PROTOGMI MUSICBRUSH – AN EXERCISE IN GENERAL MULTIMEDIA INSTRUMENT INTERFACE DESIGN

Timothy Chen, Kazushi Nishimoto*†, and Kenji Mase**

*ATR Media Integration & Communications Research Laboratories
2-2 Hikaridai, Seika-cho, Soraku-gun Kyoto 619-0288, Japan
†Japan Advanced Institute of Science and Technology, Hokuriku
1-1 Asahidai, Tatsunokuchi, Nomi-gun, Ishikawa 923-1292, Japan
Email: tichen@mic.atr.co.jp, knishi@jaist.ac.jp, mase@mic.atr.co.jp

ABSTRACT

In this paper, we discuss the interface design for instruments of general multimedia artwork. We describe the development of a prototype general multimedia instrument, the ProtoGMI MusicBrush, and show some experimental results. The MusicBrush combines virtual brush painting and music playing using position-sensitive handsets. Based on examination of existing interface conventions and on past experience with multimedia art creation systems, the important distinction between discrete and non-discrete media components was central in the development of the MusicBrush. The primary interface of the MusicBrush incorporates discrete and non-discrete input methods and we examine the usefulness of such an interface in practice.

INTRODUCTION

In this paper, we discuss the interface design for instruments of general multimedia artwork. There is a diverse array of devices that allow artists to create art, usually of a single medium. These devices include musical instruments, painting tools, and the human body. Combining the performances of two or more artists (and often their devices) can form multimedia[1] art. While single-medium art is certainly common and enjoyable, there is tremendous motivation for multimedia art as it is arguably more expressive.

The relationships between each medium in a performance are important. Often, multimedia art performances involve elements of one medium that complement elements of another. A play set to music is an example. Here, the multiple inputs are separate and the resulting output is related. In a different scheme, a performer can control two or more media simultaneously. For example, a dancer can have sensors attached that provide data for a dynamic image based on the dance movements. In such cases, output of various media is mapped from the input of a single device or performer.

The purpose of a *general multimedia instrument* is to provide an interface for controlling elements of various media simultaneously and correlatively. Other multimedia artwork systems often require several exclusive instruments only to produce a specific style of artwork. Our goal is to develop an instrument that can be generally applied to create any real-time multimedia artwork.

[1] The term "multimedia" in this paper refers to the dictionary definition of "using, involving, or encompassing several media" (http://www.m-w.com), without any technological connotations.

ATR has developed two multimedia art systems: the MusiKalScope 1 [1] and MusiKalScope 2 [3], in which music and kaleidoscopic images were created concurrently. The prevalent issue here was minimizing cognitive overload. The MusiKalScope 1 divided media creation into creative and non-creative components, which were controlled by performers and computers, respectively. MusiKalScope 2 further broke down the media space into common attributes, which were allocated to separate performers. These systems did produce a satisfactory output, but revealed several problems.

The most unexpected problem was that the performers did not feel that they had created something by themselves through the performance. We had assumed that people could create multimedia art only if the cognitive load to simultaneously operate multiple media was reasonably reduced. However, it was not sufficient.

We can assume several reasons for this problem (i.e. limited output feedback, unorthodox division of media space), but we infer that the most prominent reason stems from the unsuitable interfaces.

To determine a more suitable interface design, we investigated traditional musical instruments. We found that a scheme that uses both *discrete* and *non-discrete* input methods is advantageous for multimedia art creation instruments. This paper shows how these input schemes were implemented and discusses their effect through experimental performances with the prototype instrument.

The rest of the paper is organized as follows: first we examine how discrete and non-discrete input methods are achieved for traditional instruments and how they affect art performance. We then describe the development of the prototype instrument, the MusicBrush, focusing on the implementation of discrete and non-discrete input schemes. Finally, we show some results from performances and discuss the effectiveness of the new interface.

