Monalisa: “see the sound, hear the image”

Kazuhiro Jo
RCAST
University of Tokyo
4-6-1, Komaba, Meguro-ku, Tokyo
jo@jp.org

Norihisa Nagano
Studio 2
IAMAS
3-95, Ryoke-cho, Ogaki City, Gifu
nagano@monalisa-au.org

ABSTRACT

Monalisa is a software platform that enables to "see the sound, hear the image”. It consists of three software: Monalisa Application, Monalisa-Audio Unit, and Monalisa-Image Unit, and an installation: Monalisa “shadow of the sound”. In this paper, we describe the implementation of each software and installation with the explanation of the basic algorithms to treat the image data and the sound data transparently.

Keywords

Sound and Image Processing Software, Plug-in, Installation

1. INTRODUCTION

Many artists and researchers have tried to "see the sound, hear the image”. Kandinsky corresponded the timbres of musical instruments and colors. Based on the correspondence, he produced many paintings borrowing motifs from traditional european music [7]. John Whitney created number of motion graphics based on the theory of musical harmony [16]. With *Slow Scan*, Laurie Anderson recorded the visual information as sounds and reconstructed these images by playing the *Tape Bow Violin* [1]. *VinylVideo* also recorded images as sounds into LP analog records. It replays the moving image from the LP record with ordinal record player and a custom hardware [12]. Xenakis developed a computer system *UPIC* to allow the composer to draw music with a graphics tablet. The drawing is immediately calculated and transformed into sound by the computer [17]. *MetaSynth* translates static images into sound. It enables people to draw sound by graphically editing the spectrum structure of sound [13].

The software platform, Monalisa, also tries to "see the sound, hear the image” by treating all the image and the sound as the sequence of numbers. In contrast to such previous works, it limits its target to the image data and the sound data represented by binary codes. It consists of three software: Monalisa Application, Monalisa-Audio Unit, and Monalisa-Image Unit, and an installation: Monalisa “shadow of the sound”. Monalisa Application is standalone software that transparently treats the image data and the sound data. With this software, people open a sound data as an image data and vice versa. Sound and image effects could be added to the data. Monalisa-Audio Unit is plug-in software for sound processing applications. It wraps existing image effect plug-ins as sound effect. Monalisa-Image Unit is plug-in software for image processing applications. It wraps existing sound effect plug-ins as image effect plug-ins. Currently these software works on Mac OSX. OSX offers basic API (Application Programming Interface) for image processing and sound processing as Core Image and Core Audio [2]. Monalisa Application, Monalisa-Audio Unit, and Monalisa-Image Unit use these API as a basis of their image and sound processing. Monalisa "shadow of the sound" is an installation. It consists of custom version of Monalisa applications and a set of projector, switch, camera, speaker, and microphone situated in a room. In this installation, people hear the sound of their image and see the sound.

2. RELATED WORKS

Monalisa treats the image and the sound as the sequence of numbers represented by binary codes. *SonART* [18], the framework for image and data sonification, also offers to use data from images for sound synthesis or audio signal processing and vice verse. It is designed as standalone application and could be communicate other image and sound applications through network. *SoundHack* is a sound-file processing program [5]. Though it limits its target only for sound processing, its function "open any" allows reading image data as sound data in the same way as Monalisa application.

Because of the use of Core Image API, Monalisa Application and Monalisa-Audio Unit employs GPU (Graphics Processing Unit) for its sound processing. Several work have been doing in this area. Gallo and Tsingos investigate the use of GPU for 3D audio geometric calculation [6]. Whalen examines using pixel shaders for executing audio algorithm [14]. Zhang et al. implemented modal synthesis by using the parallelism and programmability in graphics pipeline [13]. As Monalisa software indicate alternative use of the existing effects for image and sound processing, these instance indicate alternative use of the existing hardware for sound processing.

3. ALGORITHM

Monalisa treats the image and the sound as the sequence of numbers represented by binary codes. The target image is a bitmap data that consists of pixels and the target sound is a linear monaural PCM (Pulse Code Modulation) data that consists of samples. In order to treat the image data and the sound data transparently, we have developed following two algorithms: *8-bit* and *24-bit* those sequentially transform pixels of image data into samples of sound data and vice versa.

