Chroma Palette: Chromatic Maps of Sound As Granular Synthesis Interface

Justin Donaldson
Indiana University School of Informatics
1900 E. 10th Street, Room 931
Bloomington, IN 47406
jjdonald@indiana.edu

Ian Knopke
Indiana University School of Informatics
1900 E. 10th Street, Room 932
Bloomington, IN 47406
iknopke@indiana.edu

Chris Raphael
Indiana University School of Informatics
1900 E. 10th Street, Room 933
Bloomington, IN 47406
craphael@indiana.edu

ABSTRACT

Chroma based representations of acoustic phenomenon are representations of sound as pitched acoustic energy. A frame-wise chroma distribution over an entire musical piece is a useful and straightforward representation of its musical pitch over time. This paper examines a method of condensing the block-wise chroma information of a musical piece into a two dimensional embedding. Such an embedding is a representation or map of the different pitched energies in a song, and how these energies relate to each other in the context of the song. The paper presents an interactive version of this representation as an exploratory analytical tool or instrument for granular synthesis. Pointing and clicking on the interactive map recreates the acoustical energy present in the chroma blocks at that location, providing an effective way of both exploring the relationships between sounds in the original piece, and recreating a synthesized approximation of these sounds in an instrumental fashion.

Keywords

Chroma, granular synthesis, dimensionality reduction

1. INTRODUCTION

"Granular synthesis" refers to a collection of related techniques for manipulating and reassembling audio using short sonic events, also known as "grains" [1]. Typical grain lengths will be in the range of 5 to 70 ms. (but may even be as long as several seconds). The sound source to be "granulated" may be a previously synthesized sound created for a specific audio result, or may also be based on parts of a real recording, including entire songs. Grains are usually repeated on their own or in alternating groups. At the shorter time lengths, the length of a repeated grain is a determining factor in the perceived pitch of the sound, regardless of the choice of input sound. In this way granular synthesis techniques are unique in that they combine both time and frequency domain information in a compact and highly-malleable representation.

There are many different categories of granular synthesis, such as synchronous, quasi-synchronous, and asynchronous forms, referring to the regularity with which grains are reassembled. Grains are usually windowed, both to aid resynthesis and to avoid audible clicks. However, the choice of window function can also have a pronounced effect on the resulting timbre and is an important component of the synthesis process. Granular synthesis has similarities to other common analysis/resynthesis methodologies such as the short-term Fourier transform and wavelet-based techniques. **Figure 1** shows an example of an envelope windowing and overlap arrangement for four different individual grains of sound (A, B, C, and D) arranged in an arbitrary sequential time.

Because of the large numbers of grains required by the granular synthesis process, controlling the granular procedure can be quite complicated. Hundreds or thousands of decisions must typically be made with a very short time span. Traditionally this has been dealt with by either choosing groups of grains randomly within a particular range, or slowly advancing through a sound's grains in a sequential manner while continuously repeating each successive grain. The difficulty of making so many decisions has tended to exclude more selective methods of choosing grains for a particular situation.

The methodology discussed in this paper presents a different approach to grain selection. Spectral "chroma" characteristics of sound grains are used for correlation. These correlations are then preserved (as best as possible) in a low dimensional "map", providing an easy to use synthesis control system that represents the full range of timbral expression for a given host sound source. The final system is responsive and well-suited to live performance, providing the user with a large number of expressive pitched sound possibilities in a straightforward manner.

2. Previous Work

Granular Synthesis has a long and varied history in both the

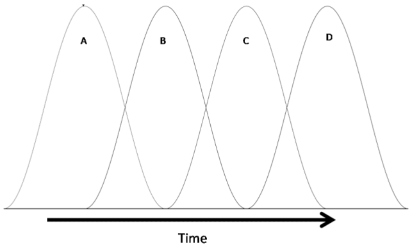

Figure 1. Granular synthesis windowing and overlap

scientific and artistic realms. The concept of viewing sound as a series of small acoustical quanta was first proposed by Denis Gabor as a means of perceptually and mathematically relating the time and frequency domains [2]. A similar approach based on the use of "grains" was later proposed and employed compositionally by Xenakis [3].

