3D GRAPHICS TOOLS FOR SOUND COLLECTIONS

George Tzanetakis

Computer Science Department
Princeton Univerisy
gtzan@cs.princeton.edu

Perry Cook

Computer Science Department and Music Department
Princeton University
prc@cs.princeton.edu

ABSTRACT

Most of the current tools for working with sound work on single soundfiles, use 2D graphics and offer limited interaction to the user. In this paper we describe a set of tools for working with collections of sounds that are based on interactive 3D graphics. These tools form two families: sound analysis visualization displays and model-based controllers for sound synthesis algorithms.

We describe the general techniques we have used to develop these tools and give specific case studies from each family. Several collections of sounds were used for development and evaluation. These are: a set of musical instrument tones, a set of sound effects, a set of FM radio audio clips belonging to several music genres, and a set of mp3 rock song snippets.

1. INTRODUCTION

There are many existing tools for sound analysis and synthesis. Typically these tools work on single soundfiles and use 2D graphics. Some attempts at using 3D graphics have been made (like Waterfall spectogram displays) but typically they offer very limited interaction to the user.

In this paper we describe some recently developed tools that are based on interactive 3D graphics. These tools can be grouped into two families. The first family consists of visualization displays for sound analysis. The second family consists of model-based controllers for sound synthesis algorithms. The primary motivation behind this work has been using 3D graphics for experimenting with large collections of sounds rather than single audio files.

Each tool is parametrized by its inputs and options. We have tried to decouple the tools from a specific analysis front-end and synthesis back-end. This is achieved through the use of different mappings of analysis or synthesis data to the tool inputs and a generic client-server architecture. That way other researchers can easily integrate these 3D graphics tools with their applications. The tool options control the visual appearance. Collections of music instrument tones, sound effects, FM radio audio clips belonging to several music genres, and rock mp3 song snippets were used for developing and evaluating these tools.

An additional motivation for the development of these tools is that recent hardware is fast enough to be able to handle audio analysis or synthesis and 3D graphics rendering at real time. All the described tools run on commodity PCs without any special-purpose hardware for 3D Graphics or Signal Processing.

2. SOUND ANALYSIS VISUALIZATION DISPLAYS

Visualization techniques have been used in many scientific domains. They take advantage of the strong pattern recognition abilities of the human visual system to reveal similarities, patterns and correlations both in space and time. Visualization is more suited for areas that are exploratory in nature and where there are large amounts of data to be analyzed like sound analysis research. Sound analysis is a relatively recent area of research. It refers in general to any technique that helps a computer "understand" sound in a similar way that a human does. Some examples of sound analysis are classification, segmentation, retrieval, and clustering. The use of 3D graphics and animation allows for display of more information than traditional static 2D displays.

Sound analysis is typically based on the calculation of feature vectors that describe the spectral content of the sound. There are two approaces for representing a sound file. It can be represented as a single feature vector (i.e a point in the high dimensional feature space) or a time series of feature vectors (i.e a trajectory).

As our focus is the development of tools and not the optimum feature front-end we are not going to describe in detail the specific features we have used. Moreover an important motivation for building these tools is exploring and evaluating the many possible features that are available. For more details about the supported features refer to [1]. The features used include means and variances of Spectral Centroid (center of mass of spectrum, a correlate of brightness), Spectral Flux (a measure of the time variation of the signal), Spectrol Rolloff (a measure of the shape of the spectrum) and RMS energy (a correlate of loudness). Other features are Linear Prediction Coefficients (LPC) and Mel Frequency Cepstral Coefficients (MFCC) [2], which are perceptually motivated features used in speech processing and recognition. In addition, features directly computed from MPEG compressed audio can be used [3, 4]. Information about the dynamic range of the features is obtained by statistically analysing the dataset of interest.

Principal components analysis (PCA) is a technique for reducing a high dimensional feature space to a lower dimensional one retaining as much information from the original set as possible. For more details refer to [5]. We use PCA to map a high dimensional set of features into a lower dimensional set of tool inputs. The extraction of a principal component amounts to a variance maximizing rotation of the original variable space. In other words, the first principal component is the axis passing through the centroid of the feature vectors that has the maximum variance therefore explains a large part of the underlying feature structure. The next principal component tries to maximize the variance not explained by the first. In this manner, consecutive orthogonal components are extracted.

