Hyperscore: A Graphical Sketchpad for Novice Composers

Morwaread M. Farbood and Egon Pasztor
MIT Media Laboratory

Kevin Jennings
Trinity College Dublin

The Hyperscore graphical computer-assisted composition system for users with limited or no musical training takes freehand drawing as input, letting users literally sketch their pieces.

Designing an intelligent, intuitive system that enables novices—particularly children—to compose music is a difficult task. We can view the problem as a spectrum of tasks that range from the development of musical algorithms for automating the compositional process to designing an appropriate interface for humans to interact with the machine. The Hyperscore software tool attempts to address both of these issues.

As a graphical environment that facilitates composition through intelligently mapping musical features to graphical abstractions, Hyperscore provides a visual analogue for what is happening structurally in the music as opposed to displaying musical events in procedural notation or as a set of parameters, as is often the case with other graphical composition systems. The fundamental idea of Hyperscore is that anyone can perform two key creative activities without musical training: compose short melodies and describe the large-scale shape of a piece. Providing graphical means to engage in these two activities forms the basis for Hyperscore's functionality.

There have been numerous past examples of graphical computer-assisted composition systems. Many of them are suited for professional musicians, using graphical objects to represent musical functions or tweak parameters. The latter category includes "traditional" commercial applications such as Digital Performer, Cubase, and Vision—multitrack sequencers that use graphical input to manipulate low-level parameters. The former category contains PatchWork/OpenMusic,[1] an object-oriented environment designed by researchers at the Institut de Recherche et Coordination Acoustique/Musique (IRCAM) to encapsulate musical functions in graphical objects that can be dragged, dropped, and interconnected to implement musical algorithms. Less conventional is Iannis Xenakis' UPIC,[2] a system that uses a large, high-resolution graphics tablet for input. Its macrocompositional level lets the user draw a time-frequency score consisting of lines, curves, and points.

Systems such as David Zicarelli's OvalTune, a program in which users create sounds and visual images simultaneously by painting with a mouse, and CyberBand,[3] developed at IBM, are designed for users who don't necessarily have musical experience. CyberBand is similar to Hyperscore in some ways: It uses riffs, or thematic fragments, as musical building blocks and higher-level modifiers to edit and refine the music. Its interface, however, is fundamentally different from Hyperscore's. It does not use drawing as a method of combining musical material and lacks the visual freedom of Hyperscore's environment.

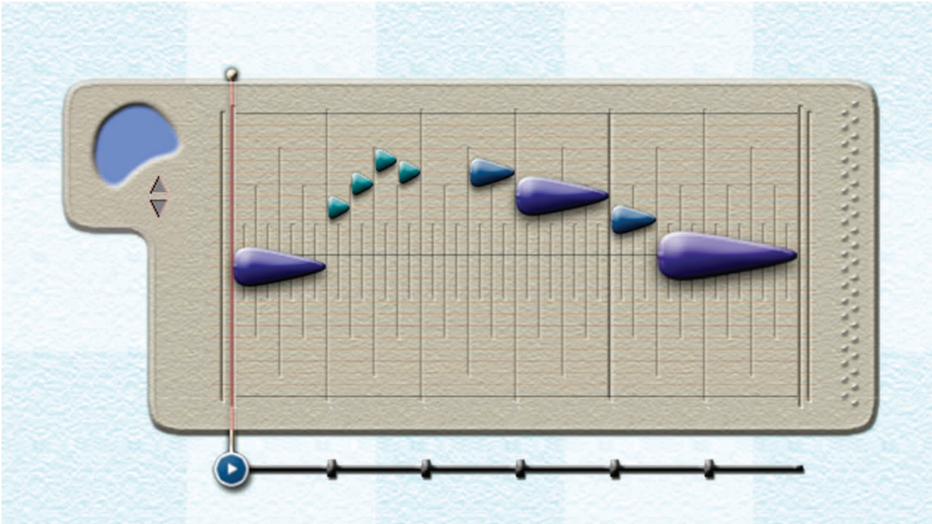
Other systems accessible to musically untrained users—including Maxis/Iwai's SimTunes (http://www.iamas.ac.jp/~iwai/simtunes/), the popular Macintosh program MetaSynth (http://www.uisoftware.com), and Morton Subotnik's Music Sketch Pads (http://www.creatingmusic.com/mmm)—can be fun to play with but don't aid the user in traditional composition.

