as a working application on an actual mobile phone. While this would have been of great benefit for both the technical status of the application and demonstration purposes, it loaded the project with two huge stumbling blocks at the very beginning. Applying high-end graphics and complex real-time audio are two of the most difficult programming challenges when creating content for mobile devices.

3. The insistence on using real time 3D for the graphics gave the project both an undue focus on the graphics issues, as well as raising the performance ambition and demands for the audio content. It became, for instance, very noticeable that the team discussions were all about the graphics quality, representation, level of animation quality etc. and not about the use of audio in the application.

4. Focus on the key issue of 'Sound as a principle driver of gamplay' became lost in the flood if technical issues and problems that needed to be resolved.

5. As the project progressed, a percentage of the team became convinced that the theoretical direction of the audio analysis and proposed application did not hold

water, and that a far simpler approach would yield better results. However, by that time such huge investments of time, effort and prestige had been lavished in the project that it was as good as impossible to reverse the direction the project was taking; Every minute from meetings, every thing you tell people outside the group, all these statements you do are investments in a certain direction. The more you invest the harder it gets to back out again. Therefore it is extremely important that you don't take on a project that is greater than you can handle and every thing you decide to produce is actually doable.

6. The time it took to create such a complex test-bed, meant that it was by its very nature inflexible and, ironically, only capable of performing according to the initial limitations of the project.

3.1. Project B. 'Beowulf'

After the decision to terminate 'Journey' was taken, there as still a very strong feeling in the B team that the research question at the heart of the project was still worth pursuing.
Audio's potential is extremely underused in most gaming genres compared to visual stimulus, especially when considering it's technical advantage over even the most up-to-date, real-time graphics:

'Since the introduction of commodity 3D-accelerated graphics cards, the evolution of graphics hardware and software has progressed at an increasing rate. Despite this, entertainment applications using real-time computer graphics are still far from producing images that approach the complexity found in reality. In contrast to graphics, the fidelity of audio found in current commodity audio cards is remarkably true to life.'[2]

A decision was taken to create a completely new project, 'Beowulf', to try and achieve what Journey had been partly designed to do: Test how sound can trigger and stimulate gameplay.
However, a fundamentally different approach was taken from the very beginning.
A practical deliverable was not specified at the start. The project's aim was to answer the research question, via an application that was intended to presented as a simple, intuitive experience for the user[3].

The prototype specification was honed to answer this question as clearly and as simply as possible. To this end, the followings design criteria were decided:

1. There would be no graphics to speak of, only a simple revealing map to help the player navigate.
2. There would be no intro text.
3. There would be no speaker or narrative speech.
4. There would be no music.
5. There would be no in-game text.
6. The game would need no instructions more than a quick explanation of the four buttons used to control movement.

It should be noted that the deliverable was not designed as a working game but, rather, as a test bed that focused all of its content onto gameplay and audio content. The intention behind the criteria was to remove any stimulus that relied on other skills than the interpretation of pure audio in an attempt to minimise contextual contamination that might affect the test subjects reaction to and interpretation off the project.

It was also important that the application should not be perceived as a game for the visually impaired. It was the relation of the sounds presented in the game to the users inherent ability to visualise events and locations that was of primary interest.

The technique of stripping down the project to an absolute minimum had a positive, two-fold effect:

1. On a technical level it was very easy to analyse the tool set needed to create the prototype.

2. It was also very easy to analyse the reactions of the test subjects; with such a focused use of sound it was no problem to determine exactly what they were reacting to, and they were able to be very concise in describing how the stimulus had affected them.

The first usable version of the prototype took a mere two weeks from the initial idea to its first working inception, limited and primitive though this undoubtedly was. However, even at this stage, it was sufficient to provide guidance and answers for the continued development of the prototype. It was even possible to begin testing some of the theoretical ideas almost immediately. The prototype proved to be very flexible and was easy to reprogram and change whenever this was deemed necessary. This ability to generate reaction early on was a huge benefit, it allowed for the test procedures to be designed with some practical insight of how the gaming experience worked. This, in turn, made for better, more-focused testing. It also allowed the team to realise certain effects early on and envisage new testing procedures that took this into consideration.
An example of this would be the early reaction to certain placeholder sounds. An environment had been designed with a carpet of old bones on the stone floor. A suitable sound was to be triggered when the game character walked across these. No existing sound of 'walking on bones' could be found so a SFX was created for the environment using eggshells and celery being snapped. It was a total surprise when early test subjects were able to identify these sounds, not with the intended inference 'bones on a stone floor' but with the actual sound source; celery and broken eggshell.
The subjects all demonstrated that we have a remarkable audio library in our heads that allows us to identify sounds with a higher degree of accuracy than was expected.
Note: It appears that as soon as even simple visual information is added to this, it overrides our ability to identify these sounds accurately. So, even though the Foley sounds were created in the same way as for an audio-visual experience, it became necessary to rethink the sound design process. It has been shown that there is a clear audio-visual correlation affecting our perception of sound and vision.

'... we showed that, audio stimuli do indeed attract a part of the viewers' attention away from the visuals and as a result, the observers find it more difficult to distinguish smoothness variations between audiovisual composites displayed at different rates, than between silent animations.'[1]

Although, we should point out that this work has concentrated on the effect of audio on our visual perception, and not the other way around.

This meant that the audio sourcing for Beowulf had to be more careful and rigorous than was initially intended. Had the audio library been an integral part of a more complex system this might have been a much harder area to adjust. Had the problem been part of a graphic representation (pre-rendered or real time 3D) the work to adjust it would have been tenfold.

One could say that the Beowulf prototype grew organically, allowing for testing and redesign as each feature was added, while always keeping the focus on the research task. The result of this was that the testing started at an earlier stage than was originally intended. The first tests were performed with a version that almost totally lacked the intended gameplay. It was basically a simple navigational experience, with all of its appeal in the various soundscapes experienced.

This in turn lead very quickly to a second prototype being constructed (Beo 2) that was able to test aspects of sound quality not addressed by the first, such as the quality of the sound file itself and the effect this has on the users perception of the environment. This had not been part of the original plan, but the testbed proved itself well able to adapt to this new role.
As the results from both prototypes are addressed in other papers, we will not deal with them here. Sufficient to say that both delivered a more than satisfactory payload for the time and energy demanded in their creation.

3.1.1. Simple prototyping is nothing new.

There are a number of examples where a deliberately simplistic or low-tech approach has been used to create innovative and even revolutionary products.
The Palm pilot was reinvented after two embarrassingly bad versions that were too ambitious and too 'clever'.
It's inventor, Jeff Hawkins, suddenly changed tack from the idea that the device would replace the PC and focused on replacing the paper based notes and devices people were carrying with them.
In order to best design this he made small wooden models that he pulled out in meetings and pretended to use:

'Here Hawkins posed a simple question: How small is small enough? His answer yielded the second principle: Small enough to fit in a shirt pocket. He paced the hallways at Palm headquarters, ruler in hand, measuring pocket sizes against small blocks of balsa wood. He designed screens and pasted down configurations of various applications.'[5]

Another good example of this is 'Spore', a new game from Will Wright created for distribution by Electronic Arts.
This product has relied on extensive and free prototyping to find the most suitable solutions to the myriad of design problems it faced. So interesting has this design process been for the industry that the prototype process has been presented at several trade shows, including GDC in San Jose and the Montreal Games Summit where it was a keynote given by Chaim Gingold and Chris Hecker, both of Maxis/Electronic Arts.

'Your prototype should be persuasive and fun. You should have to kick people out of your chair... If someone sits down and wants to play it, you win.' Chris Hecker.

http://www.gamasutra.com/phpbin/news_index.php?story=11628

3.2. Conclusions

If we are to avoid the mistake of painting ourselves into a corner and exhausting resources and time on project solutions that may well not answer our fundamental needs, we need to be very clear about what we are trying to do and extremely effective in how we achieve our goals.
Our practical tactics should allow for maximum flexibility in the application and we should be open to letting the developing prototype dictate our route of progress.
So the eventual methodology guidelines used can be stated as:

1. Define the specific focus for the project…what is the key question, then stick to this.

2. Analyse what type of prototype can best answer this key question in the most direct manner.

3. Design the simplest/easiest prototype that can do the job. Use existing tools wherever possible…if you have to create new technology, make it quick and dirty.

4. Don't concern yourself if the prototype is using a different technology than the eventual application, just as long as it answers the question effectively.

5. Start with a stripped down version of the prototype to test its ability to perform as required.

6. Keep the prototype flexible enough to be expanded and changed within the brief.

7. When all the fundamental research/development questions have been satisfactorily answered, then build the application.

8. Be prepared to throw stuff away if it doesn't work (especially the stuff you really like).

.

Warnings and observations:
One of the side effects of using really simple, if effective, prototyping techniques is the feeling within the team that it's not really a 'real' project. The speed at which initial achievements can be achieved can give the illusion of a less than satisfactory effort on the part of the design team. This is one reason why it is imperative to set up clear goals and benchmarks for the test application. If these are achieved with a minimum of effort then the team can only conclude they have performed well…
it can also be quite unnerving when the reaction to a project seems out of proportion to the perceived effort put in by the team to realize it.

References

[1] G. Mastoropoulou,K. Debattista, A. Chalmers[1]& Tom Troscianko[2] *The Influence of Sound Effects on the Perceived Smoothness of Rendered Animations*, [1]Department of Computer Science, University of Bristol.
[2]Department of Experimental Psychology, University of Bristol, UK.

[2] T. Roden , I. Parberry, *Designing a Narrative-Based Audio Only 3D Game Engine,* Department of Computer Science & Engineering, University of North Texas, Denton, Texas, USA.

[3] B. W. Boehm[1], T. E. Gray, and T. Seewaldt[2], *Prototyping vs. Specifying: a multi-project experiment*, Computer Science Department, University of California, Los Angeles, USA.
[1]also with TRW Defense Systems Group
[2]present affiliation: Universitaet Kaiserslautern.

[4] T. Fritsch, H. Ritter, J. Schiller.
User Case Study and Network Evolution in the Mobile Phone Sector. Freie Universität Berlin Takustr. 9 D-14195 Berlin, Germany,

[5]From an article by Pat Dillon, may 1998.
http://www.fastcompany.com/online/15/smallthing.html

[6]http://www.gamasutra.com/phpbin/news_index.php?story=11628

AuditoryPong – Playing PONG in the Dark

Wilko Heuten and
Niels Henze
OFFIS
Escherweg 2
26121 Oldenburg, Germany
{heuten, henze}@offis.de

Susanne Boll
University of Oldenburg,
Escherweg 2
26121 Oldenburg, Germany
susanne.boll@informatik.uni-oldenburg.de

Palle Klante
Pixelpark AG, Agentur
Friesenplatz 25
50672 Cologne, Germany
palle.klante@pixelpark.com

Abstract. Almost all computer games today are based on a visual presentation on the screen. While the visual realism advanced the main baseline of the interaction remains the same – when you switch off the display the fun is gone. In our research, we aim to provide non-visual access to computer games providing blind and sighted players with fun and pervasive interaction. As an application example we developed AuditoryPong, an interactive game that transfers the game PONG into a physical and acoustic space. The central elements of the game are transformed to moving and movable acoustic objects in a 3D acoustic environment. Based on the current acoustic game state the user moves the game paddle with body interaction or haptic devices and receives immediate acoustic feedback. Players do not need the visual display and sighted and blind people can play PONG together.

1. Introduction

Most of today's available games provide magnificent graphics to simulate fantastic worlds. The quality of the visual output constitutes a crucial factor for the game experience. Sound effects are becoming more realistic as well. However, similar to other non-visual modalities sound often serves mainly as an add-on for better entertainment without having a major impact on the game design and without transporting much primary information to the user.

There are already multiple examples were sound has a major impact not only on experiencing games, but also as a necessary part to play a successful game, e.g. first-person shooter. In previous research we developed, for example, a multimodal presentation for a mobile paper chase game. We used non-speech sound instead of a graphical map to provide information about the location of the next geo-referenced riddle [1]. Unfortunately, not all people are able to participate in the improvement of current game developments. Visually impaired and blind people do not have access to visual-oriented games, reducing their possibilities becoming entertained, having fun, getting socialized, and competing with others.

Apart from games developed for sighted players only, there are games specifically developed for user groups with special needs, e.g. auditory games for the blind. One of the first audio only games was "Sleuth - an audio experience" [2], which uses not only pure auditory cues but also speech output. Further examples of acoustic games are the sonification of the Towers of Hanoi [3] and the non-speech acoustic output for the games Mastermind and Tic-Tac-Toe developed by Targett et al. [4]. Auditory games for the blind are normally based on visual games with a very simple game idea. Their visual artefacts differ often only in their form or colour. This information can be quiet easily transferred into another modality. However, similar to visual oriented games, these games do not make use of the full potential of games in the sense of supporting the inclusion of different user groups. A game specifically developed for a blind user, for example, can typically be played only by the blind user, against a computer opponent or against other blind people in case of a multi-player game. Playing across different user groups with different visual capabilities is not possible.

In this paper, we propose the non-visual game "AuditoryPong" that can be played by sighted and blind users against each other. Based on the early published computer game PONG, we developed a new presentation metaphor and interaction techniques allowing blind people to interact within a virtual sound environment, perceiving game events, and controlling game artefacts, while sighted users still can use their well known input and output devices. The participants of the game can play at the same physical location but also over the Internet. The game design allows a sighted user to play the game without seeing a display but the player can also use a visual game interface. In AuditoryPong we bring the different user groups together as the game has been consistently translated from a visual interaction paradigm into an acoustic interaction paradigm. Through this, the game's idea, its elements, and interaction remain while enabling both user groups to hopefully win the game.

The remainder of this paper is structured as follows: The next section shortly recaps the original computer game PONG to the readers. The paper elaborates on the challenges and requirements of transferring the visual-oriented game into a pervasive acoustic and physical environment. Then we present the non-visual interaction design including the various input devices we developed or adapted for AuditoryPong. We describe the design and implementation as well as our experiences during several public demonstrations. The paper closes with a conclusion and an outline of our future work.

2. The Original PONG Game

One of the most prominent and early example of a visual computer game is PONG. Originally PONG game was invented by Nolan Bushnell and first published already in 1972 by Atari Inc. The game was first designed as an arcade game. Later it became very popular on home consoles like the Tele-Games console shown in Figure 1.

PONG is played on a two dimensional game field, which is divided into two halves. Figure 2 and Figure 3 illustrate the game layout and game elements. The user can choose between to layouts: horizontally – playing from the left to the right as shown in the figures – or vertically – playing from the top to the bottom. After starting the game, a ball represented by a small dot moves over the field from one side to the other.

Figure 1: PONG arcade console (left) and home console (right) [5].

Each player's goal is to make sure that the ball is not moving across the own baseline. To achieve this goal, the user can move a paddle along the baseline and try to hit the ball. If the player fails, the opponent scores, otherwise the ball collides and bounces back. The two other borders of the game field consist of walls. Each time the ball collides with a wall or paddle, the game plays a "beep" sound. The game is either played alone against the computer or against a human opponent. Both players are playing on the same machine at the same location.

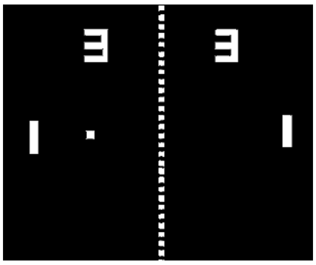


Figure 2: Original PONG game layout

3. Transferring PONG into a Non-Visual Game

In order to transfer PONG into a non-visual game, for example to make the PONG game accessible for blind people, the player needs to be able to perceive any of the visual information in a suitable modality. Furthermore the player must be able to control the main game elements. In this section, the game is analyzed in more detail to identify the information, which are exchanged between user and computer. The game field of PONG is a rectangular two dimensional area bordered by two walls on the sides and two baselines in multiplayer mode or one baseline and another wall in single player mode. The solid walls ensure that the ball rebounds if it hits the wall. Furthermore, the paddle cannot be moved through the walls. Figure 3 shows the basic game elements of PONG.

3.1. Presenting the Ball and its Location

One prerequisite to play the game is to perceive the location of the ball. For the design of AuditoryPong it is important to know when, how often, and how accurate the location of the ball needs to be perceived by the player.

The ball moves continuously within the game field. The primary user task is to manoeuvre the paddle to the position where the ball would hit the baseline. Keeping in mind that it takes some reaction time and time to move the paddle to a certain position on the baseline, we can conclude that the user must predict the ball's position in the near future. The player can achieve this by analyzing the movement of the ball. For this the ball's location must be presented continuously to the player as it is implemented in the original PONG through "moving" the ball's pixels. The accuracy and the time horizon of the prediction depends on many factors: the size of the game field in relation to ball and paddle, the velocity of the ball, and the speed the user is able to manoeuvre the paddle. If these parameters remain constant, we can conclude for the presentation of the ball's location, that the closer the ball approaches the player's baseline, the more accurate the information of the ball's location needs be presented. If the ball is near the opposite baseline, the location presentation can be less accurate.

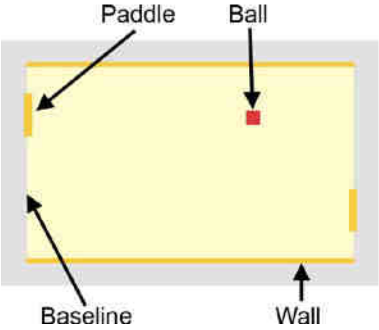


Figure 3: PONG's game elements

PONG consists of several game elements that need to be presented to the user. The user must be able to distinguish these elements and to identify the ball. The exact presentation of the ball strongly depends on the final interaction design. However from the above requirements for the location presentation, we can conclude that the ball is permanently displayed to the user. Therefore a presentation should be used, which does not annoy the user and allows an easy and accurate localization. It should not prevent the identification of other game elements. The representation must reflect the current position of the ball immediately.

3.2. Presenting the Paddle and the Field

Manoeuvring the paddle to the position on the baseline, where the player assumes that the ball will cross it requires some knowledge about the paddle. Primarily the player needs to know the current location of the paddle in relation to the ball or more precisely the location where the ball is assumed to hit the baseline. The player can then decide if and in which direction he or she has moved the paddle. In addition, the user should get an impression of the paddle's size in relation to width or height of the field. Although this relative information should be enough to play the game, it will help the player to perceive the absolute extent of the paddle to get a better idea of how far the paddle has to be moved.

An impression of the field extent helps the player to imagine the maximum distance that is covered by the paddle. The player needs to get a general image of the game layout, i.e. if the game is played horizontally or vertically.

3.3. Presenting Game Events and Moving the Paddle

It is necessary to present all events of the game and actions of the user to increase the fun of the game and to enable the users to enhance their playing abilities. Feedback on certain actions and game events – other than the movement of the ball – should be given to the user. The prominent pong-sound in the original game is one of such feedbacks. It occurs when the ball collides with a wall or paddle and informs the user about a change of the ball's movement direction. Another event is the collision of the paddle with the wall, indicating that the field border has been reached and enables the player to perceive the distance between both field borders. Feedback should also be given if the ball leaves the field, i.e. crossing one of the baselines and one player loses the game.

The only important physical activity the player needs to perform while gaming is to move the paddle along the baseline. Depending on the game layout, the paddle can be moved either from top to bottom, from left to right, or from rear to front. The design of the input interface should allow a continuous movement of the paddle to hold onto the idea of the game. In addition this makes it easier to keep track of the paddle's position.

3.4. Fairness

The social aspect is one of our cornerstones of our work: The game design allows different user groups to play against each other. Two players can choose different presentation and interaction techniques to play the game, meeting best their needs and preferences. Due to different interaction devices and the perception through different modalities it is not always possible to present the information in the same granularity level. This may result in different preconditions for the two players and to an advantage for one player. The game would become boring and frustrating. To overcome this inequity and to increase the fairness the game needs adjustable handicaps (following the idea of the golf system) to reduce the advantages of one player or to increase the advantages for the other player. Parameters for adjusting handicaps can be derived from above requirements, e.g., paddle size, paddle friction, ball speed, etc.

At first glance, PONG seems to be a very simple game. With its few game elements and rules the game is very easy to explain even if the player is not able to see. It turns out that when analysing it from the perspective of human-computer interaction, many parameters have impact on the game, its presentations and interactions. In this section, these parameters are examined, in order to provide useful information for the development of new interaction techniques.

4. Interaction Design

Interaction design defines the behaviour of interactive products. It deals with the communication between human and computer and determines the way information is exchanged. The previous section provides useful knowledge about what information needs to be exchanged for AuditoryPong. This section discusses potential interaction techniques and approaches how the information can be exchanged, such that blind and visually impaired people are able to participate in the game.

From the user's point of view, exchanging information can be categorized into output and input. Output stands for the machine's information presentation, while input means the user's ability to manipulate the state of the system. Designing input and output are parallelized tasks, because they are not independent from each other. For example, depending on the way information is presented, different interaction techniques can be applied, while others are not suitable.

In the following, we provide an overview about different interaction techniques, that we have developed in order to play AuditoryPong. We start with the output design, particularly the auditory output, as this leads to further requirements for the visual output and the input design.

4.1. Output Design

To enable users to play the game non-visually, AuditoryPong is played within a virtual sound room. In this room, virtual sound objects can be placed, representing the game elements and events identified in the previous section.

According to the requirements the ball must be presented continuously to the user. Thus, the ball is linked with a continuous sound. When the ball moves, the according sound object moves in the same direction with the same speed, representing the ball's location. Since a moving ball in the real world has no unique sound, we cannot use a typical "ball-movement-sound", which can easily be identified by the user right away. A single continuously played note would be very annoying. Thus, we use a pleasant melody that is continuously looped as auditory representative. This melody contains a wide band of frequency to ease spatial localization of the ball. The representation of the ball is the most complicated part of sound design for this game. It is the only artefact, which is heard constantly. At every moment the used tones should enable the user to localize the position of the sound in relation to the sound environment surrounding him. As we use the volume to mediate the distance between the user (the paddle) and the ball, the used sound shouldn't have any volume peaks or breaks.

Collisions of the ball with the walls and collisions of the paddles with the walls are presented by short sound effects to inform the player about the respective event. These sound effects are placed at the position where the collision took place to convey the position of the event. In order to help the user to distinguish between different events, each collision is associated with an individual sound. The sounds represent the attribute of the colliding materials. As the ball represents a flexible structure and the wall a very robust structure, the sound should represent, that the ball does not slide on the wall when colliding, but suddenly bounces back with the same angle it hits the wall. So a short hard tone should result in a realistic impression. In contrast to the ball, we can use realistic sounds to mediate collisions, determined by the materials that are colliding. To minimize the number of concurrently played sounds we decided to not represent all information in the auditory modality. Static elements like the walls should be perceived over the haptic modality. So the user has to learn the combination of information from the haptic and auditory modality. They must work together for a realistic game scenario.

In the original visual PONG game, the game is presented by a view on top of the field (bird's eye view). Due to humans' auditory perception, AuditoryPong requires a slightly different presentation of the game field. The auditory resolution is very low in vertical direction (it is hard to accurately locate sounds which are above or below our ears). Therefore, it is not recommended to use a bird's eye view for the auditory presentation, since then the ball and the paddle need to be located on vertical direction. For this reason AuditoryPong uses the perspective 3D view.

By placing a virtual listener within the sound room, the directions and distances of the sound sources regarding the listener's position and orientation can be perceived by the user. The virtual listener is either located centred and in front of the baseline, as shown in Figure 4 on the left, or located centred of the respective paddle, as shown in Figure 4 on the right. From the respective position the player listens to the ball and to any collisions. If the ball approaches to the player's baseline, the ball or more precisely the sound of the ball becomes louder. Hereby he is able to identify the distance of the ball from his position.

Furthermore the user is able to perceive the direction of the sound source. With the information of direction and distance, the user is able to appraise the location of the ball. By observing the ball's movement the player can determine the ball's speed and is then able control the paddle accordingly.

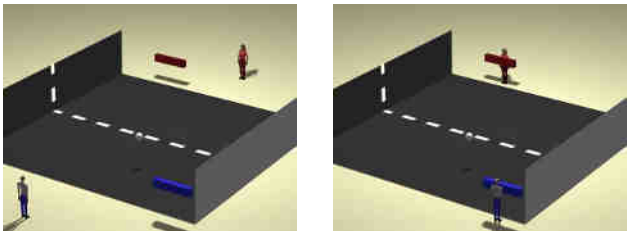


Figure 4: Virtual listener stands centred in front of the field (left) and listener is moved along with the paddle (right)

For the sound design it is important to know, that evaluations showed humans are able to distinguish between about five different positions horizontally (cf. [6]). However, these evaluations took place with static positioned sound sources. Playing PONG the user perceives the additional information through the "history" of the ball's position. So the user is able to anticipate the next position of the ball. This makes the sound design much easier.

The user has to "calibrate" to get an idea about the velocity of the ball and how fast she has to move the paddle in the right directions. This is a normal process since current soundcards use averaged head related transfer functions (HRTF) to present the virtual sound rooms. These algorithms can not reflect the player's individual hearing characteristics. When using headphones for the game the user has to adapt to the used (HRTF) that renders the sound environment. So the longer the user plays the game the better he learns this transfer function and will perform better in the game.

As shown in Figure 4, the user perceives the ball from back to front and left to right or vice versa. From this it follows that the user moves the paddle horizontally to hit the ball. Additionally to the auditory output, we equipped AuditoryPong with a visual display for sighted people. In case users like to use both presentation techniques, the visual output should comply with the auditory output, in the sense that the paddle should be moved in the same direction, in order to avoid discrepancies between different output and input techniques. The layout of the visual game field of AuditoryPong is therefore vertically, i.e. the baseline will be on the bottom of the screen.

4.2. Input Design

With the original PONG console the players use adjusting knobs to move the paddles. With these knobs the requirements described above are only fulfilled if the game is played with a visual output. The auditory output alone cannot adequately fulfil all requirements since, for example, the field's extend is not presented explicitly. Therefore, we developed or adapted different input devices to use them for AuditoryPong. We particularly address if and how the respective input device helps to fulfil the following requirements:

- Present the paddles size and location
- Convey an impression of the field's extend
- Move the paddle continuously

As these input devices are not used in everyday live in combination with the sound output, the user has to learn the usage of the input devices with the sound feedback.

We started with the traditional computer input devices mouse and keyboard that provide no information about the paddle's size and location. We connected the user's physical sensation with the paddle using a digitizer tablet and a physical slider. Finally, leaving typical computer peripherals as input devices to play the game, we allowed the players to steer the game with their own body movement using head tracking and body tracking technology. In the following we describe these input devices and their respective strengths and weaknesses in more detail.

4.2.1. Mouse and Keyboard

Mice as well as keyboards are the most common computer input devices. Both devices can be used to move AuditoryPong's paddles. If the mouse is moved to the left or the left arrow key is pressed, the paddle moves to the left. Moving the mouse to the right or pressing the right arrow accordingly results in a movement to the opposite direction.

Since mouse and keyboard are relative input devices and do not use absolute coordinates, the user cannot move the paddle to a specific position. In addition, the player receives no information about the paddle's position and size. The player only gets a rough idea of the field's extends due to the sound which is played if the paddle hits a wall. Furthermore, the keyboard does not allow continuous movement of the paddle. In general the mouse is an inadequate input device for blind and visually impaired people as it is very hard to use without the hands-eyes coordination. However, in connection with visual output, which provides a visual frame of reference, mouse and keyboard can be used for sighted user.

4.2.2. Joystick

Analogue and digital joysticks are common input devices for many computer games. For AuditoryPong we use an analogue joystick, shown in Figure 5, which provides precise information about how far the stick is moved in each direction. If the joystick is in its centred position the paddle is centred as well. If the stick is moved along the X axis the paddle accordingly moves in the same direction. The paddle hits the wall if the joystick is moved to one of its border positions.

Figure 5: Joystick provides continuous movement and absolute positioning of the paddle

The joystick is used as an absolute input device. The frame of reference is defined by the joystick's two extreme positions. Thus, the player can move the paddle to specific positions and receives information about the paddles position. However, the frame of references can not be perceived continuously and thus not very precisely and intuitively. In addition, no information about the paddle's size is conveyed. The player is assisted in getting an idea of the field's extents due to the joystick's physical resistance if the paddle hits a wall and the stick its extreme position. The joystick allows continuous movement of the paddle.

4.2.3. Digitizer Tablet and Slider

To enable the player to continuously feel the frame of reference we connected a digitizer tablet (Figure 6). The game's baseline is mapped on the X axis of the digitizer tablet. Thus, the game's left wall is at the tablet's left side, and the right wall is on the right side. The tablet itself is equipped with a physical border and tactile markers, which the player can feel with his or her hands. Using the tablet's stylus, the paddle can be moved by touching and moving the stylus on the tablet.

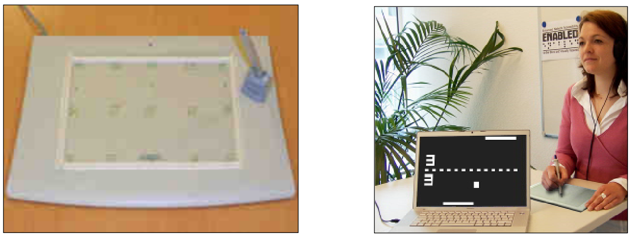


Figure 6 Digitizer tablet provides a frame of reference and absolute positioning

Markers on the tablet, which symbolise the game's walls, clearly define and convey a frame of reference which the user can perceive continuously. Thus, the player can move the paddle and receive the paddles position precisely. The player gets a clear idea of the field's extents and the stylus allows continuous movement of the paddle. However, the stylus does not provide any information about the paddle's size.

To provide the player with even more comprehensive feedback and enable him or her to identify the paddle's size we incorporated the concept of tangible user interfaces (TUI) [7]. In the context of TUIs the slider's handle is a physical artefact that is connected with digital presentation of the paddle. According to Ghazali and Dix [8] the behaviour and appearance of the paddle's physical artefact should correspond with the behaviour and appearance of the virtual paddle. Thus, we build the slider so that the physical handle has the same size as the virtual paddle. Our sliders are presented in Figure 7.