3.1 8-bit

8-bit is an algorithm for Monalisa application, Monalisa-Image Unit and Monalisa "shadow of the sound". It treats each RGB color value of pixel of bitmap data as an 8-bit sample of linear PCM data. A bitmap image data consists of pixels. Each pixel has three-color values: R (red), G (green), B (blue) and each value is defined within a certain range. We sequentially treat each color value (the order is R, G, B.) as a separate 8-bit data from the top left pixel to the down right pixel of the image data (see Figure 1).

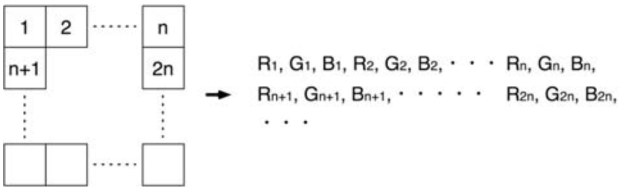
1
2
n
n+1
2n
R1, G1, B1, R2, G2, B2, • • • Rn, Gn, Bn,
Rn+1, Gn+1, Bn+1, • • • • • R2n, G2n, B2n,
• • •

Figure 1. Treatment of image data

A linear PCM sound data consists of samples. Each sample stands in a line at uniform intervals (i.e. sampling rate) and is defined within a certain range. We sequentially treat each sample as a separate 8-bit data from the start to the end of the sound data (see Figure 2).

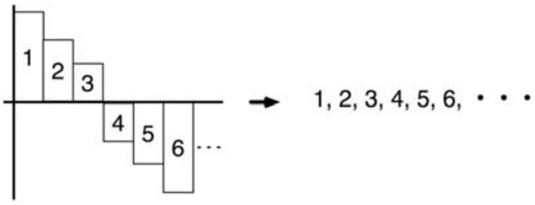

Figure 2. Treatment of sound data

In this algorithm, we exchange the 8-bit color value for 8-bit sample and vice versa. Each color value of a pixel is separately treated as a sample of a sound data and each sequence of three samples is treated as three color values (R, G, B) of a pixel (see Figure 3).

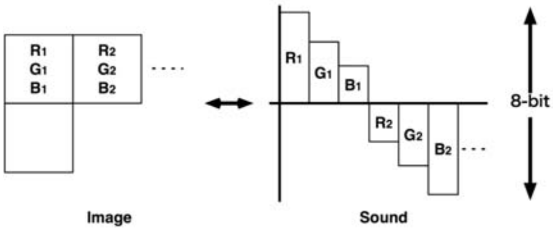

Figure 3. 8-bit algorithm

3.2 24-bit

24-bit is an algorithm for Monalisa-Audio Unit. It treats three-color values of pixel of bitmap data as a 24-bit sample of linear PCM data. We sequentially treat each color value (the order is R, G, B.) of a pixel as a combined 24-bit data from top left pixel to down right pixel of the image data. As the sound data, we sequentially treat each sample as a separate 24-bit data from the start to the end of the sound data. In this algorithm, we exchange the 24-bit color value for 24-bit sample and vice versa. Three-color values of a pixel are treated as a composite sample of a sound data and a sample of a sound data is treated as three decomposed color values of a pixel (see Figure 4).

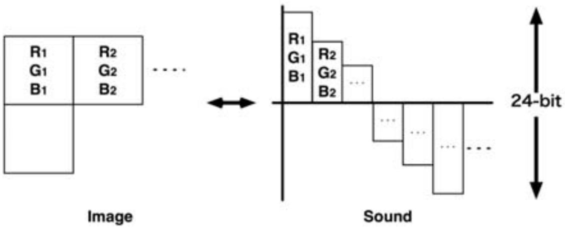

Figure 4. 24-bit algorithm

4. Monalisa: "see the sound, hear the image"

4.1 Monalisa Application

Monalisa Application is standalone software that transparently treats the image data and the sound data. Monalisa Application uses Core Image and Core Audio API as a basis of its image and sound processing. The application enables people to open any types of image data and sound data that the API supports. The data is treated as both image and sound. Based on the 8-bit algorithm, the image data can be played as sound and the sound data can be showed as image. These API also provide access to system-level plug-in architecture: Image Unit and Audio Unit. By using these plug-in, Monalisa Application offers to add image and sound effects for the data. We briefly describe two examples of the effect for the data: invert the sound, and delay the image.

Invert the sound: Invert is an image effect that inverts each color value of pixels. In Monalisa Application, each sample of the sound data is treated as separate 8-bit color value. By adding invert to the sound, each sample of the sound data is inverted within a given range (8-bit) and forms a phase-reversed sound from the original (see Figure 5).

