

Fernando Falci de Souza

**Síntese Granular em Tempo Real no
Espaço de Gabor Estendido**

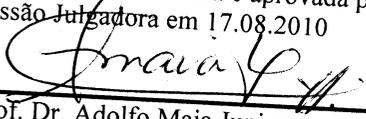
Campinas

2010

Fernando Falci de Souza

**Síntese Granular em Tempo Real no
Espaço de Gabor Estendido**

Este exemplar é a redação final da Dissertação defendida
pelo Sr Fernando Falci de Souza e aprovada pela
Comissão Julgadora em 17.08.2010


Prof. Dr. Adolfo Maia Junior
Orientador

Dissertação apresentada ao Programa de
Pós-Graduação em Música do Instituto de
Artes da Universidade Estadual de
Campinas – Área de concentração em
Fundamentos Teóricos, para obtenção do
título de Mestre em Música.

Orientador: Prof. Dr. Adolfo Maia Jr.

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE ARTES
PROGRAMA DE PÓS-GRADUAÇÃO EM MÚSICA

Campinas

2010

UNIVERSIDADE DE CAMPINAS
Nº DE REGISTRO 89098
Série 89
Ligação 10-134-10
Cód. de Acesso 1100
DATA 07/02/10
CÓDIGO T 117957

Acompanha 1 CD-ROM

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO INSTITUTO DE ARTES DA UNICAMP

So89s Souza, Fernando Falci de.
Síntese Granular em Tempo Real no Espaço de Gabor
Estendido. / Fernando Falci de Souza. – Campinas, SP: [s.n.],
2010.

Orientador: Prof. Dr. Adolfo Maia Júnior.
Dissertação(mestrado) - Universidade Estadual de Campinas,
Instituto de Artes.

1. Síntese Granular. 2. Algoritmos Genéticos. 3. Gestos.
4. Música por computador. I. Maia Júnior, Adolfo.
II. Universidade Estadual de Campinas. Instituto de Artes.
III. Título.

(em/ia)

Título em inglês: "Real Time Granular Synthesis in Extended Gabor Space."
Palavras-chave em inglês (Keywords): Granular Synthesis ; Genetic
Algorithm ; Gesture ; Computer music.
Titulação: Mestre em Música.
Banca examinadora:
Prof. Dr. Adolfo Maia Júnior.
Profª Drª. Denise Hortência Lopes Garcia.
Prof. Dr. Claudiney Rodrigues Carrasco.
Prof. Dr. Furio Damiani.
Prof. Dr. Samuel Rocha de Oliveira.
Data da Defesa: 17-08-2010
Programa de Pós-Graduação: Música.

Errata:
Área de Concentração: Fundamentos Teóricos

Emerson Luiz de Biaggi
Prof. Dr. Emerson Luiz de Biaggi
Coordenador dos Cursos de Pós-Graduação
IA/UNICAMP
Matr. 27628-6

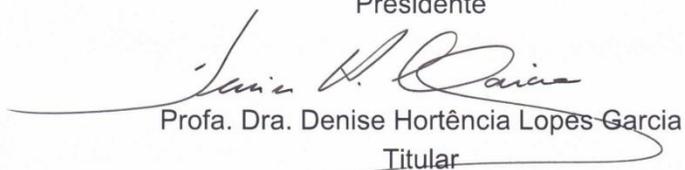
2010010729

Instituto de Artes
Comissão de Pós-Graduação

Defesa de Dissertação de Mestrado em Música, apresentada pelo Mestrando Fernando Falci de Souza - RA 79043 como parte dos requisitos para a obtenção do título de Mestre, perante a Banca Examinadora:



Prof. Dr. Adolfo Mala Junior
Presidente



Profa. Dra. Denise Hortência Lopes Garcia
Titular



Prof. Dr. Furio Damiani
Titular

Agradecimentos

Este trabalho, além de alimentar minha paixão pela ciência e pela música, merece especial importância, pois sua conclusão concretiza um novo caminho em minha trajetória pessoal. Durante quase uma década a música e a computação fizeram parte de minha vida como atividades opostas, mas neste trabalho de pesquisa as ciências da computação são colocadas a favor da música. Tornar-me mestre em música computacional e estar apto a exercer a nobre profissão de professor e pesquisador é o ponto de partida de uma nova carreira e um momento de valor inestimável.

Neste trabalho de grande valor, agradeço primeiramente ao meu orientador Prof. Adolfo Maia Júnior. Durante estes anos, em encontros semanais, além de ser mestre atencioso e inspirador, se tornou um grande amigo.

Agradeço ao Prof. Marcelo M. Wanderley pela orientação durante o estágio realizado no Canadá e, evidentemente, por ter criado esta incrível e inesperada oportunidade de morar fora do Brasil. Agradeço ao Programa de Líderes Emergentes nas Américas (ELAP) do governo canadense pelo qual este estágio se realizou, à *McGill University* e ao pessoal do *Input Devices for Music Interaction Lab* (IDMIL).

Agradeço também ao Prof. Jônatas Manzolli que, além de me apresentar aos professores Adolfo Maia Jr. e Marcelo M. Wanderley, foi quem me inspirou a buscar a carreira acadêmica em música computacional ao apresentar uma palestra a qual eu estava presente antes mesmo de iniciar meus estudos em música.

Agradeço aos Professores Claudiney Carrasco e Denise Garcia pela contribuição que trouxeram a esta pesquisa ao participarem do exame de qualificação, aos professores das disciplinas cursadas durante o programa de mestrado e à equipe do Núcleo Interdisciplinar de Comunicação Sonora (NICS-UNICAMP).

Finalmente, sou muito grato à minha família e aos meus queridos amigos por me incentivarem nesta escolha de vida e também simplesmente por estarmos juntos nesta jornada que excede nossa compreensão.

Resumo

Esta dissertação trata principalmente da pesquisa de métodos matemáticos e algoritmos com a finalidade de gerar e controlar fluxos de som em tempo real dentro do escopo da síntese granular. Inicialmente abordamos os fundamentos matemáticos da Síntese Granular proposta pelo Prêmio Nobel de Física Dennis Gabor. Em seguida apresentamos uma breve revisão histórica do uso desta técnica em composições eletroacústicas e analisamos alguns dos mais conhecidos aplicativos atuais para síntese granular. Com base nestes estudos desenvolvemos um *software* para síntese granular em tempo real que denominamos EVOGrain. Este aplicativo apresenta uma interface gráfica intuitiva e amigável com a qual o compositor controla a síntese granular desenhando, com uso do mouse, retângulos de diferentes tamanhos e posição. As coordenadas destes retângulos fornecem informação que guia a execução de um algoritmo genético, que por sua vez controla os parâmetros de um módulo de síntese granular em tempo real resultando em sonoridades e texturas que evoluem dinamicamente.

Na segunda parte da dissertação apresentamos os resultados da nossa pesquisa realizada no laboratório Input Devices for Music Interaction Lab (IDMIL) da McGill University através de um estágio pelo programa de intercâmbio canadense Emerging Leaders from the Americas Program (ELAP). Nesta pesquisa, foram feitos diversos experimentos com os chamados Instrumentos Musicais Digitais (DMIs). Os módulos do sistema original EVOGrain foram reformulados em três sistemas independentes, GranularStreamer, Genetic Algorithm Mapper e GranularDrawer, responsáveis respectivamente pela execução da síntese sonora, aplicação de algoritmo genético e síntese de imagens. Além deles, diversos controladores gestuais foram avaliados no contexto da síntese granular, sendo eles os comerciais PC-1600x, SpaceNavigator e o T-Stick desenvolvido no IDMIL, o que culminou na prototipagem de um novo dispositivo, de nossa autoria e que denominamos RedController.

