RECENT DEVELOPMENTS IN THE DIFFERENT STROKES ENVIRONMENT

Mark Zadel and Gary Scavone
Music Technology Area, Schulich School of Music
Centre for Interdisciplinary Research in Music Media and Technology
McGill University, Montreal, QC, Canada
{zadel,gary}@music.mcgill.ca

ABSTRACT

Different Strokes is an interactive software interface for solo computer music performance. It has been recently used in the context of the *d_verse* project, a series of interdisciplinary performance and video works. This application of Different Strokes has presented a number of challenges and has required the development of additional features, such as real-time granulation, multi-channel spatialization, and mappings to synthesis parameters. This paper describes these updates and the lessons learned while adapting the software for use in the *d_verse* project.

1. INTRODUCTION

Different Strokes[4, 5] is an interactive graphical interface originally designed for the context of solo laptop performance. A fundamental design goal of the software is to improve performer control and transparency in laptop performance, which is traditionally dominated by computerized automation and imperceptible mechanics. We define laptop performance to be the musical performance of software instruments through standard computer interfaces.

Since 2007, the Different Strokes environment has been used in a collaborative art project exploring themes of communication and gesture through dance, sound, poetry, and film. This project has involved artistic and technological demands on Different Strokes that have well exceeded those originally envisioned for the program. This paper describes the issues, adaptations, and lessons learned from these experiences.

2. DIFFERENT STROKES

The Different Strokes software starts as a black, full-screen canvas. The user draws strokes on the screen using a digitizing tablet, as one would in a freehand drawing application. The strokes become tracks along which “particles” can move. Particles are singularities that form the program’s basic animated element. Each stroke is associated with a sound file and each particle represents a playback head. Particles move along the strokes according to the temporal evolution of the original drawing gesture, so that quickly drawn strokes produce fast-moving particles, and similarly for slowly drawn strokes. Particles always move in the drawing direction of a stroke. A screenshot is shown in Figure 1.

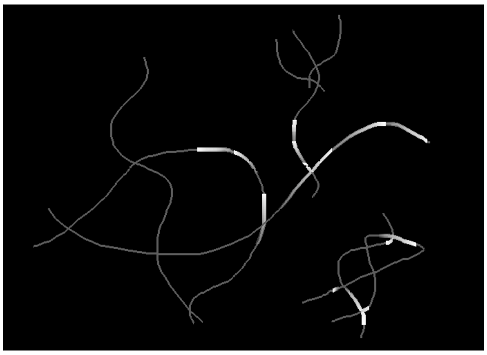
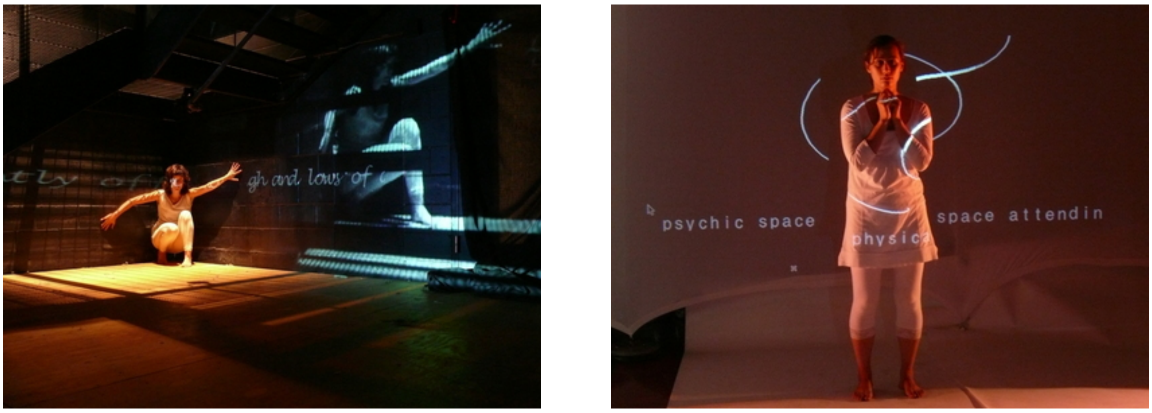


Figure 1. A screenshot of the Different Strokes interface.

