Iamascope: A Musical Application for Image Processing

Sidney Fels and Kenji Mase
ATR MI & C Research Laboratories
Seika-cho, Soraku-gun
Kyoto, JAPAN, 619-02
fels,mase@mic.atr.co.jp

Abstract

The Iamascope is an interactive, electronic kaleidoscope. The Iamascope combines computer video, graphics, vision, and audio technology for performers to create striking imagery and sound with this aesthetically uplifting interactive device. In the installation, the user takes the place of a colourful piece of floating glass inside the kaleidoscope, and simultaneously views a kaleidoscopic image of themselves on a huge screen in real time. By applying image processing to the kaleidoscopic image, users' body movements also directly control music in a beautiful dance of symmetry with the image. The image processing uses simple intensity differences over time which are calculated in real-time. The responsive nature of the whole system allows users to have an intimate, engaging, satisfying, multimedia experience.

1 Introduction

The Iamascope is a novel application for face and gesture recognition to allow unencumbered musical performance inside the Iamascope. The Iamascope is an interactive, electronic kaleidoscope made as an installation for museums, location based entertainment centers and public spaces.

Kaleidoscopes have captured imaginations all over the world since they were first invented by D. Brewster in 1816. The Iamascope is an *interactive* kaleidoscope, which uses computer video and graphics technology. In the Iamascope, the performer becomes the object inside the kaleidoscope and sees the kaleidoscopic image on a large screen (170") in real time. The Iamascope is an example of using computer technology to develop art forms. As such, the Iamascope does not enhance functionality of some device or in other words, "do anything", rather, its intent is to provide a rich, aesthetic visual experience for the performer using it *and* for people watching the performance. The Iamascope is more than a mirror-based kaleidoscope put in front of a video camera since the types of reflections possible with the computing machinery are more extensive than are possible with mirrors such as asymmetric reflections and different tiling patterns. Additionally, the use of image processing techniques on the kaleidoscopic image allows for users to create music which matches the imagery they are producing. From the perspective of face and gesture recognition, the Iamascope provides an excellent example of the effective use of image processing techniques to create a successful, responsive, easy-to-use contactless human-computer interface.

In the Iamascope, the image processing techniques add greatly to the magical appeal of the experience. When users enter the Iamascope a symphony of sounds responds immediately to their every movement along with beautiful imagery. By keeping the user unencumbered, the sense of engagement and intimacy is very high as soon as they enter. The current image processing techniques are relatively simple; using only temporal differences in intensity, however, the effect is dramatic since it is well matched to the processing rates and quality of the kaleidoscope imagery. In contrast, if the user had to wear some device it could potentially seriously impair the expressive abilities of the performer to produce beautiful imagery since anything on their bodies will be reflected in the kaleidoscope image. For some expressions this is completely unacceptable as the artistry is impeded rather than assisted by the technology.

This paper describes the Iamascope system and the vision-to-music subsystem which allows for matched musical accompaniment with the kaleidoscopic imagery. The technique is described in terms of the Iamascope, but, could easily be adapted to typical kaleidoscopes. In the first section, an overview of the Iamascope is provided to contextualize the discussion of the vision-to-music sub-system. Following, is a detailed discussion of the vision-to-music sub-system. After discussion of the musical sub-system, a brief discussion of the computer based kaleidoscope mechanism is provided. Finally, some conclusions and future work are discussed.

2 Overview of the Iamascope

A block diagram of the Iamascope is shown in figure 1. For input, the Iamascope uses a single video camera whose output is distributed to a video board with a drain to texture memory and the image processor computer. Imagery output from the Iamascope is displayed on a large (170 inch) video monitor. Audio output from the Iamascope is played though stereo speakers beside the large video monitor. In our current implementation, the video image from the camera is placed in texture memory and then the appropriate part of the video image (currently a "pie" slice also referred to as a segment) is selected to form the original image (O) which is used to create the desired reflections (O'). A multi-polygonal circle is drawn upon which the appropriate textures (original or reflected) are drawn alternately. The necessary reflections for the Iamascope are simulated with texture hardware providing frame rates of 30 frames per second. This frame rate provides low-latency, high bandwidth control of the kaleidoscopic image supporting a sense of intimacy with the Iamascope. The video image is copied into memory of the vision-to-music computer. The image processing part of the vision-to-music subsystem extracts the exact same pie slice (O) from the whole video image as is used to create the kaleidoscope imagery. By doing this, only movements which cause kaleidoscope effects will cause musical effects. A picture of a person using the Iamascope is shown in figure 2.

