

Smart Interfaces for Granular Synthesis of Sound by Fractal Organization

Abstract: This article describes software for granular synthesis of sound. The software features a graphical interface that enables easy creation and modification of sound clouds by deterministic fractal organization. Output sound clouds exist in multidimensional parameter–time space, and are constructed as a micropolyphony of statements of a single input melody or group of notes. The approach described here is an effective alternative to statistical methods, creating sounds with vitality and interest over a range of time scales. Standard techniques are used for the creation of individual grains. Innovation is demonstrated in the particular approach to fractal organization of the sound cloud and in the design of a smart interface to effect easy control of cloud morphology. The interface provides for intuitive control and reorganization of large amounts of data.

Mandelbrot's pioneering work in fractal geometry (1977, 1982) inspired composers such as Charles Dodge and Larry Austin to explore musical fractals using the computer (Dodge 1988). Around the same time various composers started to use the logistic map and other nonlinear dynamic systems to control granular synthesis and event generation (Waschka and Kurepa 1989; DiScipio 1990; Truax 1990). These musical explorations were also stimulated by the discovery that the power spectra of pitch and loudness in a wide range of musical styles approximate a $1/f$ spectral density, and much music is therefore statistically self-similar (Voss and Clarke 1975). Melodies generated by Brownian noise with density $1/f^2$ have too little variation and those generated by white noise with constant density ($1/f^0$) vary too widely (Voss and Clarke 1978). More recent work finds a spectral density of $1/f^{1.5}$ as the best generator of natural-sounding melody (Beauvois 2007).

Based on Mandelbrot's work, Barnsley (1988, 2006) developed an elegant description of fractal geometry by iterated function systems (IFS), which has been adopted as the basis for musical explorations by many others (Gogins 1991; Harley 1995; Worrall 1996).

Although powerful, Barnsley's approach privileges two-dimensional plane geometry. By contrast, sonic phenomena are conveniently described by a collection of perceptual parameters unfolding in time: pitch or frequency, amplitude, timbre, spatial location, etc. Though pitch–time space may be used

to describe the evolution of a melody or note complex, it is not a genuine two-dimensional space but rather a linking of distinct one-dimensional spaces. In general, the need to distinguish between the time dimension and each parametric dimension demands an approach towards fractal organization that differs from the standard IFS description of Barnsley. From among the geometric transforms used in fractal construction, the approach described here permits two-way stretches and shears, but avoids rotations.

Deterministic construction is used to create sound clouds as a polyphony of statements of a single input melody or note-group, freely chosen by the user; a carefully designed graphical interface permits intuitive control over morphology of the output cloud and details of the input. Clouds have sufficiently high density that individual note events or melodic lines cannot normally be discerned in the sonic output, and thus the structure of the output is best described as a "micropolyphony." But a proper understanding of this structure demands a preliminary explanation at the note-by-note level, for which we will use traditional score notation. As we shall see later in the article, as the density of the output cloud increases and the sonic output acquires genuine interest, a graphical display becomes clearer than score notation.

Preliminary Concepts

At a preliminary stage, input to the software is a melody in pitch–time space with no rests or overlaps between each of its N note events. A first iteration of the output is obtained by replacing each event i of the input with a miniature version of the input

Figure 1. Input melody (with $r_0 = 1/4$, $r_1 = 1/2$, $r_2 = 1/4$) and output melodies after one and two iterations. Addresses of events in the first iteration are 00, 01, 02, 10, 11, 12, 20, 21, 22; in the second iteration 000, 001, 002, 010, 011, etc.



melody, scaled equally in both dimensions by a ratio r_i so that it has the same duration as event i . A second iteration replaces each event of the input with scaled versions of the first iteration (see Figure 1). A third iteration replaces each event of the input with scaled versions of the second iteration, and so on. As one proceeds, ever-finer detail is added. In the limit, the output comprises N segments, each related to the whole by a similarity transformation, and thus the output melody is described as self-similar.

In practice, construction ceases after a finite number of iterations k , yielding an output of N^{k+1} events that are conveniently counted in base N , starting from zero. Digits of this base N representation yield an address $n_0n_1 \dots n_k$ for each output event, whose pitch and start time are given by the following recurrence relations:

$$p(n_0n_1 \dots n_k) = p_{n_0} + r_{n_0}(p(n_1n_2 \dots n_k) - p_0), \quad (1)$$

$$t(n_0n_1 \dots n_k) = t_{n_0} + r_{n_0}(t(n_1n_2 \dots n_k) - t_0), \quad (2)$$

where

- p_i is the pitch of input event i ,
- t_i is the start time of input event i ,
- $r_i = (t'_i - t_i)/T$ is the ratio of the duration of input event i to the total duration,
- t'_i is the end time of input event i ,
- T is the total duration of the input melody, and
- $i = 0, 1, 2, \dots, (N-1)$.

