HOW MUSICAL ARE IMAGES? FROM SOUND REPRESENTATION TO IMAGE SONIFICATION: AN ECO SYSTEMIC APPROACH

Jean-Baptiste Thiebaut
Dept. of Computer Science
Queen Mary, Univ. of London
London, UK
jbt@dcs.qmul.ac.uk

Juan Pablo Bello
Music Technology
New York University
New York, USA
jpbello@nyu.edu

Diemo Schwarz
IRCAM-CNRS STMS
Paris, France
schwarz@ircam.fr

ABSTRACT

Although sound visualization and image sonification have been extensively used for scientific and artistic purposes, their combined effect is rarely considered. In this paper, we propose the use of an iterative visualization/sonification approach as a sound generation mechanism. In particular, we visualize sounds using a textural self-similarity representation, which is then analysed to generate control data for a granular synthesizer. Following an eco-systemic approach, the output of the synthesizer is then fed back into the system, thus generating a loop designed for the creation of novel time-evolving sounds. All the process is real-time, implemented in Max/MSP using the FTM library. A qualitative analysis of the approach is presented and complemented with a discussion about visualization and sonification issues in the context of sound design.

1. INTRODUCTION

There is a long tradition of visualizing musical audio for both scientific and artistic purposes. These visual representations are usually intended to provide an alternative to audition, an inherently sequential and real-time process, by transforming the temporal dimension of audio into one or multiple dimensions in space. As a result, music visualizations allow the non-linear, and almost instantaneous, examination of events separated in time, and of forms that are only evident after an extended period of hearing. In other words, they help to overcome our inability to hear music "at a glance".

The automatic generation of a compact, informative and intuitive visual representation from audio data is an important topic on computer music research, with applications on musicology, music information retrieval and music education, to name a few [4]. In principle, an ideal representation should emphasize one or many of the high-level musical characteristics of the audio signal, e.g. melody, rhythm, harmony, timbre, dynamics, form, etc. However, it is well documented that the algorithmic extraction of this information from audio is a very difficult process, thus making existing visualizations reliant on common low-level time-frequency representations such as the Fourier or Wavelet Transforms, which are closer to the signal than to the characteristics mentioned above.

On the other hand we have western musical notation, the product of a long and complex process of development, as the best known of all high level visual representations of audio. However, it is limited by its imposition of a discrete, note-centred view of music that can be far removed from the sound phenomena that it attempts to represent. In fact, common notation is particularly badly suited to represent timbre or texture beyond common conventions regarding instrument types and families. These issues are critical in computer music composition, where notes are no longer the basic units of musical organization, and where sonic texture and its manipulation become preponderant. In the context of "infallible" sound reproduction by a computer, the composer is in a position to define sounds precisely, something that cannot be achieved through standard notation. This is all the more important since the sonic palette available is no longer limited by the mechanics of sound generation in the physical world. As a result, computer music composers have been actively involved in developing their own semantic to represent music compositions and in finding alternative, more adequate, visual representations for their music [11].

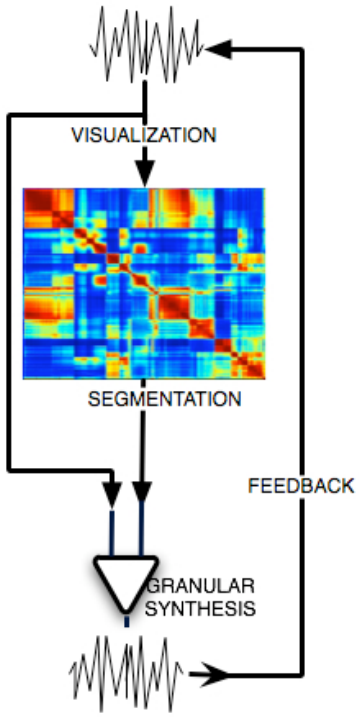

Figure 1. System's overview

Therefore, computer music provides a natural context in which to explore the connection that exists between the complementary processes of visualization and sonification. In this paper, we attempt to do so by implementing a system that recursively converts from audio to image and back in real time, thus continuously changing sounds in ways that are interesting and often unpredictable. The resulting variations are caused by the "information loss" intrinsic to the conversions. However, rather than attempting to minimize this loss, we deliberately increase it for artistic purposes. Figure 1 illustrates the process as implemented by our system.

