USING VISUAL TEXTURES FOR SONIC TEXTURES PRODUCTION AND CONTROL

Jehan-Julien Filatriau
Communications and Remote Sensing Lab
Université Catholique de Louvain
Louvain la Neuve, Belgium
filatriau@tele.ucl.ac.be

Daniel Arfib
LMA-CNRS
31, Chemin Joseph Aiguier
Marseille, France
arfib@lma.cnrs-mrs.fr

Jean-Michel Couturier
Blue Yeti
68, avenue de la Grande Conche, 17200 Royan, France
jmc@blueyeti.fr

ABSTRACT

This work takes place in the framework of a global research on the synthesis of sonic textures and its control through a gesture-based interaction in a musical practice. In this paper we present different strategies to link visual and sonic textures using similar synthesis processes; theoretical considerations underlying to this problematic are firstly exposed and several personal realizations, illustrating different approaches to design a gesturally controlled audio-visual system, are then described.

1. INTRODUCTION

Sound production and gestural control have their own specificities and belong to a field named "sound and music computing". It is interesting and useful to make a cross-comparison with other fields such as vision, perception or emotion. This paper is an attempt for the cross fertilization of different fields: visual textures making has an incredible advance in processing techniques, from which we can benefit. But also concepts behind vision can be applied to sound as soon as they are correctly transposed. The main direction of our research is to see if analogies can be done in the design and use of visual textures and the design and use of sonic textures. It completes previous articles [1] [2] which have described the algorithmic part of textures making. Our motivations behind this paper is then twofold: in chapter 2 we describe a personal view of these analogies between visual and sonic textures, supported by examples of the literature or our own prospects; in chapter 3 and 4 we present examples of the applicative research of the three authors, packaged in a way that distinguishes research done on static and dynamic images. As we will see we push forward the notion of a "malleable texture" which is especially handful for gesture control.

2. BASIC FOUNDATIONS

Before describing experiments we have carried out, we give in this section some generic considerations about the link between image and sound: firstly, we will explain why starting from visual textures making techniques could be relevant for the generation of sonic textures. Then, we will show that many techniques have been developed by the computer graphics community, enabling the synthesis of a wide range of visual textures. Finally, we will expose different strategies that can be carried out in order to link image and sound in a common generative process.

2.1. The bridge between visual and sonic textures

This bridge can be done at a technical level, of course, and next section will provide examples of such kind. But the link is also at a conceptual level of perception and recognition. The fact is that visual textures can be perceived as such once certain statistical facts arise from them; this is the perceptual fact. Among all these textures one can recognise different kinds of textures and label them; this is the recognition part. These two parts are a great importance in presence, a notion developed also by HCI and which deals with the contact with environment. All of this is also true for sonic textures: they have auditory clues that make them be textures, they are distinguishable and are part of the discriminating process of hearing in the environment.

Computer science provides us some techniques, sometimes called procedural, in order to synthesize images and sound. Are these techniques equivalent? There is still a great deal of research on how to transpose a set of methods from a field to another, and in our case this is amplified by the fact that the visual and sonic object or features are not the same. Their common ground is the mobility (a video is a moving image, a sound is a perseverance of a sonic stimulus along time) and the controllability. The malleability of visual or sonic forms can be seen from two points of view: either the object is malleable or it is fixed, but the point of view is different. Such analogies are not only words, but will be demonstrated in the image-to-sound correspondence. Sonic textures have specificities one does not have to miss: they rely on time, and sonic clues are very dependent upon the hearing process. As an example the sound of rain has a rhythmic component associated with a resonance for each drop; a static boring sound is not equivalent to the profusion of a texture image. So one has to be very careful while designing an image-to-sound system.

2.2. How to make visual textures

2.2.1. What is a visual texture?

The multiple definitions found in the literature present visual textures (Fig.1) as spatially homogenous and typically containing repeated structures subject to some random variations. Bar-Joseph highlights another important characteristic [3]: using any window of size larger than some critical size, the "information content" exhibited in the window is invariant to the window position within the given sample.

