

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/pgreecs/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 milissegundos 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 seqüências de partículas sônicas menores e mais simples (Figura 6). A teoria granular de representação sônica teve muita repercussão no meio científico; por exemplo, Nbert 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).

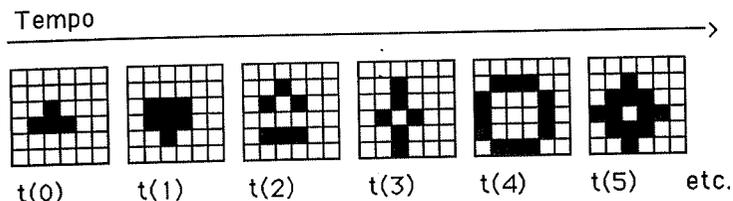
O ponto crucial para o bom desempenho de um sistema de SSG é o método utilizado para controlar a produção das partículas sônicas; exemplos: o controle da quantidade de partículas por segundo e o controle da duração de cada partícula. A grande maioria dos sistemas de SSG utilizam métodos estocásticos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

suas células vizinhas. Para que possamos visualizar o comportamento de uma FTG, o AC necessita ser programado em um computador, de tal maneira que as células sejam dispostas no monitor de vídeo como uma grade de minúsculos retângulos coloridos (ou cubos, se for um AC tridimensional); neste caso, cada cor representa um valor numérico (Wolfram, 1983; Wilson 1988; Dewdney, 1989) (Figura 1).

Figura 1: Uma função de transição global controla os valores das células de um autômato celular. Neste caso as células podem assumir apenas dois valores: 0 (branco) ou 1 (preto).



O primeiro AC foi inventado em meados da década de 60 pelo matemático John von Neumann, com a intenção de simular fenômenos biológicos, como por exemplo a *auto-reprodução*. (Cood, 1968). Ele estava interessado em construir uma máquina (mesmo que abstrata) que pudesse auto-reproduzir-se. O AC que o John von Neumann inventou consistia de uma grade bidimensional de células, cada qual podendo assumir um número finito de valores; cada valor representava um dos componentes da máquina auto-reprodutora. Ele definiu uma FTG cujo funcionamento fazia com que a "máquina" (uma figura na grade, composta de células de vários valores) se estendesse, ocupando porções vazias da grade (isto é, regiões compostas de células de valor neutro), e pouco-a-pouco fosse produzindo cópias dela mesma (isto é, fazendo com que as células neutras assumissem valores iguais aos da "máquina").

Desde então, uma grande variedade de ACs e FTGs têm sido inventadas e adaptadas para os mais diversos tipos de modelagem, nas áreas da física, biologia e computação. Recentemente, os princípios organizacionais dos ACs têm atraído a atenção dos musicólogos. Alguns compositores e pesquisadores estão investigando a possibilidade de utilização dos ACs para controlar, tanto as estruturas formais de alto nível de suas composições (forma musical), como os constituintes de baixo nível do espectro dos sons (espectro sonoro) (Millen, 1990; Hamman, 1991; Orton, Hunt & Kirk, 1991; Miranda, 1993; 1994). Em *Chaosynth* eu utilizei um AC para controlar a síntese dos constituintes espectrais dos eventos sonoros que eu tenho utilizado em várias composições, incluindo a peça *Olive Trees* (que se encontra no CD do II SBC&M).

O autômato celular ChaOs

Um breve comentário sobre a metáfora do ChaOs

ChaOs (uma espécie de acrônimo para a expressão "Chemical Oscillator", ou Oscilador Químico, em português) é um AC bidimensional que pode ser utilizado para modelar alguns fenômenos químicos e biológicos, como por exemplo um fenômeno neurofisiológico conhecido como *circuito neuronal reverberante* (Eccles, 1958; Carpenter, 1990). Neste caso, o ChaOs pode ser imaginado como um circuito composto de milhares de componentes eletrônicos idênticos, chamados de *células nervosas* (como se fosse um tecido nervoso constituído de neurônios interconectados). Em um dado momento, as células nervosas podem assumir uma variedade de valores, representados por cores diferentes. O ChaOs começa com as células nervosas nos mais variados estados, livremente distribuídas na grade. Na medida em que ele avança no tempo, a sua FTG faz com que a distribuição dos valores das células formem ciclos oscilatórios, que no monitor do computador são representados por padrões cíclicos de células coloridas (veja a Figura 5).