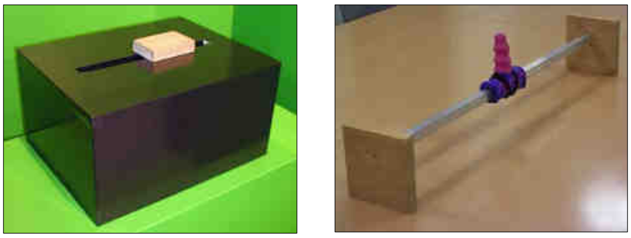


Figure 7: Slider mediate a frame of reference and the size of the paddle

4.2.4. Head Rotation

If the player uses headphones, the sound room rotates with the head, which gives an unrealistic impression of the perceived acoustic environment. In the real world a sound source in the front is equally loud on both ears. After turning the head to the right the sound can be heard stronger on the left side than on the right side. Thus, in the real world we are enabled to perceive the direction of sound sources very accurate. To achieve increased user-immersion we mimic the real world's acoustic behaviour. The orientation of the virtual listener inside the sound room is adjusted according to the orientation of the user's head. This orientation is tracked by a Flock of Birds Headtracker from Ascension Technology. The Headtracker's small sensor is attached to the headphones as shown in Figure 8. By adjusting the acoustic output according to the the head's orientation the user gets a more realistic sound impression. Thus, the user can locate the ball and other sounds more accurate.

To enforce the user to experience the improved acoustic presentation she can control the paddle by rotating the head. The paddle moves according to the user's line of sight. Moving the head to the left or to the right steers the paddle to the left or to the right on the vertical game board. By virtually "looking" at the ball the ball's sound is equally loud on both ears and the paddle is in the right position to hit the ball. Thus, it is easy to hit the ball by simply facing in the right direction.

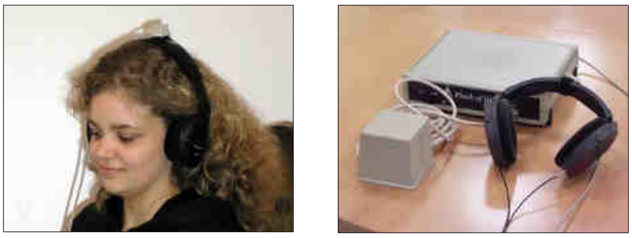


Figure 8 Moving the paddle by rotating the head

4.2.5. Body Movement

To relieve the player from sitting in front of the desk we turn PONG into a pervasive game and release the game from the computer chair. We connect the virtual game board with a real room. The physical walls of the real room represent the two walls of the game. The user body represents the paddle in physical game board. By physically moving his or her body between the two walls the user moves the virtual paddle. In addition, the position of the virtual listener within the sound room is moved according to the user's real position.

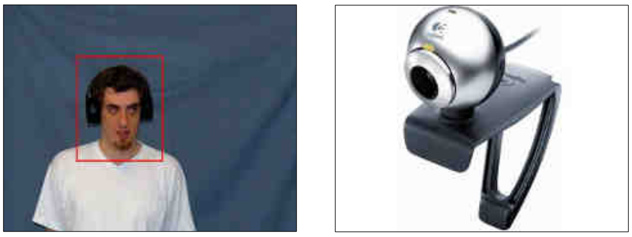


Figure 9 Pervasive interaction with face tracking

To redeem the user from any cables we use wireless headphones and determine the user's position using face-tracking technology as shown in Figure 9. The user moves in front of a Web cam that captures the scene and detects the player's face. While the user moves to the left or to the right, the camera tracks the face in real time. As there is no visual display the user relies only on the physical movement in the room and the acoustic feedback.

5. Game Demonstrator

We have developed an AuditoryPong demonstrator and tested the interaction techniques at several public events. In the following sections we describe the design and implementation of AuditoryPong and summarize our demonstration experiences.

5.1. Design and Implementation

AuditoryPong has been designed as a multiplayer game which can be played over the Internet. We implemented pong using a client/server architecture. The server processes the game logic, for example calculating the position of the ball. The thin client is responsible for the user interface, i.e. presenting the game elements and game events, and processing the user inputs. Client and server can be run on the same machine to also allow local gaming with two players. The client architecture is illustrated in Figure 10.

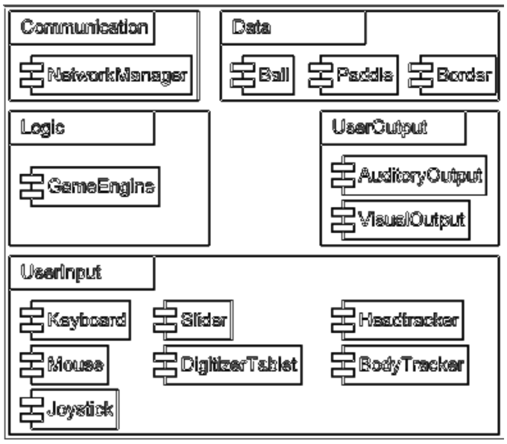


Figure 10: AuditoryPong's architecture

The architecture consists of five main components: Communication, Logic, Data, UserInput, and UserOutput. The communication component encapsulates the network communication with the server (sending and receiving game events). The data component represents the models of the game elements, stores and manages their properties, and provides functions to manipulate and access their states. The user interface is realized by the UserInput and UserOutput components. Status changes received from the game logic are processed and outputs for the currently active output device(s) are generated by the UserOutput classes. The UserInput classes receive any events from the input devices and calculate new states of the game elements. The logic component initializes and manages all other client components, provides interfaces and forwards messages between the data component, the network, and the user interface.

Besides the more essential components, our demonstrator provides accessible dialogues to configure the game. Configurations can be stored in profiles that include the input and output device, the network setup, and fairness parameters like paddle size, paddle friction, and speed of the ball. Furthermore the player can choose between different sound and graphics themes.

AuditoryPong is implemented in C++. For the non-speech 3D sound we used a sound framework for auditory displays that we have developed earlier [9]. The sound framework can be used with various sound libraries and APIs like EAX, DirectSound, A3D, FMOD and OpenAL. For the face detection and tracking we used algorithms provided by the Open Source Computer Vision Library (OpenCV) [10]. Other devices are connected via standard C++ libraries.

5.2. Demonstration Experiences

AuditoryPong has been demonstrated on several public events. The most prominent event was the Ideas Park organized by ThyssenKrupp [11] where about 3000 people dabbled in the game. Two players – mainly between 8 and 16 years old – competed at a time. One player played with a visual screen and the joystick as input device, the opponent played without visual feedback within a box using the slider to control the paddle and speakers for the sound output. The player with the visual feedback was handicapped by higher ball speed, paddle friction and a smaller paddle. After a really short familiarization time (about one minute) both players were mostly able to play the game against each other over a longer period of time. The interfaces came out as very easy to learn and extremely robust. Most players quoted to have fun with the game and were surprised that blind players were able to keep up with sighted users. The fairness functions helped to balance the different handicaps of both players.

There have been several regional events with between 10 and 300 visitors, to whom we have demonstrated AuditoryPong along with its different interfaces: Long Night of Computer Science [12], Girls' Day [13] (both supported by the German Federal Ministry of Education and Research), and dorkbot [14]. The experiences at those events provided us with further feedback about the implemented interaction techniques. Most of the players' statements can be compared with those from the Ideas Parks. The computer vision based body movement input technique turned out as not as robust as the slider or digitizer tablet due to unstable lightning conditions. However, playing a computer game without a monitor or the need to use an extra device to control the paddle provided the players with new interaction experiences.

Within the project ENABLED [15] AuditoryPong is used as a training application for blind and visually impaired people. In ENABLED applications are developed, which mediate spatial information non-visually through 3D non-speech sound. AuditoryPong serves as practice for those applications and to get familiar with the new interface through play. Interviews showed that blind and visually impaired people were able to play the game easily and that they learned the spatial relations between the different game elements without difficulties. They stated that AuditoryPong is a useful introduction to auditory displays and trains in localising synthesised sound objects.

6. Conclusion and Future Work

In this paper, we presented AuditoryPong, an advancement of the original computer game PONG, which can be played with or without a visual screen by both blind and sighted players together. We analyzed the different game elements, game events, and control requirements of the original PONG, in order to determine the necessary information and input mechanisms that need to be provided to the user for playing. Transforming the visual game into a non-visual playground, we designed a new user interface and new interaction methods: Core of the interface is the spatial non-speech audio component which presents the game elements and their spatial relations. We developed and integrated different input techniques, which enable the user to control the game. Some of these input techniques make use of haptic devices, which also present information about the game state and thus support the auditory output. Computer vision based face detection allows controlling the game by moving the own body, putting the player in a fully immersive environment. The interaction methods were evaluated through several public events.

With AuditoryPong we showed that visually oriented applications, such as games, can be made accessible for blind and visually impaired people through non-visual interfaces. From AuditoryPong we aim to learn about multimodal interaction techniques and transfer our experiences to other application areas beside games. Experimenting with different modalities and devices we develop non-visual interaction paradigms that come naturally in different stationary or mobile situation. In the future we will further improve the existing interaction techniques and investigate new non-visual interaction methods. In particular the combination of auditory displays and haptic and gesture interfaces looks promising for perceiving and exploring spatial information.

Acknowledgments

This paper is supported by the European Community's Sixth Framework programme FP6-2003-IST-2-004778. We would like to thank Daniel Nüss for his support during the development of AuditoryPong.

References

[1] Klante, P., Krösche, J., Boll, S., Evaluating a mobile location-based multimodal game for first year students. In Proceedings of the Multimedia on Mobile Devices, pp 207-218 (2005).

[2] Drewes, T.M., Mynatt, E.D., and Gandy, M., Sleuth: An Audio Experience, In Proceedings of the 2000 International Conference on Auditory Display.

[3] Winberg, F. and Hellström, SO., Investigating Auditory Direct Manipulation: Sonifying the Towers of Hanoi. In Proceedings of the International Conference on Disability, Virtual Reality and Associated Technologies, pp 75-81 (2000)

[4] Targett, S. and Fernström, M., Audio Games: Fun for All? All for Fun? In Proceedings of the International Conference on Auditory Display, pp 216-219 (2003)

[5] Wikipedia contributors, PONG, In Wikipedia The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=PONG&oldid=112959436 [last accessed August 20, 2007].

[6] Donker, H. Klante, P., and Gorny P., The design of auditory user interfaces for blind users. In Proceedings NordiCHI, pp 149-156 (2002)

[7] Ishii, H. and B. Ullmer, Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. Proceedings of the ACM Conference on Human Factors in Computing Systems (1997).

[8] Ghazali, M. and A Dix, Knowledge of Today for the Design of Tomorrow. Proceedings of the 2nd International Design and Engagability Conference (2005)

[9] Heuten, W., Wichmann, D., and Boll, S., Interactive 3D Sonification for the Exploration of City Maps. In Proceedings of the Fourth Nordic Conference of Human-Computer Interaction, 155-164 (2006)

[10] Intel, Open Source Computer Vision Library Website. http://www.intel.com/technology/computing/opencv/ [last accessed on August 20, 2007].

[11] Ideas Park (2006) Hanover, Germany: http://www.zukunft-technik-entdecken.de/?lang=en [last accessed on Feb 27, 2007].

[12] Lange Nacht der Informatik 2006 (German). http://www.informatiknacht-ol.de/ [last accessed on August 20, 2007].

[13] Girls'Day 2006 (German). http://www.bmbf.de/de/5773.php [last accessed on August 20, 2007].

[14] Dorkbot – people doing strange things with electricity. http://dorkbot.org/ [last accessed on August 20, 2007].

[15] ENABLED – Enhanced Network Access for the Blind and Visually Impaired. http://www.enabledweb.org/ [last access August 20, 2007].

Strategies for Narrative and Adaptive Game Scoring

Axel Berndt, Knut Hartmann
Department of Simulation and Graphics
Otto-von-Guericke University of Magdeburg
P.O. Box 4120, D-39016 Magdeburg, Germany
{aberndt, hartmann}@isg.cs.uni-magdeburg.de

Abstract. In this paper we briefly introduce the narrative concepts of interactive media music. Therefore, an organic correlation between music and the interactive scenario is essential. We elaborate and discuss the two basic strategies to enable an adaptive musical accompaniment. The real-time arrangement of precomposed music incorporates innovative musical structures, but often lacks flexibility, whereas an automatically generated musical soundtrack opens up great potential for the application of automatic composition systems, although their results are often said to be of minor artistic quality. Thus, we will furthermore introduce a third so far sparsely regarded approach—the hybrid combination of artistically qualitative precomposed music and flexible generative techniques. These open up new and prospective perspectives for the field of adaptive and interactive music.
This paper structures the innovative upcoming field of non-linear music with special regard to game scoring. A contrastive discussion will compare the potentials and limitations of the compositional and generative approaches and the newly introduced hybrid approach. For a stronger musical coherency we elaborate structure protective adaption principles.

1 Introduction

Film scoring is said to be the role model for computer and video games music. But while films are static and linear, computer and video games are generally interactive and non-linear. From this ontological difference, two major qualitative drawbacks emerge:

1. Music, as a linear art form as well, is hardly able to organically follow an interactive scene. To the disadvantage of coherency of musical flow and content, abrupt cuts are largely used to resynchronize music and interactive scene.
2. The musical linearity is one of the major obstacles that prevent a closer linkage to the interactive scene in respect of content. There is a considerable lack of media theoretical awareness on the narrative functions music is able to perform. Thus, game developers rarely tap the full potential of music as an instrument of expression, narration and comment.

The adjacent section 2 will give an introduction to the historically evolved functions of media music and lead to a wider range of narrative responsibilities in interactive media (e.g., computer/video games). Therefore, an adaptive and organic musical accompaniment is essential. Sections 3 and 4 will describe the two conventional basic strategies to tackle this problem, i.e., precomposed and algorithmically generated music. A further minor explored approach will be introduced and discussed in section 5. Section 6 concludes this paper.

2 Narrative Music

Combining music with different media is not new. This so called media music can be found in the earliest forms of appearance of music. Its traditional connection to movement, dance and text is well known and bred famous classical art forms like ballet, theater and opera. To an increasing degree, the composers see their role not only in decorating and duplicating non-musical content, but in mediating its deeper meaning and reflecting on it with musical means.

A typical example can be found in Felix Mendelssohn Bartholdy's oratorio Elijah. To mediate the primitivity of the prayer to the false god Baal, he used a simplistic harmonical language and counterpoint in contrast to Elijah's musically sophisticated prayer to his god. Another famous example is the final chord of Johann Sebastian Bach's St. Matthew Passion with an upwardly resolving dissonance—untypical in baroque music and thus extraordinary flashy to the coeval ear—suggesting resurrection and Easter [7, 12].

In theater, and its descendant the film, music plays a more subconscious role. In both, visual layer, sound and dialog are dominating and consciously perceived while music has to be subordinate to them in terms of its structure and development. However, this subconscious perception does not imply a minor importance within the dramaturgy. The more subconscious music acts, the more it can condition the audience in a desired manner and stimulate their perceptual direction.

Its suggestive power can sensitize the audience to emotional and analogous contents of the visuals [26]. Furthermore, its combination with images and words spans an association space of amazing unambiguousness [21]. Schneider calls this a *semantization process* [26]. Music becomes meaningful as a narrative medium that does not just express emotions and mood, but becomes a means for the expression of associations and non-verbal comments.

It can even challenge the meaning of the visuals as Eisenstein, Pudowkin, and Alexandrow describe in their famous Manifesto [11]. They distinguish between audio-visual *parallelism* and *counterpoint* in terms of the relation between music and image. Parallelism comprises those musics that follow the visual content. They express contents that are also visible in the scene (e.g, its mood, themes related to visible characters, micky mousing etc.), whereas the audio-visual counterpoint describes music that controverts the scene. It enriches or even changes the superficial meaning of the images: A happy scene accompanied by scary dissonant music seems to hide a serious invisible danger. In this way, the music acts as the speaking tube of the author. Several musicologists, like Pauli [25] and Thiel [27] supplemented this classification by a third class, comprising those musics, which add new non-visible content but do not controvert the scene, *affirmative picture interpretation and illustration*.

A more detailed contemporary classification of musical narrative functions is given by Johnny Wingstedt [29]. He distinguishes the following six classes of functions.

Emotive Class:
– emotionalize content and acting;

Informative Class:
– communication of meaning;
– communication of values;
– establishing recognition;

Depictive Class:
– describing settings;
– describing physical activity;

Guiding Class:
– attention guidance;
– masking (out) of unwanted or weak elements (e.g., projector noise, bad acting);

Temporal Class:
– provide continuity;
– define structure and form;

Rhetorical Class:
– comment, make a statement, judge.

In connection with the temporal functions, Wingstedt states: "In interactive non-linear media, such as computer games, music's ability to provide continuity is an important quality with strong potential."[29] Music helps to integrate the disrupted episodicness of the cut scenes to a whole. It connects visually separate locations of a virtual world into a bigger continuous and more believable whole.

Current computer and video games rarely apply at least a few of these functions and pauperize at a banal entertaining, at best a parallelism-based score concept. Thus, the pool of musical narrative means is extremely underdeveloped.

However, since background music in games is perceived and understood in the same way as film music [19], it can by all means perform all its narrative functions and even more. It can expand the bandwidth of information flow by disencumbering and complementing the visual channel, and support the user/player in controlling and managing complex structures (e.g., in strategy games) [19]. Its associative power can furthermore be used to influence his playing behavior and decision process. Afterwards, music can convey a response in terms of a moral comment etc.

In this role, it does not just refer to an actor on screen but to the player himself and reaches him on a more personalized and intimate level than ever. Music can take a stand on interaction as an *audio-interactive counterpoint*. This, amongst other educationally interesting potentials, lies idle today. In [2], we give a detailed discussion of this narrative concept which is only possible in interactive media.

For the success of such more sophisticated and ambitious narrative concepts the *dialectic unity of form and content*—well known to every artist—is essential. It is not enough to play a suitable piece of music and stop it when the scene ends. Like film music, interactive media music has to organically correlate with the development in the scene. Asynchronously cut music—as widely used even in contemporary up-to-date top-selling games—destroys inner-musical coherency and flow. The indifferent relation between music and scene is conspicuous; its functions seem enforced and contrived, not organic and plausible. A new type of music (not necessarily style!) is needed that is able to organically follow an interactive scene, a non-linear music that can automatically and flexibly be adapted. *Adaptive music* is needed. The following sections will discuss the three strategies to tackle this problem.

3 Precomposed Music

The musically coherent scoring of interactive scenes is a problem that the computer and video games industry is aware since approximately two decades. Hence, some of the most important solutions arose from this context. The scientific interest on this topic is much younger and began to sprout in recent years.

The so far most impressive and qualitatively best solutions incorporate not just a technical but also a compositional treatment of the problem. Precomposed music does not necessarily have to be static and linear. It can feature special structural attributes that abet a formally more open and unsealed musical layout. The performer—in this case the computer or more precisely the music engine—can deal with precomposed material and rearrange it in realtime, even during the performance.

It is at the responsibility of the composer to invest these structural attributes within his musical conception and enable such an automatic arrangement. This can feature three different characteristics: (1) a manipulation of performance-related means of expression, (2) the arrangement of the timely sequential order of musical sections, and (3) the arrangement of simultaneously playing tracks/parts.

3.1 Expressive Performance Manipulation

Without changing the composition in any way, its character of expression can already be varied by a change of its performance style. A smooth cantabile melody may appear romantic; the same melody in a faster tempo and a staccato articulation appears snappy. The potential of such performance means is used extremely rare.

One of the earliest examples of such music adaptions can be found in the classic arcade game *Space Invaders* (Toshihiro Nishikado, 1978). A repetitive stepwise descending four-tone sequence illustrates the approaching of hostile UFOs. The closer they come, the more increase speed and difficulty of the game. Likewise, the tempo of the four-tone sequence accelerates and mediates an increasing suspenseful precipitance.

The MIDI-based system *REMUPP*, presented by Wingstedt et al. [30], allows tempo changes as well. Additionally, a couple of further performance parameters can be edited, namely: articulation, reverberation, pitch transposition to different octave levels, and a timbre change by selecting different predefined instrumentation sets. These changes are applied while the playback proceeds uninterrupted.

Thus, the work with performance aspects includes the use of tempo, articulation, timbre, acoustical and technical effects, as well as, the change of dynamics (loudness) and time signature based emphasis schemes. Applying these aspects to a given piece of music premises a suitable and in detail editable representation format; wave based audio formats (e.g., Wave, MP3, and Ogg Vorbis) are inapplicable. The MIDI format is a better choice, although its direct use in games scoring decreased during the past decade because of the deficient and varying sound synthesis quality of home computer soundcards. However, this problem can be overcome by sample libraries (e.g., the famous *Vienna Symphonic Library* [28]), since they are already widely used in film scoring and the studio production of games music.

3.2 Sequential Arrangement

In the tradition of classical musical dice games (well-known is that of Mozart [24]) most adaptive game musics are composed as a rearrangeable sequence of musical sections. Examples can be found in *No One Lives Forever 2* (Monolith Productions, 2002, scored by Nathan Grigg) and *Unreal 2* (Legend Entertainment, 2003, scored by Jeremy Soule).

If interactive events necessitate a music change (e.g., the player enters a different location or is attacked), the currently playing music section will continue uninterrupted to its end where the first section of the new music joins seamlessly. Of course, the composer has to take care of the melodic, harmonic, and metric connectivity of all possible consecutive sections. To furthermore smooth this change, the developers of *Gothic 3* (Piranha Bytes, 2006, scored by Kai Rosenkranz) retouch it with different percussive effects.

More elegant is the approach of the *iMuse* engine created by Land and McConnell [20]: Jump marks can be defined at every position in the piece. They can be activated during the playback and trigger a continuation on a different position in the same or another piece. Furthermore, every mark can reference to a specialized transitional cue that leads smoothly from the particular musical context to the other piece. Applied in MIDI, these marks can also trigger changes of instrumentation, velocity, tempo, transposition, and stereo panning. The *iMuse* engine is the most sophisticated representative of the sequential arrangement that approaches and demonstrates its manifold abilities in lots of famous Lucas'Arts computer games (e.g., Monkey Island 2–4, X-Wing, and Tie Fighter).

3.3 Arrangement of Simultaneous Tracks

To get a smooth transition between two pieces of music one might be tempted to cross-fade them. Unfortunately this does not work in general. During a cross-fade both musics are hearable. The metrical and harmonic overlays of independent unsynchronized pieces generally sound quite confusing and unintentionally dissonant to the listener.

Nonetheless, cross-fading is an expedient tool for music transitions, provided that the music meets some special demands: All tracks have to be played back synchronously and are composed bearing in mind that they have to harmonize with each other. Now a cross-fade appears more appropriate although the second music, to which the cross-fade bridges, will usually not start at the beginning. But since it emanates coherently from the musical flow, it will still sound believable.

The aforementioned *REMUPP* system uses multiple tracks to change the harmonic and rhythmic complexity of a piece of music by adding or muting several parts and percussion tracks. In [3] we furthermore incorporated a distance attenuation model into the fading mechanism to automatize location-based music transitions. Since the synchronous tracks have to be musically compatible, cross-fade transitions tend to be relatively local; the change cannot be as drastic as it might be wished sometimes. Therefore, we applied the sequential arrangement strategy for further music changes which can be triggered, amongst others, by story related events.

These examples show that the creative application and combination of all the arrangement techniques for precomposed music opens up a wide range of possibilities. Since all music is precomposed, a high compositional and artistic quality can be ensured. But this is expensively paid with higher compositional effort, complexity, and a limited flexibility. Often, the music change is a long time coming because the playback has to finish a section or meet a jump mark first. These drawbacks may be less distinctive when music is generated in realtime according to interactive events or the state of the virtual world.

4 Generated Music

Generative concepts, in contrast to precomposed ones, also incur the music composing. Since most composition systems generate static concluded pieces, they are of secondary interest for the scoring of interactive scenarios. Thus, this elaboration will concentrate on those systems that stand exemplary for realtime music generation concepts that include interactivity in some way. Generally, music changes are triggered by modifications of parameters or templates.

A typical example is Biles' *GenJam* [4] that plays in duet with a human performer and improvises via a genetic algorithm approach on his musical input. Another example is Chapel's fractal-based *Active Musical Instrument* [8]. The user defines iterated functions and modifies their parameters during playback. Although aesthetically very interesting, complex, and consistent due to the nearness to serial music, this system as well as *GenJam* incorporate an actively co-performing user to continuously affect and support the system with new inspired musical ideas and material. A subconsciously listening user is insufficient.

With regard to the autonomous generation of an adaptive continuous musical accompaniment, Casella's *MAgentA* system [6] seems promising. It incorporates multiple composition algorithms and maps them to specific emotions (e.g., one for happy music, another one for sad music). If the state of the virtual scene changes, an appropriate algorithm is selected and continues the generation of background music. This concept bears great potential and leads to a new problem. For a smooth and proper transition between the musics of two algorithms (i.e., two different compositional styles!), methods for a progressive style modulation [10] are needed.

This problem as well as the description and modelling of compositional styles are widely unsolved today. Composition systems only apply contrapuntal characteristics but miss the manifold variety of meta-structural features that express the style establishing way a composer handles them to achieve a specific expression. Furthermore, orthography and grammar are not enough to create a senseful expressive natural language text. The creative process of music composition is more than a contrapuntal optimization problem; it includes the definition of a piece-specific contrapuntal material and the structure establishing application and breaking of this material in an innovative new-creating way. With the words of Harold Cohen: "Creativity is not a random walk in a space of interesting possibilities, it is directed."[9]

The human artist, whether composer or performer, is still indispensable! His inclusion as an inherent part of the concept is what the well-deserved success of *GenJam* is based on [5]. And this is the promising potential of *MAgentA*—the different composition algorithms can be procedural descriptions for very specific well-composed pieces of music that include, for instance, variation and adaption techniques. E.g., Hörnel's neural-network-based melody variation techniques can be applied for this [17, 18, 16]. Thinking about such an equal combination of human and machine leads to the new, so far sparsely regarded, hybrid approach.

5 Hybrid

Both previous concepts—precomposed music that is automatically arranged, and generated music—have their benefits and drawbacks. There is high artistic quality but limited adaptiveness on the one hand, and high flexibility but weak meta-structural coherency on the other hand. The idea arises to combine both concepts and overcome the drawbacks of one by the advantages of the other one: high quality precomposed music that is not just arranged but can be flexibly adapted. Up to now, few research has been done in this direction. Hence, there is only a small number of approaches on music adaption.

5.1 Music Adaption

In 1968 Mathews et al. had the idea to represent the pitch and rhythm of melodies as functions of time and demonstrated a running arithmetic interpolation between two folk tunes [23]. They started with *The British Grenadiers*, turned it into *When Johnny Comes Marching Home*, and ended back with the Grenadiers. Although the authors describe their results as ”nauseating” and ”jarring”, they are not uninteresting, especially for serial music. Of course, since this interpolation does not conveniently consider, e.g., tonality or motific patterns, it is hardly suited to bridge classical and popular tonal melodies or polyphonic structures.

The idea to achieve music transitions by some kind of interpolation was renewed by the *MusicBlox* concept of Gartland-Jones [13, 14] and the Music Morphing strategies of Wooller and Brown [31]. Given the start and end patterns of the transition that belongs to the corresponding precomposed musics: during the transition the start pattern is repeatedly played and with every iteration gradually reshaped towards the end pattern. Gartland-Jones realizes this with an evolutionary approach. Wooller and Brown describe three further approaches: the *interpolation* which is conceptually much alike that of Mathews, the *mixture* of start and end pattern fragments, and the *markov morphing* which might be better understood as an alternating use of two different transition matrices that describe the voice leading characteristics of the two patterns and generate a new one from them.

A further music adaption concept is presented by Livingstone et al. [22]. They suggest to change the character of expression of the one running piece of music by editing specific musical parameters, e.g., tempo, articulation, mode, harmonic and rhythmic complexity. In contrast to the aforementioned *REMUPP* system, they realize these changes with a rule-based approach: nontrivial, considering, e.g., the complex harmonic, melodic/motific, rhythmic, articulation—generally speaking meta-structural—interrelationships that have to be kept to ensure musical coherency. Again, style modulation is a challenge to be addressed.

5.2 Structure Protective Adaption

So far, really coherent music adaption is not achieved. The few recent projects concerned with this challenging and even musicologically interesting topic open up a number of promising starting points. Nonetheless it must be clarified that adaption techniques will violate the composition and cannot succeed if they do not consider the music's immanent structural features. For an adequate adaption of precomposed music that can even pass before its composer, its meta-structure has to be incorporated to give inconsistencies no chance and save musical flow.

For a stronger coherency, we suggest the application of music adaptions at the running musical material. The repetition of sections during the transition conflicts with the postulate of organic flow and has to be refused. Instead, the piece of music has to continuously play on.

Adaptions have to be structure protective and cause as little change as necessary to realize different expression characteristics or to allow a seamless transition and connection to the destination music.

Such changes should not be applied from any position in the piece but at structural boundary points. A halfway played theme or motif should not be changed or at least very carefully and unobtrusive, but its next appearance can. The starting points for the variation of a melodic phrase are its beginning, its cumulative/gestural destination points (e.g., melodic peaks), and its end (which is usually the beginning of a new phrase)—likewise, harmonic progressions. Furthermore, metrical changes (e.g., tempo changes) should only be applied at boundary points or lead to them target-oriented (e.g., ritardando and accelerando).

Hence, these macro-structural boundaries give information about appropriate starting points for the application of adaptions. Furthermore, the structure characteristics of the particular meta element give clues and limitations for protective variations. E.g., a melodic motif may be defined by the diatonic intervals and interval directions between its single notes. A rhythmic variation, a transposition, or even small interval changes in the semitone space would not destroy its motific structure but are mighty instruments to adapt the motif to a different harmonic base.

Generally, the musical morphology differentiates variations by two aspects [1]:

Subject of Variation: The *direct variation* is applied to the theme/motif, whereas the *indirect variation* retains the theme/motif unchanged but varies its accompaniment.

Type of Variation: The *strict variation* saves the harmonic and architectural characteristics of the theme/motif. Its shape and gestalt quality stay unchanged. On the other hand, the *free variation* changes not just melodic and rhythmic aspects, but also harmonic and formal. Each one of such variation can afford new gestalt and quality.

Moreover, the example showed that meta elements provide lots of clues that can help to strengthen the target-orientation of variations. Strict variations are best suited for structure protective adaptions. To furthermore appear subtly unobtrusive, the variations should be indirect, whereas direct variations are suited to connect melodic lines and ensure their melodicity.

