Christopher Keyes
Department of Music
Hong Kong Baptist University
224 Waterloo Road
Kowloon, Hong Kong
ckeyes@hkbu.edu.hk

An Introduction to ArtsSync: A New Tool for Mapping Foreground, Middle-ground and Background Musical Structures to Visual Images

Abstract

In the growing trend towards the extension of computer music to include computer graphics several primary issues become immediately clear: a) how exactly and on what level(s) to correlate the audio and graphical elements, b) the limitations inherent in the vast majority of easy-to-use software for this aim, and c) the amount of time and technical competence required to master the more low level programming environments towards this aim. This paper/demonstration will focus on these issues and how they are addressed in the "ArtsSync Project", an extensible environment for correlating audio and graphical components of multi-media projects. Its primary emphasis addresses the problems of mapping different graphical elements to different layers of musical structures, achieving the immediate moment-to-moment excitation of images while simultaneously allowing global parameters to map to medium and longer timescale structural levels of a particular composition.

1. Introduction

One can not deny the current trend towards the extension of computer music to include computer graphics. Computer music festivals that formerly would have only a few multi-media works now typically devote several concerts to them, and computer music journals are ever increasingly accompanying their issues with articles about this new 'visual music' and with accompanying DVDs. The medium has certainly captured the imagination of many computer music enthusiasts and for a variety of reasons. Perhaps the most persuasive is the great potential for enhancement to the concert experience that such graphical elements provide. Principal reasons must also include the possibilities and sheer practicality of having both music and images rendered in the digital domain, and being able to work on them with only a computer and no other expensive hardware. More recently, the integration of both sound and image creation & processing tools in the same software environments seem an especially compelling opportunity to pursue such work, despite the challenges which present themselves.

Of course composing music to accompany images (and drama) is nothing new. Apart from Opera and pieces such as Mussorgsky's *Pictures at an exhibition*, composers around the clock and around the globe are constantly working to put music to Hollywood-style films and other commercial video. What is relatively new is the central role composers now play in the process and the non-narrative context and content of much of the video elements. Continuing along the lines of Kandinsky's non-representational abstract art pieces of the early 20th century (interestingly occurring at roughly the same time as Schoenberg's a-tonal music) many current 'visual music' pieces are inherently abstract and non-narrative, almost like abstract Operas not dominated by the characters themselves. Along with this often comes the intention of composers not to allow the visual images to significantly detract from the audiences' perception and attention to the audio components.

While this presents great freedom on the one hand it poses some hard questions on the other. If the music is not intended to accompany and/or enhance the narration of a film or drama, just what will be the correlation between the audio and visual elements? On a purely practical level Jones and Nevile (2005) underscore that "Although new software tools enable vast possibilities for the correlation of audio and visual elements they also demand new conceptual approaches as well as a new level of technical competence on the part of the artist". Despite this the artistic possibilities continue to allure more and more of us towards this medium. As Roger Dannenberg notes: "Real-time interactive computer music and animation programs are fascinating…and making these programs easier to create is important so that more composers and performers can apply their talents to this interesting new art form" (Dannenberg 1993). This is precisely what the *ArtsSync* project attempts to address.

2. Background and Challenges

In terms of software, currently there exists a plethora of easy-to-use 'sound visualization' and 'VJ' (video jockey) software used for synchronizing audio and video. The visualizers of the Microsoft Window's *Media Player* and Apple's *iTunes* are probably best known and display a common weakness: although they do react spontaneously to amplitude and frequency characteristics of incoming audio, the images hardly vary from one piece of music to another. Although the iTunes visualizer does include some counterpoint of different graphical elements which can be immediately attractive, the various effects are largely not controllable and certainly not mappable to the specific structures of a given composi-

tion. The results seem more like screen-savers than multi-media art. They are also limited to renderings of abstract images. Many of the VJ applications such as *Almost Sync* or *Motion Drive* are more sophisticated in that they can process incoming and saved video streams, but they tend to focus on the most popular mixing techniques and are not very flexible in capturing a range of *vector fields* (Zettle 1999) required for more creative kinds of pieces. All are of course closed-ended platforms that can not be modified.

At the other extreme there are the graphical programming environments such as *MaxMSPJitter, DIPS, GEM* and *EyesWeb*, a real-time interactive software currently being developed at the *InfoMus Lab Laboratorio di Informatica Musicale* which emphasizes motion capture. Although all of these are highly flexible software environments, in the end they are only "a collection of objects" which require considerable investments in time to learn to use. This is sometimes not practical for all of those who are not already familiar with such programming environments (or programming itself) but who are interested in pursuing interactive projects. This is one hurdle *ArtsSync* is meant to cross.