Eu associei o comportamento do ChaOs com o ciclo evolutivo dos sons produzidos pela maioria dos instrumentos musicais acústicos: eles geralmente começam com uma distribuição aparentemente desorganizada dos seus parciais (ruído), mas o espectro organiza-se rapidamente em uma certa configuração de parciais que oscilam com regularidade durante a emissão do som. Por exemplo, num som produzido em um violino, o ataque do arco na corda primeiramente produz um ruído; somente quando a corda finalmente começa a vibrar com regularidade é que o som da nota é produzido.

O algoritmo da Função de Transformação Global (FTG)

As células nervosas do ChaOs interagem com as suas células vizinhas através do fluxo de corrente elétrica que circula entre elas. As células nervosas podem estar em um dos três estados: *polarizada*, *despolarizada* ou *queimada*. Para definir o estado de uma célula nervosa, a FTG utiliza dois valores de referência: um valor de limite mínimo (V_{min}) e um valor de limite máximo (V_{max}). Se a voltage interna da célula nervosa (V_i) for menor que o V_{min} , então esta célula está *polarizada*. Se o valor da V_i é maior ou igual que o V_{min} , então a célula está *despolarizada*. Cada célula nervosa possui um *divisor de potência* (ou potenciômetro) cuja função é manter a sua V_i menor que o V_{min} . Entretanto, quando o divisor de potência falha, a célula se despolariza. As células nervosas também possuem um *capacitor elétrico* que regula o seu grau de despolarização; a tendência, no entanto, é de tornar-se cada vez mais despolarizada. Quando a V_i atinge o V_{max} , a célula nervosa "detona" e torna-se *queimada*. Uma célula nervosa queimada é regenerada no próximo pulso do AC (isto é, a FTG automaticamente torna uma célula nervosa queimada (no pulso t) em uma célula polarizada (no pulso $t+1$)).

De um modo geral, o plano de ação da FTG é especificado através dos seguintes parâmetros:

- número n de possíveis valores para as células nervosas
- as resistências $r1$ e $r2$ do divisor de frequência das células nervosas
- a capacitância k do grau de despolarização das células nervosas
- a quantidade de pulsos t do relógio imaginário da FTG
- o tamanho da grade (isto é, o número de células nervosas)

Os estados das células nervosas são definidos por um número entre 0 e $n-1$, sendo que n não deve ser menor do que 3 (isto porque o ChaOs trabalha com células nervosas que podem assumir 3 estados; logo devemos prever pelo menos um valor para cada estado). O valor 0 corresponde ao estado *polarizado*, que é geralmente representado pela cor *branca*. Por outro lado, o valor $n-1$ corresponde ao estado *queimado*, que é geralmente representado pela cor *preta*. Todos os valores entre 0 e $n-1$ correspondem ao estado *despolarizado*. Conseqüentemente, o estado despolarizado pode possuir (quero dizer, quase sempre possui) vários graus de despolarização; quanto maior o valor de n , maior é o grau de despolarização. Várias nuances de cores entre o branco e o preto podem ser utilizadas para representar o estado despolarizado (isso fica a critério do programador).

A FTG do ChaOs é definido por 3 regras, que são aplicadas a todas as células da grade ao mesmo tempo; mas somente uma regra é selecionada para cada célula, dependendo do seu estado (n = uma certa célula da grade e t = pulsos do relógio imaginário):

(a) *se polarizada*: a célula pode ou não tornar-se despolarizada, dependendo do número de células vizinhas (neste caso, as 8 células que circundam a célula) que estão polarizadas (P_{cels}), do número de células vizinhas queimadas (Q_{cels}) e das resistências do divisor de potência ($r1$ e $r2$);

$$\begin{array}{ll} \text{SE} & cel(n)_t = 0 \\ \text{ENTÃO} & cel(n)_{t+1} = \text{int}(P_{cels}(n)/r1)_t + \text{int}(Q_{cels}(n)/r2)_t \end{array}$$