The remainder of this paper follows the structure in the figure: Section 2 discusses the strategy we follow for the visualization of audio signals; Section 3 describes how images are read as scores and reconverted into sounds; Section 4 discusses the implementation in Max/MSP using FTM and presents some examples illustrative of the performance of the system; Finally, section 5 presents our conclusions.

2. VISUALIZATION

Nearly all musical sounds are highly repetitive, with groups of similar events appearing at all temporal scales, from the very fast cycles that make up pitched sounds to slow-occurring temporal patterns that define the rhythm and/or form of a musical piece. In audio, these repetitions are closely associated with ideas of stability and predictability, as exemplified by references to the highly repetitive "steady-state" of a signal, in opposition to its "transient" and "noisy" parts. These associations are also scalable in time, with repetitions (and lack thereof) characterizing the long-term structure of a sound or piece, in terms of alternations of self-similar (predictable) and dissimilar (surprising) sections. Thus in this paper we choose to visualize audio by computing self-similarity matrices, a well-known strategy for the characterization of repetitions and structure in data (For an example in audio see [3]).

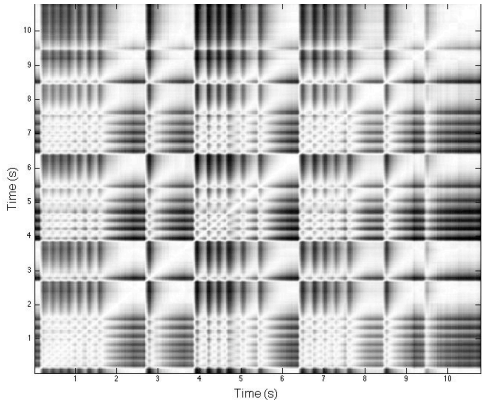

Figure 2. Self-similarity matrix of a dog barking sound.

Let us consider a digital signal that is segmented in short, sequential and partially overlapped blocks of data. We can characterize each data block with an N-dimensional feature vector, i.e. a vector of summarizing quantities that describe specific properties of a given signal segment (e.g. spectral centroid, shape, rolloff, or high-frequency content). This defines a Euclidean space where each vector can be represented by a point with N coordinates. It is possible to calculate the distance between two points in this space such that the smaller the distance is, the more similar the two feature vectors are. If we recursively perform this calculation for all possible pairs of vectors across the length of a sound, we obtain a map of the similarity of the signal with itself, i.e. a self-similarity matrix. Figure 2 shows a self-similarity matrix of an 11-seconds recording of a dog barking. Similarity between features is calculated using a simple Euclidean distance and the first 12 Mel Frequency Cepstral Coefficients (MFCC) as features (see [5] for a recent survey on common features and distances used to characterize sound similarity). These features are adequate for characterizing the textural evolution of sounds by providing a compact representation of its spectral shape (Note that for visualizing changes in, e.g., harmony or rhythm, different features can, and should, be used). In this paper, we favour this texture-based approach to sound visualization.

3. SONIFICATION

Sonification is often understood as a way of presenting abstract data in the auditory domain. There could be many reasons for this, e.g. as an aid for visual impairment or for the creation of immersive sounds environments (c.f. ICAD proceedings). We are particularly interested on the scenario where sonic events are generated for musical purposes.

In recent experiences it was observed that sonification strategies are of great use for the control of sound transformations in both the spectral [8,9] and time domain [10]. In particular, among the different data types that could be used for sound control (e.g. gestural, acoustic, stochastic), visual data was found to be amenable and helpful for the interaction. This is unsurprising if we consider human's intuitive understanding of the information conveyed by visual representations, thus granting users almost instantaneous access to the data's structure and organization.

