Audio Spray Gun 0.8 – the Generation of Large Sound-Groups and Their Use in Three-Dimensional Spatialisation

Richard Garrett
Bangor University, Wales, UK
mupe04@bangor.ac.uk

ABSTRACT

Audio Spray Gun is a computer program that simultaneously generates and spatialises large groups of sound events derived from a single sound sample. This is achieved by constraining random events to a transformable locus within a multi-dimensional parameter space, using a method analogous to the spray gun function found in computer graphics programs. Recent developments in this software include three-dimensional spatialisation and the use of metadata for the storage of sound-groups. This paper describes the program and discusses the implications of these developments for editing and spatial re-orchestration.

1. INTRODUCTION

For some years, the automated generation of numerous sound events has been explored using short samples in both granular synthesis [1] and particle systems for audio [2] but only recently has it become possible, with relatively inexpensive computers, to work with sources of longer duration.

Audio Spray Gun [3] is an experimental tool for fixed-media composition that simultaneously generates and spatialises large groups of such sources, all derived from a single sound sample (typically between 250ms and a few seconds in duration). In this approach to spatial sound synthesis [4], events are created as points constrained by a locus or 'particle zone' [5] within a virtual parameter space. By transforming this locus over time, the program produces a sequence of events, which can then be rendered to multichannel audio.

While earlier versions of the program were designed to render audio direct to eight-channel surround, version 0.8 has been extended to include three-dimensional spatialisation and the option to store sound-groups as metadata for later modification. This paper describes the program and discusses the implications of these developments for editing and spatial re-orchestration [6].

2. AUDIO SPRAY GUN

Audio Spray Gun is so named for its similarity to the spray gun function common to many computer graphics programs in which dots are distributed at random within a moving locus on a canvas. However, because the program operates in a time-based medium, its desired output is a collection of events constrained by the trajectory and transformation of the locus rather than a final static image.

2.1 Parameter Space

Audio Spray Gun 0.8 operates in a six-dimensional space comprised of three spatial dimensions, inter-onset interval (delta time), resampling rate and image spread (for use with Vector Based Amplitude Panning or VBAP [7, 8]). The program generates groups of points or sound-events within this space using simple rules that constrain otherwise random parameter choices to a specific locus which may be transformed over time by means of expansion, contraction and translation.

2.2 Sound-Events

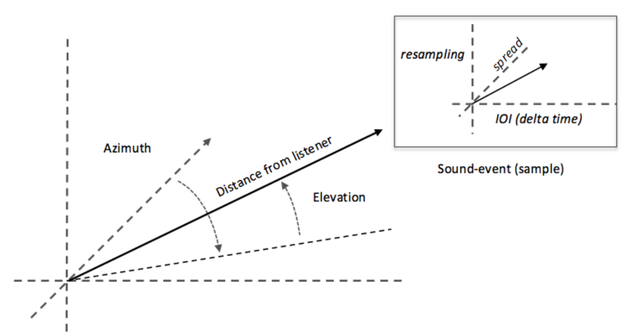

Figure 1: A Sound-Event

Each sound-event generated by the program is described by seven parameters (figure 1): the file name of the sample to be played by the event plus six co-ordinates describing its position in the space. Each event is stationary within the parameter space and has no spatial or spectral trajectory (for example, panning or pitch bend) of its own.

2.3 Loci

Consider a collection of events occurring within a fixed locus. In three dimensions, such a locus might consist of an upright cylinder (figure 2). In this example, events generated by the system occur at random points within the cylinder, giving the impression, when rendered as audio, of a body of sound with specific width, depth and height, centered on a particular location. If the locus is extended into a fourth, spectral, dimension (resampling rate), the events play back at random frequencies in a specific range but still occupy the same spatial locus; if a fifth, temporal, dimension is added, the time between events is constrained and so on.

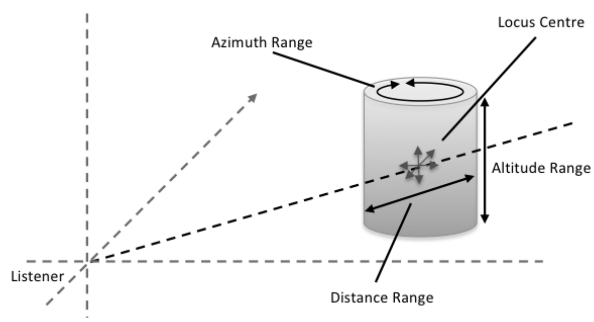

Figure 2: A 3-dimensional locus

2.4 Locus Transformation

Once such a locus has been defined, it may be modified over time by transforming its position and extent within the space. In Audio Spray Gun, this is achieved by altering six pairs of values (one pair for each of the six parameter dimensions). The first of these values locates the focal point of the locus within a given dimension and the second defines its extent about that point. By changing these values over time we can create various trajectories and transformations. These could include: modification of spatial position and extent causing the locus to moves through or expand and contract within the sound space; change of position and extent in the resampling dimension so that the spectral range of possible events moves, expands or contracts; modification of delta time to change event density from discernable individual events to thick textures; alteration of the range of spread values from point-sources to images spread across all speakers; or any combination of the above.