The kaleidoscope sub-system and the vision-to-music sub-system are all written in C with a Tcl[1] and Tk is front end and a Tk interface [4]. The kaleidoscope sub-system runs on an SGI Onyx with a Sirius video option and runs at 30 fps[2]. The vision-to-music sub-system runs on an SGI Indy (or an O2) at 15fps with full resolution (at half resolution it runs at 30 fps). The systems are set up in a client/server relationship and communicate using a TCP/IP connection with a Tcl protocol layer built on top. The vision-to-music sub-system is the server and the kaleidoscope sub-system is the client. Using the bi-directional communication channel, any changes to the settings of one sub-system synchronizes with the other.

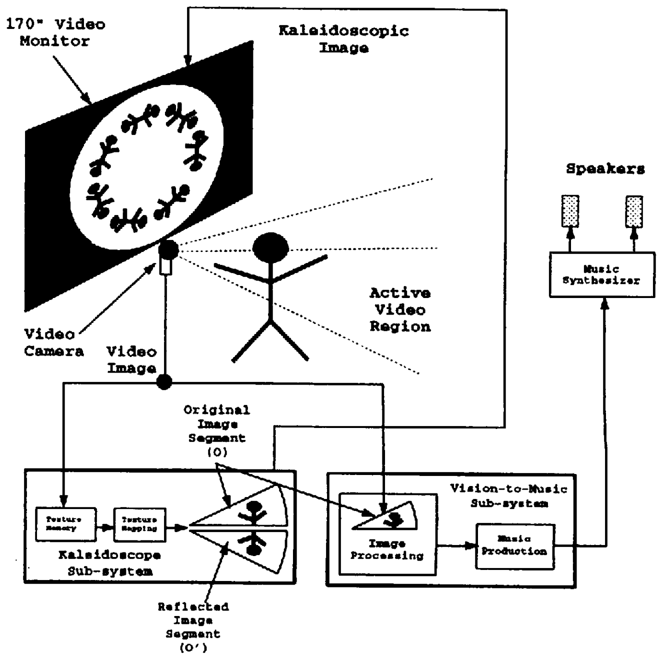

Figure 1: Block diagram of the Iamascope. The kaleidoscope sub-system runs at 30 fps on an Onyx with a Sirius video option and the vision-to-music sub-system runs at 15fps on an Indy.

[1] Tcl is a simple extendable, interpreted, scripting language and Tk is an extension to Tcl that is a GUI toolkit for the X Window System.

[2] The kaleidoscope sub-system can also run on an SGI O2 at 20fps

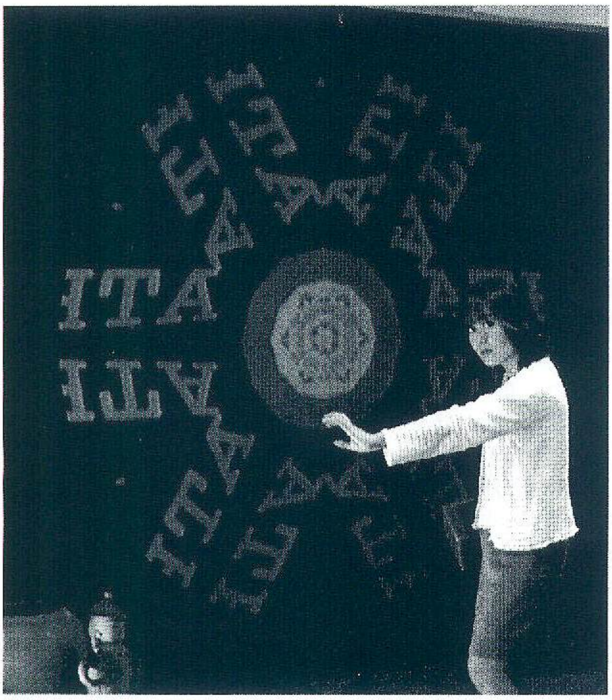

Figure 2: Example of a person enjoying the Iamascope. Note: only monochrome is shown here, the Iamascope image is in colour.