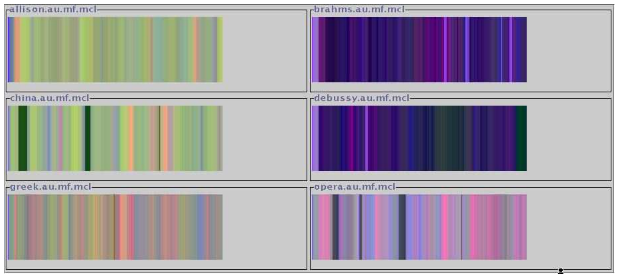

Figure 1: *Timbregrams for speech and classical music (RGB)*

Clustering refers to grouping vectors to clusters based on their similarity. For automated clustering we have used the c-means algorithm [6]. In many cases the resulting clusters can be perceptually justified. For example using 12 clusters for the sound effects dataset of 150 sounds yields reasonable clusters like a cluster with almost all the walking sounds and a cluster with most of the scraping sounds. Using visualization techniques we can explore different parameters and alternative clustering techniques in an easy and interactive way.

2.1. Case studies

- **TimbreGram** is a tool for visualizing sounds or collections of sounds where the feature vectors form a trajectory. It consists of a series of vertical color stripes where each stripe corresponds to a feature vector. Time is mapped from left to right. There are three tool inputs corresponding to the three color components for each stripe. The *TimbreGram* reveals sounds that are similar by color similarity and time periodicity. For example the *TimbreGrams* of two walking sounds with different number of footsteps will have the same periodicity in color. PCA can be used to reduce the feature space to the three color inputs. The current options are RGB and HSV space colormapping.

  Fig. 1 shows the *Timbregrams* of six sound files (each 30 seconds long). Three of them contain speech and three contain classical music. Although color information is lost in the greyscale of the paper, music and speech separate clearly. The bottom right sound file (opera) is light purple and the speech segments are light green. Light and bright colors typically correspond to speech or singing (Fig. 1 left column). Purple and blue colors typically correspond to classical music (Fig. 1 right column). Of course, the mapping of features to colors depends on the specific training set used for the PCA analysis.

- **TimbreSpace** is a tool for visualizing sound collections where there is a single feature vector for each sound. Each sound (feature vector) is represented as a single point in a 3D space. PCA can be used to reduce the higher dimensional feature space to the input x, y, z coordinates. The *TimbreSpace* reveals similarity and clustering of sounds. In addition coloring of the points based on automatic clustering and bounding box selection is supported. Typical graphic operators like zooming, rotating and scaling can be used to interact with the data. A list of the nearest points (sounds) to the mouse cursor in 3D is provided and by clicking at the appropriate entry the user can hear the sound.

Figure 2: *GenreGram showing male speaker and sports*

  Sometimes it is desirable to automatically hear the sound points without having to select them individually. We are exploring the use of principal curves [7] to to move sequentially through this space.

- **TimbreBall** is a tool for visualizing sounds where the feature vectors form a trajectory. This is a real-time animated visualization where each feature vector is mapped to the x, y, z coordinate of a small ball inside a cube. The ball moves in the space as the sound is playing following the corresponding feature vectors. Texture changes are easily visible by abrupt jumps in the trajectory of the ball. A shadow is provided for better depth perception (see Fig. 3).

- **GenreGram** is a dynamic real-time audio display targeted towards radio signals. The live radio audio signal is analysed in real-time and is classified into 12 categories: *Male, Female, Sports, Classical, Country, Disco, HipHop, Fuzak, Jazz, Rock, Silence* and *Static*. The classification is done using statistical pattern recognition classifiers trained on the collected FM-radio dataset. For each of these categories a confidence measure ,ranging from 0.0 to 1.0, is calculated and used to move up or down cylinders corresponding to each category. Each cylinder is texture-mapped with a representative image of each category. The movement is also weighted by a separate classification decision about if the signal is *Music* or *Speech*. *Male, Female*, and *Sports* are the *Speech* categories and the remaining are music. The classification accuracy is about 90% for the *Music/Speech* decision, 80% for the *Speech* categories, and 45% for the *Music* categories. The focus is the development of the display tool therefore the classification results are not a full scale evaluation and are provided only as a reference.

