Categorising complex dynamic sounds

EDUARDO RECK MIRANDA,†‡ JAMES CORREA‡ and JOE WRIGHT§

†Sony Computer Science Laboratory Paris, 6 Rue Amyot, 75005 Paris, France
‡Laboratório de Música Eletroacústica de Santa Maria, Rua Riachuelo 1110–32, 90010-272 Porto Alegre, Brazil
§Nyr Sound, 8 Cringle Avenue, Barnemouth BH6 4HX, UK
E-mail: †miranda@csl.sony.fr, ‡jcorreia1@uol.com.br, §joe@nyrsound.com

Chaosynth is a cellular automata-based granular synthesis system whose capabilities for producing unusual complex dynamic sounds are limitless. However, due to its newness and flexibility, potential users have found it very hard to explore its possibilities as there is no clear referential framework to hold on to when designing sounds. Standard software synthesis systems take this framework for granted by adopting a taxonomy for synthesis instruments that has been inherited from the acoustic musical instruments tradition, i.e. woodwind, brass, string, percussion, etc. Sadly, the most interesting synthesised sounds that these systems can produce are simply referred to as *effects*. This scheme clearly does not meet the demands of more innovative software synthesizers. In order to alleviate this problem, we propose an alternative taxonomy for Chaosynth timbres. The paper begins with a brief introduction to the basic functioning of Chaosynth. It then presents our proposed taxonomy and ends with concluding comments. A number of examples are provided on this volume's *Organised Sound* CD.

1. INTRODUCTION

Granular synthesis works by generating a rapid succession of very short sound bursts called granules (e.g. 35 ms long) that together form larger sound events (figure 1). The results tend to exhibit a great sense of movement and sound flow. This synthesis technique can be metaphorically compared with the functioning of a motion picture in which an impression of continuous movement is produced by displaying a sequence of slightly different images at a rate above the scanning capability of the eye (Miranda 1998).

The composer Iannis Xenakis is commonly cited as one of the mentors of granular synthesis. In the 1950s, Xenakis developed important theoretical writings where he laid down the principles of the technique (Xenakis 1971). The first computer-based granular synthesis system did not appear, however, until Curtis Roads (1991) and Barry Truax (1998) began systematically to investigate the potential of the technique.

Thus far, most of these systems had used stochasticity to control the production of the granules, e.g. to control the waveform and the duration of the individual granules. We devised a system called Chaosynth (Miranda 1995) which uses a different method: it employs a *cellular automaton* (Cood 1968). As will be demonstrated below, cellular automata have proved more efficient than stochasticity to drive the production of the sound granules.

Before we proceed, we would like to note that the term 'granular synthesis' has sometimes been associated with a musical signal processing technique whereby a recorded sound is chopped into tiny pieces, shuffled and reassembled in various ways for playback. Chaosynth does not use prerecorded sounds; it is a real synthesizer in the sense that it synthesises all sounds from scratch.

By way of a related work, we cite an early attempt at a cellular automata-based granular synthesizer described by Andy Hunt and co-workers in a paper presented at the International Computer Music Conference in 1991 (Hunt, Kirk and Orton 1991).

2. THE CELLULAR AUTOMATA ENGINE

Cellular automata (CA) are computer-modelling techniques originally introduced in the 1960s by von Neumann and Ulan (Cood 1968); since then they have been repeatedly reintroduced and applied for a considerable variety of modelling purposes. In general, CA are implemented as a regular array or matrix of variables, or simple processing units, called cells. Each cell may assume values from a finite set of integers and each value is normally associated with a colour (figure 2). The functioning of a cellular automaton is displayed on the computer screen as a sequence of changing patterns of tiny coloured cells, according to the tick of an imaginary clock, like an animated film. At each tick of the

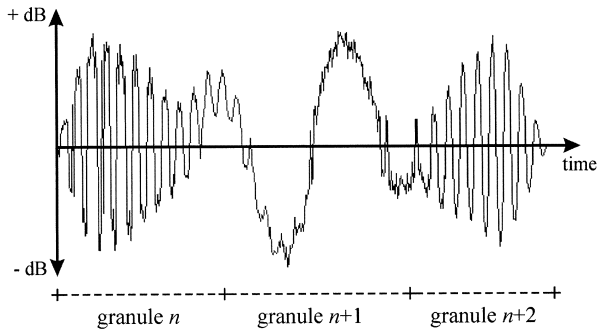


Figure 1. A rapid succession of tiny sounds form larger complex sounds.