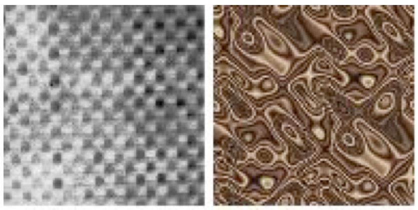

Figure 1: *Examples of natural textures (at left) and synthetized textures (at right).*

2.2.2. *Visual textures synthesis*

Visual texture synthesis has received an increasing attention from the computer graphics research community over twenty years. The objective is to generate, from an original sample, an endless output texture perceptually similar to the reference image. First attempts on texture synthesis focused on the development of well-known procedural textures [4]. After these, methods starting from the analysis of an input image have been carried out: those techniques model visual textures as sample from probabilistic distribution [5] and more recently as Markov Random Fields [6]. A lot of techniques dedicated to the synthesis of animated textures have also been developed: some of them extend techniques used for fixed images [3][7], while other methods rely on more or less complex physical models to render dynamic natural phenomena like waterfalls, fire, or waves motion [8]. Advances in these synthesis algorithms coupled to increasing possibilities offered by computer science have made possible high quality synthesis of fixed and time-varying visual textures. These images are potentially interesting candidates for an image-to-sound process based on static image. However those methods are generally computationally intensive and not adapted to real-time constraints, which make them not suitable for a dynamic image-to-sound process. For this aim, an alternative solution may be offered by fractal textures, described in the next section.

2.2.3. *Fractal textures*

A fractal is a rough or fragmented geometric shape that can be subdivided in an infinity of elements, each of which being, at least approximately, a reduced-size copy of the whole. Fractals are especially used in computer modeling of irregular patterns and structures in nature such as clouds, mountains, or branches of trees that do not correspond to simple geometric shapes. Fractals present some characteristics that make them close to visual textures: indeed fractals are generally self-similar, independent of scale and often referred to as "infinitely complex".

Iterated Function System (IFS) is one of the simplest ways to generate fractal images [9]. Fractals of this type are created by applying a series of affine transformations to an initial point through a number of iterations. A weighted probability factor is associated to each transformation, in order to favor certain configurations during the iterative process. From a geometrical point

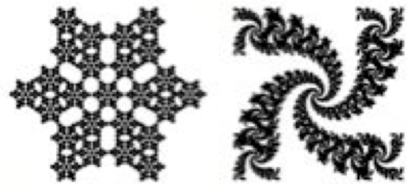

Figure 2: *Examples of IFS fractal textures.*

of view, this process corresponds to a combination of elementary transformations (translations, rotations, compression and shears) allowing after few iterative cycles to build up a large range of fractal shapes like trees, spirals or snowflakes (Fig.2). The complexity of these algorithms depends on the number of points composing the image and the number of iterations of the process; small values of these parameters allow to compute and render in real-time quite complex fractal shapes. Furthermore, by varying parameters describing affine transformations, one can easily distort these shapes and obtain animated image. This is the strategy we have adopted in one of our image-to-sound instrument, as described in section 4.2.

2.3. How to link visual to sonic textures

There is no one-way procedure to link vision and sound. We have developed three points of view, starting from the "image from a sound" thinking, going to the "pixel image sonification" and ending with the "equivalence of the processes". Each of them gives a territory for the sound, which can be explored by gesture.

2.3.1. *Considering time as a spatial coordinate*

We are accustomed to sonagrams (Fig.3) and this way we can consider a 2D image being in fact a 1D vector using time as the second coordinate. This means to derive from a static image an evolving sound.

