SOM-G, a Language for Granular Synthesis

Paulo Roberto Gonçalves, Aluízio Arcela

Departamento de Ciência da Computação- Universidade de Brasília (UnB)
CEP 70910-900, Brasília - DF - Brazil

roberto@cic.unb.br, arcela@cic.unb.br

Abstract. *The syntax of SOM-G —a sound synthesis language for granular synthesis— is described. Since SOM-G is a descendant from the additive synthesis language SOM-A, it has inherited most of its concise and efficient syntatic structure, being arranged so as for allowing the definition of orchestras of granular synthesis instruments having a high degree of control over granular parameters as well as the interpretation and rendering into digital audio of polyphonic scores using these instruments.*

1. Introduction

Since the postulation of the acoustical quanta and their relation to the theory of hearing [Gabor 1947] and to the theory of information [Moles 1969], along with the first proposal for a composition theory based on sound grains by Iannis Xenakis [Xenakis 1963], granular synthesis opened new horizons in sound synthesis. The early implementations of granular synthesis in computer systems showed its relevance but also has evidentiated some difficulties in controling the sound quality [Roads 1985].

A sound grain is a short piece of sound which is modulated by a time function or envelope as shown in Figures 1 and 2. In granular synthesis, a complex sound is generated by a fast sequence of sound grains, so that a granular synthesis instrument can be thought as a list of hundreds or thousands of sound grains, each of them being defined by a set of parameters. However, in order to granular synthesis becomes a meaningful sound synthesis technique, it is necessary that each generated grain has an individual parametrization.

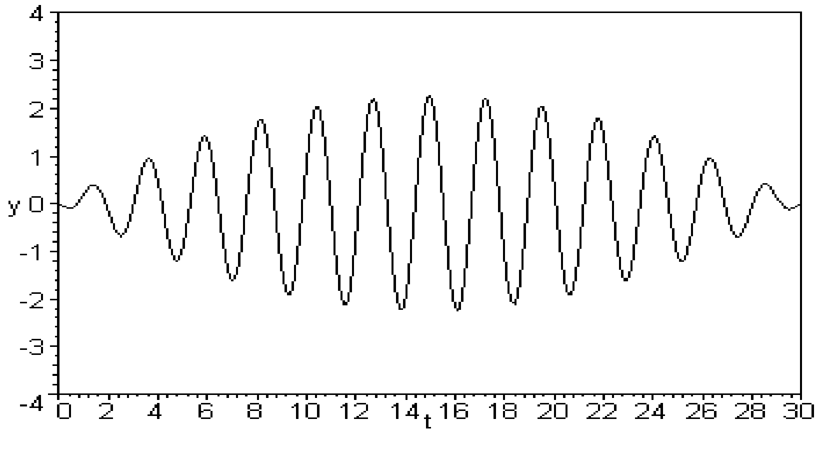

Fig. 1. A sinusoidal sound grain, modulated by a parabolic envelope

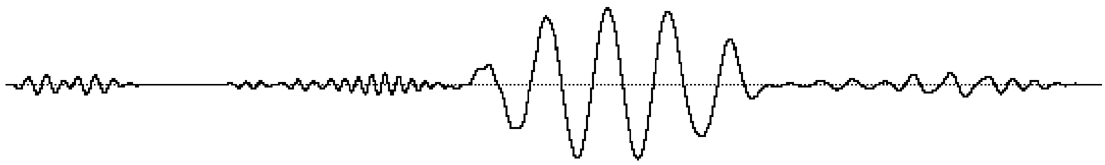

Fig. 2. A sequence of grains with different frequencies, intensities, durations and phase.

A granular sound synthesis language must be flexible and efficient when dealing with the required level of control of the grain generation. The sound synthesis language SOM-G descends from SOM-A, a language for additive synthesis [Arcela 1994] often used for scores computed by the time-trees. SOM-G provides a means for the specification of a set of granular synthesis instruments —called *granular orchestra*— by specifying one by one the grains for each instrument. SOM-G also allows the writing of polyphonic scores and its implementation (in Java plus JavaSound) to be interpreted and rendered into sound. Since the SOM-G's interpreter provides the framework for the synthesis, one could focus his efforts in the design of granular instruments or in applications that generate algorithmically SOM-G granular charts.