When a particle encounters an intersection between strokes, it divides into two and each particle moves out of the intersection along the two stroke paths. This is the main axiom that governs the simulation and allows loops to contain continuously orbiting particles. See [4, 5] for more information about common stroke shapes and system behaviours.

The original software design used variable speed sample playback to generate sound. A pre-loaded sound file is selected via the keyboard and associated with subsequently drawn strokes. The entire sound file is mapped to the stroke such that the sound file’s beginning and end correspond to the beginning and end of the stroke. As a particle moves along the stroke, it acts as a playhead that scrubs through the wavetable according to its movement. A consequence of this mapping is that playback speed is dictated by drawing speed: quickly drawn strokes may play back at a high pitch and slowly drawn strokes may play back at a low pitch, depending on the length of the stroke and the length of the associated sound file. Changes in drawing speed result in correspondingly changing playback speeds.



(a)

(b)

Figure 2. Images from *d_verse* project rehearsals. Note the projection of the Different Strokes interface in (b).

3. THE D_VERSE PROJECT

d_verse: transitional algorhythms of gesture is a series of performance and video works exploring the themes of communication and gesture, directed by poet pk langshaw of Concordia University, in Montreal. It brings together a number of different disciplines—dance, film, sound art, and poetry—and explores the permeability of the barriers between them. The works combine live performers and various real-time, digital processing systems to create an active, responsive whole. *Line* is the common element between the different disciplines and is used as the medium of artistic communication between them.

Visually, the works involve dancers onstage in a black box environment. The dancers move in response to the other performance elements or according to prearranged choreography. A commonly explored aspect of these interactions has involved the projection of video onto or around the dancers, with accompanying real-time spatialized audio. The projection elements include real-time processing of captured video, interactive Flash animations, and the Different Strokes interface. Photos from project rehearsals are shown in Figure 2.

An important element of the *d_verse* project is the close interrelation between the individual disciplines. At various points in the works, each artistic element is expected to take the lead, while the others follow, reinterpret and react. Each discipline is responsible in part for creating the narrative, and they are all engaged in a dialogue with each other.

Different Strokes was chosen as a major audio-generating component of the *d_verse* project because of its tight coupling of audio and visual media, as well as its use of line. Work has focused on extensions to the Different Strokes interface to accommodate the needs of the project, and in modifying its sound generation algorithms to meet the group's aesthetic goals. Timothy Sutton, a Concordia University graduate, acts as composer and is responsible for coordinating the overall musical aesthetic.

Different Strokes is used to control audio processes and to simultaneously give a visual representation of that control. A large part of the audio in *d_verse* comes from Different Strokes, treated with outboard processing in Max and external hardware signal processing. Recordings of recited poetry form a fundamental component of the soundscape, either played back directly or through a granular synthesis algorithm. The system now also allows recited poetry to be recorded in real-time while a stroke is drawn for subsequent granulation.

3.1. d_verse and Different Strokes

The *d_verse* project's focus on interrelated disciplines mirrors the integration of audio and visuals in Different Strokes. As *d_verse* is realized through real-time performance and dynamic gesture, Different Strokes is a natural fit in this context.

One of the main themes of the *d_verse* project is the exploration of gestural line. Different Strokes evokes gestures in multiple ways: the amplification of the performer's physical drawing gesture, and the flowing, gestural movements of the animated particles. The movements of the animated particles along the strokes resonate with the dancers' movements. Since the project is based around words, both spoken and written, the software can also become an activated writing surface for written text.

While gestural and real-time qualities of Different Strokes lend themselves well to the *d_verse* project, certain aspects needed to be updated to work effectively in this context. These are elaborated upon in the following section.

4. FEATURE UPDATES

Certain features have been added to the software over the development phase of the project. These include multichannel support, granular synthesis functionality, mappings from the simulation features to synthesis parameters, and updates for robustness.

4.1. Multi-Channel Audio and Sound Placement

The application was updated with a multi-channel signal path to allow sound spatialization. The Synthesis ToolKit's (STK) new StkFrames structure[2] was introduced into the code to easily manage blocks of multi-channel audio.

