PERGAMON

Computers & Graphics 23 (1999) 277–286

COMPUTERS & GRAPHICS

Chaos & Graphics

Iamascope: a graphical musical instrument

Sidney Fels[a,*], Kenji Mase[b]

[a]*Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada, V6T 1Z4*
[b]*ATR MI&C Research Laboratories, Seika-cho, Soraku-gun, Kyoto, Japan, 619-02*

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.

Keywords: Electronic kaleidoscope; Interactive multimedia; Contactless machine interface; Virtual musical instrument; Video-to-texture mapping

1. Introduction

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 as shown in Fig. 1. The types of reflections possible with the computing machinery are more extensive than are possible with mirrors with effects 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 human computer interaction, 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.

* Corresponding author. Tel.: + 001-604-822-5338.
E-mail address: ssfels@ece.ubc.ca (S. Fels)

0097-8493/99/$ - see front matter
PII: S0097-8493(99)00037-0

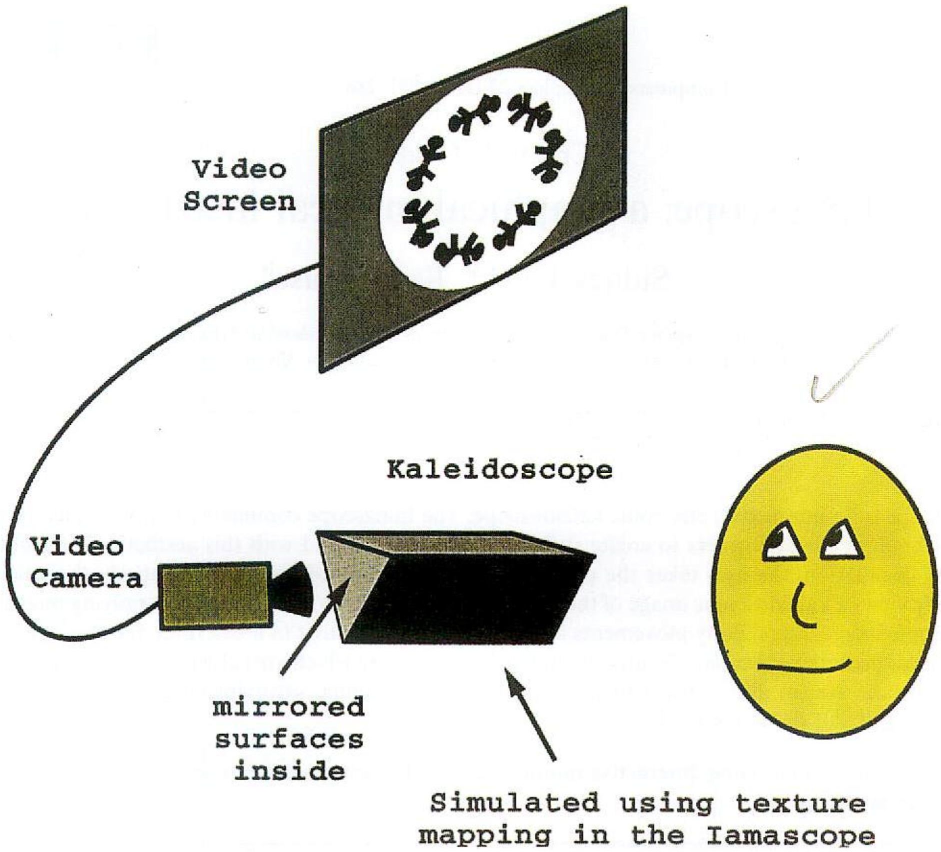

Fig. 1. Putting a kaleidoscope in front of a video camera. This will provide an interactive kaleidoscope like the Iamascope; however, the reflections possible will be limited to those available with real mirrors. The Iamascope uses texture memory to simulate the mirrors. Notice, that putting a kaleidoscope in front of a video monitor fed from a video camera also provides an interactive kaleidsocope; however, the user has to be looking through the eye-piece all the time.