Palavras chave: Síntese Granular; Algoritmos Genéticos; Controladores Gestuais; Música Eletroacústica.

Abstract

This dissertation deals mostly with the application of mathematical methods and algorithms in order to generate and control streams of sound in real time within the scope of granular synthesis. Initially we studied the mathematical foundations of Granular Synthesis proposed by the Nobel Prize in Physics, Dennis Gabor. Next, we presented a brief historical review of the use of this technique in electroacoustic compositions and analyzed some of the most popular software of granular synthesis available today. We developed a new software for real-time granular synthesis we named EVOGrain. This application provides an intuitive and friendly graphical interface where the composer controls the granular sound stream by drawing rectangles of different sizes and positions on a window interface using the mouse. The coordinates of these rectangles provide information that guide the resulting process of a running genetic algorithm, which in turn controls the parameters of a granular synthesis engine in real time resulting in sound streams and textures that evolve dynamically.

The second part of this dissertation presents the results of our research in the Input Devices for Music Interaction Laboratory (IDMIL) of McGill University through the Canadian Fellowship Emerging Leaders from the Americas Program (ELAP). In this research, several experiments with so-called Digital Musical Instruments (DMIs) took place. The modules of the original system EVOGrain were reformulated into three independent systems, GranularStreamer, Genetic Algorithm Mapper and GranularDrawer, responsible respectively for the synthesis of granular sounds, application of genetic algorithm and image synthesis. In addition to this diversification of software, some gestural controllers were evaluated in the context of our granular synthesis model, the PC-1600X, SpaceNavigator (available commercially) and the T-Stick (developed at the IDMIL), culminating in the prototyping of a new device designed by us and named RedController.

Key Words: Granular Synthesis; Genetic Algorithm; Gestural Controllers; Electroacoustic Music.

Sumário

| | |
|--|------------|
| 1 INTRODUÇÃO | 1 |
| 1.1 PUBLICAÇÕES..... | 3 |
| 1.2 ESTRUTURA DA DISSERTAÇÃO..... | 3 |
| 1.3 CD-ROM ANEXO..... | 4 |
| 2 SÍNTESE GRANULAR | 5 |
| 2.1 A TEORIA DE GABOR..... | 7 |
| 2.2 SÍNTESE GRANULAR ANALÓGICA..... | 12 |
| 2.3 SÍNTESE GRANULAR DIGITAL..... | 13 |
| 2.4 COM PONDO COM GRÃOS..... | 14 |
| 2.5 ESTADO DA ARTE EM SÍNTESE GRANULAR..... | 20 |
| 2.5 MOTIVAÇÃO..... | 25 |
| 3 O SISTEMA EVOGRAIN | 28 |
| 3.1 MÁQUINA DE SÍNTESE GRANULAR EM TEMPO REAL..... | 29 |
| 3.2 ALGORITMOS GENÉTICOS..... | 33 |
| 3.3 INTERFACE INTUITIVA E AMIGÁVEL..... | 39 |
| 3.4 INTEGRAÇÃO COM OUTROS SISTEMAS ATRAVÉS DE MENSAGENS OSC..... | 42 |
| 3.5 ARQUITETURA FINAL DO SISTEMA..... | 44 |
| 4 SISTEMAS DERIVADOS DO EVOGRAIN | 46 |
| 4.1 GRANULARSTREAMER..... | 47 |
| 4.2 GENETIC ALGORITHM MAPPER..... | 51 |
| 4.3 GRANULARDRAWER..... | 56 |
| 5 CONTROLADORES GESTUAIS E INSTRUMENTOS MUSICAIS DIGITAIS | 60 |
| 5.1 EXPERIMENTOS COM SPACENAVIGATOR..... | 63 |
| 5.2 EXPERIMENTOS COM T-STICK..... | 65 |
| 5.3 EXPERIMENTOS COM PC-1600X..... | 67 |
| 5.4 O PROTÓTIPO REDCONTROLLER..... | 68 |
| 6 RESULTADOS | 73 |
| 6.1 RESULTADOS OBTIDOS COM O SISTEMA EVOGRAIN..... | 73 |
| 6.2 RESULTADOS OBTIDOS COM OS SISTEMAS DERIVADOS DO EVOGRAIN..... | 78 |
| 7 CONSIDERAÇÕES FINAIS E PERSPECTIVAS | 86 |
| REFERÊNCIAS | 91 |
| ANEXO A : ARTIGO PUBLICADO NO 7º CONGRESSO DA AES BRASIL 2009 | 94 |
| ANEXO B : ARTIGO PUBLICADO NO 8º CONGRESSO DA AES BRASIL 2010 | 101 |

1 INTRODUÇÃO

Esta dissertação trata principalmente da pesquisa de métodos matemáticos e algoritmos com a finalidade de gerar e controlar fluxos de som (pistas) em tempo real dentro do escopo da síntese granular. Inicialmente abordamos os fundamentos matemáticos da Síntese Granular proposta pelo premio Nobel de Física Dennis Gabor. Três artigos deste descrevem tais fundamentos matemáticos (GABOR,1946,1947,1952). Além de tratar de uma técnica que manipula o material sonoro em seu limite micro-temporal, abordamos também o problema de classificação, ou taxonomia, destes sons através da Espectromorfologia proposta por Dennis Smalley (SMALLEY, 1997) e sobre a física e psicofísica do som (ROEDERER, 2002) neste contexto. Grandes avanços foram feitos no campo da síntese granular na década de 90 sendo os maiores responsáveis por tais avanços os compositores Curtis Roads e Barry Truax. Em seu livro **Microsound** (ROADS, 2001), que é um clássico da síntese granular, Roads resume quase tudo que foi feito nesta área até 2001. Estudamos em detalhe este livro e diversos de seus exemplos de síntese granular foram implementados. Neste livro também Roads fornece outras classificações dos sons granulares.

Com base nestes estudos foi possível iniciar o desenvolvimento de um novo método matemático e alguns algoritmos para síntese granular em tempo real disponibilizando também uma interface amigável, isto é, uma interface de fácil compreensão por qualquer pessoa. Devido ao nosso interesse em portabilidade, os programas computacionais derivados da análise teórica foram escritos na linguagem JAVA, sendo que a primeira abordagem buscou utilizar os recursos já existentes da biblioteca *jMusic*. Por ser uma biblioteca que visa disponibilizar recursos para manipulação de música em suas mais diversas formas, suas funcionalidades acompanham um excesso de estruturas, classes e funções. Por um lado este excesso permite a aplicação mais geral da *jMusic* no desenvolvimento de sistemas sonoros e musicais, mas por outro fica prejudicada a desempenho do software, em especial quando o objetivo é a manipulação de parâmetros sonoros em tempo real. Com o objetivo de melhorar o desempenho do algoritmo, foi experimentada uma segunda abordagem onde retiramos deste todas as estruturas e lógicas

desnecessárias à síntese granular. Assim a biblioteca jMusic foi substituída por uma biblioteca dedicada mais “enxuta” e de autoria própria.

Como mencionamos acima, foi elaborada uma interface gráfica intuitiva e amigável onde o compositor controla a síntese granular desenhando, com uso do mouse, retângulos de diferentes tamanhos e posição, cujas coordenadas fornecem informação (parâmetros de entrada ou input) que guia a evolução temporal de um fluxo de sons granulares. Esta interface controla a execução de um algoritmo genético que permite que uma família de grãos de som evolua continuamente em direção a uma segunda família a qual denominamos “alvo”. Com isso obtemos texturas e timbres naturais evoluindo dinamicamente em tempo real, com baixa latência, superando a dificuldade de alguns métodos de síntese em apresentar uma evolução do timbre satisfatória em tempo real. Este sistema foi denominado **EVOGrain**.