MEDIA CHOICES

Many types of art exhibit a combination of sound and visual forms and movements. It seems logical for a multimedia *instrument* to focus on creating output for these two media, as was the case with the MusiKalScope. The MusicBrush will continue with this approach. Extending the general multimedia instrument to encompass more forms of media is not our current objective. This is a future direction.

There is a difference between how directly a performer can control music and visual art creation. Drawing basic images is intrinsically simpler than playing any musical instrument since the image

0-7803-6536-4/00/$10.00 (c) 2000 IEEE

output is a direct function of gesture movement, whereas musical instruments maps gesture movement to music output. There are varying degrees of performer-to-music directness: gesture can control tempo directly, but the actual sound of the instrument is often either indirectly produced (strings, percussion) or modified (brass, woodwind) by gesture control. It is this extra unintuitive step that makes musical instrument training necessary.

This difference will make it difficult to create an interface that can provide sufficient control over the output, while adhering to the established conventions of art creation. The MusicBrush interface tries to maintain the intuitiveness of creating visual art by gesture and serve as an adequate input method for music.

INTERFACE CONVENTIONS

Since the general multimedia instrument will be used for the performance of multimedia art in general, it should share some features with existing art creation devices. Many of these are musical instruments used to perform music of any form. As mentioned previously, these instruments exhibit an important distinction between input methods: discrete and non-discrete.

Discrete input is distinctly separable and controllable within the input range. Repeating discrete input sequences should not be difficult. Piano keys and trumpet valves are examples of discrete input mechanisms.

Non-discrete input is precisely the opposite; the input space is continuous which makes it difficult for precise control and thus it is more challenging to repeat the exact same input. Hitting a piano key with variable strength and blowing a trumpet with variable intensity demonstrate non-discrete input.

Discrete input allows people to reproduce melodies that are composed and transcribed beforehand as well as to perform melodies spontaneously. Non-discrete input often is used for nuances of a musical piece that add an organic dimension to performance. By incorporating these two schemes, people can express their musical creativity. Without either of them, it becomes very hard to achieve sufficient musical performances.

Thus, we think that a general multimedia instrument as with all instruments, will require an interface that allows discrete and non-discrete control. However, since the output of this instrument is not solely musical, input methods of other media—visual, in particular—must be examined.

With visual art creation, discrete control seemingly does not play as large a role as non-discrete control. This can be attributed to the fact that the composer and the performer are usually the same person in visual art, while they are different in music. Thus, reproducibility is not often required in traditional painting, sculpture, and other similar forms of visual art. Artists do not need to describe the creation procedure to allow other people to reproduce the same artwork—hence no discrete input (or notational analog to the musical score) is required.

There are, however, forms of visual art where reproducibility is necessary, e.g. ballet, opera, and play. Some notational methods exist to describe choreography, script, scenario, etc., although there is no universal standard like music notation. Discrete input can be defined to achieve reproducibility in these forms of visual art. A new kind of real-time drawing may be created in near future where a composer creates the "score" of a drawing and a performer reproduces it with his own expression.

We are targeting visual art where reproducibility is essential as the primary visual creation of the general multimedia instrument. In addition, we also expect to realize ideas more easily using discrete input for visual art.

MUSICBRUSH ESSENTIALS

The creation of the MusicBrush focuses on conceptualizing an ideal interface for generating multimedia artwork. The final outcome of the artwork—currently the music and images—is also important, but the fact remains that even if an astounding output is generated, this would be moot if it requires unreasonable effort on the part of the performer.

The MusicBrush allows the performer to draw strokes on a virtual canvas using his arms and to play melody on a handheld button interface. Hence, the instrument is aptly coined "MusicBrush." The visual output consists of a form of real-time drawing, but a notational method has not been determined. Figure 1 shows a simple representation of the system architecture and the flow of data.

There has been work done on creating music using gestures. Wright et al. [4] used a digitizing tablet and its handheld input devices to create music by applying drawing motions on the tablet. Fine non-discrete control was achievable with drawing gestures, but the only discrete mechanisms available were a few buttons, which were neither numerous nor positioned well enough to be suitable for discrete music input. The *DanceSpace* by Sparacino et al. [6] has a musical and graphical output based on dancer movements. Again, there was no discrete control for music because the music was generated in harmony with the background music.