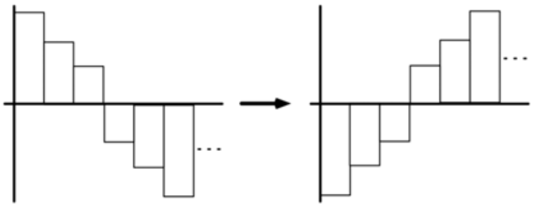

Figure 5. Invert the sound

Delay the image: Delay is a sound effect that delays each sample of the sound data with a given interval and compounds it to the original data. In Monalisa Application, each color value of pixels is treated as separate 8-bit sample. By adding delay to the image, each color value of pixels is delayed with a given interval and added to the original image (see Figure 6).

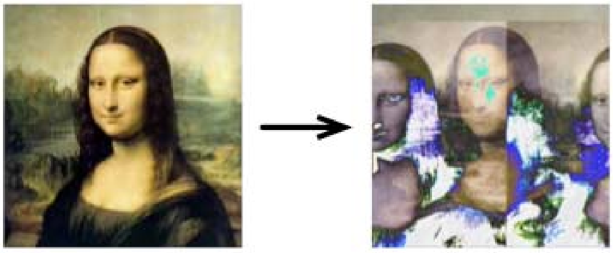

Figure 6. Delay the image

4.2 Monalisa-Audio Unit

Monalisa-Audio Unit is plug-in software for sound processing applications. Currently it works on Audio Unit host applications on Mac OSX (e.g. Apple GarageBand, Apple Logic Pro). It enables people to add several kinds of image effects for the sound data in real-time by wrapping existing Image Unit plug-in as Audio Unit plug-in. In Audio Unit host applications, the plug-in behaves as a single Audio Unit plug-in. By using this plug-in, the sound data is split into separate bitmap image by given buffer size along to time line. Based on the 24-bit algorithm, each sample of the sound data is treated as a pixel of a 24-bit image data. Image effects (i.e. Image Unit plug-ins) can be added to the sound data through Monalisa-Audio Unit. Several kinds of Image Unit plug-ins are pre-installed on Mac OSX. Each pixel of the processed image is re-treated as a sample and formed a new sound data in real-time.

4.3 Monalisa-Image Unit

Monalisa-Image Unit is plug-in software for image processing applications. Currently it works on Image Unit host applications on Mac OSX (e.g. Pixelmator, Apple Motion). It enables people to add existing sound effects for the image data within several applications by wrapping existing Audio Unit plug-in as Image Unit plug-in. It works not only for the static image data, but also for the motion graphics data by treating it as a collection of static image. The plug-in works with standalone software Monalisa-Image Unit Generator. The software separately wraps each existing Audio Unit plug-in as an Image Unit plug-in. In Image Unit host applications, each wrapped Audio Unit plug-in behaves as a single Image Unit plug-in. Based on the 8-bit algorithm, each color value of the pixel is treated as an 8-bit sample of the sound data. Audio effects (i.e. Audio Unit plug-ins) can be added to the image data as standard Image Unit plug-ins. Several kinds of Audio Unit plug-ins are pre-installed on Mac OSX. Each sample of the processed sound is re-treated as a color value of pixel and formed a new image data in real-time.

4.4 Monalisa "shadow of the sound"

Monalisa "shadow of the sound" is an installation that represents the essence of Monalisa application. It was premiered at Open Space at NTT InterCommunication Center from 9th June 2006 to 11th March 2007. A total of about 10,000 people participated in the work during the exhibition period. It consists of custom version of Monalisa applications and a set of projector, camera, microphone, switch, and speaker situated in a room (see Figure 7).

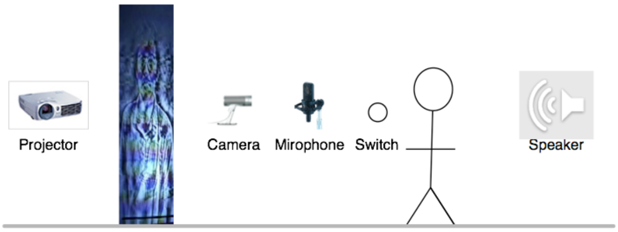

Figure 7. Setting of Monalisa "shadow of the sound".

