Spectral and 3D spatial granular synthesis in Csound

Oscar Pablo Di Liscia

[1] Programa de Investigación Sistemas Temporales y Síntesis Espacial de Sonido, PICT-2015-2604, Escuela Universitaria de Artes, UNQ, Argentina.
odiliscia@unq.edu.ar

Abstract. This work presents ongoing research based on the design of an environment for Spatial Synthesis of Sound using *Csound* through granular synthesis, spectral data based synthesis and 3D spatialisation. Spatial Synthesis of Sound may be conceived as a particular mode of sonic production in which the composer generates the sound together with its spatial features. Though this type of conception has long lived in the minds and work of most composers (specially in electroacoustic music), some strategies applied here were inspired by the work of Gary Kendall [13]. Kendall makes specific mention to both Granular Synthesis and Spectral data based synthesis as examples of resources through which the composer may partition the sonic stream in both the time domain and the frequency domain, respectively. These procedures allow a detailed spatial treatment of each one of the obtained parts of a sound which, in turn, may lead to realistic or unusual spatial images. The aim is not to describe in detail granular synthesis, nor spectral data based synthesis or sound spatialisation techniques, but to describe the particular strategies in the design for the aforementioned purposes.

Keywords: Spectral data based synthesis, granular synthesis, sound spatialisation.

1 Introduction and theoretical background

This work presents an environment for Spatial Sound Synthesis (from here on abbreviated as SSS) using *Csound* through granular synthesis, spectral data based synthesis and 3D spatialisation. This is a part of the author's ongoing research project, in which one of the main objectives is the development of software for high level programming environments (such as *Csound*, *Pure Data* and

Super Collider) for SSS. Another of the software developments being pursued was already presented in Di Liscia [5]. The techniques of Granular Synthesis, Spectral Data Based Synthesis and Spatialisation will not be treated in detail, since those are well known subjects in Computer Music and Digital Signal Processing[1].

SSS[2] may be conceived as a particular mode of sonic production in which the composer generates the sound together with its spatial features. Though this type of conception has long lived in the minds and work of most composers (specially in electroacoustic music), some strategies applied here were inspired by the work of Gary Kendall [13]. Analyzing the function of sound spatialisation in electroacoustic music, Kendall places significant emphasis on the interplay between the perceptual grouping and the spatial features of sound. He makes specific mention of both granular and spectral data based synthesis, as examples of resources through which the composer may partition the sonic stream in both the time domain and the frequency domain, respectively. These procedures allow a detailed spatial treatment of each one of the obtained parts of a sound which, in turn, may lead to realistic or unusual spatial images. Since granular synthesis may modify the "normal" time development of a sound it may be used, along with spatialisation, to generate ambiguities related with the number of virtual sources and their spatial projection. On the other hand, spectral data based synthesis in conjunction with spatial processing makes possible the modification of the spatial projection of the spectral components also leading to the aforementioned ambiguities.

Generally speaking, Granular Synthesis (from here on abbreviated as GS) is a technique based on the juxtaposition of small portions (called grains) of a source audio signal. In a typical GS application, the user is allowed to set (and eventually change over time) several parameters of a grain sequence, the most common of which are: duration, temporal gap, source audio signal, amplitude envelope, peak amplitude, and pitch shifting. The use of GS on electroacoustic and audio visual creation is very well known and widely documented (see, among others, Roads [21], Batty et al [2], and Del Campo [4]), as well as electronic works by many composers. There are many excellent GS opcodes in *Csound* some of the most important being *sndwarp, granule* and *partikkel*, [8]).

Sound synthesis using spectral data may compromise several analysis techniques, but some of the most well-known and used are based on the Fast Fourier Transform (FFT) analysis (See Moore [17], Embree & Kimble [6], Moorer [18],

---

[1] Some appropriate references on each one will be provided in the next section.