The system

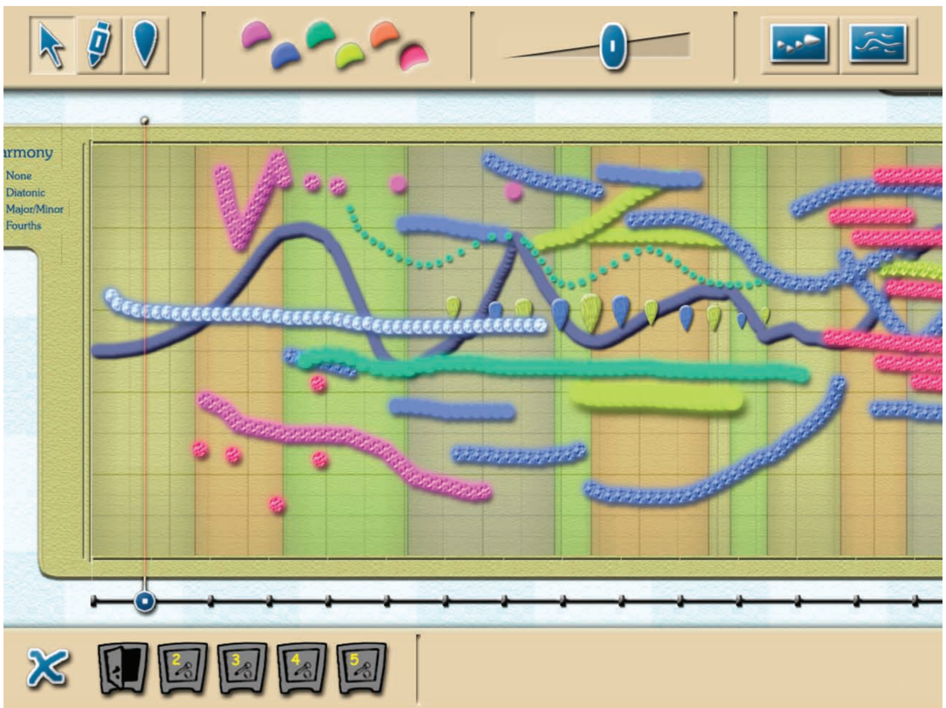
Hyperscore, a Windows application written in C++ using DirectX, consists of an expansive, zoomable canvas where users can create any number of musical fragments and whole pieces. Users can position these musical objects anywhere on the canvas and can view four different levels of zoom for ease of editing. The first step in composing a piece is creating some melodic material in motive windows (see Figure 1). The window's vertical axis represents pitch (spanning two octaves), while the horizontal axis represents time. Users can stretch or shorten the window depending on how long the motive is. Colorful droplets represent notes, and users add them by clicking on the grid. The system interprets blank spaces as rests.

The user chooses a color for each motive and composes a piece by selecting different colored pens and drawing into a sketch window (see Figure 2). Every time