The spatial placement of mono synthesis voices is one of the particular design challenges in our development efforts. There is one synthesis voice per stroke, associated with a single "sounding" particle. There are many features associated with this particle that could be used in the mapping: its position in space, its normalized position along a stroke, its speed, and stroke curvature at its position being just a few examples. The panning algorithm implemented uses the particle position in the global 2D plane to determine the sound's position in space. Thus, a large loop containing a particle that orbits the whole screen will produce a sound that circles the central listening position. This spatialization was implemented using the VBAP[1] algorithm.

It remains to be seen if this panning strategy will suit the project. The projected drawing action is sometimes required to overlap with a particular part of the physical space (for example, drawing on the dancer's arms), and this might interfere with the audio performer's ability to place the resulting sound in a particular position in the sound space. The algorithm will be tested in future rehearsal sessions and adjusted as necessary.

4.2. Granulation

The *d_verse* project called for real-time processing of spoken poetry. Granulation of an audio input buffer was added to the application using the Granulate object from STK, with input via the appropriate facility in RtAudio. A new "granular" stroke type was added to the software for this functionality, selectable via a predefined keystroke. Each granular stroke owns its own input buffer and synthesis voice. As the pen touches the tablet, recording is turned on and the input buffer is filled. Recording stops when the pen is lifted.

A challenge in integrating granular synthesis was to define the mapping from the simulation state to the granular parameters. There are many individual parameters in a granular synthesis model that can be controlled, and many parameters in the simulation that can be queried.

One option that was discussed was to associate vectors of predefined granular synthesis parameters with various points in the 2D drawing plane. The 2D position of a sounding particle would be used to determine a granular parameter mapping by linearly interpolating the spatial settings, allowing the parameters to evolve in time as a particle moves along a stroke. This scheme would suffer, however, from the same issues mentioned in the previous section: having to draw on particular parts of the 2D surface could conflict with having to follow the onstage movement of the physical dancers.

Our current solution is to control granular synthesis parameters externally via a MIDI fader box. This decouples the synthesis control from the simulation, and makes it possible to effect various timbres independent of the drawing position. While this deviates from the notion of using only the interface to control the sound output, the added flexibility was needed for this particular case. The granular parameters are set globally, and affect all of the granular synthesis processes identically.

4.3. MIDI input and output

MIDI input and output was added to Different Strokes to help communicate with external software. Since the audio code in the application was done at a low level in C++ using STK, the implementation effort required to add various audio effects is high compared to other prototyping languages such as Pd or Max. This fact led to the decision to use external software processing and plugins to help vary the sound instead of reimplementing existing audio effects. This has the added benefit of distributing the computational load for audio processing across multiple computers. The external system is fed the Different Strokes audio, as well as control data in the form of MIDI messages.

MIDI capabilities were added via the RtMidi class[3]. This required a mapping from simulation events to equivalent MIDI messages, communicating potentially interesting simulation events such as "pen up," "pen down," particle creation, and other information. An aesthetically interesting mapping from these messages to processing parameters was devised by the composer. As mentioned above, MIDI input is also used to modify the parameters of the granular synthesis processes. The MIDI subsystem will soon be augmented with an OSC system, as OSC is more expressive and powerful than MIDI.

4.4. Other Changes

Various other smaller changes were made to the codebase as well. The graphics were changed to greyscale and the strokes were made thicker in order to be detectable by video cameras for processing and reprojection. An issue with this change is that the graphical feedback indicating the sound file associated with each stroke (previously indicated by stroke colour) is now missing.

The original Different Strokes application exhibited a feedback situation where certain stroke topologies (double loops) would result in an ever-increasing number of particles. The program was modified to eliminate this effect by imposing a small, fixed upper limit on the number of particles allowed per stroke. This solution leads to some interesting temporal effects, where combinations of strokes do not always repeat their movement regularly. Pseudorandom repetition and stuttering is observed, and this nonlinearity can be used to interesting creative ends.

Moving particles scrub through the wavetable according to their speed and position. As new particles are added, the playhead jumps to the new position, creating appealing cutting effects. Unfortunately, this also introduces discontinuities and clicks in the audio output. A crossfading feature was created to help alleviate this problem. Each stroke is associated with two wavetable playback voices, and discontinuous jumps are smoothed by fading between the old and new playhead positions.