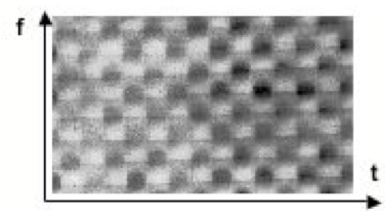

Figure 3: *"Images from a sound" thinking.*

Using images as potential sonagrams is however not straightforward. A sonagram can be considered as the modulus of a sliding short Fourier transform analysis, and the reconstruction can be made according to a phase-vocoder approach [10]. But not every image can be considered as a sonagram. There is a relationship between the points of a sonagram which can be explained this way: a point in a sonagram will give rise to a gaboret (sinusoid enveloped by a window) in the time domain. When taking again a sonagram from this gaboret, one finds the reproducing kernel, which establishes relations between adjacent points. This is not usually fulfilled by arbitrary images. This means that one cannot consider any image as a "valid" sonagram. The other very important point is that not only modulus is necessary, but phases are also important and if not given, they must be estimated. This is not an easy task, and different methods have been suggested [11] [12], which give only an approximate result. One of them consists in going back and forth from an image to a sound, each time reassigning the value of the target modulus to the image, while others assign phases along spectral lines or transients. It is also possible to use the relation between the derivative of phase and modulus according to frequency and modulus. We did a combination of these techniques, as will be demonstrated in section 3.1.

An alternative to the phase-vocoder is the sum of sinusoids approach, where each line of the image corresponds to the envelope of a sinusoid. This forces the frequencies to be on a scale, a strong fact that can be played with, but cannot be called the reconstruction of a sonagram. However, this approach is musically significant and has been widely used by programs such as Metasynth [13].

The gestures that can be associated to this way of doing depend if we work out of real-time or in real-time. A work in studio will depend on gestures linked to computer processing, and only after the sound is rendered. We will show in section 3.1 a description of such a possibility using a database of natural images. When using this correspondence in real-time, we can have two strategies:

- either we consider an image as fixed, so that the gesture will develop sound according to a time unfolding and some frequency and amplitude transformations,
- or we create new images according to action gestures, and unfold these images with modulation gestures which eventually modulates the parameters of the image to sound transcription.

2.3.2. Considering the temporal evolution of an image

This is a classical way to see things, as far as the video capture is now easily made. Here we consider the external aspect of a visual process — for example motion of a dancer in front of a camera — and derive from it features that can be transposed in a sound [14]. If the image is an evolving visual texture and the sound a sonic one, the idea is to find a mapping for the conversion of one to the other. The concept is to have a malleable image which drives a sonic process. The mapping between the two processes can be very loose, because in one way one tries to capture the dynamism of the image as well as its textural content. Such a mapping will be demonstrated in section 4 with the “Filtering string” (1D version of a texture) and with the “Sonic fern” using a 2D image. A subset of this approach relies on the exploration of this image via different pathways to drive the sonic process (Fig.4). The equivalent would be a moving camera over a fixed image, and the focused area gives the present slice of sound. We can say that the initial image is totally innocent, but the way we position successive images gives a sense. An example of this approach will be demonstrated in the “Texture Scratcher” instrument in section 3.2.

2.3.3. Linking both visual and sonic generative processes

In this approach we start from an algorithm of visual texture synthesis, but instead of focusing on the result of this process, one focusses on the process itself, and this way we can link the destiny of a visual object to the destiny of the sonic object (Fig.5). If a texture destiny is written in the unfolding of a program, the data which enables this unfolding will be the constituents of the sound. This way the spirit of an algorithm can be expressed in two media. This way is very powerful, as soon as one can really use the same algorithm for visual and sonic texture making. We are in front of the huge problem of the visual feedback of a sonic process, a field addressed by new research thesis works [15][16].

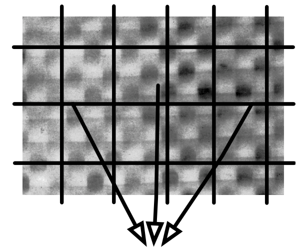

Figure 4: *“Pixel image sonification” technique.*

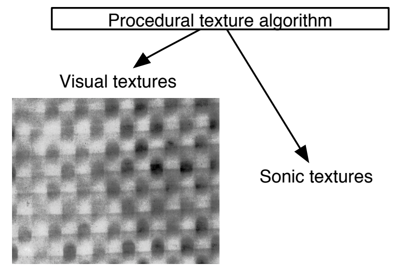

Figure 5: *“Equivalence of the processes” technique.*

3. APPLICATIONS WITH FIXED IMAGES

After these fundamental considerations, we describe in two following sections four realizations done by the authors involving image and sound in the same synthesis process and driven by a gestural interaction: first examples are based on static images (section 3) whereas following ones rely on dynamic images (section 4),

