From the visualization of sound to real-time sonification: different prototypes in the Max/MSP/Jitter environment

Anne Sedes, Benoît Courribet et Jean-Baptiste Thiébaut

Centre de recherche en Informatique et Création Musicale (CICM),

Université de Paris VIII - MSH Paris Nord – France
{asedes,bcourribet,jbthiebaut}@mshparisnord.org

Abstract

This paper presents current aesthetic and software development research by the authors on the theme of visualization of sound.
After describing general trends and strategies in the field of sound/visual relations, the problem of the representation of time is evoked. Sound visualization is then considered as an interface for sound design and interactive music composition through the presentation of software currently in development, using time-frequency analysis or terrains in 3D immersive environment as tools for manipulating sound.

Introduction

We consider the interest of visualizing sound in the context of visual software interfaces, as well as in the interactive arts. Hence, we may imagine a visual space in two dimensions, presented as an overview or in a subjective view, understood as the control space of different dimensions of a sound space. Sound object may be represented as spheres, or color region may visualize parametric variation zone. A controller, as a keyboard mouse, or any joystick may then allow exploring sensitive variation of sound and music.
Beyond scientific visual representations in one hand, and imaginary, poetic and literary illustrations on the other hand, new approaches of sound visualization are emerging due to our relation with the computer science tools, evolving on the mode of emulation and transduction between the man and the machine. In this context, the temptation of real time sonification of image produced by visualization of sound is emerging.

1. About the visualization of sound

We are going to describe several approaches and strategies generally used in order to give a visual representation of a sound or a musical work. We have decided to sort these approaches considering the way the representations are linked formally to the music itself. By "linked formally", we mean that the visual and audio parts are sharing data in a way or another. In the following section, we will not be discussing the goals that may lead to these strategies, but rather describing their "modus operandi".
First, we can think about the visual representation of a sound analysis or even the audio signal data itself. The result of the sound analysis is simply displayed with only a few possibilities for changing its visual properties (color scales, range…): for example, the waveform of a sound or its spectrum. It is important to notice that it is possible to use either a still image or a series of images in time (a video): displaying a waveform the way an oscilloscope does or STFT (Short-Time Fourier Transform) frames instead of the spectrum of the whole sound. Dynamic images are more suited to render the dynamic dimension (which is fundamental) of sounds: with a still image, it is possible to foresee what is going to happen and to look "back in time" whereas dynamic images refer to the transiency of the sound phenomenon.
Then, still close to the representation of the audio signal data, we find the so-called "visualizations" in software such as Winamp or iTunes: one of the representations described above is used as a raw material then processed with visual filters (blur, geometric deformations, feedback processes…). Most of the time, even if the sounds are really different, the visual results are similar from one to another.
Still dealing with sound analysis, we have the possibility not to display the result of the analysis by itself, but to use this result in order to generate visual data or to modify a pre-existing video. For example, in the Egosound software (Sedes, Courribet, Thiebaut. 2003), the amplitude of a sound is controlling the radius of a 3D OpenGL sphere or the saturation of its color. Another example would be the detection of transients into a music work in order to trigger video processing: a pre-existing video would be processed *according to* the music. The central concept of *mapping* rises from this approach: how will we share data between the domains of sound and video, which relationships do make sense? These questions are being extensively studied

thanks to the Max/MSP/Jitter environment, which makes it possible to perform both audio and video processing within the same software.

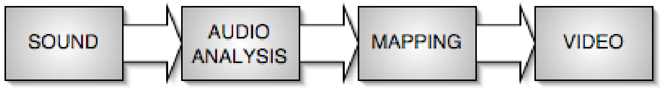

Figure 1 : a sequential approach to sound/visual relations

It is also possible to share data between these two domains without the sequential aspect described previously: global parameters can be sent in parallel to an audio unit and to a video processing unit. The *Egosound* software is a good example of this possibility: the cartesian coordinates of a virtual sound source (we will call them "intentional" parameters) are sent both to an audio spatialization engine and to an OpenGL renderer, so that the software displays what the user would see (the moving sound source) if the sound was visible. Benoît Courribet also used an audio/video granulation engine for his project Isol with percussionist Julien Tardieu: each time a grain of sound was produced, a 3D structure was deformed on the screen.