  In addition to being a nice demonstration tool for real-time automatic audio classification, the *GenreGram* gives valuable feedback both to the user and the algorithm designer. Different classification decisions and their relative strengths are combined visually revealing correlations and classification patterns. Since the boundaries between music genres are fuzzy, a display like this is more informative than single classification decision. For example, most of the times a rap song will trigger *Male Speech*, *Sports* and *HipHop*. This exact case is shown in Fig. 2.

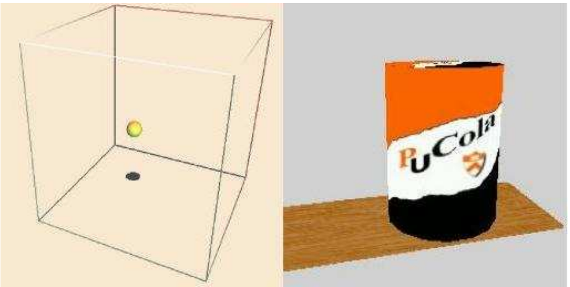

Figure 3: *TimbreBall* and *PuCola Can*

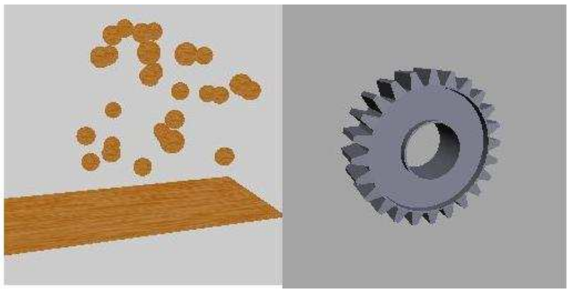

Figure 4: *Gear* and *Particles*

3. MODEL-BASED CONTROLLERS FOR SOUND SYNTHESIS

The goal of virtual reality systems is to provide the user the illusion of a real world. Although graphics realism has improved a lot, sound is very different from what we experience in the real world. One of the main reasons is that unlike the real world, objects in a virtual world do not make sounds as we interact with them. The use of pre-recorded PCM sounds does not allow the type of parametric interaction necessary for real world sounds. Therefore 3D models that not only look similar to their real world counterparts but also sound similar are very important.

A number of simple interfaces have been written in Java3D, with the common theme of providing a user- controlled animated object connected directly to the parameters of the synthesis algorithm. This work is part of the Physically Oriented Library of Interactive Sound Effects (PhOLISE) project [8]. This project uses physical and physically-motivated analysis and synthesis algorithms such as modal synthesis, banded waveguides [9], and stochastic particle models [10], to provide interactive parametric models of real-world sound effects.

Modal synthesis is used for sound generating objects/systems which can be characterized by a relatively few resonant modes, and which are excited by impulsive sources (striking, slamming, tapping, etc.). Banded waveguides are a hybridization of modal synthesis and waveguide synthesis [11], and are good for modeling modal systems which can also be excited by rubbing, scrapping, bowing, etc. Stochastic particle model synthesis (PhISEM, Physically Inspired Stochastic Event Modeling) is applicable for sounds caused by systems which can be characterized by interactions of many independent objects, such as ice cubes in a glass, leaves/gravel under walking feet, loose change in a pocket, etc.

These programs are intended to be demonstration examples of how sound designers in virtual reality, augmented reality, games, and movie production can use parametric synthesis in creating believable real-time interactive sonic worlds.

3.1. Case studies

- **PuCola Can** shown in Fig. 3 is a 3D model of a soda can that is slid across various surface textures. The input parameters of the tool are the sliding speed and texture material and result in the appropriate changes to the sound in real time.
- **Gear** is a 3D model of a gear that is rotated using a slider. The corresponding sound is a real-time parametric synthesis of the sound of a turning wrench.
- **Feet** is a 3D model (still under development) of feet stepping on a surface. The main input parameters are gait, heel/toe timing, heel and toe material, weight and surface material. A parametric physical modling of human walking then drives both the graphics and synthesis models.
- **Particles** is a 3D particle model of particles (rocks, ice cubes, etc) falling on a surface. The density, speed and material input parameters are driving both the 3D model and the corresponding physical modelling synthesis algorithms.