An output event in iteration k with address $n_0n_1 \dots n_k$ is obtained from the event in iteration $(k-1)$ with address $n_1n_2 \dots n_k$ by a scaling of r_{n_0} relative to origin (p_0, t_0) followed by translation

to the point (p_{n_0}, t_{n_0}) . After several iterations most output melodies are microtonal, moving by intervals smaller than a semitone. But with few iterations and an input with sufficiently wide pitch range, traditional staff notation can be used to illustrate the process of construction (as in Figure 1).

The visual perception of self-similarity in the plane rests on our facility for recognizing similar figures—a facility that is exercised whenever we look at identical objects situated at varying distances in a three-dimensional field. Within carefully prescribed limits, there is no doubt that we perceive similarity in the time dimension: A rhythm played back at different tempi is still understood as the same rhythm. At very high tempi, uneven rhythms are smoothed out in our perception (Abel 1972; Van Noorden 1975), and as we cross the 20-Hz threshold, rhythm starts to be perceived as timbre. At very low tempi, durations exceed the “perceptual present” of around 3 seconds (Pöppel 1997) and rhythm evolves into form (Stockhausen 1962).

The perception of similarity, or scaling, in the pitch dimension is much less certain. For example, most listeners would not understand the arpeggios (C–D♭–E♭), (C–D–F♯), and (C–E♭–A) as fundamentally the same. But if such arpeggios were played with identical rhythm and dynamics, we would understand them as closely related.

Basic Model

The foregoing, preliminary model yields an ornate melodic line rather than the cloud of events required

Figure 2. Input melody,
with $r_0 = r_1 = r_2 = 1/3$.



for granular synthesis. A more advanced model is obtained by the following simple modification of the recurrence relations from Equations 1 and 2:

$$p(n_0 n_1 \dots n_k) = p_{n_0} + r_{n_0}^\alpha (p(n_1 n_2 \dots n_k) - p_0), \quad (3)$$

$$t(n_0 n_1 \dots n_k) = t_{n_0} + r_{n_0}^\beta (t(n_1 n_2 \dots n_k) - t_0). \quad (4)$$

This has far-reaching consequences: it yields sound clouds constituted as a polyphony of statements of the input melody and it mitigates concerns regarding the perception of self-similarity in pitch. At each stage of this new construction a scaling of r_i^α is applied to the pitch dimension and r_i^β to the time dimension. Convergence to a limit requires $\alpha > 0$ and $\beta > 0$, yet this is of little concern because construction always ceases after a finite number of iterations and corrective scaling of the output is usually applied prior to synthesis. When the limit does exist, independent scaling of each dimension ($\alpha \neq \beta$) ensures that it is self-affine rather than self-similar, since the output is now the union of N subsets, each related to the whole by a two-way stretch (an example of an affine transformation). For this reason, α and β are referred to as affine exponents. Equations 3 and 4 lead, in turn, to the closed forms

$$p(n_0 n_1 \dots n_k) = p_{n_0} + \sum_{i=1}^k (p_{n_i} - p_0) \prod_{j=0}^{i-1} r_{n_j}^\alpha, \quad (5)$$

$$t(n_0 n_1 \dots n_k) = t_{n_0} + \sum_{i=1}^k (t_{n_i} - t_0) \prod_{j=0}^{i-1} r_{n_j}^\beta, \quad (6)$$

which also hold for the preliminary model above, with $\alpha = 1$ and $\beta = 1$.

A wide variety of output morphologies result from the interplay of exponents α and β on a given input melody. Some feeling for this behavior can be gained by studying the score examples in Figures 2–4. When α or β takes a noninteger value, however, score notation cannot represent the irrational output

Figure 3. Outputs after one iteration based on the input of Figure 2, with α set to 2, 1, 0, and -1 ($\beta = 1$ throughout).



values generated and a graphical display is required instead (for example, Figures 5 and 6).