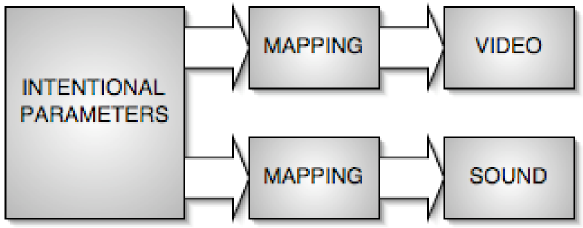

Figure 2 : a parallel approach to sound/visual relations

Finally we can consider many possibilities where no formal link exists between the audio and video parts: the most obvious example in this case is the pop music video clip. Assuming one decides whether or not the video is going to match with the sound characteristics, the results range from short plotted movies with no apparent link to the music (Radiohead's Karmapolice, for example) to extreme synchronization to the music like in Autechre's Gantz Graf video clip. It's interesting to notice that in the last example, it's hard to find out whether or not a formal link exists. Director Alex Rutterford has created this video clip, synchronizing 3D structures frame by frame according to the sonic events. It appears to be a visual annotation of a musical work, in the same way that it's possible to draw colored shapes and pictures on top of a sonogram in the GRM's Acousmographe software: though the sonogram represents a canvas, it is possible for the user to draw whatever shape he wants.

We have to note that one of the main problems occurring when trying to visualize sound is the choice of a representation for time. Time is often represented in an explicit way as a dimension of the image (usually horizontally from left to right, like in a music score), which modifies our perception of the music. Because all evolutions past and future can be seen at the same time, the mechanism of protensions and retensions described by philosopher Husserl is altered when the timeline is explicit.

2. visual interface of sound

We have described many ways to give visual representations of sound or music, but it seems like we have focused on either analytic or artistic ways to consider the sound-visual relations. But what we will discuss now is the possibility to create a visual representation of a sound in order to use it as an interface for interactive music composition or audio signal processing.

We have seen previously that we can give what we will call an "intentional" visual representation of a sound by displaying graphically some parameters used for sound generation, audio signal processing, or even structural composition. Now we can think about manipulating this visualization in order to interact with the sound or music.

Following the concepts developed under the term of WaveTerrain Synthesis, we have decided to represent parameters by 2D monochrome terrains, the value of the parameter being mapped with the intensity of the colour. These terrains are then superposed and used as textures for a ground in a 3D virtual environment.

An image can be seen as the superposition of three planes, one for each of the red, green and blue (RGB) channels. Because the Jitter environment uses this formalism, we have decided to control three parameters, each one being linked to one of the RGB planes.

Hence, by moving virtually on this ground, the user is controlling the evolution of many parameters, outputting the values at the position where he is located. For every trajectory made on the ground, the user imprints his timeline, experiencing the structure of a piece actively and interactively. Though the 3D movement algorithm is used here with mouse control, it has been tested with many sensors (ultrasound, the "leaning chair" of installation La Terre ne se meut pas…) and we are willing to use it with motion capture systems.

Below, we can see a trajectory drawn on a terrain (clockwise from top-left corner) and the resulting evolutions of parameters (intensity of the colour, and amplitude/time representations) for each plane.

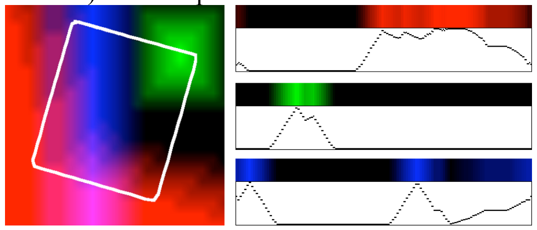

Figure 3: a trajectory on a terrain leads to unique parameter evolutions

Many software can be used to design terrains, we have developed our own interface in order to create terrains easily by drawing directly onto them:

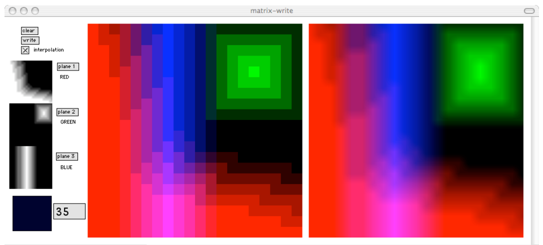

Figure 4: our terrain design software, on the right, interpolation is used.

Anne Sedes and the choreograph Laurence Marthouret work with such an interface to control adaptive sound spaces in real time for the stage.

3. The *Sonos* software

We will now present a software named *Sonos*. The background of this work comes partly from the field of graphical synthesis (Curtis Roads 1996), which took several forms. First of all, the UPIC system (Unité Polyagogique Informatique du CEMAMu) conceived by Iannis Xenakis and associates researchers in 1977, which has evolved until now as real-time UPIC. In 1993, Vincent Lesbros developed the Phonogramme software. And recently, the Metasynth software (U&I software 1997) which is advertised by : “Paint with sound, compose with light”.