Therefore, a sufficiently extensive knowledge of the music's meta-structure is necessary. It can be provided by additional meta files that contain the results of, e.g., an automatic analysis. Since ambiguity is a common problem, we suggest a semi-automatic analysis, as described in [15].

6 Conclusion

In this elaboration we presented a systematization of the so far sparsely developed field of adaptive music. We differentiated three basic approaches that weight the role of the human composer and the music arranging or generating machine very different. The noticeably serious differences between precomposed and automatically generated music in terms of artistic quality and adaptability/flexibility make the hybrid approaches—although in their infancy—seem very promising. They leave the art creating process at the real artist, i.e., the human composer, and employ the machine beyond the humanly possible—the immediate adaption in response to interactive events in a virtual environment.

We have discussed the problems of the hybrid music adaption approaches with regard to inner-musical coherency and flow. The music's immanent meta-structure has to be considered to a much higher degree! Otherwise, adaptions and variations will perturb it. Therefore, we presented a new conceptional approach towards a structure protective music adaption technique.

Finally, a flexible musical accompaniment—whether precomposed, generated, or both in combination—is essential for an organic linkage with the manifold possible details and narrative aims of the interactive scenario and smoothes the way for interactive media music to become the weighty narrative instrument that its ancestor, i.e., linear media music, already is. Like film music, which had to develop appropriate structures for the connection to moving images half a century ago, interactive media music has to find its way, too.

References

[1] G. Altmann. * PI>?1@I3<7 $CHA 7B@<H78 #>B &1B62P3< A >M 7>ID>7@B PB6 B1@I7B. Schott, Mainz, Germany, 8th revised edition, jan. 2001.

[2] A. Berndt and K. Hartmann. Audio-Interactive Counterpoint. In . CPB65 * PI>3 1B6 M47 * CQ>B; 'A 1; 7, University of London, England, sept. 2007. Institute of Musical Research.

[3] A. Berndt, K. Hartmann, N. Röber, and M. Masuch. Composition and Arrangement Techniques for Music in Interactive Immersive Environments. In P6>C * CIM@ OVVK4 ! CB9G CB . CPB6 >B %1A 7I, pages 53–59, Piteå, Sweden, oct. 2006. Interactive Institute, Sonic Studio Piteå.

[4] J. A. Biles. GenJam: Evolutionary Composition Gets a Gig. In , HC3776>B; I C9 M47 OVVO ! CB= 97H7B37 9CH 'B9CHA 1MCB / 73<BC@C; T ! PHH3P@A , Rochester New York, USA, sept. 2002.

[5] J. A. Biles. GenJam in Transition: From Genetic Jammer to Generative Jammer. In , HC3776>B; I C9 M47 : M4 'BM7H B1MCB1@ CB97H7B37 CB %7B7H I MQ7 HM7, Milan, Italy, dec. 2002.

[6] P. Casella. Music, Agents and Emotions. Licentiate thesis, Engenharia Informática e de Computadores, Instituto Superior Técnico, Universidade Técnica, Lisboa, Portugal, july 2002.

[7] E. Chafe. J.S. Bach's St. Matthew Passion: Aspects of Planning, Structure, and Chronology. (CPH B1@C9 M47 A 7H 31B * PI>3C@C; >31@ . C3>7M I, 35(1):49–114, spring 1982.

[8] R. H. Chapel. - 71@M A 7 @CH M A >3 * PI>3 . TI= M7A I $HCA $HI3M1@ 1B6 ! <1CM B $PB3M CB I4 / C= R1H6I 1B 3MQ7 * PI>31@ BI M PA 7B M PhD thesis, University Pompeu Fabra, Department of Technology, Barcelona, Spain, sept. 2003.

[9] H. Cohen. A self-defining game for one player: on the nature of creativity and the possibility of creative computer programs.) 7CB1H6C * PI>3 (CPH= B1@35(1):59–64, feb. 2002.

[10] P. Dickinson. Style-modulation: an approach to stylistic pluralism. / <7 * PI>31@ / >A 7I, 130(1754):208–211, apr. 1989.

[11] S. M. Eisenstein, W. I. Pudowkin, and G. W. Alexandrow. Manifest zum Tonfilm. In F.-Z. Albersmeier, editor, / 7SM7 LPH / <7CH7 67I $ >@A I. Reclam, 3rd edition, 1998. 1st release 1928.

[12] J. Freeman-Attwood. Review: St Matthew Passion. / <7 * PI>31@/ >A 7I, 131(1767):265, may 1990.

[13] A. Gartland-Jones. MusicBlox: A Real-Time Algorithm Composition System Incorporating a Distributed Interactive Genetic Algorithm. In G. Raidl, editor, , HC3776>B; I C9 #CC0 CH?= I<CDI L#PHC%, OVVN5 KM4 #PHCD71B ! CB97H7B37 >B %7B7M B , HC; H1A A >B; , pages 490–501, Berlin, Germany, 2003. Springer.

[14] A. Gartland-Jones and P. Copley. The Suitability of Genetic Algorithms for Musical Composition. ! CBM7A DCHIHT * PI>3 - 7Q>7R, 22(3):43–55, 2003.

[15] K. Hartmann, D. Büchner, A. Berndt, A. Nürnberger, and C. Lange. Interactive Data Mining & Machine Learning Techniques for Musicology. In ! CB97H7B37 CB 'BM7H6>I3>D@B1HT * PI>3C@@;T OVVJ E! '* VJ F, Tallinn, Estonia, aug. 2007.

[16] D. Hörnel.) 7HB7B A PI>?1@I3<7H . M4P?MPH7B PB6 . M4@ A >M B7PHCB1@7B + 7MU7B. Shaker Verlag, Aachen, 2000.

[17] D. Hörnel and P. Degenhardt. A Neural Organist improvising baroque-style melodic variations. In , HC3776>B;I C9 M47 'BM7HB1M4CB1@! CA DPM7H * PI>3 ! CB97H7B37, pages 430–433, Aristotle University, Thessaloniki, Greece, 1997. International Computer Music Association.

[18] D. Hörnel, J. Langnickel, B. Sieling, and B. Sandberger. Statistical vs. Connectionist Models of Bebob Improvisation. In , HC3776>B;I C9 M47 'BM7H= B1M4CB1@! CA DPM7H * PI>3 ! CB97H7B37, pages 244–247, Beijing, China, 1999. International Computer Music Association.

[19] K. Jørgensen. On the Functional Aspects of Computer Game Audio. In P6>C * CI M@D OVVK4 ! CB9G CB . CPB6 >B %1A 7I, pages 48–52, Piteå, Sweden, oct. 2006. Interactive Institute, Sonic Studio Piteå.

[20] M. Z. Land and P. N. McConnell. Method and apparatus for dynamically composing music and sound effects using a computer entertainment system. United States Patent Nr. 5,315,057, may 1994. filed nov. 1991.

[21] Z. Lissa. IM47M4? 67H $>@A A PI>?. Henschel, Leipzig, Germany, 1965.

[22] S. R. Livingstone, R. Muhlberger, and A. R. Brown. Playing with Affect: Music Performance with Awareness of Score and Audience. In PI= M4I@I>1B ! CA DPM7H * PI>3 ! CB97H7B37, Queensland, Australia, 2005.

[23] M. V. Mathews and L. Rosler. Graphical Language for the Scores of Computer-Generated Sounds. , 7HID73M4Q7I C9 + 7R * PI>3, 6(2):92–118, Spring–Summer 1968.

[24] W. A. Mozart. Musikalisches Würfelspiel: Anleitung so viel Walzer oder Schleifer mit zwei Würfeln zu componieren ohne musikalisch zu seyn noch von der Composition etwas zu verstehen. Köchel Catalog of Mozart's Work KV1 Appendix 294d or KV6 516f, 1787.

[25] H. Pauli. Filmmusik: Ein historisch-kritischer Abriß. In H. Chr. Schmidt, editor, * PI>? >B * 1II7BA 76>7B. Schott, Mainz, Germany, 1976. modified in 1977 and 1981.

[26] N. J. Schneider. &1B62P3< $>@A A PI>? '4 * PI>?= 6H1A 1MPH;>7 >A B7P7B " 7PM13<7B $>@A . Verlag Ölschläger, München, Germany, 2nd edition, 1990.

[27] W. Thiel. $>@A A PI>? >B %7I3<>3<M7 PB6 %7;7B= R1HM Henschelverlag Kunst und Gesellschaft, Berlin, Germany, 1981.

[28] Vienna Instruments, 2007. http://www.vsl.co.at/.

[29] J. Wingstedt. Narrative functions of film music in a relational perspective. In '. * # , HC3776>B;I8 . CPB6 0 CH@I MC " >I3CQ7H, Spain, 2004. International Society for Music Education.

[30] J. Wingstedt, M. Liliedahl, S. Lindberg, and J. Berg. REMUPP—An Interactive Tool for Investigating Musical Properties and Relations. In + '* # =V:8 + 7R 'BM7H91 37I 9CH * PI>31@#SDH7I= I>CB, pages 232–235, Vancouver, Canada, 2005. NIME -05.

[31] R. Wooller and A. R. Brown. Investigating morphing algorithms for generative music. In / <>H6 'M7H= 1M4CB4 / <>H6 'BM7HB1M4CB1@! CB97H7B37 CB %7B7H= 1M4Q7 . TIM7A I >B M47 #@73M4CB>3 HM1, Melbourne, Australia, dec. 2005.

Sound-based Gaming for Sighted Audiences – Experiences from a Mobile Multiplayer Location Aware Game

Inger Ekman, inger.ekman@tml.hut.fi
Telecommunications Software and Multimedia Laboratory,
Helsinki University of Technology
P.O.Box 5400
FIN-02015 HUT

Abstract. Game audio has been suggested as a means of enhancing the play experiences on mobile devices. However, the field has seen little practical research on the subject. This study investigated the role of sound design in a mobile pervasive game in a prototype mobile game called The Songs of North. We examine the challenges faced by designers of mobile sound and demonstrate how some of the challenges can be overcome. Our design demonstrates how using sounds as a primary information channel can facilitate the use of physical movement as a main game mechanics. However, results from user tests highlight that using sound to convey information is an unfamiliar game mechanic to sighted players. Pervasive game sound also challenges the players' personal sound environment as it competes with music and causes social disturbance. We discuss the implications on mobile game sound in general and suggest directions for future work.

1 Introduction

The last years have seen a marked increase in interest in game audio. Until recently, sound has often played only a marginal role in the game design and development process. Lately sound has been receiving more attention in game development and sound is considered an important component in creating pleasurable gaming experiences. Last years have also seen a growing interest in game audio research. A strong motivator behind this interest is still game accessibility (see e.g. the IGDA accessibility white paper [17]), as sound is sought as a means of making computer games accessible to visually impaired players. However, another application often referred to in relation to the future of audio gaming is mobile games. The growing popularity of mobile gaming encourages development in this area, and developers and researchers alike are seeking ways in which mobile games could compete with pc or console gaming. Since graphics display capacities of mobile devices are limited, alternatives such as sound are explored with great interest. Audio does indeed seem promising for these new platforms. Some authors have even suggested that audio-mostly games could turn out to be much more immersive than visual video games, since the lack of visual information will provide more room for player imagination [21][25]. Another benefit with audio games is that they do not require the player to stay in front of a screen, thus allowing 360 degrees of freedom [25]. This also allows the use of more physically active forms of game interaction, such as a wider range of gestures. Another possibility is to incorporate movement from place to place as a gaming element, such as was done in the games *Pirates!* [4], *Can You See Me Now* [12], and *The Songs of North* [8][20].

Despite the inspiring potential of mobile gaming, nearly all practical research on audio gaming has been made on pc, using tools and hardware far beyond those available for mobile gaming. Even with research primarily considered for mobile use, implementation and testing has been performed on pc. We agree with several authors [26][1] that mobile platforms are bound to evolve, perhaps faster than their tabletop counterparts. However, we argue that working with functioning applications and being able to test in real use situations is fundamental for gaining an understanding of sound in the context of mobile gaming. This work addresses the lack of practical work and implements a mobile multiplayer game. We investigate the development of an audio-mostly game for sighted audiences and examine the role of sound throughout the process. To assess the freedom provided by audio, the game uses physical movement as a main form of interaction. The game is implemented as a playable prototype, which is complete with story and main game mechanics to facilitate a real playing experience.

The structure of the paper is as follows: First we take a brief look at related research in the areas of game sound and mobile audio. We then describe the two prototype implementations of our game, The Songs of North. Next, we consider the iterative process of sound design and cover some crucial implementation issues along the way. Finally we discuss how players felt about sound-based gaming based on test results from two game tests. Last we conclude with a discussion on the future of mobile sound and provide some directions for further research.

2 Related research

The development of games for visually impaired players has produced several studies investigating audio-based gaming for different genres such as action [2][1], arcade shooter [23] and mystery/adventure [7]. A more conclusive list of available games can be found at [3]. Much of the research effort has been finding ways to express various forms of game information in audio form and produced valuable knowledge for audio interface design. Especially game audio has contributed to research on audio-based navigation, whether in 2D [24] or 3D environments [1]. More general questions arise around the combination of informative sound with aesthetics [13]. Hand-in-hand with the practical work on game audio have been theoretical investigations on the functional aspects of game sound and meaning construction [9] [19] [14].

Whereas most of this research has focused on sound for visually impaired, many of the methods will benefit user interfaces designed for sighted people as well. However, when relying on audio information is concerned, skill levels may be different for sighted players. Also, there is little knowledge about sighted players' attitudes towards audio information. In response to outright questions, Cunningham et al. [5] found that only a few players were interested in playing audio games. There is also little information about how well sighted people can use audio-based game information for play. Further, most of this information comes from situations not involving visual information at all. Hadrup et al [15] designed a real-space auditory game, *Dark Circus* for sighted players with an intention to provide new user experiences and alternative modes of interaction. However, the game required players not to see each other, so they used a pitch-dark room as game area. Oren [24] investigated the reception of the same audio signals with and without graphics for a platform game. The audio-only group consisted of sighted, blindfolded players, who ended up using double the time completing the game as compared to the non-blindfolded control group.

Most of the research on audio-only or audio-enhanced playing has focused on the use of sound in conventional pc or console gaming. Liljedahl et al. [21] and Roden et al. [26] both explicitly consider the use of audio for gaming on mobile devices, but their implementation was implemented on modern pcs. Hiipakka, Lorho & Holm [16] implemented sonification of a memory game for a Compaq iPAQ, but the game itself was developed mostly as a means of testing auditory menus and did not make use of mobility in the design.

Sound has also been used in some pervasive games. *Can You See Me Now* consisted of chasers who by moving in the physical world were attempting to capture players navigating a virtual model of the city via the Internet. The chasers used walkie-talkies to communicate with each other, while the players could overhear this conversation and make game decisions based on what they heard. [12] In *Pirates!* players guide their pirate ships through oceans by walking around in the physical space; arriving on islands would notify players of this with a sound “Land ahoy!”. [4] The pervasive live-action role playing game *Prosopopeia* also employed sound-related props in interesting ways, e.g. by using a modified reel-to-reel tape recorder equipped with a hidden cellular phone to receive messages from the spirit world. [18] None of these games, however, revolve extensively around audio information.

3 The Songs of North

The Songs of North [8][20] is a multiplayer enhanced reality game, in which players take the role of shamans inhabiting the spirit game-world. The game draws on inspiration from the Finnish mythology, especially the epic Kalevala. The spirit world (game world) exists in parallel with the physical world and players use physical movements to navigate the spirit world.

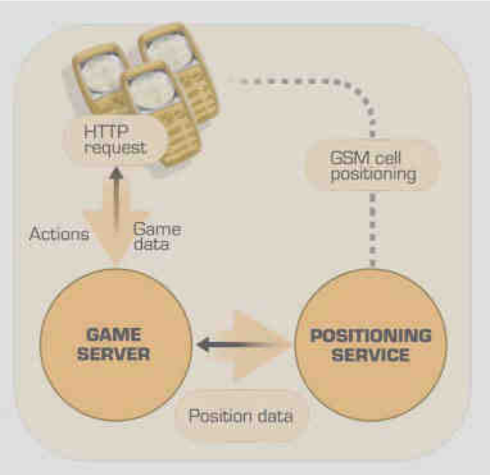


Figure 1. Overview of game architecture.

Players’ movements are detected and fed into the game using GSM cell positioning. In addition to moving around in the physical world, players interact with the spirit game-world by drumming a magic drum, implemented in the phone’s interface.

Players can interact with each other as well as with the many non-player characters (NPCs) of the world. The interaction can take the form of fights or collaboration. The game also supports messaging between players. The story of the game revolves around the legendary Sampo, a machine that is able to produce all the riches of the world. Sampo has been destroyed in a battle and pieces are now scattered all around the world. These pieces are powerful magic elements, but they also have destructive power. The players now have two options: As they find the pieces of Sampo, they either return them to the earth by dropping them into swamps or keep the pieces of Sampo and use the magic to gain personal power (at the expense of the well-being of the world). In addition to either destroying or keeping pieces of Sampo, the game contains various quests. Some quests cannot be performed alone, and require co-operation with other shamans

The game is designed to be played continuously, so that gameplay will integrate with other activities of the player. Interaction with the game requires moving around a large area, with in-game content laid out over the city distances over 10km. Movement by any means of transportation, although interaction primarily designed for slower velocities (walking, by bike).

The spirit world exists in parallel with the physical world. Some places in the real world have significance to the game. For example, a portal to the underworld may be situated in a graveyard, water spirits inhabit areas near lakeshores, and so forth.

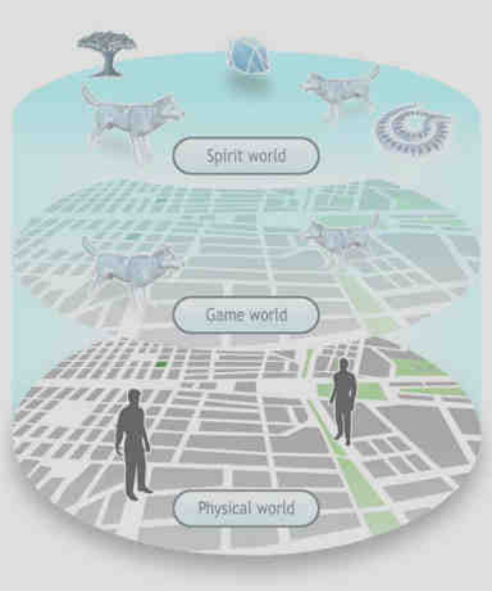


Figure 2. Schematic view of the game world. The spirit world is superimposed on the physical world, creating a game space where physical location and game location intertwine.

4 Sounds in The Songs of North

The sounds of the game were designed with two intentions in mind: First, the sounds should encourage players to use listening as a game mechanic. Second, the game sounds should also invoke a sense of presence, allowing the players to experience the game world as an enhanced component of physical reality. Game sounds provide a link between the real and the fictive; they provide the fuzzy area where game events mix with the texture of ordinary life.

Game sounds are used both in the shaman drum interface, as well as to convey information about the game. The basic idea behind the audiovisual design of the game is that the spirit world can be heard real-time, whereas getting up-to-date visual information requires casting spells with the shaman drum (mobile device). In reality, audio is never completely up-to-date; without player activity the client application will only poll the game server once every 30s. However, audio information is used in such a way as to create an illusion that the player constantly has a real time contact to the game world. Since the game will automatically get information about new events, sound allow the players to just leave the game running and monitor the situation as it develops, either over time or due to their movements in the physical world.

This design decision liberates players from watching the game device while on the move. However, since the game's visual interface is available at all times (in the player's pocket) we avoid some problems of audio-only information. Players can always use visuals and confirm any ambiguous sound information by investigating the drum interface. Visual support also makes introducing new sound signals less of a problem than it is in audio-only interfaces; it is up to the player to decide when they can rely on audio-only information without consulting the visual display. Moreover, performing actions in the game will require drumming and attending to the mobile device. Thus, whenever players want to issue spells, they will also be in a position to watch the device and receive visual feedback.

Playing-by-listening also provides a compromise between active playing and not attending to the game at all, which is crucial for the pervasive nature of the game. In order to encourage play throughout the day, players must be able to perform their everyday activities while playing without loosing a sense of what is happening in the game.

The game features four main types of sound:

- The sounds of the players’ own drum,
- Event sounds such as the sound of drummed spells and unique sounds for various spells and attack types,
- Presence sounds signifying that certain characters, places and items are nearby, and
- Ambient environmental sounds that represent the world’s current state (good, neutral or evil)

4.1 Technical platform

A founding decision throughout the project was that we wanted to implement a functional prototype in order to facilitate testing in a realistic setting with actual players. The game client was implemented in Java over the MIDP1 (Mobile Information Device Profile) and runs on Nokia N-Gage (Series 60) phones. This combination provided the best available solution at the time of our research.

From a sound design point of view, the situation was challenging: The Java-implementation (MIDP1) for Nokia N-Gage did not support more than one sound channel, neither did it provide much control over sound playback. With MIDI we had continuous control over volume, but other controls were lacking. We could also set the volume of linear files, however this could not be modified during playback. After the test, some players reported technical issues (hardware or firmware problems) that caused sound files to always play on full volume. The phone's space limit for a single Java application was approx. 3M. This capacity had to be divided between the whole game including code, graphics and sounds. As is typical of mobile phone speakers, the sound output was optimized around speech frequencies and all lower frequencies inaudible.

Part of these limitations were due to the setup we chose; e.g. using Symbian instead of Java would have allowed much greater control over sound playback (Symbian was not an option due to time limits). Nevertheless, on a more general level the situation is illustrative of the kind of limitations faced in mobile environments [6].

4.2 Sound implementation

Initially, we opted on doing all game sounds in MIDI, however some way into development we noticed that there were problems with memory management associated with these files. Using more than around 10 MIDI files would cause the game to freeze and the only means of releasing the files was to shut down the game application. We thus turned to wav and used MIDI only for a few sounds that required continuous volume control.

A challenge with mobile sound is coping with various listening environments. In order to minimize the risk that sounds were masked by environment noise, we encouraged players to use headphones. This also sought to minimize social disturbance caused by sound playing on public places.

We had no means of knowing which direction the player was facing. Because of this, we decided against using stereo effects for sound positioning as this could have been falsely interpreted as direction advice. Also, since only one file could play each moment, we did not have to concern about separating different sound effects by location in the auditory field.

4.3 Representing information

Since only one sound could play at each time, sounds had to be queued in order of play priority. As the drum is the player’s primary tool for manipulating the game world, a design decision was made to give the topmost priority to the player’s own drum hits. Whenever a player uses the drum to make a spell, all other game sounds quiet down and the drum sound is played. To guarantee the legibility of game sound information, the drum interface sound is the only with power to interrupt playback; all other sounds play out fully. After drum sounds next priority is held by the spell sounds of self-issued spells (indicating spell success or failure). This guarantees maximum feedback on player actions, since the only sound to interrupt self-issued spells is the drum sound of the player initiating a new spell.

In all other cases, sounds were prioritized according to the in-game context. In many cases it is acceptable that sounds that have no immediate effect on player status be played back after a small delay. This is due to the fact that game time is never guaranteed to be the same for all players. The client will poll the server for game status when a player issues spells, or otherwise automatically poll the server every 30s. Most often the player activity will match the amount of events so that intense situations create more polls and thus the client’s view of game situation stays up to date. However, in a worst-case scenario, a single poll can return more event data than can be played immediately. In general, priority is given to event sounds (spells) targeted at the player character, but in some cases presence of important objects (pieces of Sampo) will override less critical event sounds. The decision to always play self-initiated spell sounds first means that in intense situations the player's awareness of their own actions is prioritized over incoming spells. Thus they may not always be (sonically) aware of their character player's health, but will always know whether or not spells are issued correctly. On the other hand, player status information is made very clear in the visual interface.

Apart from event sounds, all objects, characters and places produce sounds when they are in the proximity of the character. Every time the client polls the game server, the audio queue is re-examined to decide, what sounds should be played. When no events require attention, sounds of objects, as well as the state sound describing the nature of the game world are queued depending on object priority and chronological order, making the most recent sounds last in the queue. Sounds are popped and played from the queue with a random amount of silence between them. To avoid unnecessary repetition, the ambient sound (representing the game world's current state), which would otherwise always be in the queue, was played only once every three minutes.

4.4 Iterations between prototypes

The first implementation of Songs of North, along with results from the first player tests, is described in detail in [8]. Next we will consider some iterations made between prototypes.

The sound design of the first version relied heavily on the Nordic mythology and Finnish nature for its sounds. The game interface showed a traditional shaman drum made of wood and animal skins, on which the game world would be depicted by the pattern of bones lying on the skin. However, without bass, we were not able to make the drum sound as big as we would have liked. Together with the reality that players were in fact pushing the buttons of the mobile phone, we felt the drumming could be developed further. For the second prototype, we wanted to make the drum interface better integrated in the game story and make drumming part of the magic.

The new spirit world in the second prototype still borrows many sounds from nature. However, the emphasis is now taken upwards, with sounds inspired by winds and flying. This accentuates that magic is something that goes on 'up there', and resonates well with how people think about mobile telephony. Similarly, to better match with the technical nature of the mobile interface and keypress-drumming, the drum was given a techno-treatment. The new interface suggests the player is something of a DIY shaman; pounding a drum made of scrap metal players use keys and screws to magically communicate with the spirit world. The accompanying sounds clank and ring giving the feeling of a technical device possessed with magic powers.

Turning to symbolic sounds allowed a more systematic approach to the representation of game information. Sounds design was guided by informative and aesthetic values, but without the limitation of having to produce 'realistic' animal sounds. In the new version, all character sounds share some basic acoustic properties that make them easy to recognize as character sounds (instead of character sounds just relying on similar iconic relationships as was the case with animal sounds). We also made the new character/item sounds shorter and more musical than before. In addition to making them more socially acceptable, we assumed that making sounds less iconic could make them somewhat more tolerant of repetition.

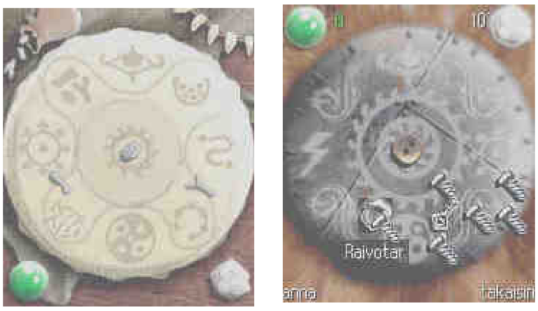


Figure 3. The two visual versions of the shaman drum interface reflect changes in the theme: The first prototype depicts game items as bones and gold nuggets on a drum made of animal skin. In the second prototype the drum was made of metal and bones substituted by buttons, screws and keys – things you might find in your pocket.

To further support playing-by-listening, some crucial combinations of character and item sounds were layered to provide new sonic symbols. Especially, characters holding valuable pieces of Sampo would reveal their possessions through modified presence sounds. Another special case of sonic symbols is the spell sounds. Spells are layered with information to support various levels of listening. The harmonic structure is informative of the nature of the spell, with especially attack spells markedly dissonant. Further, spell timbre will give more detailed information of the spell's function and effects. Spell sounds also carry rhythmic signature from drumming, so listening to the spell more closely will reveal how complex the drumming pattern was that issued them. On a basic level, just listening to the length and complexity of the spell in general will tell players something of the opponent's skill level. Further, with sufficient training, it should be possible to recognize some spells just by paying attention to the pattern.

A final major change was concerned with how to present the different roles visual vs. audio information. Specifically, the first version contained some visual information in the form of animations. While the animations were very informative, such as showing a slowly dying fire after being hit by a fire spell, they also caused people to believe they were looking at realtime information, when in fact the visual interface was static and information. This lead to serious problems, as players would use the static display on their shaman drum for navigation without realizing that the positions of items had nothing to do with the current game situation.

In order to avoid these serious problems, we decided to skip the animations and make more distinct the fact that visual interface was static in order to minimize false conclusions. We also devised new play rules that ensured that the game would not go silent for long times, whereas the drum would deactivate after a while.

5 Testing the game

The gameplay experience and playability of the iterated prototype was evaluated during a two-week play-test period. 16 players were provided a Nokia N-Gage and a SIM-card each for the play-test period. Players were informed that the sounds carried information about the game world. The game website also provided a few example sound files for players to get familiar with outside the game, but at least some players remained unaware of this option throughout the testing period.

On the whole players were critical about the sounds of the game. Part of the low acceptance of sounds had to do with the way sounds were prioritized. It soon became clear that the methods for selecting which presence sounds to play were inadequate to deal with the more crowded areas of the game world. During the game, some players also carried items with them to their homes; this stashing habit soon turned the game areas responding to that location into areas of sonic chaos.

In general, most sounds had been interpreted they way they were intended to, often with the help of the visual interface. In general, the symbolic sound design was well received, but all sounds were criticised for too much repetition. Listening for pieces of Sampo was a special case. Pieces of Sampo held much significance in the game and accordingly responses to Sampo sounds were markedly excited. An interesting aspect of how sounds would be described was that many players made references to sounds that were clearly influenced by the physical location of the sound. For example, our game had swamps, where players could destroy pieces of Sampo. Depending on the location where players had heard this sound, they would describe it as everything from pouring beer into a glass to flushing a toilet. Whereas traditional games have rather strong means of fixing their sound references this is not always the case. Especially in augmented reality games, as the sounds become part of the environment, the environment influences how they are understood.

Since the game was addressed at an audiences of sighted people, we were interested in how players would react to using sound. In our experience, sighted people are initially rather uncomfortable using sound as their primary information channel. This attitude was clearly observable also in the player interviews. Using sound as a source of information divided players. While some players reported actively using sound as a means of playing, others had turned off the sounds after only a few times of listening. For some players, it was clearly preferable to monitor the game world by looking at the device even if it meant interrupting what they were doing.

Despite general suspicion towards sound information, most players nevertheless mentioned using sounds for some purpose. Sounds of spells were used as a means of knowing whether a spell had been successful or not. This is rather natural for interface sound. However, one player mentions using this feature to cast spells without looking at the mobile device at all. Most players also make some comment about listening for characters, places and items. However, this function seems to have been most useful in the outskirts of the city, where items were sparser.