3 Vision-to-Music Sub-system

The vision-to-music sub-system has two parts, image processing and music production. The image processing is responsible for capturing the video image, extracting the correct part of the image and calculating intensity differences. The music production part is responsible for converting a vector of intensity differences into MIDI signals to control a MIDI synthesizer.

3.1 Image Processing

A block diagram of the image processing system is shown in figure 3. The function of the image processor is to divide up the active video region into bins and compute the average intensity difference between the current bin and the previous (in time) bin. The kaleidoscope sub-system controls which part of the full image (i.e. the pie slice O in figure 1) is used to create the kaleidescopic image. The same pie slice is extracted by the image processing system. The pie slice is then divided into bins as shown in figure 3. In each bin, the average intensity difference is calculated from the grayscale values of each pixel using:

$$Ibin_i = \frac{\sum_{p \in bin_i} \sqrt{(g_p(t) - g_p(t-1))^2}}{number\ of\ pixels} \qquad (1)$$

where $g_p(t)$ is the grayscale value of pixel p at time t. At 640x480 image resolution and pie slice of 20 degrees, this calculation can be done at 15 fps on an SGI Indy computer. The vector of intensity differences for all the bins is sent to the music production part of the sub-system. All the image processing code is written in C.

3.2 Music Production

The music production part of the vision-to-music sub-system runs every time a new vector of bin intensity differences is received from the image processor. Each bin represents a semitone offset from the root note of the current key selected by the music production system. The offsets are chosen so that each bin in ascending order is associated with a I, III, or V note from the current key in ascending order. For example, if the current key is C then bin 0 represents a 0 offset (C note), bin 1 represents an offset of 4 (E note), bin 2 represents an offset of 7 (G note), bin 3 represents an offset of 12 (C note, one octave higher) and so on. When the image intensity difference for a bin exceeds a threshold a "note ON" MIDI signal is sent to the MIDI synthesizer. The velocity parameter of the "note ON" signal (i.e., how hard the note is hit) is a function of the image intensity difference. The larger the difference the harder a note is hit.

After a timeout period, the note will be turned off with a "note OFF" MIDI signal. The timeout period is

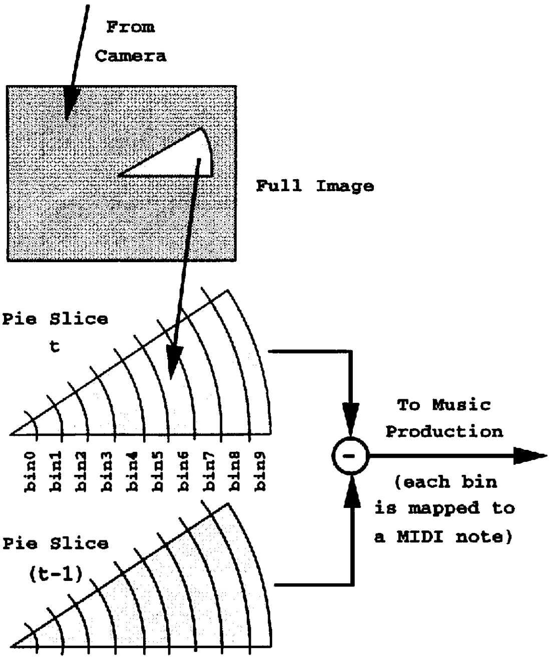

Figure 3: Diagram showing image processing in the vision-to-music sub-system.

selected to match the instrument being used; for example, for a piano the "note OFF" signal is sent 250msec after the "note ON" signal is sent. Some instruments available with typical MIDI synthesizers sound best if they are sustained a long time. These sounds include instruments such as strings and the Goblin instrument. For these instruments the "note OFF" signal is sent after a long delay, i.e., 2000msec. For each instrument a different "note OFF" delay must be decided upon for best performance. Notice, that the "note OFF" signal is not sent as soon as the bin intensity difference goes below threshold since doing so requires the user to explicitly control the duration of each note and the tempo of the music. Instead, using the technique above, the tempo is effectively set by the time between frames; that is, the time from the "note OFF" signal to when the next possible bin intensity update arrives. With a frame rate of 15 fps a pleasant tempo is achieved. Currently, we are considering ways to implement tempo control as a user settable parameter independant of frame rate.