3.1. Visual textures seen as sonagrams

We are tempted to use the wide data banks of visual textures and exploit them as sonagrams. It is not an easy game, because in a way textured images are 2D and not concerned with time, whereas sound is 1D and develops with time. Anyway an interesting point of view is to force a visual texture so that it represents a sonagram. Though a brute force approach of a phase reconstruction from the arbitrary image can be used, the sound results are very metallic. So we decided to use an alternative approach; on a sonagram the horizontal lines can be interpreted as sinusoid partials, whereas the vertical lines can be interpreted as transients. In a sinusoidal +transients +noise approach, it is then convenient to separate an original image into the sum of three images representing these components. This can be done by convoluting the image with three 2D filters, each one selecting a particular aspect of the original image (Fig.6). Starting from a sonagram it is then possible to reconstruct the sine waves, transients and additional noise. We applied this technique with an image taken from the widely used Brodatz database of natural textures (Fig.7).

Each of these three images is then independently treated with programs corresponding to the acoustic facts it belongs to:

- for image 1 (left), vertical lines are assigned phases corresponding to transients. This means phases are zero for maxima and turn at natural rate around these maxima.

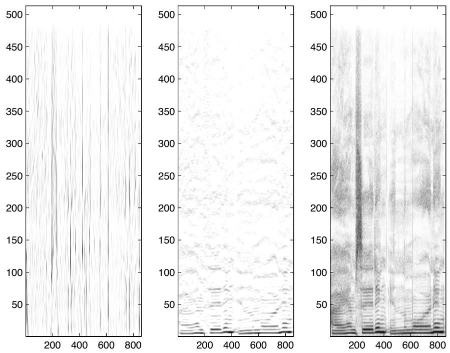

Figure 6: *Separation of a sonagram into three subimages.*

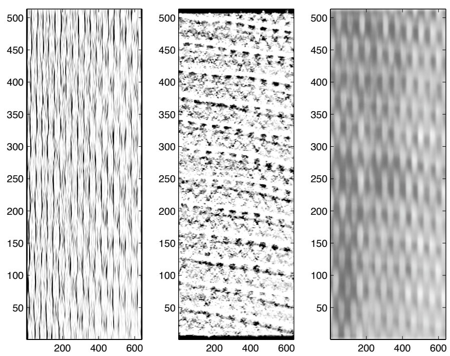

Figure 7: *Separation of a Brodatz texture into subimages.*

- for image 2 (middle), the phases for horizontal lines are developed corresponding to the spectral lines (maximum of amplitude) using techniques similar to SMS [17] (the spectral line is at the top of a parabola for a Gaussian window) or using the fact that the horizontal derivative of the phase is proportional to the vertical derivative of the modulus.
- for image 3 (right), phases are assigned a random value.

These three images are then rendered and the sum of these sonic signals gives a texture which holds the characteristics of the three images (sine plus transients plus noise) and a good intuitive mixing helps find a good balance between them. Presently we have designed only studio techniques corresponding to this processing. This means we use transformations on the images to obtain digital audio effects. But we do plan experiments where images will be created or retrieved according to gesture, and their temporal development altered by modulation gestures.

3.2. The Texture Scratcher

In this instrument, described in detail in [1], the relation between image and sound is quite different than in the previous "sonagram" approach: here sound synthesis does not consist anymore of a basic translation image-to-sound, but is now based on a gesture-controlled exploration of a visual space. This instrument is a real-time adaptation of the Functional Iteration Synthesis (FIS) [18] implemented in Max-MSP-Jitter environment and using a tablet-screen and a joystick as gestural interfaces. FIS is a special application of the well-known wave terrain synthesis, where an orbit is traced on a three-dimensional surface (the wave terrains) to generate a waveform corresponding to the variation of elevation of the trajectory on the terrain.