There were many reasons for players not to use sound. The most common was that players preferred listening to their own music over game sounds. One player stated that the reason for not listening to sounds were that the headphones felt unpleasant. For some players, the redundancy of information, i.e. audio not providing anything extra, was seen as a reason against listening.

6 The future of mobile game sound?

While some people complained of social disturbance and annoying audio, the most common reasons for turning off the was to listen to music. Notably, sceptic voices within the industry such as mobile sound designer Peter Drescher [6] have argued that mobile game sound will have little function, when personal audio /music listening will allow people to choose their favourite band over game beeping. If this were to happen, it would have tremendous impact on sound design in general, but pose especially grave questions on the possibility of pervasive gaming. These games are dependent on maintaining a sense of awareness over extended time and not being able to use sounds would seem detrimental to this effect.

Persuading players not to turn off game sounds will require a shift in design attention. Whereas sound designers now work with an awareness of the hardware settings of the players, pervasive games will require also paying attention to the various playing contexts of the game. Designing for a mobile context thus involves more than making sounds that sound as desired on mobile speakers. Because mobile games are played within a social setting, sounds should be designed to fit into the environment of everyday activity. This is especially true for pervasive games.

A simple example of added contextual awareness would be to monitor background noise levels for decisions about suitable playback volume. Another possible context to design for is the presence of other media products while gaming. One obvious possibility is to allow more choice of music within the game. However, mobile devices could also be designed to facilitate for several sound streams at once, making it possible to play both personal music and game audio from various processes at once. With inventive use of location information and analysis of background sounds, games could become sensitive to the play context. Some inspiration for the integration of game sounds with contextual information can bee sought in the works on wearable and adaptable sound environmens. *Sonic City* [22] explored mobile interaction and wearable technology as a tool for music creation. *Sonic City* is a music system, which creates a personal soundscape based on a persons physical movement, local activity and urban ambiance. In another project, *Nomadic Radio*, different information sources received by the mobile device are combined to construct a dynamic information experience to the user [27]. Wearable music devices can also monitor body state and influence it via the choice of music, like the *BodyResT* system designed for relaxation [11]. This mechanism would optimally be part of a larger control system for negotiating game sound depending on listening environments (noisy/quiet), social settings (e.g. busy/available) and in game context (events). While some contextual information is easy to deduce, such as information about noise levels, much work is still needed before games will be able to react correctly the varying social contexts of gaming. Also further research is required to properly understand the construction of pervasive mobile experience and how game sound influences the perception of everyday life.

7 Acknowledgements

In addition to the author, the following people were involved with the design and implementation of The Songs of North prototype: Petri Lankoski (game server, project lead); Jussi Lahti (client application, lead game designer), Jani Nummela (graphics). The research project was funded by National Technology Agency TEKES, Codetoys, the Finnish national lottery Veikkaus, Nokia Research Center and TeliaSonera Finland.

8 References

[1] Andresen, G. Playing by Ear: Creating Blind-Accessible Games. Gamasutra, May 20, 2002.

[2] Atkinson, M.; Gucukoglu, S.; Machin, C. & Lawrence, A. Making the Mainstream Accessible: Redefining the Game. Sandbox Symposium, Boston Massachusetts, July 29-30, (2006)

[3] Audio-Games website: www.audiogames.net. [Accessed August 22, 2007.]

[4] Björk, S.; Falk, J.; Hansson, R. & Ljungstrand, P. Pirates! Using the Physical World as a Game Board. Interact 2001, July 9-13, Tokyo, Japan, (2001)

[5] Cunningham, Stuart; Grout, Vic & Hebblewhite, Richard. Computer Game Audio: The Unappreciated Scholar of the Half-Life Generation. Proc. AudioMostly, Piteå, Sweden, 2005.

[6] Drescher, P. Could Mobile Game Audio BE More Annoying?! http://digitalmedia.oreilly.com/pub/a/oreilly/digitalmedia/2006/04/26/could-mobile-game-audio-be-more-annoying.html?page=1 (2006) [Accessed August 22, 2007.]

[7] Drewes, T.; Mynatt, E. & Gandy, M. Sleuth: An Audio Experience. Proc. International Conference on Auditory Display. Atlanta, GA, USA, (2000)

[8] Ekman, I.; Ermi, L.; Lahti, J.; Nummela, J.; Lankoski, P. & Mäyrä, F. Designing Sound for a Pervasive Mobile Game. Proc. ACM SIGCHI Advances in Computer Entertainment, Valencia, Spain, 110-116, (2005)

[9] Ekman, I. Meaningful Noise: Understanding Sound Effects in Computer Games. In Digital Arts and Cultures, Kopenhagen, Denmark, (2005)

[10] Eriksson, Y & Gärdenfors, D. Computer Games for Children with Visual Impairments. Proc. 5th International Conference of Disability, Virtual Reality & Associated Technologies, Oxford, UK, 79–86, (2004)

[11] Fagerlönn, J. BodyResT - A prototype using music responding to heart rate for stress reduction. Masters thesis, Luleå University of Technology, Dept. Computer Science / Media Technology. (2005)

[12] Flintham, M.; Anastasi, R.; Benford, S.; Hemmings, T.; Crabtree, A.; Greenhalgh, C.; Rodden, T.; Tandavanitj, N.; Adams, M. & Row-Farr, J. Where On-Line Meets On-The-Streets: Experiences With Mobile Mixed Reality Games. Proc. SIGCHI conference on Human factors in computing systems, Ft. Lauderdale, Florida, USA, 569 – 576, (2003)

[13] Friberg, J. & Gärdenfors, D. Audio Games: New Perspectives on Game Audio. Proc. Advances in Computer Entertainment Technology. Singapore, (2004)

[14] Grimshaw, M. (2007). The acoustic ecology of the first-person shooter. Unpublished PhD thesis, University of Waikato, New Zealand, http://www.wikindx.com/mainsite/acousticEcologyOfTheFirst-PersonShooter.pdf (2007) [Accessed August 22, 2007.]

[15] Hadrup, R.; Jakobsen, P. S.; Juul, M. S.; Lings, D. & Magnúsdóttir, ?. Designing an Auditory W-LAN based Game. http://www.soundk.com/papers/manchester.pdf (2004) [Accessed August 22, 2007.]

[16] Hiipakka, J.; Lorho, G. & Holm, J. Auditory Navigation Cues for a Small 2-D Grid: A Case Study of the Memory Game. Proc. International Conference on Auditory Display. Kyoto, Japan, (2002)

[17] IGDA. Accessibility in Games: Motivations and Approaches. International Game Developers Association white paper. http://www.igda.org/accessibility/IGDA_Accessibility_WhitePaper.pdf (2004) [Accessed August 22, 2007.]

[18] Jonsson, S.; Montola, M.; Waern, A. & Ericsson, M. Prosopopeia: experiences from a pervasive Larp. Proc. ACM SIGCHI Advances in Computer Entertainment Technology, Hollywood, California, USA (2006)

[19] Jørgensen, K. On the Functional Aspects of Computer Game Audio. Proc. AudioMostly, Piteå, Sweden, (2005)

[20] Lankoski, P.; Heliö, S.; Nummela, J.; Lahti, J.; Mäyrä, F. & Ermi, L. A Case Study in Pervasive Game Design: The Songs of North. Proc. NordiCHI, Tampere, Finland, 413–416, (2004)

[21] Liljedahl, M.; Papworth, N. & Lindberg, S. Beowulf – An Audio Mostly Game. Proc. ACM SIGCHI Advances in Computer Entertainment, Salzburg, Austria, (2007)

[22] Mazé, R. & Jacobs, M. Sonic City: Prototyping a Wearable Experience. Proc. 7th IEEE International Symposium on Wearable Computing, (2003)

[23] McCrindle, R. J. & Symons, D. Audio Space Invaders. Proceedings of the 3rd International Conference of Disability, Virtual Reality & Associated Technologies, Alghero, Italy, 59–65, (2000)

[24] Oren, M. Speed Sonic Across the Span: Building a Platform Audio Game. Extended abstracts, Conference on Human Factors in Computing System, San Jose, CA, USA, 2231 – 2236, (2007)

[25] Röber, N. & Masuch, M. Playing audio-only games: a compendium of interacting with virtual, auditory worlds. Proc. DiGRA 2005 Conference: Changing Views – Worlds in Play, (2005)

[26] Roden, T.E.; Paberry, I. & Ducrest, D. Toward mobile entertainment: A paradigm for narrative-based audio only games. Science of Computer Programming 67, 76-90, (2007)

[27] Sawhney, N. & Schmandt, C. Nomadic Radio: Speech and Audio Interaction for Contextual Messaging in Nomadic Environments. ACM Transactions on Computer-Human Interaction, 7 (3), 353–383, (2000)

[28] Targett, S. & Fernström, M. Audio Games: Fun for All? All for Fun? Proc. International Conference on Auditory Display, Boston, MA, USA, (2003)

Sonic interaction design: case studies from the Medialogy education

Stefania Serafin, Smilen Dimitrov, Steven Gelineck, Rolf Nordahl and Olga Timcenko.
Medialogy, Aalborg University Copenhagen,
Lautrupvang 15, 2750 Ballerup, DK,
sts@media.aau.dk

Abstract.
In this paper, we share our experience with students' projects undertaken during the Spring Semester 2007 of the Medialogy education at Aalborg University in Copenhagen. We describe different projects which combine problem based learning with interaction and sound design.

1 Introduction

Medialogy is an education which was established at Aalborg University in Denmark in 2002. The main aim of Medialogy is to combine technology and creativity in the design, implementation and evaluation of interactive media products.

As part of the Medialogy education, the problem based learning (PBL) approach is adopted [3], in which students spend the first half of a semester following courses, and the second half solving a problem through a project.

In this paper, we introduce different projects implemented during the Spring semester 2007 in the 4th Semester of the Bachelor Medialogy education at Aalborg University in Copenhagen. Three sub-themes for projects were suggested: sound in games, alternative musical instruments and sound in products. Students were divided in 8 groups of 5-6 people each. The projects implemented by the students covered all these three categories.

The theme of this semester was Interaction design. Students were asked to find problems related to the design of a physical interface embedded with sensors, with auditory feedback, and evaluated using usability techniques.

2 Interaction design

In the pedagogical system developed at Aalborg University, each semester has a theme. Such theme is supported by several courses, known as project supporting courses (abbreviated as PE courses in danish). All students, working in groups, develop along the semester a project which is related to the theme and uses knowledge and experience obtained following the courses.

In this particular case, the theme of the semester was Interaction Design. Students learned how to design and build new interfaces embedded with sensors. At the end of the semester, they built physical objects which could be touched, squeezed, moved around, and which sent information of such actions to the computer. Students also acquired skills in sound design and how to program sound effects. Moreover, they learned about design principles, usability studies and evaluation techniques for interaction design.

The courses described in the following section are the PE courses supporting the project.

3 Description of the courses

3.1 Measurement of user experience

Measurement of user experience is a 2 ECTS course. The main goal of the course is to persuade students that probability of developing really useful products highly increases by involving all stakeholders into the design process as early as possible, even at the idea stage. [5] is used as a textbook for this course, as it fully advocates the main goal of the course. But, somehow students tend to believe that usability testing could be done only at fairly advanced prototypes however, in order to show them the opposite, an example from praxis is used. [2] describes how professionals from a major mobile-phone company (Nokia) are using paper-prototypes for testing a new WAP-phone. That article is used as additional course material. The students are encouraged to perform different usability procedures and testing several times during the project work.

3.2 Physical Interface Design

The Physical Interface Design course is a 1 ECTS course. The course provides the students with knowledge about design principles regarding physical interfaces and physical computing. It is also the aim of the course to connect the other courses of the semester in a meaningful way demonstrating their importance on a more global scale.

During the course the students are presented with design fundamentals that apply to everyday physical interfaces (telephone, coffee machine, bottle opener, etc.) using Donald Norman's *The Design of Everyday Things* [4] as main reference.

Extending these fundamentals the students are furthermore presented with the advantages and additional challenges provided by the integration of new media and the new interaction capabilities, entailed by this integration.

Everything is taught from a user centred point of view. During the course the students are presented with related projects from the real world of physical computing showing where the design is successful or has failed. During the last lecture the students create their first physical interface using a kit of simple sensors and materials provided to them.[1] They produce a video explaining their decisions and present their creation.

3.3 Audio design

Audio design is a 3 ECTS course which introduces students to the concept of digital sound manipulation and synthesis in real-time. As a prerequisite, students have basic knowledge of auditory perception, but have generally no previous experience with sound design and sound synthesis. The class is divided into a theoretical part in which concepts of sound design are taught, and an hands on part in which the theories are implemented in the real-time environment Max/MSP.[2]

The course starts by introducing the concept of digital audio (sampling and quantization) and sound effects in time domain such as loop, delay, echo, flanger, chorus, etc. Then the concept of a digital filter is introduced, and different filters such as low-pass, high-pass, band-pass and comb filter are presented.

In the second part of the course, sound synthesis is taught, and the advantages of creating sounds from scratch as opposed to manipulating existing recordings is discussed. Different synthesis techniques such as additive, subtractive, modulation synthesis (ring modulation, amplitude modulation and frequency modulation), as well as granular synthesis are introduced. For each technique, examples are illustrated of its use in interactive applications such as game and digital arts. Moreover, exercises allow the students to experiment on their own on the technique.

Given the difficulty of finding an appropriate textbook for the course, the instructor decided to write her own lecture notes.[3] This proved to be rather beneficial for the students.

Creating sounds from scratch which are aesthetically pleasing is a rather complex task, especially for unexperienced students. As it will be discussed during the description of the projects, some students were quite successful in achieving this goal, and rather proud of their result. Others preferred to work on existing samples and manipulate them according to their needs.

3.4 Sensors Technology

Sensors Technology is a 3 ECTS course, divided in 15 lectures. As work with sensors implies work with electronics, the course introduces electronics, with special perspective on sensors, and their utilization in context of user interaction with rich media software applications.

This stated coverage of the course content, can sometimes span several semesters and different courses in traditional bachelor electronic engineering studies. To manage to fit all these discussion areas in a meaningful manner in a single semester course, the course is organized as follows. It starts with a brief discussion of the concepts involved in interfacing software and hardware (including data acquisition hardware), and continues with a brief review of atomic theory, including the concept of free electron, and electric properties of matter. Elementary electrostatics is introduced, on a microscopic scale, as a basis for understanding voltage - and extended to the context of conductive materials, and Ohm's law on a macroscopic scale. This leads to a brief overview of circuit theory, where elementary analysis techniques are introduced, from which the voltage divider is central (as being both relatively easy to analyze, and being applicable with a number of off-the-shelf sensors). At the same point, a hydraulic analogy to electric circuits is introduced, as a tool to simplify the discussion of the different electronic elements.

The circuit theory stays mostly on the level of DC analysis with resistors. From that point on, an introduction to the capacitor, diode, and transistor are given, as elementary electronic elements. The main discussion focuses upon their role in an electric circuit, and a corresponding hydraulic analogy; several elementary circuits are provided for each element, along with analysis. Using linear approximations as models (where possible, say for the diode and the transistor) allows that the analysis stays within the linear domain,

[1] Teleo Starter Kit, two potentiometers, one button, one strain gauge. Implemented using Max/MSP.

[2] www.cycling74.com

[3] Available online at: http://www.media.aau.dk/∼sts/ad/

and remains simple. In addition, sensor variants of each element are discussed.

The course concludes with discussion of the operational amplifier as a circuit element, as well as a brief overview of available, more complex sensors (like IR distance detectors), which provide relatively simple electric interfaces. During the course duration, in-class demonstrations (during lectures) and lab exercises (after lectures) are held, where the students solder and measure the circuits discussed in class - which provides the elementary experience with circuit assembly and soldering, prerequisite to the application of sensor technology in the semester projects.

Given the difficulty of finding an appropriate textbook for the course, the instructor decided to compile his own lecture notes, as well as produce several visualisation applets.[4] Since this was also a need for the audio design class, it seems like a suitable textbook covering topics related to the Audio mostly conference is needed.

Students also learn how to interface different sensors to the Teleo starting kit[5] and the Arduino board.[6]

4 Description of projects

According to the pedagogical model developed at Aalborg University, all projects start with the formulation of an initial problem. Through the development of a product, students can address the problem, and evaluate if it has been solved from a human centered perspective. It is therefore important for the students not only to be able to develop and application, but also to put it in a context and properly assess the validity of their solution.

Students were allowed to choose the problem statement they wanted to address. The only limitation imposed was the need for the application to be interactive, to contain an alternative sensorial input and auditory feedback, and to be evaluated according to the techniques learnt in the Measurement of user experience course.

To delimit the choices, three themes were offered:

Three groups worked on sound in games: the first group built an alternative physical interface to a ball game, which was implemented in the form of a squeezable ball with pressure sensors and accelerometers. The second group designed a game in which players were asked to interact at a certain tempo, while the third group designed a game in which players were wearing a physical interface capturing jumping. Concerning the theme of alternative musical instruments, one group designed a ”soundgrabber”, i.e., an interface in which sounds could be moved around in space and thrown away with physically meaningful gestures. A second group designed a new interface to enhance singers’ performances. Concerning the last category (sound in products), one group designed a wobble board enhanced with sensors, in order to be able to control a game with auditory feedback for enhancing the experience of reabilitation. Other groups examined how new forms of interaction and auditory feedback can improve ordering services in a bar or helping reducing the use of electric appliances for reducing global warming.

4.1 Beat-bandit

This group explored the possibility of designing audiovisual games in which audio is an intrinsic part of the gameplay, and investigated if a physical interface can make the user more engaged in the sensation of audio.

To achieve this goal, they designed a game called Beat-bandit, in which the player is asked to shoot some targets following a beat, in a similar fashion as done in the game Rez for Playstation 2 by Sega (2002), but with the aim of designing a less abstract game environment.

The controller designed to play the game was a plastic gun enhanced with IR sensors. The sensors technology adopted showed some lack of precision, which prevented proper testing of the complete setup.

16 subjects tested the game and completed a questionnaire. Figure 1 shows a test subject while trying the beat bandit game with a traditional mouse and keyboard interface versus the gun interface. Overall, the test subjects enjoyed the fact that they were supposed to follow a beat, but had difficulties in understanding properly the rules of the game and which rhythm they were supposed to adjust to.

Figure 1: Subjects testing the beat bandit game.

[4] Available online at http://media.aau.dk/∼sd/st/

[5] makingthings.com

[6] www.arduino.cc/

4.2 Game mechanics' effect on score

In this project, students addressed the statement formulated in [1], claiming that in games that use sound effectively, an expert player's score is lower with the audio turned off that it is when the audio is turned on.

The group was interested in extending this statement and investigating the effect of using an alternate controller together with sound effects and see if this increased the player's score. To achieve this goal, an alternative controller shaped as a hand-held ball was designed, embedded with four pressure sensors and one accelerometer, as shown in Figure 2. Such controller was inspired by the Squeezables developed at the MIT Media Lab [6].

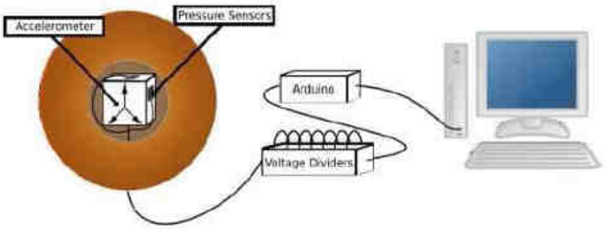


Figure 2: The controller designed to test the game mechanics' effect on score.

The game was tested with novice and expert players (in this case the developers of the game). Results show that novice players had a hard time adjusting to the new controller, while expert players performed slightly better while using the novel controller compared to the traditional keyboard.

The effect of audio on the player 's performance was also tested, and proved to give no conclusive increase when playing with or without sound, both in the expert and novice tests.

4.3 "SSS" - Jumping 2D platform game

This group started with looking for means to increase the physical activity of players during game playing. The resulting product is a traditional 2D game, implemented in Adobe Flash, where the user interacts through a handheld button device, and a couple of FSR sensors fitted on the users shoes. The user can control the game avatar by physically jumping and crouching. The project report, besides game design, discusses traditional 2D platform games, as well as problems of perspective of the user and immersion; audio design of game sounds using FM and granular synthesis; and problems of sensor implementation, fitting, and interfacing.

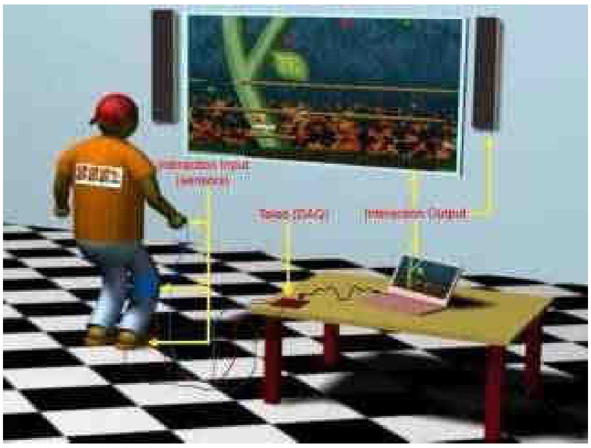


Figure 3: A setup diagram of "SSS" jumping 2D platform game.

4.4 Sound grabber

The main idea behind the sound installation called Soundgrabber is to understand if it is possible to touch a sound by grabbing it from a bucket and positioning in different locations. As can be seen in Figure 4, the Sound grabber is a physical interface designed as a semi-circle, enhanced with four cones. At the bottom of each cone a speaker is placed. Each cone is embedded with light sensor, which allow to detect the distance of objects from the cone itself. The user interacts with the Sound grabber using a glove embedded with a bend sensor. By bending the hand inside a bucket, users are able to grab a sound, listen to it (thanks to the speaker embedded inside the glove) and release it in one of the bucket.

The Soundgrabber was publicly demonstrated at the Sound Days event in Copenhagen in June 2007. Sound experts and naive visitors tried the installation, and provided enthusiastic feedback. Figure 4 shows a visitor at the event.

4.5 Singer interface

The idea of this project was to empower a singer to manipulate her own voice in real time, while singing on the stage. The singer should use a handheld device, in a form of a short stick. The device should, by help of two dual-axis accelerometers, be able to measure 3-dimensional acceleration of singers hands moving during the performance.

Depending on direction of acceleration (up/down, left/right or front/back), different filters could be activated and would modulate the voice. Implemented effects were: delay, flanger and tremolo. The group

Figure 4: The sound grabber at Sound Days.

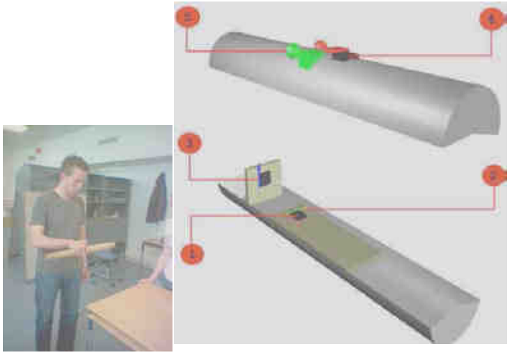


Figure 5: A fellow student testing the device (left) and Cross-section image of the Control Device 1) Sensor 2) X-axis of Horizontal sensor 3) X-axis of Vertical sensor 4) Button 5) LED- diode

worked with a singer and performer, and tested a prototype with her. Based on her experiments with the device, the group concluded that significantly more work is needed before they might produce a useful device. Although in technical sense mapping between performers motions and sound filters works correctly, there is a huge issue how to make this mapping natural, in a sense that a desired effect is achieved during stage-performance, and it feels natural for a singer.

This was yet another example of an ambitious project, where the students did not succeed to leverage the whole potential of the technology they had in hands, and where usability issues were greater than technical matters.

4.6 ”BarZar” Bar Ordering System

This group started by thinking of ways to improve the efficiency of the process of ordering drinks at a bar. The resulting product is a system, consisting of a screen, PC and corresponding button sensors, to be fitted on each bar table for the customers, and one fitted at the bar for the bartender. The software is implemented in Adobe Flash, PHP and a database backend, and both an administration and a user interface are demonstrated. The project report, besides application design, discusses other systems of similar kind; how a typical auditory environment in a bar is taken into account in design of application sounds; and problems of sensor implementation, fitting, and interfacing.

4.7 Wobble active

This is an example of apparently modest, but overall very well executed project. A wobble board is a device for ankle rehabilitation after an injury. It is documented that exercising on a wobble-board is a very efficient way to rehabilitate ankles, both after sport injury or after any other kind of ankle injury. The issue is that the wobble-board training is quite boring, and patients tend to neglect it, as soon as they stop feeling physical pain although prolonged exercises are critical for successful rehabilitation. The students group had an idea to use a wobble board as an input device for a computer game, as they assumed that playing games would increase motivation of doing otherwise boring exercises. They cooperated with a physiotherapist during the project, trying to develop a simple arcade games that would encourage proper exercising. The main idea is that with a help of 4 bending sensors it is possible to measure inclination of the board in space, and to map those inclination into mouse movements on 2D screen. The group also implemented a very interesting way of clicking in the game, by keeping the position constant over a certain field for a certain amount of time. Combining with left/right and up/down motion, the wobble active board this way can perform all functions of a computer mouse. Moreover, as it is well known that balancing is much more difficult task with eyes closed, the group made experiments with sound feedback lower and calmer sounds for equilibrium states, and more loud sounds when the position of the wobble board is approaching undesirable states. The group made a functional prototype together with two tested games, the physiotherapist they have been working with has shown lots of appreciation for their work, and they are now looking for other possibilities in using this project.

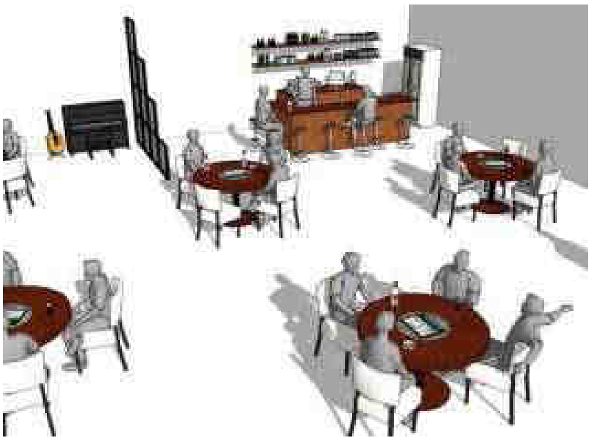


Figure 6: A scenario of use of "BarZar" bar ordering system.

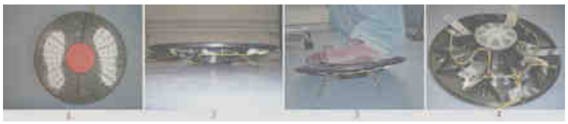


Figure 7: Top view. 2: Side view. 3: WobbleActive in action. 4: The underside.

4.8 AntiC02

The main idea of this project is to design a prototype of a home-system which will inform the users that their energy consumption is too high. For example, one of the cases when the system should react is when the window is open, and the radiator under the window is on. The energy is clearly wasted at that point. Measures could easily be done with several touch and temperature sensors, so that part of the project was not a big challenge. The main challenge is: what the nature of the feedback should be? If the message appears on some screen, it can easily be ignored. Sound comes as a natural choice but sound feedback has an inherent drawback if it is too obstructive or unpleasant, people tend to simply turn it off. Thus the group was trying to find informative, but not unpleasant or disturbing sounds and, finally, after a serial of user-testing, has settled to mimicking sounds from nature wind, raging from breeze to severe wind, and waterfall sound. Interesting, but not sufficiently finished part of the group work was testing different synthesized sounds with potential users, in order to figure out the most both informative and pleasant sound for the purpose of the system. The group also designed an appearance of a screen-saver and background of a touch-screen to be used with their system. The design idea was that clean and calm appearance should indicate that everything is as it should be, and rough, cluttered background indicates that something is going wrong within the system. However, a real product would require significantly more testing on likeability and unobtrusiveness of the system, as several one-hour tests could hardy prove that the system will not be previewed as obtrusive on prolonged daily use. This is an example of an ambitious and well-thought, but fairly poorly executed project.

5 Conclusion and discussion

In this paper, we introduced different projects performed by Medialogy students at Aalborg University in Copenhagen during the Spring semester 2007.

We find that the PBL approach helps students structuring their project work, facilitating them in the formulation an initial problem, which is then solved by designing an interface embedded with sensors tested using standard usability techniques.

A disadvantage is the fact that to acquire so many skills in a single semester is a quite demanding task, both from the students' and teachers' perspective.

As mentioned in the previous section, the scope and quality of the different projects largely varied among the different groups. However, all groups showed the ability to work problem based, and design novel input devices connected to auditory feedback.

References

[1] W. Buxton, B. Gaver, and S. Bly. The Use of Non-Speech Audio at the Interface. *Tutorial Presented at CHI91*, 1991.

[2] T. Jokela, J. Koivumaa, J. Pirkola, P. Salminen, and N. Kantola. Methods for quantitative usability requirements: a case study on the development of the user interface of a mobile phone. *Personal and Ubiquitous Computing*, 10(6):345–355, 2006.

[3] A. Kolmos, L. Krogh, and F.K. Fink. *The Aalborg PBL model: progress, diversity and challenges.* Aalborg University Press, 2004.

[4] Don Norman. *The Design of Everyday Things.* Basic Books (Perseus), 2002.

[5] J. Preece, Y. Rogers, and H. Sharp. *Interaction design: beyond human-computer interaction.* Wiley, 2002.

[6] G. Weinberg and S.L. Gan. The Squeezables: Toward an Expressive and Interdependent Multiplayer Musical Instrument. *Computer Music Journal*, 25(2):37–45, 2001.

Topical Segmentation of Audio for Tertiary Courseware

Katrin Hartmann, *k.hartmann@gcal.ac.uk**
Tom Buggy, *t.buggy@gcal.ac.uk*
School of Computing and Mathematical Sciences, Glasgow Caledonian University

Abstract: We are interested in making available to students structured recordings of tutorial dialogues. Such material is called Tertiary Courseware . It has been shown that the availability of Tertiary Courseware can assist in learning.

Previous attempts to provide Tertiary Courseware have found that it is enormously time consuming to process and prepare recorded discussions. For the use of Tertiary Courseware to become widespread automatic tools for the preparation, indexing and segmentation of the material are essential.