Após quase dois anos realizando a pesquisa e desenvolvendo o sistema **EVOGrain** surgiu uma oportunidade única e de grande valor de realizar um estágio pelo programa de intercâmbio canadense chamado *Emerging Leaders from the Americas Program* (ELAP). Neste estágio, realizado no *Input Devices for Music Interaction Lab* (IDMIL) na *McGill University*, abordamos uma nova vertente da nossa pesquisa com a experimentação de Instrumentos Musicais Digitais (DMIs), cujos resultados incorporamos a esta dissertação. Os módulos do sistema original **EVOGrain** foram separados em três sistemas independentes, **GranularStreamer**, **Genetic Algorithm Mapper** e **GranularDrawer**, responsáveis respectivamente pela execução da síntese sonora, aplicação de algoritmo genético e síntese de imagens. Além desta diversificação de *software*, diversos controladores gestuais foram avaliados no contexto da síntese granular, sendo eles os comerciais **PC-1600x**, **SpaceNavigator** e o **T-Stick** desenvolvido no **IDMIL**, o que culminou na prototipagem de um novo dispositivo, de nossa autoria e que denominamos **RedController**.

1.1 Publicações

Durante o processo desta pesquisa foram elaborados e publicados dois artigos nos anais do Congresso da Sociedade de Engenharia de Áudio (AES) em 2009 e 2010. O primeiro “*Evolutive Processes for Granular Synthesis in Extended Gabor Spaces*” contém uma breve revisão histórica da síntese granular e do uso de algoritmos genéticos em música e descreve fundamentalmente a implementação do sistema **EVOGrain**. O segundo, “*An Input Device To Control Granular Sound Processes*”, propõe o uso de DMIs para iniciar processos sonoros granulares com contornos específicos descritos no artigo *Spectromorphology* (SMALLEY, 1997). Estes dois trabalhos estão nos anexos A e B desta dissertação.

1.2 Estrutura da Dissertação

Iniciaremos o segundo capítulo com a apresentação da teoria matemática que fundamenta a síntese granular, seguida de um breve histórico deste interessante método de composição e de seu uso na música eletroacústica antes e depois do advento da tecnologia digital. Prosseguiremos identificando alguns elementos musicais presentes em peças de Curtis Roads e Barry Truax. Apresentaremos também uma análise de sistemas para síntese granular de uso livre (*freeware*) e finalmente discorreremos sobre a motivação da nossa pesquisa frente ao estado da arte da Síntese Granular.

No capítulo 3 apresentaremos o fundamento teórico da técnica de programação conhecida como Algoritmos Genéticos. Em seguida, descreveremos os detalhes da implementação da segunda versão do sistema **EVOGrain**, dividida em três módulos principais: a máquina de síntese granular em tempo real, o algoritmo genético e a interface. Serão apresentadas instruções de uso e documentação de como integrar o **EVOGrain** com outros sistemas através de mensagens Open Sound Control (OSC).

Os capítulos 4 e 5 são dedicados à descrição das atividades realizadas no estágio na *McGill University*, sendo que o quarto capítulo apresentará os sistemas criados a partir dos módulos do EVOGrain, o **GranularStreamer** para síntese de granular, o **G. A. Mapper**

para o algoritmo genético e o **Granular Drawer** para a interface. No quinto capítulo descreveremos os experimentos realizados com alguns instrumentos digitais disponíveis no IDMIL como o **PC-1600X**, um painel de *sliders*, o **SpaceNavigator**, um super mouse 3D, e o **T-Stick**, uma interface alternativa desenvolvida por Joseph Malloch no próprio IDMIL (MALLOCH, 2008). Apresentaremos, por fim, um novo controlador gestual alternativo que denominamos **RedController**, desenvolvido durante nosso estágio no IDMIL sob a supervisão do Prof. Dr. Marcelo M. Wanderley.

No capítulo 6, analisaremos os resultados sonoros obtidos com o uso dos diversos sistemas desenvolvidos dentro da perspectiva da Espectromorfologia (SMALLEY, 1997) e algumas considerações poéticas. Finalmente, no capítulo 7 apresentaremos uma conclusão sobre os temas abordados nesta dissertação e algumas perspectivas para pesquisa futura.

1.3 CD-ROM Anexo

Esta dissertação é acompanhada de um CD-ROM cujo conteúdo está dividido em quatro pastas. A pasta *Artigos* contém arquivos PDF dos dois artigos publicados no decorrer desta pesquisa, a pasta *Dissertação* contém um arquivo PDF com o texto integral desta dissertação, a pasta *Exemplos* contém doze arquivos MP3 com exemplos sonoros referenciados no capítulo 6 desta dissertação, e finalmente, a pasta *Sistemas* contém três outras pastas onde são encontradas versões executáveis dos aplicativos **EVOGrain**, **GranularStreamer** e **G.A.Mapper**.

2 SÍNTESE GRANULAR

O som é uma onda mecânica que se propaga longitudinalmente em um meio material elástico através de compressões e expansões deste meio que se alternam. Esta propagação de uma frente de compressão carrega uma forma de energia chamada de Energia Elástica e quando atinge a fronteira com outro meio de propagação a energia é parcialmente refletida e parcialmente absorvida. Quando a superfície atingida é um tímpano humano, a energia absorvida continua se propagando pelo ouvido médio e pelo ouvido interno até atingir as células ciliares da membrana basilar onde, se a energia estiver acima de um limiar de ativação, é convertida em impulsos eletroquímicos que se propagam pelos neurônios do sistema nervoso central (ROEDERER, 2002). Este fenômeno psicoacústico é a percepção do som.

O processo de produzir artificialmente as rápidas sucessões de compressões e expansões de uma onda sonora teve início com técnicas mecânicas. As diversas técnicas de produção sonora sejam elas mecânico-acústicas (grupos de instrumentos ou mesmo uma orquestra), eletromecânicas (antigo órgão Hammond), eletrônicas analógicas (primeiros sintetizadores) ou ainda eletrônicas digitais, sempre estiveram ligadas aos avanços tecnológicos dos meios de emissão acústica. O som como seqüência de compressões e expansões pode ser convertido num sinal elétrico periódico $s(t)$ como ocorre, por exemplo, em um microfone. Inversamente, o sinal $s(t)$ pode ser convertido num som. O exemplo mais simples é um sintetizador primitivo composto por um oscilador que um sinal elétrico cuja intensidade varia como uma onda senoidal com amplitude e freqüência determinada, este sinal é então convertida em uma onda acústica através de um alto-falante (ROADS, 2001).

Em 1963, Max Mathews publicou um artigo histórico no qual apresenta o computador digital como um instrumento musical para se obter “sons a partir de números” (SMITH III, 2002). A partir daí, a busca por representações digitais do sinal sonoro passou a ocupar uma posição muito importante, dada a possibilidade de implementação de algoritmos de síntese no computador. Neste cenário, o sinal elétrico $s(t)$ tem uma representação discreta (digital) dada por uma seqüência de valores da amplitude de $s(t)$

amostrados em intervalos de tempo regulares. Quando a amostragem é realizada em intervalos suficientemente pequenos (a cada 1,36 milissegundos, ou a 44.100 vezes por segundo) é possível representar toda a faixa de frequências que está dentro do limites da audição humana (de 20 Hz a 22.050 Hz) (MOORE, 1990).

Diversos métodos de síntese sonora digital foram desenvolvidos ao longo de décadas de pesquisa: síntese aditiva, síntese subtrativa, síntese FM, modelagem física e síntese baseada em amostras (*sampling*). Um dos métodos mais conhecidos é a *Síntese Aditiva de Fourier* no qual o sinal $s(t)$ é representado por uma somatória de funções periódicas do tipo seno e cosseno, conhecidas também como *Parciais Harmônicas* ou simplesmente, *Parciais de Fourier*.

