社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.
MVE99-90 (2000-03)

汎用的マルチメディアアート演奏装置のインタフェースデザインに関する一検討

ティモシー・チェン*, 西本一志*†, 間瀬健二*

*ATR 知能映像通信研究所
〒619-0288 京都府相楽郡精華町光台 2-2
tichen, mase@mic.atr.co.jp

†北陸先端科学技術大学院大学
〒923-1292 石川県能美郡辰口町旭台 1-1
knishi@jaist.ac.jp

本稿では、汎用的に使用可能なマルチメディアアート演奏装置の実現のための、操作インタフェースのデザインについて検討する。さらに、この検討結果に基づき開発したプロトタイプである MusicBrush を紹介するとともに、試用実験を通してインタフェースデザインの適切さについて検証する。デザインに関する検討は、既存のアート製作装置（特に楽器）のインタフェースの分析と、筆者らのこれまでのマルチメディアアート製作装置に関する研究開発経験に基づいて行われた。その結果、自由な演奏表現の実現には、離散的属性と連続的属性の操作手段の分離と統合が重要であることがわかった。MusicBrush は、この知見に基づき開発された。MusicBrush は、相互に関連性を持つ音楽と映像によるマルチメディアアートを演奏するための装置であり、位置センサを持つハンドセットによってこの両属性の統合操作を実現している。

キーワード：マルチメディア、インタフェース、マルチメディア演奏装置、設計指針

INTERFACE DESIGN OF GENERAL MULTIMEDIA INSTRUMENTS

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

*ATR Media Integration & Communications Research Laboratories
2-2 Hikaridai, Seika-cho, Soraku-gun Kyoto 619-0288, Japan
tichen, mase@mic.atr.co.jp

†Japan Advanced Institute of Science and Technology, Hokuriku
1-1 Asahidai, Tatsunokuchi, Nomi-gun, Ishikawa 923-1292, Japan
knishi@jaist.ac.jp

In this paper, we discuss the interface design for instruments of general multimedia artwork. As a result of this design investigation, a prototype instrument, the MusicBrush was developed and demonstrated. Combining virtual brush painting and music playing using position-sensitive handsets, the MusicBrush is used to create correlative audio-visual art. The design investigation was based on an examination of existing interface conventions and prior experience with multimedia art creation systems. The distinction between discrete and non-discrete media components was an important issue and thus was central in the development of the MusicBrush. As such, the primary interface of the MusicBrush incorporates discrete and non-discrete input methods, and we demonstrate the suitability of such an interface in practice.

Keywords: multimedia, interface, multimedia instrument, design

INTERFACE DESIGN OF GENERAL MULTIMEDIA INSTRUMENTS

I. INTRODUCTION

In this paper, we discuss the design of an interface for instruments of general multimedia art. There is a diverse array of devices that artists use to create art, usually in a single medium. These devices include musical instruments, painting implements, and the human body. Performances of two or more artists and their respective tools can be combined to form multimedia art. While single-medium art is common and enjoyable, there is tremendous motivation for producing multimedia art as it is arguably more expressive.

The relationships between media 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 but the resulting outputs are related. In a different scheme, a performer can controls two or more media simultaneously. For example, a dancer can have attached sensors that provide data for a dynamic image based on their 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 systems used to create multimedia artwork often require several exclusive instruments only to produce a specific type of artwork. Our goal is to develop instruments that can be generally applied to create any real-time multimedia artwork.

ATR's MusiKalScope 1 [1] and MusiKalScope 2 [2] were two multimedia art systems in which music and kaleidoscopic images were concurrently created. The MusiKalScope 1 divided media creation into creative and non-creative components, which were controlled by performers and computers, respectively. MusiKalScope 2 further divided the media space into common attributes, which were allocated to separate performers. These systems did produce a satisfactory output, but revealed several problems.

An 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, this 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 a prominent reason stems from the ill-suited interfaces. To determine a more suitable interface design, we investigated traditional musical instruments. As a result, we found that a scheme that uses both *discrete* and *non-discrete* input methods is essential for multimedia art instruments. This paper shows how these input methods were implemented and discusses their effectiveness through experimental performances with the prototype instrument.

The rest of the paper is organized as follows: first we examine the discrete and non-discrete input mechanisms of traditional instruments and how they influence 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.