Organised Sound **5**(2): 95–102 © 2000 Cambridge University Press. Printed in the United Kingdom.

https://doi.org/10.1017/S1355771800002065 Published online by Cambridge University Press

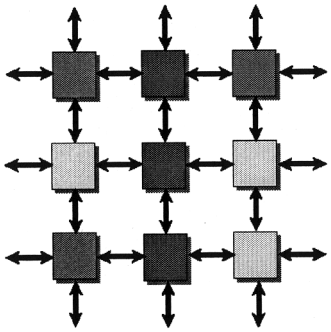


Figure 2. Cellular automata are normally represented as an array of interacting processors referred to as cells.

clock, the values of all cells change simultaneously, according to a set of transition rules that take into account the values of their neighbourhood. The cellular automaton that drives Chaosynth consists of a matrix of identical cells containing a variable that is updated at each cycle, according to the states of its four or eight neighbours. At a given moment t, cells can be in any one of the following conditions: (i) *quiescent*, (ii) in one of n states of *depolarisation*, or (iii) *collapsed*. The metaphor behind this CA is as follows: a cell interacts with its neighbours through the flow of electric current between them. There are minimum, V_{min}, and maximum, V_{max}, threshold values which characterise the condition of a cell. If its internal voltage V_I is under V_{min}, then the cell is quiescent (or polarised). If it is between V_{min} (inclusive) and V_{max} values, then the cell is being depolarised. Each cell has a potential divider which is aimed at maintaining V_I below V_{min}. But when it fails (i.e. if V_I reaches V_{min}), the cell becomes depolarised. There is also an electric capacitor which regulates the rate of depolarisation. The tendency, however, is to become increasingly depolarised with time. When V_I reaches V_{max}, the cell fires and collapses. A collapsed cell at time t is automatically replaced by a new quiescent cell at time $t + 1$. The functioning of the CA is determined by a number of parameters, including the number n of possible cell values or colours ($n > 3$), the resistors R_1 and R_2 for the potential divider, the capacitance k of the electric capacitor and the dimension of the grid. These are Chaosynth user-specified settings.

In practice, the condition of a cell is represented by a number between 0 and $n - 1$ (n = number of different conditions). A value equal to 0 corresponds to a quiescent condition, whilst a value equal to $n - 1$ corresponds to a collapsed condition. All values in between exhibit a degree of depolarisation. The closer the value gets to $n - 1$, the more depolarised the cell becomes. The functioning of the automaton is defined by three rules simultaneously applied to each cell, selected according to its current condition. The rules are as follows:

- *If quiescent*: the cell may or may not become depolarised at the next tick of the clock ($t + 1$). This depends upon the number of polarised cells in its neighbourhood, the number of collapsed cells in its neighbourhood, and the resistance of the cell to being collapsed.
- *If depolarised*: the tendency is to become more depolarised as the clock t evolves.
- *If collapsed*: a burned cell at time t generates a new quiescent cell at time $t + 1$.

The automaton tends to evolve from an initial random distribution of cells in the matrix towards an oscillatory cycle of patterns (figure 3). The reason we use this CA in Chaosynth is because its behaviour resembles the way in which most of the natural sounds produced by acoustic instruments evolve: they tend to converge from a wide distribution of their partials to oscillatory patterns. Our technique for rendering sounds from this behaviour is presented below.

3. RENDERING SOUNDS FROM THE CELLULAR AUTOMATON

Each sound granule produced by Chaosynth is composed of several spectral components. Each component is a waveform produced by a digital oscillator which needs two parameters to function: frequency (Hz) and amplitude (dB). In Chaosynth, the oscillators can produce various types of waveforms such as sinusoid, square, saw tooth, and band-limited noise. The CA controls the frequency and duration values of each granule, but the amplitude values are set up by the user beforehand via Chaosynth's Oscillators Panel, illustrated in figure 4. The system can, however, interpolate between two different amplitude settings in order to render *sound morphing* effects (Howard and Angus 1996).

The mechanism works as follows: at each cycle, the CA produces one sound granule (figure 5). In order to visualise the behaviour of a CA on the computer, each possible cell condition is normally associated with a colour, but in our case we also associate these conditions with various frequency values. For example: yellow = 110 Hz, red = 220 Hz, blue = 440 Hz, and so forth; these are arbitrary associations. Then we subdivide the matrix of the automaton into smaller uniform sub-grids of cells and associate a digital oscillator to each sub-grid (figure 6). At each cycle of the CA, i.e. at each snapshot of the CA animation (sic), the digital oscillators associated with the sub-grids simultaneously produce signals, which are added in order to compose the spectrum ω of the respective granule as

$$\omega = \sum_{n=1}^{N} S_n ,$$

where S_n stands for the spectrum of the signal produced by oscillator n, and N is the total number of oscillators associated with the matrix.