Published by the IEEE Computer Society



1 Motive window for creating melodic material.

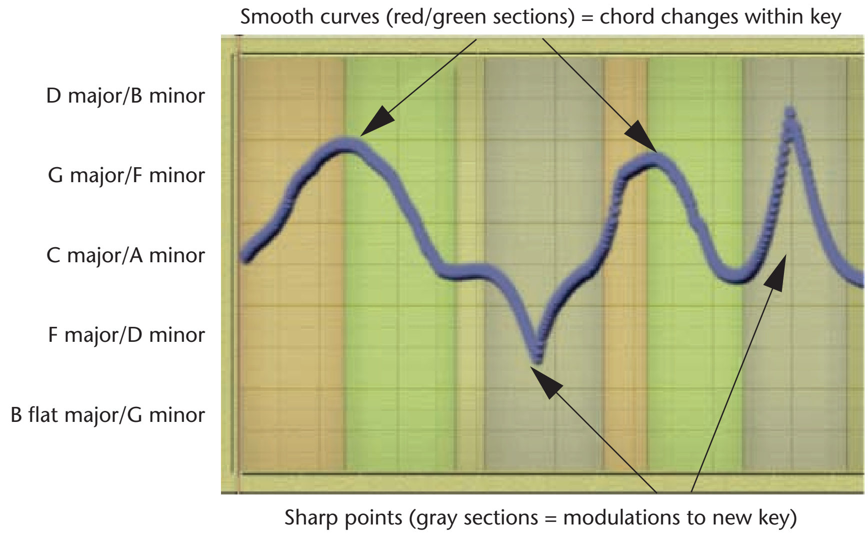


2 Hyperscore screenshot showing four motive windows and one sketch window.

the user draws a line of a particular color, Hyperscore adds the motive mapped to that color to the piece. The start and end points of the line determine how many times a motive repeats. That is, a fixed pixel-to-duration metric calculates the length of time a line plays. Drawing the line straight makes the motive repeat with the precise melodic intervals of the original motivic material. Curves and bends in the line impose a pitch envelope on the motive's repetitions but do not alter the melodic contour to the point that the new material is unrecognizable from the original motive.

Users can reshape the lines after drawing them by right clicking and then dragging. Professional drawing packages such as Adobe Illustrator represent curves as linked Bezier segments where the user explicitly clicks and drags the vertices of the Beziers' control polygons. This type of control-point curve editing can become tedious. In Hyperscore the user can grab the curve and pull. Physically simulated, the curve responds in a natural manner, and users can make small or large changes in the same way they would drag a rope. A list of connected two-dimensional points represents each line, and a spring with a fairly stiff spring constant models the line segment between each pair of control points.

Other line-editing features include cutting and pasting, changing instrumentation, and increasing or decreasing playback volume. By default, the instrumental sound mapped to each line is pizzicato (plucked orchestral strings). Changing the instrumentation also changes the line's appearance. By default, it is textured; it becomes smooth when the sound changes to bowed orchestral strings. The current version of Hyperscore



3 Harmony line.

only offers these two string sounds. All sound output is MIDI, and either the computer's sound card or an external MIDI synthesizer can act as the output device. We could easily add more instruments in future versions of Hyperscore, but we would probably need a more visually striking representation to differentiate between multiple instrument options.

The lines drawn into a sketch window combine to form larger, multivoiced segments of music (a maximum of 30 simultaneous voices). These lines can communicate a musical gesture when effectively interwoven and overlapped. Hyperscore facilitates composition through this combination of creating a visual representation of the large-scale structure of a piece and the process of integrating musical lines. This representation provides high-level control over the dramatic arc of the piece as a whole as well as the placement of individual motivic elements.

Shaping music

Hyperscore also addresses harmony in a couple different ways. In the simplest example, harmony can be a single chord without a reference point, without regard to what precedes or follows it. Users can add individual chords consisting of three simultaneous voices to the sketch window. They are displayed as colored droplets, with each color representing a different harmony type: major, minor, augmented, diminished, and so forth.

Defining transitions from one chord to another is the first step toward adding functional harmony. This can be as insignificant as the prolongation of a previous chord or harmonic function or as far-reaching as a move to a new key. Hyperscore lets users describe these types of harmonic progressions by shaping a central line. Depending on whether the curves in the line are going up or down and depending on their shape, the computer chooses relevant chords.

Users can choose from between four harmony styles: none; diatonic (all chromatic pitches are changed into diatonic pitches); major/minor (tonal harmony based on Bach-style harmonization); and fourths (based on chords constructed from fourths rather than thirds as in the case of regular tonal harmony).