In a recent work [10], we propose a new strategy for the sonification of visual data. Specifically, this work was concerned with the generation of a music composition from a portrait. The concept developed for this piece was centred around the question of whether the visual structure of the portrait, at both micro and macro levels, could be transformed into the auditory domain. The portrait was analysed from bottom to top in a pixel-by-pixel basis, and segmented in regions of similar luminance (determined by measuring the luminance difference between contiguous pixels). Each region was characterized by its mean luminance and length (the total number of neighbouring pixels that make it). As a whole, more than 4000 regions were found. The length and luminance of each region were mapped, respectively, to the duration and density parameters of a

granular synthesis algorithm. In this project, we used a database of sounds that were selected according to the different macro parts of the image: the blouse, chin, cheeks, lips, eyes and hair. Although the low-level process was identical for every part of the composition we found that using different samples enhanced the contrast between the different parts, while the choice of samples ultimately lets the composer be in control of the aesthetic outlook of the piece. Such a strategy enables infinite textural possibilities while automating the more structural aspects of the music (micro and macro articulations). This strategy was later reused to sonify videos. This time, the quantity of movement was the characteristic used to determine articulations.

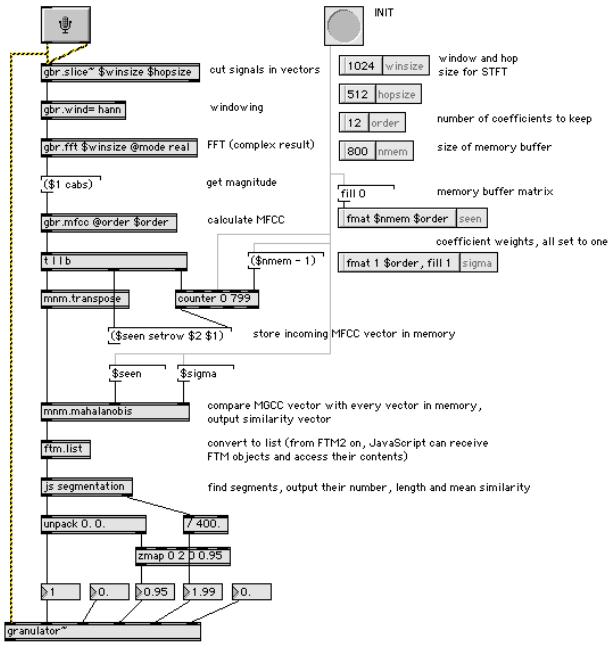

Figure 3. Max/MSP/FTM patch of the self-similarity calculation and distribution of the parameters to a granular synthesis tool.

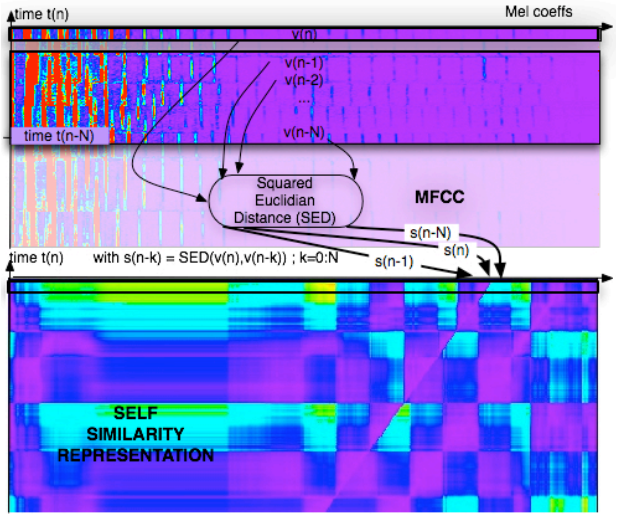

Figure 4. Diagram of the process of self-similarity computation in real-time.