Apesar do método de Síntese Aditiva de Fourier obter excelentes resultados e estar bem estabelecido como técnica de síntese sonora, o mesmo apresenta algumas dificuldades de representação de sons complexos e naturais (ROADS, 2001), principalmente daqueles cujo transiente é muito rápido. Para este tipo de síntese precisamos, primeiramente, considerar o fato que nenhuma fonte sonora real é capaz de produzir o som de uma única onda senoidal. Além disso, no que se refere à duração e ao comportamento da envoltória de volume de um som, a Síntese Aditiva de Fourier decompõe o som em senóides que por hipótese, duram infinitamente, enquanto que os sons produzidos por fontes reais apresentam envoltórias (ou curvas de amplitude) finitas, no sentido que começam com valor zero, num certo intervalo de tempo são não nulas, e depois voltam ao valor zero. O modelo mais simples para este tipo de comportamento é o chamado modelo ADSR - *Attack, Decay, Sustain e Release*, formada de segmentos de reta (funções lineares), como na Figura 1.

Métodos tradicionais de síntese focados no domínio da frequência (em contraposição ao domínio do tempo) costumam apresentar limitações na reprodução deste comportamento de envoltória temporal, em especial nos períodos de ataque e decaimento onde o sinal sonoro apresenta uma intrincada variação de energia entre suas componentes espectrais, como já mencionamos no caso da Síntese de Fourier. A geração de som pelo método de síntese granular oferece uma alternativa aos métodos de síntese tradicionais, pois, ao decompor sons longos e complexos em uma grande quantidade de “pequenos

sons”, “quanta de sons” na definição de Gabor (GABOR, 1947), cuja duração é da ordem de milissegundos, dá maior ênfase à dimensão do tempo ao invés da dimensão da frequência, como veremos a seguir na exposição da Teoria de Gabor sobre Síntese Granular.

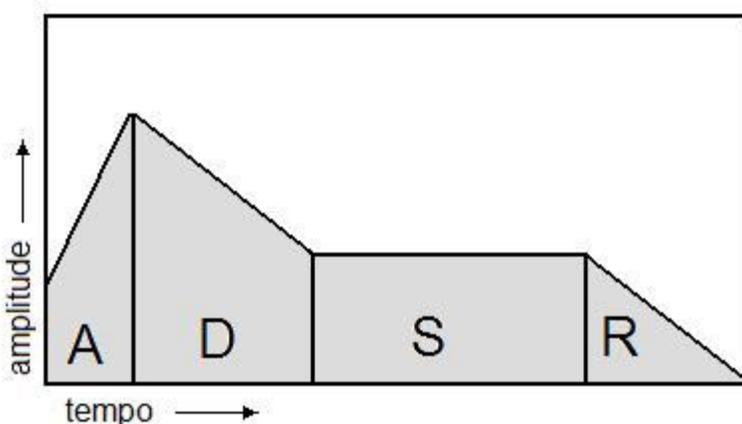


Figura 1 Modelo ADSR de envoltória de volume.

2.1 A Teoria de Gabor

O som é um fenômeno ondulatório, com este modelo ondulatório podem-se explicar, de maneira simples, vários fenômenos sonoros como interferência, reflexão e outros. O que não é bem conhecido é que o som também pode ser interpretado como um conjunto de “partículas elementares” de energia sonora. Em 1947, Dennis Gabor publicou um artigo onde introduz, matematicamente, ou formalmente, o conceito do *Quantum Acústico*, definido como a “menor partícula de som” que contém energia suficiente para ser audível, isto é, capaz de ser detectada pelo aparelho sensorial acústico humano (GABOR, 1947). A consequência desta abordagem é que um som longo e complexo poderia então ser decomposto em uma grande quantidade de “quanta” de som de curta duração (da ordem de milésimos de segundos, para o ouvido humano). Mais tarde Roads denomina estes quanta de “partículas gaussianas” (ROADS, 2001), pois apresentam uma envoltória na forma de uma Curva de Gauss.

Gabor reconheceu que a representação do sinal sonoro como uma função $s(t)$ leva a uma interpretação enganosa do fenômeno acústico, quando na verdade, nenhum sinal sonoro pode ser detectado em um intervalo de tempo nulo. Para se escutar um sinal sonoro é necessário observá-lo por certo período de tempo, o ouvido humano como detector não é ideal, ou seja, não consegue distinguir todas as frequências que compõem o som detectado, a não ser, idealmente, num intervalo de tempo infinito, como mostra GABOR (1947). A idéia aqui é que qualquer “análise por janelamento” implica certo compromisso entre a resolução do sinal analisado (ou percebido pelo ouvido) em tempo e frequência. Por exemplo, se queremos distinguir bem (com pequeno erro) a frequência de um som, teremos que ouvi-lo com um tempo “razoavelmente longo”, ou nas palavras de Roads: “alta resolução em frequência requer a análise de um número grande de amostras. Isto implica uma janela de longo tempo” (ROADS, 2001). Inversamente podemos ouvir (ou detectar) sons de muito curta duração (da ordem de milissegundos), mas o preço a ser pago por isto é a perda de resolução na identificação das frequências que compõem o sinal sonoro. Ouvimo-los como breves *tics* de ruído.

Gabor traduz isto na linguagem da Teoria da Comunicação: “Considerando que queremos classificar as frequências dentro de uma largura de banda W , não podemos medir intervalos de tempo menores $1/W$ ” (GABOR, 1947). Gabor propõe então uma representação do som que considere simultaneamente a função *Densidade de Energia* associada ao sinal sonoro em função do tempo $s(t)$ e a função *Densidade de Energia* associada à representação espectral $S(f)$ (Transformada de Fourier) do sinal $s(t)$. Matematicamente isto pode ser formalizado da seguinte maneira: se denotamos Δt como o intervalo de tempo da análise de um som, e Δf como a largura da banda na análise de frequências, então vale a seguinte desigualdade:

$$\Delta t \times \Delta f \geq 1 \quad (1)$$

Esta desigualdade é bastante similar à famosa Relação de Incerteza de Heisenberg da Mecânica Quântica, razão pela qual Gabor denomina estes sinais elementares de “*quanta acústico*”, ou ainda “*quanta de informação*”. Assim, Gabor combinou tempo e

freqüência dispondo-os, numa representação cartesiana, como eixos ortogonais (tempo na vertical e freqüência na horizontal), conforme a Figura 2. Este espaço bidimensional é conhecido usualmente como Espaço Tempo-Freqüência de Gabor.