One reason for having a graphical notation system in the form of freehand drawing is to provide the user with an expressive means of shaping musical direction. Drawing a contour is a simple and intuitive way to depict areas of harmonic tension and resolution. The harmony line (see Figure 3) that runs through the center of each sketch window can control major/minor and fourths harmony types. By default, every sketch window comes with a flat harmony line. Clicking and dragging actions shape the line. Color bands appear to indicate the line's parsing. Flat areas are not colored and represent regions with stable (or functionally tonic) harmonies in the current key.

Upward areas, colored red, result in unstable or subdominant/dominant harmonies. Downward areas, which naturally follow upward areas, resolve the previous unstable harmonies. The local texture of the line at the beginning of each section influences the chords chosen in the flat, upward, and downward sections. Pointed or spiked regions indicate a change in key or modulation. The y-value of the tip of the spike determines the new key. The center is mapped to C major; moving the point up adds sharps and moving the point down adds flats.

Applications and future work

We designed the current version of Hyperscore specifically for traditional forms of composition. It's suited for use as both an educational tool and as a way to explore musical creativity. We have tested it extensively with children who have composed short pieces for string orchestra as part of Tod Machover's Toy Symphony project (http://www.toysymphony.net). The Toy Symphony project aims to introduce children to creative music making through the use of specially designed hardware and software. See the "Using Hyperscore for the Toy Symphony Project" sidebar.

Users can save Hyperscore pieces as MIDI files, a standard format that can be read into any notation program, such as Finale or Sibelius. This makes it easy to go from Hyperscore format to musician-readable format, giving a composer the option of sketching out a

Using Hyperscore for the Toy Symphony Project

Hyperscore has provided the primary vehicle for composition activities for Tod Machover's Toy Symphony, a large project involving children, orchestras, and technology. During the course of the project, children in Europe (Dublin, Glasgow, and Berlin) and the US (Boston and New York) have worked with the software to compose pieces for string orchestra, some of which local professional orchestras or string quintets then performed in concert.

Hyperscore workshops in each of the five Toy Symphony locations consisted of five sessions involving between seven and twelve children. Composers, musicians, and music educators acted as mentors who introduced children to the software and guided them through the composition process. Children had minimal difficulty in operating or navigating the software and by the end of the first session had moved on to the actual composition work. In almost all cases, children successfully completed the task of composing a short piece for string orchestra within the five sessions.

Workshops typically proceeded through several stages. At each stage, mentors made suggestions for core ideas and concepts upon which children might focus as they worked, but they encouraged children to diverge and explore their own ideas at all stages. Mentors generally adopted a reactive rather then proactive role. That is, they made themselves available to help with working out the details, draw attention to areas that might need further consideration, and provide support and affirmation throughout the process. In general, the mentors refrained as much as possible from leading the children or imposing their own compositional priorities. Some examples of the types of musical ideas explored include making motives go together and building a piece from beginning to ending.

Researchers at the Centre for Research in IT in Education, Trinity College Dublin, are in the process of carrying out a rigorous academic evaluation of Hyperscore as part of ongoing research into technology and music learning. We closely observed a large number of children throughout the process of composing using Hyperscore, both during the Toy Symphony project and in independent evaluations. Analysis of the resulting data is ongoing, but at this point we can make some general observations. The real power of the Hyperscore interface is the manner in which the software makes the composition task manageable by modularizing it. It's much easier for children and naive composers to relate to the notion of making small bits of music and then assembling those bits into a larger work than it is to start with a completely unstructured task. The compositional process generally begins in the motive window, within which children explore ideas relating to rhythmic and melodic character. Because of the motivic nature of the Hyperscore compositional process, this is in many ways the most crucial stage, as the overall character of the final piece will largely depend on the character of the constituent motives.