[2] To the knowledge of the author, the first authors that coined the name of *Spatial Sound Synthesis* were Bresson and Schumacher [3].

and Wessel & Risset [26]). High resolution analysis combined with a model which attempts to represent the attributes of a sound taking into account its deterministic and stochastic parts is also a well known improvement of FFT based analysis techniques (see [7], [20] and [23]). *Csound* provides several opcodes for FFT-based Analysis-Synthesis, some of the most important being the groups of opcodes for *pvoc, pvsanal* [9] and *ATS* [10] [20].

Sound spatialisation by computer means involves a huge group of techniques and resources. In the opinion of the author, these may be classified in three groups: a) the ones related to the virtual sources location, b) the ones related to the sources directivity, and c) the ones related to rooms/environments. *Csound* offers several excellent opcodes for source location and room/environment treatment. For the former, it is possible to use the *intensity panning* [14], [15], the *VBAP* [21] and the *Ambisonics* [16] techniques. The latter may be achieved via several pre-designed reverberators, using networks of IIR filters, *multitaps* and *delay* units connected in series and parallel, or performing fast convolution [11] with impulse responses of rooms / environments. The *spat3d* opcode [25] provides 3D sound spatialisation of both the direct signal and the early echoes.

2 Per grain 3D spatialisation and spectral GS in Csound

Though *Csound* provides several very powerful GS opcodes, none of them makes feasible individual 3D spatialisation and spectral treatment of overlapping grains in a way that the author found suitable for his purposes. Fortunately, *Csound* provides all the opcodes and resources required to design a GS environment with the capacity of individual processing of each grain generated. The environment that will be presented includes an instrument that implements GS (*the_grainer*) by calling recursively another instrument (*the_grain*) that generates each grain with its own spectral and 3D spatial features, and another instrument (*greverb*) that provides multi-channel reverberation for the complete stream of grains. In what follows, the details of each one of the mentioned instruments, their capacities, and their interaction are discussed.

the_grainer instrument is the part of the GS environment that creates a stream of grains computing all its features and calling with the appropriate parameters another instrument (*the_grain*) that will be described in the next section. The parameters involved are the usual ones in GS synthesis, plus the ones related to 3D spatialisation and spectral features. The user may set the grain's duration, audio source sound file, amplitude envelope function, temporal gap, peak amplitude, audio starting read point on the audio source sound file, pitch transposing, spatial location (azimuth angle, elevation angle and distance) and spectral features (these will be further explained).

For each one of the aforementioned parameters, there is the possibility of setting a base value plus a random deviation value that may change over the grain stream. There are, at present, four ways of setting these two values, which are handled by *ad hoc* macros for the user convenience (details can be found in the source code an its documentation). The audio source for the grains must be, at present, a sound file. The user may, however, have a "pool" of audio source sound files out of which a particular audio source sound file for each one of the grains of a stream may be selected by means of the aforementioned methods.

Since for the spectral processing of the grains, the *pvsanal* / *pvsadsyn* [9] opcodes are presently used, there is the possibility of setting the offset bin value, the bin increment value and the number of bins for the synthesis value which may also be invariant or change over the duration of the stream of grains.

the_grain instrument synthesizes the grains with the parameters computed by its caller instrument (*the_grainer*). The parameters are invariant over the duration of each grain.

As mentioned, the *pvsanal* / *pvsadsyn* opcodes are presently used for the spectral processing of the grains. The time domain signal of the grain generated is sent to *pvsanal* opcode and the frequency domain signal generated by it is sent to *pvsadsyn* opcode to generate the time domain signal of the grain with the spectral modifications requested.

The spatialisation of the grains is achieved mainly using the Ambisonics [16] technique through the *spat3di* [25] and the *bformenc1* [12] opcodes. The *spat3d* opcode provides 3D sound spatialisation computing both the direct signal and the early echoes (using the image method [1]) for a virtual source and listener located into a virtual room whose features must be set by the user in a *Csound* Table. The output of this opcode was set to B-Format First Order Ambisonics (which will produce the four signals usually termed in Ambisonics parlance as W, X, Y and Z) and the echo recursion was set to 2 (the number of early echoes computed will be 24 in this case). After that the first order Ambisonic signals are encoded using *spat3di* and stored in the first four cells of the output array, if requested, the Second or Third Order Ambisonic B-Format of the direct signal only is computed using the *bformenc1* opcode and the remaining signals[3] are added to the corresponding remaining cells of the output array which will -then-contain a *mixed-order* Ambisonic set of signals (MOA). This strategy was al-