The value of α influences the pitch structure of the output (see Figure 3). When $\alpha > 1$, profiles become progressively smoother and output pitches cluster ever more closely around the input. For $\alpha < 1$, pitch profiles become rougher, a trend that continues for $\alpha < 0$ as output pitches are “repelled” from input values and the magnitude of local change exceeds long-term change. The effect of α on output profiles is analogous to the influence of an exponent α on time series generated by noise with a power density spectrum $1/f^\alpha$ (Voss and Clarke 1978; Beauvois 2007).

The value of β influences the time structure of the output (see Figure 4). When $\beta = 1$, the output melody, like the input, has no gaps or overlaps between events. When $\beta > 1$, the output breaks apart into individual statements of the input melody separated by silences. And as β rises further, output events cluster ever more closely around start times of the input. When $\beta < 1$, individual statements of the input melody start to overlap one another to create a polyphony. It is typically under such conditions that sound clouds suitable for granular synthesis are obtained (for example, Figures 5 and 6). For $\beta < 0$, overlaps increase and the durations of output events become large compared to the input.

Figure 4. Outputs after one iteration based on the input of Figure 2, with β set to 2, 1, 0, and -1 ($\alpha = 1$ throughout).



Figure 4

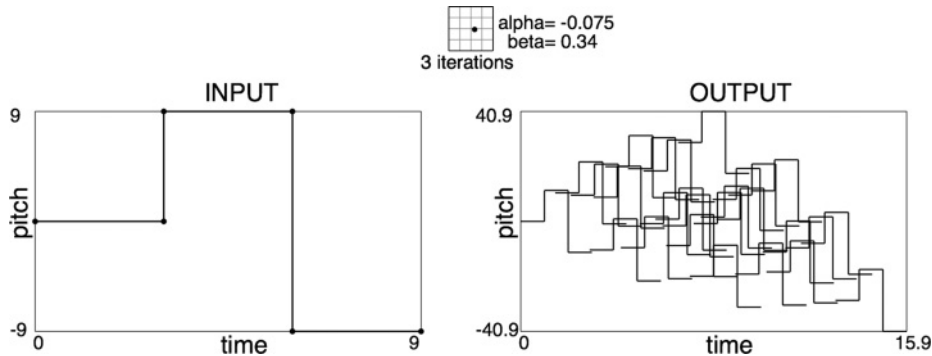


Figure 5

In the special case $\alpha = 0$ there is no scaling of input pitches, and when $\beta = 0$ there is no scaling of input durations. Under such circumstances Equations 5 and 6 specify repeated convolution of the input with itself, or autoconvolution, and if the input sits on a given lattice, so too does the output. Concerns regarding our perception of self-similarity in the pitch domain are thus dispelled when $\alpha = 0$.

Extending the Model

The preceding model may be extended and generalized. A first generalization allocates multiple

Figure 5. Graphical interface, showing the input melody of Figure 2, and an output cloud after three iterations, $\alpha = -0.075$, $\beta = 0.34$. Vertical lines clarify the polyphonic structure of the output.

parameters to input events and a second generalization allows linear change of such parameters over time. Finally, gaps and overlaps between input events are permitted.

First Generalization: Multiple Parameters

A first generalization of the model allocates multiple parameters to each input event: pitch, amplitude, timbre, spatial location, etc. For each parameter, the adoption of a scale that is perceived as linear when associated with a corresponding synthesis definition ensures that the graphic display bears a close

Figure 6. The same output as in Figure 5, after six iterations. Vertical lines are omitted from the display.

Figure 7. The same cloud as Figure 6, showing amplitude versus time for $\alpha = 0.813$, $\beta = 0.34$.