2. Granular Parameters

There are two optional parameters that affect all the grains: the envelope and the spectral contents. If the envelope of the grains remains undefined, it is assumed a bell-shaped function, whereas if the spectral content of the grains is ommited, the waveform of the grains will be a sinusoid. There are other granular parameters that affect all the grains. These parameters are referred to as *individual* in this paper. In SOM-G, grain parameters are passed to the synthesis functions through the instrument definition, so that each grain is a part or belongs to an instrument. The instrument must be able to play different notes, so that some parameters must be passed as having a relative value. The individual parameters handled by SOM-G are the following:

- *Relative frequency*: Inside an istrument, the relative frequency for each grain is specified as a real number that when multiplied by the frequency of one note played by the instrument will give the absolute frequency of the grain.
- *Relative intensity*: The intensity of the grain is defined as a real number that after beeing multiplied by the intensity of a note played by the instrument it belongs will produce the final intensity of the grain.
- *Initial phase angle*: The initial phase of the grain.
- *Duration*: The absolute duration of the grain in seconds. Althought frequency and intensity of the grains are defined in relation of the frequency and intensity of the notes to be played, the duration of each grain is defined as an absolute value. When a note is played with a duration that is shorter than the whole duration of the list of grains in an instrument definition, all the grains which start before the duration of the note will not sound.
- *Stereophonic distribution*: A number that specifies the balance of the grain between left and right channels.
- *Time delay* between sucessive grains: The elapsed time from the start of the previous grain in the list to the start of the currently defined grain. As in the case of the grain duration, the delay between grains is absolutely defined and invariant and will remain unchanged even when notes of different durations are played.

The time delay between grains can be set to zero, so that two or more grains with different parameter sets can overlap entirely and compose events having very complex spectral content even if a sinusoid waveform is used for the grains.

3. Syntax and Semantics of SOM-G

3.1 The *VAL* operator

SOM-G programs are called *granular charts.* The *VAL* operator must be the first statement in a granular chart. *VAL* states the time lapse $\mathbf{t_1} - \mathbf{t_2}$ to be interpreted (notes whose starting time lays out of this interval will not be played), the sampling frequency $\mathbf{F_a}$, and three optional modifiers: **T** (a positive number that multiplies all starting times and durations of the notes), **Tr** (a positive number that multiplies all frequencies of the notes) and **N** (a positive number that multiplies all the amplitudes of the notes). It can be applied in two forms:

$$(\mathbf{VAL}\ t_1\ t_2\ F_a) \quad \text{or} \quad (\mathbf{VAL}\ t_1\ t_2\ F_a\ T\ Tr\ N)$$

3.2 The *FRM* operator

This is an optional operator which sets the form of the envelope of the grains. It must preceed all instrument definitions in a granular chart. The FRM operator must be applied in the following way:

$$(\mathbf{FRM}\ y_1\ x_1\ y_2\ x_2\ ...\ y_n\ x_n)$$

The values y_i x_i set the ordinate and the abscissa of a point of the envelope function in a cartesian system of coordinates. The whole envelope function will result from an interpolation of these points. Each x_i value must be in the closed interval [0,720].

3.3 The *ESP* operator

This is also an optional operator. *ESP* sets the spectral content of the grains. It is applied as follows:

$$(\mathbf{ESP}\ f_1\ I_1\ f_2\ I_2\ ...\ f_n\ I_n)$$

The values f_i I_i specifies the frequency order and the relative amplitude of a spectral component of the waveform respectively. The resultant waveform will be normalized, and as said before, the missing of an *ESP* operator in a granular chart means the use of a sinusoidal waveform. An *ESP* operator must preceed all instrument construction in a granular chart.

3.4 The *INS* operator and the construction of granular instruments

The *INS* operator allows the construction of a granular instrument as a list of grains. The form of application of *INS* is:

(INS <name> << list of grains>>)

The string <*name*> sets the instrument name, and each grain in the list has the form $(\mathbf{t_d\ f_b\ I\ d\ f_i\ o})$, where:

- t_d is the time delay in miliseconds between the starting time of the actual grain and the starting time of the next grain denoted.
- f_b is the relative fundamental frequency of the grain. As said before, the absolute frequency of the grain will be determined only when a note is played by the instrument it belongs, as its relative frequency multiplied by the frequency of the note to be played.
- I is the intensity of the grain, also defined as a relative value.
- d is the duration of the grain, expressed in miliseconds.
- f_i is the initial phase of the grain.
- o is the distribution in two stereophonic channels. It can assume values from 0 to 1, being $o = 0$ the situation where the grain will sound entirely in the left channel.

