Wavelets as a Multiresolution Analysis and Synthesis Technique for Sound Timbres Edition

REGIS ROSSI ALVES FARIA
Computer Music Group
LSI - Laboratório de Sistemas Integráveis
Escola Politécnica da Universidade de São Paulo
(regis@lsi.usp.br)
JOÃO ANTÔNIO ZUFFO
LSI - Laboratório de Sistemas Integráveis
Escola Politécnica da Universidade de SãoPaulo
(jazuffo@lsi.usp.br)
Av. Professor Luciano Gualberto, 158, Tv.3
05508-900 São Paulo - SP Brasil

Abstract

Real acoustic instruments' sound pureness still stands an appreciable distance ahead of the artificial instruments' one, as synthesizers, in timbre quality as much as in the virtually infinite variability and modulation possibilities.
Analysis and synthesis techniques have been employed for years as a tool to analyse timbres' temporal-spectral characteristics, and re-synthesize them from these extracted parameters.
In this paper we show the utilization of Wavelets - a mathematical tool employed in signal processing with additional advantages over the classic Fourier transforms - as a technique for sound analysis, edition and synthesis under a multiresolution scheme. The central idea is to explore the capacities of this technique in altering fine timbre components in the time-frequency (scale) spaces, in order to produce improved timbres.

Introduction

Several analysis/synthesis techniques for signal processing have been employed through years now in the field of music. Successive development of computational resources -both in hardware and software- adding to a growing interdisciplinary interaction with other knowledge branches (as applied mathematics, biology and others) propitiate the emergency of efficient and sophisticated techniques to process several types of signal - including musical ones.We introduce wavelets as one these emergent tools, coming from the marriage of several mathematical branches and engineering and computational techniques.

So far the realism and sound pureness produced with real-world musical instruments have not been surpassed: the infinite number of physical parameters, and also the musicians' psychological impressions, turn the musical experience into a complex phenomenon, hard to be efficiently modeled in computer applications.

One way towards the enrichment of music representation on computers is the processing of the musical signals in both time and frequency domains in order to work out instruments' timbres - whether they are acoustical or not. Acoustic instruments have complex tones. The spectral evolution in an attack reveals different evolution in time of the various harmonics. Even at the "steady-state" the tone undergoes many micromodulations and other variations that are perceptible. Also at the end of a note the harmonics do not decrease uniformly: each frequency component reveals a different collapse in time. The musician in stage imposes several modulations to the sound. His or her style and mood are translated into complex modulations and different intensities during performance. Add to this the influence of local acoustics, giving birth to effects like reverberation, etc. The final sound is a signal full of details hidden in its time evolution.

One can easily sense large variations in intensity and recognize major pitches. One can also recognize timbres, sense some vibratos and other effects. However, a human listener often interprets these parameters in an emotional context, where feelings are translated into a great spectral variability in time, e.g, fine modulations and "micro-structures" mixed inside the signal.

It happens that these details are present inside the signal at several different scales. As in a map, one can see large areas at larger scales, and details at smaller scales. It is desirable an unbounded resolution signal edition system, where changes can be made over the signal at different scales.

The wavelet framework is adequate for this purpose. We show how wavelets can be used to implement a multiresolution signal edition system, dealing with the signal at different scales, and in an efficient and concise manner.

Wavelets

The application of wavelets to signal processing is only a few years old. The theory is based on representing generic functions in terms of basic building blocks. This is an "atomic decomposition" algorithm, as the Fourier transform.

Looking back over the history we find seven different origins of wavelets: the idea of focusing a signal over different scales came up independently in many fields of physics, mathematics and engineering. In 1984, Grossman and Morlet introduced an unified framework, giving birth to the first definition of a wavelet. Mallat and Meyer (1986) formulated the multiresolution analysis, a natural framework for understanding and constructing wavelet bases. From then on the number of contributions and applications of this theory has grown substantially.