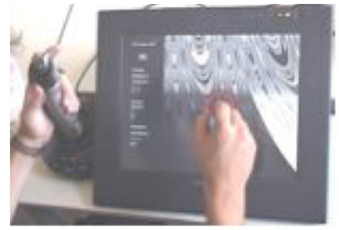

Figure 8: *The Texture Scratcher.*

Specificity of FIS comes from the way the terrains are built: terrains are here computed by iteration of non-linear functions, which make them extremely complex and close to visual textures. Sound is produced by scratching areas of the terrain displayed on the screen with the stylet (Fig.8); terrain scratching is carried out either through a direct way using only the stylet, or through an indirect way by adding a joystick enabling to generate more complex orbits by a parametric control. In this instrument, the resulting sound is not a direct translation of the image. Image is used as a support for the sound synthesis, and though the outline of image has an influence on the sonic result, characteristics of the sound primarily depend on the way to explore the image (orbit shape, velocity, periodicity).

4. APPLICATIONS WITH DYNAMIC IMAGES

4.1. The Filtering String

The "Filtering String" [19] was our first attempt to drive both visual and sound processes by a common gestural control. In this instrument, the shape of a virtual slow-moving string is used to drive a filter bank fed with noise as input. The string shape, based on a spring-mass model, is displayed on the screen and varies the gains of 32 filters. The user acts simultaneously on the image and sound production by interacting with the dynamic system thanks to a graphic tablet and a multi-touch surface: one hand, using the stylet, configures the intrinsic parameters of the string (damping, tension, stiffness) while the other one applies forces on it by exerting pressure on the multitouch surface. This changes equilibrium position of the system and makes the string oscillating, implying dynamic fluctuations in the frequency spectrum of the sound.

4.2. The Sonic Fern

The idea underlying this new instrument is to compute a set of points representing in one hand a graphical object — a fern — and driving in other hand a sonic process. Both processes are be conducted by the same gestural control, which makes the fern move and the corresponding sound evolve in a parallel way. The first step for this is to construct an image shape from a set of input parameters specified by the user. The second step is to connect to the output data an algorithm that creates a sonic texture.

4.2.1. *Construction of the fern by iterated function system*

As described in section 2.2, Iterated Function System (IFS) is a simple way to generate complex image at a low computational cost [9]. For this instrument we use for the control of sound synthesis parameters one of the most famous shape generated by IFS: the Barnsley's fern (Fig.9).

Figure 9: *The classical Barnsley's fern (at left) and deformed ferns (at middle and right).*

The iterative algorithm executed to build this kind of fractal image using IFS is described below. Images constructed by this way are composed of P points, each point being the result of N iterations. The iterative process starts with a couple (x_0,y_0) chosen at the origin, and for each iteration n, a couple of coordinates (x_n, y_n) is computed from previous values (x_{n-1}, y_{n-1}) applying one of the four predefined affine transformations chosen randomly. After N iterations, a point of coordinates (x_N, y_N) is drawn on the image, and the whole process restarts for the next point. The recursive nature of the IFS guarantees that the whole is a larger replica of each frond, that is each leaf of the fern is the same as the full fern.

```
for j = 1 : NumOfPts
% set initial position
x = 0; y = 0;
for i = 1 : iterations
% random draw assigning which transformation
% will be applied for this iteration
p = rand(1);
if(p < p1) k = 1;
else if(p < (p1 + p2) k = 2;
else if(p < (p1 + p2 + p3) k = 3;
else k = 4;
%Compute xn+1 and yn+1
xn+1 = ak * xn + bk * yn + ck;
yn+1 = dk * xn + ek * yn + fk;
xn = xn+1;
yn = yn+1;
end;
pts(j, 1) = xn; pts(j, 2) = yn;
end;
```

The rendering of the fern needs thus four affine transformations T_k, k=1,2,3,4, each of them being characterized by six coefficients [a,b,c,d,e,f] as below:

$$T_k = \begin{cases} x_n = a_k * x_{n-1} + b_k * y_{n-1} + c_k \\ y_n = d_k * x_{n-1} + e_k * y_{n-1} + f_k \end{cases} \tag{1}$$

An image is thus characterized by a matrix M of 24 coefficients $[a_k, b_k, c_k, d_k, e_k, f_k]$ with k=1,2,3,4. By varying these coefficients, one can change the resulting shape of the image. Small changes in one or few coefficients will have as consequence to deform the fern (rescaling, rotation, inclination) whereas varying all the coefficients will render an image very far from the original fern. Furthermore, a continuous variation of one or several of these parameters allows to obtain a user-controlled malleable fern.