One of the first people to actively champion granular synthesis was Curtis Roads, as well as developing one of the first computer-based implementations [4], and a great deal of additional research since that time [5]. Granular synthesis has also been used specifically to synthesize speech, such as in the FOF [6] and VOSIM systems [7]. Many other excellent implementations exist, such as Barry Truax's real-time POD system [8].

Chroma research originates with Roger Shepard's research into representing the perceptual structure of pitch as a two-dimensional helix, instead of the usual one-dimensional representation [9]. This organization of pitch highlights octave relationships, grouping pitches into the familiar "pitch classes" that are part of the basis of western music . Chroma have proven useful in recent years in a number of different music information retrieval contexts such as key-matching [10],[11], and audio thumbnailing [12] because of their ability to represent frequencies in a musically-meaningful way.

Barry Moon proposed a method of indexing grains using amplitude, zero crossings, and some frequency information as part of a live performance interface [13], and this methodology was later extended by Miller Puckette. [14] Other granular synthesis control systems based on cellular automata [15] and finite-state machines [16] have been used. One recent project has even extended the granular idea to include both video and audio, using audio analysis of a human voice for grain selection before resynthesis [17]. While quite different in both conception and aesthetics David Wessel and Ali Momeni also proposed a geometric interface for the control of musical material [18].

3. Calculating Chroma and Height Energy

The "Chroma Palette" focuses on providing a useful method of selecting and utilizing sonic grains as pitched elements according to Roger Shepard's chroma theory. Encoding songs into a series of chroma block profiles can provide a useful representation of the pitched musical events present in the piece, and how these events change over time. Most importantly, it captures the similarity between grains of sound according to how we perceive them, and not according to their raw spectral energy profiles.

Calculating the chromatic and height energy distributions of an FFT distribution is as follows:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad (1)$$

Equation 1 is the classic Discrete (Fast) Fourier Transform that expresses a sequence of N samples as a spectral representation of k "bins".

$$F_k = \frac{SR \times k}{N} \qquad (2)$$

The frequency representation *F(k)* of these bins is expressed in **Equation 2** using the sample rate (SR) of the original signal.

$$M_k = 69 + \lfloor 12 \log_2(F_k/440) \rfloor \qquad (3)$$

Equation 3 translates the frequency bins into a continuous version of the MIDI standard *M(k)*, given a diapason of 440Hz (With the standard reference of A' at 69).

$$C_j = \sum_{k:M_k \bmod 12=j} |X_k| \qquad (4)$$

Equation 4 is the MIDI representation modulus twelve, which gives a chromatic distribution *C(j)* from the key of C to B.

$$H_j = \sum_{k:\lfloor M_k/12 \rfloor = j} |X_k| \qquad (5)$$

Equation 5 is simply the MIDI representation divided by twelve, giving the height distribution *H(j)*.

The Chroma Palette algorithm modifies these calculations slightly. First, the frequency information is normalized and constrained to the standard MIDI range (0 to 127, or 8.17Hz to 12,543.85Hz). Any energy outside this range is removed.

This process gives twelve bins of aggregate energy for a chromatic representation, and eleven for a height representation in the context of the aforementioned MIDI range constraints. However, the height representation in the Chroma Palette uses a twelve bin representation for both distributions, and this does not affect any of the results. Finally, the chromatic and height distributions are concatenated to give the final representation of a given signal segment, called an "augmented chroma distribution". As a final note, the height energy does not correspond directly to the conventional notation for musical octave. However, it does not matter how the height or chroma bins are indexed. All that matters is that the twelve part logarithmic equivalence interval among the frequencies is maintained. The mapping technique will work regardless of the diapason or octave/height values used.