In the current work, we draw on the various sonification experiences we had in the past, while incorporating a few novel, and rather important, modifications. First, we incorporate an interactive aspect, by processing sounds in real time, acquired either from a sound file or a microphone. Secondly, we use a sound-generated image, i.e. a self-similarity matrix, to control transformations on this sound flow. As in our previous work, we use a granular synthesizer with parameters controlled by the analysis of the visual representation. This analysis consists of the segmentation of a self-similarity vector, computed and displayed every 512 samples, into sequences of similar values. For each vector the algorithm calculates the number of sequences, their mean and length.

4. IMPLEMENTATION

A prototype of the proposed real-time process has been implemented as a Max/MSP patch using the FTM library. A snapshot of the patch including blocks for audio analysis, self-similarity calculation, segmentation and synthesis can be seen in Figure 3.

4.1. Real time audio analysis with FTM

As mentioned before, we use Mel-Frequency Cepstral Coefficients (MFCC) as features to represent the information on the audio stream. These features are calculated from the input signal following the standard process (see [5] for an explanation) illustrated in Figure 5: for each block of the signal the discrete Fourier transform is calculated; the logarithm of the magnitude spectrum is taken; a filterbank is used to map frequency (*f*) components to the mel scale, according to the relationship:

$$mel = 1127.01048 \cdot \log\left(1 + \frac{f}{700}\right)$$

Finally, the discrete cosine transform (DCT) of the mel power spectrum is calculated thus generating the coefficients (of which we use the first 12).

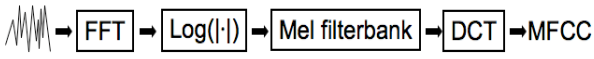

Figure 5. Block diagram of MFCC calculation

As can be seen in Figures 3 and 4, each similarity vector (a row of the self-similarity matrix) is computed as the square Euclidean distance between each incoming vector of the first *order* MFCC values, and the last *nmem* vectors, stored in a circular buffer that corresponds to the "memory" of the algorithm (upper plot of Figure 4). This can be expressed as:

$$s(n,n-k) = \sum_{i=1}^{12} \left(v_i(n) - v_i(n-k)\right)^2, k \in [0,N]$$

where k indexes positions in the buffer, and $i \in [1,12]$ refers to the MFCC coefficient number. The resulting self-similarity vector, at the top of the matrix, is calculated such that the current time position *t(n)*,

indicated by a write pointer, shifts one position to the right of the display with every cycle. This ensures that the displayed matrix corresponds to the standard definition of a self-similarity representation showing a clear diagonal indicating the distance between each vector and itself (from bottom-left to top-right in the plot).

The quick and efficient real-time implementation of MFCCs and the self-similarity matrix is possible through a combination of the FTM and MnM extension frameworks [6, 1] for Max/MSP that add powerful data structures, visualisation and operators, such as a floating point matrix, together with the Gabor arbitary-rate signal processing library [7].

Previous attempts to perform these computations using Jitter were unsuccessful, as this framework cannot handle the computational complexity of matrix operations at audio signal rate.

4.2. Segmentation

The final step of the analysis process consists of the real-time segmentation of each similarity vector. The aim of this calculation is to analyze the image in such a way that the resulting parameterization allows the control of a sound synthesis process (to be discussed in Section 4.4). The segmentation function reads the bin-by-bin values of the self-similarity vector, comparing each value with the running mean within the current segment. If the difference is more than a threshold ∂ (in our case 0.05), then the existing region is set to end at the previous bin, while a new region, starting at the current bin, is defined. The calculation carries on until the end of the vector is reached.
The segmentation algorithms returns: the total number of segments on each vector, and a 2-dimensional array containing the mean and length values of all segments. As will be explained in Section 4.4. the number of segments controls the density of grains of a granular synthesizer such that, the more dissimilar the sound is the less dense the synthesis would be.