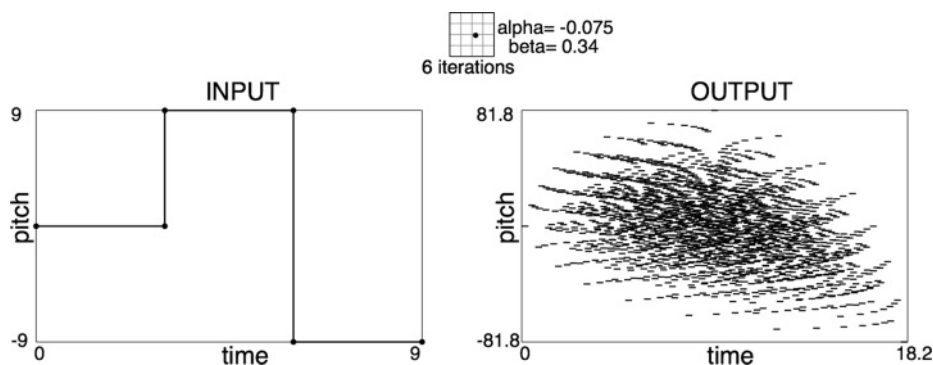


Figure 6

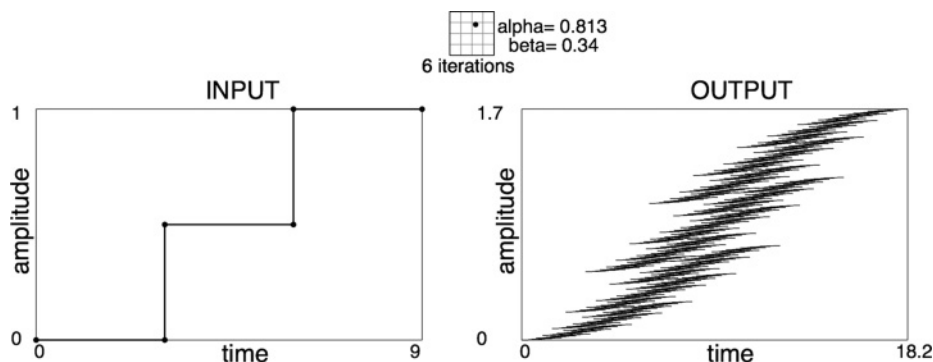


Figure 7

relationship to the sonic result, thereby facilitating close control over design of the output sound. It is for this reason that pitch–time space rather than frequency–time space has been adopted. But for many parameters, linearization is fraught with difficulty. Despite the pioneering work of Stevens (1957, 1972) and Fletcher and Munson (1933), this is certainly the case for perceived loudness. For many other parameters, especially those governing timbre, linearization may be unattainable and in such cases aesthetic criteria will guide the choice of a metric. Currently, the software permits up to ten different parameters to characterize events, but this can be extended. To create the output, each parameter is associated with its own value of an affine exponent α and its own number of iterations—but a single value of β still governs organization of time (see Figures 6 and 7).

Second Generalization: Linear Change of Parameters

In a second generalization, individual events of the input melody are permitted linear change of each parameter over time. Given an input event i with parametric gradient m_i , the transformation associated with this event entails scaling by r_i^α and r_i^β , followed by a shear parallel to the parameter axis that transforms an event of zero parametric gradient into one with gradient m_i . The new recurrence relations are given by the following equations:

$$p(n_0 n_1 \dots n_k) = p_{n_0} + r_{n_0}^\alpha (p(n_1 n_2 \dots n_k) - p_0) + r_{n_0}^\beta m_{n_0} (t(n_1 n_2 \dots n_k) - t_0), \quad (7)$$

$$t(n_0 n_1 \dots n_k) = t_{n_0} + r_{n_0}^\beta (t(n_1 n_2 \dots n_k) - t_0), \quad (8)$$

Figure 8. Input melody comprising four events, and output after two iterations, $\alpha = 0.55$, $\beta = 0.45$.

Figure 9. Output based on the same input, after seven iterations, $\alpha = 0.55$, $\beta = 0.45$.