The computer graphics and the image processing techniques are not particularly advanced from a scientific perspective. What is of interest is the advance in providing an aesthetically satisfying interaction technique with computer graphics. According to John Bates, the Silicon Surfer:

> This is the first computer installation that I have tried that worked this well, gave me a very satisfying and creative experience and which actually got me emotionally involved with it. It is like playing a very satisfying and responsive instrument.[1]

The interaction techniques used to create the Iamascope provide fuel for pursuing novel directions to create new interaction techniques. Many parts are required to create the interaction including: balancing computer graphic and sound qualities, providing real-time, responsive feedback in both media, providing the *correct* mapping type between body and media, and understanding aspects of intimacy with the interface [1]. From a more pragmatic perspective, the Iamascope has direct application as both an artwork for enhancing artistic creativity of people and a graphical musical instrument for performers to master for dance and musical performance. The Iamascope is a *graphical musical instrument* since performers play the Iamascope to control both imagery and music.

This paper describes the Iamascope system, consisting of the kaleidoscope sub-system and the vision-to-music sub-system which allows for matched musical accompaniment with the kaleidoscopic imagery. In the first section, an overview of the Iamascope is provided. Following, is a detailed discussion of the kaleidoscope sub-system and the vision-to-music sub-system. Finally, related work, conclusions and future directions are discussed.

[1] See http://www.radionet.com/surf/surf970807.html.