This paper investigates methods for topical segmentation of an audio dialogue. Topical segmentation splits a discussion into non-intersecting units where each unit contains material related to a particular topic. In order that segmentation be automatic and robust we consider simple statistical models of the transcribed text of the dialogues. We survey the literature on statistically-based segmentation, with emphasis on models where the text is considered as a bag of words. A model which has been shown to reflect the occurrence of topical words well is the G-Model of Katz. Katz' G-Model has been applied to topical segmentation of text documents by Reynar with very good results. We use Reynar's technique to attempt the topical segmentation of tutorial dialogues.

We show that the results of using the G-Model as applied by Reynar leads to poor results when applied to tutorial dialogues. An analysis of Reynar's model shows that the statistical model used is inconsistent. We demonstrate this inconsistency and derive a new, consistent, model. The performance of this model on a standard set of pre-segmented texts is demonstrated and compared with the Reynar model. The performance of the two models on a set of conversational dialogues is also shown. The use of automated audio transcription inevitably means that transcription errors will occur. We consider various models of transcription error and show how transcription errors affect the accuracy of the topical segmentation process for both Reynar's and our models.

1. Motivation

Dialogues have been successfully used as a tool to improve student learning. Experiments using Tutored Video Instruction (TVI) [1] demonstrated that students who collaboratively watch a previously recorded video of a lecture with a tutor and discuss its issues in between received higher grades compared with those students who attended the original lectur*e.

This beneficial effect has also been achieved with computer supported online discussions. Distributed Tutored Video Instruction (DTVI) [2] is a collaborative approach to group study in which students and a tutor collaboratively watch the previously recorded video of a lecture online. Controls enabled all participants to pause the synchronized video and discuss its content amongst the group. In an controlled experiment the students who participated in the discussions achieved better results compared with students who didn't.

Tertiary Courseware (TC) makes the dialogues that occur as a by-product of forming an understanding of a topic available as a learning resource in Higher Education [3] [4]. Various forms of such dialogues have been successfully used in computer-based teaching.

Using Task Directed Discussions (TDDs) thirty hours of tutorial discussions were recorded on video, and 200 clips extracted to form multimedia Tertiary Courseware material [5][6]. In experiments, 36 students were divided into two groups. One group had access to primary courseware only, the other to primary courseware with links to the tertiary courseware embedded. The group with access to the "vicarious" dialogues engaged in more discussion, had more relevant dialogues, and spent less time off-topic that the group with access to only the primary material.

In another experiment captured discussions were used to support students' understanding of parsing a tree structure. Transcripts of the dialogue between a tutor and a student as they attempted a series of discourse representation structure (DRS) exercises, together with an animated movie of a syntax tree or diagram, were made available to students. The purpose of the dialogues is to provide students with an opportunity to learn how to do these exercises, by observing the activity of other students. [6]

There is a more general case for making discussions accessible. Conversations with clients can be useful training material. [7] Discussions that take place during project meetings may contain a record of decisions as well as their justifications. In every day life people listen to other people's conversations as witnessed by programmes such as Question Time and Any Questions.

Most computers now provide programs for synchronous real time conversations in the form of text, such as Google Talk or MSN messenger, audio, such as Skype or video, such as NetMeeting. At the same time storage capacity is such that it is affordable to record the whole life span of a human being. [8]

The previous examples of Tertiary Courseware used materials that were manually segmented and structured. The effort involved in such manual pre-processing prevents this kind of courseware being employed on any larger scale. It is important that the dialogues be made accessible automatically to an acceptable standard.

A recorded conversation is linear in nature. An often used linguistic structure for discourse is as a sequence of utterances or turns. Separately, the participants' foci of attention during a conversation can be modelled as a set of focus spaces. [9] Several focus spaces can be accessible at any one time in a conversation. The structure of focus spaces can be modelled as a stack. The concepts from lower spaces are often available from higher spaces. Each sequential segment belongs to a focus space.

For a conversation to be used effectively as a learning resource requires focus spaces, which are individual sub topics, to be accessible without having to listen through a whole conversation up to the point of interest. Topical segmentation aims to separate a document, such as a conversation, into topically cohesive

[*] Address for correspondence

units. Once individual focus points are identified they can be made accessible through standard information retrieval techniques such as indexing and searching.

Many approaches exist to the segmentation of video material, which exploit scene changes, subtitles, lexical cues and their correlation just to name a few. [10] [11] However, in the context of tutorial dialogues the information that determines the boundaries of a topical segment are contained in the audio track or more precisely in the words that make up the conversation. Any video track if available only contains the torso of the participants.

This paper investigates the application of statistical topic segmentation algorithms for the purpose of segmenting conversations. Even though the algorithm is applied to perfect transcripts of conversations at the moment any chosen method must be sufficiently robust to be applicable to automatically transcribed conversations. Speaker independent automatic transcription of continuous speech will add a significant amount of noise to the transcript. It is not feasible to train the speech recognition engine on all participants in tutorial dialogue. The recognition rates of speaker independent speech recognition systems vary but even the best estimates include an error rate of 5%. [12] Other reports document word error rates of approximately 60%. [13] Additionally, even using a large vocabulary not all terms will be recognized, especially domain specific words are not likely to be part of the vocabulary. Spontaneous continuous speech adds another source of error though the use of utterances and other incompletely formed sentences.

We focus here on statistical topic segmentation algorithms look at a text as a “bag of words” and attempt to establish topical boundaries through the application of statistical methods.

2. Topic Segmentation Algorithms

There are various techniques for automatic topic segmentation. Some are briefly discussed here before analysing one algorithm in more detail. A more detailed review of topic segmentation algorithms can be found here. [14]

Vocabulary Management Profile [15] uses a graph of the average number of newly introduced words in successive intervals of a document to determine, amongst other things, topical boundaries. It is argued that a relatively large number of newly introduced words indicates topical change because new topics often introduce new words.

The Lexical Cohesion Profile (LCP) method [16] computes the cohesion or similarity of words in a given segment of text. Lexical cohesiveness is semantic similarity in words. [17] The similarity is calculated using a semantic network which is build from an English dictionary. If the words in a segment of text have a high cohesion value they are similar and the words are therefore considered to be part of one topical unit. If the words in a segment have a low cohesion they cross a topical boundary. Minima in cohesion denote topical boundaries. However, the method does not identify all topical boundaries in a text. Lexical cohesion is hard to determine. Apart from words having more than one meaning, changes in meaning due to sequencing of words are hard to measure. The word “bin” is different in meaning from “liner”, but the sequence of both, i.e. “bin liner”, has a different meaning again. The capture of the correct context of words and context shift is subject of much ongoing research.

Text Tiling [18] is a method that uses changes of lexical repetition for the separation of text into subtopics. Units or “tiles” of text are compared using a similarity score that is based either on the number of common non-stop words in adjacent sections or on the number of newly introduced words. Breaks in the lexical repetition indicate topic boundaries. This algorithm is of particular interest because it only uses term repetition to identify subtopics. It doesn’t require an additional thesaurus or other knowledge base. The evaluation indicates that the algorithm is quite robust. It was implemented to identify topical change amongst Japanese newswire documents with reasonable success.

Local Content Analysis (LCA) is used to expand the vocabulary expressed in a sentence in order to establish broader base for any similarity measure between sentences. [19] The similarity between all pairs of sentences is established and a score of similarity is calculated by adding up the similarity score for all possible individual segments. Each potential segment is ranked according using the score. The eventual decision on the segment boundaries is based on a final score that takes into consideration the final score and the probability of the segment length. This method is particularly useful when the length of segments is variable and can be very short so that there too few words in common to establish topic boundaries based on word occurrences alone. The method requires that sentence boundaries are known.

Many topic segmentation algorithms exploit certain cue words and phrases which indicate the beginning or end of a new story. This is particularly applicable to the segmentation of broadcast news stories where phrases such as “Joining us now from ….” or “Now over to ….” are frequently used to start a new story and “Good bye from …” denotes the end of a broadcast. However, these clue words are very domain specific. This report focuses on a technique that can be easily applied and is domain independent.

3. Methodology

The algorithm further investigated here describes a segmentation technique for the segmentation of text documents into sections that deal with disjoint topics. [20] For a series of hypothetical boundaries in a document, such as sentence boundaries or longer pauses during speech or any arbitrary points of choice, it is decided whether this point in the document is a topic boundary or not.

In order to evaluate a hypothetical topic boundary the text is divided into two sections; the section of text preceding a hypothetical boundary, called region one, and the section of text following the boundary, called region two. A topic boundary occurs if the probability that region one and region two do not discuss the same topic is higher than the probability that region one and region two discuss the same topic or in other words if the unconditional probability of seeing the words in region 2 is higher than the conditional probability. The length of one region is chosen as the average size of one topic from a training corpus of similar documents. Based on the published algorithm this is a length of 230 words.

The probability that the words in region 2 are used independently of the words in region 1 is calculated as the product of the probabilities of a word w occurring k times or:

$$P_{two} = \prod_{w} \mathrm{P}(k, w)$$

The probability that region 1 and region 2 discuss the same topic is calculated as the probability of seeing the words in section two given the context C, i.e. the words in region one or:

$$P_{one} = \prod_{w} \mathrm{P}(k, w \mid C)$$

The formula that is used to determine whether it is likely that the words in region two were generated independently from the words in region one is based on the G model. [21] The (unconditional) probability of a word w occurring k times in a document is calculated as follows:

$$P(k,w) = (1-\alpha_w)\delta_{k,0} + \alpha_w(1-\gamma_w)\delta_{k,1} + (\frac{\alpha_w\gamma_w}{B_w - 1}(1 - \frac{1}{B_w - 1})^{k-2})1 - \delta_{k,0} - \delta_{k,1})$$

With:

- α_w being the probability that a document contains at least one occurrence of word w
- γ_w being the probability that w is used topically (more than once) in a document given that it occurs at all.
- B_w being the average number of occurrences in documents with more than one occurrence of w.
- $\delta_{x,y}$ being a function with the value of 1 if x = y and 0 otherwise.

The formula expresses the probability of a word w occurring k times in a document as the sum of the probability that the word occurs 0 times, the probability that the word occurs once, i.e. the word is a non-topical word, and the probability that a word occurs more than once, i.e. that the word is a topical word.

The three parameters α_w, γ_w and B_w were originally established for each word from a training corpus of similar documents. However, the algorithm still performed well, even for segmenting Spanish broadcast news, when a constant value for α, γ and B was used, which is the value assigned to unknown words based on training data from English newswire documents.

Table 1 shows the formulae used for calculating the conditional probabilities given certain key numbers of occurrences of a word in region 1 and region 2. The formulae consider 0, 1, 1 or more (1+) and 2 and more (2+) occurrences of a word.

Occurrences in region 1	**Occurrences in region 2**	**Conditional probability**
0	0	$1-\alpha$
0	1	$\alpha(1-\gamma)$
0	2+	$\frac{\alpha * \gamma}{B-1}(1-\frac{1}{B-1})^{k-2}$
1	0	$1-\gamma$
1	1+	$\frac{\gamma}{B-1}(1-\frac{1}{B-1})^{k-2}$
2+	0+	$\frac{1}{M(B-1)}(1-\frac{1}{B-1})^{k-2}$

Table 1: Formulae for the calculation of conditional probabilities [20]

The algorithm was initially trained and tested by Reynar on the HUB-4 Broadcast News Corpus, a collection of transcribed television news broadcasts, annotated with topic boundaries by the producer. The performance of the segmentation algorithm was compared using precision and recall against:

- An algorithm that randomly computes the same number of segment boundaries as were contained in the original documents.
- The TextTiling algorithm (Hearst, 1994).
- An improved version of the TextTiling algorithm.
- A maximum entropy model, which also exploits domain cues, common content word bi-grams, common named entities, synonyms, first use of words after the boundary and the use of pronouns in the first five words after the topic boundary.

The word frequency algorithm performed better than all apart from the maximum entropy model. It achieved a precision of 0.55 and a recall of 0.52. Precision is the proportion of correctly identified boundaries out of all identified boundaries. Recall is the proportion of correctly identified boundaries out of all correct boundaries.

Since any given word can either: not occur at all, occur once or occur more than once the sum of all those probabilities should add up to 1. In other words it is certain that any given word occurs not at all, once or more than once in a section.

The formulae for the calculation of the conditional probabilities is based on the assumption that "having seen 0 occurrences in block 1 does not provide any information about how many are likely to be observed in block 2, so the conditional probability is simply the probability under the original model" [14, pg. 97]. The G model assumes the probability of seeing k instances of word w in a document as:

$$P_w(0) = (1 - \alpha). \qquad (1)$$

The conditional probability for seeing 0 occurrences of word w in region 2 of the document is calculated as: (1- α), see Table 1.

Bayes' theorem states that the probability of event A given event B is equal to the probability of an event being A and B over the probability of event B or :

$$P(A|B) = \frac{P(A \cap B)}{P(B)}. \qquad (2)$$

Applied to the calculation of conditional probabilities for the occurrences of words in region 2 of the document the probability of seeing 0 occurrences of a word in region 2 given 0 occurrences of this word in region 1 is equal to there being 0 occurrences of this word in the whole document over the probability of seeing 0 occurrences in region 1 or:

$$P(0_2|0_1) = \frac{P(0_2 \cap 0_1)}{P(0_1)}. \qquad (3)$$

The probability of seeing 0 occurrences of word w in the document was earlier described as (1 - α).

Substituting (1) in (2) leads to:

$$P(0_2|0_1) = \frac{(1-\alpha)}{P(0_1)}. \qquad (4)$$

The same formula is used to calculate the probability of seeing a word in region 2; therefore:

$$\frac{(1-\alpha)}{P(0_1)} = (1 - \alpha). \qquad (5)$$

The only condition under which this is possible is: $P(0_1) = 1$, in other words when it is certain that there are no occurrences of a word in region one. This leads to a contradiction since the same argument can be used to calculate the conditional probabilities for any word. Therefore, there can be no words in the first half of the document which unrealistic. This is false, therefore the original assumption of $P_w(0) = (1 - \alpha)$, must be false.

The G model does not make any assumption about the distribution of words inside a document. It only provides an assumption of the probability of a word occurring various times. Choosing (1 - α) as the estimate for the conditional probability of a word occurring in region 2 given that it doesn't occur in region 1 assumes that this word can't occur in the first half of the document. Logically there are situations for which this can be imagined, such as ending a document with the word "finis". However, this seems to be more likely an exception rather than a rule.

4. The Computation

The unconditional probabilities for k occurrences of word w, $p_w(k)$ are given by the Katz model [21]:

$$p_w(0) = 1 - \alpha \qquad (1)$$

$$p_w(1) = \alpha(1-\gamma) \qquad (6)$$

$$p_w(k), k \geq 2 = \frac{\alpha\gamma}{B-1}(1 - \frac{1}{B-1})^{k-2} \qquad (7)$$

Where α is the probability that a word occurs in a document at least once, γ is the probability that the word occurs topically, i.e. more than once if it occura at all, and B is the average number of occurrences of a word when it occurs more than once.

In order to estimate the conditional probability of k_2 occurrences of a word w in the second half of a document given k_1 occurrences in the first half we assume a uniform distribution for words within a given document. This means that any particular instance of w has a probability of 1/2 of being in either the first or second half of the document.

The probability of $j, j \geq 0$ occurrences of w in the first half of the document and $k-j, k \geq j$ occurrences in the second half can be computed as:

$$p_w(j, k-j) = \binom{k}{j} p_w(k)/2^k \quad (8)$$

where $\binom{k}{j}$ is the number of ways of selecting j items from k, and equals $k!/(j!(k-j)!)$.

The conditional probability of k_2 occurrences of a word in the second half of a document given k_1 occurrences in the first half is:

$$p_w(k_2 \mid k_1) = p_w(k_1, k_2) / \sum_{i=0}^{\infty} p_w(k_1, i) \qquad (9)$$

The probability of k_1 occurrences in the first half of the document is $\sum_{i=0}^{\infty} p_w(k_1, i)$. The value of the expression $\sum_{i=0}^{\infty} p_w(k, i)$ is given as follows:

$$\begin{aligned}
\sum_{i=0}^{\infty} p_w(0,i) &= \frac{1}{2}(1-\gamma)\alpha + \frac{\gamma\alpha}{2B} - \alpha + 1 \\
\sum_{i=0}^{\infty} p_w(1,i) &= \frac{1}{2}\alpha(1-\gamma) + \frac{\alpha(3B^2 - 8B + 4)\gamma}{2(B-2)B^2} \qquad (10)(11)(12) \\
\sum_{i=0}^{\infty} p_w(k,i)(k>1) &= \frac{2\alpha\left(1 - \frac{1}{B-1}\right)^{i}(B-1)^2\left(\frac{B}{B-1}\right)^{-i}\gamma}{(B-2)^2 B}
\end{aligned}$$

5. Experiments

The algorithm with the was evalutated on manually transcribed conversations. Three conversations, of approximately 30 minutes duration were transcribed. In addition, as a control, the sonnets of Shakespeare were entered with segment boundaries at the end/beginning of each sonnet, and the sonnets as a whole treated as a single document with topic changes at sonnet boundaries.

The Lucene package [22] was used for the initial preparation of the documents. The following processing steps were performed:

1. The documents were tokenized.
2. Common stop words were removed.
3. Porter stemming [23] was applied to each word to produce a token.

The transcribed conversations contained 14, 16, and 19 segments ranging in length from 20 to 300 tokens. The sonnets contained 154 segments, with an average length of 55 tokens.

In order to present the algorithms in the most favorable light token frequency statistics were gathered for each document and used as the parameters of the Katz model. [21] The probabilities of occurrence were computed using Good-Turing smoothing [24] of the occurrence frequencies in each document. This means that we will estimate the probability of 3 occurrences of a token as greater than 0 even if we did not observe 3 occurrences in any of the document segments.

In line with the work of Reynar a fixed window was used as an estimate of the size of document segments. For the conversations the window size ω was 100 tokens, for the sonnets 55 tokens.

Given a set of ordered tokens $S = t_0, \ldots, t_n$ which constitute a sub-document, the tokens are analyzed and we produce a set of token-occurrence pairs, $\{(t_0, n_0), \ldots, (t_i, n_i), \ldots (t_m, n_m)\}$ where n_i is the number of times t_i occurs in S. The probability of S, using the Katz model is $\prod_i p_{t_i}(n_i)$. In order to avoid arithmetic overflow with very small numbers we use surprisals, representing the *weight* of S, $w(S)$ as $\sum_i -\log p_{t_i}(n_i)$.

In order to estimate topic boundaries we proceed as follows:

1. Split the token stream at ω. Let the left ω tokens be L and the right ω tokens be R. We compute the weights of L and R as well as the conditional weight of R given L, as described above.
2. The split is advanced one token. Weights and conditional probabilities for L and R are recomputed. This computation can be performed efficiently using a sliding window.
3. The process stops when R comes within ω tokens of the end of the document.

The output of this process is an estimate of the weights of L and R, together with the conditional weight of R as the second half of a topical section given L as the first half, at each possible split between ω tokens and $N - \omega$ tokens, where N is the number of tokens in the document and ω the sliding window size.

6. Conclusions

A typical plot of weights and conditional weights is shown in Figure 1.

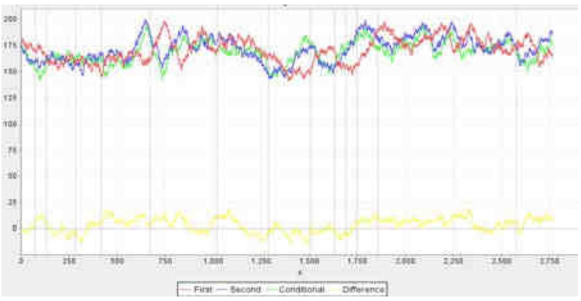


Figure 1: Plot showing unconditional and conditional probabilities for an discussion from IT Conversations

The weight of L is shown in red, the weight of R in blue, the conditional weight in green, and the difference between the weight of R and the conditional weight in yellow. The continuous vertical lines show topic boundaries.

For each of the documents, using both Reynar's original algorithm and our corrected computation of conditional probabilities we found very little correlation between the difference of R and the conditional weight. There is no discernible correlation between the weights and the segment boundaries.

For each of the documents, using both Reynar's original algorithm and our corrected computation of conditional probabilities we found very little correlation between the difference of R and the conditional weight. There is no discernible correlation between the weights and the segment boundaries.

There are several reasons for the failure of the "bag of words" model to provide satisfactory topical segmentation.

1. Reynar's experiments used news stories as the topical units. It islikely that there are greater and more concentrated differences invocabulary between news stories than there are between segments of a conversation.
2. The underlying topical model which partitions a conversation into disjoint segments does not accurately represent the structure of real conversations. This is also true inthe domain of poetry where themes can recur between poems, although in a different context.
3. There is an inevitable imprecision in determining topic boundaries using the "bag of words" model since the words are presented sequentially, and words which are significant to the weight of a segment are likely to occur away from the topic boundaries. It is quite possible that this is more likely to be true in conversations, as opposed to news stories, since journalists tend to present the main topic of a story at the beginning, and will summarise the topic at the end.

The corrected conditional probability model produced significantly different output, but made no difference to the accuracy of the results with respect to our segmentations. We believe that this shows that the basic premise that a "bag of words" approach to topical segmentation can work in this context is false for the reasons outlined above.

We have not analyzed the fit of the Katz G-model to our data. It is possible that the fit is poor and that a model, better suited to conversational dialogues, would improve algorithm performance. We believe this to be unlikely but it may deserve further testing.

7. References

[1] Gibbons, J. F., W. R. Kincheloe, et al. (1977). "Tutored Videotape Instruction: a new use of electronics media in education." Science 195: 1139-1146.

[2] Sipusic, M. J., R. L. Pannoni, et al. (1999). Virtual Collaborative Learning: A Comparison between Face-to-Face Tutored Video Instruction (TVI) and Distributed Tutored Video Instruction (DTVI). SMLI TR-99-72. Mountain View, Sun Microsystems Laboratories. http://research.sun.com/ics/dtvi.html.

[3] T. Mayes. Learning technology and groundhog day. In W. Strang, V.B. Simpson, and D. Slater, editors, *Proceedings of Hypermedia at Work: Practice and Theory in Higher Education*. University of Kent at Canterbury, 1995.

[4] McKendree, J., K. Stenning, et al. (1998). "Why observing a dialogue may benefit learning." Journal of Computer Assisted Learning(14): 110-119.

[5] T. Mayes and F. Dineen. Developing tertiary courseware through capturing task directed discussions. In *World Conference on Educational Multimedia, Hypermediaand Telecommunications (EDMEDIA 1999)*, pages 1061-1066, 1999.

[6] Stenning, K., McKendree, J., Lee, J., Cox, R., Dineen, F., and Mayes, T. 1999. Vicarious learning from educational dialogue. In *Proceedings of the 1999 Conference on Computer Support For Collaborative Learning* (Palo Alto, California, December 12 - 15, 1999). C. M. Hoadley and J. Roschelle, Eds. Computer Support for Collaborative Learning. International Society of the Learning Sciences, 43.

[7] http://www.itconversations.com/shows/detail1682.html

[8] Gemmell, J., Lueder, R., and Bell, G. 2003. The MyLifeBits lifetime store. In *Proceedings of the 2003 ACM SIGMM Workshop on Experiential Telepresence* (Berkeley, California). ETP '03. ACM Press, New York, NY, 80-83.

[9] Grosz, B. J., and Sidner, C. L. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*12(3):175.204.

[10] Chua, T., Chang, S., Chaisorn, L., and Hsu, W. 2004. Story boundary detection in large broadcast news video archives: techniques, experience and trends. In *Proceedings of the 12th Annual ACM international Conference on Multimedia* (New York, NY, USA, October 10 - 16, 2004). MULTIMEDIA '04. ACM Press, New York, NY, 656-659.

[11] Maybury, M. T. 1998. Discourse cues for broadcast news segmentation. In *Proceedings of the 17th international Conference on Computational Linguistics - Volume 2* (Montreal, Quebec, Canada, August 10 - 14, 1998). International Conference On Computational Linguistics. Association for Computational Linguistics, Morristown, NJ, 819-822.

[12] Huang, X., Alleva, F., Hwang, M., and Rosenfeld, R. 1993. An overview of the SPHINX-II speech recognition system. In *Proceedings of the Workshop on Human Language Technology* (Princeton, New Jersey, March 21 - 24, 1993). Human Language Technology Conference. Association for Computational Linguistics, Morristown, NJ, 81-86.

[13] Brown, E., Srinivasan, S., Coden, A., Ponceleon, D., Cooper, J., Amir, A., and Pieper, J. 2001. Towards speech as a knowledge resource. In *Proceedings of the Tenth international Conference on information and Knowledge Management*

(Atlanta, Georgia, USA, October 05 - 10, 2001). H. Paques, L. Liu, and D. Grossman, Eds. CIKM '01. ACM Press, New York, NY, 526-528.

[14] Reynar, J.C. (1998). Topic Segmentation: Algorithms and Applications. Ph.D. thesis, University of Pennsylvania, Department of Computer Science.

[15] Youmans, Gilbert. 1991. A new tool for discourse analysis: The vocabulary-management profile. Language, 67(4):763-789.

[16] Kozima, H. 1993. Text segmentation based on similarity between words. In *Proceedings of the 31st Annual Meeting on Association For Computational Linguistics* (Columbus, Ohio, June 22 - 26, 1993). Annual Meeting of the ACL. Association for Computational Linguistics, Morristown, NJ, 286-288.

[17] Kozima, H. and Furugori, T. 1993. Similarity between words computed by spreading activation on an English dictionary. In *Proceedings of the Sixth Conference on European Chapter of the Association For Computational Linguistics* (Utrecht, The Netherlands, April 21 - 23, 1993). European Chapter Meeting of the ACL. Association for Computational Linguistics, Morristown, NJ, 232-239.

[18] Hearst, M. A. 1997. TextTiling: segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.* 23, 1 (Mar. 1997), 33-64.

[19] Ponte, J. M. and Croft, W. B. 1997. Text Segmentation by Topic. In *Proceedings of the First European Conference on Research and Advanced Technology For Digital Libraries* (September 01 - 03, 1997). C. Peters and C. Thanos, Eds. Lecture Notes In Computer Science, vol. 1324. Springer-Verlag, London, 113-125.

[20] J. C. Reynar. Statistical models for topic segmentation. In *Proceedings of the37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 357-364, Morristown, NJ, USA, 1999. Association for Computational Linguistics.

[21] M. Katz S. Distribution of content words and phrases in text and language modelling. *Nat. Lang. Eng.*, 2(1):15{59, 1996.

[22] http://lucene.apache.org

[23] M.F. Porter, 1980, An algorithm for suffix stripping, *Program*, 14(3) pp 130−137.

[24] I. J. Good. The Population Frequencies of Species and the Estimation of Population Parameters. *Biometrika*, Vol. 40, No. 3/4 (Dec., 1953), pp. 237-264

Evolution of Interactive Audiobooks

Cornelius Huber, Niklas Röber, Knut Hartmann, and Maic Masuch

Computer Games Research Group
Department of Simulation and Graphics
Otto-von-Guericke-University Magdeburg, Germany

Abstract. Audiobooks and radio plays have enjoyed an increase in popularity over the recent years and are still growing. One of the reasons are their ease of use, but also a deepened immersion due to an auditory presentation of the story. Interactive Audiobooks combine the advantages of such an auditory narration with interactive elements known from computer games, therefore allowing a more intense perception by focussing on an auditory presentation.

This paper summarizes the ideas behind interactive audiobooks and sketches the various developments and improvements over the last year. Here we focus especially on the user interface design, used for story sonification and game interaction, as well as on the development of a mobile sound/story engine that allows an implementation on mobile devices.

1 Introduction

An acoustic narration of a story line has several advantages. It integrates the listener into the story by requiring an active participation in order to reconstruct the fictional universe. This reconstruction takes place using the listeners own imagination and phantasy, therefore adding and building it from personal information and experiences. Oral presentations are therefore considered to be much more stimulant and immersive than audio/visual presentations. Additionally, auditory narrations are easier to author and design and require less hardware for their presentation. Furthermore, auditory content is easier to perceive and evaluate than audio/visual depictions.

The recently introduced *Interactive Audiobooks* take the approach of auditory narration one step further by combining the advantages of complex (non-)linear narratives with interactive elements from computer games [1]. These interactive elements represent parts of the story line and actively integrate the listener/player as part of the story. The underlying engine is thereby designed with a varying degree of interactivity, that allows anything between a passive listening to the story up to an interactive audio-only computer game.

2 Interactive Audiobooks

The authoring and the design of immersive, non-linear plots remains one of the main challenges in interactive digital storytelling. The main advantages of interactive audiobooks are a non-linear story line, but also a varying degree of interaction. Both become feasible with the utilization of story trees, which offer an interesting alternative to create a non-linear story design [2]. This method is also often used by game designers to provide alternative plots and endings that are consistent with the player's action and interaction. Our system is based on such a story-graph structure and further extended by interaction nodes. These interactive nodes represent essential parts of the story line that can be actively played by the user. The interaction nodes comprise story-related mini-games and dialogs, but also techniques to interfere with the story-line and to influence the story's characters. As they are based on narrative nodes, they also contain narrative content, in case the listener does not wish to interact. Here the system chooses among possible directions and the story continues depending on previously made decisions. Figure 1 shows a simplified story-graph, in which *narrative nodes* (light/green) and interactive nodes (dark/blue) intertwine. The story starts at the top with the root node and traverses down till it reaches an end condition (*terminal nodes*). The graph branches at predefined points, at which decisions and challenges in the form of interactive parts are placed. The perceived story line depends on the players interaction, but also on the character's behavior, and can be re-played multiple times to explore alternative pathways and endings.

As all information is presented acoustically, sophisticated techniques for information sonification, as well as interaction are required. Here we employ several techniques which are commonly used in 3D virtual auditory environments [3], but have especially concentrated on an intuitive interaction paradigm that can be implemented using a minimal interface. The first prototype was built around a regular gamepad that was used for interaction and control [1]. A preliminary user evaluation, however, revealed several difficulties and problems, of which the interface design and uncertainties whether or not interaction was possible were among the severest. Starting from there, several different approaches were evaluated, leading to a modified design of the interface/interaction paradigm. The main character/storyline is now affected and indirectly controlled based on four types of interaction: *thinking*, *aggressive* and *defensive* behavior, and *passive* waiting. In the current implementation, this approach has proven to be successful, although an elaborate user evaluation needs to be performed.