3.5 *EXE* and *STP* operators

The operators *EXE* and *STP* establish blocks of notes to be played by the instruments defined in a granular chart. The blocks must appear after all instrument definitions, and has the following form of aplication:

(EXE t_a t_b)

<<notes>>

(STP)

where t_a and t_b are the limiting times of the part of the block that must be played, expressed in seconds. In other words, (EXE t_a t_b) means "play all notes having starting time between instant t_1 and instant t_2". SOM-G allows the definition of one or more blocks of notes in a granular chart, and each one will be interpreted as an independent polyphonic score. The notes are specified in the following way:

(<instrument> t_i dur f a)

where

- *<instrument>* is the name of the instrument that will play the note
- t_i is the starting time of the note (in seconds)
- *dur* is the duration in seconds of the note
- f is the base frequency in Hertz
- a is the amplitude of the note.

3.6 *FIM* operator

The *FIM* operator indicates the end of a granular chart, and is applied as follows:

(FIM)

4. A grammar for SOM-G

A granular chart has four parts: the first one, referred to as <def_par>, passes parameters for the synthesis. The second one, referred to as <inst_list>, contains the definiton of the instruments. The third one contains the score, referred to as <score>, and the last part is the operator *FIM*, that indicates the end of the granular chart. The whole granular chart has the following structure:

<granular_chart > → <def_par>
<inst_list>
<score>
(FIM)

The parametrization by the *VAL* operator is referred to as *<param>,* the optional parametrization of envelope as <def_env> and the optional parametrization of the spectral content of the grains as <def_spec_content>:

<def_par> → <param>
| <param> <def_env>
| <param> <def_spec_cont>
| <param> <def_env> <def_spec_cont>
| <param> <def_spec_cont> <def_env>

<param> → ***(VAL*** *<flt_lit> <flt_lit> <int_lit> <fl_lit> <fl_lit> <fl_lit>* ***)***
| ***(VAL*** *<flt_lit> <flt_lit> <int_lit>* ***)***

<def_env> → ***(FRM*** *<env>* ***)***

Inside the parametrization of the envelope there is a set of points, referred to as <env>:

<env> → <point>
| <point> <env>

<point> → <int_lit> <int_lit>

As said before, the parametrization of the spectral content for the grains is passed as a list of spectral components referred to as <spec_cont>. Inside this list, each spectral component is referred to as <spec_cmp>.

<def_spec_cont> → ***(ESP*** *<spec_cont>* ***)***

<spec_cont> → <spec_cmp>
| <spec_cmp> <spec_cont>

<spec_cmp> → <int_lit> <flt_lit>

The next part is the list of instruments or orchestra, referred to as <inst_list>. In the construction of each instrument, referred as to <inst_def>, the operator *INS* is followed by a list of grains, referred to as <grains_list>. Each grain in a list is referred to as <grn>.

<inst_list> → *<inst_def>*
| *<inst_def> <inst_list>*

<inst_def> → ***(INS*** *<name> <grains_list>* ***)***

<name> → *<char_lit>*

<grains_list> → *<grn>*
| *<grn> <grains_list>*

<grn> → ***(*** *<flt_lit> <flt_lit> <flt_lit> <flt_lit> <flt_lit> <flt_lit>* ***)***

The score, referred to as <score>, has one or more blocks of notes:

<score> → *<block>*
| *<block> <score>*

<block> → ***(EXE*** *<flt_lit> <flt_lit>****)*** *<note_list>* ***(STP)***

Each block contains a note list:

<note_list> → *<note>*
| *<note> <note_list>*

<note> → ***(*** *<char_lit> <flt_lit> <flt_lit> <flt_lit> <flt_lit>* ***)***

The literals were referred to as:

<char_lit> → A literal of alfanumeric characters
<flt_lit> → A floating point literal
<int_lit> → An integer literal.