2.5 Sound-groups

Each time the program is run, it creates an array of sound-events chosen at random from within the constraints of a locus undergoing some combination of the transformations described above.

Each time such a sound-group is created, the characteristics of its constituent events may vary significantly but its gross features remain strictly defined by the program. Thus, when the same transformation is executed a number of times, the results obtained exhibit strong similarities at the meso-time[1] scale but differ at the level of individual sound-events.

Sound-groups generated in this fashion can be made up of several hundred events running over periods of a few seconds to a few minutes. While each event is stationary and distinct, the overall effect of so many overlapping sounds can produce dramatic apparent motion around the sound space.

3. IMPLEMENTATION

In Audio Spray Gun, the trajectory and transformation of a sound-group is defined using twelve function generators, one for each of the twelve values described above. Two modes of operation are available so that these functions are dependent either upon time elapsed (as fraction of a predefined total duration) or on the number of events played so far (as a fraction of a predefined total number).

The general format of these functions is

$$value(x) = add + f\,(mult \times env(x)) \qquad (1)$$

where *add* and *mult* are constants and $env(x)$ is the height of a curve or envelope (0 – 1) at point x. This point is defined as either

$$x = (time\ now)/(total\ duration) \qquad (2)$$

or

$$x = (index\ of\ current\ event)/(total\ number\ of\ events) \qquad (3)$$

according to mode.

The function f can be selected from a number of options, for example:

linear	$add + (mult \times env(x))$	(4)
rand	$add + random(mult \times env(x))$	(5)
rand2	$add \pm random(mult \times env(x))$.	(6)

Typically, the trajectory of the locus will be described by linear functions and its extent will be controlled by random functions.

As each event is generated, the program calculates the position and extent of the locus by evaluating these twelve functions. It then selects a random point from within that locus, which it converts to an event location with respect to the listener.

This process repeats in real time until the end-point of the functions is reached. As each event occurs, it is played back

[1] Time periods measured in seconds that govern the local (as opposed to global) structure, see [1] p.14.

as audio and added to a data array so that the sound-group may be replayed later or stored as metadata.

4. INTERFACE

The program interface consists of three windows: the parameter window, the launch window and the display window.

The parameter window contains twelve graphical function generators that define the trajectory and transformation of the sound-group. Each function generator (figure 3) uses a multislider to display a function curve which can edited using the mouse or selected from a menu of pre-defined shapes. The function curve can also be either inverted or reversed using a single mouse click and the 'mode' menu allows the user to choose which type of function (see section 3) will be output. Spatial curves may be defined in spherical, cylindrical or Cartesian coordinates.

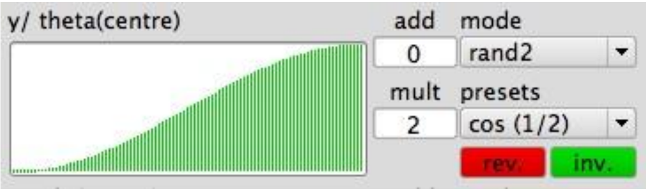

Figure 3: A function generator

The launch window holds controls for program execution and data storage. From this window, the user can trigger and replay sound-groups; select the sound file from which events will be created; load and save parameter sets; select between time and event mode of operation or save sound-groups as metadata. The user can also define the length of the group in either events or seconds; set the maximum gain for individual events; and apply a delay to each event proportional to its distance from the listener. This delay is approximately equivalent to the speed of sound in air (340 ms^{-1}).

The display window shows an animation of the spatial locations of events as they occur.

Audio Spray Gun is written entirely in SuperCollider 3.6.3. [9].

5. METADATA

From version 0.8 onwards, the program can output information in three file formats: as parameter files containing all the variables used to define a particular sound-group; as interleaved audio files for hardware playback or recording to a DAW; or a given instance of a group can be written to file as a SuperCollider archive.

In this third option, one particular instance of the sound-group is stored as a header and an event array. The header contains global elements like the filename of the sample used, master gain, delay status and so on, and the array stores a list of events as points in the parameter space.

This introduces a number of possibilities for post-production.

5.1 Portability between spatialisation systems

Now that sound-groups may be stored as events in an abstract space, it becomes possible to take a stratified approach to spatial composition [10]. Sound-groups may first be rendered to audio in the studio and arranged there and then, at a later date, be re-rendered to the same plan, for concert performance by whatever method (for example VBAP, Ambisonics or Wave Field Synthesis) and for as many speakers are available at the target site.

Thus it will, hopefully, become possible to compose works that can be spatially re-orchestrated for a variety of performance systems with relative ease.

5.2 Surgical editing

In broad-brush techniques such as this, it is often difficult to perform surgical edits on near perfect results. While it may generally be quicker to run the process a few more times until one obtains a better version, it is now possible to edit inconvenient events within the metadata and then re-render the modified material.