4.3. An eco-systemic approach

Eco-systemic approaches are inspired by natural ecosystems, where the cycle of production and destruction is regulated by the system itself. The production of an ecosystem (the output) is part of the system and can be considered, upon creation, as input to the system. DiScipio [2] proposes the use of eco-systemic approaches in systems of sound generation. Such systems, in DiScipio's point of view, are interesting as they interact both with the environment and their own production, with the potential emergence of outstanding music activity or movement.
The sonic system we present here is inspired by this approach. As discussed earlier, we start by analysing and visualizing the incoming sound. Once transformed, the modified sound serves both as output and, through a feedback loop, as part of a mixed input to the system's next iteration. The evolution of the sound differs widely depending on the input, e.g. sound sequences tend to be more stable when calling pre-recorded sound files, and richer and more unstable when using ambient sounds as input (e.g. from a microphone). This evolution depends largely on the used synthesis approach. This is, for this implementation, a granular synthesizer. While we do not claim that this is the best sound engine for this process, we suggest that it provides a useful mechanism for illustrating the ideas put forward by this experimental setup.

4.4. An Implementation Using Granular Synthesis

In this prototype we use Benoit Courribet's *granulator~*. This granular synthesis tool receives 5 parameters in addition to the sound that can be changed in real time. These parameters are: the size of grains, their delays, feedback, density and the shape of the envelope. The number of sequences per vector is mapped to the density value such that the less sequences there are, the highest the density of grains would be. The grain size is controlled by the sequence's length, while the feedback is controlled by its mean value. The other parameters are directly instantiated by the user or assigned from the preset values we have stored in the program.
It is worth noting the high quantity of information generated by the segmentation process: a similarity vector, containing up to 800 segments, is analyzed every 1024 samples at a sampling rate of 44.1 kHz. This represents up to 68900 values per second, more than we can efficiently manage using our current synthesis approach. To partly cope with such high information density, we use successive segmentation values to control two different granulator~ objects, each assigned to a different stereo channel. The appropriate management of such a large amount of information calls for a synthesis methodology that makes optimal use of this data. This is beyond the scope of this paper, but is certainly part of our ongoing efforts at improving the system.

4.5. Discussion

In our experience, the interaction seems to be enhanced by the real-time use of sound, not only as input, but also as a control mechanism. This bypasses the usual dependency on a set of abstract parameters that are external to the sonic process itself. Furthermore, the sound production is supported by a real time visual display that intuitively represents musical content.

Another important aspect is the ability of the system to trigger temporally evolving sounds, making it a useful tool for sound design. The resulting temporal structure is the product of the feedback loop and of the algorithm's re-interpretation of low-level textural changes in the incoming sound into temporal variations of grain density in the output sound. Since this transformation depends entirely on the used timbres, there is no specific sound identity, or limited palette,

that can be attached to this approach (as is the case with common sound synthesis techniques). This is a desirable feature on a sound design application. Figure 6 exemplifies the complexity of similarity-driven time domain transformations. The upper plot of Figure 6 shows the spectrogram of a click sound used as input to the system, while the bottom plot represents the spectrogram of the stereo output. We can observe that, although the frequency content is not strongly modified, the temporal organisation is highly dissimilar but well organised, i.e. not random.

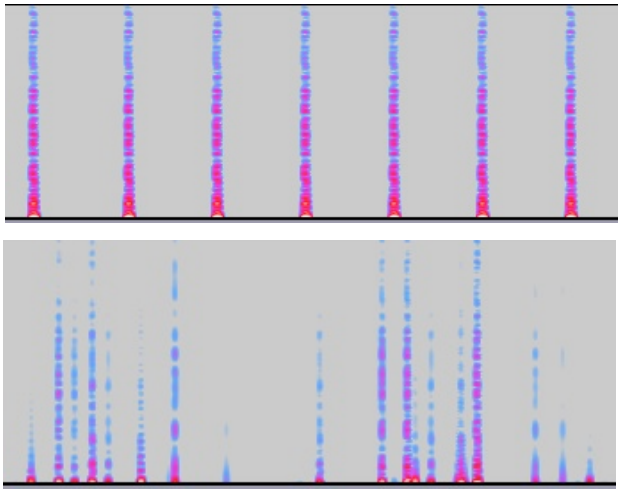

Figure 6: (top) Spectrogram of the original sound and (bottom) spectrogram of the output sound.