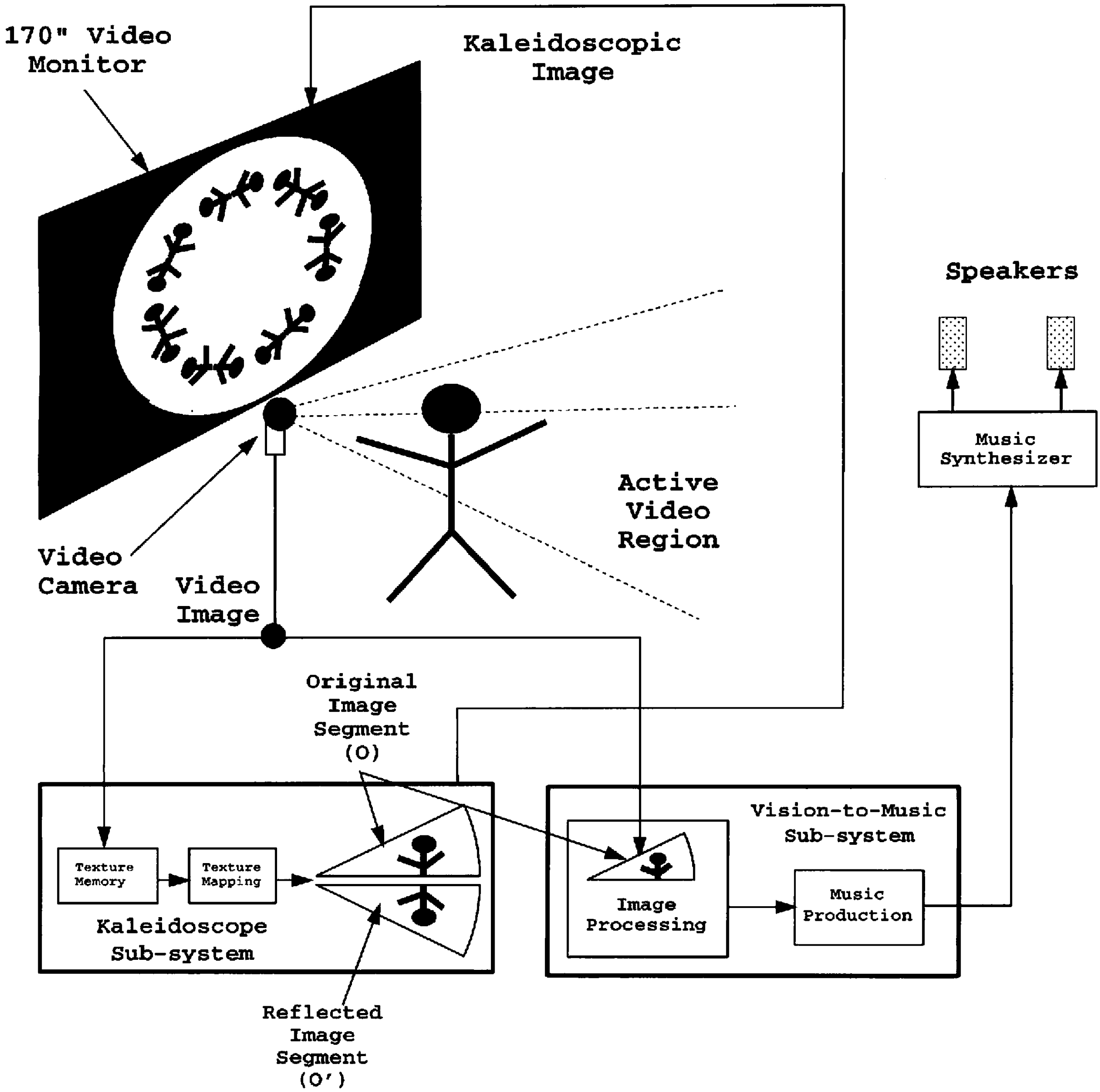

Fig. 2. Block diagram of the Iamascope. Output from the video camera feeds into both the kaleidoscope sub-system and the vision-to-music sub-system. The kaleidoscope sub-system places the video image into texture memory and extracts the root image to form the kaleidoscope. Texture hardware performs the mirror reflections and the output is displayed on the large screen. The vision-to-music sub-system uses the same root image as input and performs motion analysis on sub-sections of the root image. The output from the motion analysis is processed and converted to MIDI signals to control a MIDI synthesizer. The kaleidoscope sub-system runs at 30 fps on an SGI O2 R5000 and the vision-to-music sub-system runs at 15 fps on an Indy.

2. Overview of the Iamascope

A block diagram of the Iamascope is shown in Fig. 2. 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 in) video monitor. Audio output from the Iamascope is played though stereo speakers beside the large video monitor. In our current implementation,

Fig. 3. Example of a person enjoying the Iamascope.

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 Fig. 3.

The kaleidoscope sub-system and the vision-to-music sub-system are all written in C with a Tcl[2] front end and a Tk based interface [2]. The kaleidoscope sub-system runs on an SGI O2 R5000 at 30 fps. The vision-to-music sub-system runs on an SGI Indy (or an O2) at 15 fps 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 subsystem 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.

3. Kaleidoscope sub-system

The kaleidoscope sub-system in the Iamascope uses a single video camera as input. The video input feeds directly into texture memory. Once in texture memory, the root image for the kaleidoscope is extracted. Creating mirror reflections using texture mapping is straightforward; only a reversal of the texture indices is required. The flexibility of texture memory allows many types of kaleidoscopic effects to be created easily and in real-time.

The typical kaleidoscope used in the Iamascope is based on the reflections found in a two-mirrored kaleidoscope since, in our experience, it provices the highest level of interactivity. 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

[2] Tcl is a simple extendible, interpreted, scripting language and Tk is an extension to Tcl that is a GUI toolkit.

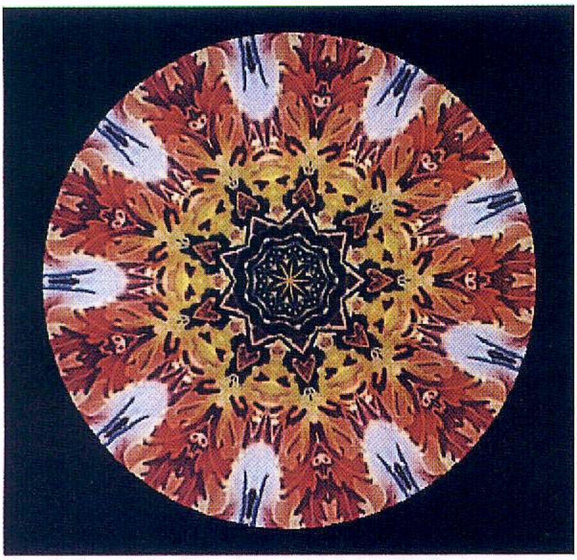

Fig. 4. Snapshot of an image created with the two mirror kaleidoscope.

alternation of the original image and its mirrored reflection (as illustrated in Fig. 2). 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 Fig. 2. For example, if we use a 20 degree pie slice then there will be 18 segments which make up the circular image. The odd segments will have the original image and the even ones will have the mirror reflection. An example of the two-mirrored reflections produced by the Iamascope is shown in Fig. 4. 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 centre 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]

The Iamascope currently supports several other types of reflections. The main types of reflections include:

1. *Three-mirror kaleidoscope*: The symmetry with a three mirror kaleidoscope creates an image which extends to infinity, filling the whole screen. The angles of the three mirrors form a triangle and can be adjusted. An example of images with this kaleidoscope is shown in Fig. 5.
2. *Three-mirror kaleidoscope on a sphere*: Typically a three mirror kaleidoscope image extends to infinity. However, by drawing the reflected triangles on a sphere composed of many triangles, the image can be "wrapped" around the sphere. One interesting effect with this mapping is that the triangles are not completely uniform around the whole sphere, thus, the texture mapping stretches the image appropriately causing some distortion. Fig. 6 shows an example of such an image. Additionally, the shape of the triangles can be adjusted for different looking symmetries. The sphere can also spin.
3. *Three-mirror kaleidoscope on a sphere from the inside*: The same as the previous reflection except you view the reflections from inside the sphere.
4. *Four-mirror kaleidoscope*: The symmetry with a square arrangement of mirrors extends to infinity. This arrangement is not particularly appealing or interactive.

Of course, other tilings and symmetries may be used for different effects, such as found in [3]. An excellent discussion about some of the possibilities for computer based kaleidoscopes can be found in [4]. The key point for a beautiful effect is to make sure that the reflections always line-up, that is, the reflected image is always adjacent to the original so that the point at which they join is identical on each side. This makes the reflections seamless. An added feature of using texture mapping is that the textures will be distorted to fit the polygon that the texture is being mapped on to. So, as long as the texture indices are set up to line up the reflection with the original, the texture mapping will distort the image to make sure there are no seams. These non-uniform reflections add to the possible types of surfaces, tilings and mapping shapes that can be done with the Iamascope.

Another useful parameter which may be controlled is the rate of rotation of the image.[4] This is equivalent to rotating the eyepiece of a kaleidoscope and keeping the objects in the viewing chamber fixed. The effect this has is to create a smoothly time-varying kaleidoscopic image. This is not conducive to interaction since the location of the extracted image is changing. As the root image rotates, the user must try to track it. This is difficult. However, it is very useful as an "attract" mode since the

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

[4] On the O2 we rotate the image, on the Onyx we rotate the position for extracting the image.

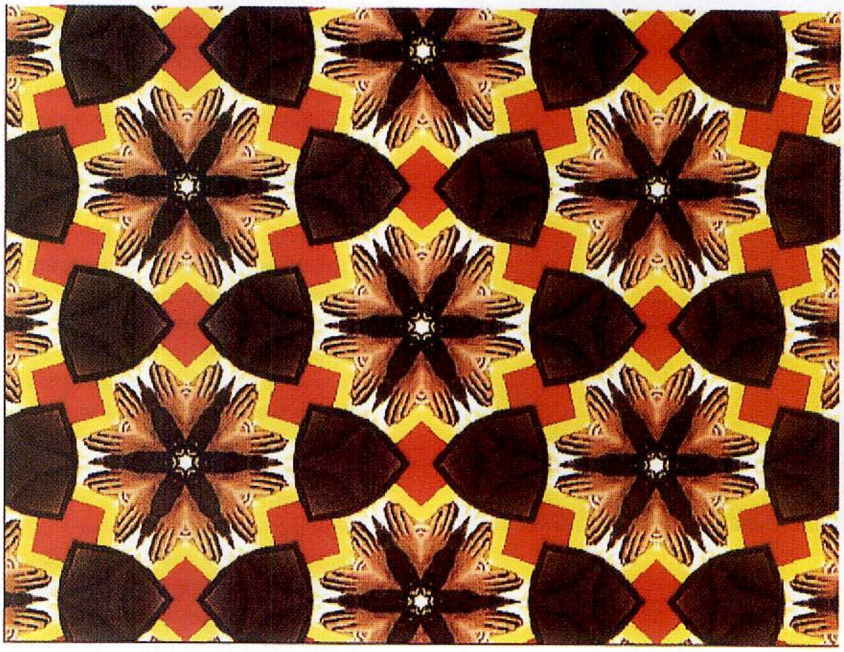

Fig. 5. Snapshot of the three mirror kaleidoscope image. Notice, there is a performer's hand in the image.

