
Version 1.1

David Topper  
Virginia Center for Computer Music  
University of Virginia  
topper@virginia.edu

Juraj Kojs  
Virginia Center for Computer Music  
University of Virginia  
kojs@virginia.edu

ABSTRACT

This paper discusses recent upgrades to VScore: A Real-time Visual Application for Scoring Music [1]. This application enables the user to design digital musical score from digital images and movie files. These digital media are played back in an ordered sequence. Both the performer and composer can program the text notes into the timeline. The entire visual score can be played, paused, forward, and rewound. VScore is available under the GNU Public License. New features include network based control (for multiple performers), an on-screen cursor, thumbnail outline, speed control, and external control via MIDI and Max/MSP.

1. INTRODUCTION

Although some notation applications support basic timeline display [2, 3], they are overly expensive. Other applications [4, 5, 6] are less expensive, but are non-dynamic. VScore is a free application which suggests a novel digital approach to scoring music and its dynamic display.

VScore software facilitates performance of compositions based on absolute time. The application enables sequential display of images in a simple and elegant manner. In a performance, VScore offers a solution to problems such as page turning and stop-watching.

The composer assembles their own score by inserting image and/or movie files into a timeline and assigning their duration. The absolute-time based music notated in graphic and traditional notation were utilized in VScore with equal success. The complete VScore timeline can be controlled by a conductor and displayed to performers over a network connection. The performer is able to insert performance notes in order to predict the coming cues and changes.

VSCORE 1.0 SUMMARY

VScore as a standalone application was originally developed using GTK/GNOME library to manage the general interface (e.g., pulldown menus and widget placement) and QuickTime for images. For better performance and common platform availability, the program was re-written using the Qt GUI library to handle both.

The main feature of the application is the timeline, which enables placing and images in a sequential order.
The interface is similar to a spreadsheet. By clicking on a cell for a filename, a file menu selection interface is presented. Similarly, the performer can customize their score by inserting various text remarks. The notes are timed and framed by the starting and ending temporal points in the score. The user types in the text and temporal points in seconds as shown in Figure 3. Further, the text can be displayed in one of the nine regions, which divide the screen into equal parts. Thus, if the scored music involves multiple performers, each performer can use a separate layer to mark their individual remarks. Each performer can choose from five available colors to further individualize their texts.

![Figure 3. VScore 1.0. Note storage and placement window.](image)

One goal in VScore 1.1 was to add “drag and drop” ability to user and composer notes in a timeline. The planned functionality is similar to transparent desktop sticky notes. As of the writing of this paper, that feature is not yet implemented.

2. VSCORE 1.1

2.1. Network Control

The original version of VScore could only be used by a single performer. VScore 1.1 facilitates network control via an entire ensemble via TCP connections. Each running copy of VScore can communicate with other running versions. Privileges can be set within the given project to allow one or multiple copies to control playback, pause and movement (forward and reverse) through the timeline. Notes are stored individually for each performer.

2.2. Movie Files

VScore 1.1 enables importing of images as well as movies through a QuickTime widget. Often the composer prepares the score as a movie in another application such as Macromedia Flash [7] and provides the performer with a complete product. This new feature allows synchronization between static media such as photos, dynamic media such as movie files, and composer/performer specific information and queues stored in VScores “notes.”

2.3. On-screen cursor

VScore 1.0 had a very simple moving button beneath the image area. The position of this button indicated the location and progress through the entire score. Based on feedback from performers and composers, VScore 1.1 offers an additional timeline element. The cursor is a practical pointer, which temporal progression unit on the screen can be defined either as 0.5 or 1 second. At each interval, the cursor moves from left to right on the screen. This is very useful for image-based scores, where precision is needed. The cursor can be set to indicate overall absolute progress through the score or progress within a given timeline event, indicating when the next event is about to occur.

![Figure 4. The on-screen cursor is a solid semi-transparent line scrolling through the screen in assigned intervals.](image)

2.4. Thumbnail outline

In order to better facilitate compositional setup of a score, a timeline thumbnail viewer has been added to VScore 1.1. Rather than simply viewing the spreadsheet menu, the thumbnail outline enables the viewer to retain the feeling of the score as a whole throughout the performance. The outline displays the thumbnail of the images, which are in the score. A pointer indicates on which image or “page” in the score the performer is currently located.

In the original version of VScore, playback speed was static. The playback speed can now be modified via a multiplier, allowing sections or the entire score to be played back at different tempos.
in compositions where the temporal precision is of primary concern. A metronome window can be also activated from the View menu. This feature enables the performer to set the display metronome mark. A flashing button then appears on the screen at the intervals set by the performer. For the scores with constant pace, this feature can be preferred to time following.

2.8. Documentation

The main features of VScore are now documented in a pdf included with the application. A basic tutorial also guides the novice user through key aspects of the program. Similarly, a few small template examples of compositions are included.

3. Conclusion and Future Directions

VScore 1.1 represents a set of major improvements to the original version, based on feedback from performers and composers. Unmentioned in this paper are minor changes to internal program code that have fixed minor programming bugs, which manifested during specific conditions. It remains a primary goal of this project to distribute the application freely and receive further feedback and suggestions to further functionalities from the users. Based on continued feedback, we are evaluating online repositories such as Sourceforge.net.

Future directions for the development of VScore include more stress testing. Since the application is designed to be used in a performance situation, bugs and crashes are simply not tolerable. The QT GUI library runs on Mac OS, Windows, and Linux platforms. VScore currently runs only under Mac OS X. Future work will include ports to Windows and Linux.

4. REFERENCES