(b) *se despolarizada*: uma célula despolarizada tende a tornar-se mais despolarizada ainda, dependendo da capacitância k e do grau de polarização de sua vizinhança (isto é, das 8 células vizinhas). O grau de despolarização é calculado pela soma dos valores das células vizinhas ($SVal$), dividida pelo número de células vizinhas que ainda se encontram polarizadas (P_{cels});

$$\begin{array}{ll} \text{SE} & 0 < cel(n)_t < n-1 \\ \text{ENTÃO} & cel(n)_{t+1} = k + \text{int}(SVal(n)/P_{cels}(n))_t \end{array}$$

(c) *se queimada*: uma célula nervosa queimada é regenerada no próximo pulso do relógio, isto é, torna-se polarizada;

$$\begin{array}{ll} \text{SE} & cel(n)_t = n-1 \\ \text{ENTÃO} & cel(n)_{t+1} = 0 \end{array}$$

O controle da síntese de partículas sônicas

Embora os princípios funcionais e metafóricos do ChaOs intuitivamente sugerem que ele pode ser utilizado para controlar a produção de partículas sônicas, eu tenho que confessar que eu encontrei sérias dificuldades para elaborar um método de controle que produzisse eventos sonoros interessantes;

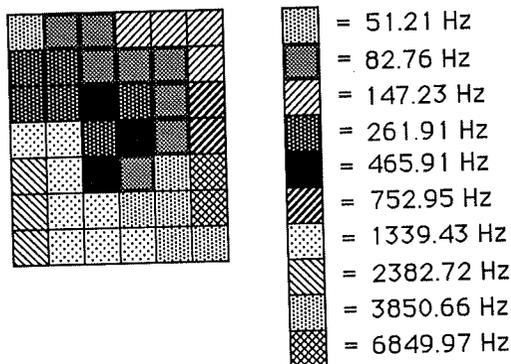
relacionar os valores numéricos produzidos pelo ChaOs com os parâmetros de um instrumento de "síntese granular" foi uma tarefa que demandou muita experimentação. Eu apresento a seguir o método que me pareceu mais razoável e que produziu, em minha opinião, os sons mais interessantes.

O método de controle

O espectro de cada partícula sônica produzida pelo *Chaosynth* é composto de vários parciais, sendo que cada parcial é uma onda senóide produzida por um oscilador (Figura 4). Para produzir uma onda senóide, um oscilador necessita de três parâmetros; frequência (em Hz), amplitude (em dB) e duração (em milissegundos) da onda. O ChaOs se encarrega de controlar os valores das frequências dos parciais (isto é, o conteúdo espectral) e a duração de cada partícula, enquanto que os valores das amplitudes dos parciais (isto é, o envelope espectral) são definidos *a priori* pelo musicista. O sistema *Chaosynth* relaciona os diferentes estados do ChaOs com os valores de frequências pré-estabelecidos pelo musicista (Figura 2).

A grade de células nervosas é dividida em várias sub-grades de mesmo tamanho, sendo que cada sub-grade é responsável pelo funcionamento de um oscilador. A Figura 3 ilustra uma grade de 693 células nervosas, distribuídas entre 9 sub-grades. A cada pulso t do relógio imaginário do ChaOs, o sistema *Chaosynth* produz uma partícula sônica, cujas frequências dos seus parciais são calculadas pela média aritmética das frequências relacionadas com os valores das células nervosas das sub-grades correspondentes aos osciladores.

Figura 2: O sistema relaciona os diversos estados das células nervosas com valores de frequência especificados pelo musicista.



Além das amplitudes dos parciais, o musicista também estabelece *a priori*: (a) o tamanho da grade, (b) a quantidade de sub-grades (ou melhor, a quantidade de osciladores), (c) a distribuição das células nervosas entre os osciladores, (d) o número de possíveis valores para as células nervosas e as frequências relacionadas com estes valores, (e) as resistências $r1$ e $r2$ do divisor de potência do ChaOs, (f) a capacitância k do capacitor que regula o grau de despolarização e (g) o número de pulsos t do relógio da FTG.