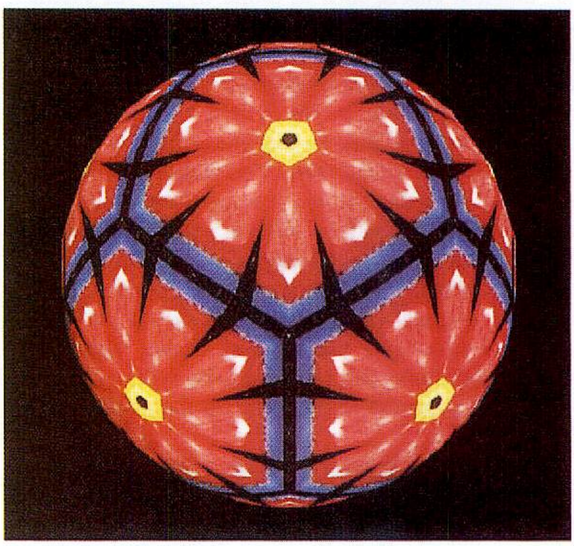

Fig. 6. Snapshot of the three mirror kaleidoscope wrapped around a sphere. The rpm of the sphere can be adjusted.

image is dynamic. Typically, when there is no user movement detected in the Iamascope for sometime, the Iamascope automatically begins to rotate the image slowly. As soon as movement is detected, the standard two mirror arrangement with a static root image is re-started.

In addition to the above, other controls are available which can be interactively exploited. The Iamascope has controls for:

- changing root image background mixing colour; this allows for the kaleidoscopic image to be shaded,
- image brightness,
- background; usually in interactive mode we use a black background, but in attract mode an interesting background is used to paste the two-mirror based circle on,
- starting position for drawing the root image; this allows the symmetry lines to be oriented. For example, with a 20° angle there are 18 lines of symmetry. By starting the root image so that one edge is vertical, other symmetry folds will also be vertical, dividing the image in half (this parameter was suggested by an anonymous user at Siggraph97).
- inlay; the Iamascope can display the whole image captured by the video camera, along with the current root image outlined in red, in a small sub-window. This can be helpful to orient the user.

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

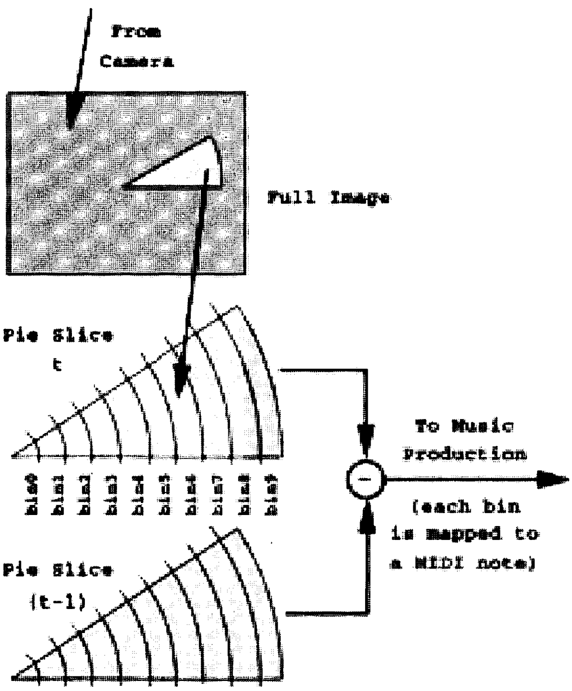

Fig. 7. Diagram showing image processing in the vision-to-music sub-system.

into an "attract" mode which shows a dynamically changing image. Currently, we randomly cycle through some of the interesting reflection options while spinning the image at a slow rate. 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.

4. 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.

4.1. Image processing

A block diagram of the image processing system is shown in Fig. 7. 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 Fig. 2) is used to create the kaleidoscopic image. The same pie slice is extracted by the image processing system. The pie slice is then divided into bins as shown in Fig. 7. In each bin, the average intensity difference is calculated from the grayscale values of each pixel using:

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

where $g_p(t)$ is the grayscale value of pixel p at time t. At 640×480 image resolution and pie slice of 20°, 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.