The frequency values F for each oscillator p are determined by the arithmetic mean over the frequency values

https://doi.org/10.1017/S1355771800002065 Published online by Cambridge University Press

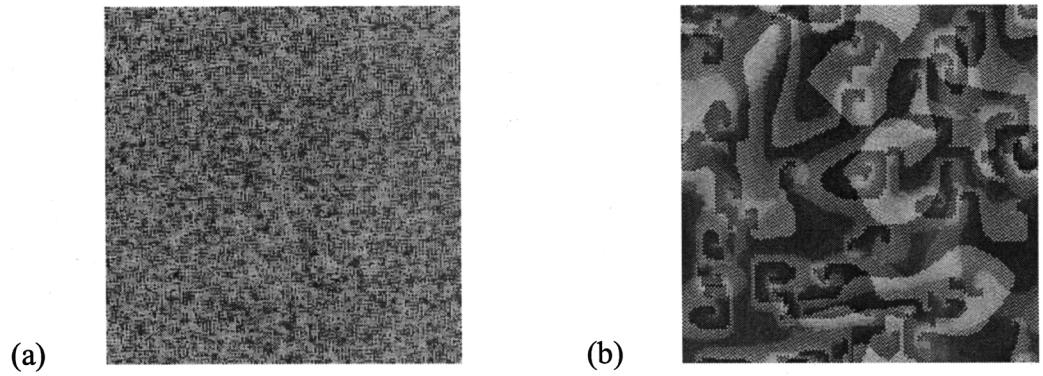


Figure 3. The CA of our system tends to evolve from (a) an initial random distribution of cells in the grid towards (b) an oscillatory cycle of patterns.

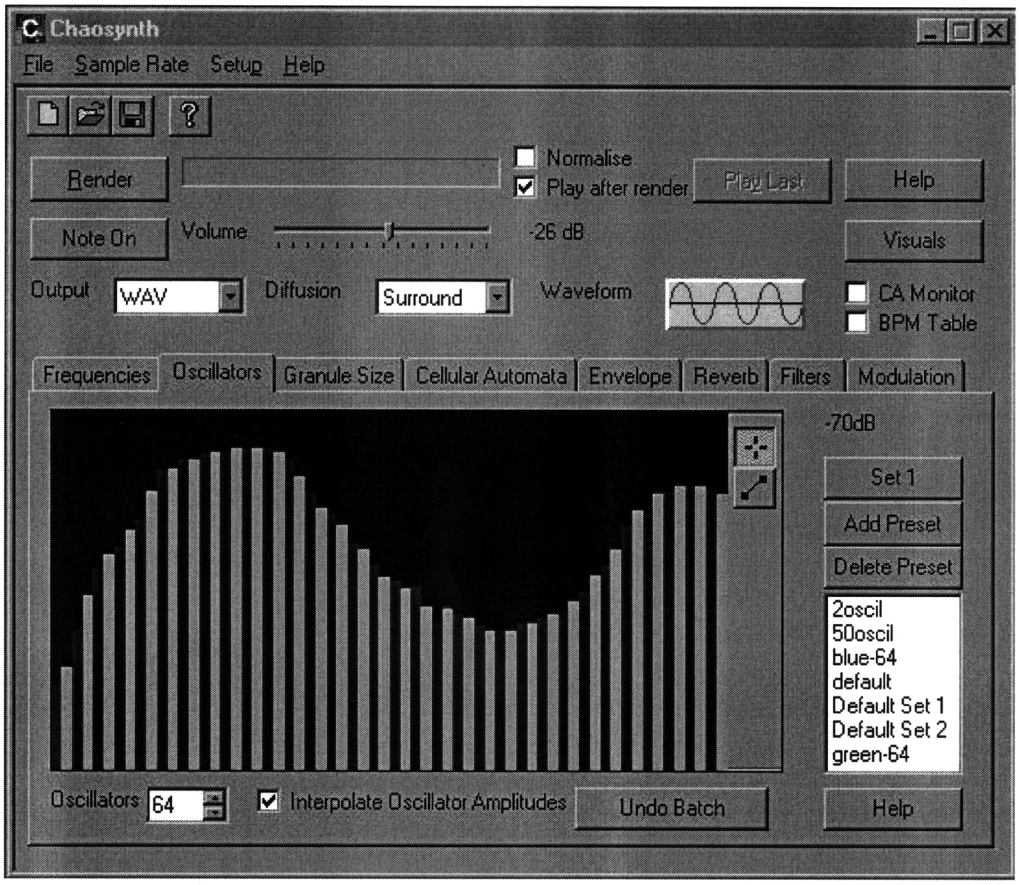


Figure 4. A sample screenshot of Chaosynth featuring the Oscillators Panel. Each column on the main window controls the amplitude for an oscillator.