Fourier techniques are an ideal tool for studying stationary signals, decomposing them into linear combinations of trigonometric waves (sines, cosines). Musical signals may be classified as nonstationary signals, where transients and other non predictable events might happen, and the wavelets techniques are an ideal tool for studying such signals, decomposing them into linear combinations of wavelets.

Wavelet Representation

Wavelets consist in a family of basis functions $\psi_{j,k}$, in $L^2(\mathbf{R})$, obtained from a single mother-wavelet ψ by dilations (scaling) and translations (shifts).

$$\psi_{j,k} = 2^{j/2}\psi(2^{j/2}t - k) \qquad \text{,where } j,k \in \mathbf{Z}.$$

There are infinite possible families like this, and the usefulness of them is linked to some desirable properties they must possess. The mother wavelet ψ should verify some important properties: be a localized pulse that decreases to zero and has integral zero. Also, it should verify the admissibility condition (C_ψ) below:

$$C_\psi = \int_0^\infty |\psi(\xi)|^2 . |\xi|^{-1} . d\xi < \infty \quad \text{and} \quad \int_{-\infty}^{\infty} \psi(t).dt = 0$$

where $\Psi(\xi)$ is the Fourier transform of ψ.

The family $\{\psi_{j,k}\}$ above is an orthonormal basis of $L^2(\mathbf{R})$. That is $<\psi_{j,k}, \psi_{l,m}> = \delta_{j,l}.\delta_{k,m}$ $(j,k,l,m \in \mathbf{Z})$, and every function $f(t) \in L^2(\mathbf{R})$ can be written as

$$f(t) = \sum_j \sum_k d_{j,k}\psi_{j,k}(t) \qquad \text{(S)}$$

where the wavelet coefficients $d_{j,k}$ are given by

$$d_{j,k} = < f(t), \psi_{j,k}(t) > \qquad \text{(A)}$$

We are interested in wavelet functions whose binary dilations and dyadic translations are sufficient to represent all functions *f(t)* in $L^2(\mathbf{R})$. Observe that the family $\{\psi_{j,k}\}$ covers infinite scales $a=2^{-j}$, and performs a frequency band analysis when generating the wavelet coefficients. We have frequencies separated in consecutive octaves (2^j), a natural scaling factor in music.

Wavelet Transform

Unlike Fourier analysis, the integral form of the wavelet transform is intimately related to the wavelet series representation: the coefficients $d_{j,k}$ of *f(t)* are precisely the values evaluated by the integral wavelet transform (not shown) at the dyadic positions in the binary dilated scales. The wavelet representation simultaneously localizes *f* and its Fourier transform with the multiscale analysis capability. Since there are real-time algorithms for calculating the coefficient sequences and for recovering *f* from these sequences, we will center our attention in the discrete signal analysis and synthesis through discrete wavelet transforms.

The formulas in equations (S) and (A) are a simplified form of the synthesis and analysis processes, respectively.

Timbres Edition

An unbounded resolution signal edition system should support:

- an efficient representation of the signal at different scales. *It should be possible to "see" details at different scales.*
- operation on the signal at different scales. *It should be possible to process the signal sequence (at scale a) with known processing methods -as filtering, modulation, addition and subtraction of other signal sequences, applying envelopes, etc.*
- propagation of changes. *Changes to the signal on a level (scale) must propagate to other levels, in a non-redundant manner, and without loss of information. In other words, levels must be connected.*

A multiresolution analysis (and synthesis) approach is the choice for implementing the above processing structure. In the next session, we introduce the concepts of a multiresolution analysis framework, and its properties. It is presented the pyramidal algorithm, and its implementation with filter banks. Thus a natural connection between filter trees in discrete time and the multiresolution in continuous time is made, showing that filter trees can implement multiresolution analysis. Finally, it is shown how filter trees lead to wavelets.