II. INTERFACE CONVENTIONS

Since the GMI 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. 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. In addition, 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 challenging to reproduce the exact same input. Hitting a piano key with variable velocity 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 is often used to add nuances to a musical piece, which adds an organic dimension to a performance. By incorporating these two schemes, people can express their musical creativity. Without either of them, it becomes more difficult to achieve a pleasing musical performance.

Thus, we think that the GMI, as with all instruments, will require an interface that allows discrete and non-discrete control. However, since the output of the GMI is not solely musical, the categorization of input 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 found 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.

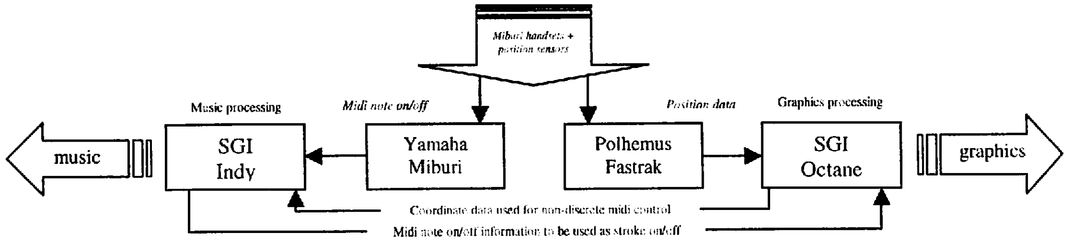

Figure 1: System architecture for the MusicBrush

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. By defining discrete input in these forms of visual art, we can be achieve reproducibility. A new kind of real-time drawing may be created in the near future where a composer creates a "score" for a drawing and a performer reproduces it with his own interpretation.

We are targeting visual art with an emphasis on reproducibility as the primary visual creation of the GMI. Seeing what discrete art objects are created will hopefully lead to a categorization and notation system.

The GMI is by no means limiting itself to audio and visual media, nor were these particular media arbitrarily selected. In fact, in computer terminology, multimedia implies audio and video (i.e. multimedia encyclopedia). Many types of art also exhibit a combination of music and imagery. Naturally, the general multimedia instrument should first focus on these two media.

III. MusicBrush Basics

The creation of the MusicBrush focuses on conceptualizing an ideal interface for generating multimedia artwork. The final outcome of the artwork is also important, but the fact remains that even if an astounding output were generated, this would be moot if it requires unreasonable effort on the part of the performer. Thus, we aimed to create a simple and intuitive interface to make it easy for all users to enjoy creating multimedia art.

This MusicBrush allows the performer to draw strokes on a virtual canvas using his arms and to play melody on a handheld button interface. 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. [3] 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. [5] has musical and graphical output based on the movements of a dancer. 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. The interface required distinct input mechanisms for both discrete and non-discrete input. For this purpose, we combined two devices: the Yamaha Miburi, and the Polhemus Fastrak 3Space.

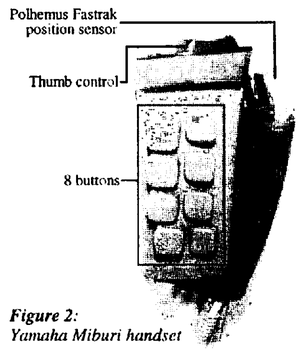

Figure 2: Yamaha Miburi handset

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 freedom of moving arms, useful for 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.

The MusicBrush uses these devices to take discrete and non-discrete input and map it to sound and image output. The following sections describe the mappings in detail, which are summarized in table 1.

III.1. 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 one root note to the next was a simple solution. Several restrictions arose: sharp and flat notes would be excluded, the notes were confined to one octave, and the eight notes would have to correspond to a fixed scale type (i.e. minor or major, etc.).

Table 1: MusicBrush input to output mapping

Input	Discrete/Non-discrete	Sound Output	Discrete/Non-discrete	Image Output	Discrete/Non-discrete
Miburi buttons	Discrete	Pitch	Discrete	Stroke on/off	Discrete
Miburi button press strength	Non-discrete	Volume	Non-discrete	Stroke width	Non-discrete
Left hand x,y position	Non-discrete	N/A		Stroke coordinates	Non-discrete
Right hand y position	Non-discrete	Pitch bend	Non-discrete	N/A	
Left hand z position	Non-discrete	Volume	Non-discrete	Stroke width	Non-discrete
Right hand x position	Non-discrete	Panning	Non-discrete	N/A	

The thumb controls of the handsets are used as modifier buttons. For example, the right hand thumb control changes the key of the active notes up or down an octave. This was not very intuitive, by any means, but it provided a quick workaround to the inherent limitations of the handset. The eight buttons alone, however, is sufficient for beginners to create a rudimentary melody. The modifier keys are intended for more adept users interested in expanding the musical versatility of the instrument.