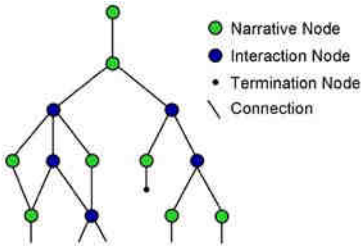


Figure 1: Story Tree Structure.

2.1 System and Design

The developed system is divided into two parts and consists of an authoring environment (Figure 2) and a runtime system. At the moment the runtime system is also located on the PC-platform, but currently being ported for mobile implementation. The initial implementation was based on OpenAL/EFX and employed only partially 3D sounds and EFX effects for the sound rendering. Using mobile devices and a new sound engine, we hope to utilize and exploit these functionalities stronger and integrate them into the scene presentation and even into some of the mini games. For the mobile implementation, we target cell phones, PDAs and mobile gaming consoles. Recently, we have developed a fast and sophisticated DSP sound engine that supports several streams and DSP filtering in realtime, and which runs on nearly all mobile platforms. The next task will be a conversion of the runtime system including the story engine and the current user interface setup, which has been laid out with mobile devices in mind.

Figure 2 shows an overview of the authoring environment that is currently used to create and design interactive audiobooks. The screen is mainly divided into three parts, of which the left one shows the entire story, while the middle screen displays the current selected scene along all contained narrative and interaction nodes. The right part of the authoring environment is used to specify interactions, design mini-games and to assign sounds and ambient music to the nodes.

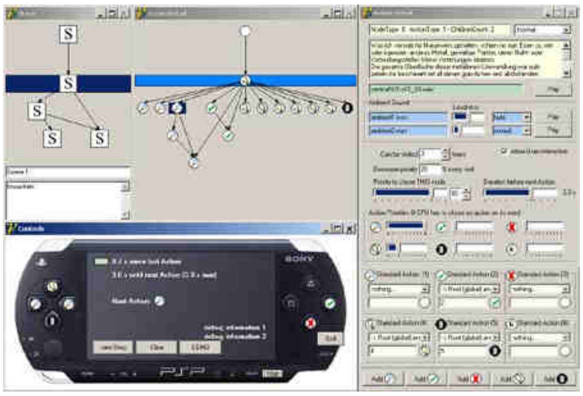


Figure 2: Authoring Environment.

So far several smaller interactive audiobooks have been created, with one being an adaptation of a short story from Edgar Allan Poe and another a story that combines several myth of the Magdeburg Cathedral [3]. In the latter attempt, the story has been designed more tightly around the interactive audiobook's concept, to ensure a higher acceptance among the users.

3 Summary and Conclusions

We have discussed the concept of interactive audiobooks along several improvements and modifications that have evolved over the last year. Here we focussed especially on the user interface design and an implementation for mobile devices.

Although, the new interface has proven to work very well, several ideas remain for future improvements. The current sound rendering only allows sound spatialization and room acoustics to a minimal degree. Also topics like multi-player and augmented audio reality are yet to be explored and will hopefully soon extend the possibilities of interactive audiobooks.

References

[1] N. Röber, C. Huber, K. Hartmann, M. Feustel, and M. Masuch. Interactive Audiobooks: Combining Narratives with Game Elements. In *Proceedings of TIDSE Conference*, pages 358–369, 2006.

[2] K. Hartmann, S. Hartmann, and M. Feustel. Motif Definition and Classification to Structure Non-Linear Plots and to Control the Narrative Flow in Interactive Dramas. In *3rd Int. Conf. on Virtual Storytelling*, pages 158–167, 2005.

[3] N. Röber and M. Masuch. Leaving the Screen: New Perspectives in Audio-only Gaming. In *Proceedings of ICAD*, 2005.

Game Sound Education at the Interaction Design Department of the Zurich University of the Arts - Between Research Laboratory and Experimental Education

Daniel Hug
Interaction Design Department
Zurich University of the Arts
daniel.hug@zhdk.ch

Abstract. Game design education is still a rather rare subject at educational institutions, even less common is the topic of game sound design. At the Interaction Design department of the Zurich University of the Arts (ZHdK), Switzerland, we have introduced several interconnected courses that teach aspects related to game sound design, taking into account the special requirements at a school for art and design. In order to deal with the lack of best practices and examples that could help us orientate and define such a program we develop the courses along research questions, investigating at the same time the theoretical foundations by using the courses in game sound design as experimental laboratories. This iterative cycle of teaching and analysis helps us to advance both education and research.

1 Introduction

Since the arrival of the AdLib Music Synthesizer Card (1987) and the Creative Labs Sound Blaster 1.0 (1989) game audio has played an increasingly important role in the market. Therefore, when the game design curriculum was introduced at the Interaction Design department (IAD) of the Zurich University of the Arts (ZHdK)[1] in 2003 it was soon clear that also sound design should play an important role in the curriculum. When game sound was finally introduced in 2005 the question was: What should such a program contain, what are it's elements and how should they be structured?

In this paper I describe our approach of integrating game sound education and research in a loop that facilitates mutual fertilization and makes it possible to advance both aspects. In the first section I will outline the current situation of game audio education. Then I will introduce the main research questions that are relevant to students of game audio design at an university of the arts. In the last section I will describe the curriculum at ZHdK and show the mutual links between education and research[2].

2 The State of the Art in Game Audio Education and Research

The most obvious first step to find out how to develop a curriculum was to look at existing educational opportunities. In fact, courses in entertainment technology or game design are offered at many universities (see, for example, [31]). However, while there are countless courses on graphic design, 3D modeling, or even programming available, sound related courses, let alone game sound related ones, are still relatively rare. In the private sector things do not look much different, even at well-known institutions such as the Games Academy [1], finding something about sound design in the curricula is difficult. The SAE institute [30], probably the most renowned international school for audio engineering, offers game design related degrees in collaboration with the Qantm college [12], but apart from digital audio, editing and mixing courses also here nothing can be found that is specifically related to game sound design. The same is true for the International Game Developer Association's Special Interest Group on Game Education [2].

A potentially helpful source, the Game Audio Education Working Group of the Interactive Audio Special Interest Group (IASIG), describes it's mission and aims as follows:

> ”Game audio requires a myriad of specialized audio skills. Currently few web sites and books provide resources to educate new people who want to get involved in game audio. This project sets out to offer an additional resource for people to use in learning about game audio. Its overall goals include:
>
> - Shortening the learning curve of new audio personnel in game audio by providing resources that people can learn from.
> - Creating a clear reference for terminology and creation methods in the world of game audio.
> - Supplying schools with information to integrate game audio education into their curricula.”[22]

In fact the IASIG Wiki [23] offers an extensive list of relevant aspects to be covered in game audio education. Unfortunately, as the working group has been closed, there is not too much activity on the Wiki and many questions, especially around designing for interactive media and games, remain unanswered.

2.1 Game Audio for Art and Design

There is no doubt, getting to know and master the tools of the trade has to be covered in a curriculum on game audio design. However, especially at an university of the arts, essential skills for students to acquire are to be able to develop original aesthetic designs and the ability to express oneself artistically. While this requires a lot of experience that cannot be taught directly, there are some areas of knowledge that form an essential basis to acquiring those competences: Understanding the importance of communication, semiotics and aesthetics while providing the functionality required for usability and game-play.

In order to find more information about what should be part of a game audio curriculum satisfying these requirements we also looked at publications. There are some excellent books on game audio, however they focus on the practical aspects of design and

[1] http://iad.zhdk.ch

[2] Please note that the emphasis of this work lies on a general notion of sound and not specifically on music or voice, although it may include it as well.

implementation and do not really deal with semiotics or functional aesthetics. Some works focus more on sound design and business [29], others on aspects of programming [38], [6] or the development process as a whole [7]. Theoretical frameworks for film sound design, such as those developed by Chion [9] or Flückiger [17], may help to develop theories for game sound, but they usually are not sufficient to understand game sound phenomena completely.

Since a few years this situation is slowly changing, though, as more and more online and offline sources explicitly related to this topic appear. Interesting online resources are for example the game sound design section of filmsound.org [8], the audio track on Gamasutra [20] or the website of the Game Audio Network Guild [24]. Collins runs a website that provides a rich source on game audio, among others a glossary and a description of functions of game audio [13]. There is also an increasing number of scientifical offline publications which are of interest. Ekman has published an interesting paper introducing a theory of diegesis adapted for computer games [16]. Friberg and Gärdenfors developed approaches to developing a categorization for game sound and include in their work aspects of works from Chion [19]. Moreover there is an upcoming article by Collins, which specifically deals with participatory and non-linear aspects of video games audio [14]. But many of these sources seem to focus on music rather than on sound in general. There is obviously a need for more systematic research on semiotics and functional aesthetics of game sound. My goal is, to contribute to this research on game audio by integrating it into game education. For this, first the core questions of research have to be clarified.

3 Topics of Game Sound Research at ZHdK

As mentioned, semiotics, functional aesthetics and individual expression are the main topics for art and design related game sound education. In this section I want to describe more in detail what I mean by this.

The first requirement, which separates games from other interactive applications is *immersion* and support of *flow* [15]. And here sound plays an important role, because it is omnidirectional, multidimensional and encompassing. Moreover the ears cannot be closed [37, 27]. However, achieving immersion requires more than good headphones or a surround sound setup; Shedroff, for example, reminds us that a good story, well told, can often immerse us deeper than a technically sophisticated artificial audiovisual environment [34]. Some important requirements for immersiveness are adaptivity, completeness and continuity in the sound design [25]. But many questions as to what facilitates immersive experiences remains unanswered. This is why I consider immersion being an issue of design and art rather than technology.

We all know the role of sound in giving abstract or fictional objects a ”physical” existence, helping to suspend the disbelief in the artificial world. But the functional-aesthetic role of sound goes beyond that: It actually reveals to us the ”inner nature” of things and processes, it *”characterizes”* them, a property which is used extensively in computer games. Therefore, the objects we interact with in a game are not just ”there”, they also reveal their character and meaning in the context of a specific situation in the game. They acquire meaning beyond their simple ”existence”.

This leads us to the next aspect of game sound design that I consider relevant for our courses: The relationship between *sound and action*. The meaning revealed though sound I just wrote about is revealed through interaction with the game world. And this is one of the essential differences to film sound: We do not passively watch and listen to a virtual world, we experience it through action.

An example of how sound modifies the meaning and perception of action is how sound connects processes. In film, sound plays an important role in providing continuity between cuts, it can make processes seem continuous even if they are montaged. In games there are no cuts in the same way as in movies. Instead, we may encounter situations like the following: In ”XIII” (Ubisoft 2003) a first aid shelf is opened by clicking on it with the mouse and when we approach it further we take out a first aid box. But what actually happens upon the mouse click is this: First there is the door opening itself with a ”squeek”, our avatar still holds the gun in his hand and doesn't move. Second, the first aid box just disappears, however, we hear a metallic sound of a box put in place somewhere. What we have here are two things: First, the substitution of missing graphical information through sound, and second, the sonic completion of abstract user actions, like mouse button presses, into complex virtual actions.

The action related functions of sound have again an importance on an archaic, psychological level: They satisfy our desire for the manifestation of our action and influence in the world through sound, a concept that Chion calls *”ergo audition”* [11]. I consider this a very important factor in creating overall enjoyable game experiences. Of special interest is a related effect that I call *”differential of power”*: If I as a player trigger a loud, powerful sound with a small movement, it gives me the feeling of power and influence, a weak and fragile sound however is perceived as a sign of weakness and powerlessness [26].

Last but not least, there is the concept of *”systemic composition”* of both the virtual sonic objects and the virtual soundscape in a game, which introduces a musical perspective into game sound design. This means that we can study the sounds of a game as a kind of ensemble composed of a multitude of sound emitters, which are differently configured and placed, and the player's movement through the environment and her interaction with objects in it generates a dynamic sonic arrangement and mix. This means one has to design a functional-aesthetic system of sounds in time and space rather than what designing a mix means in linear media like film.

To summarize: Meaning in games is constructed in the triad of (inter)action, vision and audition[3] in a nonlinear, modular system of objects and events, with a strong mutual influence between those elements, very much comparable to the audiovisual contract that has been described for film [9]. Therefore, systemic sound design, the relationship with action and the meaning creation processes are the core questions for both research and education at the game audio program of the Interaction Design department at ZHdK.

4 Using Education for Research

I have laid out the main areas of interest in game sound research and education at IAD/ZHdK. But how to go about actually researching and teaching these topics? How can research and education be linked in a fruitful way? And: Why should they actually be linked?

As mentioned before, we aim at connecting education and research in a tight loop because there is not enough substantial and structured knowledge to form the curriculum that is supposed to treat questions of semiotics and functional aesthetics. Therefore

[3] Also haptics and possibly other sensorial channels could be included, but sound and vision are certainly the most relevant senses in games today.

we develop the theoretical foundations by using the courses as experimental laboratories, using an iterative cycle of teaching and research. A big advantage of this approach is also the momentum that can be achieved by mobilizing students. In fact, the students often introduce surprising and fresh approaches to analytical and theoretical questions that can help to reorient the research. From this point of view it is actually an advantage that there is not an established "must have" curriculum for game sound design yet.

The courses are project oriented, often several courses like sound design, 3D modeling and programming are joined in a common project. Like this many valuable and well documented cases for further study can be generated in a relatively short time[4].

In parallel to the courses a *Toolbox for Game Sound Analysis* has been developed and is modified in an iterative way, using the insights gained from practical course and project work. This "living document" serves as a central point of reference and documentation of the actual situation in the process of developing the theory for game sound. Providing the toolbox for the students helps to overcome the fundamental difficulty of understanding and speaking about sounds and sonic experiences in our visually dominated society.

In the following section we will present the elements of this curriculum, how they feed into each other and into research, and how they help to refine the theoretical and didactical foundations on which game sound education can build.

5 Course Descriptions and Cases

5.1 The Basis for Everything: Acoustic Display

In this first course module the foundations for all subsequent courses are laid. The four-day course for students in the second semester is carried out together with the students of the Interaction Design program, as the content is relevant to both game and interaction designers.

5.1.1 Course Description

The topics are the following:

Phenomenology of sound and hearing: First of all, even before acoustics, comes the question of how we experience sound in our everyday "Lebenswelt" (to borrow a term coined by Husserl). Investigating the different modes of hearing and their influence on how sounds are perceived and valued is another important topic that is often underestimated or forgotten.

Speaking about sound: This is the first element in a kind of "conceptual sound toolbox". It focuses on basics of sound creation and propagation on a very high level, introducing terms like pitch, volume, timbre, envelope, and the influence of space and material on sound. Building on these elements the concept of everyday listening and sound events by Gaver [21] is introduced. This is combined with the concepts of "objet sonore" developed by Schaeffer [32] (see also [10]) and "sonic effect" [3].

Notation of sounds and sound events: An approach to notation of everyday sounds is introduced. This is derived from Schafer's work on soundscape analysis [33] which is a simplification of Schaeffer's taxonomy.

Soundscape analysis: Starting from Schafer's work the notion of the soundscape as a sonic system, composed by interacting sounding objects, is discussed. In order to approach sound also from a communication oriented point of view the aspects of acoustic communication and acoustic community are introduced [33, 35].

Basic action-sound relationship: This topic is about understanding the action-sound relationships in the physical world, how our perception and interpretation of objects and processes are influenced by sound and how these fundamental experiences shape our understanding of the world.

First and second order semantics: The next step is to look at the many ways that meaning is created through sound, starting with film analysis. Important sources of knowledge are here [36] and [17]. This framework also includes aspects of material, setting, scenography and subjectivization.

Sound in the context of interaction: Finally, the fields of acoustic display and sonification are introduced, ranging from basic functions of acoustic display like alarms, status information etc. to the concept of sonification, acoustic icons and earcons.

All of these topics are deepened and discussed in class assignments such as protocoled sound walks or action - sound analysis for everyday objects. The course's final project is the design of an acoustic display for an interactive application. The outcome can be something like functional prototypes for drag & drop sonifications or sonically enhanced objects of everyday use. An example of this course module is documented in [18].

5.1.2 Relevance to Research Questions

Several links between education and research can be identified here:

Speaking about sound is connected to the important question of how everyday sounds can be described and categorized. All areas of research and education can build on such systems of description. The exercises allow to test the vocabulary and notation strategies under practical conditions (e.g. when recording sounds and communicating about them with the team for design decision making) and help to improve them iteratively.

Semiotics and aesthetics: As mentioned before, understanding film sound is a good starting point due to the extensive theoretical material available. This way specific aspects of game sound design can be analyzed in comparative studies, adapting the methods from film sound study.

The investigation of *action-sound relationships* and the *soundscape as a sonic system* is fundamental to understand the nature of sound design for interactive media.

5.2 Game Sound Analysis

This course (about 3 days in total, for students of the second semester) is the first in the program that addresses game sound specifically. It builds upon the foundations mentioned above.

5.2.1 Course Description

After a warm-up listening exercise a theoretical and methodological introduction is given, providing a *Toolbox for Game Sound Analysis*. This toolbox contains, among other elements:

- *General description of the sounds*: what sounds can be heard, what categories (language, music, SFX; subcategories like NPCs, buttons, etc.; categories of everyday listening or sonic objects, (see 5.1.1)) can be identified, what moves and where, description of spatial properties and description of the "sound objects" encountered, etc.;
- *Semantic categories*, such as archetypes, symbols, keysounds, stereotypes, leitmotifs, etc.;
- *Functional aspects*:
  - *Related to interaction*: general communication (direct, indirect and environmental communication [5]),

[4] See also the website of the game design program of IAD: http://www.gametheory.ch

disambiguation, focus of attention, meaning for actions, substitution for haptics, etc.

  - *Related to narration and dramaturgy*: emotional cues, influence on perception of time, character development, etc.
  - *Related to space*: navigation, setting, scenography, construction of the virtual soundscape (sonic landmarks, keynote sound, acoustic definition, see [33])

- *"Musical" analysis*: What is the "style" and aesthetics of the game? How do the sounds "work" together aesthetically? How can the "systemic mixing" be judged?
- Also contained in the toolbox is the notational system developed by Schafer mentioned at 5.1.1.

All analyses in the course are done with the help of this toolbox, which ensures the comparability of the results.

Analysis of 2D Arcade Games from the 80s: In this exercise, the sounds of classic arcade games from the 80s (especially from the Atari2600 video game console) are analysed. The analysis of the (technically) rather simple sounds used in those games has several benefits:

- Many basic principles of functional aesthetics of game sound are already contained in the design;
- The limited amount of sounds makes it easier to illustrate those principles;
- Many "standard" sounds still used today have their origin in games from that time, such as jumping sounds or error sounds;
- Last but not least, the careful analysis of the sounds makes it possible to discover their actual richness and variety and shows that aspects of aesthetics and emotional design have always played an important role in game sound design.

Analysis of 3D Games: For this exercise the analysis toolbox is extended by questions related to spatial situation and setting of the sonic events, again based on the notational method described by Schafer (see 5.1.1). This includes questions about where the sound source is located, if and how it moves, how the geographical environment shapes the sound and its propagation, and what other sources are audible in the virtual soundscape.

Also the notion of the acoustic community is investigated in more depth. What does the player hear? What do non player characters (NPCs) "hear"? Especially since more advanced listener models for NPCs have been introduced, such aspects have gained importance for game design.

Comparative Analysis: In the last part of the course the students engage in a more complex project. The goal is to conduct a comparative analysis of two games, using the theoretical and methodological tools provided. Examples of possible comparisons are:

- A "historical analysis", comparing a 2D arcade game from the 80s with a modern 3D title of the same genre;
- A comparison of a category of functional sounds (e.g. hit- or pick-up sounds) between games of different genres;
- The analysis of the transformation between a 2D game and it's three-dimensional counterpart, e.g.. Zelda: A Link to the Past (Nintendo 1991) and Zelda: Twilight Princess (Nintendo 2006)[5].

5.2.2 Relevance to Research Questions

Also here, the first aspect is that the terminology and theoretical concepts can be evaluated through practical application, further developing them towards a specific theory of game sound. Also action-sound relationships specific to games can be investigated and differences between genres can be classified. The notation of sounds, originally created for soundscape studies conducted in physical space, can be tested also in virtual soundscapes. The understanding of the mechanisms of virtual soundscapes can be improved (e.g. aspects of orientation and navigation compared to general "atmospheric backdrop"). For example, a group of students analyzed the soundscape composition of the role-playing (RPG) game Gothic II (JoWood 2002) and described, how the soundscape was supporting mainly orientation and navigation while sonifying battles as rather dull events, underlining the character of "serious work" that battles in RPGs have.

Especially the comparative analyses can serve as input for specific research issues. An example of how the results of such analyses can potentially be used for research is a comparison conducted by a group of students: They compared the sounds of the multiplayer shooter Quake Arena (Activision 1999) with the single-player shooter Serious Sam (Gathering of Developers 2001). The sounds of Quake were very simply structured and had a signal-like quality, while the sounds of Serious Sam were very complex, multi-layered constructions with a lot of variety and effects. From this it can be hypothesized that the sound design for a multiplayer game which is used in e-sports requires an approach directed at optimal recognizability and "legibility" of the sounds and the emotional and narrative component is rather secondary, while in a game like Serious Sam each weapon has a complex "personality" and the general aim is to overwhelm the player in an audiovisual feast.

5.3 Game Sound Design

5.3.1 Course Description

This course lasts about four days and is aimed at students of the 4th semester. The first exercise is to design a small audio play of about one minute duration. Designing an audio play is an excellent exercise for the essential skill of combining and layering sounds in order to create richer meaning. It forces the students to first imagine precisely how e.g. a machine is constructed or what clothes a person wears, in order to design the sounds that evoke a visual image in the listener's head. Furthermore the power of added metaphoric or symbolic sonic cues to the sounds can be investigated.

The game sound toolbox is supplemented by a vocabulary for describing SFX, derived from Beauchamp [4]. This vocabulary contains terms like "whooshes" that are not based on a clear categorization but rather on everyday practice of sound designers.

Constructing Modular Soundscapes: In this assignment the acoustic community and the system of sounding objects introduced in game sound analysis serves as a conceptual background to understanding the possibilities and potentials of today's 3D sound engines, which allow to create not only two dimensional (stereo with fixed channels) soundscapes but actual sound emitter systems that simulate "real" sounding objects in a spatial environment, with the so-called listener object moving between them (see, for example, [28]). The theoretical and conceptual background described above enables the students to think about the

[5] That such 2D-3D analyses have a relevance beyond purely academic questions is clearly shown by the advent of recent titles such as Crush (Sega 2007) or Super Paper Mario (Nintendo 2007) which seamlessly switch between "classic" 2D views and 3D presentation.

technological possibilities in more challenging ways and use it not only to simulate the sonic ”reality” but to transform it and reconstruct it, creating new functional-aesthetic sonic experiences. An example is given below in section 5.4.

Designing Gameplay with Sound: Finally, the students integrate the acquired design skills in a project of their own. Usually, the students use the Torque game engine (GarageGames) or the Unreal editor (Epic) to create 3D games in the context of a specific aesthetic framework or genre. For example, last year's assignment was inspired by the movie Alphaville (Godard 1965). In addition, aspects of design process management are discussed and methods like rapid prototyping or tools like the development process map [7] applied.

5.3.2 Relevance to Research Questions

The design process is conducted along the steps described in game sound analysis and serves to test the analytical framework against a real-life design situation. Here it soon becomes clear, if the conceptual background actually helps to improve design results in any way and if it is useful to formulate and verify design related hypotheses. Moreover, in a design process also new questions for research arise. The introduction of the sound categorization vocabulary commonly used in the industry (e.g. for ”SFX” libraries) allows to critically compare it with the more structured approaches mentioned above (see 5.1.1).

5.4 Experimental Game Design

This last course in the curriculum aims at integrating all components from the previous courses into one big framework. Here the wealth of the theoretical and practical expertise acquired by the students in the prior courses is brought to bear in order to create original and expressive experiences. Usually the projects are developed by combining courses from sound design, 3D design, storytelling and programming. The goal is to cross borders of genres and media and to design for extraordinary contexts, involving for examples aspects of physical space. Often real clients are involved. The best works are usually presented to a larger public and in the annual diploma show, which serves also as a showcase for the whole design department.

An example project is ”Inside/Outside”, which was carried out in fall 2006. The aim of this project was to analyze and deconstruct the real space of the ZHdK buildings and reconstruct them in a game. The special treat was that the presentation was done in the auditorium which disposes of a 360° video-beam setup and a 7.1 surround installation. The project played with the real space of this auditorium, the whole school, and virtual de- and reconstructed versions of it. Sound was used extensively in this project, e.g. to transport the identities of places or transgress physical spaces. The results were presented in the auditorium to the public (See figures 1 and 2).

Figure 1: Final presentation of inside/outside to the public.

Figure 2: One of the projects of inside/outside used mobile videoconferencing to create a game combining virtual and real space

6 Conclusion

In this paper I have described how game sound design is handled in both education and research at the Interaction Design department of the Zurich University of the Arts. There are only few examples that could help to orient the creation of a curriculum for game sound design that suits the requirements of a education in art and design. Also the theoretical foundations are rather thin, albeit growing slowly. I intend to contribute to the research in this domain by linking education and research in our courses in such a way that they mutually enrich and enable each other.

I have described the courses and listed the main topics discussed in them, describing how they relate to research questions. The research questions are focusing on semiotics and functional aesthetics, because I consider these being of special relevance for students in art and design. As the program is still rather young there are not yet many final results or documented impacts on game sound theory but I'm convinced that this will change as the courses iterate.

Doing research in the field of game audio shares some of the challenges also facing design research in general. The question at the core is: how can design research produce scientifically relevant findings? In the approach described here I use theoretical concepts and structured methods in everyday design tasks with students. By documenting results and adapting the ”Toolbox for Game Sound Analysis” I can create a experimental laboratory of empiric study within an educational context which, as I believe, will lead to scientifically relevant results very soon.

Very often game audio is considered to be an add-on which will be taken care of by a musician or sound designer. The gifted sound wizard, the inspired musician are somehow mystical figures that contribute something to game design that everybody considers an important part of the experience, but nobody really understands. In my opinion it is important to demystify the role of the game sound designer and open this field to all game design students. Ultimately it will greatly improve the design process and in the end the quality of an overall game design if everybody - also the programmer, the 3D artist or the project manager - know and understand the role sound plays in games.

References

[1] Games Academy [online]. Available from: http://www.games-academy.de [visited 8.8.2007].

[2] International Game Developers Association. Game Education Special Interest Group [online]. Available from: http://www.igda.org/education [visited 11.6.2007].

[3] Jean-François Augoyard and Henry Torgue, editors. *Sonic experience - A Guide to Everyday Sounds*. McGill-Queen's University Press, Montreal, 2005.

[4] Robin Beauchamp. *Designing Sound for Animation*. Elsevier, Burlington, MA, 2005.

[5] Daniel Bernstein. Creating an Interactive Audio Environment [online]. 1997. Available from: http://www.gamasutra.com/features/19971114/bernstein_01.htm [visited 11.6.2007].

[6] James Boer. *Game Audio Programming*. Charles River Media, Massachusetts, 2003.

[7] Alexander Brandon. *Audio for Games: Planning, Process, and Production*. New Riders, Berkeley, 2005.

[8] Sven E Carlsson. Game Sound Design [online]. Available from: http://www.filmsound.org/game-audio [visited 23.8.2007].

[9] Michel Chion. *Audio-Vision: sound on screen*. Columbia University Press, New York, 1994.

[10] Michel Chion. *Guide des Objets Sonores - Pierre Schaeffer et la recherche musicale*. Buchet/Chastel, Paris, 2nd edition, 1995.

[11] Michel Chion. *Le Son*. Editions Nathan, Paris, 1998.

[12] Qantm college. Games programming [online]. Available from: http://www.qantm.com/study_areas.cfm?id=3 [visited 20.8.2007].

[13] Karen Collins. Glossary of Game Audio [online]. Available from: http://www.gamessound.com/glossary.htm [visited 15.8.2007].

[14] Karen Collins. An Introduction to the Participatory and Non-Linear Aspects of Video Games Audio. In Stan Hawkins and John Richardson, editors, *Essays on Sound and Vision*. Helsinki University Press, Helsinki, Forthcoming.

[15] Mihaly Csikszentmihalyi. *Flow: the psychology of optimal experience*. Harper & Row, New York, 1990.

[16] Inger Ekman. Meaningful Noise: Understanding Sound Effects in Computer Games. In *Proceedings of the 2005 conference on Digital Arts & Culture*, 2005.

[17] Barbara Flückiger. *Sounddesign: Die virtuelle Klangwelt des Films*. Schüren Verlag, Marburg, 2001.

[18] Karmen Franinovic, Daniel Hug, and Yon Visell. Sound Embodied: Explorations of Sonic Interaction Design for everyday objects in a workshop setting. In *Proceedings of the 13th international conference on Auditory Display*, 2007.

[19] J. Friberg and D. Gärdenfors. Audio Games: New Perspectives on Game Audio. In *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, 2004.

[20] Gamasutra. Game Audio Track [online]. Available from: http://www.gamasutra.com/php-bin/article_ display.php?category=1 [visited 24.8.2007].

[21] W. W. Gaver. What in the world do we hear? An ecological approach to auditory event perception. *Ecological Psychology*, (5):1–29, 1993.

[22] Interactive Audio Special Interest Group. IASIG Game Audio Education Working Group [online]. Available from: http://www.iasig.org/wg/closed/eduwg [visited 28.8.2007].

[23] Interactive Audio Special Interest Group. IASIG Wiki [online]. Available from: http://www.iasig.org/wiki [visited 28.8.2007].

[24] Game Audio Network Guild [online]. Available from: http://www.audiogang.org [visited 24.7.2007].

[25] Daniel Hug. Hear Me Interact. In G. Buurman, editor, *Total Interaction - Theory and practice of a new paradigm for the design disciplines*. Birkhäuser, Basel, 2005.

[26] Daniel Hug. *Sounds for Movers and Shakers*. forthcoming, 2007.

[27] Don Ihde. *Listening and Voice: A Phenomenology of Sound*. Ohio University Press, Athens, Ohio, 1976.

[28] Creative Labs. OpenAL Documentation [online]. Available from: http://www.openal.org/documentation.html [visited 23.8.2007].