Multiresolution Analysis

A multiresolution analysis consists of a sequence of successive approximation (closed) spaces V_j. Each subspace V_j is contained in the next subspace V_{j+1}. A function in one subspace is in all higher (finer) subspaces:

$$\ldots V_{-1} \subset V_0 \subset V_1 \subset \ldots \subset V_j \subset V_{j+1} \subset \ldots$$

A function *f(t)* decomposed into these spaces has a piece in each subspace. The piece (projection of *f(t)*) in V_j is called $f_j(t)$. The union of all subspaces is $L^2(\mathbf{R})$, and the intersections between subspaces is a null space ($\bigcap_{j\in Z} V_j = \{0\}$). There are additional requirements:

- Completeness: $f_j(t) \rightarrow f(t)$ as $j \rightarrow \infty$, and Emptiness: $\| f_j(t) \| \rightarrow 0$ as $j \rightarrow -\infty$
- V_{j+1} consists of all rescaled functions in V_j: $f(t) \in V_j \Rightarrow f(2t) \in V_{j+1}$
- Shift invariance: $f(t) \in V_j \Rightarrow f(t - 2^{-j}.k) \in V_j$
- There exist a basis for each subspace V_j: $\{\phi_{j,k}\ , k\in \mathbf{Z}\}$ is an orthonormal basis **for** $\mathbf{V}_j\ , j\in\mathbf{Z}$.

We call ϕ the "*scaling function*" of the multiresolution analysis.

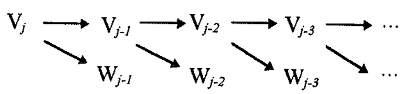
The function $f_{j+1}(t)$ in V_{j+1} has a better resolution than f_j in V_j. The missing portion necessary to aproximate $f_{j+1}(t)$ from f_j is in a new subspace W_j: $\mathbf{\Delta f_j = f_{j+1}(t) - f_j}$, where $\mathbf{\Delta f_j \in W_j}$. From the subspace point of view, $V_j \oplus W_j = V_{j+1}$. The new family of subspaces are the orthogonal complement of V_{j+1} in V_j. It follows that

$$V_{j+1} = W_j \oplus W_{j-1} \oplus W_{j-2} \oplus \ldots \quad \text{or} \quad V_{j+1} = W_j \oplus W_{j-1} \oplus W_{j-2} \oplus \ldots \oplus V_0\text{, which implies that}$$

$$f_{j+1}(t) = \Delta f_j + \Delta f_{j-1} + \ldots + \Delta f_1 + \Delta f_0 + f_0$$

Naturally, it follows that the union of all subspaces W_j is also the whole space $L^2(\mathbf{R})$, and the requirements above also apply to the family of (closed) subspaces W_j. The family of functions $\{\psi_{j,k}\ , k\in\mathbf{Z}\}$ constitutes an orthonormal basis for W_j . More: The whole collection $\{\psi_{j,k}\ , j,k\in\mathbf{Z}\}$ constitutes an **orthonormal basis** for $L^2(\mathbf{R})$, which is called **a wavelet basis** of $L^2(\mathbf{R})$, with $\psi_{j,k}(t) = 2^{j/2}\,\psi(2^j t - k)$ (to maintain the coherence with $\phi_{j,k}$).

The structure that connects the subspaces V_j to W_j is a pyramid, as below:



We begin our calculations at some unit scale. The wavelets are a basis for the whole space $L^2(\mathbf{R})$, but the scaling function ϕ at j=0 and the wavelets with $j\geq 0$ are a more practical basis. We can recover a *f(t)*, decomposed into a set of subspaces V_j and W_j:

$$f(t) = f_0(t) + \sum_{j=0}^{+\infty} \Delta f_j = \sum_{-\infty}^{+\infty} \Delta f_j = \sum_{-\infty}^{+\infty} d_{j,k}\psi_{j,k}$$

where $d_{j,k}$ **are the wavelet coefficients** of *f(t)*. From $f_0(t)$ (level V_0) and we extract the other $f_j(t)$ from mathematical operations over the pyramid. We can stop at some scale, say 2^{-J} (level J, where lies $f_J(t)$) with enough high frequencies components (finer) to reproduce an exact signal. In the opposite direction (down the pyramid), decomposing f_0 into successive coarser aproximations, we obtain less resolved descriptions of *f(t)*, at larger scales -as in a map.