An alternative note mapping considered resembled that of a trumpet, where a combination of pressed keys corresponds to a certain note. The advantage of this was that there potentially could be many more notes possible. The disadvantage was that with this versatility comes increased complexity. A combination of buttons needed to produce a note will always be more difficult to execute than the single key-single note convention. This will also make it difficult to produce notes in harmony. Furthermore, the eight-button configuration is not standard on any existing musical instrument, so a mapping that would mimic a standard convention like the trumpet configuration would be impractical.

While discrete control is input solely by the buttons, 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 the volume variable for the duration of the note.

III.2. Graphical Control

The simplistic graphics produced by the MusicBrush is prototypical for further refinement. 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 strokes and of the entire artwork is large issue. The nature of colour demands that it can be continuously controlled. In practice, however, the application of colour is rarely controlled in such a manner. Artists of material or digital art must paint with an active colour selected from a finite palette. Mixing or applying other "effects" to the basic content consisting of discrete colours creates nondiscrete colour content.

The MusicBrush originally used an arbitrary colour mapping. A colour was assigned to each of the twelve possible keys. The brightness of the active colour was a function of the note position within the octave. An arbitrary mapping of colour to key was chosen because there is no natural colour progression comparable to the linear key progression from one C key to the C key one octave higher.

The current interface experiments with continuous colour selection. A colour palette that contains all RGB colours is shown in a separate window. The position of the left handset is mapped to a cursor location that points to a certain colour. Thus the user can continuously change the brightness of a colour, as well as relatively smoothly move to a different colour.

Brief testing shows that strokes with pleasant colour gradients can be created, but more concentration was required to monitor and traverse the colour palette. The disadvantage to this scheme is that there is absolutely no correlation between colour and music. A variable palette dependent on key could be a solution.

IV. ATR EXHIBITION

During the ATR Open House held on November 4 and 5, 1999, the MusicBrush system output was hooked up to a large rear-projection screen so that users could easily see what they create (fig. 3). While the performer used the MusicBrush to create melody, accompaniment was provided by a digital piano.

During the first day, the MusicBrush was interfaced with Tomoko Yonezawa's Tangible Sound (TS) project [4]. Her system provided a chord to which the MusicBrush's active key is set. This collaboration seemed to be theoretically suitable: the MusicBrush would produce the foreground melody while the TS set the background music. Unfortunately, the collaboration was not very successful. The chord progression and its timing was not regular, and there were often long periods where the chord did not change, and thus the playable range of the MusicBrush 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, and the user could control the timing of the chord 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 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.

Figure 3: Performance during ATR Open House

The MusicBrush was indeed simple to operate. The player was able to produce some melody because he had used the handsets during development. Being a piano player was also an advantage. Nevertheless, he was not as successful at being able to produce any decent images quickly.

For the next few hours, he 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 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. Most did not pay attention to the extra controls available, confirming that a simple intuitive interface was indeed essential.

V. EVALUATION AND CONCLUSION

From the brief trial of the MusicBrush during the Open House, we saw that an interface with primary discrete/non-discrete input mechanisms is suitable for creating simple multimedia artwork. The multimedia (music and images) output obtained had basic correlations and was relatively conventional (as opposed to "abstract" unconventional output such as kaleidoscopic images).

While the interface is simple, musicians and non-musicians alike will have to learn how to operate the MusicBrush as a brand-new instrument. Considering intuitiveness as an important factor for interface design, we needed to make some compromises when putting the devices together. In the future, we hope to create custom-made handsets that have features similar to traditional musical instruments. This can perhaps reclaim some of the intuitiveness that was lost with the Miburi.

Finally, audience confusion on how the system operated was minimal. More importantly, however, we would like to know if performers can readily make progress with the MusicBrush, and how well he can express his creativity. Another area of study is how well this system will create reproducible artwork, and whether some sort of notation system can be derived for real-time image creation.

The MusicBrush is an ongoing project. We are encouraged by the observations during the Open House. We look forward to expanding the system and continuing on the road that leads to the ultimate General Multimedia Instrument and subsequent popularization of this exciting art form.

REFERENCES

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

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

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

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

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