In terms of art, finding multi-media works with compelling music and compelling images working together is sometimes challenging. One common problem lies distinctly in the correlation between visual and musical elements. Formerly the visual elements are realized using a completely different software environment from the audio. This then leads the composers/musicians tied to the pre-existing and fixed temporal elements of the video. One solution to this is to compose rather temporally ambiguous music, an approach that does work and may allow for some degree of musical spontaneity. It lacks though the synergetic "magic" that comes from the exact correlation of multiple elements on a millisecond time scale. Another often unsatisfying aspect of many of these works is the lack of *depth* in the video, the counterpoint of different simultaneous musical ideas either juxtaposed or superimposed with foreground, middleground, and background elements.

Norman McLaren (Miller, G. 1981) a pioneer in pre-computer animation achieved great results, for his time, and often wrote and correlated his own music for his films. Unfortunately, he was not trained composer, which mars some of his pieces, and the technical quality of the images (flicker, jitter etc.) lies far below current standards. His work does though utilize both elements effectively. John Whitney's works also deserve special attention. *Copper Island* (Burns 2003**)** and *FAKTURA* (Miller, D. 2003) are recent works highly regarded by many in the field. For many they have both compelling images and music, however the correlation between the two are limited.

2.1 The Rift

What these previous approaches illustrate is a rift between the immediacy of capturing the moment-to-moment audio data versus the structural mapping of other aspects of a composition that occur over a longer timeframe. In fact when critiquing iTunes/Microsoft-Windows Media Player-type visualizers Roger Dannenberg concludes "There is a common temptation to draw connects between music and image at a very superficial level...as soon as the obvious connections from sound to image are made the image ceases to be interesting or challenging. A better approach…is to make connections between deep compositional structure and images" (Dannenberg 2005). Taking perhaps the opposite view, Jones and Nevile (2005) write: "In both audio and visual domains as more elements are added the listener/viewer becomes less able to devote attention to them all simultaneously. This isomorphism…indicates that mapping single musical voices in a composition to a single graphical element is a good approach for making visual experiences that can accompany music in a meaningful way."

3. Our Approach in *ArtsSync*

Our intention in creating *ArtsSync* is to do both; creating an easy-to-use environment which achieves the immediate moment-to-moment excitation of abstract renderings, movies, still images, and live camera feed while simultaneously allowing all parameters to map to longer timescale structural levels of a particular composition. In our prototype interface the length of the piece is input, and various parameters can be chosen to respond to various characteristics of the analyzed audio or change dynamically by a series of break-point curves where the shape can be clearly seen and the time scales calculated according to the length of the piece (see figure 1). Another advantage of this approach is that if one inputs a length that is shorter than the actual composition, one can see the entire graphical components on a much shorter time scale, fast-forward if you will, which facilitates the fine tuning of parameter changes over the larger time scale.

The results are highly animated visual components with a very high degree of correlation to various levels of the audio components, and thus provide potential for enhanced concert experiences. The concerns are to avoid the pitfalls of static 'screen-saver' images of most 'visualizers' as well as images and music that seem poorly coordinated.

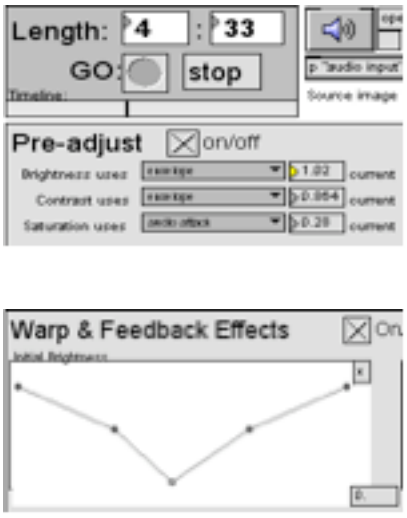

Figure 1. *GUI with length of the piece (upper left) and breakpoint curves for values*

3.1 Architecture

The *MaxMSPJitter* programming environment was used for the software implementation of *ArtsSync*. An extendable, open architecture was constructed in order to ensure flexibility and longevity; new effects modules can easily be added by third parties with *MaxMSPJitter* programming skills using a developers' template. The *ArtsSync* architecture also automatically adapts itself to a user-definable image resolution, so that the software can take advantage of future advancements CPU/GPU clock speeds to create arbitrarily high resolution output images.