Remember, each bin is associated with an offset from the root note of the current key. How is the key and root note chosen? The key and the root note is chosen automatically by the Iamascope. One of the options in the user interface is to list a key sequence. The Iamascope will cycle through the key sequence, changing keys at some user selected interval. A melodically pleasant sounding key sequence is "C F G Cminor" with the key change set to occur every 2.5 seconds. The root note is hard coded one octave below middle C. Of course, any sequence can be chosen according to the taste of the performer. This scheme is like a very simplified version of what is happening in the RhyMe system [1].

In a typical installation of the Iamascope there are 10 bins. Thus, the user has 10 MIDI notes which they control. The first bin is at the narrow side of the slice, corresponding to low notes with successive bins going farther to the outside of the slice with correspondingly higher notes. When the user moves inside one of the bins, the intensity difference goes above threshold causing the corresponding note to turn on. As they move through many bins, several notes will be sounded nearly simultaneously which will all come from the same chord in the current key. If the instrument is a single note based instrument, like a piano or a harpsichord, the effect sounds like strumming a 10 string guitar where the key and chord is held for you (though the notes have the timbre of the selected instrument).

The music production part also implements a motion detector. If the total image intensity differences of all the bins remains below threshold for a long time a "NO MOVEMENT" signal is sent to the kaleidoscope sub-system. The kaleidoscope sub-system uses this signal to go into automatic mode to display a pleasant, dynamic kaleidoscope effect. As soon as the music production system recognizes motion a "HAVE MOVEMENT" signal is sent to the kaleidoscope subsystem so it can return to its normal mode (see section 4).

The music production system is written in Tcl and behaves as a *Tcl server*. The image processing part of the vision-to-music sub-system is a C client which is connected to the Tcl server. The image processing part sends data to the music production system as a simple Tcl command. Likewise, the kaleidoscope subsystem also connects to the vision-to-music server as a Tcl client, sending and receiving Tcl commands.

4 Kaleidoscope Sub-system

The typical kaleidoscope used in the Iamascope is based on the reflections found in a two-mirrored kaleidoscope. However, in the Iamascope a pie slice (segment) from the original video image is used instead of a triangular slice typical of two-mirror kaleidoscopes. Thus, if the arc angle of the slice is an even integer divisor of 360 degrees a circular image is formed using the alternation of the original image and its mirrored reflection (as shown in figure 1). The even integer multiple arc angle is required so that alternation of the original image with its reflection will exactly fill the circle as shown in figure 1. For example, if we use a 30 degree pie slice then there will be 12 segments which make up the circular image. The odd segments will have the original image and the even ones will have the mirror reflection. Three aspects of this method provide a beautiful effect. First, as the segments exactly fill the circle and there is always a reflected image paired with the original slice the boundaries of each segment will exactly line up without any perceivable discontinuity. Second, since a "pie" slice is used as the original there is a singularity at the centre of the image. This singularity allows the users movement to be perceived relative to the outside edge of the circle and the centre. Third, the pie slice allows for different visual scales to be used from the image. The outer edge of the slice captures a large area of the video image while towards the center of the image only a small area is captured for the reflections. By placing objects close to the centre of the slice it is more difficult to recognize it in the kaleidoscope image allowing for more abstract forms of expression in the image[3].

Additional controls are available which can be interactively exploited. The Iamascope has controls for changing the number and configuration of the mirrors, image background mixing colour, image brightness, arc angle, slice angle rotation speed etc. Currently, the vision-to-music sub-system activates some of these controls when it detects no movement in the image. When there is no movement in the image for an extended period of time the vision-to-music sub-system sends a signal to put the kaleidoscope sub-system into a default mode which shows a dynamically changing image. The main control to achieve this affect rotates the location in the original image where the pie slice is extracted. This is equivalent to spinning the camera giving a smooth beautiful rotating kaleidoscope effect. Using the vision-to-music sub-system as a motion detector in this fashion contributes to the success of the Iamascope in an exhibit setting.

5 Related Work

Related work to the Iamascope fall into two categories; systems which uses image processing to control music and research on different ways to support musical performance to lower cognitive load on performers.