Part *a* of **Figure 2** shows an example spectral representation of a small segment of music taken from a larger piece. This segment represents a single note played on a bass oboe. This segment is a single continuous note over the segment, but it has several spikes present in the Fourier representation (part *a*). Each spike is labeled with its equivalent chroma value and height value. Below part *a* are two figures which show the chroma and height energy (part *b* and *c* respectively) for the above spectral representation. The chromatic and height representations of this segment are much clearer than the comparable FFT information, although they sacrifice some of the details of the FFT. For instance, a chord of A1 and G2 could theoretically equal the chroma and height distributions of a chord of A2 and G1 using the given method. Also, none of the total energy of an individual segment is preserved as a dimension for correlation since the energy for each block is normalized. Furthermore, segments from the embedding that are below a specified volume are removed. The chroma and height distributions are concatenated together to create the final distribution used for analysis. The example distribution in **Figure 2** would produce the final distribution in **Figure 3**. Note how this distribution is essentially just a combination of the distributions from **Figure 2b** and **Figure 2c**. This combination of chroma and height distributions, the *augmented chroma distribution*, is the basis for the dimensional reduction algorithm used to generate the interface map.

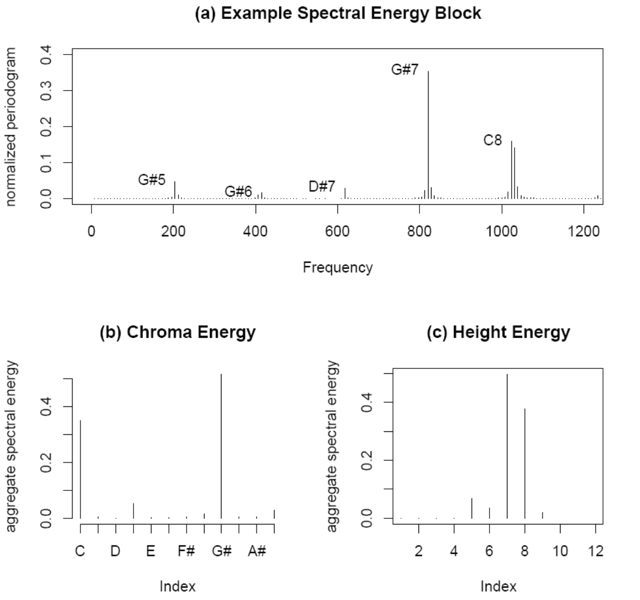

Figure 2. Example spectral, chroma, and height distributions

4. Low-Dimensional Embedding Methodology

Each FFT block is transformed into a twenty-four bin chroma representation. The basic method is as follows:

1. A normalized FFT spectral decomposition is performed on the desired music source at a standard bin size.
2. Each FFT bin for a given block is transformed into an analogous chroma semitone value and a height parameter value, which are then added to the corresponding semitone and height bins for the analogous chroma block in the augmented chroma distribution. This process is repeated for each FFT block.
3. A Euclidean distance matrix is calculated from the resulting matrix of chroma blocks. Each cell in the matrix is the Euclidean distance between the row and column distributions from the chroma block matrix. This establishes how dissimilar a given chroma block is from the other chroma blocks in the array.
4. The distance matrix is then processed with a metric multidimensional scaling method into a two dimensional representation suitable for visualization.

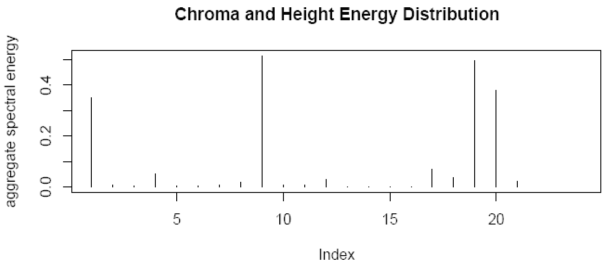

Figure 3. Example augmented chroma distribution

The dimensionality reduction operation applied to the full version of the bass oboe song produced the plot in **Figure 4.**

The Chroma Palette algorithm uses standard methods of calculating distance and metric dimensionality reductions. For more details on these techniques, please check [19] for more information.

5. Embedding Interpretation

Generally, low-dimensional representations of acoustic events created in this fashion behave as follows:

- The segments of sound with pure chromatic and height characteristics will be positioned away from the center of the plot.
- The sounds with more uniform distributions of chromatic and height energy will be located in the center.
- Similar sounding chromatic elements sometimes arrange themselves in long, fairly straight lines (depending on the amount/manner of noise filtering and thresholding used). This behavior can also often be related to vibrato in the performance.