associated with the states of the cells of their corresponding sub-grid as

$$F_p = \frac{\sum_{n-1}^{N} \phi_n}{N},$$

where ϕ_n represents the frequency of cell n, and N is the total number of cells of the sub-grid. Suppose, for example, that each oscillator is associated with 9 cells and that at a certain cycle t, 3 cells correspond to 110 Hz, 2 to 220 Hz and the other 4 correspond to 880 Hz. In this case, the mean frequency value for this oscillator at time t will be 476.66 Hz. An example of a grid of 400 cells allocated to 16 oscillators of 25 cells each is given in figure 6. The duration of a whole sound event is determined by the number of CA iterations and the duration of the granules; for example, 100 iterations of 35 ms granules results in a sound event of 3.5 s duration.

This mapping method is interesting because it explores the emergent behaviour of the CA in order to produce sounds in a way which resembles the functioning of some acoustic instruments. The random ini-

https://doi.org/10.1017/S1355771800002065 Published online by Cambridge University Press

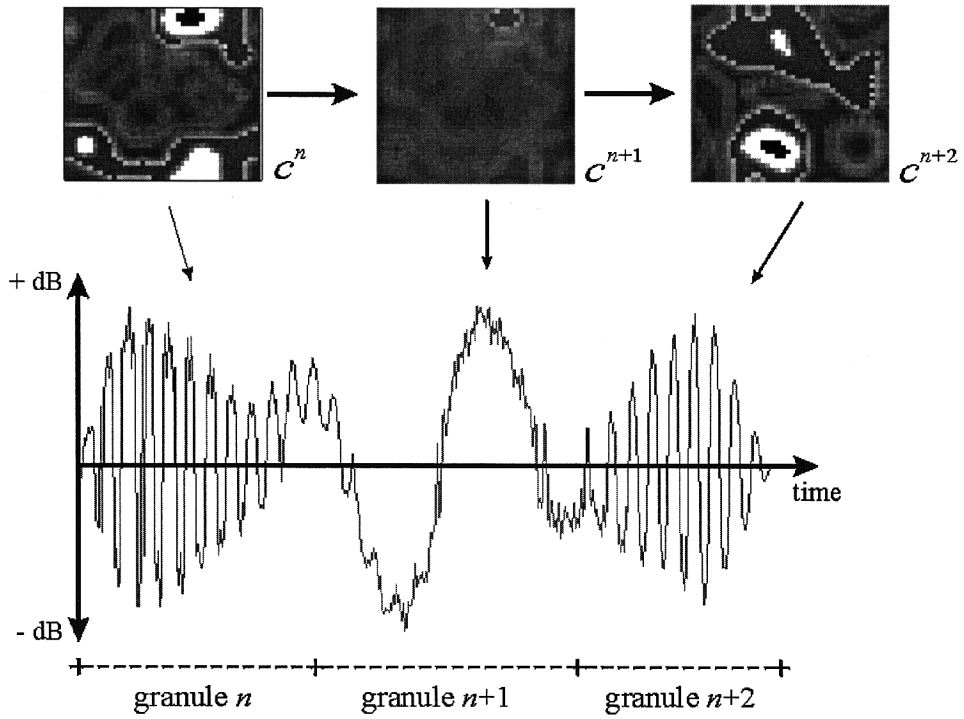


Figure 5. Each snapshot of the cellular automaton produces a sound granule. (Note that this is only a schematic representation, as the granules displayed here do not actually correspond to these particular snapshots.)

tialisation of cells in the grid produces an initial wide distribution of frequency values, which tend to settle to an oscillatory cycle. This behaviour resembles the way in which the sounds produced by most acoustic instruments evolve during their production: their harmonics converge from a wide distribution (as in the noise attack time of the sound of a bowed string instrument, for example) to oscillatory patterns (the characteristic of a sustained tone). Variations in tone colour are achieved by varying the frequency values, the amplitudes of the oscillators, and the number of cells per oscillator. Different rates of transition, from noise to oscillatory patterns, are obtained by changing the values of R_1, R_2 and k.

4. CHAOSYNTH AND SOUND DESIGN

Chaosynth has proved to be an incredibly powerful synthesizer, whose abilities to produce unusual sounds are limitless. As examples of works composed using sounds almost entirely produced with Chaosynth we cite *Olivine Trees* by Miranda and a movement of the suite *Winter Studies* by Correa; the former has been awarded the bronze medal at the 20th Luigi Russolo International Composition Competition, in Italy. (Recordings of these pieces can be obtained directly from the authors.)