The MusicBrush's approach virtualizes the canvas for the purpose of drawing art on a big screen, at the expense of haptic feedback. This also opens up the possibility of 3-dimensional input. The interface required distinct input mechanisms for both discrete and non-discrete input, as mentioned previously. For this purpose, we

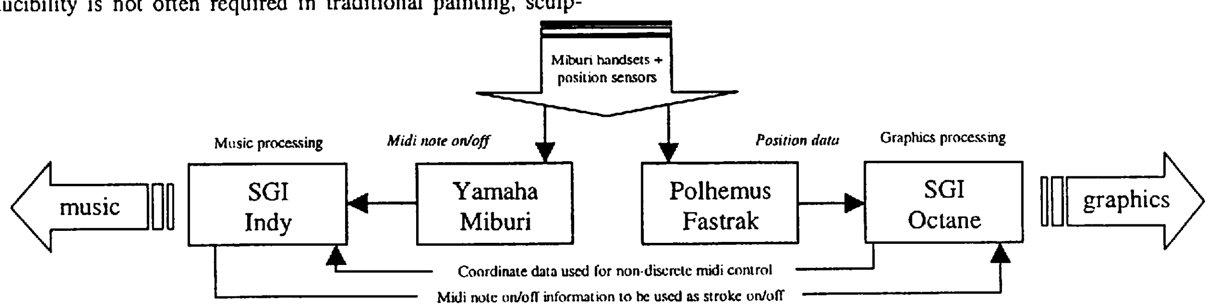

Figure 1: System architecture for the ProtoGMI MusicBrush

combined two devices: the Yamaha Miburi, and the Polhemus Fastrak 3Space.

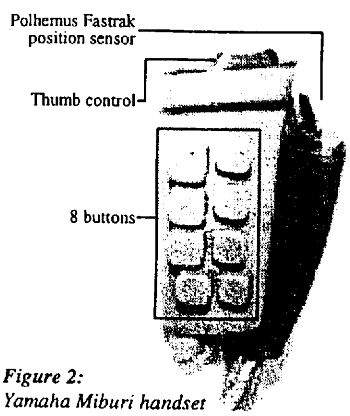

Figure 2: Yamaha Miburi handset

The Miburi is an existing, albeit non-traditional, MIDI instrument. Its two handsets with eight buttons and a thumb control each facilitate simple discrete input (fig. 2). The portable handsets allow the arms to move freely, thus allowing non-discrete input. The Polhemus Fastrak 3Space sensor device is used to capture hand position. Position sensors are attached to the handsets that give measurements based on a central source using electromagnetic fields.

MUSIC CONTROL

The buttons on the handsets are used to provide discrete MIDI note-on/off information. The factory default note mapping of the Miburi was not useful without information from the bend sensors, which were not used. A linear progression of notes from a root to the root one octave higher was a simple solution. Several restrictions arose, however. Sharp and flat notes would be excluded, the notes were confined to one octave, and the eight notes would have to correspond to a certain scale type (i.e. minor or major, etc.). Some of these restrictions were addressed by the use of the thumb controls.

While discrete control is input solely by the button, non-discrete control is handled both by the buttons and the performer's gestures. The buttons act as keys on a piano, whose velocity is based on the strength of the button press. This corresponds to the initial volume of the produced note.

For gestures, there are several possible directions of movement, but the standard Cartesian axes are used for simplicity. In music, these movements modify the nuances of panning, pitch bend, and volume. Panning corresponds to the x-axis movement, and pitch bend is set to the left hand's y-axis movement. The z-axis of the right handset is mapped to volume (variable for the duration of the note).

GRAPHICAL CONTROL

The gesture-driven input mechanism is a template for how non-discrete input will be handled. With the freedom afforded by three dimensions, fine non-discrete input can be obtained and simultaneously applied to many different attributes. It is also a good mechanism for creating visual art, as it is a technique familiar with conventional artists and intuitive for people in general.