When entering the room, each participant saw his/her image projected on the screen. When he/she pushed the switch, the image was captured as a static bitmap image data and the light of the room was gradually decreased to the darkness. The image data was transformed to the application and automatically played as a stream of sound through the speaker. Then, the microphone captured the stream of sound and transformed it to the application. The application re-projected the incoming sound data as an image on the screen from top left to down right of the screen. In this installation, the re-projected image reflects the reverberation of the room as duplications of the image (see Figure 8).

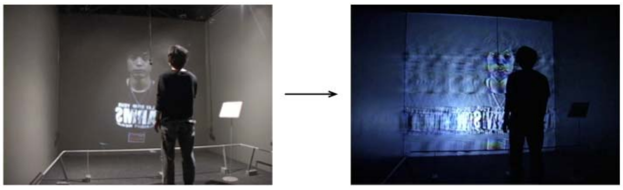

Figure 8. Original image and re-projected image.

We have developed two custom versions of Monalisa application. One for image capture and sound production, and another for sound capture and image production. Both applications were installed in different PC. The first one (PC1) was connected to the camera, the projector, and the speaker. It captured the image, and played the image as the sound. The other one (PC2) was connected to the microphone and the speaker. It captured the sound, and projected the sound as the image. To control the trajectory of the installation, we employed MaxMSP [4] in other PC (PC3). It was connected to the switch, the light, and a video switcher. When it received a signal from the switch, it sent a capture message for PC1 through Open Sound Control (OSC) [9]. The light was controlled by DMX controller through MIDI. When the light turned to the darkness, PC3 sent a play message for PC1, a capture message for PC2, and switched the video switcher from PC1 to PC2. After PC2 projected whole captured sound, PC3 increased the light to ordinal level (see Figure 9).

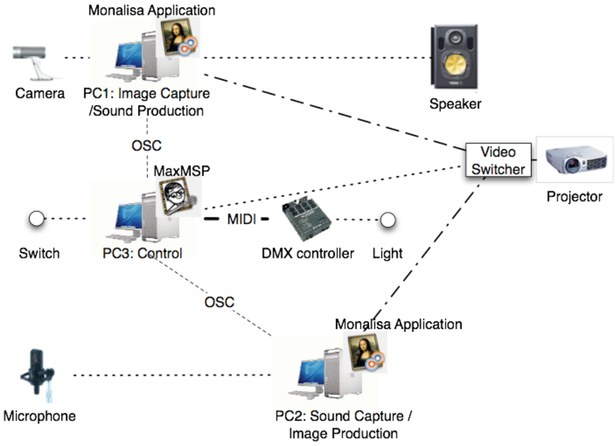

Figure 9. System of Monalisa "shadow of the sound".

5. DISCUSSIONS

In Monalisa, image data become sound data and vice versa. As Whitelaw cited from the email of Christopher Sorg, "all data inside the computer is essentially the same, ... either with ears or eyes, or whatever senses we care to translate the switching of 1s and 0s into..." [15]. While sonification or visualization lay emphasis on the use of sound / image to help a user monitor and comprehend whatever it is that the sound / image output represents [8], our software platform gives people new modes of manipulation to employ the sound data and the image data as materials to produce their own creative works. Sometimes, such unintended use may results in horrible noise, while other times it can produce wondrous tapestries of sound [3]. The software platform also enables people to use existing image plug-ins in existing sound processing applications and vice versa. This experience suggests us to access old media objects in new ways congruent with information interfaces we use in our everyday life [10].

We have developed two algorithms: 8-bit and 24-bit. While 8-bit provides one to one relationship in each color value and sound sample, 24-bit provides one to one relationship between each pixel and sample. Therefore the result of adding effects for data has difference in each algorithm. For instance, with 8-bit, the resulted sound will be low bit like old hip-hop sample. In contrast, 24-bit retains the resolution of sound data which most of existing sound processing application provides. We think their distinguished characters and the variety of expressions are not a trivial function. Therefore we plan to provide the selection of two algorithms in our future release.

Due to the technical limitation, we currently fixed the buffer size of 4096 in Monalisa-Audio Unit. It limits the range of transformation and inhibits to treat whole sound data if it is not consists of 4096 samples. Therefore we also plan to add adjusting mechanism for buffer size in future release.

While situating the installation Monalisa "shadow of the sound", we have conducted several observations. Due to the limited space, we briefly introduce two following observations.