The main criticism that we have received from Chaosynth's potential users, however, is that due to its newness and flexibility, it has been very hard to explore its potential: users other than Chaosynth's makers themselves do not have a clear reference from which to begin the sound design process. The problem is that standard software synthesis systems take for granted a taxonomy for synthesis instruments that is inherited from the acoustic musical instruments tradition: i.e. woodwind, brass, string, percussion, etc. Sadly, the most interesting synthesised sounds that these systems can produce are simply referred to as *effects*. This scheme clearly does not meet the demands of more innovative software synthesizers. In order to alleviate this problem, we propose an alternative taxonomy for Chaosynth timbres. This new taxonomy departs from a framework for sound categorisation inspired by Pierre Schaeffer's (1966) concept of sound maintenance, and to some extent by an article written by Jean-Claude Risset in the book, *Le timbre, métaphore pour la composition* (1991). We assume that the most important characteristic of the sounds produced by Chaosynth, and granular synthesis in general, is their spectral evolution in time (Miranda 1998). Our proposed taxonomy is introduced below. Note that we will use the term 'instrument' to refer to specific Chaosynth setups that produce the sounds of the respective categories.

4.1. The proposed taxonomy

The general structure of the proposed taxonomy is summarised in the table. We have defined five general classes: Fixed Mass, Flow, Chaotic, Explosive and General Textures.

https://doi.org/10.1017/S1355771800002065 Published online by Cambridge University Press

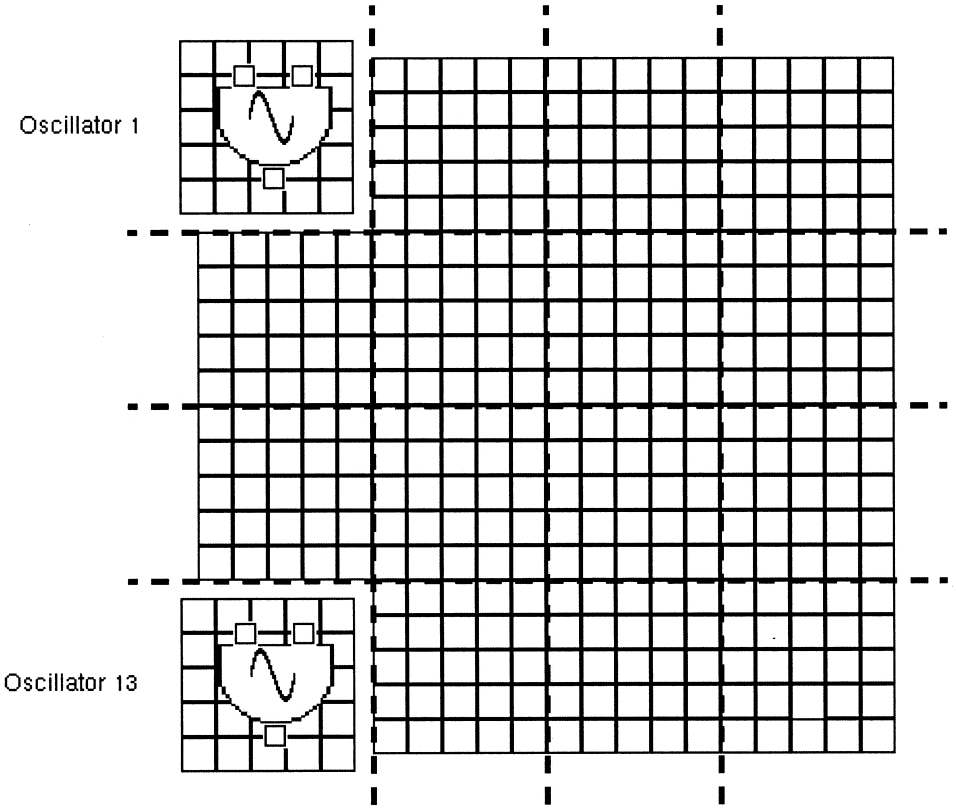


Figure 6. An example of a grid of 400 cells allocated to 16 digital oscillators.

Table

Fixed Mass	Flow	Chaotic	Explosive	General Textures
Lighten	*Cascade*	*Insects*	*Metallic*	*Textures*
Darken	*Landing*	*Melos*	*Woody*	*Gestures*
Dull	*Raising*	*Boiler*	*Glassy*	
Elastic	*Lift*	*Windy*	*Blower*	
Melted	*Crossing*	*Noises*	*Drum*	
	Drift			

4.1.1. Fixed Mass

This class comprises those instruments that produce sounds formed by a large amount of small granules. The overall outcome from these instruments is perceived as sustained sounds with a high degree of internal redundancy; hence the label 'fixed mass'. Notice here that by fixed mass we do not mean fixed pitch, but rather a stable and steady spectrum where the frequencies of the grains are kept within a fixed band. Sometimes this creates a sense of pitch, but this phenomenon is not a mandatory condition. We have defined five sub-classes for this class: whilst *Lighten* are bright instruments that produce sounds rich in high frequencies, *Darken* are those instruments that produce sounds rich in low frequencies; *Dull* instruments produce muffled sounds, and finally *Elastic* and *Melted* are named after the auditory sensation they create. Our first example on this volume's *Organised Sound* CD is a typical fixed mass sound.

