A Visualization of Music

Sean M. Smith
Department of Computer Science
Texas A&M University

Glen N. Williams, Ph.D.
Department of Computer Science
Texas A&M University

INTRODUCTION

Currently, the most popular method of visualizing music is music notation. Through music notation, an experienced musician can gain an impression of how a particular piece of music sounds simply by looking at the notes on paper. However, most listeners are unfamiliar or uncomfortable with the complex nature of music notation. The goal of this project is to present an alternate method for visualizing music that makes use of color and three-dimensional space.

PROCESSING MUSIC DATA

For an effective visualization of music, the data used must reflect upon the information found in an actual score. MIDI (Musical Instrument Digital Interface) data files, which contain this information, are utilized for this project. A MIDI data file only stores the control information for a musical composition as opposed to the sounds produced by a performance of the music. This is similar in nature to the manuscript notation (or sheet music) commonly used by musicians.

The program is designed to parse input MIDI data files and generate graphical representations based on this information. These representations are dependent on a mapping function described in the next section. The code is written in ANSI C and makes use of Open GL.

MAPPING MUSIC DATA TO 3-D SPACE

Defining a Mapping Function

The characteristics of music are used to define a mapping function. This function transforms input music data into three-dimensional graphical representations. The mapping process is divided into two steps. The first step defines a mapping function for individual tones, while the second step specifies a mapping function for instruments within a particular orchestration.

Mapping Tone Data

Tones generated by instruments are represented by colored spheres in the visualization. The characteristics of each sphere are partially dependent on three properties that describe musical tones--pitch, volume, and timbre.

The pitch (or note value in MIDI terms) of a particular tone defines the "highness" or "lowness" of that note. Every musical instrument plays notes within their limited pitch range. The minimum and maximum values of this range are typically instrument dependent. Based on

Proceedings of the 8th IEEE Visualization '97 Conference
1070-2385/97 $10.00 © 1997 IEEE

the MIDI standard, pitches have values restricted to the range [0, 127]. The range of pitches for each instrument is mapped to locations along the vertical y-axis.

Because music seldom makes use of the entire range of pitches, the lower and upper pitch ranges may not be utilized. To compensate for this, the program determines the minimum and maximum pitch values and scales the vertical axis accordingly. Consequently, the minimum and maximum values displayed represent the minimum and maximum pitch values for that particular piece of music.

The volume (or velocity in MIDI terms) of a single tone designates the loudness or softness of the tone. In the visualization sequence, the volume of a note is represented by the relative size of each displayed sphere. Each sphere's radius is restricted to the range [0, 127]; this corresponds to the minimum and maximum MIDI volume values for a tone.

Timbre is the most difficult characteristic of a tone to describe. It is often referred to as the "color" of a tone produced by a musical instrument. The timbre of a tone is dependent on many factors. Based upon the General MIDI (GM) specification, instruments with similar (but not necessarily identical) timbres are divided into instrument groups. Color is used to show distinctions between each instrument group. These colors are selected on the basis of visibility during the visualization.

Mapping Instrument Data

An individual instrument plays a sequence of tones with specified rhythms throughout a piece of music. The tones produced by an instrument are visualized as a sequence of spheres of various shapes and sizes. (The collection of all sequences of spheres represents the entire musical score.)

For this visualization, the individual instruments are mapped to particular values along the horizontal x-axis. The program is written to displace the instruments evenly along this axis. The maximum allowable radius of each sphere (which represents the maximum volume level of a note) is determined by this spacing. In other words, the volume range [0, 127] is mapped to the range [0, r] where r is the largest possible sphere radius. As a result, the size of a note at maximum volume is dependent on the number of instruments performing a piece of music. To compensate for this, a global scale factor is applied to each sphere's radius. Without a scale factor, the notes in a visualization sequence with many instruments would be too small to view.

Rhythm refers to the flow of the music based on time. In a musical score, each instrument plays tones with according to a predetermined rhythm. In other words, every note occurs at a certain point in time and lasts for a specified length of time. As the spheres are generated in the visualization, they are placed in locations along the time axis (or z-axis) according to their specified start time. The spheres remain on the screen for their entire duration. As a result, tones that are audible for longer periods in the actual music remain visible for longer periods of time in the visualization.

DESCRIPTION OF VISUALIZATION SEQUENCE

Selection of Music

The first piece of music selected is an arrangement of the first movement from *Octet in F Major, Op. 166 (D.803) for Strings and Winds* which was composed by Franz Schubert. The orchestration consists of a clarinet, a bassoon, an F horn, two violins, a viola, a cello, and a contrabass. Both the tempo and time signature remain constant throughout the piece of music. The second selection is *Tanz, Uf Dem Anger* from Carmina Burana which was composed by Carl

Proceedings of the 8th IEEE Visualization '97 Conference
1070-2385/97 $10.00 © 1997 IEEE

Orff. The particular arrangement used is based on a concert band version written by John Krance. This piece was selected to demonstrate the program's ability to adapt to changes in tempo and time signature.