The *ArtsSync* patch itself is divided into three main sections: *Controllers*, *Image Generators* and *Effects*.

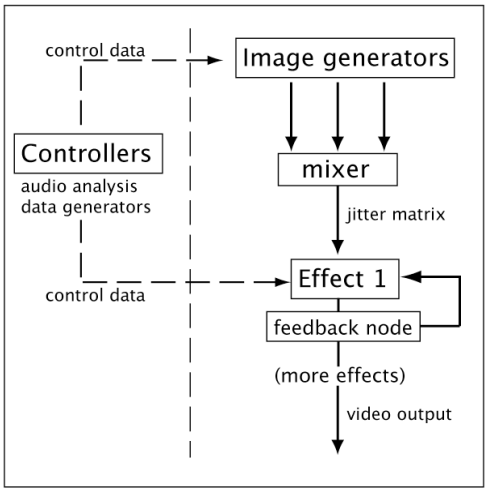

Figure 2. *Overview of the* ArtsSync *MaxMSPJitter patch architecture, showing the connections between the three main sections.*

The *Controllers* section consists of modules which generate control data to be sent to any module in the other sections. There are two types of controller modules: *audio analyzers* and *data generators*. *Audio analyzers* look at the incoming audio input and do real-time analysis to extract interesting information. At present, several spectrum-based amplitude followers and peak detectors are implemented, and there are plans to add more analysis modules in the near future. *Data generators* include the aforementioned break-point curves, MIDI controllers and oscillators, as well as random and chaos number generators. The output of all the modules in the *Controllers* section, regardless of the source (i.e. analysis or generator), is normalized to the range 0. to 1. (In the case of peak detectors the *bang* message is used.) This control data is then scaled to a relevant range at the destination module in one of the other sections. This standardization of control data is an essential first step in assuring the high degree of correlation sought between the audio and graphic elements. It also gives the user a chance to experiment freely, as control data can be switched from one module to another without having to manually recalculate scaling parameters.

The *Image Generators* section consists of modules which can translate control data from the *Controllers* section into video images. A simple example is the *solid color* module, which can be set to generate a solid colored frame, the intensity of which can be influenced by the amplitude of the incoming audio stream or other controller. Another example is the *12 Band to radial module*, which takes the amplitude data from the *12 Band envelope follower* analysis module and transforms it into a circular pattern which reacts to changes in the spectrum of the sound. Other *Image Generators* include a QuickTime image player and two different waveform generators. Since *ArtsSync* is built upon an open-ended architecture, there is no limit to the number of *Image Generator* modules that can be implemented. There are currently several other modules in development, and custom modules can easily be added by a third-party programmer. And since each module can be activated or deactivated individually, unused modules can be switched off to conserve CPU power. At present, three *Image Generator* modules can be active at any given time, and mixed using a simple cross fader or by means of a variety of arithmetic operations. Figure 3 shows an example of two generator modules, a simple waveform generator and a Quicktime image player, blended together using an additive operator.

The *Effects* section is where transformations are performed on the source image generated in the *Image Generator* section, using parameters received from the *Controllers* section. There are at present several effects modules implemented, ranging from simple brightness/contrast/saturation adjustments to radical image warping effects. Once again, there is no limit to the number of effects modules that can be implemented in the future, and individual modules can be activated or deactivated to conserve CPU power.

Figure 3*. Two frames of video output from ArtsSync, demonstrating the coupling of audio amplitude with a brightness/contrast effect module.*

A significant feature of the *Effects* section is the implementation of several image feedback nodes. The video stream can be "tapped" from any of six effects modules and sent back into the stream at an earlier stage in the effects chain. When combined with a rotation/zoom effect module, this enables users to create the familiar *iTunes*-like abstract 'visualizer' effect as shown in Figure 4. Since feedback nodes are available at every effect insert in the video stream, highly complex, dynamic images can be created using a minimum of CPU resources.

