GRANULAR TOOLS FOR REAL-TIME SOUND PROCESSING, EXAMPLES OF MAPPING WITH VIDEO

Benoît Courribet

Centre de recherche en Informatique et Création Musicale,
Université de Paris VIII
Maison des Sciences de l'Homme Paris Nord, Saint-Denis, France
bcourribet@mshparisnord.org

ABSTRACT

This paper describes software developed by the author on the theme of granular processing and statistical control of parameters. After having exposed the context of this work, both on aesthetical and technical levels, a set of Max/MSP abstractions used for real-time processing is presented. These tools propose spectral and spatial variations to the standard granular techniques. The architecture of these objects, and proposals for control interfaces are presented.
Then, examples of mapping between granular audio processing and video processing are presented.

1. INTRODUCTION

I've always considered granular processing of sounds as a very interesting field to explore both in aesthetic and conceptual ways [1].
Granular techniques make it possible for the composer to access several levels of sound organisation and compose music along multiple time scales.
The set of software tools I'm about to present is based on basic granular processingtechniques. It has been extensively used in my most recent artistic works, including "Isol" for cymbal, live processing and video, the "Solus Ipse" piece used for the multimedia installation "La Terre ne se meut pas", the electroacousic pieces "Formerself" and "time.mass.velocity" and live improvisations with musicians (clarinet, guitar, saxophone, flute…),

2. REAL-TIME GRANULATION TOOLS

The main goal that led me to create these tools was the possibility to granulate incoming sound sources in real time. I had already experienced this by using the VST Plug-in "Shuffling" from the GRM-Tools software bundle [2]. But I wanted to build a generic granulation engine, so I could extend its functionalities and propose variations on the processing. Moreover, I wanted this engine to require very little computational cost.
To describe briefly the granulation of sound, we can say that it consists in extracting short fragments (1 ms – 100 ms) of the incoming sound, windowing these grains and rearranging them in time. For any precision on granular synthesis and processing, I highly encourage the reader to refer to Curtis Roads' "Microsound", which is the definite guide to granular techniques [3].
I will only say that what interested me in granulation of sound was the de-linearization of the time stream, which can range from subtle temporal blurring (and tone-blurring at the same time) to extreme scattering of the sound particles with a wide and continuous palette of sonorities.

We will now describe successively the main granulation object and the subsequent developments and variations.

2.1. The granulator~ object

When I first started to dig into the existing granulation tools for Max/MSP, I realized that most of the time, the granulation techniques were used in order to achieve time-stretch and pitch-shifting processing. This requires the storage of the incoming sound into a buffer (in term of Max/MSP objects). At that time, I was not very familiar with the use of buffers and could not easily figure out how to store continuously incoming sound into buffers avoiding synchronization and looping artifacts. Following a remark I had read in Curtis Roads' "Microsound", I decided to use a delay line.
The granulation engine is controlled by 5 parameters:
Grain size: S (in milliseconds), Maximum delay: D (in milliseconds), Feedback coefficient: F ($0 \leq F \leq 1$), Density: P ($0 \leq P \leq 1$) (actually, P stands for "Probability"), Envelope coefficient: E ($0 \leq E \leq 1$)
The incoming sounds passes through the delay line, the delay time is taken randomly between 0 and D every S milliseconds. The result is that every S milliseconds, a

new fragment of the incoming sound is read. This process is synchronized with an amplitude envelope generator: the delayed signal passes through this generator, which triggers an envelope with a probability of P (with P=0, none of the grains are output, with P=1, they're all output). The envelope coeficient E controls the shape of the envelope: a trapezoïdal window is used, with E=0 it becomes rectangular, with E=1 it becomes triangular. The feedback coefficient controls the amount of signal re-routed in the delay line, it helps with obtaining resonant filter-like sounds.

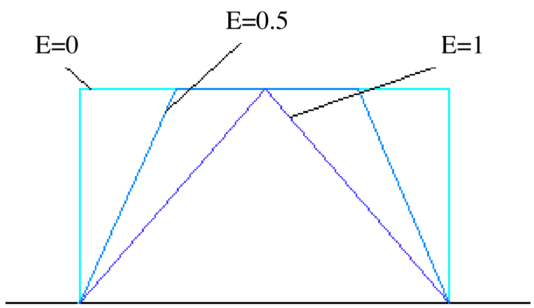

Figure 1. *Influence of the envelope coefficient on the shape of the window*

Of course there are many limitations due to the delay line characteristics. One of these limitations concerns the density of grains; it's not possible to obtain a grain density higher than 1 (i.e. overlapping grains). The feedback parameter can somehow help to obtain perceptively denser streams of grains. Another limitation is due to the real-time aspect of the processing[1]. The read pointer of the delay line cannot go "into the future" contrary to when the sound data is already stored.

Based on this granulation engine, several objects have been developed, proposing spectral and spatial variations to the basic granulation technique.

2.2. Some variations

2.2.1. granulatorq~

Each grain passes through a bandpass filter before output. The cutoff frequency of the filter is taken randomly (on a logarithmic scale) for each grain between a minimum and maximum value set by the user. The quality factor (Q) is also adjustable for the whole stream. With this tool, it is possible to achieve liquid-like tones and all sort of resonant percussive textures.

2.2.2. granulator-freq~

The incoming sound is first subdivided into 5 signals according to 5 contiguous frequency bands (by spectral filtering technique) and then, each one of these signals is routed into an individual granulator~. Set carefully, this tool adds complexity and width compared to the standard granulation.