[29] Aaron Marks. *The Complete Guide to Game Audio*. CMP Books, Kansas, 2001.

[30] School of Audio Engineering. Games Programming: Diploma of Game Design [online]. Available from: http://www.sae.edu/courses/diploma_of_game_design [visited 2.8.2007].

[31] Randy Pausch and Don Marinelli. Combining the Left and Right Brain. *Communications of the ACM*, 50:50–57, 2007.

[32] Pierre Schaeffer. *Traité des objets musicaux*. Seuil, Paris, 1966.

[33] R. Murray Schafer. *The Soundscape: Our Sonic Environment and the Tuning of the World*. Destiny Books, New York, 2nd edition 1994 edition, 1977.

[34] Nathan Shedroff. *experience design 1*. New Riders, Indiana, 2001.

[35] Barry Truax. *Acoustic Communication*. Ablex, 2nd edition, 2000.

[36] Theo van Leeuwen. *Speech, Music, Sound*. Palgrave Macmillan, 1999.

[37] Wolfgang Welsch. Auf dem Weg zu einer Kultur des Hörens? In *Grenzgänge der Ästhetik*. Reclam, Stuttgart, 1996.

[38] Martin D. Wilde. *Audio Programming for Interactive Games*. Focal Press, Oxford, 2004.

Associating Graphical Objects with Musical Tempo

Jukka Holm, Visiokatu 1, FIN-33720, Tampere, Finland. Email: jukka.a.holm@nokia.com
Antti Aaltonen, Visiokatu 1, FIN-33720, Tampere, Finland. Email: antti.aaltonen@nokia.com

Abstract. Portable music players and music-oriented mobile phones are becoming increasingly popular. At the same time, the available storage space is growing and enabling music collections of thousands of songs. The navigating in and interacting with such large number of songs can be a tedious task with the device that has a small display and limited interaction capabilities. One attractive alternative to current list- or tree-based solutions is interactive visualizations, which facilitate browsing and organizing personal music libraries and thus ease in selecting what kind of music to listen to. The essential idea of these visualizations is to replace the traditional UI with a dynamic, adaptive, and interactive image. To design the visualizations, the authors have arranged several online questionnaires on how people map images to music. This paper presents the results of one of the questionnaires, focusing on mapping musical tempo to various visual features such as the number of objects, shape, size, orientation, color, and blur.

1. Introduction

The music business is in a transition as the sales of CDs drop and online sales of digital music grow. In addition, portable music players (e.g. Apple iPod) and music-focused mobile phones are becoming increasingly popular and thus, music consumption is going mobile. The increased storage space of the music devices allows storing of more songs in the users' collections. For instance, the newest iPods can store up to 20 000 songs. Navigating in and interacting with such a large collection can be a hard task on a small screen of a mobile device that has limited interaction capabilities. The user may forget his Whitesnake and Steep albums for a long time, because he never manages to scroll the artist list to that point. Instead of using current list-based presentations, one alternative is to create new interactive visualizations of musical content, which facilitate browsing and organizing personal music libraries and thus ease in selecting what kind of music to listen to.

Information visualization is about using "*computer-supported, interactive, visual representation of abstract data to amplify cognition* [12]." For example, the following ways are proposed for amplifying cognition: shifting workload from a human's cognitive system to their perceptual system, reducing searching, and enhancing recognition of patterns. Hence, the idea is to replace the current list or tree presentations of a music library with a dynamic, adaptive, and interactive visualization. Certain visual elements of the picture are mapped to user's own music collection or a music recommendation service. For example, the user clicks on a sharp black object and playback of fast heavy metal from his music collection starts, or clicking a round green object starts the playback of slow country music.

To study visualizations for music, we arranged a set of online questionnaires on how people map images to music. This paper presents the results of one questionnaire, concentrating on mapping musical tempo to various visual features of images.

2. Related Work

Lots of previous research, applications, and online services exist for visualization of large music collections, generation of playlists, etc. In the following, we focus on those where tempo has been used as one parameter for selecting the songs.

Van Gulik's artist map user interface [1,2] provides an overview to an entire music collection or its subset by calculating the similarities between artists and visualizing them on the screen. Colors in the UI can be mapped to, e.g., mood, genre, and tempo. In one of the examples, clustering was based on release year (*x*-axis) and tempo (*y*-axis). The coloring was done on tempo as follows: very slow = blue, slow = cyan, medium = green, fast = yellow, and very fast = red.

Zhu [8] developed "*an algorithm to automatically estimate human perceptions on rhythm and timbre of a music clip. Then, based on these two values, each music clip is mapped into a 2D (timbre-rhythm) space*." In the visualization, brightness decreases from top to bottom and tempo from right to left.

Musicovery [9] is an interactive web radio, where the user is able to select music according to mood (dark, energetic, calm, positive), danceability, decade, tempo, and genre. Tempo has been mapped to the *y*-axis, and decreases from top to bottom.

Moody [3] is a mood-based playlist generator for iTunes. To start with, the user has to tag his whole iTunes library according to mood. This is done along two axes, where *y*-axis represents intensity and *x*-axis denotes happiness. The axes are color-coded in such way that red represents intensive but sad music, yellow intense and happy music, blue calm and sad music, and green happy but calm music. The mood is saved in the comments field of the ID3 tags. Once the library is tagged, the user can define new playlists based on mood.

In Musiclens [4], the user can adjust sliders to get new recommendations. One of the sliders adjusts tempo, which decreases from top to bottom.

With the UPF Music Surfer [5] the user can select an artist, album, or track, or upload his own track to find similar music from the service. The similarity maps can be arranged according to e.g. tempo (BPM), timbre, rhythm, genre, danceability, key, or tonality (major/minor).

In the case of Apple iTunes, the user can e.g. create smart playlists that match certain BPM conditions (e.g. add song to playlist if its tempo is higher than 120 BPM).

3. Online Questionnaire

To study how people associate musical tempo with various visual features of images, an online questionnaire was arranged. The questionnaire consisted of 17 pictures where one visual attribute at the time was varied in a set of objects. In each picture, the participants were asked to compare different objects and then select which one seems to have the highest (or the lowest) tempo. Some of the visual features (e.g. size and location) were selected because they are prominent and logical, and others were based on previous research and our sketches for visual music player user interfaces.

The call for participation was sent to c. 200 employees of a large international company that has employees of several nationalities. Participation was voluntary and the participants did not receive any kind of compensation for participating in the study. 75 persons answered the questionnaire, and 80% of them were male and 20% female. 63 % (47 persons) were Finnish, 8% (6 persons) Indian, 11% (8 persons) Chinese, and the rest from various nationalities such as American, Canadian, Irish, German, and Mexican. Nearly 90% of the participants were 25-40 years old, and majority had engineering background. Next, we discuss the results of the questionnaire in more detail.

4. Results

Due to the high number of Finnish participants, it is obvious that their votes dominate in the results. We also studied how the votes were split between the different nationalities, but did not notice any significant differences. However, we acknowledge that the number of non-Finnish participants is low, which means that this finding is merely suggestive.

4.1. Number of Objects

In the first part of the questionnaire, the participants were told that each option (A, B, C in Figure 1) represents a single song with a different tempo. They were asked to select one option in order to listen to fast music. The participants were also able to select option ”No opinion.”

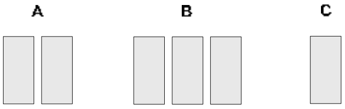


Figure 1: Associating the number of objects with tempo.

As expected, most of the participants (84.0%) considered that option B has the highest tempo. The other options received only a few votes (option A 4.0% and option C 6.7%) and 5.3% of the participants had no opinion. Consequently, the result implies that the number of objects could be a good parameter for representing tempo, and high numbers of objects map to faster tempos than a low numbers.

4.2. Size of Objects

In the next question, the participants were asked which of objects shown in Figure 2 they associate with fast music.

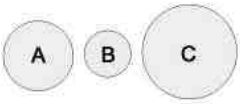


Figure 2: Associating the size of objects with tempo.

The majority of participants (56.0%) preferred the small-sized object for presenting fast songs. However, over one fourth (28.0%) favored the large objects and 10.7% did not have an opinion. Interestingly, the mid-size (option A) was favored by 5.3%, which could be explained by that fact people did not read the instructions carefully enough (e.g.,” please do not make your selection based on the location or order of pictures”). Also, the size difference could have not been adequate even though the factor of diameter was varied from 1 to 1.5 and to 2. As a result, the size of object alone is a poor variable for representing tempo.

4.3. Shape

Next we studied how varying the number of corners of a shape would affect perceiving tempo. The participants were asked to select shapes for listening to slow and fast music (Figure 3).

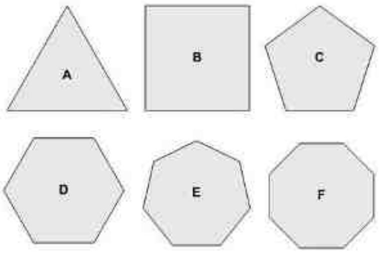


Figure 3: Associating different shapes with tempo.

In both cases the votes were evenly divided between three different objects. Analyzing the results and the comments of the participants revealed that the majority associated the higher number of corners with a faster tempo (Table 1).

Table 1: Votes for shapes with different number of corners.

Fast music			Slow music		
Option	Number of votes	% of votes	Option	Number of votes	% of votes
A	18	24.0	A	20	26.7
B	2	2.7	B	23	30.7
C	2	2.7	C	4	5.3
D	1	1.3	D	3	4.0
E	21	28.0	E	0	0.0
F	24	32.0	F	17	22.7
No opinion	7	9.3	No opinion	8	10.6

Figure 3 shows that as the number of corners increases, the shape starts to become increasingly circular. So, we studied how associating tempo to an object without corners (circle) would relate to objects with sharp corners (Figure 4). The task was the same as with the previous set of objects. The selection of shapes A (circle) and D (star) was based on our findings from comics [10,11] and art. In comics, spiky shapes are often used in balloons to illustrate that a character is aggressive or shouting.

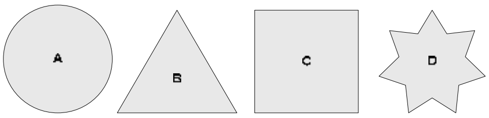


Figure 4: Associating circle, triangle, square, and star shapes with tempo.

The results were encouraging (Table 2). 84% of the participants mapped the star to fast music, while the circle was associated with slow music (68%). The reasons for associating the star with fast music were, e.g., that a shape with many corners implies fast music; sharp corners resemble fast beats; and the object looks like thunder and explosion. The circle was associated with slow music, for example, because a simple shape relates to slow music; less corners imply a slower tempo; and the circle is a round, calm, and peaceful shape. As the percentage values are so high, it seems that our comics-like approach could be an interesting alternative for representing musical tempo.

Table 2: Votes for circle, triangle, square, and star.

Fast music			Slow music		
Option	Number of votes	% of votes	Option	Number of votes	% of votes
A	4	5.3	A	51	68.0
B	4	5.3	B	9	12.0
C	1	1.4	C	9	12.0
D	63	84.0	D	1	1.3
No opinion	3	4.0	No opinion	5	6.7

4.4. Blur

This part of the questionnaire studied how blurring objects affects to perceiving tempo (Figure 5).

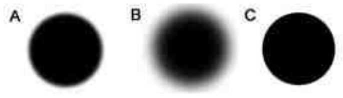


Figure 5: Associating blurring with tempo.

Nearly half of the participants linked blurred objects to slow music (49.3%) and sharp objects to fast music (52.0%). However, 32.0% of the participants mapped the blurred object to fast music and 24.0% the sharp object to slow music. The conflicting result implies that blurring is a poor way to present tempo.

4.5. Foreground Objects Versus Background Objects

Next we studied could the depth order of objects represent tempo and in this case, do people associate fast music with objects closer to or further away from the foreground.

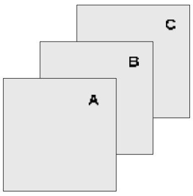


Figure 6: Associating objects on foreground and background with tempo.

The votes were split evenly between foreground (25.3%) and background (20.0%), while over half of the participants (53.3%) did not have an opinion. Consequently, using the depth order for representing musical tempo is a very poor attribute.

4.6. Vertical Location Versus Horizontal Location

In this part of the questionnaire, the participants were instructed that each rectangle represents a collection of songs and each circle (marked with A-X) symbolizes a single song (Figure 7). In the case of first rectangle, the participants were asked which of the songs is faster: A or B? The same question was repeated for each pair of circles shown in Figure 7. The participants were also able to select option “both songs have the same tempo” or “no opinion.”

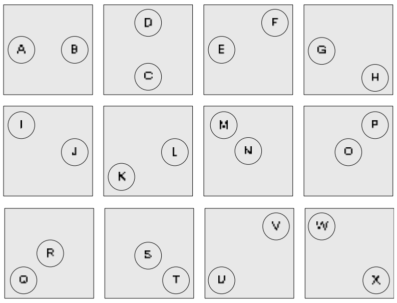


Figure 7: Associating location with tempo.

Based on the results (Table 3), people map tempo to the *y*-axis in such a way that faster songs are located on the top and slower songs on the bottom. Most votes (c. 40-60%) were consistently given to circles having the highest *y*-coordinate value (see bolded figures in Table 3). Those circles that also had the highest *x*-coordinate value got the greatest number of votes. In the case where the circles A and B had the same *y*-coordinate value, most votes (40.0%) were given to option “both songs have the same tempo.”

Despite the low differences in percentages of votes, we consider mapping tempo to the *y*-axis to be a promising parameter. Since on average one fourth of the participants did not express their opinion, the *y*-axis should be labeled clearly (as in e.g. [9]) or the visualization should be supported with some other visual parameters presented in this paper.

Table 3: Votes for circles in different locations.

Fast music					
Option	% of votes	Option	% of votes	Same tempo %	No opinion %
A	8.0	B	32.0	40.0	20.0
C	5.3	D	**52.0**	18.7	24.1
E	5.3	F	**54.7**	18.7	21.3
G	**45.3**	H	18.7	8.0	28.0
I	**45.3**	J	18.7	8.0	28.0
K	9.3	L	**50.7**	17.3	22.7
M	**44.0**	N	13.4	13.3	29.3
O	4.0	P	**60.0**	12.0	24.0
Q	6.7	R	**58.7**	9.3	25.3
S	**40.0**	T	21.3	9.4	29.3
U	5.3	V	**58.7**	16.0	20.0
W	**48.0**	X	14.6	10.7	26.7

4.7. Orientation

To study the effect of orientation, the participants were asked which of the rectangles (Figure 8) seem to have the highest and the slowest tempos.

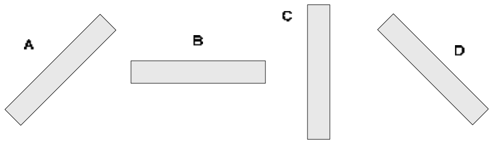


Figure 8: Associating orientation with tempo.

For the fast music, 38.7% of the participants selected option A (rising slope) and 28% voted for option B (pointing up). The reasons were, e.g., that pointing up implies fast tempo, tempo increases to the right, upward slope implies fast music, and going upwards means that the beat gets faster. In the case of slow music, option B received 50.7% and option D had 20% of the votes. The reasons for selecting B now were, e.g., that the object looks like lying, lazy, sleeping, or flat, and thus relates to slow tempo. Again nearly 20% of the participants did not have an opinion. This combined with relatively low percentages suggest that orientation only is a poor choice for presenting musical tempo.

4.8. Size Versus Location

Next we studied if size dominates over location or vice versa. Participants were instructed that each picture (A-D in Figure 9) represents a collection of music and each rectangle represents a single song with a different tempo. They were asked to select one part in each picture to listen to fast music. In the cases A and B, the participants could select top, bottom, and no opinion. For the cases C and D, the options were left, right, and no opinion.

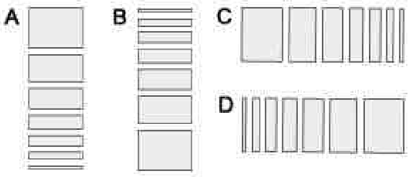


Figure 9: Associating objects with different sized rectangles with tempo.

The results (Table 4) suggest strongly that size dominates location. As discussed in Section 4.2, people tend to associate small objects with fast tempos. In the case of picture A, 60.0% of the participants associated the bottom of the picture (small rectangles) with fast songs. In the next case, the small rectangles were now located on the top that received 80.0% of the votes. This finding relates also to cases C and D: the small rectangles located on the right (C) and left (D) sides of the pictures receive most of the votes (78.7% and 60.0%, respectively). However, location may have a small effect on the results; when the small rectangles are located on top or right, the percentages are higher.

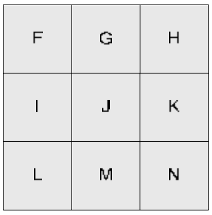
Table 4: Votes for objects with different sized rectangles.

	Fast music			
	Number of votes	**% of votes**	**Number of votes**	**% of votes**
Option	**A**		**B**	
Top	26	34.7	60	80.0
Bottom	45	60.0	11	14.7
No opinion	4	5.3	4	5.3
Option	**C**		**D**	
Left	12	16.0	45	60.0
Right	59	78.7	25	33.3
No opinion	4	5.3	5	6.7

Due to the high percentages of the most voted options, it can be concluded that this type of graphical presentation may be a good way to present musical tempo.

4.9. Location

In the next question, the participants were asked which cells in the grid (Figure 10) have the fastest and the slowest tempo.



F	G	H
I	J	K
L	M	N

Figure 10: Associating grid location with tempo.

The results were as anticipated. 38.7% of participants considered cell H to be the fastest, and 34.7% considered cell L as the slowest. This is in line with Section 4.6, which indicated that fast songs have higher *y*-coordinate values than slower ones. The rest of the cells received only few votes – roughly 3% on the average. Still, the amount of participants that did not have an opinion was roughly 40% which reduces the suitability of using this attribute alone.

4.10. Color

After this we studied how tempo associates with colors. We used 12 colors (red, green, yellow, blue, black, white, pink, cyan, gray, orange, brown, and purple) recommended for color-coding [6]. The following introduction was given to the participants: "There are 12 songs in a music collection. Each color is mapped to a single song and each song has a different tempo."

First the participants were asked to select a color for the fastest song. The majority (60.0%) of participants picked red, while the other votes were evenly divided between the other colors (Table 5). The participants were also asked to justify their selection, and the reasons for selecting red were, e.g., aggressive, fast, danger, heat, fire, active, root chakra, passion, and hot end of RPM meters. These reasons are easy to understand and also in line with western color symbolism (see e.g. [7]). Interestingly, all Chinese participants (8) considered red to be the fastest color (for Finnish the number was 61.7%). One explanation mentioned was that red represents happiness in China, and fast music also often represents happy emotion.

Next the participants were asked to select one or two other colors for fast songs. The most voted colors were yellow (22.6%) and orange (17.4%). Red came again third (13.0%), which is explained by the votes of people who did not consider red as the fastest in previous question. The main reasons for selecting orange and yellow were, e.g., fast, warm, hot, getting hotter, close to red, happy, and bright colors that feel active.

Table 5: Votes for mapping colors to fast tempo.

Fastest (75 votes)			**Other fast** (115 votes)		
Option	**Number of votes**	**% of votes**	**Option**	**Number of votes**	**% of votes**
Red	45	60.0	Red	15	13.0
Green	0	0.0	Green	0	0.0
Yellow	2	2.7	Yellow	26	22.6
Blue	1	1.3	Blue	6	5.2
Black	5	6.7	Black	9	7.8
White	4	5.3	White	7	6.1
Pink	0	0.0	Pink	8	7.0
Cyan	3	4.0	Cyan	4	3.5
Gray	1	1.3	Gray	1	0.8
Orange	5	6.7	Orange	20	17.4
Brown	0	0.0	Brown	1	0.9
Purple	1	1.3	Purple	7	6.1
No opinion	8	10.7	No opinion	11	9.6

Surprisingly, white was the most voted for the slowest color (20.0% of all votes). The next choices were blue (13.3%), black (12.0%), gray (12.0%), and brown (10.7%), while 13.3% of participants did not state any opinion (Table 6). The reasons for selecting these colors were that, e.g., white is a calm, steady, stable, slow, neutral, and bland color; blue is cool, slow, serene, and cold; gray is a dull color that seems to be slow; and brown is calm, down to earth, solid, and boring. When asked that what other colors could be mapped to slow songs, the most popular answer was gray (18.6% of votes) followed by brown, purple, and blue. The reasons were e.g. that a large purple area may be relaxing and blue is the opposite of red.

Table 6. Votes for mapping colors to slow tempo.

Slowest (75 votes)			**Other slow** (118 votes)		
Option	**Number of votes**	**% of votes**	**Option**	**Number of votes**	**% of votes**
Red	0	0.0	Red	1	0.8
Green	6	8.0	Green	7	5.9
Yellow	1	1.3	Yellow	9	7.6
Blue	10	13.3	Blue	11	9.3
Black	9	12.0	Black	6	5.1
White	15	20.0	White	7	5.9
Pink	1	1.4	Pink	8	6.8
Cyan	2	2.6	Cyan	7	5.9
Gray	9	12.0	Gray	22	18.6
Orange	1	1.3	Orange	2	1.7
Brown	8	10.7	Brown	15	12.7
Purple	3	5.0	Purple	11	9.3
No opinion	10	13.3	No opinion	12	10.2

The final question was which colors should be mapped to songs having a medium tempo (i.e. songs which are not fast nor slow either). Two colors, namely green and blue, rise slightly above the others with 22.9% and 16.1%, respectively (Table 7).

Table 7. Votes for mapping colors to medium tempo.

Medium (118 votes)					
Option	**Number of votes**	**% of votes**	**Option**	**Number of votes**	**% of votes**
Red	1	0.9	Green	27	22.9
Yellow	2	1.7	Blue	19	16.1
Black	1	0.8	White	2	1.7
Pink	9	7.6	Cyan	2	1.7
Gray	13	11.0	Orange	13	11.0
Brown	10	8.5	Purple	9	7.6
No opinion	10	8.5			

As a conclusion, red seems a safe choice for representing fast music. In the case slow or medium songs, the given votes for the most preferred colors are so widely distributed that there is no solid answer for the selection of right colors.

4.11. Circles and Spider Web

The participants were next asked which parts of Figure 11 have the fastest tempo.

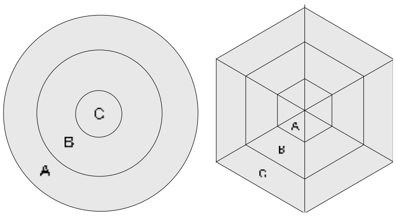


Figure 11: Associating circle and spider web objects with tempo.

In both cases, the results were roughly similar; 64.0% of participants associated the center of pictures with the highest tempo. The outer ring and “no opinion” received practically the rest of the votes (varying between 12% and 22%), as the ring on the middle received only one vote (1.3%) in the circle case.

When asked the reason for the selection, many commented that they selected the center because they consider fast to be small. Again, this is in line with the results of Section 4.2, where 56.0% of the participants voted for small-sized objects to represent fast songs.

4.12. Triangle

In this question, the participants were asked which corner of a triangle (Figure 12) could be mapped to fast music.

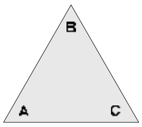


Figure 12: Associating the corners of a triangle with tempo.

The preferred option was the top (45.3% of the votes), while bottom left and right corners received only 6.7% and 12% respectively. However, a large portion of the participants (36.0%) did not have any opinion on the question. The results are in line with the other results of this paper, but as the percentages are low, it seems that using the corners of a triangle may not be a suitable way for representing musical tempo.

4.13. Lightness

In this part of the questionnaire, the participants were asked which part (top, bottom, left, or right) of the pictures A-D in Figure 13 they would select to listen to fast music.

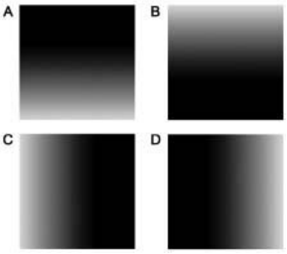


Figure 13: Associating lightness with tempo.

In the cases A and B, where the lightness changes vertically, half of participants considered that top of the picture should be mapped to faster songs than the bottom. One fourth associated the bottom of the picture to fast music (Table 9). The direction of lightness change thus seems to determine the tempo axis, and top dominates over bottom and lightness. In the cases C and D, the lightness changes horizontally and the results resemble those of pictures A and B. However, the distribution is now more even between left and right: while 48% (C) and 42.0% (D) of participants associated the right side with fast songs, over 30% preferred left.

Table 9: Votes for pictures with varying lightness.

Fast music					
A			**B**		
Option	**Number of votes**	**% of votes**	**Option**	**Number of votes**	**% of votes**
Top	38	50.7	Top	37	49.3
Bottom	20	26.7	Bottom	19	25.3
Left	0	0.0	Left	0	0.0
Right	4	5.3	Right	5	6.7
No opinion	13	17.3	No opinion	14	18.7
C			D		
Option	**Number of votes**	**% of votes**	**Option**	**Number of votes**	**% of votes**
Top	1	1.3	Top	2	2.7
Bottom	0	0.0	Bottom	0	0.0
Left	23	30.7	Left	24	32.0
Right	36	48.0	Right	36	42.0
No opinion	15	20.1	No opinion	13	17.3

The results seem to follow the logic that the direction of lightness change defines the axis for mapping the tempo. The differences between the percentages are low, thus implying that lightness may not be the best way to present tempo.

4.14. Vertical and Horizontal Bars

In the last part of the questionnaire, the participants were asked which area (A-C, Figure 14) seems to have the highest tempo.

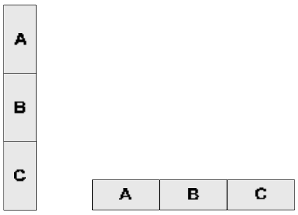


Figure 14: Associating vertical and horizontal bars with tempo.

As earlier, the results (Table 10) indicate that areas located on top or right should be mapped to faster songs than areas located on bottom or left. However, this result is largely dominated by the Finnish people as almost 70% of them had this opinion.

Table 10: Votes for vertical and horizontal bars.

Fast music					
Vertical bar			Horizontal bar		
Option	Number of votes	% of votes	Option	Number of votes	% of votes
A	48	64.0	A	7	9.3
B	0	0.0	B	1	1.3
C	6	6.0	C	44	58.7
No opinion	21	21.0	No opinion	23	30.7

5. Conclusions and Future Work

We presented the results of our online questionnaire that focused on mapping musical tempo to various visual features. It should be noted that the results are dominated by the Finnish participants.

According to the results, the number of objects is a good way of presenting tempo and high numbers of objects map to faster tempos than a low numbers. Also the comics-like approach, where tempo was mapped to shapes with varying number of corners, was successful as the majority of participants associated the higher number of corners with a faster tempo. Especially the star shape (84.0%, fast music) and circle (68.0%, slow music) received excellent results.

Over half of the participants (56.0%) preferred small-sized objects for presenting fast songs. When combined with some other visual feature, the percentage increases further. E.g., in the case of circle and spider web objects that were split into multiple regions, 64.0% of participants answered that the center of pictures (which had the smallest area) has the highest tempo. The participants were also shown pictures where the size of rectangles increased from left to right, top to bottom, or vice versa. When the small rectangles were located on the top and large ones on the bottom, 80.0% of participants associated the top with fast music.

Our results indicate strongly that people map tempo to the y-axis (instead of x-axis) in such a way that faster songs are located on the top and slower songs on the bottom. However, the percentage values varied depending on the pictures and the formulation of the question. When asked which one of two circles in a rectangle has the highest tempo, most votes (c. 40-60%) were consistently given to circles having the highest y-coordinate value. For a triangle, the preferred option was the top with 45.3% of the votes. In the case of a vertical bar, the topmost part received 64.0% of the votes. Because of relatively low percentage values, the y-axis should be labeled clearly or the visualization should be supported e.g. with some other visual parameters discussed in the paper.

Also in the case of vertical change of lightness, roughly half of participants mapped the top of the picture to faster songs than the bottom. In the case of a horizontal change, the right side dominated.

The mapping of colors to tempo was also studied. As a conclusion, red (60%) seems to be a good choice for representing fast music. In the case slow or medium songs, the votes for the preferred colors were more distributed, and thus there is no answer for the selection of colors. Likewise, the level of blur and depth order seem to be poor alternatives for representing tempo.

In the future we will arrange more online questionnaires on mapping images to music. Potential topics include e.g. symbols, fonts, facial expressions, avatars, user taken photos, listening history as well as favorite songs, albums, and artists. It would also be interesting to study how much the results improve if two or more visual features at the same time are used to represent a single musical parameter such as tempo. The results of all questionnaires will be used to design new user interfaces for mobile music players, which will be then implemented and user tested.

References

[1] Van Gulik, R., Vignoli, F., van de Wetering, H., *Mapping Music in the Palm of Your Hand, Explore and Discover Your Collection*, Proceedings of ISMIR Conference (2004)

[2] Van Gulik, R., Vignoli, R., *Visual Playlist Generation on the Artist Map*, Proceedings of ISMIR Conference (2005)

[3] *Moody*, http://www.crayonroom.com/moody.php (2007)

[4] *Musiclens*, http://www.musicline.de/de/$255dae7db14eabefc87799d69cb952e7/recoengine04/start/ml/musiclens (2007)

[5] *UPF Music Surfer*, http://musicsurfer.iua.upf.edu (2007)

[6] Ware, C., *Information Visualization*, 2nd Edition, San Francisco, Morgan Kaufmann Publishers, p. 125 (2004)

[7] *Color Symbolism*, http://en.wikipedia.org/wiki/Color_Symbolism (2007)

[8] Zhu, J., Lu, L., *Perceptual Visualization of a Music Collection*, IEEE Transactions on Speech and Audio Processing, Vol. 10, No. 5 (2002)

[9] *Musicovery*, www.musicovery.com (2007)

[10] McCloud, S., *Understanding Comics*, New York, HarperCollins Publishers, (1994)

[11] *The Essential Punisher Vol. 1*, Marvel Comics, (2004)

[12] Card, S. K., Mackinlay, J. D., Shneiderman, B., *Readings in Information Visualization: Using Vision to Think,* San Francisco, Morgan Kaufmann Publishers, (1999)

Special Presentations

MacDonald, Raymond
Keynote: Music – a universal language 181

Herre, Jürgen
Keynote: Personal Audio – From simple sound reproduction to personalized interactive rendering 181

Ashby, Simon
Dynamic Audio Mixing for Games 182

Grünenberg, Reginald
Beyond Broadcasting's First Real Technology: Interactive Streaming........ 183

Johnsen, Gisle
Interactive composing for young students........ 186

MacDonald, Raymond

Music – a universal language

We are all musical; every human being has a social biological guarantee of musicianship. Not a vague utopian ideal but rather a conclusion drawn by an increasing number of academic researchers involved in investigating the foundations of musical behaviour. Moreover, the earliest communications between a parent and a child are essentially musical and to respond emotionally to music is a defining feature of our humanity. We can indeed sing before we can talk.