Two well known systems which use vision processing to control music are DanceSpace [2] and the Mandala system from The Vivid Group. In DanceSpace, the performer's dance is captured by a video camera and used to control music and computer graphics. The movement of the performer is mapped such that the dancer's hands and feet control virtual musical instruments. The pitch of the music is controlled by the dancer's head height. At the same time, computer graphics are created and controlled by the dancer's motion. A coloured outline of the dancer's body is successively represented. With DanceSpace, various music styles can be played; however, the sounds generated are always continuously ascending or descending, significantly impacting the quality of the music generated.

In the Mandala system, users' bodies are extracted from the background image and mixed with prerecorded video imagery. There are virtual "hot-spots" which the user can touch with their hands. These "hot-spots" are implemented by processing images over time. These "hot-spots" behave as buttons which the user controls to trigger musical events. This success of this application shows the usefulness of having fast response, simple image processing for real-time applications. One difficulty with the Mandala system is knowing where the buttons are since the only feedback is visual before the button is pressed.

One of the advantages of vision based interfaces is that they leave the user unencumbered with devices. Unfortunately, this also limits the user from having feedback other than audio and visual. Some applications work better with other types of feedback including force, tactile, taste etc. Music expression is especially susceptible to having force and tactile feedback. However, in the Iamascope, the user does not have any force or tactile feedback, thus, the type of information they can provide to control musical expression is limited. Some research has been performed which tries to support musical expression with this impoverished information source.

Specifically, the RhyMe system in [1] describes a computer assisted jazz improvisation system. Of importance is that the system allows a user to play jazz improvisation along with some jazz music background music by hitting keys[4]. The system ensures that the user only plays from a choice of theoretically correct notes according to the theory. Thus, the user need only focus on the overall mood of the piece and the tempo when playing. For systems which only use vision as their input, this approach seems naturally suited. From the perspective of the Iam-

[3] Likewise, the user can also move closer or farther from the video camera to get different scaling effects.

[4] "Keys" are any type of button actuator, such as a piano key or a virtual, vision based active zone.

ascope, the principles of providing only "good" notes for the user to choose from was used which matches the control available using the image processing used. Other systems which support musical production include DanceSpace [2] and Brush de Samba [3].

6 Conclusions and Future Work

The simple image processing techniques used in the Iamascope greatly add to enjoyment and expression possibilities of the Iamascope. The key features which make it a successful application of image processing techniques are:

1. allows for a contactless human-machine interface
2. runs in real time
3. allows music to match kaleidoscopic imagery

There are several directions to take to improve the vision-to-music sub-system. First, currently only image intensity difference is being used to control music. This works very well to keep the music synchronized with any motion which is occurring in the kaleidoscope image; however, it would be useful to also track the human performer. In this case, the performer's hands and face could be used to control different aspects of the musical performance or kaleidoscope imagery. An alternate controller could track other properites in the image such as colour, texture and velocity. Second, in the current system the tempo of the music is a function of the frame rate. We intend to change this by using a higher frame rate and having "note OFF" times be user settable. Third, the music system uses a simple mechanism to keep the music euphonic. By improving the musical performance support by the computer more interesting forms of musical expression should be possible. One possibility we are pursueing is combining the image processing part of the vision-to-music sub-system with the RhyMe system. In this case, the performer in the Iamascope will play improvisation with a band accompaniment. Another possibility is to use the vision-to-music system to control overall aspects of a band. In this situation, a musician (or musicians) play music to accompany a dance performance of someone in front of the Iamascope. The music played by the musician can be processed so that it controls some of the kaleidoscope properties, i.e., background colour, pie slice size, etc. At the same time, the movements of the performer in the Iamascope can control overall qualities of the music being played, i.e., timbre, tempo, etc. Much research remains to be done to determine how best to utilize the MIDI controller implemented by the vision-to-music sub-system to explore the varieties of musical expression possible.

References

[1] S. S. Fels, K. Nishimoto, and K. Mase. Musikalscope: A graphical musical instrument. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems (ICMCS'97)*, pages 55–62, Jun 1997.

[2] S. Flavia. Choreographing media for interactive virtual environments. Master's thesis, Media Arts and Sciences, M.I.T., 1996.

[3] A. Kotani and P. Maes. An environment for musical collaboration between agents and users. In *Proc. of Lifelike Computer Characters*, page 54, September 1995.

[4] J. K. Ousterhout. *Tcl and the Tk Toolkit*. Addison-Wesley, New York, 1994.