The Wavelet decomposition and reconstruction algorithms

We need to have our signal *f(t)* described at different scales. It is desirable the ability to go from a coarse aproximation of *f(t)* towards a finer one, where more details are available (better resolution), and vice-versa, and perform operations on the signal at chosen scales. The multiresolution framework offers the environment to accomplish these operations. In this scheme, projections of *f(t)* into subspaces V_j and W_j are related by:

$$f_J = f_{J-1} + g_{J-1}\text{, and by iteratation follows that } f_J = g_{J-1} + g_{J-2} + \ldots + g_0 + f_0 .$$

There is a intimate relation between $\phi(t)$ and $\phi(2t-k)$ and between ψ and $\phi(2t-k)$ known as the *two-scales relation*:

$$\phi = \sum_n h_n \phi_{1,n} \quad \text{and} \quad \psi = \sum_n g_n \psi_{1,n}$$

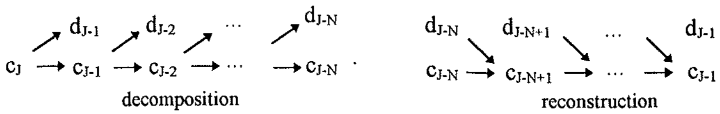
since $\phi \in V_0 \subset V_1$ and $\psi \in W_0 \subset V_1$. From these relations we derive the decomposition formulas:

$$c_{j-1,k} = \langle f, \phi_{j-1,k} \rangle = \sum_n h_{n-2k} c_{j,n} \quad \text{and} \quad d_{j-1,k} = \langle f, \psi_{j-1,k} \rangle = \sum_n g_{n-2k} d_{j,n}$$

We can define now f_j and g_j as $f_j = \sum_k c_{j,k}\phi_{j,k}$ and $g_j = \sum_k d_{j,k}\psi_{j,k}$. It is clear that $C_{j,k} \in V_j$ and $\mathbf{d_{j,k} \in W_j}$. Since $f_j = f_{j-1} + g_{j-1}$, the reconstruction algorithm is

$$c_{j+1,k} = \sum_n [\overline{h_{k-2n}} c_{j,n} + \overline{g_{k-2n}} d_{j,n}]$$

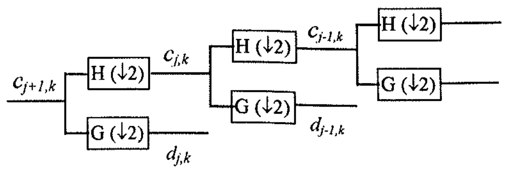
This is **wavelet decomposition** and **reconstruction**, which can be schematize in a analogous pyramid algorithm:



It is the recursive nature of wavelets algorithms that make them computationally fast and efficient.

Filter banks implementation

A multiresolution pyramid analysis can be implemented with filter banks, structured in a tree format, as below:



H is a low-pass filter, which computes averages. G is a high-pass filter, which computes differences. The downsampling steps (↓2) get even numbered components from the input sequence. The averages and downsampling go on indefinitely, each step taking us from a finer level to a coarser one (as in the multiresolution step from V_j to V_{j-1}). In real computations we can start from a fine scale 2^{-J} at level J and go down the tree towards level $j=0$, which, for example, can be normalized with $\Delta t=1$. If the input vector x(n) has lenght $N=2^J$, we reach level $j=1$ with 2 inputs, almost the coarsest level. In the filter tree, the input sequence x(n) corresponds to the coefficients of $c_{J,k}$.