Em outras palavras, cada partícula sônica é o resultado de uma síntese aditiva (Dodge & Jerse, 1985) de ondas senóides (Figura 4); na medida em que o ChaOs se desenvolve, os valores de frequência associados aos valores das células nervosas das sub-grades são utilizados para computar os valores das frequências dos osciladores que produzem cada senóide da partícula. A duração de um evento sonoro é determinada pelo número de pulsos do relógio do ChaOs; por exemplo, 100 pulsos produzindo partículas de 30 milissegundos cada, resulta em um evento sonoro de 3 segundos.

Este método de controle é interessante porque ele explora o comportamento do ChaOs para produzir eventos sonoros; esse comportamento pode ser associado ao funcionamento de alguns instrumentos musicais acústicos, como por exemplo o violino e a voz humana. Os sons destes instrumentos começam com uma distribuição livre e (aparentemente) desorganizada de seus parciais. Entretanto, os parciais organizam-se rapidamente em uma configuração espectral "pseudo-regular". O ChaOs começa com as células nervosas, dos mais diversos valores, livremente distribuídas na grade, mas na medida em que ele se desenvolve, as células tendem a organizar-se em formações ondulantes e

oscilatórias (Figura 5); a velocidade do processo de organização é controlada pelas resistências $r1$ e $r2$ e pelo capacitor k .

Figura 3: Uma grade de 693 células nervosas, distribuídas entre 9 sub-grades, sendo que cada sub-grade se relaciona com um oscilador.

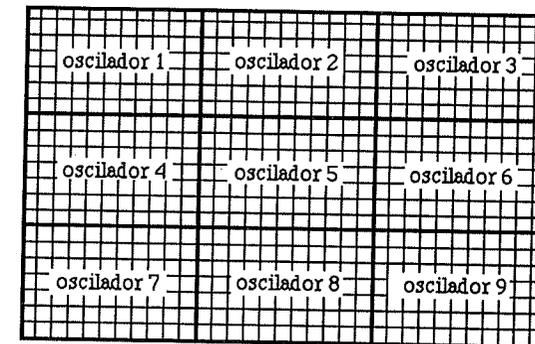
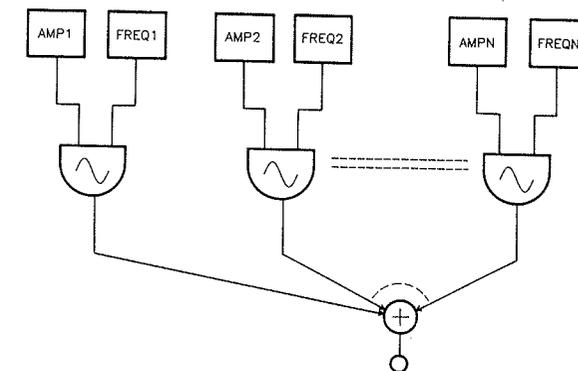


Figura 4: Cada partícula sônica é o resultado de uma síntese aditiva de ondas senóides.



Os sons do *Chaosynth*

Eu tenho sintetizado sons utilizando até 40 valores diferentes para as células nervosas (isto é, 40 valores de frequência diferentes) e uma grade de 25.000.000 (5.000 x 5.000) células nervosas, distribuídas entre 25 osciladores (isto é, cada partícula pode possuir até 25 parciais) (Figura 6).

Os sons resultantes são na sua maioria muito parecidos com os sons de água em movimento; eu costumo apelidar os sons do *Chaosynth* de "sons líquidos e borbulhantes". O *Chaosynth* produz sons "líquidos" e "borbulhantes" nos mais variados graus de densidade, de acordo com a duração individual de cada partícula (por exemplo, 100 a 600 milissegundos cada). Variações na coloração do timbre são controladas pela variação dos valores das frequências associadas aos valores das células nervosas, enquanto que o grau de transição entre ruído e padrões oscilatórios é controlado pelos valores do capacitor k e dos resistores $r1$ e $r2$ do ChaOs.

Maiores informações sobre a relação entre os controles do sistema e os sons que ele sintetiza podem ser obtidas no relatório que está disponível na World Wide Web (veja o endereço no final do resumo em inglês, na primeira página deste artigo).