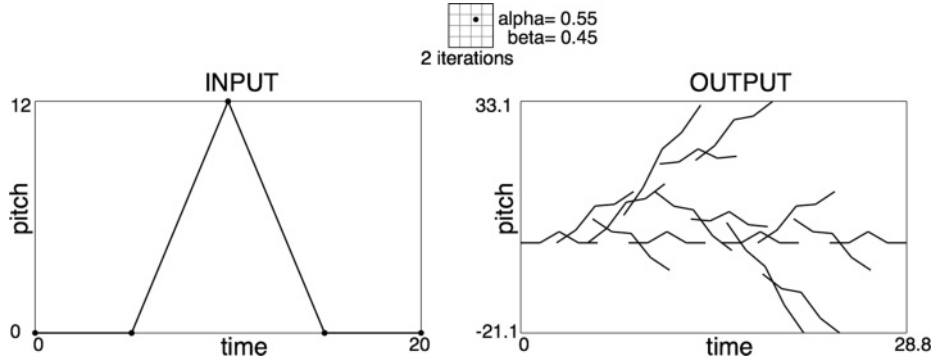


Figure 8

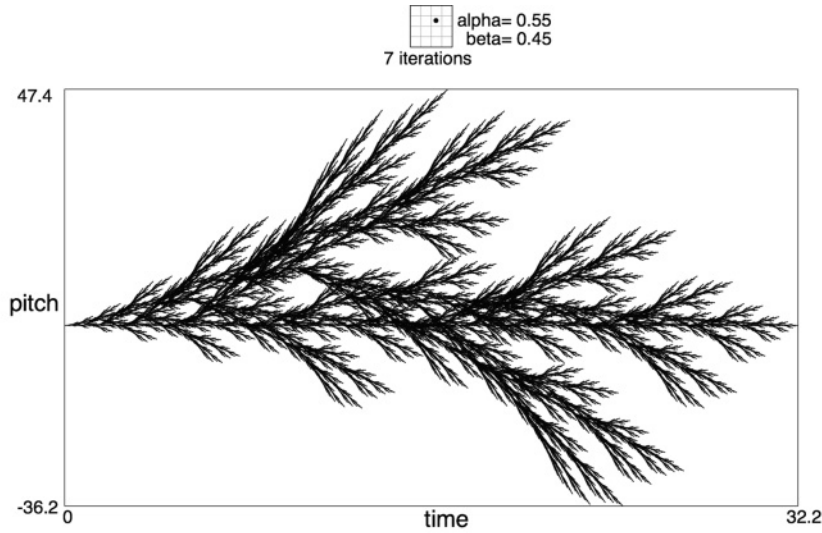


Figure 9

where

$m = (p'_i - p_i)/(t'_i - t_i)$ is the parametric gradient of each input event, p'_i is the end parameter value of input event i , and t'_i is the end time of input event i .

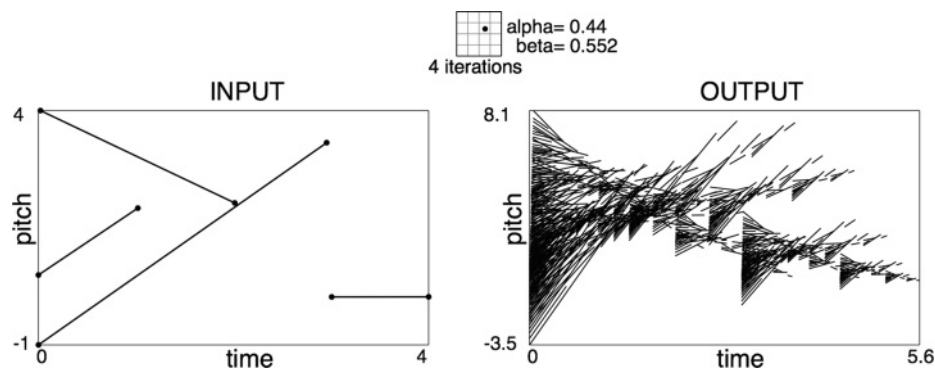
The final term of Equation 7 is responsible for the shear. The closed form for $t(n_0 \ n_1 \ \dots \ n_k)$ remains as in Equation 6, but the closed form for $p(n_0 \ n_1 \ \dots \ n_k)$ is cumbersome and is not shown here. Still, it is worth noting that the parametric gradient of each

output event is given by:

$$m(n_0 n_1 \ \dots \ n_k) = m_{n_0} + \sum_{i=1}^k m_{n_i} \prod_{j=0}^{i-1} r_{n_j}^{\alpha-\beta}. \quad (9)$$

With the introduction of nonzero gradients, new output morphologies result: dendritic forms, arborescences, and so on (see Figures 8 and 9; further examples follow later in the article). If $\beta = 1$ and the input melody is continuous, the output is a fractal interpolation function (Barnsley 1986; Monroe

Figure 10. Input comprising four overlapping events, and output cloud after four iterations, $\alpha = 0.44$, $\beta = 0.552$.



1995) whose roughness is controlled by the value of α .

Third Generalization: Gaps and Overlaps between Input Events

A final generalization permits both gaps (silences) and overlaps between input events. The input is no longer restricted to being a melody, but may be any arbitrary arrangement of note events (see Figure 10). When overlapping events predominate, the user may choose to define the scale values r_i (Equations 1 and 2) as a proportion of the sum of durations of all events, rather than a proportion of the bounding duration. But apart from this, no new protocols are required.