In practice, we can assume the values of signal sequence f(n) as the coefficients $c_{J,k}$ (actually f_J), and the analysis will provide the next level coefficients $c_{j,k}$ and $d_{j,k}$.. This last contains the details of f(n) separated into band-frequencies. In the synthesis, we invert the processing direction, as in continuous multiresolution synthesis, the only difference being in executing an upsampling (↑2) instead of a downsampling from a level to another.

The operations in a filter tree, as above, can be conveniently implemented through matrices multiplications.To transform N-lenght input sequence to its N coefficients (generated in the decomposition) it is necessary a NxN matrix. It is like solving a linear system. The inverse transform operations (the synthesis) involves the inverse matrix:

Synthesis: $\mathbf{x} = \mathbf{W}.\mathbf{b}$ Analysis: $\mathbf{b} = \mathbf{W}^{-1}.\mathbf{x}$

where **b** is the vector of the N coefficients, **W** is the wavelets matrices obtained from the filter banks coefficients, and **x** is the input vector (signal sequence).

The choice of the high-pass and low-pass filters will exert a strong influence on the properties verified by these matrices. For example, if we have an orthogonal filter bank (with orthogonal H and G) the correspondent

(filter bank) matrix will be orthogonal. By means of a proper normalization, the orthogonality turns into orthonormality, and, thus, $\mathbf{W}^T.\mathbf{W} = \mathbf{I} \Rightarrow \mathbf{W}^{-1} = \mathbf{W}$. This turns the transform into a *fast transform*, because these matrices can be factorized into 2 or 3 matrices, with many zeros entries, and the number of numerical operations can be dramatically reduced. Actually, it can be proved that the number of multiplications for the *fast wavelet transform* is bounded to less than 2.T.N, where T is the number of the filter coefficients. In other words, the algorithmic complexity is O(N).

Now, let's show the bridge that leads filter trees to wavelets-based multiresolution schemes.

There are many parallels between a filter tree in discrete time and a pyramidal multiresolution in continuous time:

Filter banks (discrete time)	**Wavelet multiresolution scheme (continuous time)**
filter bank tree	multiresolution pyramidal structure
downsampling ($v(n)=y(2n)$, $\omega \rightarrow \omega/2$)	rescaling $t \rightarrow 2t$
lowpass filter	averaging with $\phi(t)$
highpass filter	detailing with $\psi(t)$
orthogonal matrices	orthogonal basis
analysis bank output	wavelet coefficients
synthesis bank output	sum of wavelet matrices
product of filter matrices	fast wavelet transform

The construction of a wavelet basis was connected previously to the existence of a scaling function ϕ. It is now appropriate to show the connection between the low-pass filter choice and the scaling function.

Dilation and wavelet equations

The low-pass filter coefficients $c_{j,k}$ (n) are the link that leads to wavelets. The operation {H(↓2)} in the pyramid algorithm might be, in theory, executed indefinitely. It consists of a recursive operation. Suppose that ϕj,k is one basis at level j in the filter banks. At level j-1 the basis is ϕj-1,k, as if it was computed by a filtering/downsampling operation. It is a two-scale relation, required for multiresolution analysis:

$$\phi^{(i+1)}(t) = \sum_k h(k)\phi^{(i)}(2t-k)$$

where $(i+1)$ and (i) indicates a recursive calculus. This is called the cascade algorithm. If those functions $\phi^{(i)}$ converge as $i \rightarrow \infty$ take the limit of the iteration, which is the *dilation equation*:

$$\phi(t) = \sum_k h(k)\phi(2t-k)$$

This, in multiresolution language, means that the space V_0 is contained in V_1. The wavelet subspace W_o is also in V_1, and there exist a similar relation connecting $\psi(t)$ and $\phi(2t-k)$, but this time through the high-pass filter portion, i.e. the second channel in the filter bank. This is called the *wavelet equation*:

$$\psi(t) = \sum_k g(k)\phi(2t-k)$$

This is how filter banks leads to wavelets representation. The trick in constructing a wavelet-basis is in the choice of the filters. Not all filters leads to wavelets. The filter must verify some important properties in order to be useful. The orthogonality theorem says that if the cascade algorithm converges, and if the coefficients c(k) and k(k) come from an orthogonal filter bank, then they lead to an orthonormal basis $\phi_{j,k}$ and an orthonormal wavelets basis $\psi_{j,k}(t)$. Generally, if $H(\omega=\pi) = 0$, there is convergence of $\phi^{(i)}(t)$ to $\phi(t)$, and when $|H|^2$ is halfband the $\phi(t)$ is orthogonal to its translates.

There are numerous other conditions and special properties of some filters that lead to specific wavelets representations, but it is not our aim to go further in mathematical and filter banks issues for the moment.

One only has to keep in mind that we can construct wavelets using filter banks, and as we can conceive infinite types of (qualified) filters, it is possible to obtain infinite wavelets basis. This is one advantage over Fourier techniques, which only take into account trigonometric sines and cosines.

Conclusions

We think that a wavelet-based multiresolution analysis and synthesis environment is an efficient framework to examine musical signals, inherently non-stationary, and possessing finite energy. One attractive is the possibility of extrapolating the higher resolution limit, creating a "detail" at the top-most level (in W_J) and expanding to a new higher level. A modification in the signal at level j should affect only the elements in W_j, since through the wavelet basis this change will be propagated onwards.

A detail in W_j can be created by processing the sequence that represents the signal in this level with known signal processing techniques. Wavelets operations are feasible in real-time. Real-time sound signals edition, however, is dependent on the efficiency of the editions operations itself.

One important point to stand out is on the choice of the filter bank/wavelet basis. Some wavelets are better than others to treat a specific signal type. For example, an algorithm that is excellent for data compression can be a disaster when applied for analysis. A large portion of the work lies in the research of an optimal basis for the type of signal that will be processed. Performances of different algorithms should be compared, as in benchmark tests.

In the case of speech and music, since quality judgments are greatly influenced by "human-factors", it is also advisable to take into account the opinion of musicians.

Rather than proposing a final technique, this paper has shown the multiplicity of research routes and alternatives in music synthesis utilizing wavelets techniques.

References

Daubechies, I. *Ten Lectures on Wavelets.* CBMS-NSF Regional Conference Series on Applied Mathematics. Philadelphia, SIAM, 1992.

Strang, G.; Nguyen, T. *Wavelets and Filter Banks.* Workshop Notes: San Jose State University, jan (13-16) 1995. Wellesley-Cambridge Press, 1995. /to be published/

Chui, C.K. *An Introduction to Wavelets.* San Diego, Academic Press, 1992.

Meyer, Y. *Wavelets: Algorithms and Applications.* Philadelphia, SIAM, 1993.

Risset, J.C. *The computer, music and sound models.* In: Combes, J.M. et al. Wavelets: time-frequency methods and phase space: proceedings of the International Conference, Marseille, dec.1987. 2ed., Springer-Verlag, p.102-123, 1989.

Jaffe, D.A. *An overview of criteria for evaluating synthesis and processing techniques.* In: XIV Congresso da Sociedade Brasileira de Computação: I Simpósio de Computação e Música:Anais, Caxambu, aug. 1994, p.53-61.

Oppenheim, A.V. *Discrete-time signal processing.* New Jersey, Prentice-Hall, 1989.

Acknowledgments

This paper has been produced in connection to MSc program research works, under support of CAPES.