4.1.2. Flow

These are instruments whose outcome gives a clear sensation of movement due to a continuous change of the fundamental frequencies of the grains. These sounds are said to have a variable mass, but not an unpredictable one as the frequencies of the granules tend to move collectively in one direction. Six sub-classes for this class were defined according to the direction of the spectral movement: *Cascade* and *Landing* both comprise instruments whose outcome is of a descending direction, with the difference that the movement of the former is smooth and light, whereas in the latter it is fast and vigorous. Conversely, there are *Raising* and *Lift* instruments, where the movement is ascending. The *Crossing* instruments produce sounds that display a cross movement between two groups of frequencies, one group going upwards and the other going downwards. Finally, the sub-class *Drift* produces sounds whose sense of direction is not entirely clear; there is some flow, but frequencies of the grains tend to 'drift' up or down. Our third sound example on the *Organised Sound* CD is an example of a *Cascade*-type of sound, and figure 7 shows its spectrogram.

https://doi.org/10.1017/S1355771800002065 Published online by Cambridge University Press

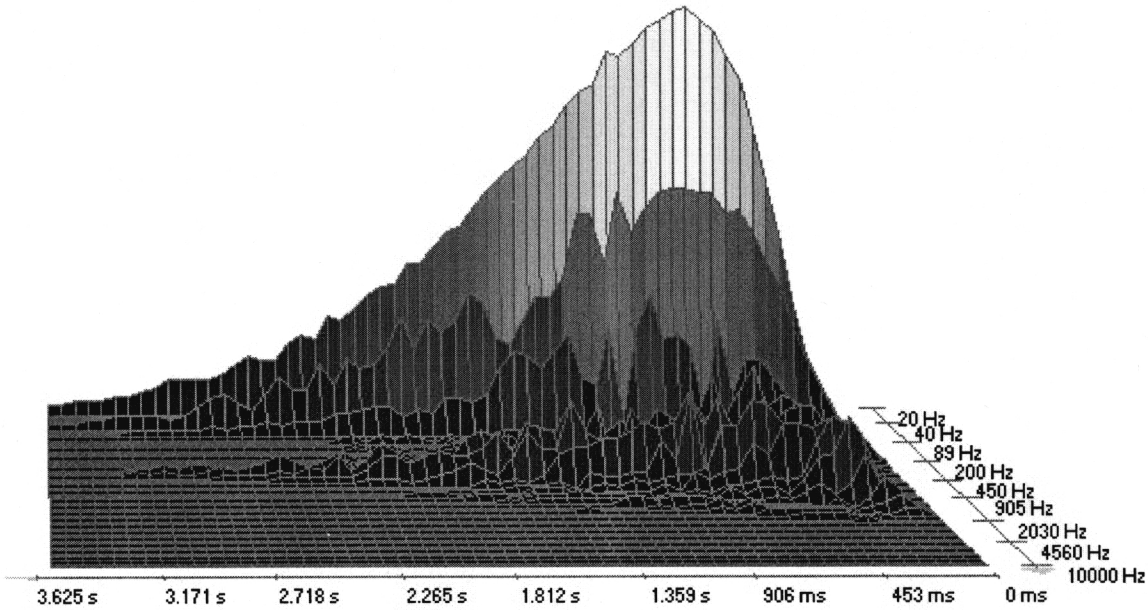


Figure 7. A *Cascade*-type of sound.

4.1.3. Chaotic

This class of instruments produce sounds with a very large degree of spectral movement. These sounds are of highly variable mass, the movement of the grain frequencies is unpredictable, and they do not have an overall direction. We have defined five sub-classes of chaotic instruments: *Insects* are those instruments whose outcome resembles the noises produced by insects. These sounds have more activity in the high frequencies. *Melos* produce sounds whereby the individual grains can be perceived as individual notes, as if they were fast melodic fragments, whereas the sounds from a *Boiler* instrument resemble the sound of boiling liquids. *Windy* and *Noises* produce sounds that resemble or are derived from white-noises: the former produces an auditory sensation of blowing air, whereas the latter produces noise similar to an AM radio off station. Figure 8 portrays the spectrogram of a *Melos* sound and it can be heard as the sixth example on the *Organised Sound* CD.