Complex clusters and arrangements of nodes (representations of grain segments on the map) will occur according to the underlying sonic signal characteristics, and are not easily classified or described. In general, while this method compresses a dense amount of information into a two dimensional plane, it can be difficult to predict precisely what a given piece of music will look like, or how a given embedding correlates to the original sound. A method of animating active nodes in the plots in time with the music was developed to better understand the relationship between the musical signal and the resulting two dimensional embedding. An example of this animation is given as a movie clip, available online[1].

6. Stress and Dimensional Simplex

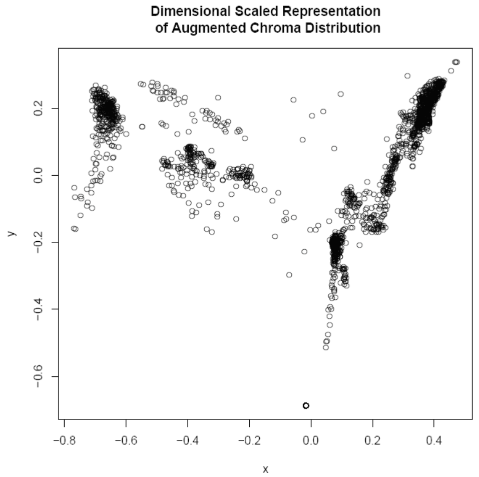

Figure 4. Example dimensional reduction embedding

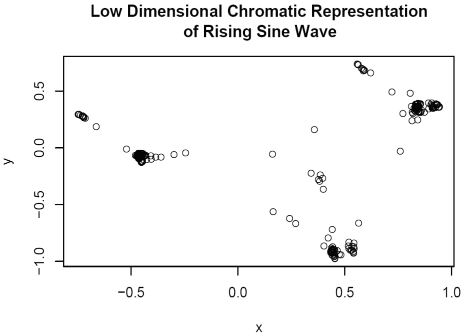

Figure 5. MDS representation of rising sine tone

As the chroma distributions approach a higher level of overall variance, the scaled representation will lose its ability to correctly represent the dissimilarity between blocks as distances in two dimensional space. The error in the dissimilarity represented by distances between chromatic blocks in the two dimensional embedding is known as *stress*, a common result of trying to compress a much higher dimensional distribution into a lower dimension. It is often simply not possible to correctly represent a complex musical piece in a two dimensional space in this fashion, even after normalizing and converting the Fourier distribution into a simpler augmented chromatic distribution. For example, a constant level of dissimilarity (distance) between *N+2* blocks in an *N* dimensional embedding will cause stress. If this level of dissimilarity is the maximum dissimilarity represented in the dimensionality reduction, the embedding will take on a simplex shape [20]. With twenty three potentially independent dimensions, dissimilarity matrices of chroma distributions will commonly have higher levels of stress in lower dimensions.

The Chroma Palette normalizes each block according to its total energy. The normalization simplifies the original distribution, allowing similar sounding notes to correlate closely to each other regardless of their original energy or volume. However, this process constrains the maximal dissimilarity between any two blocks, and increases the likelihood that a non-Euclidean dissimilarity/distance profile exists in the overall distribution, thus increasing the chances of forming a simplex shape. This is the main reason that such a shape is prevalent among the chroma palette maps generated in this fashion.

Even though the embeddings suffer from an expected level of stress and exhibit common simplex shapes in their layout, the representation can still convey meaningful information about the *dominant* chromatic features of the musical piece, and function as useful interaction surfaces for controlling these chromatics features.

7. Other Issues

Chromatic representations of acoustic phenomenon are only meant to capture distinctly pitched sounds. If the signal contains a large amount of noise, resonance, reverberation, or percussion, then the representation will be far less coherent. Low-dimensional representations of music with these issues will not tend to show coherent clustering for any of the pitched elements present, and the individual segments of such a piece will also most likely not suffice as a "palette" for pitched granular synthesis.