Sonos is a real-time interactive software. It is a visual interface based on the STFT analysis. The aim of *Sonos* is to transform the sound helped by a graphical interface of the sound itself after analyze. In order to manipulate the sound graphically, we apply a STFT to a stream of input samples and draw it on a visual window. Scaled to a logarithmic view, the sonogram is then graphically modifiable. The resulting matrix of this transformation is then synthesized. The real-time quality of this software is conceptually complex, as the time itself can be modified with a scaling of its axis. According to the transformation, the output sound may have a completely different temporality. The gesture control is real-time: a modified parameter causes immediate transformations of the sound.

3.1. A time / frequency / amplitude representation

These programs propose a time / frequency / amplitude representation of the sound and graphical function to draw the sound in time. The conceptual problem is to map a graphical gesture to a musical result, to transpose musical aesthetic consideration to a visual interface. This approach is highly creative but somehow conducts the user to draw a STFT. Influenced by real-time emerging interface like the one we presented earlier, our goal is to provide a more visual control of the sound synthesis.
The basic material of the process is the sound itself. The image on which we apply graphical transformation is a STFT representation. We are more dealing with digital effects than with synthesis. This working progress software has two main methods. The whole process is described in figure 4.

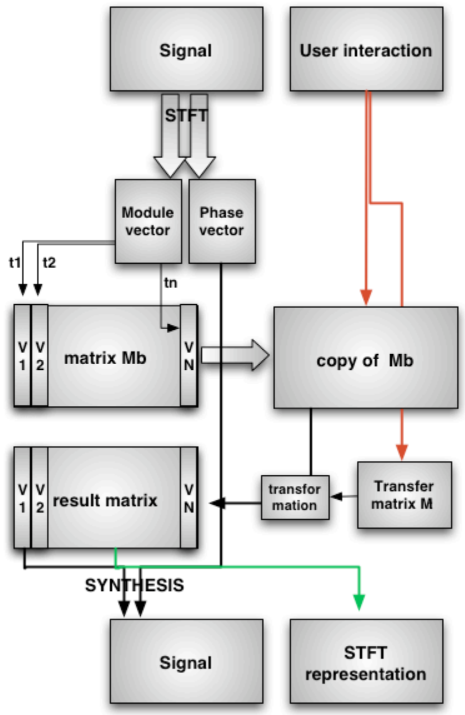

Figure 5 : signal and data flow in the Sonos approach

3.2. First transformation method: dealing with sound like with image

As the STFT represents amplitude, frequency and time, the transformation of this representation would necessary modify amplitude, frequency or time. This kind of transformation has been implemented with a phase vocoder (PV). Regarding sound like an image allows transformations that have not been implemented with a PV.

Considering that the sum of n vector of the STFT constitute a matrix, *Sonos* applies transformations to a whole set of vector. The basic sound is sequentially written in the matrix. For the display the matrix is resized to a logarithmic view so the user can act on the image the closer to its perception. The operations on the sound are inspired by the manipulation of images in the Jitter environment. Among them we focused on the rotation, zoom, blur and saturation. The rotation of the matrix has a progressive shift effect on the partials which creates a kind of glissando. Its duration depends of the size of the matrix. When the rotation is equal to $\pi/2$, the continuous partials of the basic sound become a noisy attack, and the transient attack becomes a continuous sound. This causes a deep modification of the time-domain of the input sound that is considered in the new system as a frequency axis. The horizontal zoom-in and zoom-out functions occur timestretch effects while the vertical zoom shift the frequency (Luke Dubois 2003).

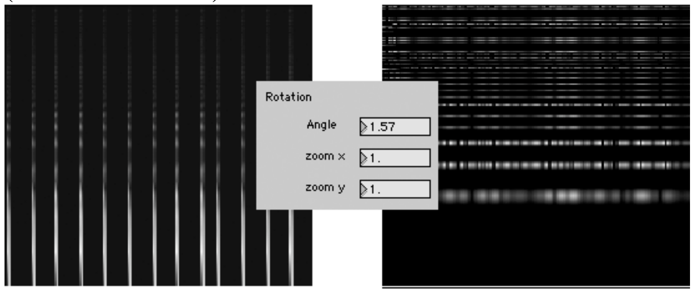

Figure 6 : ***The left image represents the STFT of a repeated digital click. Transient attack becomes continuous sound on the right image after a π/2 rotation. the abscissa axis represents time, the ordinate represents frequency.***

Some other graphical transformations are implemented in *Sonos* like blur, interpolation or saturation. They have sensible effects on sound.