---

[3] In the case of Second Order Ambisonic B-Format, these will be the five signals (R, S, T, U and B) and, in the case of Third Order Ambisonic B-Format, seven signals (K, L, M, N, O, P and Q) will be included as well.

ready used by Noisternig et al [19] in order to reduce computational expenses while also minimizing a potential quality loss in the perceptual results. Generally speaking, MOA systems take advantage of the human auditory system's spatial acuity in the horizontal plane, thus using a higher resolution in this plane compared to the resolution used for other directions [24]. Though this is not strictly the case in the present development the aim is similar, since it is commonly assumed that both, the early echoes and the dense reverberation, require less spatial definition than the direct sound.

The output of the instrument is set accordingly to the *nchnls* variable of *Csound*. At present there are four types of possible values: *nchnls*=2 will set stereo (*UHJ* trans-coded) output, whilst *nchnls*=4, *nchnls*=9 or *nchnls*=16, will set respectively First, Second and Third B-Format Order Ambisonics outputs.

The dense reverberation is achieved by a third instrument (*greverb*) by means of fast convolution using the *pconvolve* [11] opcode.

3 Conclusions and future improvements

The environment presented is in continuous assessment and development. However, even in the present state, it proved to be a very powerful, perceptually effective, and versatile tool for SSS applied to electronic composition.[4]

Since there is not enough space for an extensive treatment on the use of the environment for SSS, just two cases -taken from the examples included in the code- will be briefly addressed here. In the first case a broad frequency band sound, synthesized granulating a pitched note of a single sound source, performs a circular movement around the audience. As the movement evolves, the sound "drops" three specific audio bands of its frequency components each remaining steadily in the spatial zones through which their "source" sound has passed. At the same time, the granulation of each one of the partial bands becomes more apparent because the duration of the grains becomes gradually smaller than their gap times. In the second case a sequence of speech sounds is divided into two granulated streams: one containing vowels and the other containing occlusive consonants. This allows a distinctive spatial treatment of each one and, furthermore, the vowel stream is divided into two frequency bands which are segregated in space differently as well.

---

[4] All the referred *Csound* code fully commented (and with several examples that the reader is encouraged to test and analyze) is available at:
https://github.com/odiliscia/the_grainer

Future developments may include, among other, specific Ambisonic decoding stages, the use of higher order Ambisonics orders, the improvements of the distance cues using specially designed filters, the use of other spectral based techniques and the design of a graphic interface to handle the complexity of the environment more comfortably.

4 Acknowledgements

The author thanks Universidad Nacional de Quilmes[5] and FONCyT[6], Argentina, for the support of this research.

5 References

1. Allen, J. and D. Berkley: Image Method for Efficiently Simulating Small Room Acoustics. Journal of the Acoustical Society of America, 912–915 (1979).
2. Batty, J., et al: Audiovisual granular synthesis: micro relationships between sound and image. In: Proceedings of The 9th Australasian Conference on Interactive Entertainment: Matters of Life and Death, pp. 8, Australia (2013).
3. Bresson J., Schumacher, M.: Compositional Control of Periphonic Sound Spatialization. In: Proceedings of 2nd International Symposium on Ambisonics and Spherical Acoustics, Paris, France, (2010).
4. Del Campo, Alberto: Microsound. In: Scott Wilson, David Cottle, and Nick Collins (Eds.) The SuperCollider Book. The MIT Press, London, UK. pp. 463–504 (2010).
5. Di Liscia, Oscar Pablo: Granular synthesis and spatialisation in the Pure Data environment. In: PDCon 2016 Proceedings. Waverly Labs, NYU, New York, USA. pp.25–29. http://www.nyu-waverlylabs.org/pdcon16/proceedings/ (2016).
6. Embree, P. & Kimble, B.: C languaje algorithms for DSP, Prentice Hall, New Jersey, USA (1991).
7. García, G. and Pampin, J.: Data compression of sinusoidal modeling parameters based on psychoacoustic masking. In Proceedings of the International Computer Music Conference, Beijin (1999).
8. Heintz, Joachim et al: *Csound FLOSS Manual.*
   http://write.flossmanuals.net/csound/f-granular-synthesis/ (Last access: 05/2017)
9. Heintz, Joachim et al: *Csound FLOSS Manual.*
   http://write.flossmanuals.net/csound/i-fourier-analysis-spectral-processing/ (Last access: 05/2017)