4.2. 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. Many schemes are possible for musical control based on the input from the image processor. We chose a production scheme that did not require any absolute positioning of the body and would play euphonic music to match the beautiful kaleidoscope images. Within these constraints there is room for some musical control and expression by the performer, though, it does not provide as much as a physical musical instrument such as a guitar. Viewing the Iamascope as an interactive artwork, the scheme used is suited for the creation of the aesthetic experience. When viewed as an instrument for performance, the current scheme is best for novices. Currently, we are exploring other forms of musical control within the Iamascope framework that will allow more complex musical interaction and expression.

In the current system, each bin of the intensity differences from the image processing stage 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 selected to match the instrument being used: for example, for a piano the "note OFF" signal is sent 250 ms 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., 2000 ms. 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 independent 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 Csus4" with the key change set to occur every 2.5 s. 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 [5].

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 sub-system so it can return to its normal mode (see Section 3).

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 sub-system also connects to the vision-to-music server as a Tcl client, sending and receiving Tcl commands.

4.3. Related work

Work related to the Iamascope falls into two categories; systems which use image processing to control music and/or graphics and research on different ways to support musical performance to lower cognitive load on performers.

There are numerous systems which use vision processing to control imagery and/or music including: DanceSpace [6], the Mandala system from The Vivid Group, the Very Nervous System by David Rokeby, Recollections by Ed Tannenbaum, MIC Exploration Space [7] and Liquid Views [8]. Further, considerable research has been done in creating novel musical interfaces [9].

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 pre-recorded 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 [2] 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.[5] 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 Iamascope, 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 [6] and Brush de Samba [10].

5. Conclusions and future work

The use of textures to create a flexible, interactive kaleidoscope is very successful. This simple mechanism provides an exciting interactive art piece for anyone to enjoy. The Iamascope system was displayed in the Electric Garden at Siggraph97 [11] and enjoyed by more than one thousand people. This interactive artwork demonstrates an effective synthesis of texture mapping techniques for mirror reflections in real-time and image processing for musical accompaniment.

The simple kaleidoscopic reflections and image processing techniques used in the Iamascope allow user's to enjoy and express themselves as the explore the possibilities of the Iamascope. The key features which make it a successful interactive multi-media experience are that it

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

While the current kaleidoscope sub-system is very interactive and satisfying there are a number of interesting directions to explore for new forms of expression. These include:

1. non-linear reflections,
2. mapping reflections onto other 3D objects,
3. separating user from background and mixing image with computer graphics for the root image,
4. merging images from different cameras for remote, cooperative Iamascopic experiences.

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 properties 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 pursuing 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.

Acknowledgements

We are grateful for the contributions made by Dirk Reiners, Takumi Tadashi and Ryohei Nakatsu for assisting with this project.

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

References

[1] Fels SS, Mase K. Intimacy as an index for understanding human-computer interaction. Proceedings of the Workshop on Interactive System and Software. December 1997:211.

[2] Ousterhout JK. Tcl and the Tk toolkit. New York: Addison-Wesley, 1994.

[3] Glassner A. Andrew glassner's notebook: a periodic tiling. IEEE Computer Graphics and Applications. 1998;18:83–90.

[4] Pickover C. Mazes for the mind: computers and the unexpected. New York: St. Martins Press.

[5] Fels SS, Nishimoto K, Mase K. Musikalscope: a graphical musical instrument. Proceedings of IEEE International Conference on Multimedia Computing and Systems (ICMCS'97), June 1997:55–62.

[6] Sparacino F. Choreographing media for interactive virtual environments. Master's Thesis, Media Arts and Sciences, MIT, 1996.

[7] Sommerer C, Mignonneau L. Mic exploration space. Visual Proceedings of SIGGRAPH'96: The Bridge. 1993:17.

[8] Fleischmann M, Bohn C, Strauss W. Rigid waves – liquid views. Visual Proceedings of SIGGRAPH'93: Machine Culture. August 1996:124.

[9] Paradiso J. Electronic music interfaces: new ways to play. IEEE Spectrum Magazine 1997;34:18–30.

[10] Kotani A, Maes P. An environment for musical collaboration between agents and users. Proceedings of Lifelike Computer Characters. September 1995:54.

[11] Fels SS, Reiners D, Mase K. Iamascope: an interactive kaleidoscope. Visual Proceedings of SIGGRAPH'97: The Electric Garden. August 1997:76–7.