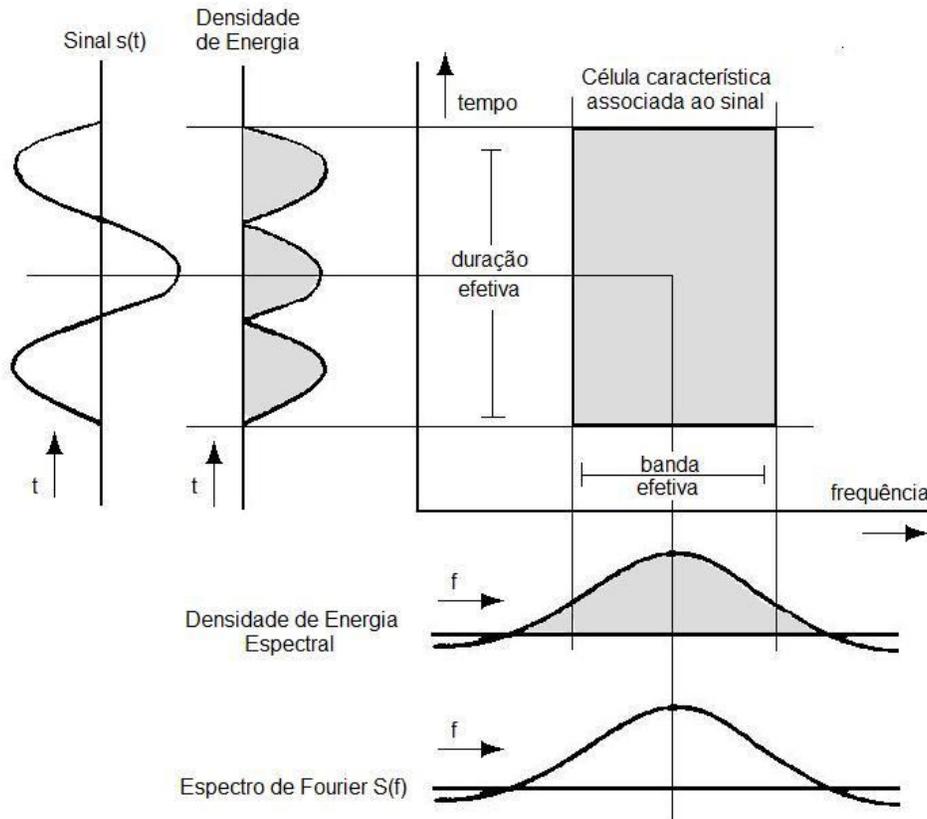


Figura 2 Teoria do Quantum Acústico de Gabor.

Um “quantum” de som é representado então como um retângulo com larguras Δt e Δf e com área mínima, isto é, $\Delta t \times \Delta f = 1$. Observe que da desigualdade (1) resulta que tempo e freqüência são quantidades inversamente proporcionais. Estes retângulos foram denominados por Gabor de *células*. Para cada uma delas está associado um quantum acústico. Qualquer sinal sonoro real (que tem um início e fim) ocupa certo retângulo no Espaço de Gabor cuja área é sempre maior ou igual a 1. Esta área pode então ser decomposta em uma rede (ou matriz) de células (retângulos) de Gabor ou, dito de outra forma equivalente, o sinal é representado por uma soma de quanta acústicos (Figura 3).

Do ponto de vista matemático, Gabor tomou sinais elementares (ou quanta acústicos) como senóides moduladas por funções gaussianas, isto é, sinais que podem ser expressos como:

$$s(t) = \exp\{a^2 \cdot (t-t_0)^2\} \cdot \{\cos(2\pi f t) + i \cdot \text{sen}(2\pi f t)\} \quad (2a)$$

ou equivalentemente,

$$s(t) = \exp\{a^2 \cdot (t-t_0)^2\} \cdot e^{i2\pi f t} \quad (2b)$$

onde f_0 é a frequência da função senoidal.

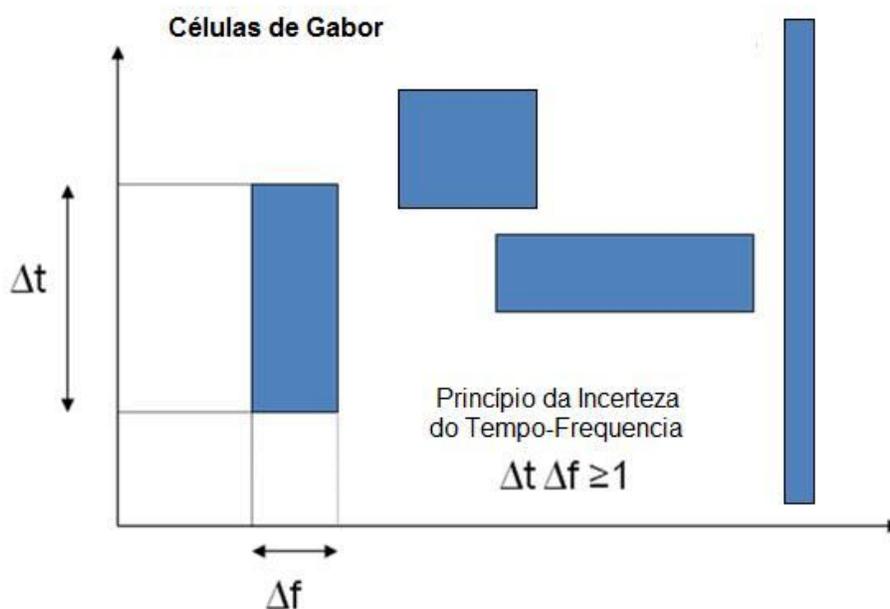


Figura 3 Princípio da Incerteza no Espaço Tempo-Frequência.

A Transformada de Fourier $S(f)$ (espectro do sinal) apresenta o mesmo comportamento gaussiano. A largura da Gaussiana de $s(t)$ é dada por $\Delta t = \pi^{1/2} / a$, e a largura da Gaussiana que modula o espectro $S(f)$ é dada pela relação inversa, $\Delta f = a / \pi^{1/2}$. Claramente vale então a desigualdade (1) e neste caso elementar vale exatamente a igualdade na fórmula (1).

Logo que a teoria de Gabor foi publicada, diversos experimentos sobre os limites da percepção humana do som foram iniciados. Muitos destes experimentos investigavam os limites da discriminação de frequências, tempo, intensidade e silêncio. Um experimento em particular constatou que, quando dois eventos sonoros ocorrem separados por um intervalo inferior a 10ms (ou até 21ms dependendo de cada indivíduo), a audição humana os percebe como um único som. Ou seja, a audição humana também ocorre em “janelamentos”, o que resulta nos efeitos de ambigüidade presente no nível temporal do microsom como fusão e mascaramento (GABOR, 1947).

Apesar de Gabor estar interessado em obter um método para análise do som, sua teoria resulta que, de forma inversa à decomposição do som, é possível compor sons mais complexos por meio da emissão de uma grande quantidade de “partículas de som” que se fundem sendo percebido como uma única estrutura sonora (ROADS, 2001). Encontramos alguns exemplos de sons caracteristicamente granulares na natureza como, por exemplo, o som das gotas de uma cachoeira, de pedras rolando, de um enxame de abelhas e também como aplausos, murmúrios, motores a combustão. Inclusive, é possível obter texturas granulares com uso de instrumentos acústicos, isto ocorre mais claramente em instrumentos de percussão como maracas e chocalhos (onde cada guizo emite a sua partícula de som), e também em orquestrações como *pizzicato* de cordas ou *col legno* (tocar as cordas com o lado de madeira do arco), som de válvulas pressionadas nos metais e sopros, etc.

Atualmente, a Síntese Granular (*Granular Synthesis* - **GS**) é pensada como um método de síntese de som cujo paradigma teórico continua sendo a teoria do Quantum Acústico, mas cada partícula de som, denominada *grão*, tem uma definição mais flexível do que as partículas de som de Gabor. Por exemplo, Roads considera sons granulares como aqueles cujo tempo de duração é de 1 a 100 ms, não importando a banda de frequência, ou seja, o seu conteúdo espectral (ROADS, 2001). Ao focar no domínio do tempo, a Síntese Granular apresenta-se como alternativa a métodos tradicionais como Síntese Aditiva de Fourier que focalizam o domínio da frequência. Sons, timbres, texturas e música são aqui obtidos pelo seqüenciamento temporal de grãos, minúsculos “pedaços de sons”, da ordem de um a uma centena de milissegundos.

2.2 Síntese Granular Analógica

O uso musical da síntese granular tem início antes da era digital. Webern, representante da Segunda Escola de Viena, estabeleceu uma tendência de atomizar cada vez mais o material sonoro. Em diversas de suas peças é possível destacar esta tendência em características como a pequena duração total da peça, grande presença do silêncio e uso de materiais sonoros pequenos, mas compostos por uma grande quantidade de notas de curta duração. Com o advento das possibilidades de manipular o som através de meios eletromagnéticos, as escolas de Darmstadt e Colônia deram continuidade a esta tendência estética ao focalizar no “ponto” (pequeno trecho de som) como menor unidade de material composicional, em oposição às notas tradicionalmente escritas em partitura (THOMSON, 2004).