Figure changes the sound: The resulted sound was produced from the image of the participant. Therefore, the figure of the participant affects the character of the sound, for example, white T-shirt produced higher frequency and border of striped shirts made kind of rhythmical sound.

Sound / Image equipment affects image / sound: We have tested several video cameras, microphones, speakers and lights. When we changed these equipments, the quality of the sound equipment affects image and vice versa. For instance, the sensitivity of the video camera affects the spectrum of the sound and the directivity of the microphone affects the clearness of the image. These characters of Monalisa show potential as alternative tools to check the quality of sound and image equipments.

The possibility of computational process as a material for artistic creation [9] is not fully investigated yet. We are interested to explore the alternative way for image and sound processing. We anticipate that our initial explorations of the software platform stimulate new ideas for the instruments for image and sound productions.

6. ACKNOWLEDGEMENTS

This work was developed under the support of FY2005 IPA Exploratory Software Project (Project Manager: KITANO Hiroaki) provided by Information-Technology Promotion Agency (IPA) and NTT InterCommunication Center. We also would like to thank Nao Tokui and Kazuo Ohno for their valuable comments and Karl D.D. Wills for his excellent graphic design.

7. REFERENCES

[1] Anderson, L. THE RECORD OF THE TIME Sound in the Work of Laurie Anderson, NTT Publishing Co., Ltd., 2005.

[2] Apple, Core Audio, Core Image, http://www.apple.com

[3] Cascone, K. The Aesthetics of Failure: "Post-Digital" Tendencies in Contemporary Computer Music. Computer Music Journal, Vol. 24, No. 4, December 2000, pp. 12-18.

[4] CYCLING74. MaxMSP, http://www.cycling74.com/products/maxmsp

[5] Erbe, T. SoundHack: A Brief Overview, Computer Music Journal, Vol. 21, No. 1, Spring 1997, pp. 35-38.

[6] Gallo, E. and N. Tsingos. Efficient 3D Audio Processing with the GPU. In GP2, ACM Workshop on General Purpose Computing on Graphics Processors, 2004.

[7] Kandinsky, W. Concerning the Spiritual in Art, New York: Dover Publications Inc, 1977.

[8] Kramer, G. (ed.) Auditory Display - Sonification, Audification, and Auditory Interfaces. Addison-Wesley, 1994.

[9] Maeda, J. Design by Numbers, Cambridge, MA: MIT Press, 1999.

[10] Manovich, L. The anti-sublime ideal in data art. http://www.manovich.net/DOCS/data_art.doc, 2002.

[11] OSC, Open Sound Control, http://cnmat.cnmat.berkeley.edu/OSC/

[12] Sengmuller, G. VinylVideo TM, Leonardo - Volume 35, Number 5, October 2002, pp. 504-504.

[13] Wenger, E. MetaSynth, http://metasynth.com

[14] Whalen, S. Audio and the Graphics Processing Unit. http://www.node99.org/projects/gpuaudio/gpuaudio.pdf, 2005.

[15] Whitelaw, M. Hearing Pure Data: Aesthetics and Ideals of Data-Sound, in Arie Altena (ed.) Unsorted: Thoughts on the Information Arts: An A to Z for Sonic Acts X, Amsterdam: Sonic Acts/De Balie, 2004.

[16] Whitney, J. Digital Harmony: On the Complementarity of Music and Visual Art, McGraw-Hill, Inc., New York, NY, 1981.

[17] Xenakis, I. Formalized Music, rev. ed. Stuyvesant, New York: Pendragon Press, 1992.

[18] Yeo, W. S.,, Berger, J., and Lee, Z., SonART: A framework for data sonification, visualization and networked multimedia applications, in Proceedings of the Internaional Computer Music Conference, Miami, FL, USA, November 2004.

[19] Zhang, Q., Ye, L., and Pan, Z. Physically-Based Sound Synthesis on GPUs, Entertainment Computing - ICEC 2005, Springer Berlin / Heidelberg, Volume 3711/2005, pp. 328-333.

8. APPENDIX

Monalisa application is under development for next release. Monalisa-Audio Unit and Monalisa-Image Unit are downloadable from following URL.

http://nagano.monalisa-au.org/?page_id=351

A Japanese techno musician Junichi Watanabe employs Monalisa-Audio Unit to produce his latest album "LITTLE SQUEEZE PROPAGANDA" (ADDL-004, AsianDynasty, 2007).