2.2.3. granulator-space~

The information found in Curtis Roads' Microsound on the theme of Spatialization of sound particles didn't mention real-time spatialization of individual grains into pluriphonic spaces. Each grain is positioned in the pluriphonic space thanks to a spatialization engine. The user defines a circle (by setting its center and radius); the grains will be output within this spatial zone. For now, the spatialization engine is the ambipan~ (ambisonic-B format) external for Max/MSP developed by Rémi Mignot at CICM following an abstraction I had developed earlier. Of course, it is possible to change the spatialization algorithm and thus ensure compatibility for specific systems such as 5.1 for example.

2.3. Control interfaces

In this paragraph, I will present some possibilities on controlling the tools described previously. I have always been interested in continuous variations of sound morphologies and I think that continous variation of statistical parameters is an interesting way to perform high-level sound control. I have developed a generic tool for Max/MSP allowing the user to store sets of parameters into presets and then interpolate between these presets. Combining this tool with the probalistic techniques inherent to granular processing leads to a wide range of dynamic gestures: coalescence, aggregation…
I have also developed some graphic interface MaxMSP patchers in order to control specific objects:

For the granulator-space~ object, a click and drag movement into a graphic window sets the position of the center (click) and length of radius (drag):

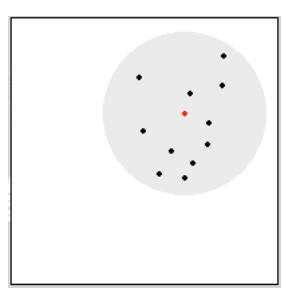

Figure 2. *A graphical interface for the granulator-space~ object. Grains are produced within the zone drwan by the user.*

1 Real-time refers here to the fact that an incoming sound is processed in real time, it doesn't point out real-time control that can also be achieved with stored sound files.

For the granulator-freq~ object, one can graphically set the crossover frequencies of spectral filters along a visualization of the spectrum of the incoming sound:

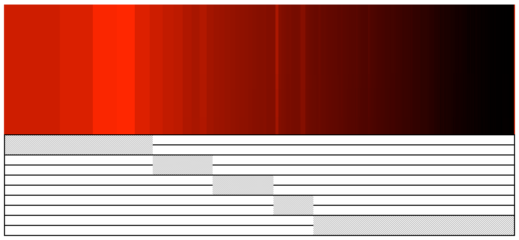

Figure 3. *A graphical interface for the granulator-freq~ object. Cumulative spectrum (top) and frequency bands (bottom).*

3. EXAMPLES OF SOUND/VISUAL MAPPING TECHNIQUES

In the field of sound visualization, many strategies are explored. Contrary to general visualization techniques, I've decided to focus on local strategies and thus explore what I will call a synchronous approach.
The synchronous approach consists in routing a set of parameters (they're called intentional parameters because they pre-exist both the audio and video) simultaneously to video and audio processing modules. The advantage of such an approach is that some data is hard or cannot be extracted from processed audio. For exemple, it's hard to retrieve a delay time from a sound processed by a delay line with feedback. However, before the processing is applied to the audio, it is easy to use a common parameter to process video. While giving the possibility to synchronize audio and video events in an automated way, it leaves room for creativity and autonomy of both media.

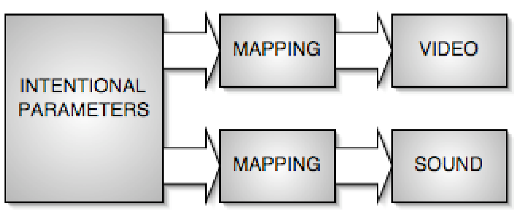

Figure 4. *A representation of the synchronous approach of sound/video relations*

This approach has been used both in real-time and non-real-time contexts. The "Isol" piece contains audio and video parts, linked by the same real-time granulation engine: a percussionist was playing a semi-improvised part on a ride cymbal; a contact microphone was connected to a computer to provide real-time processing of the incoming sound. A random clock was used to generate the grains and at the same time deform an OpenGL 3D structure and change the point of view of a 3D scene. The whole piece consists in continuous changes in sound and visual morphologies, as parameters like audio feedback and visual persistence, size of grains and scale of the structure are linked.

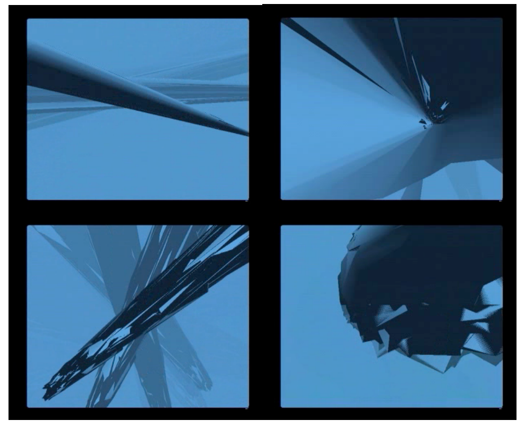

Figure 5. *Stills from the "Isol" piece.*

In a non-real time context, I've used the same clock to drive an audio granulation engine and to provoke a jump in the reading of a movie. Depending on the size and densitiy of grains, many visual effects can be obtained, from subtle dynamic blurring, to extreme randomization. The choice of the video source is extremely important, and slow travellings or zooms are preferred. The "dream" in the beginning of the second part of Andrei Tarkovsky's Stalker is an excellent example.

4. REFERENCES

[1] H. Vaggione, "Son, temps, objet, syntaxe" *in Musique, Rationalité, Langage. Cahiers de Philosophie du langage n° 3,* Editions de l'Harmattan, Paris, 1998, pp. 169 - 202.

[2] E. Favreau, "Les outils de traitement GRM Tools" *JIM98 Proceedings*, CNRS-LMA, Marseille, 1998.

[3] C. Roads, *Microsound*, MIT Press, USA, 2001.