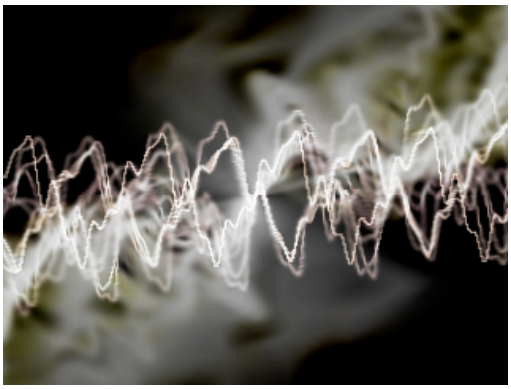

Figure 4. *The familiar iTunes "visualizer" effects can be easily created in ArtsSync; however now the user has absolute control over all effects parameters.*

4. Applications

4.1 Of Space and Time

Of Space and Time (2004), a multi-media work for percussion, real-time interactive graphics and multi-channel audio is one of my first works to use the ArtsSync application. It incorporates many of the same abstract waveform effects used in the iTunes software. However, changes in color, feedback, brightness, image warp evolve over time matching the rising and falling 'dramatic curve' of the music, following its climaxes and lulls. Other parameters, such as color, reflect different overlapping musical structures in the composition. As delays with feedback are used in the audio portion in some places, the feedback nodes mentioned above are employed on the visual component and seem to compliment each other well. Of course the waveforms follow the frequency and amplitude of the live player allowing for the spontaneous visual connection on the shortest millisecond timescale.

4.2 Orion

As with *Of Space and Time*, my composition *Orion* achieves additional correlation of audio and graphical elements by using similar and/or analogous processing techniques on both visual and audio components. As a simple example I used a deep space image taken by the Hubble Space Telescope of the Orion Nebula, which appears as a massive and colorful gaseous cloud. For solo piano (equipped with MIDI sensors) the audio component consists of two main elements: a subtle processing of the piano's acoustic sound with reverberation and slight harmonization (to match the gaseous colors of the nebula) and two sets of audio samples triggered by the piano's MIDI output. These sampled sounds, using equal-tempered 24 and 36-tone tunings to contrast with the piano's sound, are projected around the audience within an octaphonic loudspeaker array, appearing to be in constant motion.

The corresponding graphics portion is also composed of two elements. The first element is the actual photo of the Orion Nebula. Similar to the reverberation process which 'scatters' the piano's sound off of virtual surfaces, the nebula image is subjected to the subtle 'sprinkling' effect of a matrix operator that uses probability functions to scatter matrix cells, displaced by a random amount which is determined by the amplitude of the piano's incoming audio stream. The result is a noisy "cloud" of data during loud sections contrasting with quieter sections where the image has little or no processing. The second element are two waveforms (the familiar oscilloscope-like writing of the audio data into the matrix with feedback etc.) that react to the various frequencies present in the incoming audio stream. The position of the two waveforms on the projection screen however would also be in constant motion and use the same position data of the sampled sounds in the speaker array, thus being exactly synchronized. As a result of using the same position data, the visual and audio are quite similar and thus lead to a high degree of correlation between the two elements.

5. Conclusion

In working on multi-media pieces using the ArtsSync Project's program we have found it invaluable and expediant towards the creation of works that corellate different musical structures, with dispariative timescales, to a variety of graphial elements. We also find that the use of the MaxMSPJitter environment, along with the open architexture of the program itself, greatly aids in the addition of newer modules and the adaptation of the program to a vareity of new projects. It is hoped and planned that such an open architecturewill allow for its continued development towards the extension of computer music to computer graphics.

References

Burns, K. 2003. "Copper Island" *Computer Music Journal*. DVD 27, 4. 3.

Dannenberg, R. 1993. "Software Design for Interactive Multimedia Performance" *Interface Journal of New Music Research*. 22, 3. 213-228.

Dannenberg, R. 2005. "Interactive Visual Music: A Personal Perspective" *Computer Music Journal*. 29, 4. 25–35.

Jones, R. Nevile, B. 2005. "Creating Visual Music in Jitter: Approaches and Techniques" *Computer Music Journal*. 29, 4. 55–70.

Miyama, C. Rai, T. 2003. "Introduction of DIPS Programming Technique" Proceedings of the *2003 International Computer Music Conference*. San Francisco: International Computer Music Association. 459-462.

Miller, D. 2003. "FAKTURA" *Computer Music Journal*. DVD 27, 10.

Miller, G. 1981. "The Eye Hears, the Ear Sees" (video recording) : Norman McLaren, film maker / B.B.C. [and] the co-operation of the National Film Board of Canada; written and directed by Gavin Millar; producer, Margaret Dale Pub info (Sandy Hook, Conn.): Video Images.

Zettle, H. 1999. *Sight Sound Motion*. 3rd ed. New York: Wadsworth.