3.3 Second transformation : graphical transformations using a transfer matrix

The second approach of graphical transformations regards the use of matrixes as control interface. Each plane of a matrix is relied to a transformation, and each pixel stores a value. The user colors the matrix to perform transformations. The horizontal axis represents the time domain, and the vertical axis represents the frequency domain. The drawing matrix is a transfer matrix which is supplied by the basic visualization image and provide a new matrix (i.e. a new sound and a new image). The transfer matrix is sixteen time smaller than the basic matrix. The goal of this reduction is to get a processor gain and to divide the frequency bands into 32 in place of 512. The reduction is applied to the time domain as well. It implies that a transformation is applied to a time band of 371ms. Indeed, each vector corresponds to 11.6 ms of signal, i.e. 512/44100, and 11.6 * 32 = 371.

For the frequency domain, the transfer matrix is scaled according to an exponential law of the form :

$$f(x) = c.2^{k.x}$$

with *c* and *k* constant.

We present here the implementation of two transformation planes. The first transformation consists in a frequency filter. The value stored in each pixel of the plane is a 8-bits value, of a range of [0..255]. This value is scaled to [0..1] and is multiply by the value contain in the bins of the corresponding frequency band. The result, which is the new module of the STFT is displayed in real-time. A cursor shows the current position of the synthesis.

The second transformation uses the green plane. It implements a frequency delayed line. The intensity of the color of a bin is also scaled to [0..1]. The value is then interpreted as a delay parameter which allows to repeat a part of the signal corresponding to the colored bin. The delayed sound is then delayed and added to the signal of the following bin.

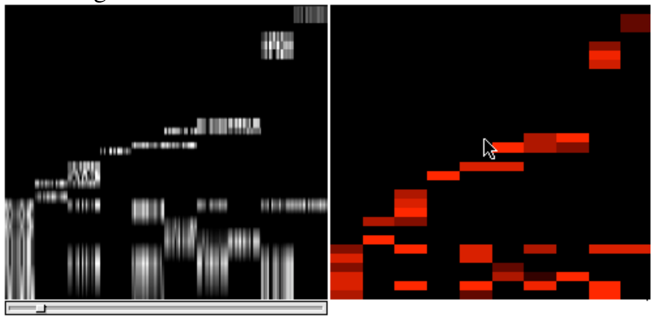

Figure 7 : the right image is the filter interface. The left image is the result of the basic sound filtered by the transfer matrix

The transfer matrix supplies the delay values. These values are multiply with a copy of the resulting matrix, which provide a new matrix with the delayed signal. This matrix is then shifted of the size of a bin and added to the result matrix. Using a transfer matrix to transform a STFT avoid the problem encountered with the different graphical synthesis software: draw a STFT to create sounds. Furthermore, the potential offered by Jitter to control image in real-time provide many ways of manipulating sound with such a process.

Conclusion

We have evoked here some approaches of the visualization of sound. We have talk about the visualization as an interface for the sound's parameter. The work in progress project *Sonos*, in the continuity of the graphical synthesis, propose to integrate real-time interaction in the synthesis process and in sound transformation. Perspectives in interactive arts and inter-media are numerous. They could also concern the development of interfaces and operational tools for work, sequence and mix sound and multimedia images.

References

Dubois, L. *Jitter_pvoc_2D* (2003). Max/MSP Abstraction
Lesbros, V. *Phonogramme* (1993). Software.
Husserl, E. Leçon pour une phénoménologie de la conscience intime du temps (PUF 1964)., PARIS
Raczinski. J.-M., G. Marino and M.-H. Serra. 1991. "New UPIC system demonstration." In B. Alphonceand B. Pennycook, eds. *Proceedings of the 1991 International Computer Music Conference*. San Francisco : International Computer Music Association. pp. 567-570.
Roads, C. *Computer Music Tutorial* (1996). MIT Press.
Sedes, A. , Courribet, B.,Thiebaut J.-B. (2003). "Egosound, an egocentric interactive and real-time approach of sound space". Proceedings of the DAFX-03, London.
Sedes, A. , Courribet, B.,Thiebaut J.-B., (2003). "Visualisation du sonore avec le logiciel egosound: work in progress". Actes des journées d'informatique musicale 2003, Montbéliard
Sedes, A. (2003). "Espaces sonores, actes de recherches", éditions musicales transatlantiques, Paris, 2003.
Wenger, E. *Metasynth*. 1997. Software, U&I Software.