---

[5] http://unq.edu.ar

[6] http://www.agencia.mincyt.gob.ar/frontend/agencia/fondo/foncyt

10.Heintz, Joachim et al: *Csound FLOSS Manual.*
http://write.flossmanuals.net/csound/k-ats-resynthesis/ (Last access: 05/2017)

11.Heintz, Joachim et al: *Csound FLOSS Manual.*
http://write.flossmanuals.net/csound/h-convolution/ (Last access: 05/2017)

12.Heintz, Joachim et al: *Csound FLOSS Manual.*
http://write.flossmanuals.net/csound/b-panning-and-spatialization/
(Last access: 05/2017)

13.Kendall, Gary: La interpretación de la espacialización electroacústica: atributos espaciales y esquemas auditivos. In: Basso, Di Liscia y Pampin (Eds.): Música y espacio: ciencia, tecnología y estética. Editorial de la Universidad Nacional de Quilmes (2010).

14.Karpen, R. (1998), Locsig Space opcode Documentation. In: *The Csound Manual.*
http://www.csounds.com/manual/html/locsig.html (Last access: 05/2017)

15.Karpen, R. (1998), Space opcode Documentation. In: *The Csound Manual*
http://www.csounds.com/manual/html/space.html (Last access: 05/2017)

16.Malham, Dave: "El espacio acústico tridimensional y su simulación por medio de Ambisonics". In: Basso, Di Liscia and Pampin (Eds.): Música y espacio: ciencia, tecnología y estética, Editorial de la Universidad Nacional de Quilmes, pp. 161–202 (2010).

17.Moore, F. R.: An introduction to the mathematics of DSP, Part II. CMJ 2(2):38–60, MIT Press, USA (1978).

18.Moorer, J. A.: The use of the Phase Vocoder in Computer Music Applications. JAES, 26(1/2): 42-45 (1978).

19.Noisternig, M., Musil, T., Sontacchi, A., Höldrich, R.: A 3d real time rendering engine for binaural sound reproduction. In Proceedings of the 2003 International Conference on Auditory Display, Boston, MA, USA, 6-9 (2003). (Last access: 05/2017)
https://www.researchgate.net/publication/228747180_A_3D_real_time_Rendering_Engine_for_binaural_Sound_Reproduction (Last access: 05/2017)

20.Pampin, J., Di Liscia, P., Moss, P., Norman, A.: ATS user Interfaces. In: Proceedings of the International Computer Music Conference, Miami University, USA (2004).

21.Pulkki, V.: Virtual sound source positioning using vector base amplitude panning. JAES, 45(6) pp. 456–466 (1997).

22.Roads, C.: Microsound. The MIT Press, England (2004).

23.Serra, X. and Smith J. O. III: A Sound Analysis/Synthesis System Based on a Deterministic plus Stochastic Decomposition, Computer Music Journal, 14(4), MIT Press, USA (1990).

24.Trevino, J., Koyama, S., Sakamoto, S., Suzuki, Y.: Mixed-order Ambisonics encoding of cylindrical microphone array signals. In Journal of Acoustic Science and Technology, 35, 3, The Acoustical Society of Japan, (2014).

25.Varga, I.: Spat3d opcode Documentation, In: The Csound Manual.
http://www.csounds.com/manual/html/spat3d.html (Last access: 05/2017)

26.Wessel, D. and Risset, J.: Exploration of Timbre by Analysis and Resynthesis. In: The Psichology of Music, D. Deutsch (Ed.), Academic Press. pp. 26–58 (1985).