5. CONCLUSIONS

There are a number of reasons that motivated us to explore the use of visual representations in real-time sound design processes. First, we wanted to develop a sonification strategy based on a structured image, which is itself a representation of sound. For this we have chosen to visualize sound using self-similarity, a widely used approach in specific areas of computer music research, e.g. in music information retrieval, mainly for music analysis purposes. In turn, this visual representation was segmented into regions and re-interpreted into sound by means of a granular synthesizer that reacts to the length and height of those regions. These ideas draw on previous initiatives, such as Iannis Xenakis' UPIC system and the U&I program MetaSynth, where drawing abilities are required to create sounds.

Second, we wanted to explore whether such representations, linked to a musical application, could be helpful to musicians controlling sound production. Thus we have implemented a real-time, interactive and eco-systemic approach that is able to produce novel sounds on-the-fly that are entirely controlled by the textural characteristics of the sonic input. In this respect, our system is related to the concept of adaptive effects [12], where the characteristics extracted from a control audio signal affect the sound itself. In the future we plan to work closely with composers and sound artists in order to enhance the system's capabilities.

6. REFERENCES

[1] Bevilacqua F., Muller R., Schnell N., MnM: a Max/MSP mapping toolbox. New Interfaces for Musical Expression. Vancouver : May 2005, p. 85-88

[2] Di Scipio, A., 2003, Sound is the interface: From interactive to ecosystemic signal processing (Organised Sound 8/3: 269-277), Cambridge University Press

[3] Foote, J.: Visualizing Music and Audio using Self-Similarity. In Proceedings of ACM Multimedia '99, pp. 77-80, Orlando, Florida, November, 1999.

[4] Isaacson, E.: What You See Is What You Get: on Visualizing Music. Proceedings of the International Conference on Music Information Retrieval, London, UK , September 2005, pp. 389-395.

[5] Pampalk, E.: Computational Models of Music Similarity and their Application in Music Information Retrieval. PhD Thesis, Vienna University of Technology, Vienna, Austria, 2006.

[6] Schnell, N., Schwarz, D. "Gabor, Multi-Representation Real-Time Analysis/Synthesis," in Proceedings of the 8th International Conference on Digital Audio Effects, DAFx'05, Madrid, Spain, 2005.

[7] Schnell, N., Borghesi, R., Schwarz, D., Bevilacqua, F. and Muller, R., FTM -- Complex Data Structures for Max, in Proceedings of the International Computer Music Conference, ICMC, Barcelona, Spain, 2005.

[8] Sedes, A., Courribet B., and Thiebaut J. B. 2004. Visualization of Sound as a control interface. In Proceedings of the Digital Audio Effects Conference, Naples.

[9] Thiebaut, J.-B., Visualisation du son et réversibilité, l'exemple du logiciel Sonos Proceedings of Journées d'Informatique Musicales 2005 (JIM 05), Saint Denis, France

[10] Thiebaut, J.-B., 2006, Portrait, composition for electroacoustic and violin, commission of the National Portrait Gallery

[11] Thiebaut, J.-B., Healey, P., 2006, Sketching: Visual Interfaces to Musical Composition, DMRN workshop, Queen Mary University of London, UK

[12] Verfaille, V. "Effets Audionumériques Adaptatifs: Théorie, Mise en Oeuvre et Applications en Création Musicale Numérique", PhD dissertation, Université Aix-Marseille II, 2003