The Interface

The interface is shown in Figure 11 and its use demonstrated in a video at http://www.mitpressjournals.org/doi/suppl/10.1162/COMJ_a_00374. Both the interface and the sound synthesis routines have been coded in James McCartney's SuperCollider environment. A wide variety of output morphologies result from the interplay of exponents α and β upon the input, and they are easily explored via the interface. A mouse drag in the output window is tracked and vertical motion is mapped to α , horizontal motion is mapped to β . By doing so the user is afforded the impression that mouse

movement is controlling the morphology of the cloud: stretching, skewing, or folding it.

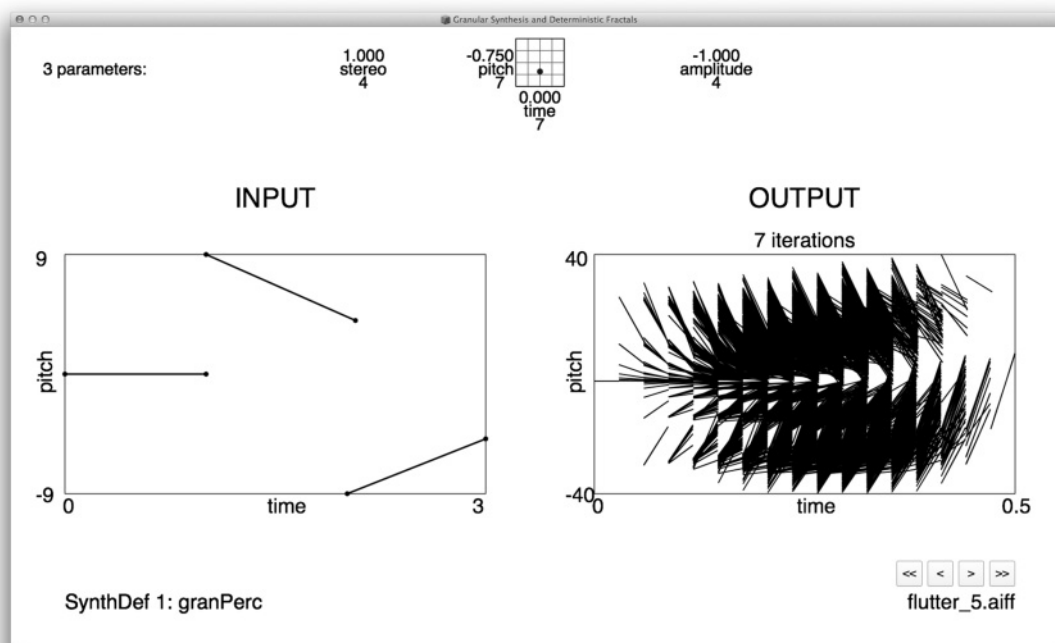
Names of all the parameters used to define the input and output are displayed across the top of the screen. The value of α is displayed immediately above each parameter name, and the number of iterations immediately below. The value of β is displayed top center of the screen, labeled as the exponent associated with time, and the number of iterations for time is shown immediately below (see Figure 11). The user can view the input and output in each parametric dimension versus time, using a key press to move between parameters. Parameter names wrap around so that the current parameter is always located in the middle of the display, where a small grid shows a point plotted in α - β space.

Optional color display of the output cloud shows each grain colored according to the value of a second parameter chosen by the user: blue represents lowest values, red highest, and intermediate values are mapped across the spectrum. Shift-drag in the output window changes the value of α for this second parameter, resulting in striking color changes of the cloud.

Using the mouse, input events can be moved, added, or deleted while display of the output is instantly updated. If required, the input can also be edited in a text file containing all the input data. In addition to the standard operations of inversion, retrograde, and retrograde inversion, the software permits shear of the input in two orthogonal directions. Shear parallel to the time axis leaves parameter values unaffected but skews

Figure 11. Screenshot of the interface. In this example input events start and finish at 0, 1, 2, and 3 seconds. The output has $\alpha = 1.0$ for stereo position,

-0.75 for pitch, -1.0 for amplitude, and $\beta = 0$. Seven iterations are used for pitch and time, and four for stereo position and amplitude.



events in time, so it transforms a chord comprising simultaneous events of different pitch into a rising or falling arpeggio. Shear parallel to the parameter axis leaves time values unaffected but skews the parametric gradient of every event, so transforming events of fixed pitch into glissandi with identical gradients. Operations within the interface are carried out by some 80 distinct key commands that are described in a Help file.