4.2.2. *Implementation of IFS object in Max-MSP*

As IFS offers a quite simple method to compute set of points representing moving graphical object, we tried to implement this algorithm in Max-MSP environment, with the objective to link output data provided by IFS to a sonic process. Because available Max-MSP libraries are not suited to the execution of iterative process, we programmed in C our proper Max object, that we called IFS. This object enables to compute and render with Jitter any IFS fractal image in real-time (Fig.10). User is able to specify global parameters (number of points and iterations) and vary independently each value of the coefficient matrix M. By this way, it is a very convenient tool for the drawing of a large range of animated fractal shapes in Max-MSP-Jitter environment. In the experiments described below, the fern was composed of 256 points calculated after 15 iterations.

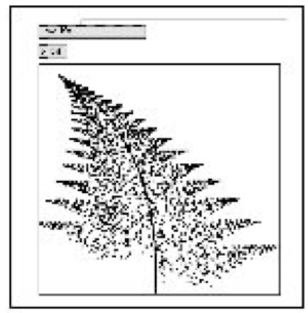

Figure 10: *Barnsley's fern construction algorithm.*

4.2.3. *From image data to sound parameters*

Many strategies may be adopted to map the data of the fern shape to sound synthesis parameters. We developed two of them, very simple, based on granular techniques:

- in the first case, each point constituting the fern triggers a sound grain extracted from a sample, as in classical granulation techniques. Characteristics of the grains vary according to the motion of the fern: horizontal coordinate of the point is used to determine the position of the extracted grain in the original sample whereas the vertical coordinate is assigned the length, amplitude and pitch shifting factor to each grain.
- in the second case, grains are not anymore extracted from a sample but synthesized by noise filtering. Each point of the fern is assigned a noisy grain, that is a narrow noise band which parameters are mapped with image data as follow: the central frequency of the band is given by the horizontal coordinate of the point on the image, its bandwidth by the vertical position and the amplitude of the grain is controlled by the motion speed of the point.

4.2.4. *Gesture-controlled deformation of the fern*

The approach taken here for linking image to sound offers the possibility to use a common gestural control to drive both visual and sonic processes. For the mapping we use the "Max drum", a gestural interface enabling to track the position of two sticks in a 3D space. The objective of the mapping was to give the user the ability to animate the shape of the fern through his gestures. For this we chose to vary simultaneously four coefficients (a_2, b_2, d_2, e_2) of the matrix M. We developed two ways to gesturally control these parameters:

- instantaneous deformation : here values of the four coefficients are directly mapped to X and Y position of each stick. This mapping allows the user to actually sculpt the shape of the fern by means of the two sticks.
- oscillating deformation : in this second way, we consider the fern as a pseudo rigid object, able to oscillate around a equilibrium position when forces are applied to it. User stretches the fern in the same way than previously before to release it by getting up the stick in order to make the fern oscillate. The oscillation is achieved by varying the four coefficients following the motion of a spring-mass model. This is done thanks to the scansynth object originally developed for the Filtering String instrument [19].

Gestures associated to each mapping are of different natures: in the first case, user acts instantaneously on the shape of the image by modulation gestures, whereas in the second, an action gesture the stretch and release of the fern triggers the subsequent evolution of the parameters which will be unfolded after that time.

4.2.5. Future works

Here we tried to link visual and sonic textures in the same synthesis process relying on an IFS fractal image. These experiments provided interesting results, especially on the relation between gestures, image and sound. Indeed a strong correspondence exists in the dynamism of resulting image motion and sound fluctuations. Now we need to pursue in this track by investigating other fractal shapes, more "textured" than the fern, or by trying other mapping strategies to enforce correspondence between image and sound features. We are only at the start of the evaluation of the musical possibilities offered by this instrument, which in fact will belong to each composer or experimenter.