4.1.4. Explosive

This class resembles traditional acoustic percussion instruments. The sounds here are of short duration, fast attack, practically no sustain, and short decay; in general terms, these sounds have a short impulsion and are either of a fixed or variable mass. This class comprises five sub-classes, organised according to the resemblance of their sound to those from real percussive materials: *Metallic, Woody, Glassy, Blower* and *Drum*. *Blower* instruments produce fast explosion-like sounds with a very high energy. In the *Drum* class we include those instruments whose sounds are very similar to membrane-type drums. Our tenth example on the *Organised Sound* CD is a typical example of an Explosive *Metallic* sound; its spectrogram is shown in figure 9.

4.1.5. General Textures

These are instruments that produce what we refer to as *Textures* and *Gestures*. The outcome of *Textures* is a cloud of sounds with long duration and variable mass. The classic example here is the typical bubbly and cloudy-like stereotype texture that is normally associated with granular synthesis. The second sub-class, *Gestures*, generate sounds that resemble some standard musical instrument techniques, such as a cello glissando, or metaphorical sounds such as the noise of a steam train, a helicopter or a Klingon phaser. An example of a cello-like glissando produced by Chaosynth is given as the thirteenth example on the *Organised Sound* CD.

5. CONCLUSION

Chaosynth's ability to produce unusual complex dynamic sounds is limitless. Most sounds produced by the system cannot be found in the real acoustic world and yet they often sound 'natural' and pleasing. The perceived naturalness of Chaosynth sounds is due to the behaviour of the cellular automaton that controls the synthesis parameters. However, due to its newness and flexibility, potential users have found it hard to explore as there is no referential framework to hold on to when designing new sounds. In order to alleviate this problem

https://doi.org/10.1017/S1355771800002065 Published online by Cambridge University Press

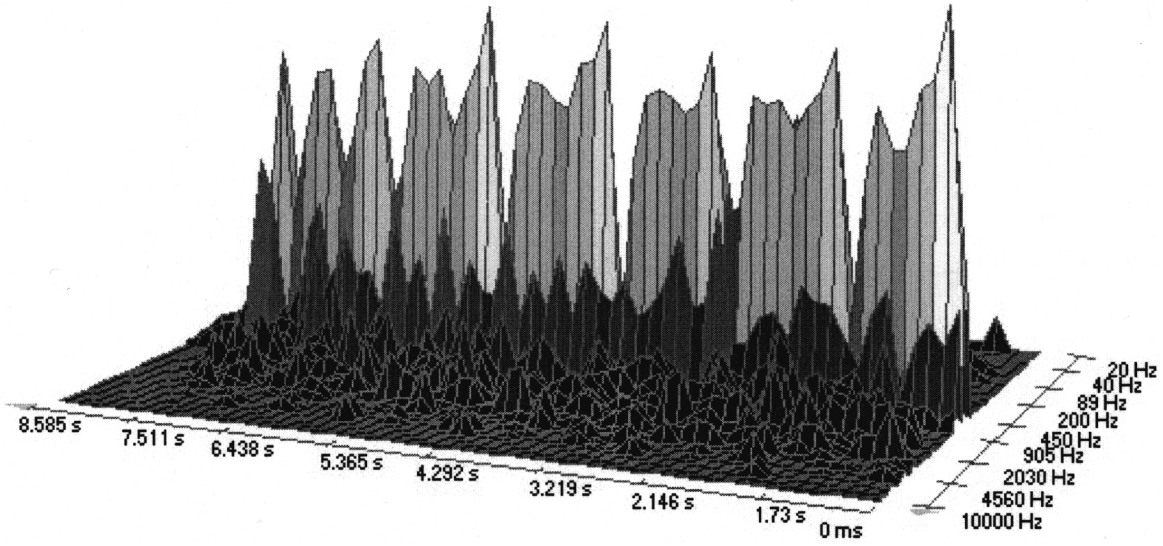


Figure 8. The spectrum of a *Melos* sound.