Figura 5: Na medida em que o ChaOs se desenvolve, as células tendem a se organizar em formações ondulantes e oscilatórias.

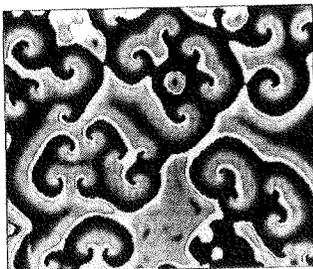
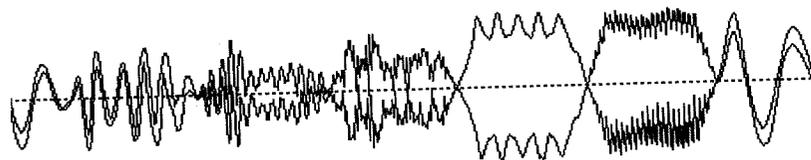


Figura 6: A representação de uma sequência de partículas sônicas produzida por Chaosynth.



A arquitetura do sistema e o uso de computação paralela

A arquitetura do *Chaosynth* está ilustrada na Figura 7. Os sons são sintetizados utilizando o compilador Csound (Vercoe, 1991). O Csound é uma linguagem para síntese sonora no qual o musicista especifica um programa de síntese (isto é, um instrumento digital) em um *arquivo-orquestra* e uma lista numérica de parâmetros de síntese (para o *arquivo-orquestra*) em um *arquivo-partitura*. Quando o compilador do Csound é ativado, estes dois arquivos são processados e o "som" é gerado e arquivado em um *arquivo-sonoro*.

A interface gráfica do *Chaosynth* fornece vários "knobs" e "sliders" para a especificação dos diversos parâmetros do sistema (por exemplo, as frequências associadas aos valores das células nervosas e as amplitudes dos osciladores). Uma vez especificados os parâmetros, o musicista então ativa o ChaOs, que por sua vez produz um *arquivo-partitura* (compatível com o *arquivo-orquestra* do sistema). Finalmente o compilador Csound é ativado e o som então é sintetizado.

A versão atual do sistema (1.0) utiliza computação paralela para tornar a FTG do ChaOs mais rápida. Para paralelizar a FTG eu utilizei a ferramenta PUL-RD, desenvolvida no Centro de Computação Paralela de Edimburgo (EPCC) (PUL-RD significa Parallel Utilities Library - Regular Domain). A ferramenta PUL-RD utiliza uma técnica chamada *decomposição de domínios regulares* (ou "regular domain decomposition", em inglês) para paralelizar programas baseados em autômatos celulares cuja grade pode ser decomposta regularmente (Brown, 1994).

A técnica de decomposição de domínios regulares foi desenvolvida para computar problemas cujos dados podem ser organizados em grades de considerável tamanho. Ela divide a grade em sub-grades regulares e as distribui entre os diversos processadores do computador paralelo para o processamento concorrente. Eu utilizei a PUL-RD para dividir a grade de células nervosas em sub-

grades, de tal forma que cada sub-grade seja relacionada com um oscilador (Figura 8). Neste caso, todos os parciais de cada partícula sônica são computados em paralelo.

Figura 7: A arquitetura do sistema.

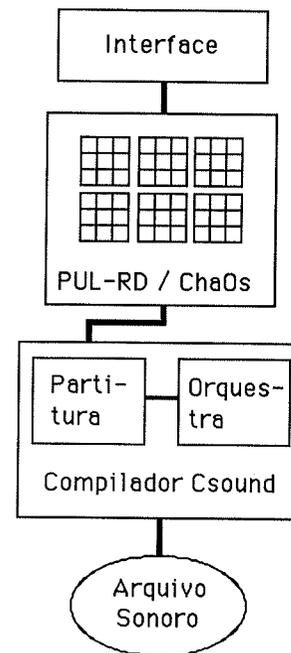
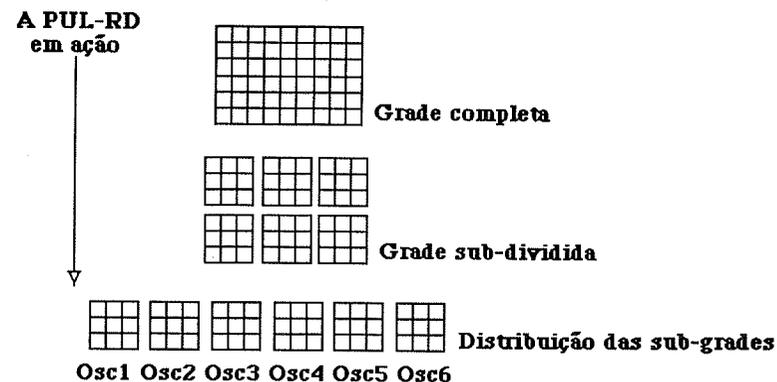