The C++ code was also upgraded to the most recent version of STK (4.3.1), and refactoring was done to improve the code.

5. UNEXPECTED CHALLENGES

There were a number of interesting, unexpected issues that cropped up over the course of our group development sessions.

5.1. Key Commands in Low Lighting

The Different Strokes interface design has used key commands for selecting the stroke drawing mode. For example, the user hits the "2" key to select wavetable number two, and subsequently drawn strokes are associated with this wavetable until the mode is changed again. In adding extra features, new keystrokes were added, and there are now a fair number of meaningful keys for use in this project. Relatively precise targeting is required in the performer's non-dominant hand to find the desired key.

d_verse rehearsals have occurred in dark black box environments, making it very difficult to see the keyboard. One drawback of using many individual keyboard commands is the fact that it is difficult to find individual keys without adequate lighting. While using a small light or a back-lit keyboard is a quick solution, simplified keyboard commands or an alternate method of selecting modes may be preferable in the future to allow effective performance in these conditions.

5.2. Parallax and Drawing on Dancers

One of the more aesthetically appealing effects in the *d_verse* project is when the interface strokes are drawn in order to make their projection fall on the dancers and props in an interesting way. This is visually attractive, but can be difficult due to the parallax from the differing positions of the projector and the performer's point of view. If these are not lined up closely, it can be challenging to target a particular part of the physical space. A certain amount of occlusion from the dancers' bodies also occurs that can make targeting difficult.

Different Strokes is an inertial system, in the sense that drawn strokes remain on the canvas until they are explicitly deleted by the performer. If the performer draws on the dancer and he or she moves away, the drawn stroke remains in place until it is deleted. It is extra work for the audio performer to clear strokes that are no longer needed visually.

5.3. Visual Versus Audio Precedence

In developing the project, there was a difference of perspectives on how to use the software. The system was designed as a controller for audio performance. The visual aspect serves as an interface and a visualization, but not necessarily an aesthetic end in itself. In *d_verse*, artistic visuals are central to the project, and the desire was to use the software in a primarily visual way with the audio as a sonification of the graphics. The near-term compromise may be to augment the visuals with external video processing to make them more subtle and aesthetically interesting. The response to this question is still being developed.

6. CONCLUSION

The original Different Strokes software has required a number of modifications and extensions to deal with the specific requirements of the *d_verse* project and to address the unique challenges it poses. Feature additions, such as multi-channel spatialization, granular synthesis, and MIDI functionality have been required. Mappings have been developed to link the spatial and visual world of the Different Strokes interface to synthesis and processing parameters. In particular, mappings to sound spatialization and external synthesis have been developed. The project has required alternative control strategies as well, such as using external controllers for manipulating synthesis parameters. Unexpected issues have cropped up over the course of the project, including the limitations of key commands or the relative focus on visuals versus audio.

While some of these modifications are particular to the *d_verse* project, this work has served as an interesting exercise in design and implementation, and has highlighted areas of the software that would benefit from refinement. Most of these improvements will remain in the Different Strokes application for the long term.

7. REFERENCES

[1] Ville Pulkki. Virtual sound source positioning using vector base amplitude panning. *Journal of the Audio Engineering Society*, 45(6):456–466, 1997.

[2] Gary Scavone. STK homepage. <`http://ccrma.stanford.edu/software/stk/`>, 8 February 2008.

[3] Gary Scavone and Perry Cook. RtMidi, RtAudio, and a synthesis toolkit (STK) update. In *Proceedings of the International Computer Music Conference*, pages 327–330, Barcelona, Spain, 2005.

[4] Mark Zadel and Gary Scavone. Different strokes: a prototype software system for laptop performance and improvisation. In *Proceedings of the Conference on New Interfaces for Musical Expression*, pages 168–171, Paris, France, 2006.

[5] Mark Zadel and Gary Scavone. Laptop performance: Techniques, tools, and a new interface design. In *Proceedings of the International Computer Music Conference*, pages 643–648, New Orleans, LA, 2006.