Exploring Music Visualization Using MIDI and Java

Nathan Fox, Dr. John Carlis, Department of Computer Science and Engineering

Introduction

Sheet music has been used to represent a musical piece. One problem with sheet music is its difficulty in displaying a wide range of pitches on a single staff. Also, to some, it is unclear what the relationship is between two notes at a glance. Most importantly, sheet music has difficulty illustrating volume changes, and it does not allow for a continuum of pitches.

This research explored displaying notes on a spiral instead of a staff. The spiral uses spokes to represent common tones, and the notes in the music are represented as collections of cylinders along a time axis perpendicular to the spiral [Belifuss et al., 2003].
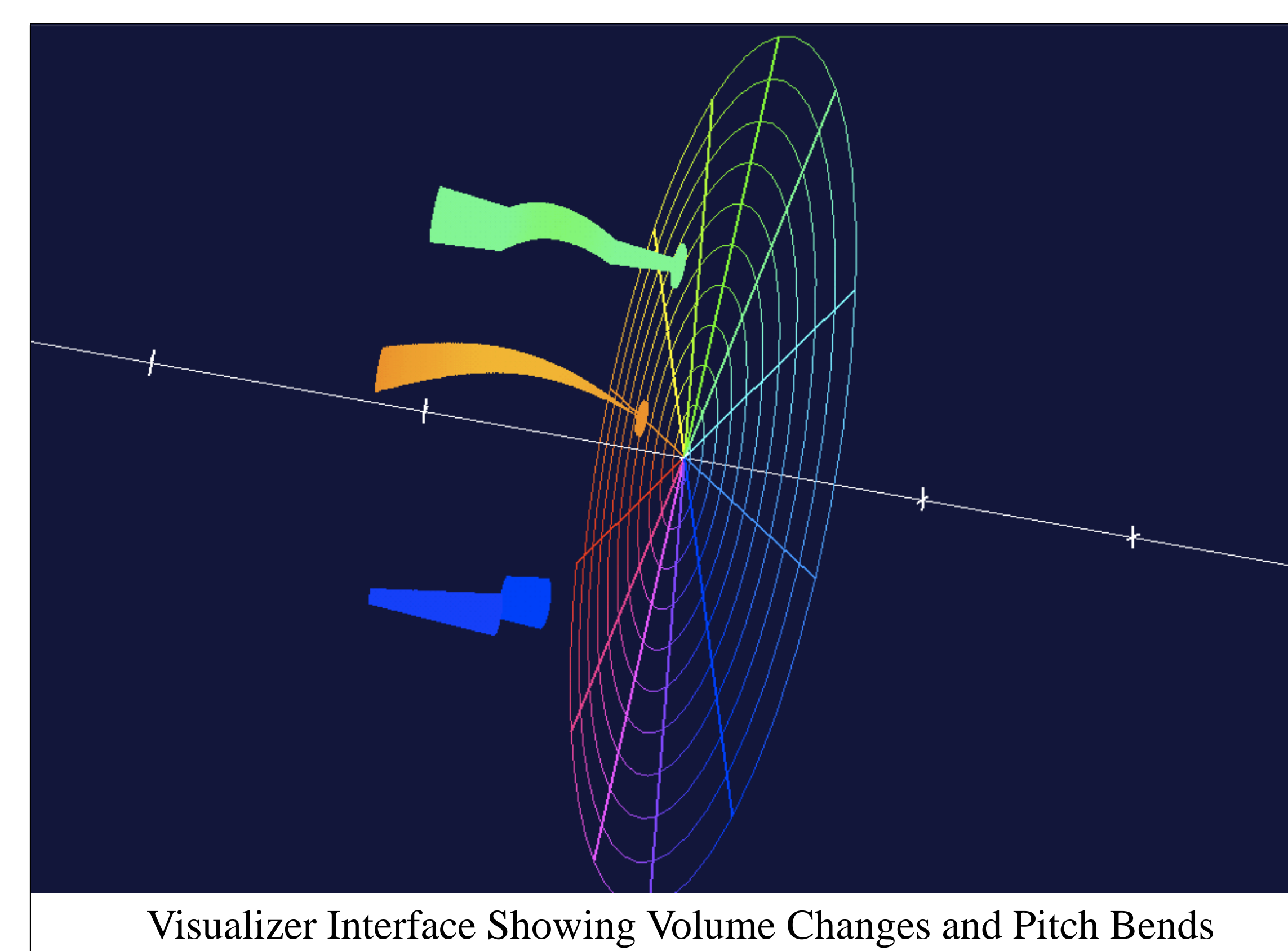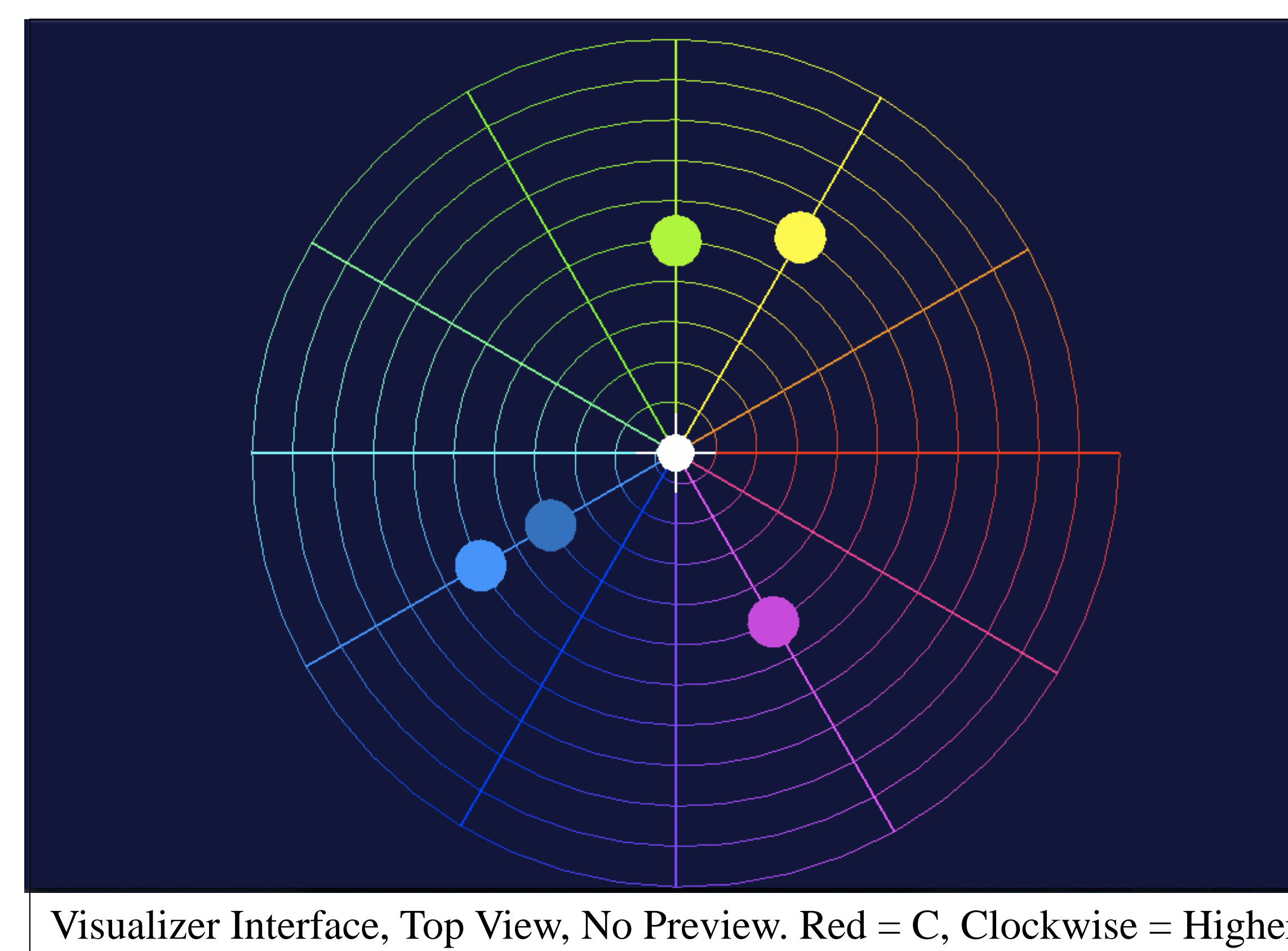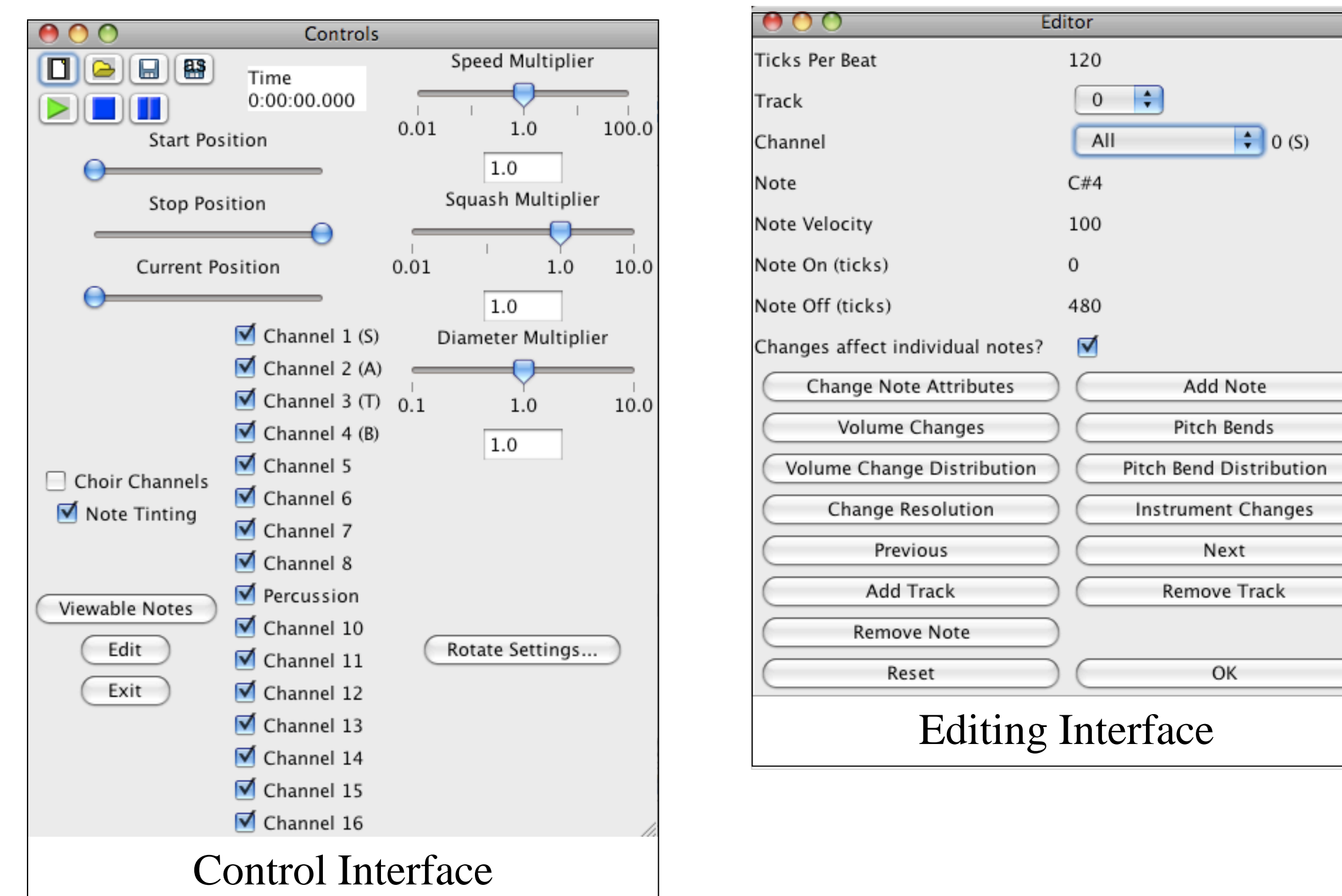
A three-dimensional musical display would be difficult to use effectively when read from a page. Computers, though, can represent such a structure usefully. Hence, we tried to develop a working piece of software based on the spiral display.