5.3 Logical editing

Once a sound-group has been written to metadata, it becomes possible to change it within SuperCollider by searching for events whose parameters meet certain conditions and modifying them either by changing parameter values or by treating the sound file used by those particular events.

For example, because its only spectral dimension is resampling, Audio Spray Gun lends itself to the generation of long sounds whose low frequency components are of extended duration and tend rumble on after higher frequency components have died out. It is possible to use metadata to locate lower frequency events and apply gain envelopes to them, constraining their length so as to compose more abrupt features.

5.4 Substitution of different sound files

A further use of metadata is to substitute different sound samples for the one used when the group was first generated. This can be done globally, for example by rendering a second copy of a sound-group with a different sample and perhaps layering it over the first, or locally, using probability to skew the choice of sound sample over the duration of the group.

No formal tools for manipulating metadata have yet been implemented but a number of experiments in re-rendering and logical editing have been tried with some success.

6. AUDIO SPRAY GUN IN USE

6.1 Composition and Metadata

This author used earlier octophonic versions of Audio Spray Gun in a number of works over the period 2013 -14. The method of operation was to develop sound-groups in Audio Spray Gun and then record a number of variants to eight-channel audio. The process of composition was then to select the best versions and arrange them in Ableton Live, balancing the material largely by adjusting the gain of each group as a whole without recourse to panning and with only minimal use of 'fades'.

The introduction of metadata however, has allowed this practice to change and the author is now developing methods of writing along the lines laid out in section 5.1 with the intention of writing music whose "score" exists as a series of points in a virtual space waiting to be rendered for whatever performance venue is offered.

6.2 Perceived Motion

The generation of multiple static events along a trajectory can give a strong sense of spatial motion and the distribution of sounds around a finite spatial locus can suggest the presence of a single body as opposed to a mass of discrete points. At times, these impressions of motion and physical volume can be sustained even when traversing the acoustically hollow space found at the centre of a multichannel speaker array.

6.3 Spread

Trajectories which feature increased spread as the locus approaches the central listening position or the lowest point in the resampling curve can increase the sense of a solid mass moving through the sound space.

The utility of random spread is still open to question but experiments suggest that if spread values are heavily biased towards point sources and only a few near-monophonic events are introduced, more diffuse textures can be obtained without the total obliteration of other spatial features.

7. CONCLUSIONS

Audio Spray Gun allows the user to construct, spatialise and audition groups of sounds containing hundreds of individual samples in a matter of minutes, producing output it would take days or even weeks to create by manipulating individual samples in a DAW.

With the addition of periphonic audio and metadata storage it is hoped that Audio Spray Gun will become a serious tool for the creation of object-based spatial compositions that are transportable across multiple performance systems.

Acknowledgements

This work was done as part of a PhD studentship funded by the Arts and Humanities Research Council of the United Kingdom.

8. REFERENCES

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

[2] D. Kim-Boyle, "Sound Spatialization with Particle Systems." *Proceedings of the 8th International Conference on Digital Audio Effects (DAFx-05)*, Madrid 2005, http://jim.afim-asso.org/jim2005/download/19.%20Kim-Boyle.pdf

[3] R. Garrett, "In Flight and Audio Spray Gun: Generative composition of large sound-groups." *eContact! 17.3 Toronto International Electroacoustic Symposium 2014: 8th edition of TIES*, 2015, http://econtact.ca/17_3/garrett_audiospraygun.html

[4] N. Fonseca, "3D particle systems for audio applications." *Proceedings of the 16th International Conference on Digital Audio Effects (DAFx-13*), Maynooth, Ireland 2013, http://dafx13.nuim.ie/papers/25.dafx2013_submission_50.pdf

[5] M. Schumacher & J. Bresson, "Spatial Sound Synthesis in Computer-Aided Composition." *Organised Sound*, vol. 15, no. 3, pp 271-289, 2010.

[6] E. Lyon, "The Future of Spatial Computer Music." *Proceedings of the 11th Sound & Music Computing / 40th International Computer Music Conference*, Athens 2014, pp. 850-854.

[7] V. Pulkki, "Virtual sound source positioning using vector based amplitude panning." *Journal of the Audio Engineering Society*, vol. 45 no. 6, pp. 456-466, 1997.

[8] V. Pulkki, "Uniform spreading of amplitude panned virtual sources." *Proceedings of the 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New York 1999, http://lib.tkk.fi/Diss/2001/isbn9512255324/article3.pdf

[9] J. McCartney, "Rethinking the computer music language: SuperCollider", *Computer Music Journal*, vol. 26, no. 4 pp. 61–68, 2002

[10] N. Peters, T. Lossius, J. Schacher, P. Baltazar, C. Bascou & T. Place, "A Stratified approach for sound spatialization." *Proceedings of the 6th Sound and Music Computing Conference*, Porto, 2009.