5. An example of granular chart

Here is presented a very simple granular chart. In this chart, a single note at the base frequency of 110 Hz is played by an instrument.

```
( VAL 0  1  44100  1  1  0.8 )
( FRM 0 0 50 180 70 360 50 540 0 720 )
( ESP 1 100 2 50 3 20 )

( INS GRAN
(  0 1 20 30 0 0.5  ) ( 30 2 10 30 0 0.5 ) ( 20 3 20 30 0 0.5 ) ( 10 4 10 30 0 0.5 )
( 20 5 25 30 0 0.5 ) ( 10 4 10 30 0 0.5 ) ( 20 2 15 30 0 0.5 ) ( 20 1 10 30 0 0.5 )
( 15 1 20 30 0 0.5 ) ( 20 2 10 30 0 0.5 ) ( 10 3 25 30 0 0.5 ) ( 15 2 10 30 0 0.5 )
( 25 3 10 30 0 0.5 ) ( 10 1 10 30 0 0.5 ) ( 20 3 20 50 0 0.5 ) ( 15 2 10 30 0 0.5 )
( 15 1 10 30 0 0.5 ) ( 20 4 10 30 0 0.5 ) ( 10 3 30 50 0 0.5 ) ( 20 2 10 30 0 0.5 )
)

( EXE t_a t_b )
```

```
(GRAN   0  1   110   100)
( STP )

(FIM)
```

The optional operator FRM was applied to define the envelope for the grains. The form of the envelope is represented bellow.

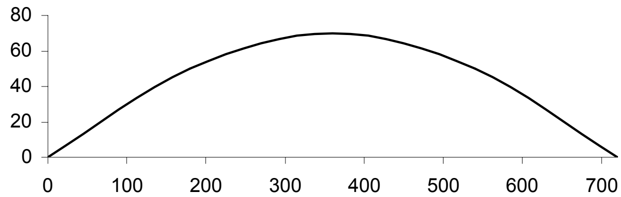

Fig. 3 Shape of the envelope defined in the sample granular chart.

The operator ESP was also applied to define the waveform for the grains.

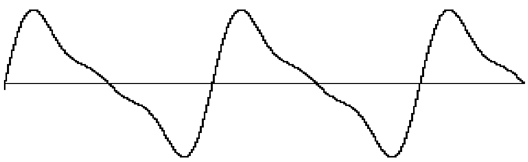

Fig. 4 The waveform defined by ESP operator

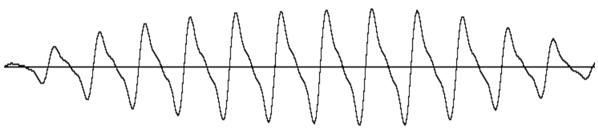

Fig. 5 Shape of the grain

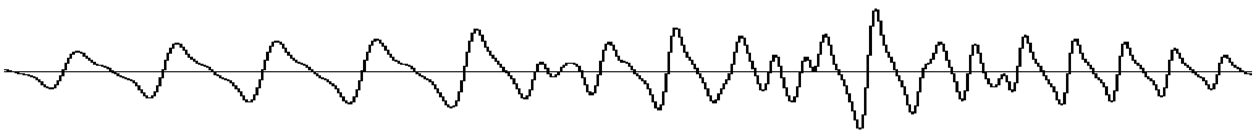

Fig. 6 Part of the signal corresponding to the example granular chart

6. Conclusion

A language for granular synthesis can be a powerful tool for sound synthesis research. There is a number of possible directions for subsequent studies. The algorithmic generation of granular instruments and granular charts should receive special attention, because a large number of grains must be specified. Providing SOM-G with interacting capabilities in a real time context also seems to be very interesting.

References

Arcela, A. “A Linguagem SOM-A para Síntese Aditiva”. *Anais do I simpósio Brasileiro de Computação e Música.* 33–43, SBC, Caxambu, MG, 1994.

Gabor, D. “Acoustical Quanta and the Theory of Hearing”. *Nature 4044*, 591 – 594, 1947.

Moles, A. *Teoria da informação e percepção estética.* Rio de Janeiro, RJ: Tempo Brasileiro, 1969.

Roads, C. “Granular Synthesis of Sound”. in Roads, C., *Foundations of Computer Music*, Cambridge, Massachussets: MIT Press, 1985.

Xenakis, I. “Musiques Formelles”. *La revue musicale, double numéro 253 et 254.* Paris, France: Éditions Richard-Masse, 1963.