Design Decisions

•Java was the language of choice. The new software is based on a similar program written by Dr. Carlis's students in 2003. That was written in C++ and was Windows-specific. The new software is cross-platform, and the intent is to eventually create a web application.

•The software can only work with MIDI files. The original software also had this restriction. We decided to keep it because of the simplicity of the file format and the existence of a powerful MIDI library in the Java API.

•For 3D graphics, we use Java3D. This is a relatively simple suite. Though it is old (1999), its documentation has been kept up-to-date as Java has evolved.

•Over the course of the project, the researcher learned about software development techniques. Hence, much of the oldest code is poorly written and difficult to read, though it did separate the different graphical components into classes. Later code is much more modular, and it takes advantage of an earlier-built abstraction of a MIDI file that was not being used.

Acknowledgements

This research would not have been possible without the assistance of Dr. John Carlis of the Department of Computer Science, the Undergraduate Research Scholarship, and the Undergraduate Research Opportunities Program (UROP).

Control Interface

Editing Interface

Visualizer Interface, Top View, No Preview. Red = C, Clockwise = Higher

Visualizer Interface Showing Volume Changes and Pitch Bends

Software Features

•Spiral display for music with changeable viewing orientation, zoom, and preview time, and
•Notes represented as collections of cylinders
•Tinted notes for easy differentiation of similar notes in different octaves
•Option for coloring voice channels uniformly
•Percussion channel represented on z-axis as white cylinders
•Ability to turn specific channel displays on or off
•Display of volume as note diameter and pitch bends as displacement of notes (not in old software)
•Ability to edit MIDI files (not in old software)
  • Adding and removing tracks and notes
  • Editing volume changes and pitch bends on a per-note basis
  • Changing instruments and resolution

Conclusion

The spiral display makes seeing relationships between notes quite easy, as intervals are simply angles between spokes in the spiral. As a result, this software could eventually be used to compose music. In its current form, though, adding notes is a slow process, so the editing capabilities are best used to tweak existing files at the moment.

A unique feature about this software is the ability to edit volume and pitch for a single note. MIDI ties these changes to a channel, which can contain many notes. Thus, to accomplish this feat, this software "promotes" required notes to new tracks and channels, thereby leaving the notes in the old channel unchanged.

This software is not yet complete. There are still some bugs, and there are plans to add additional features. For example, notes cannot be moved or copied between channels in the current version, and the editor is still somewhat separate from the display. The goal is to have a good, novel piece of software for music visualization and editing.

References

Belifuss, J. and Knatterud, L. 2003. Spirals. *University of Minnesota*.

Chew, E. and François A. 2004. MuSA.RT: Music on the Spiral Array. Real-Time. *University of Southern California*. http://imsc.usc.edu/research/project/musart/musart_nsf8.pdf (Accessed December 19, 2009).

Title: Exploring Music Visualization Using MIDI and Java
Contact information: foxxx340@umn.edu
University of Minnesota Department of Computer Science

UNIVERSITY OF MINNESOTA
Driven to Discover℠