5. CONCLUSION

This work gives a major contribution to the following question: in which way available image processing techniques dedicated to visual textures can be exploited for the sonic textures production? We have experimented different strategies to link image and sound in the same process and results exposed here let us think that the visual textures may be involved at several levels in sonic processes. Among the most promising, the track of mapping a malleable graphic object to a sonic process offers a wide range of attractive possibilities in term of gestural interaction.

6. REFERENCES

[1] J.-J. Filatriau and D. Arfib, "Instrumental gestures and sonic texture," in *Proc. Sound and Music Computing (SMC) Int. Conf.,* Salerno, Italy, 2005, [Online] http://www.tele.ucl.ac.be/~jjfil/smc05_Filatriau_Arfib.pdf.

[2] G. Eckel, D. Rocchesso, and G. Strobl, "Sound texture modeling: A survey," in *Proc. Sound and Music Computing (SMC) Int. Conf.,* Marseille, France, 2006, [Online] http://assoc.orange.fr/gmem/smc06/papers/8-soundtexturemodelingSMC06.pdf.

[3] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman, "Texture mixing and texture movie synthesis using statistical learning," *IEEE Trans. Visualization and Computer Graphics*, vol. 7, no. 2, pp. 120–135, July 2001.

[4] K. Perlin, "An image synthesizer," *Computer Graphics*, vol. 19, no. 5, pp. 287–296, July 1985.

[5] J. S. De Bonet, "Multiresolution sampling procedure for analysis and synthesis of texture images," in *Proc. ACM SIGGRAPH,* New Orleans, USA, 1997, pp. 361–368.

[6] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *Proc. IEEE Int. Conf. on Computer Vision, Kerkyra, Greece*, 1999, pp. 1033–1038.

[7] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, "Texture optimization for example-based synthesis," *ACM Trans. Graphics*, vol. 24, no. 3, pp. 795–802, July 2005.

[8] S. Thon and D. Ghazanfarpour, "Ocean waves synthesis and animation using real world information," *Computer Graphics*, vol. 26, no. 1, pp. 99–108, 2002.

[9] M. Barnsley and H. Rising, *Fractals Everywhere*. Boston Academic Press Professional, 1993.

[10] M. Dolson, "The phase vocoder: A tutorial," *Computer Music J.*, vol. 10, no. 4, pp. 14–27, 1986.

[11] J. Laroche and M. Dolson, "New phase vocoder technique for pitch-shifting, harmonizing and other exotic effects," in *Proc. IEEE Workshop Appl. of Dig. Sig. Proc. to Audio and Acoust.,* New Palz, NY, 1999, pp. 91–94.

[12] D. Griffin and J. Lim, "A new model-based speech analysis/synthesis system," in *Proc. IEEE Int. Conf. Acoust., Speech, and Sig. Proc. (ICASSP'85),* Tampa, USA, 1985, pp. 513–516.

[13] U. Software, "Metasynth," Retrieved June 29th, 2006, [Online] http://www.uisoftware.com/MetaSynth/.

[14] A. Camurri, G. De Poli, A. Friberg, M. Leman, and G. Volpe, "The MEGA project: analysis and synthesis of multisensory expressive gesture in performing art applications," *J. New Music Research*, vol. 34, no. 1, pp. 5–21, 2005.

[15] J.-M. Couturier, "Utilisation avancée d'interfaces graphiques dans le contrôle gestuel de processus sonores," Ph.D. dissertation, University of Aix-Marseille II, 2004.

[16] S. Jordà, "Digital lutherie: Crafting musical computers for new musics' performance and improvisation," Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, 2005.

[17] X. Serra and J. Bonada, "Sound transformations based on the SMS high level attributes," in *Proc. COST-G6 Workshop on Digital Audio Effects (DAFx-98),* Barcelona, Spain, 1998, pp. 138–142.

[18] A. D. Scipio, "Synthesis of environmental sound textures by iterated non linear functions and its ecological relevance to perceptual modeling," *J. New Music Research*, vol. 31, no. 2, pp. 5–21, 2002.

[19] J.-M. Couturier, L. Kessous, and D. Arfib, "Expressiveness and digital musical instrument design," *J. New Music Research, special issue on Expressive Gesture in Performing Arts and New Media*, vol. 34, no. 1, pp. 125–136, 2005.