Having made some number of motives, children then move on to drawing them in the sketch window. In both windows, children's initial approach might be broadly divided into those with a visual focus and those with an aural focus. Some children try to make visual patterns, where the sound output is of secondary importance, while some children immediately focus on the graphical elements as a means to control the sound. In some cases, children will make seemingly random actions simply to see what happens.

In fact, this randomness is an inescapable feature of interactions with interfaces of this type. In designing a nondirective, open interface to facilitate creative music making, it's inevitable that users will have the facility to exhibit this type of behavior, with obvious implications for learning. Rather than try to control for this in the interface itself, the approach here is to make the interface as open ended as possible and then provide guidance and feedback via mentors during the workshop process. By a process of nondirective questioning, children's attention can be focused on those areas that seem to lack intent or cohesion. In this way they can be guided into the decision-making process that is at the heart of any creative art. The role of the interface in this process is to act as a canvas, a vehicle to explore and think about fundamental musical ideas while engaging in real, meaningful creative work.

composition in Hyperscore and then editing in standard notational format.

We would like to add many other features. Most would make Hyperscore more useful to professional composers as well as novices. At the top of the list is direct editing at the individual note level within the sketch window and improving the harmony line algorithm to allow more precise control. Adding external methods of inputting motivic material—both audio and MIDI—would also be useful.

Asking the computer to do more of the work might also help users. One approach would provide a special computer pen. Drawing annotations with this pen would give the computer license to add extra material not controlled by the user. If a user is dissatisfied with a particular section and cannot see a way to improve it, adding a computer pen annotation would add musical material generated at Hyperscore's discretion.

Another idea is a reverse Hyperscore, where the input is a piece of music (in MIDI format, for example) and the output is a Hyperscore sketch. This would be a much more difficult task than the current graph-to-music approach. We would need to provide some concrete method to break down a piece into basic motivic elements, perhaps by doing a statistical analysis of recurring rhythmic, melodic, and harmonic patterns.

These are just a few of the diverse possibilities for improving and modifying Hyperscore. We hope that the next version will further our goal of making it an effective tool for musical and educational purposes. ■

Acknowledgments

Morwaread M. Farbood and Egon Pasztor wrote Hyperscore. We thank Kevin Jennings, director of pedagogy for the Toy Symphony project, for creative contributions to Hyperscore. This project took place under the supervision of Tod Machover at the Massachusetts Institute of Technology and was generously funded by the MIT Media Laboratory, CSK, and Sega.

References

1. G. Assayag et al., "Computer Assisted Composition at Ircam: PatchWork and OpenMusic," *Computer Music J.,* vol. 23, no. 3, 1999; http://www.ircam.fr/equipes/repmus/RMPapers/CMJ98.
2. H. Lohner, "The UPIC System: A User's Report," *Computer Music J.*, vol. 10, no. 4, 1986, pp. 42-49.
3. J. Wright et al., "CyberBand: A 'Hands On' Music Composition Program," *Proc. Int'l Computer Music Conf.*, Int'l Music Assoc. and Program of Psychoacoustics of Aristotle Univ. of Thessaloniki, 1997, pp. 383-386.

Morwaread M. Farbood *is a PhD student at the MIT Media Laboratory. Her research interests include algorithmic composition and machine-learning approaches to modeling musical structures. Farbood received a BA in music from Harvard University.*

Egon Pasztor *is a PhD student at the University of California, Berkeley. His research interests include graphics, art, style, and design in all forms as well as interactive systems, simulated physics, and software engineering. Pasztor received an MS in media arts and sciences from the MIT Media Laboratory.*

Kevin Jennings *is a research fellow and PhD student at the Centre for Research in IT in Education at Trinity College Dublin and Media Lab Europe. As a classical guitarist, he has performed widely in both Ireland and Europe, including solo recitals, chamber music, and concerto performances.*

Readers may contact Morwaread M. Farbood at MIT Media Lab., 20 Ames St. E15-443, Cambridge, MA 02139; mary@media.mit.edu.