Music plays a greater part in the everyday lives of more people than at any time in the past. This is partly the result of the extremely rapid technological developments that have occurred in the last two decades or so, allied to the increasing commercialisation and economic power of the music industry. In the developed countries of the world at least, the widespread availability and relative inexpensiveness of MP3 players, the internet, the MIDI interface, the video recorder and more means that a vast diversity of musical styles and genres is available to us as listeners. The ways in which people experience music – as 'consumers', fans, listeners, composers, arrangers, performers or critics – are far more diverse than at any time in the past, as are the range of contexts in which this takes place. The topics of musical identities and musical communication will be introduced by summarising recent work that suggests the ubiquitous nature of music within modern life signals possible evolutionary functions of music and that our musical tastes and preferences are crucial indicators of who we are.

This lecture will also present overviews of a number of research projects highlighting the multifaceted ways in which music plays an important role and influences our life. Research studies utilising an Indonesian Gamelan that highlight the use of music for individuals with special needs will demonstrate how music can be utilised to facilitate psychological and musical developments. A summary of a project investigating the provision of music within Scottish education will be presented. This study highlights the fundamental importance of improvisation within musical development and it will be suggested that improvisation is currently a much under used strategy within music education. A number of studies showing how music can be used for pain and anxiety relief in hospitals will also be presented

In summary, I aim to demonstrate how music is a fundamental channel of communication: it provides a means by which people can share emotions, intentions, and meanings even though their spoken languages may be mutually incomprehensible. It can also provide a vital lifeline to human interaction for those whose special needs make other means of communication difficult. Music can exert powerful physical effects, can produce deep and profound emotions within us, and can be used to generate infinitely subtle variations of expressiveness by skilled composers and performers.

Herre, Jürgen

Personal Audio – From simple sound reproduction to personalized interactive rendering

For a long time, one of the foremost goals in audio engineering has been seen in the faithful transmission or reproduction of recorded sound for the listeners. More recently, with the growing use of computers and digital signal processing to create a rich multimedia experience, the notions of personalized and interactive audio have gained increasing importance: For many applications, users desire the ability to interact with the reproduced sound in an intuitive way and adapt it to their personal preferences. Applications like karaoke/play-along, gaming and teleconferencing call for new levels of interactivity. This talk discusses some relevant steps on the long road towards comprehensive interactivity. Special attention will be given to technologies that enable both bitrate and computationally efficient representations of interactive sound, including the ongoing developments within the ISO/MPEG standardization group.

Ashby, Simon

Dynamic Audio Mixing for Games

Dynamic Audio mixing techniques have been the mainstay of the film and postproduction industries for years. Given the linear nature of these mediums, dynamic audio mixing is easily controlled and predictable. Mixing game audio brings with it many challenges, including performance constraints and the non-linear event based triggering of in-game sounds. Current game console technology now provides enough processing power to allow Sound Engine architectures to support advanced audio pipelines by incorporating real-time dynamic mixing functionality. Using real-game practical audio examples, this session will demonstrate the many positive benefits that dynamic audio mixing can have on modern sound design.

Session Topics
The dynamic mixing concepts presented in this session can be applied at three different conceptual levels:

- Atomic – focuses on dynamic mixing techniques that can be applied to sound entities at the atomic level, only affecting individual objects. Real game audio examples will be used in the following areas of interest: parameter value randomization, distance attenuation, and controlling audio parameters in real-time using game data.

- Group – presents advanced mixing techniques that result from the dynamic interaction of atomic objects and parameters combined with behavioral and mixing hierarchies. Examples will include: setting audio parameters via event triggers, mix bus routing, auto-ducking, and voice management.

- Global – demonstrates mixing techniques that apply to all levels of sound entities. Practical examples include occlusion and obstruction, game states, and game environments.

Grünenberg, Reginald

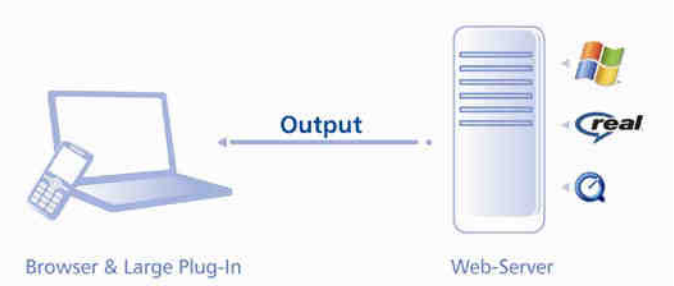
Beyond Broadcasting's First Real Technology: Interactive Streaming

The convergence of digital media requires a new paradigm of interactivity. Traditional broadcasting is already challenged by a swarm of new applications called weblogs, wikis, social software, mashups and RSS. But the dazzling career of the resulting participative and collaborative media culture that is summarized as Web 2.0 is not based on new technologies. Everything was in place for Web 2.0., existing tools and programming languages like AJAX were just combined in a smart way. Now, Interactive Audio Streaming – later also Interactive Video Streaming – comes along as the first technology that in itself really goes Beyond Broadcasting. Every single client device gets its own individual stream, the composition of which is permanently controlled by the user's interaction, e. g. through cursor movements, clicks, speech or his/her GPS data. The number and kind of audio sources that can be switched or mixed in real-time is virtually unlimited: pre-recorded, dynamically generated (e. g. text-to-speech) and live audio signals can be mixed into one interactive stream. This technology is platform, codec, bandwidth and client agnostic (no plug-in). From this perspective stationary and mobile Internet are just the same. It's a new audio-interface for every IP enabled client, be it IPTV, network printer, set-top box, car, mobile phone or PC.

Interactive Streaming enables a huge variety of new applications in the fields of e/mcommerce, e/m-learning, e/m-entertainment, enabling web and online advertising. It will end the era of silent movie on the Internet, boost Web 2.0, provide an infrastructure for the convergence of digital media and make Beyond Broadcasting becoming the promise of a new generation of interactive, dialogue based technologies. We will show some compelling demos and pilot cases.

The Internet is young. It is still a visual and text medium. It has not yet learned to speak with us. Of course there is now some rich media content, most prominently Flash movies like the ones on Youtube. But this is not what we mean. Rich media content nowadays is not interactive at all, not dialogue based. It is far below the possibilities of the Internet and IP based networks. It is an impressive monologue that is still slave to the old Broadcasting Paradigm. Everybody hears and sees the same digital content. No chance to (co-)create, control, monitor or influence the rich media content coming on my client device. The poor principle of this interaction between server and client is "Play/Stop".

Fig. 1 Common Streaming Technologies



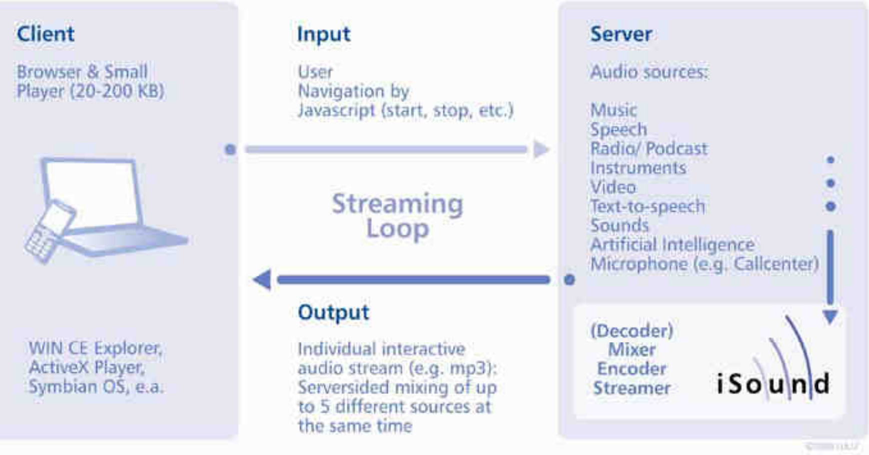
Audiantis, a software start-up based in Berlin and Tokyo, has developed a technology that heralds the end of this silent film age for the Internet and all intranets. And it is the first of a new generation of technologies that will leave behind the Broadcasting Paradigm. With a fundamentally new procedure called Interactive Audio Streaming websites on all IP enabled client devices with audio output can be upgraded with a sensitive and intelligent audio layer.

It is nothing less than a new audio interface. The visual interface will be enriched and becomes more attractive, exciting, informative, entertaining and – more emotional. The ultimate USP (unique selling proposition) of this audio technology is the quick creation of emotion. A novel or a movie also creates emotions by the meaning of its narration. But that takes far more time. The immediate access to the stimulation of emotions is sound. The creation of emotion is not (necessarily) an end in itself or just another technique of manipulation and social control. In an increasingly complex world the emotional dimension is best prepared to embed naked information that doesn't mean anything to the perceiving and reflecting subject. Sound provides an emotional sense of orientation and increases at the same time dramatically the memorizing capacity of the brain. This of amplification of the sensuous capacities requires that the multimedia industry discovers a neglected dimension for creative design and online interaction: Audio.

As an IT start-up we are faced with a rare problem, a problem which only then arises when a new technology, with its cargo of enormous configurative potential, has to be brought safely down to land in the marketplace, descending from the great heights of innovation. In order to be meaningfully applied, in order for it to realise its potential, this technology presumes a comprehensive knowledge of its medium. For the successful implementation of our interactive audio streaming, with which all types of browser supported applications in intra and Internet can be made audio interactive, fundamental knowledge is required of an area that, as such, doesn't presently exist: Interactive Audio Design. The multimedia industry has to date hardly made any efforts to develop the medium of audio. This presents us with the problem that, to all intents and purposes, we need to create a new profession before we can confidently place our product in the hands of our sales, and above all, integration partners. We find ourselves in an embarrassing situation: We intend to deliver the technology for a new sound dimension for the Internet, however are not able to place anyone on the corresponding mixing desk. Up to now audio has not been a priority for multimedia designers. In fact they have shown themselves to be satisfied with the existing dominance of image and text on the Internet. The multimedia industry has managed to reintroduce the silent film by way of the Internet. In contrast, we seek to discover the sound button for the next era of the Internet. For this purpose we developed together with the Fraunhofer Institute in Ilmenau guidelines for interactive audio interface design. Here I would like to praise the excellent diploma thesis written by Björn Ullrich under Prof. Heidi Krömker, Entwicklung und Evaluation von Audio Design Guidelines für eine interaktive Streaming-Software. We have just translated this text into Japanese and it will help us a great deal to teach web-designers to come up with compelling audio interfaces.

But what exactly is Interactive Audio Streaming? As mentioned above, the streaming technology presently employed on the Internet functions like radio and television transmissions, according to the simple sender–receiver principle. Pre-produced files are streamed to the user computer in a monotonous fashion, without any further interaction from the server. Our procedure can extract a great deal more. It builds an uninterrupted connection between the client and the server. We call it a navigated streaming loop. The system's intelligence and therefore its complexity are entirely situated on the server side. All that is required on the client side is a tiny player (1-200 KB, depending on the OS of the client device) that is transmitted at the tip of the stream and, when the user allows, is automatically installed. Or we use other applications as the player for the interactive stream, most importantly Flash.

Fig. 2 Interactive iSound© Streaming



Client
Browser & Small Player (20-200 KB)
WIN CE Explorer, ActiveX Player, Symbian OS, e.a.
Input
User Navigation by Javascript (start, stop, etc.)
Streaming Loop
Output
Individual interactive audio stream (e.g. mp3): Serversided mixing of up to 5 different sources at the same time
Server
Audio sources:
Music
Speech
Radio/ Podcast
Instruments
Video
Text-to-speech
Sounds
Artificial Intelligence
Microphone (e.g. Callcenter)
(Decoder)
Mixer
Encoder
Streamer
iSound

Instead of the simple turning on and off of a requested stream file, the client-server system establishes a permanently navigable circuit which can be controlled by the cursor, i.e. movements of the user's mouse pointer. The stream has a new quality as the user continually controls and modulates its configuration. And suddenly each user has his individual stream.

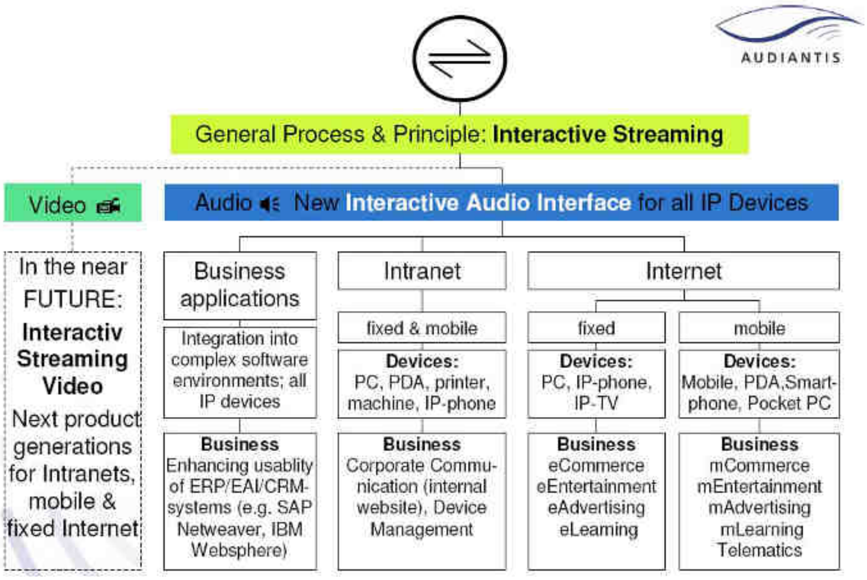
For the first time users can personalize their streams because language, gender of the speaker, background music, help functions etc. are just different data sources that be mixed interactively – that means anytime during the streaming session – and in real-time into the stream. Besides, bandwidth is not an issue. iSound© streaming can use any codec and even switch dynamically between them depending on the client and the available bandwidth.

In the standard scenario of an audio-interactive website the user controls the composition of the interactive stream by means of mouse-over commands of the cursor. The result is a reduction of the high communication inhibition threshold built up by the existing "Point & Click" GUI paradigm. The requirement that every interaction be initiated by a mouse click continually places the user in a decision-making situation. Click actions have to an extent a binding character. Whoever clicks on a banner or a link either leaves the site or at the least expresses their agreement with the opening of a new browser window covering the previous one. Whoever clicks allows data to be transferred to their computer in the case of a download or even permits money to be transferred from their account. Whoever clicks may have even signed a contract as in the case of online auctions or the acceptance of licence conditions. Click actions quite frequently initiate interactions with a high degree of commitment and with far reaching consequences. This is without doubt an intended and useful facility as the Internet is also there for the completion of valuable and legally validated transactions.

However a freer, less binding form of movement for visitors to websites is missing. As the interactive stream is continuously available – especially in the case of mobile Internet where it is often started long before the page has configured – and audio events can be initiated with a latency of on average 1 second, we have decided in favour of the mouse-over approach. Thus we can create very smooth audio interfaces that seamlessly switch and overlay without disruption various audio sources. We only understood very lately that this is indeed the first step beyond the Broadcasting Principle. Initially we just wanted to put some intelligent sound on websites. For this purpose we invented the process of "real-time mixing of signals within the source", the very heart of our iSound© server and central claim of our patents. Then we progressively discovered how much and what kind of content we can mix into the stream, namely pre-recorded, live and dynamically created sound like the one from text-to-speech software. At the next level we saw that the composition of the stream does not necessarily need to be controlled by a cursor. Why not giving the user's GPS data the control over the interactive stream? And now there is no limit any more. How about metabolic data like blood pressure or temperature? How about financial data from stock exchange? Most recently the CEO of the Japanese company Weathernews suggested an iSound© based internet radio project that is directly connected to earthquake early warning systems.

All those applications have become possible because we have opened the classical stream that was imprisoned between a simple storage medium on the server-side and a proprietary plug-in on the client-side. Client and server are now open variables on both ends of the stream. On the server-side we can mix any data content – not only audio but also e. g. commands to trigger events on the client – into each individual stream. On the client-side we can choose existing or design new controls for the interactive stream. Thus we can now define what Beyond Broadcasting means: the receiver has in principle unlimited and permanent control over the content transmitted by the sender. Interactive Audio Streaming with iSound© is, as far as we know, the first technology that is governed by this new paradigm. And the vast field of applications indicates that there is something fundamentally new about this process that we only grasped very slowly. As the iSound© server is codec and client agnostic, furthermore the core is extremely small with 430 KB programmed in standard C++. Thus it can be easily integrated in any soft- and hardware environment.

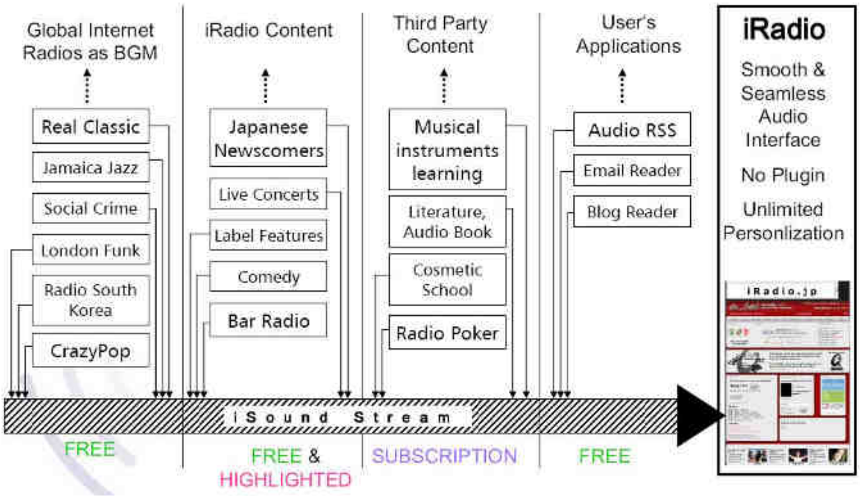
Fig. 3 Fields of Application of Interactive Streaming



AUDIANTIS
General Process & Principle: Interactive Streaming
Video
Audio New Interactive Audio Interface for all IP Devices
In the near FUTURE:
Interactiv Streaming Video
Next product generations for Intranets, mobile & fixed Internet
Business applications
Integration into complex software environments; all IP devices
Business
Enhancing usablity of ERP/EAI/CRM-systems (e.g. SAP Netweaver, IBM Websphere)
Intranet
fixed & mobile
Devices:
PC, PDA, printer, machine, IP-phone
Business
Corporate Communication (internal website), Device Management
Internet
fixed
Devices:
PC, IP-phone, IP-TV
Business
eCommerce
eEntertainment
eAdvertising
eLearning
mobile
Devices:
Mobile, PDA,Smart-phone, Pocket PC
Business
mCommerce
mEntertainment
mAdvertising
mLearning
Telematics

Of course we are looking forward to apply this principle to video streaming. But there is a lot of R&D ahead of us. In the meantime we will start as many new audio applications as possible in order to learn more about this new paradigm. One of our new and exciting experiments is an internet radio based on iSound©. It is our Japanese iRadio Project that will start this year on www.iradio.jp. Motorola and Apple had an iRadio project until the dropped their alliance in 2005. We also target like they did the mobile phones. And it will be much easier for us to get access to the devices as we don't need huge plug-ins.

Fig. 4 Internet Radio Project with iSound Streaming: www.iRadio.jp



At the first level we want to aggregate selected (or potentially all) internet radios similarly to Shoutcast. We want to use them as background music and eventually feature or even promote them. The second level is the dedicated Japanese content for independent music where we will feature new artists and bands. On the third level we want to over vocal content on any kind. We will provide an open API so that third parties can offer their content through our iRadio. The fourth level is where my own contents are mixed into the radio stream, e.g. audio RSS or e-mail reader. The visual interface will not just be a website but at the same time the iRadio itself. The user or rather the listener can personalize and schedule the radio programme by navigating through a liquid and smooth audio-interface. We believe that this will be a most compelling user experience. And we believe that it prepares a huge and now technology based leap of digital media in a new era Beyond Broadcasting.

Johnsen, Gisle

Interactive composing for young students – The Music Delta system

It is a fact that young people today spend most of their time listening to music and surfing on the Internet. At the same time there is a great need in the education system to encourage the student's creativity. It is important that the education system use this fact and incorporate the use of Internet and music in the education of kids and young people.

Grieg Music Education (GME) has developed a system that enhance the creativity of kids and young people by using Internet as a communication system and music as a tool for the students to be creative.

The GME system called Music Delta, is the first system worldwide that use all the possibilities that you may find in a Learning Management System (LMS). Music Delta is delivered as a teaching package through the schools LMS. It opens directly in the LMS and make use of all the tools within the LMS. The administrator give the students access to what he / she wants them to use within Music Delta. In addition the students may upload whatever text, picture, video, music or sound they have in their own portfolio and use their owm materials directly in Music Delta. All their work within Music Delta will also be saved in their portfolio in the schools LMS. GME is the first company that have developed educational tools and packages build on the AICC and SCORM standards.

Learning Management Systems

The use of Learning Management Systems is a new and modern way for the education system to organize the students work and many countries now incorporate the use of Learning Management Systems (LMS) in the education. In Norway the kids now start using an LMS at the age of six.

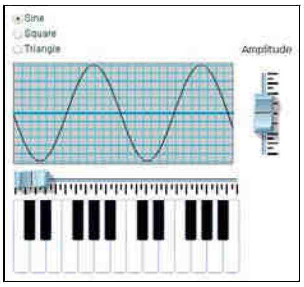
The Learning Management Systems also open up for digital and interactive teaching programs. The publishing companies have started developing digital resources to be used instead of ordinary books. Anyway, most of the digital resources used today, are only traditional books presented on the Internet, containing only text and pictures. And they do not enhance the kids creativity. Grieg Music Education's system, Music Delta, really enhance the creativity of kids and young people by using Internet as a communication system and music as a tool for the students to be creative. Music Delta, make it possible for kids and young people to work and collaborate with music across classrooms and homes and across borders of different kinds. In MusicDelta students can create music, share music ideas, viewpoints and recorded music in addition to mix their own productions with professional productions. MusicDelta enable students from different countries to work online, sharing and composing music together. They may publish their new music on their own ArtBlog. Music Delta – AN INTERACTIVE ROOM FOR KIDS AND YOUNG PEOPLE Music Delta is an interactive room where the users log in from any computer. They all have their own profile with a personal username and password.

Within Music Delta the users can meet other kids from the school, the city, the country or from the whole world, depending on what the teacher or school decide. In this way they may invite other kids to collaborate, creating new music. Creating music is one of the main purposes in Music Delta. Here the users find many interactive tools, in which they may create their own music. After producing their own music, the users may publish the music or music video on their own blog within Music Delta. If the school permits it, they may also publish their new compositions on their own webside, on Myspace etc.

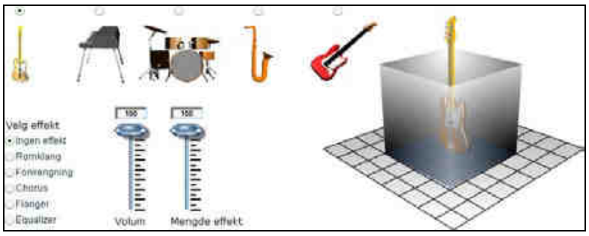
One important aspect is that Music Delta opens for collaboration between students from different parts of the world.

A brief introduction to the interactive tools in the Music Delta

Learn about Sound: This is an interactive tool that let the students learn how a sound is created and how to manipulate the sound. The students can try how the theories work by using the interactive tools.

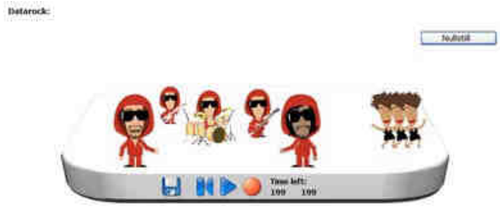


Manipulating the sound: This is a tool that learn the students how to manipulate the sound played by instruments by the use of digital effects like Chorus, Reverb, Distortion, Flanger and Equalization

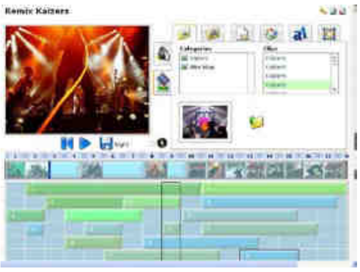


Composition: In Music Delta the students can read about composition and how to create their own music, and after having read the theories, they can hear and see famous composers from different parts of the world, talk about how they work when composing. Music Delta has several webcasts with popular artists, who talk about how they work with music.

This is a picture from the webcast with the song writer from the Spanish band Ojos de Brujo, who tells about how he is creating music. Then the students can put together their own favourite band on their personal WebStage. In Music Delta they get access to music clips from real artists that they can put together to create their own music production. In the WebStage they can change the Volume and Panning.

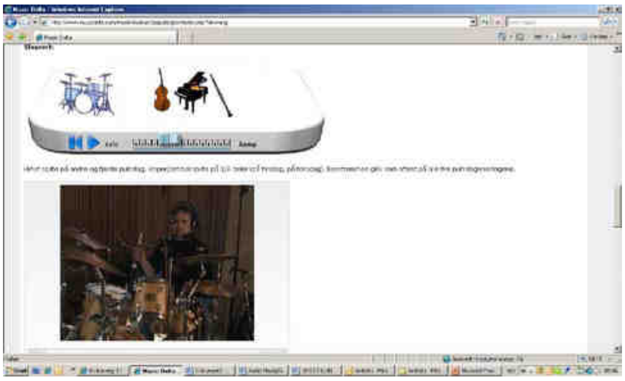


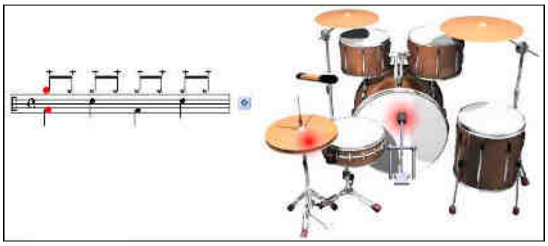
After this, the students can start composing their own music by the help of sound clips from original artists, or they can upload files with music they have created themselves. In the same way, they can put together their own music video by using professional pictures or uploading their own pictures.



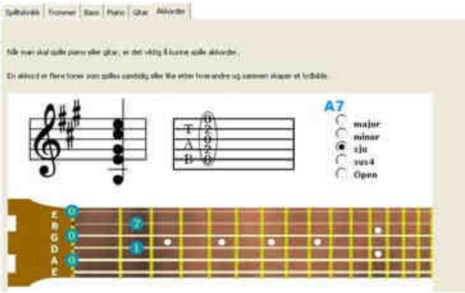
In Music Delta, the students can create their own animation with the help of moveable figures from the Henrik Ibsen play: Peer Gynt. This is an in an interactive tool, where professional actors have recorded parts from Peer Gynt. The students can access Peer Gynt music from different parts of the world, and put together their own Peer Gynt music. The students can actually put together a completely multimedia production, controlling the text, music, sound effects, backgrounds and moveable figures. The students can publish the result as an animation with pictures, sound and music in their own ArtsBlog.

Performing: In the performing part, the students get access to interactive tools, learning them how to play different instruments. All the interactive lessons are presented as interactive, internet based notation. In addition the students can watch animated instruments playing along with the notation.

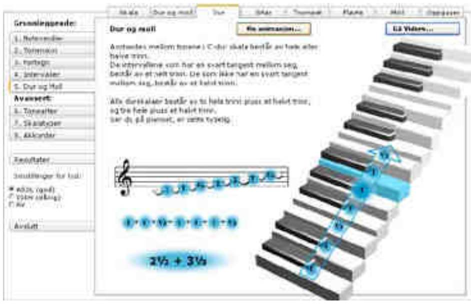




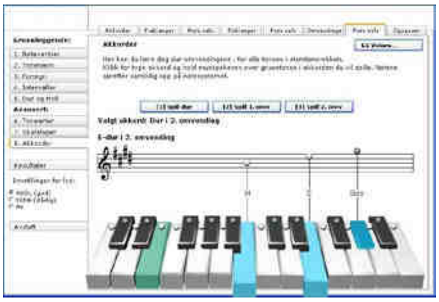
Music Delta contains interactive chord finders, that the students can actually play on, by moving the mouse over the strings or keys.



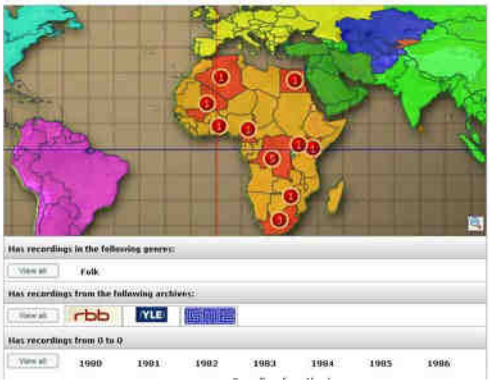
Music Delta teaches the students to read and write music. The tool is interactive and the students get immediately response. Music Delta has also an advanced part, where the students can



learn advanced music theory. This part is also interactive, which means that the students can test their own skills at all levels. *The music and art of the world:* In Music Delta the students



get access to new tools, helping them to learn all about the music and arts of the world. This particular tool contains a map combined with a time line.The map gives the students access to music from all the world, articles about music from all the world and videoclips showing arists and instruments from all the world. The map helps the student to find and learn about music styles, artists, composers, music from different parts of the world and historical periods.
The Time line helps the students accessing important facts from



Has recordings in the following genres:

View all Folk

Has recordings from the following archives:

View all rbb YLE

Has recordings from 0 to 0

View all 1980 1981 1982 1983 1984 1985 1986

the history of music and art. The map also allows the students to listen to music from different composers and artists.