Figura 8: A ferramenta PUL-RD divide a grade em sub-grades regulares e as distribui entre os diversos processadores do computador paralelo.



Conclusão

O *Chaosynth* é um sintetizador cujos parâmetros de síntese são controlados por um autômato celular (AC) chamado ChaOs. A técnica de síntese do *Chaosynth* se assemelha muito com a técnica de síntese granular, pois ela produz uma grande quantidade de partículas sônicas simples, que no total formam eventos sonoros mais complexos. Pelo menos duas características diferem o *Chaosynth* dos sistemas tradicionais de síntese granular: (a) o fato de que eu utilizo o ChaOs para controlar a produção de partículas sônicas e (b) cada partícula é o produto de uma síntese aditiva de várias ondas senóides, enquanto que nos sistemas tradicionais cada partícula é composta por apenas uma senóide.

A versão atual do sistema utiliza apenas um método para mapear o comportamento do ChaOs com os parâmetros de síntese. Em futuras versões eu pretendo deixar esse mapeamento em aberto, de tal forma que o musicista possa inventar e experimentar outros métodos.

No momento eu estou investigando a possibilidade de paralelizar o módulo de síntese sonora, com o objetivo de minimizar ao máximo o tempo necessário para computar as amostragens digitais (ou *samples*) para produção do som. Assim o *Chaosynth* também poderá funcionar como uma espécie de instrumento musical, tocado por intermédio de controladores apropriados; como por exemplo a *luva interativa* que está sendo desenvolvida na Unicamp, pela equipe do Jônatas Manzolli, cujos resultados de pesquisa também estão sendo apresentados no II SBC&M. A longo prazo, eu pretendo unir o *Chaosynth* ao sistema CAMUS (também de minha autoria). O sistema CAMUS utiliza os autômatos celulares para controlar níveis mais altos de composição; por exemplo, o nível formal da composição de fragmentos musicais (Miranda, 1993; 1994).

O *Chaosynth* foi implementado em um supercomputador CM-200, no Centro de Computação Paralela de Edimburgo (EPCC), na Escócia, com o auxílio técnico de Martin Westhead and Robert Fletcher.