The MusicBrush currently allows the performer to draw rudimentary 2D images. Moving the arms while pressing a button will produce a stroke onscreen that traces the path of the movement. To simulate an actual brush, the initial width of the stroke depends on the initial strength of the button press, and also varies as the performer varies the z-axis displacement between the handset and the source.

The colour of the stroke also depends on the note and the current key. A colour was arbitrarily assigned to each of the twelve possible (major) keys. An arbitrary mapping of colour to key was chosen because there is no natural colour progression as there is a linear key progression from one C key to the C key one octave higher. Currently colour is treated as a discrete attribute, but the nature of colour demands it to be continuously controllable. This is an interesting future direction.

PERFORMANCE

During the ATR Open House held on November 4 and 5, 1999, the MusicBrush was hooked up to a large rear-projection screen (fig. 3). While the performer used the MusicBrush to create melody, accompaniment was provided by a digital piano.

On the first day, the MusicBrush was interfaced with Tomoko Yonezawa's Tangible Sound (TS) project [5]. Her system provided a chord to which the MusicBrush's active key is set. This collaboration seemed suitable: the MusicBrush would produce the foreground melody while the TS provided accompaniment. Unfortunately, this combination was not very successful. The chord progression and timing was not regular, and thus the key of the melody remained the same, as were the produced colours.

For the second day, the MusicBrush was disconnected from the TS. A preprogrammed chord progression was sent to the piano for accompaniment, and the user could advance to the next chord in the progression with a push of a thumb control. To provide more dynamic feedback, the background colour also changed when the accompaniment changed. The colour was arbitrarily mapped—similar to stroke colour.

Since the buttons on both handsets produced the same notes and pitch bend was mapped to left hand movement, the right handset was designated the drawing hand. The left thumb control keys were set to provide sharp notes and to advance the chord progression. Incrementing and decrementing the octave was handled by the right thumb control keys.

The Open House was the first field test of the MusicBrush prototype. Visitors watched as the author (Chen) performed with the

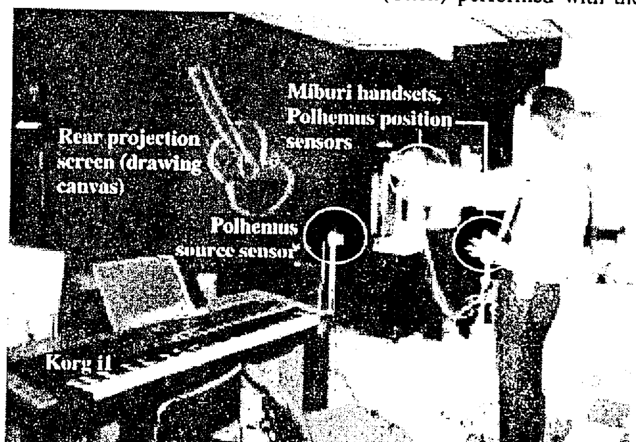

Figure 3: Performance during ATR Open House

device while resident demonstrators observed his progress through the two days. The brevity of the Open House served as additional motivation for reducing the learning curve as much as possible. The MusicBrush was indeed simple to operate. Pressing buttons and moving the hands does not take much effort. The player was able to produce some melody because he had used the handsets during development. Being a piano player was also an advantage. He was not as successful at being able to produce any decent images quickly.

For the next few hours, he constantly struggled with trying to draw better images while not allowing the quality of music to decrease. To more precisely control the placement of strokes required more concentration, and thus he often paid less attention to the melody. For much of the morning, he was producing little more than monochromatic scribbles.

In the afternoon, the player was able to draw more coherent objects. A clear-screen feature was mapped onto a thumb control and this eliminated the cluttering of strokes that plagued the morning's efforts. Focusing on producing more of a picture with a fewer number of strokes, he was able to create a simpler but cleaner image.

Overall, the fellow exhibitors were quite impressed by the progress the demonstrator made in the two days. At the end, he was able to intentionally produce a melody accompanied by colourful, recognizable visual output. It was evident that the employed interface made it easy to create and re-create similar images and music; it also allowed him to express nuances both in drawing and playing music.