Linear scaling of the output cloud, both in time and in each of its parametric dimensions, is usually applied prior to synthesis via a secondary interface. For example, the high density of events at the start of Figure 10 may demand an overall reduction in amplitude to avoid distortion, or the cloud in Figure 9 may require a wider overall pitch range in order for us to hear its detailed internal pitch structure. When synthesis is complete, a vertical cursor scrolls through the display of the cloud during playback of the resulting sound.

Using buttons at the bottom right of the screen, the user can navigate through a succession of output

sound files created during the current session; in Figure 11 these files are named flutter.0 through flutter.5. The name of the SuperCollider synthesis definition used to create grains is displayed at bottom left of the screen.

Discussion of Results

A wide range of sound clouds have been synthesized by the author in two-, four-, and eight-channel sound using a range of grain types, including short sampled sounds and synthetic types. A selection of stereo outputs is presented in the video at http://www.mitpressjournals.org/doi/suppl/10.1162/COMJ_a.00374. Cloud morphologies inherit their characteristics from the input and from the choice of affine exponents. The sonic results have greater interest across a range of time scales than similar sounds created by stochastic granular synthesis. There is a stronger sense of purpose in their unfolding through time and more interest in their detail.

Successful results are achieved by paying attention to the pairing of grain type with cloud morphology, design of the input profiles, allocation of affine exponents to each parameter, and spatialization of grains.

If input events are situated on a time lattice, the output cloud manifests significant periodicity when β takes integer values, and this periodicity is particularly pronounced if β is 1, 0, or -1 (Figure 11). If such a periodic output cloud is synthesized using grains with a percussive amplitude envelope and the overall time scale adjusted so that the output period is less than 1/20 sec, one will hear a single pitched sound with a fluctuating timbre that reflects internal characteristics of the grains.

By contrast, when β takes noninteger values the output cloud does not, in general, demonstrate periodicity, even if the input is situated on a lattice (Figures 5–10). Such aperiodic clouds are the more common class of output and prove to be the most useful musically. When synthesized, the internal characteristics of grains (e.g., their pitch and timbre) typically remain perceptible as part of an overall sound mass in much the same way as clouds whose time structure is stochastic.

The ability to specify a different number of iterations for each parameter has rich compositional potential. For the cloud shown in Figures 6 and 7, one might in addition define an input profile for the stereo position of grains and specify just one iteration for this parameter. The output will then comprise $3^2 = 9$ overlapping sub-clouds, each characterized by its own stereo position and each containing $3^5 = 243$ grains. Color display of such a cloud will reveal its structure.

Curtis Roads has drawn attention to the importance of parameter linkage in granular synthesis: the dependency, whether direct or inverse, of one grain parameter to another parameter (Roads 2004, pp. 125–129). Such linkages have been explored by (1) creating input profiles for different parameters that are closely related, or that are inversions of one another, and (2) direct coding within SuperCollider synthesis definitions. Weaker, behavioral linkages can be established by allocating the same affine exponent to two or more parameters.

Granular synthesis has traditionally been used to create gradually evolving sound textures, but by choosing an input with many overlapping events, as in Figure 10, this software can produce an output cloud with a marked concentration of amplitude at its start, yielding a powerful, percussive sound. A short sound object created with this software may also be used as a grain for another, higher-level cloud with quite a different structure, thereby creating complex sound objects that are best described as “multifractals” (Harte 2001). Such a multifractal approach is one way of managing the 20-Hz perceptual discontinuity, since two independent time structures can be designed on either side of the divide.

A much earlier version of this software was published and distributed by the Composers’ Desktop Project. It was used by the author to create the acousmatic work *Ebb and Flow*, and to organize rhythmic density in his String Quartet No. 1 and *Chicago Fall* for acoustic and electronic instruments (Rhys 1996).

Future Developments

Michael Barnsley (2006) introduced the concept of a “V-variable fractal” as a hybrid between two or more iterated function systems and a “superfractal” as the collection of all such possible hybrids. These hybrid forms possess a greater degree of irregularity than those generated by a single IFS (see, for example, Figure 5.27 in Barnsley 2006) and are proposed as a better fit to many of the geometrical forms encountered in nature. The software and interface described in this article can readily be developed to permit generation of such hybrid sound clouds, and thereby create hybrids between two or more sound objects.