Referências Bibliográficas

- Carpenter, R.H.S. (1990), *Neurophysiology*, Edward Arnold, GB.
- Cood, E.F. (1968), *Cellular Automata*, Academic Press, GB.
- Brown, S. (1994), *Parallellising Grid-Based Applications with PUL-RD*, relatório técnico, Edinburgh Parallel Computing Centre, GB.
- Dewdney, A.K. (1989), "A cellular universe of debris, droplets, defects and demons", *Scientific American*, August:88-91.
- Dodge, C. and Jerse, T. (1985), *Computer Music - Synthesis, Composition and Performance*, Schirmer Books, UK.
- Eccles, J.C. (1958), *The Physiology of Imagination*, WH Freeman & Co (reimpressão de artigo publicado na revista *Scientific American*), EUA.
- Garbor, D. (1947), "Acoustical Quanta and the Theory of Hearing", *Nature*, 159(1044):591-594.
- Hamman, M. (1991), "Mapping Complex Systems using Granular Synthesis", *Montreal ICMC 1991 Proceedings*, 475-478.
- Millen, D. (1990), "Cellular Automata Music", *Glasgow ICMC Proceedings*, 314-316.
- Miranda, E.R., Nelson, P. and Smaill, A. (1992), "ChaOs: a Model for Granular Synthesis using Cellular Automata", *Annual Report & Project Directory 1991-92*, Edinburgh Parallel Computing Centre, 153-156, GB.
- Miranda, E.R. (1993), "Cellular Automata Music: An Interdisciplinary Project", *Interface*, 22(1):3-21.
- Miranda, E.R. (1994), "Cellular Automata Music Composition", *Languages of Design*, 1994(2):105-107.
- Orton, R., Hunt, A., and Kirk, R. (1991), "Graphical Control of Granular Synthesis using a Cellular Automata and the Freehand Program", *Montreal ICMC 1991 Proceedings*, 416-418.
- Roads, C. (1991), "Asynchronous Granular Synthesis", *Representation of Music Signals*, De Poly, G. et al (editores), MIT Press.
- Truax, B. (1988), "Real Time Granular Synthesis with a DSP Computer", *Computer Music Journal*, 12(2):14-26.
- Vercoe, B. (1991), *Csound Manual*, disponível via Internet ftp no diretório *music* da máquina *media-lab.mit.edu*.
- Westhead, M.D. *Chaosynth: a System for Granular Synthesis using a Cellular Automaton*, EPCC Summer Course Project Report, Edinburgh University, GB.
- Wiener, N. (1964), "Spatial-Temporal Continuity, Quantum Theory and Music", *The Concepts of Space and Time*, Capek, M. (editor), Reidel.
- Wilson, G. (1988), "The life and times of cellular automata", *News Scientist*, October(8):44-47.
- Wolfman, S. (1983), "Statistical mechanics of cellular automata", *Reviews of Modern Physics*, 55(3):601-623.

One Tool, Two Programs and Several Ideas for Composition with Spectral-Modelling Synthesis

CELSO AGUIAR

Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University
Stanford, California 94305-8180 USA
aguiar@ccrma.stanford.edu

Abstract

Sound can be approached from complementary perspectives. This paper describes aspects of one of my works, 'Piece of Mind' for tape, where time domain techniques derived from 'musique concrète' are blended to powerful spectral modelling (SMS) in the processing of natural sounds. Programs written in collaboration with Xavier Serra have made SMS available in CLM's synthesis environment. Deterministic components are synthesized by a highly efficient IFFT algorithm that relieves the delay of pleasure composers have historically accumulated with additive synthesis. Bell sounds and singing voice are hybridized or cross-faded to provoke ambiguity of identity. On the other hand, the way SMS handles its stochastic part exposes the fact that dsp guys and composers may not always speak the same language. Being aware of what synthesis techniques are geared to and the ability to present proper solutions are first requirements. A graphical interface to display SMS information on a single view is introduced.

SOME PREMISES

A prediction was once made by researcher Julius O. Smith III where he affirmed that time domain techniques for sound synthesis and transformation would migrate, or be absorbed in the future, into frequency domain ones (Smith, 1991). By time domain techniques he meant all synthesis techniques derived from 'musique concrète' like sampling and granular synthesis. Under frequency domain he listed several techniques including spectral modelling synthesis (SMS) and inverse Fourier synthesis.

His prediction was made about four years ago and considered the fact that sampling and granular techniques can add very interesting new colors to the sonic palette, but they are very difficult to control. To obtain control over these techniques, more general sound transformations would be required. As these transformations are to be understood in terms of what we hear, and the best way to understand what we hear is through the short-time spectrum it seems logical to assume that time domain techniques point towards spectral modelling.

The word 'control' and its meaning in his text may stir polemic if directly transposed to a compositional context. Composers, including myself, may want to deal with the uncontrollable, and that stands very far from avoiding control over materials. However, I still think his idea is interesting enough to deserve some discussion and consideration. In one side we have time domain techniques that are straight-forward to implement but which impose barriers to effectively manipulate sounds. On the other hand we have very powerful analysis-synthesis techniques which allow us to modify sound in very effective ways but whose implementation is not as immediate as in the previous case and which involve much more computation than the mere reading of files or of samples of recorded samples.

For the composer the amount of delayed pleasure associated with additive synthesis, for example, is such