Stockhausen trouxe uma importante contribuição para a síntese granular ao compor em 1952 o movimento *Étude* da peça *Kontakt*. Nesta composição granular pioneira, Stockhausen utiliza diversos geradores de impulso analógicos, cada um deles conectado a outros componentes como filtros passa-baixa, filtros passa - alta, gravadores de fita em loop e circuitos de realimentação com diferentes arquiteturas (ROADS, 2001). É interessante notar que, ao compor com síntese granular, pequenas alterações na estrutura do grão (conteúdo espectral, envoltória, forma de onda, duração) resultam globalmente em alterações nas propriedades da textura sonora. Ou seja, alterações no nível micro-temporal resultam em efeitos que são perceptíveis no nível macro-temporal do som.

Em 1957, Stockhausen publicou o polemico artigo “...*how time passes...*” (STOCKHAUSEN, 1957) onde propõe uma visão composicional unificada para as diversas camadas temporais da estrutura musical. Definindo um *período* como o tempo de repetição entre ciclos, o ato musical de compor é estudado, ou analisado, em camadas relacionadas à sua duração. Assim estuda-se separadamente forma, frase, motivo, ritmo, bem como melodia. O principal conceito utilizado neste artigo é de que pulso e altura são eventos da mesma natureza, mas com períodos diferentes. Um pulso nada mais é do que uma altura com frequência infra-sônica, enquanto uma altura é um pulso com período muito curto ao ponto de ser percebido como um som contínuo e, por serem eventos da mesma natureza,

devem ter igual tratamento musical. Nesta época Stockhausen estava trabalhando com serialismo integral e assim pretendia obter uma série de 12 números que fosse aplicável para as pulsações de forma idêntica à série de alturas melódicas.

Iannis Xenakis foi o primeiro a formalizar um possível modelo de composição para síntese granular dividindo o tempo em unidades iguais (síncrona) e de curtíssima duração as quais denominou telas (*screens*). Cada tela é representada por uma matriz ocupada por grãos de som. Uma seqüência de telas recebe o nome de livro (*book*). Este modelo está presente em seu famoso livro *Formalized Music* (XENAKIS, 1971). Em 1958, Xenakis trabalhava com o GRM (*Group de Recherches Musicales*) em Paris para produzir diversas peças para fita magnética, entre elas a composição granular “*Concrete P.H.*”, na qual usou uma gravação do som de carvão sendo despejado como material composicional (um som natural, mas de característica granular). A fita magnética foi fatiada em uma grande quantidade de pedaços pequenos, mas de tamanho variável, e então reorganizada e colada. O resultado é uma textura rugosa, com estalidos brilhantes e secos, que está em constante transformação. Xenakis foi também o primeiro a utilizar o termo microsom (*Microsound*) para se referir a peças onde o controle do compositor se verifica no nível do Quantum Acústico (ROADS, 2001).

2.3 Síntese Granular Digital

Apesar da possibilidade de se compor música granular com recursos analógicos, e até mesmo acústicos, a Síntese Granular apresenta mais recursos e facilidade de manuseio no domínio digital. Muitas das principais contribuições nesta linha de pesquisa foram feitas por Curtis Roads (THOMSON, 2004), que realizou uma grande quantidade de estudos em Síntese Granular desde 1978. Daí em diante foi estabelecido um respeitável vocabulário de estruturas sonoras como nuvens, pulsares, *grainlet*, *trainlet* e outros; bem como uma diversidade de métodos de síntese granular, granulação de sons gravados, análise e re-síntese com contribuições de vários pesquisadores/compositores as quais são apresentadas por Roads em seu famoso livro “*Microsound*” (ROADS, 2001).

Os programas de Síntese Granular desenvolvidos por Roads não operavam em tempo real, isto é, era necessário especificar todos os parâmetros do programa antes de executá-lo, para posteriormente obter o *output* em um arquivo de som. Barry Truax foi pioneiro ao realizar Síntese Granular em tempo real com o uso de um microprocessador de sinal digital chamado DMX-1000 (TRUAX, 1988) no Instituto de Sonologia de Haia, na Holanda. Este dispositivo de hardware dedicado sintetizava os grãos em tempo de real e permitia a alteração nos parâmetro da síntese a qualquer momento. Em *Riverrun*, composta em 1986, Truax faz analogia às partículas de água de um rio em constante fluxo, a peça é um exemplo de controle da Síntese Granular em tempo real.

Uma vez estabelecida uma linguagem granular, e com o avanço da tecnologia digital, pesquisas recentes tem se voltado para o uso de modelos matemáticos para controle de alto nível sobre os parâmetros da síntese. Desta forma busca-se obter novas estruturas sonoras a partir de um conjunto mínimo de parâmetros. DiScipio (DISCIPIO, 1990) controlava os parâmetros de síntese através de Modelos Fractais, Sistemas Dinâmicos e Teoria do Caos. Nesta mesma direção mais formal, Maia e Miranda (MAIA, 2005) desenvolveram um sistema desenvolvido em MATLAB denominado **FuzzKov** baseado em Conjuntos Fuzzy e Cadeias de Markov. A Teoria dos Grafos, como estrutura de controle, também foi proposta por Valle e Lombardo (VALLE, 2003). Pode-se inferir destes exemplos que uma grande profusão de sistemas computacionais para síntese granular pode ser obtida como aplicações de modelos matemáticos e algorítmicos para geração e controle de um fluxo de sons granulares.

2.4 Compendo com Grãos

Quando Gabor fundamentou um novo modelo de representação do fenômeno físico do som, também deu origem à pesquisa sobre uma nova técnica de síntese sonora que mais tarde veio a ser conhecida como síntese granular. Ao permitir que compositores manipulem o material sonoro ao nível do microsom, cria-se um imenso espaço de possibilidades sonoras, ou seja, a técnica de manipular o microsom abre novos horizontes, não somente à síntese digital de áudio, mas também à composição musical. Neste recente e diversificado

ambiente os conceitos tradicionais de ritmo, melodia e harmonia se mostram ineficazes para auxiliar o compositor (VAGGIONE, 1996). Uma das alternativas para encontrar ferramentas mais adequadas para orientar a composição com síntese granular é recorrer às peças musicais e textos de grandes contribuintes da área como, por exemplo, Curtis Roads e Barry Truax. Realizando uma breve análise em peças musicais destes dois expoentes da síntese granular foi possível identificar algumas formas recorrentes de organização dos grãos. Das peças de Curtis Roads destacam-se duas formas que denominamos *palavras granulares* e *estruturas clássicas*, enquanto que *texturas* e *processos* são exemplificados pela obra de Barry Truax.

Durante décadas de trabalho, Curtis Roads desenvolveu uma série de sistemas de síntese granular nos quais diferentes métodos de controle e organização dos grãos resultavam em algumas sonoridades particularmente interessantes. Ao longo dos anos estas sonoridades puderam ser agrupadas construindo-se uma taxonomia de estruturas granulares, a qual Roads apresenta em seu livro *Microsound* (ROADS, 2001). Apesar de Roads também fazer uso da síntese granular para formar texturas e conduzi-las dentro de um espaço de parâmetros, o que queremos destacar de sua obra musical é o uso do que denominamos *estruturas clássicas* e *palavras granulares*.

Estruturas clássicas são sons de natureza granular com sonoridades características, muitas das quais sintetizadas pela primeira vez pelo próprio Roads, e de uso recorrente em peças musicais do gênero. Sistemas digitais de síntese granular são controlados por uma multiplicidade de parâmetros que podem apresentar grandes variações resultando em um imenso espaço de possibilidades sonoras. No entanto em algumas regiões específicas desse universo sonoro multidimensional podem ser geradas as sonoridades características das *estruturas clássicas*. Nos resultados do capítulo 6 serão apresentados exemplos sonoros de diversas *estruturas clássicas*.