Conclusion

Among the variety of approaches used for the organization of grains in the granular synthesis of sound, reviewed by Roads (2004), the technique described here has rich compositional potential and warrants further investigation. The sonic results

have greater interest across a range of time scales than do similar sounds created by stochastic granular synthesis. They demonstrate greater purpose as they unfold in time and more interest in their detail. Despite the mathematical description presented in this article, the graphical interface allows the software to be used in an intuitive fashion and constitutes a powerful tool for organization of multidimensional grain data. The technique will be refined and developed further in preparation for more widespread use by the musical community.

Acknowledgments

Thanks are due to Rajmil Fischmann (University of Keele, UK), who guided the earliest phase of this work, Tom Hall (Anglia Ruskin University, UK), who introduced me to the SuperCollider environment and provided valuable advice thereafter, and to undergraduate researchers Alex Baldwin and Joshua Brown, who helped develop the interface and synthesis routines, supported by funds from Anglia Ruskin University.

References

- Abel, S. 1972. "Duration Discrimination of Noise and Tone Bursts." *Journal of the Acoustical Society of America* 51(48):1219–1223.
- Barnsley, M. 1986. "Fractal Functions and Interpolation." *Constructive Approximation* 2(1):303–329.
- Barnsley, M. 1988. *Fractals Everywhere*. San Diego, California: Academic Press.
- Barnsley, M. 2006. *Superfractals*. Cambridge, UK: Cambridge University Press.
- Beauvois, M. 2007. "Quantifying Aesthetic Preference and Perceived Complexity for Fractal Melodies." *Music Perception* 24(3):247–264.
- DiScipio, A. 1990. "Composition by Exploration of Non-Linear Dynamic Systems." In *Proceedings of the International Computer Music Conference*, pp. 324–327.
- Dodge, C. 1988. "A Musical Fractal." *Computer Music Journal* 12(3):10–14.
- Fletcher, H., and W. Munson. 1933. "Loudness, Its Definition, Measurement and Calculation." *Bell System Technical Journal* 12.4:377–430.
- Gogins, M. 1991. "Iterated-Functions Systems Music." *Computer Music Journal* 15(1):40–48.
- Harley, J. 1995. "Generative Processes in Algorithmic Composition: Chaos and Music." *Leonardo Music Journal* 28(3):221–224.
- Harte, D. 2001. *Multifractals: Theory and Applications*. London: Chapman and Hall/CRC.
- Mandelbrot, B. 1977. *Fractals: Form, Chance, and Dimension*. New York: Freeman.
- Mandelbrot, B. 1982. *The Fractal Geometry of Nature*. New York: Freeman.
- Monro, G. 1995. "Fractal Interpolation Waveforms." *Computer Music Journal* 19(1):88–98.
- Pöppel, E. 1997. "A Hierarchical Model of Temporal Perception." *Trends in Cognitive Sciences* 1(2):56–61.
- Rhys, P. 1996. "Ebb and Flow, Chamber Concerto, Chicago Fall, Mill Green Music, Three Songs, String Quartet No1." PhD dissertation, Keele University, Department of Music.
- Roads, C. 2004. *Microsound*. Cambridge, Massachusetts: MIT Press.
- Stevens, S. 1957. "On the Psychophysical Law." *Psychological Review* 64(3):153–181.
- Stevens, S. 1972. "Perceived Level of Noise by Mark VII and Decibels (E)." *Journal of the Acoustical Society of America* 51(2B):575–601.
- Stockhausen, K. 1962. "The Concept of Unity in Electronic Music." E. Barkin, trans. *Perspectives of New Music* 1(1):39–48.
- Truax, B. 1990. "Chaotic Non-Linear Systems and Digital Synthesis: An Exploratory Study." In *Proceedings of the International Computer Music Conference*, pp. 100–103.
- Van Noorden, L. P. A. S. 1975. "Temporal Coherence in the Perception of Tone Sequences." PhD dissertation, Technische Hogeschool Eindhoven, Institute for Perception Research, Eindhoven, The Netherlands.
- Voss, R., and J. Clarke. 1975. "1/f Noise in Music and Speech." *Nature* 258(5533):317–318.
- Voss, R., and J. Clarke. 1978. "1/f Noise in Music: Music from 1/f Noise." *Journal of the Acoustical Society of America* 63:258–263.
- Waschka, R., and A. Kurepa. 1989. "Using Fractals in Timbre Construction: An Exploratory Study." In *Proceedings of the International Computer Music Conference*, pp. 332–335.
- Worrall, D. 1996. "Studies in Metamusical Methods for Sound and Image Composition." *Organised Sound* 1(3):183–194.