Visitors were also allowed to try out the system. Understandably, it was not very easy to use. Drawing was handled better in general, and they had no problems with identifying the resulting creation.

CONCLUSIONS

From the brief trial of the MusicBrush system during the Open House, several goals were accomplished:

- The separation of discrete and non-discrete attributes was effective in simplifying usability and allowed the performer and audience to readily identify what was being created.
- The multimedia (sound and image) output obtained had basic correlations.
- Traditional sound and image output was obtained (cf. "abstract" unconventional outputs).

Of course, there are many issues that need to be addressed:

- Can performers readily make progress in the use of this system, and how well can he express his creativity?
- The interface is novel and unconventional, thus detracting from intuitiveness. How much of a factor is this when it comes to evaluating accessibility for novice users?
- How can intuitiveness and versatility be improved even further? Does this always necessarily come at the expense of simplicity?
- How can additional attributes be added to the system? What is the maximum amount of control a performer can or should be allowed to handle?
- How will this system interface with other traditional and/or non-traditional instruments? Should this interfacing even be considered?
- How well will this system create reproducible artwork? Can there be a consistent notation derived for real-time image creation?

The ProtoGMI MusicBrush is an ongoing project. We are encouraged by the observations during the Open House. Here are some future possibilities:

The interface is still a makeshift combination of devices. Streamlining the components is prudent, though it is preferable to perform further trials. The handset buttons are not in a conventional layout. Using a more traditional layout may facilitate intuitiveness, yet this must not compromise any of the other general multimedia instrument goals. Acceleration sensors and sensor gloves are other devices that will provide a greater range of sensor input. Wireless connections are almost essential, as the MusicBrush's maze of wires was often a hassle.

The system currently supports the output of a rudimentary melody, often consisting of single notes. While the current interface does support multiple notes, an easier way to create these would be preferred. Simultaneous control of multiple instrument voices will be desired. A further development is the usage of this system to create accompaniment, which would correspond to a richer visual output as well. Obviously the interface will have to be revamped for such a purpose.

Drawing in 3D is a possibility, as well as virtual sculpting. Adding sensor gloves can greatly improve tactile input capabilities, as demonstrated by Mulder's Sound Sculpting [2] project. Snapshots of images corresponding to a portion of music can be saved and displayed in a mosaic displaying a temporal visual representation. Finding a more suitable colour mapping would be helpful.

We look forward to exploring these possibilities and continue on the road that leads to the ultimate general multimedia instrument and subsequent popularization of this exciting art form.

ACKNOWLEDGEMENTS

The authors would like to thank Ryohei Nakatsu for his support, and Sidney Fels and Tomoko Yonezawa for invaluable discussion with regards to this project. We would also like to thank the University of British Columbia and the Co-op Japan Program for sponsoring the research internship for Timothy Chen.

REFERENCES

[1] **Fels, S., Nishimoto, K., & Mase, K.** MusiKalScope: A Graphical Musical Instrument. *IEEE Multimedia*, Jul.-Sep. 1998, 26-35.

[2] **Mulder, A., & Fels, S.** Sound Sculpting: Manipulating Sound through Virtual Sculpting. *Proceedings of the Western Computer Graphics Symposium '98*, 15-23.

[3] **Nishimoto, K., Fels, S., & Mase, K.** Towards Multimedia Orchestra: A Proposal for an Interactive Multimedia Art Creation System. *IEEE ICMCS Proceedings 1999*, 900-904.

[4] **Wright, M., Wessel, D., & Freed, A.** New Music Control Structures from Standard Gestural Controllers. *ICMC Proceedings 1997*, 387-390.

[5] **Yonezawa, T., Mase, K.** Interaction of Musical Instruments Using Fluid. *Transactions of the Virtual Reality Society of Japan 2000*, Vol. 5, No.1.

[6] http://vismod.www.media.mit.edu/~flavia/projects.html