Chaosynth

Um sistema que utiliza um autômato celular para sintetizar partículas sônicas

Eduardo Reck Miranda

Compositor, Mestre em Tecnologia Musical, Doutor em Música & Inteligência Artificial

Rua da República, 541 Apt. 317, Porto Alegre - RS, Brasil

E-mail: miranda@music.ed.ac.uk

Resumo em inglês: In this paper I introduce *Chaosynth*, a new sound synthesis system which uses a cellular automaton to produce sounds. *Chaosynth* functions by generating a large amount of short sonic events, or particles, in order to form larger, complex sound events. The synthesis technique of *Chaosynth* is inspired by granular synthesis. Most granular synthesis techniques uses stochastic methods to control the formation of the sound events. *Chaosynth,* however, uses a cellular automaton. I begin the paper by introducing the basics of granular synthesis and explain the functioning of the *Chaosynth* technique. I then introduce the basics of cellular automata and present ChaOs: the cellular automata used in *Chaosynth.* I also explain how ChaOs controls the synthesis parameters and how I used parallel computing to accelerate its performance. I conclude the paper with some final remarks and suggestions for further developments. An early version in English of this paper can be found in *Leonardo* Vol. 28, No. 4 (Journal of the International Society for the Arts, Science and Technology, MIT Press). A project report is available in the World Wide Web site of Edinburgh University: http://www.music.ed.ac.uk/pgrecs/eduardo/chaosynth_report/epcc_project.html. **Palavras chaves**: síntese sonora, autômatos celulares e música, modelagem simbólica de circuitos neuronais, computação paralela.

A Síntese Sonora Granular e o sistema *Chaosynth*

A Síntese Sonora Granular (SSG) é uma técnica para síntese de eventos sonoros complexos. O funcionamento da técnica SSG tem como princípio a produção de milhares de minúsculos eventos sonoros simples, ou *partículas sônicas* (por exemplo, partículas de 30 milisegundos cada), que ao todo formam eventos sonoros complexos.

Esta técnica de síntese tem como base a *teoria granular de representação sônica* proposta na década de 40 pelo físico Dennis Garbor (1947). A teoria propõe que os sons de morfologia complexa são compostos por sequências de partículas sônicas menores e mais simples (Figura 6). A teoria granular de representação sônica teve muita repercussão no meio científico; por exemplo, Nobert Wiener, uma das maiores autoridades da teoria da informação, inspirou-se nas idéias de Dennis Garbor para medir o grau de informação de uma mensagem sonora (Wiener, 1964). O compositor Iannis Xenakis foi o primeiro a utilizar, na década de 60, uma teoria de representação granular para fins musicais (Xenakis, 1971). Entretanto, foi somente na década de 80, com a popularização dos computadores de alto desempenho, que as teorias de Dennis Gabor e Iannis Xenakis tiveram a oportunidade de serem postas em prática por compositores de um modo geral. Desde então, várias variantes da técnica SSG têm sido propostas e utilizadas; veja por exemplo (Truax, 1988; Roads, 1991).

O ponto crucial para o bom desempenho de um sistema de SSG é o método utilizado para controlar a produção das partículas sônicas; exemplos: o controle da quatidade de partículas por segundo e o controle da duração de cada partícula. A grande maioria dos sistemas de SSG utilizam métodos estocásticos (isto é, probabilísticos) para esse fim. Em *Chaosynth* eu proponho um método diferente: o método proposto utiliza um autômato celular chamado ChaOs (Miranda, Nelson & Smaill, 1992; Westhead, 1993).

Introdução aos autômatos celulares e seus princípios "musicais"

Os autômatos celulares são modelos matemáticos de sistemas dinâmicos e não-lineares, onde espaço e tempo são expressos por valores discretos e finitos. Um autômato celular (AC) é geralmente representado por um arranjo matemático (de 2 ou 3 dimensões) de variáveis discretas chamadas *células*. Os valores destas células definem o *estado* do AC. Estes valores mudam constantemente, em sincronia com o pulso de um relógio imaginário. A mudança dos valores das células é controlada por uma *função de transição global* (FTG), que determina o valor de uma célula em função dos valores de