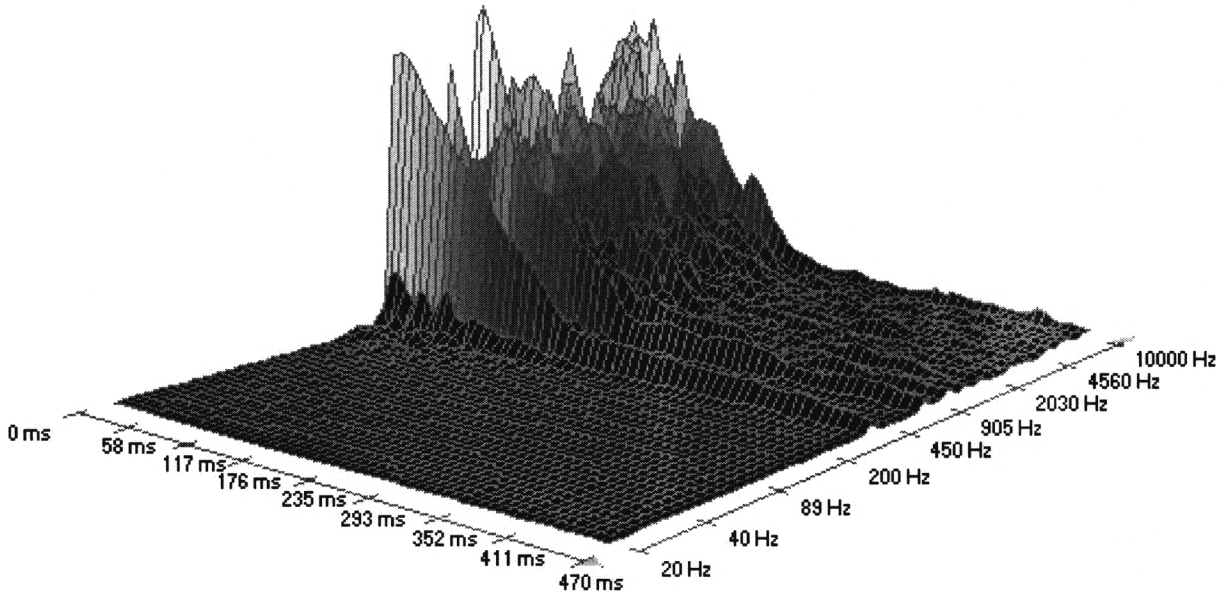


Figure 9. The spectrum of an Explosive *Metallic* sound.

we have proposed an alternative taxonomy for Chaosynth sounds.

The development of the proposed taxonomy is the result of massive experimentation with the system and there are still some inconsistencies to be addressed. We feel, for example, that the General Textures class still is too vague in comparison with the others. We are currently devising a large catalogue of instruments classified according to this taxonomy to distribute to Chaosynth users. Our aim is to gather as much feedback as possible from the users in order to implement further improvements.

Chaosynth is now available for Macintosh and for Windows platforms. For more information please refer to the Internet site: http://www.nyrsound.com

APPENDIX

Sound examples on the *Organised Sound* CD

Chaosynth Sample 01: Fixed mass, *Darken*
Chaosynth Sample 02: Fixed mass, *Elastic*
Chaosynth Sample 03: Flow, *Cascade*
Chaosynth Sample 04: Flow, *Cascade*
Chaosynth Sample 05: Chaotic, *Windy*
Chaosynth Sample 06: Chaotic, *Melos*

https://doi.org/10.1017/S1355771800002065 Published online by Cambridge University Press

Chaosynth Sample 07: Chaotic, *Insects*
Chaosynth Sample 08: Chaotic, *Noises*
Chaosynth Sample 09: Chaotic, *Melos*
Chaosynth Sample 10: Explosive, *Metallic*
Chaosynth Sample 11: Explosive, *Blower*
Chaosynth Sample 12: Explosive, *Metallic*
Chaosynth Sample 13: General Textures: *Gestures*

REFERENCES

Cood, E. F. 1968. *Cellular Automata*. London: Academic Press.

Howard, D. M., and Angus, J. 1996. *Acoustics and Psychoacoustics*. Oxford: Focal Press.

Hunt, A., Kirk, R., and Orton, R. 1991. Graphical control of granular synthesis using cellular automata. *Proc. of the ICMC*, pp. 416–18.

Miranda, E. R. 1995. Granular synthesis of sounds by means of a cellular automaton. *Leonardo* **28**(4): 297–300.

Miranda, E. R. 1998. *Computer Sound Synthesis for the Electronic Musician*. Oxford: Focal Press.

Risset, J.-C. 1991. Timbre et synthèse des sons. In J.-B. Barrière (ed.) *Le timbre: métaphore pour la composition*. Paris: IRCAM/Christian Bourgois Editeur.

Roads, C. 1991. Asynchronous granular synthesis. In G. de Poli *et al*. (eds.) *Representations of Music Signals*. Cambridge, MA: MIT Press.

Schaeffer, P. 1966. *Traité des objets musicaux*. Paris: Editions du Seuil.

Truax, B. 1988. Real time granular synthesis with a DSP computer. *Computer Music Journal* **2**(2): 14–26.

Xenakis, I. 1971. *Formalized Music*. Bloomington, IN: Indiana University Press.

https://doi.org/10.1017/S1355771800002065 Published online by Cambridge University Press