A segunda forma de organizar os grãos que destacamos das composições de Roads é a formação de estruturas sonoras denominadas *palavras granulares*. Esta é uma poética onde cada grão de som é interpretado como um fonema. Diferentes fonemas são então agrupados em “palavras” com a duração de pequenos motivos musicais, de meio segundo até 5 segundos. A Figura 4 apresenta um exemplo de “palavra” com duração de dois

segundos e meio encontrada na peça Pictor Alpha de Roads. Por meio da escuta e da análise do espectrograma podemos separar as cinco sílabas que a compõe: um pulsar de curta duração (I) seguido de um ataque grave (II) e de uma reverberação bem rugosa (III), uma sílaba percussiva é repetida duas vezes (IV) e então a “palavra” termina com um trecho de ruído (V). Estruturas semelhantes são encontradas em diversas peças da obra de Curtis Roads, como por exemplo, Organic, Half-Life e Volt Air.

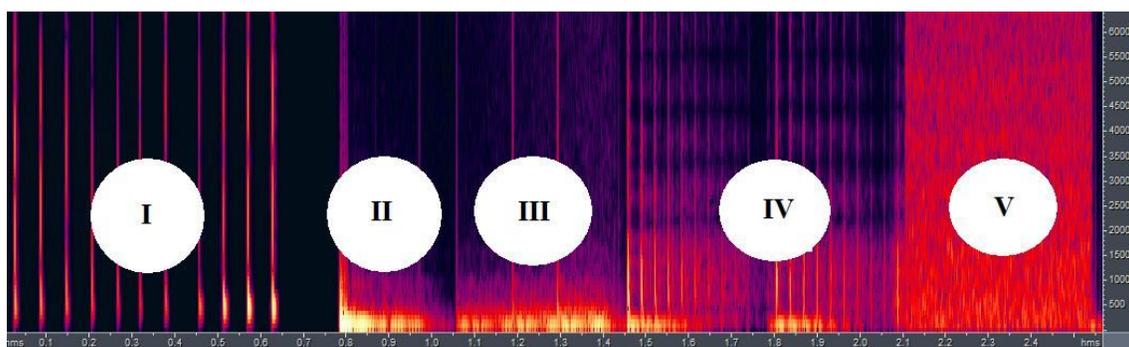


Figura 4 Uma “palavra” de 2,5s da peça Pictor Alpha (2003) de C. Roads

Podemos considerar esta pequena estrutura sonora como um objeto musical rígido por dois motivos: primeiramente porque apresenta suas secções (sílabas) muito claramente e com contornos bem delimitados e, em segundo lugar porque não apresenta um processo de transformação gradual que seja notável no espectrograma. Apesar desta aparente rigidez, Roads manipula este material sonoro com tal maestria que confere ao mesmo uma organicidade muito bela e digna de ser considerada uma obra prima da música eletrônica. Utilizando uma hierarquia de texturas, estruturas clássicas e palavras, Roads compõe paisagens sonoras que nos remetem a uma floresta habitada por uma fauna sintética, onde cada animal digital profere seu som específico formando longos diálogos. Também é notável o fato de Roads declarar sua preferência ao uso de material puramente sintetizado (conforme entrevista concedida por Curtis Roads para o canal de televisão de internet VBS, entrevista publicada em 11 de Maio de 2009 no endereço <http://www.vbs.tv/watch/motherboard/curtis-roads>). Segundo o mesmo não há nada de

errado com o material concreto ou amostrado (*sampling*), porém a paleta de sons sintetizados também apresenta a sua beleza e potencial artístico.

É bem conhecido que, no nível do microsom os domínios do tempo e da frequência estão intimamente relacionados, formando uma zona de crepúsculo nos limites da percepção humana. Quando dois eventos sonoros de curta duração (dezenas de milissegundos) ocorrem em um intervalo de tempo muito pequeno (em média, menor que 50 milésimos de segundo) ocorre o fenômeno psicoacústico conhecido como mascaramento. Este fenômeno acontece quando o aparelho auditivo humano percebe os dois eventos isolados como um único evento sonoro, ou seja, o primeiro evento esconde o ataque do segundo. Em termos de síntese granular, quando grãos são emitidos com uma densidade em torno de 20 grãos por segundo ou superior, ocorre fusão dos sons e o que se percebe é uma massa sonora contínua. Desta forma é possível manipular o material sonoro ao nível temporal do microsom para formar texturas compostas com grãos pouco variados (texturas monocromáticas), com grãos muito diversificados (texturas multicromáticas) ou mesmo combinar diversas camadas ou fluxos em uma única textura mais rica.

Uma vez estabelecida uma textura sonora, pequenas alterações nos parâmetros que controlam a síntese de cada grão resultam em alterações nas características da textura. Assim, a síntese granular se apresenta como técnica composicional que enfatiza mais o processo de transformação do som no tempo, realçando as nuances do espectro sonoro. E, portanto, em certo sentido, é um estilo composicional derivado do espectralismo, definido aqui como uma música que focaliza o timbre e o coloca em primeiro plano como o mais importante elemento da estrutura ou linguagem musical (REIGLE, 2008).

Barry Truax explora esta técnica composicional em diversas peças de sua autoria como, por exemplo, em sua célebre composição *Riverrun* (1986). Nesta peça os grãos de som são poeticamente interpretados como gotas de água que formam o corpo caudaloso de um rio. Ao observar apenas um trecho do rio, temos a impressão de que este é formado sempre pela mesma matéria, mas sabemos que o que ocorre verdadeiramente é um fluxo constante de partículas que dão vida a uma intrincada paisagem sonora (TRUAX, 1990).

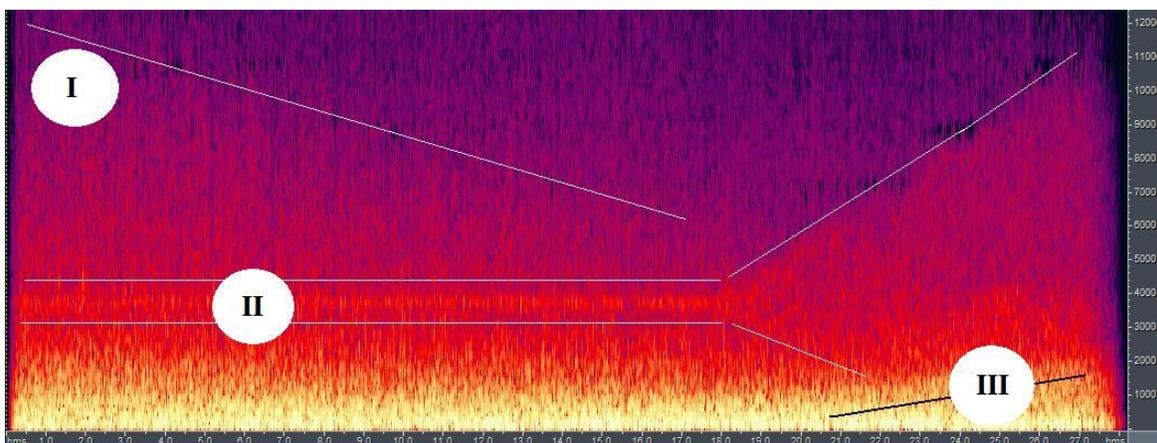


Figura 5 Trecho da peça Riverrun de Barry Truax (aprox. 30s).