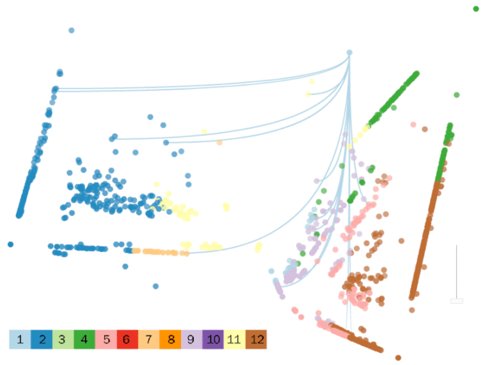

Figure 6. Chroma Palette interface

Figure 5 shows the distinctive triangular distribution of points in the plot of a rising sine wave. This sound highlights the problems of having "too much" disparate frequency information in the grains comprising a signal. The plot shape occurs as a near constant level of dissimilarity occurs between the individual grains under analysis. This shape is also reminiscent of **Figure 4**, which shows that this issue can occur even in monophonic examples with a limited note range.

8. System and Interface Implementation

The granular synthesis interface uses the two dimensional chroma representation as its main display. The maps are calculated from a sound source using the R programming language [21], including the packages for acoustic analysis (tuneR) [22]. The flash applet displays the generated map, and applies a coloration scheme to help distinguish the dominant chroma for each node. Interacting with the applet is done by simply moving the cursor over one or more nodes. When the cursor is close to one or more nodes, the flash applet resolves which nodes are within range, and sends the block index of the given nodes to a granular synthesis engine, which re-synthesizes the original sound present in those blocks. In the current implementation of the Chroma Palette, this message is passed from a Tablet PC over a wireless network connection to a server attached to an amplifier and speakers.

8.1 Interface

The interface, shown in **Figure 6** contains the map of another chroma embedding, a simple legend to indicate how the coloration correlates to the dominant chroma in each segment block, and a small slider (barely visible in the lower right), which allows the selection radius for the cursor to be increased or decreased. When one or more individual blocks are triggered, lines extend from the block to the next block in the original signal sequence. This gives the user an understanding of the sequential relationship of the chromatic elements on the map, and how they in turn relate to the original musical signal.

Performing with the interface is simple. All it requires is a pointing device of some sort, such as a mouse, Wacom tablet or that provided with the Table PC we have used here. As the pointer is moved within the graphical environment, grains are immediately sounded and synthesized. Grain production can be

[1] http://xavier.informatics.indiana.edu/~jjdonald/mdsfft/oboe.mpg

setup to sound only when the pointer is directly over a grain, or placed in a "sample-hold" mode where the last grain is sounded until a new one has been selected. Use of a pointer device with additional degrees of freedom enables additional possibilities, such as controlling the amplitude through device pressure, or selecting a wider "search pattern" for activating grains with a circular range centered around the indicated position.

8.2 Granular Synthesis

This project uses a custom real-time implementation of granular synthesis constructed in the Pure data environment [23]. Up to 32 time-offset grains can be used at a single time, in multiple shapes and sizes, with 16 voices providing the most aesthetically-pleasing results. sound files are stored in PD arrays, which allows for easy indexing. Both time and waveform displays are continuously updated. While primarily used in a synchronous granular synthesis role in this project, asynchronous "cloud" synthesis is also possible, as well as control of other parameters such as amplitude, grain duration, grouping, panning.

The synthesizer can be controlled directly from the PD control window. This sort of slider and number-box control is largely sequential and tends to produce classic granular synthesis sounds, such as the effect of slowly "crawling" through a sound file that has traditionally been used [8]. However, it is also possible to control the parameters of the synthesizer through a network connection.

Figure 7 shows an example of the chroma palette at use with a standard Tablet PC computer. **Figure 8** shows the overview of the system configuration, including the distinction between the interface system and the synthesis system. Communication requests are encoded using a simple TCP/IP protocol, with a bare number indicating a grain sample to use (the most common request), and other requests encoded with simple letter prefixes. While UDP requests could easily be used in place of TCP, the bandwidth has so far not been taxing enough on our local network to warrant this step. In practice, with the speed of today's portable computers, it is perfectly possible to run both the synthesis engine and a simple graphical interface on the same machine. However, it is easy for this system to accommodate complex 3-D graphical systems or small network-enabled wireless PDA devices should the need arise.