Frame Generation

In order to create the visualization, frames are generated by the program and then stored as image files in RGB format. In order to synchronize the visualization with the music, the interval between successive frames is calculated by the program. After a single frame is generated, the frame generation rate is recalculated due to the possibility of a change in tempo within the music.

Soundtrack Generation

The music for both arrangements is performed using the Cakewalk (Version 3.0) sequencing program. The instrument sounds used in both visualizations are generated with a Yamaha QY-300 sequencer and a Roland Sound Canvas sound module. The actual sounds produced by these two devices are recorded and mixed on an SGI Indy workstation.

Coordinate Axes

Coordinate axes are located in the background throughout both visualizations. Each axis represents a different parameter in the visualization. The red axis represents pitch, the blue axis represents instrument, and the green axis represents time. The axes rotate relative to the rotations performed in the visualized scene. The origin of the axes is located at the "middle C" pitch value.

Note History

As each tone ends, a history marker is used to represent its location and duration. The markers are represented by small (relative to the actual note representations) colored spheres. Each marker is scaled along the time axis according to the original tone's duration. In other words, the markers for tones that are held for longer periods appear as ellipsoids, while the markers that represent shorter tones remain sphere-like in appearance. The shorter staccato tones in Orff's *Tanz* all appear to be very similar in length. On the other hand, the differences in note duration can easily be seen in the note history of Schubert's *Octet in F Major*.

The color scheme for the history markers is based upon the color of the original tone. Each marker is assigned the same hue value as the original note, but only a fraction of the original saturation value. Consequently, the markers all appear to be lighter in color than the original tone colors. Also, as each marker "ages", the color intensity level decreases. As a result, the past markers appear to be darker than the more recently generated markers. This effect is used to give a better sense of depth in the visualization.

Legend Information

A legend appears on the right hand side of the screen to assist the viewer in interpreting the visualization. Information such as title and composer, instrument group color key, and axes color key appears only for specified amounts of time. Each of these segments of information is displayed for approximately five seconds. Other information that is dynamic in nature is present at all times in the legend. This information includes current tempo, current time signature, and the current metronome beat value. Due to the dynamic nature of this information, each value is recalculated and updated with each new frame.

Proceedings of the 8th IEEE Visualization '97 Conference
1070-2385/97 $10.00 © 1997 IEEE

CONCLUSION

This paper and the accompanying video describe one method of visualizing music in three-dimensional space. The implementation of this method shows that music visualization is an effective technique; although, it is certainly not the only possible method for accomplishing the task. Throughout the course of this project, several variations and alternative approaches were discussed. The final version of this project reflects the decisions that were made in order to present the best possible representation of music data.

Proceedings of the 8th IEEE Visualization '97 Conference
1070-2385/97 $10.00 © 1997 IEEE

REFERENCES

[AIKI 96] J. Aikin, "Mysteries of MIDI," *Music and Computers*, pp. 75-77, January/February 1996.

[BRAU 94] C. Braut, *The Musicians's Guide to MIDI*. Paris, France: Sybex, 1994.

[DODG 85] C. Dodge and T. A. Jerse, *Computer Music: Synthesis, Composition, and Performance*. New York: Schirmer Books, 1985.

[FOLE 93] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice*, Reading, MA: Addison-Wesley Publishing Company, Inc., 1993.

[HECK 95] J. Heckroth (Crystal Semiconductor Corp.), *Tutorial on MIDI and Music Synthesis*. La Habra, CA: The MIDI Manufacturers Association, 1995.

[HOFF 89] C. R. Hoffer, *The Understanding of Music*. Belmont, CA: Wadsworth Publishing Company, a Division of Wadsworth, Inc., 1989.

[LIPS 89] E. Lipscomb, "How Much for Just the MIDI?", *Benchmarks*, October 1989, (*Benchmarks* is the North Texas Computing Center Newsletter).

[MEYE 91] C. Meyer, "General MIDI (GM) and Roland's GS Standard", *Electronic Musician*, August 1991.

[NEID 93] J. Neider, T. Davis, and M. Woo, *OpenGL Programming Guide*. Reading, MA: Addison-Wesley Publishing Company, Inc. and Silicon Graphics, Inc., 1993.

[PENF 95] R. A. Penfold, *Advanced MIDI User's Guide*, Glasgow, England: PC Publishing, 1995.

Proceedings of the 8th IEEE Visualization '97 Conference
1070-2385/97 $10.00 © 1997 IEEE