4. IMPLEMENTATION-DATASETS

The sound analysis is performed using MARSYAS [1] an object-oriented framework for audio analysis. The sound synthesis is performed using the Synthesis Toolkit [12]. The developed tools written in Java 3D act as clients to the analysis and synthesis servers. The software has been tested on Linux, Solaris, Irix and Windows (95,98,NT) systems.

Four datasets were used as the basis for building and testing our tools. The first dataset consists of 913 isolated musical instrument notes of bowed, brass and woodwind instruments recorded from the McGill MUMS CDs. The second dataset consists of 150 isolated contact sounds collected from various sound effects libraries. This set of 150 sound effects is composed of everyday “interaction” sounds. By interaction we mean sounds that are caused and changed directly by our gestures, such as hitting, scraping, rubbing, walking, etc. They range in size from 0.4 to 5 seconds, and are the focus of two other projects, one on the synthesis of interaction sounds, and one on the perception of such sounds. The remaining two datasets are 120 FM radio audio clips, representing a variety of music genres, and 1000 rock song snippets (30 sec long) in mp3 format.

These datasets were used to build the sound analysis visualization tools. Using realistic data sets is the only way to debug and evaluate visualization tools. In addition to feature extraction techniques, clustering and PCA require a training data set in order to work. Evaluating visualization techniques is difficult and we have not performed any direct user studies on the visualization systems (although we have performed segmentation and similarity retrieval studies on the datasets). We have found that the tools make perceptual sense and in some cases have provided us with insights about the nature of the data and the analysis techniques we can use.

5. FUTURE WORK

The main directions for future work are the development of alternative sound-analysis visualization tools and the development of new model-based controllers for sound synthesis based on the same principles. Of course new analysis and synthesis algorithms will require adjustment of our existing tools and development of new ones. More formal evaluations of the developed tools are planned for the future. Integrating the model-based controllers into a real virtual environment is another direction of future work.

6. ACKNOWLEDGEMENTS

This work was funded under NSF grant 9984087 and from gifts from Intel and Arial Foundation.

7. REFERENCES

[1] G. Tzanetakis and P. Cook, "Marsyas: A framework for audio analysis," *Organised Sound*, 2000, (to appear).

[2] M. Hunt, M. Lennig, and P. Mermelstein, "Experiments in syllable-based recognition of continuous speech," in *Proc. 1996 ICASSP*, 1980, pp. 880–883.

[3] D. Pye, "Content-based methods for the management of digital music," in *Proc.Int.Conf on Audio, Speech and Signal Processing, ICASSP*, 2000.

[4] G. Tzanetakis and P. Cook, "Sound analysis using mpeg compressed audio," in *Proc.1999 IEEE Int.conf. on Audio, Speech and Signal Processing ICASSP00*, Istanbul, 2000.

[5] L.T.Jolliffe, *Principal Component Analysis*, Springer-Verlag, 1986.

[6] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, 1973.

[7] T. Hermann, P. Meinicke, and H. Ritter, "Principal curve sonification," in *Proc. Int. Conf on Auditory Display, ICAD 2000*.

[8] P. Cook, "Toward physically-informed parametric synthesis of sound effects," in *Proc.1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA99*, New Paltz, NY, 1999, Invited Keynote Address.

[9] G. Essl and P. Cook, "Measurements and efficient simulations of bowed bars," *Journal of Acoustical Society of America (JASA)*, vol. 108, no. 1, pp. 379–388, 2000.

[10] P. Cook, "Physically inspired sonic modeling (phism): Synthesis of percussive sounds," *Computer Music Journal*, vol. 21, no. 3, September 1997.

[11] J.O. Smith, "Acoustic modeling using digital waveguides," in *Musical Signal Processing*, C. Roads, S. T. Pope, A. Piccialli, and G. De Poli, Eds., pp. 221–263. Netherlands: Swets and Zietlinger, 1997.

[12] P. Cook and G. Scavone, "The synthesis toolkit (stk), version 2.1," in *Proc.1999 Int.Computer Music Conference ICMC*, Beijing, China, October, 1999.