9. Advantages

Unlike other sound synthesis methods, one of the interesting aspects of granular synthesis is the use of existing or recorded sound material as the basis for generating new sounds. This can be seen as a form of sampling, but with the intention that the sound product will "re-synthesized" in various ways that will likely be unrecognizable as the original sound and is not restricted to any periodic rhythmic structure.

Figure 7. Example of Chroma Palette Use with Tablet PC

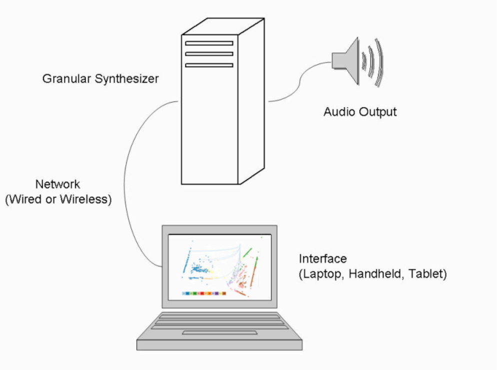

Figure 8. System Overview

The chroma palette techniques described here represent each audio vector as a collection of grains with a unique 2-dimensional distribution. However, groupings of grains with similar timbres and tonalities are also generated and are easily perceived (and further emphasized through the use of coloring). This amounts to a musical interface that will be unique and different for each sound file, and yet still preserves a high-enough degree of visual coherence as to make the usage of the system self-explanatory no matter what performance material is used. Moreover, as the basic performance technique is pointing at different grains (or grain groupings), each unique layout produces a new performance landscape that has it's own possibilities for interaction and sonic exploration.

A second advantage of the system that may not be so obvious is the separation of the interface and the sound production system. One of the difficulties that continuously encumbers live performance with computers or "laptop music" is the necessity of amplifying the sound-production mechanisms. Because the interface is also the same device that produces the sound, and yet has no natural sound of its own, this tends to necessitate having a somewhat bulky, stationary computer on a stage, with audio lines stretching across a hall to a mixing board in the middle of the audience. Performances tend to embrace the character of what could be referred to as a crazed office worker[2]. More portable computers certainly exist with wireless interfaces, but direct transmission of audio requires a lot of bandwidth and may exhibit transmission difficulties that are anathematic to the kind of reliable performance situations that computer musicians strive to achieve.

An obvious solution to this, and the one we have adopted here, is to isolate the relatively lightweight performance interface from the synthesis engine, and to only send control data wirelessly between the two machines. The use of control data instead of audio greatly reduces the bandwidth requirements. The synthesis computer can also now be stationed near to the mixing environment, greatly simplifying the live performance setup and eliminating the need for audio cables cluttering the stage and performance space. The interface is reduced to being simply "what the performer needs" and can be much more portable, as well as enabling a wide range of aesthetic choices beyond the Tablet PC we have used here.

[2] Thanks to David Zicarelli for this analogy.

Also, the separation of the two systems makes it possible to easily attach radically different control interfaces, possibly from completely different computing languages or environments, with the only requirement being the ability to engage in standard network communications. This is especially desirable as the normal graphical environment of PD can be somewhat limiting. At present, graphical interfaces have been implemented in both Flash and Perl/TK. Java, and C++/OpenGl are also being considered. Additional possibilities would be to use an HTML page with an image map (useful for underpowered devices with web browsers such as cell phones or Palm Pilots), or a digital classroom-style electronic "drawing board" that would allow the audience to easily see the graphical interface on the stage.

There is a huge potential here for artists using this device to have highly-personalized, portable interfaces for performance that are both gesturally responsive, repeatable from one instance to another (in other words, not randomly generated) and yet can still be easily understood by anyone, regardless of their level of comfort with technology.

10. Future Work

The Chroma Palette sounds best with pitched elements. Non chromatic elements, such as non-pitched percussion, can be disruptive when they are sorted within a group of pitched elements. One potential modification would be for the chroma layout and analysis algorithm to separate pitched and non-pitched segments, representing the two types of sound differently in the layout, perhaps in separate areas of the interface.

The underlying augmented chroma eigendecomposition process is also a subject for future study. The general characteristics and features of the eigenvalues and eigenvectors of the original signal (as a dissimilarity matrix) need to be investigated and explained further.

The mechanism and manner of interaction with low-dimensional musical embeddings is another opportunity for further work. The authors are interested in moving from the proposed single point of interaction/synthesis, to a multiple point of interaction technique (such as with "multi-touch" interfaces), to a multiple point/multiple person interaction technique (perhaps using a Vicon system to capture dance performance movements as input streams in three dimensional space).

11. References

[1] Roads, C. (2001a). *Microsound*. MIT Press.

[2] Gabor, D. (1947). Acoustical Quanta and the Theory of Hearing. *Nature*, *159*(4044), 591-594.

[3] Xenakis, I. (2001). *Formalized Music: Thought and Mathematics in Composition*. Pendragon Pr.

[4] Roads, C. (1991). Asynchronous granular synthesis. *Representations of musical signals table of contents*, 143-186.

[5] Roads, C. (2001b). Sound composition with pulsars. *Journal of the Audio Engineering Society*, *49*(3), 134-147.

[6] Rodet, X., Potard, Y., & Barriere, J. (1984). CHANT Project: From the synthesis of the singing voice to synthesis in general. *COMP. MUSIC J.*, *8*(3), 15-31.

[7] Kaegi, W., & Tempelaars, S. (1978). Vosim-a new sound synthesis system. *Journal of the Audio Engineering Society*, *26*(6), 418-425.

[8] Truax, B. (1982). Timbral construction in Arras as a stochastic process. *COMP. MUSIC J.*, *6*(3), 72-77.

[9] Shepard, R. (1964). Circularity in judgements of relative pitch, *JASA*, 36, 2346-2353.

[10] Peeters, G. Chroma-based estimation of musical key from audio-signal analysis.

[11] Peeters, G. (2006). Musical key estimation of audio signal based on hidden Markov modeling of chroma vectors. *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06), Montreal, Quebec, Canada*, 127-131.

[12] Bartsch, M., & Wakefield, G. (2005). Audio thumbnailing of popular music using chroma-based representations. *Multimedia, IEEE Transactions on*, *7*(1), 96-104.

[13] Moon, B. (2001). Temporal filtering: framing sonic objects. *Proceedings of the International Computer Music Conference*, 342-345.

[14] Puckette, M. (2004). Low-dimensional parameter mapping using spectral envelopes. *Proceedings, International Computer Music Conference, Miami.*

[15] Miranda, E. (1995). Granular Synthesis of Sounds by Means of a Cellular Automaton. *Leonardo*, *28*(4), 297-300.

[16] Valle, A., & Lombardo, V. A two-level method to control granular synthesis. *XIV CIM - Proceedings of the XIV Colloquium on Musical Informatics 2003*, 136-140.

[17] König, S. sCrAmBlEd?HaCkZ! Retrieved January 31, 2007, from http://www.popmodernism.org/scrambledhackz/.

[18] Wessel, D. and Ali Momeni. (2003). *Characterizing and Controlling Musical Material Intuitively with Geometric Models*. Proceedings of the 2003 conference on New Interfaces for Musical Expression, 54-60.

[19] Kruskal, J., & Wish, M. (1978). *Multidimensional Scaling*. Sage Publications Inc.

[20] Torgerson, W. (1965). Multidimensional scaling of similarity. *Psychometrika*, *30*(4), 379-393.

[21] R Development Core Team. (2006). *R: A Language and Environment for Statistical Computing*. Vienna, Austria. Retrieved from http://www.R-project.org.

[22] Ligges , U., Weihs, C., Preusser, A., & Heymann, M. (2005). *tuneR: Analysis of music.*

[23] Puckette, M. (1996). Pure data: another integrated computer music environment. *Proc. the Second Intercollege Computer Music Concerts, Tachikawa*, 37-41.

[24] De Poli, G., & Piccialli, A. (1991). Pitch-synchronous granular synthesis. *Representations of musical signals*, 187-219.

POSTERS