Assim, Truax realiza gradualmente pequenas alterações em seus grãos para obter um processo de transformação sonora que é percebido no nível textural. Trata-se da composição do “macro” a partir de manipulações do “micro”, de forma análoga à realizada nas artes gráficas com o uso das estruturas matemáticas conhecidas como fractais. Contrariamente ao exemplo de Roads apresentado acima, não é possível encontrar nesta peça nenhum trecho de curta duração (como 2,5s no caso do exemplo de Roads) que contenha objetos sonoros com contornos bem definidos. Na realidade se utilizarmos uma janela de apenas 2,5 segundos para analisar esta peça, não localizaremos quase nenhum trecho que não seja uma textura homogênea e constante. Porém, ao observar o espectro sonoro de uma distancia maior, isto é, quando observamos um trecho de 30 segundos conforme exibido na Figura 5, podemos ver claramente os contornos espectrais deixados pelos processos de manipulação dos grãos. Particularmente em “II”, podemos ver um fluxo de grãos inicialmente concentrados em uma estreita banda de frequência com centro em 4000 Hz. Após 20 segundos os grãos deste *fluxo* começam a ser dispersos dentro de uma banda de frequência de largura crescente.

Um segundo exemplo de processos de transformação sonora na obra de Barry Truax é encontrado na peça Wings Of Nike (1987). Na Figura 6 vemos um trecho de vinte segundos desta peça no qual podemos notar a presença de três camadas sonoras. Tanto a camada “I”, uma forte vibração no registro grave, como a camada “II”, uma textura bem dispersa em termos de banda de frequência e escurecida pela baixa amplitude, atuam como

material sonoro de sustentação para a camada “III”, que por sua vez exerce o papel de “solista”, não somente por estar no registro mais agudo deste segmento sonoro, mas principalmente por ser o ponto mais atraente à nossa atenção, devido ao seu movimento espectral linear descendente com abertura de banda, ou seja, neste lento gesto sonoro, podemos observar um processo de dispersão contínua de grãos limitados pelas fronteiras de uma banda de frequência, que se expande linearmente em direção às frequências mais graves.

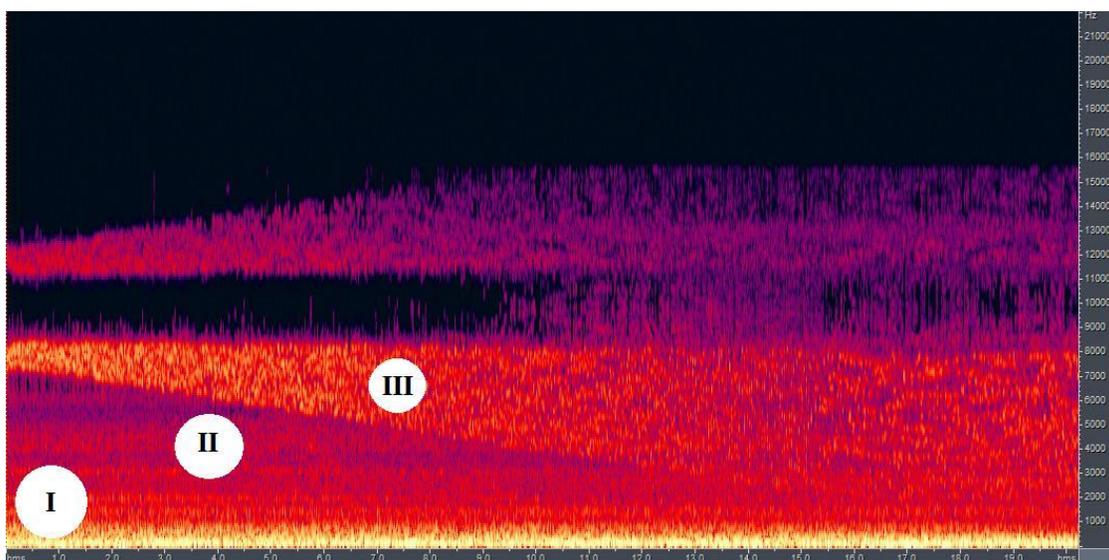


Figura 6 Espectro de um trecho de 20 segundos da peça Wings of Nike de B. Truax

Atualmente, obras de Barry Truax, Curtis Roads e muitos outros compositores que utilizam síntese granular podem ser analisadas dentro da perspectiva da Espectromorfologia de Dennis Smalley a qual pode ajudar na orientação de processos de transformação sonora (SMALLEY, 1997). No caso particular da síntese granular, cada grão de som deixa uma marca espectral muito bem delimitada de acordo com sua frequência, duração, amplitude, forma de onda e envoltória. Isto torna possível compor quase que diretamente sobre o espectro sonoro colorindo-o com uma paleta de grãos. Por meio da variação dos parâmetros da síntese é possível realizar processos de transformação das texturas assim como os

descritos na seção de Movimento e Crescimento (*Motion and Growth*) do artigo de Smalley.

Separando o material sonoro em múltiplas camadas é possível compor seções onde uma delas se comporta como “solista” (mais aparente) acompanhado de outras camadas, ou seções onde diversas camadas se transformam simultaneamente alternando o plano que ocupam, entre o primeiro plano e o plano de fundo (*foreground e background*), formando uma rede de processos. Duas camadas podem também convergir em uma única massa sonora ou, de maneira oposta, os grãos de uma única textura podem ser transformados de forma a separar a textura inicial em duas direções diferentes passando então a ser percebida como duas camadas distintas. Uma única camada é composta por partículas de som que, como vimos, Gabor denominou quanta acústicos. Aumentando a quantidade (densidade de grãos por segundo), tamanho (duração) ou energia (amplitude) destas partículas é possível, por exemplo, transformar uma textura de natureza difusa ou rarefeita em texturas densas.

Finalmente, a própria quantidade, a velocidade e a variação de direção dos processos tornam-se parâmetros a serem controlados pelo compositor, que então contrasta seções mais estáveis com momentos de muita turbulência (CLARKE, 1996). No capítulo 6 são apresentados diversos exemplos sonoros obtidos com o uso do **EVOGrain** e do **GranularStreamer** onde usamos os termos definidos na Espectromorfologia de Smalley para descrever os processos como também a terminologia de Roads e Truax.

2.5 Estado da Arte em Síntese Granular

Nesta seção são apresentados alguns softwares para Síntese Granular disponíveis para download pela internet (último acesso em 06/07/2010). Alguns destes sistemas são de uso livre (*freeware*), outros disponibilizam uma versão gratuita para testes por um período de tempo limitado. Foram avaliados cinco sistemas para síntese granular: **Granulator**, **Granulab**, **Kenaxis**, **AL** e **Chaosynth**. Após realizar uma breve navegação e experimentação com tais sistemas foi possível identificar suas particularidades e suas características em comum, conforme descrições a seguir.



Figura 7 A interface do sistema Granulator

Os sistemas **Granulator** (<http://www.nicolasfournel.com/granulator.htm>) e **Granulab** (<http://www.abc.se/~re/GranuLab/Granny.html>) são muito semelhantes em sua forma de operação e nos resultados apresentados. O primeiro sistema analisado é denominado **Granulator** e apresenta a interface exibida na Figura 7. O Granulator lê um arquivo WAVE e a partir deste gera grãos conforme a configuração de doze parâmetros como, por exemplo, “quantidade de grãos por segundo”, “duração de cada grão”, “*pitch*” (velocidade de leitura da forma de onda) e outros. Todos os parâmetros são determinados por um valor central e um delta de variação, o que garante diversidade ao material sonoro gerado. O sistema pode ser controlado com o mouse, mas também permite associar cada um dos botões giratórios a um controlador MIDI.

Assim como o Granulator, o sistema **Granulab** apresenta um conjunto de onze parâmetros, exibidos na interface como *sliders* verticais, também compostos por um valor central e um valor de variação. No Granulab, o conteúdo de cada grão pode ser uma forma de onda senoidal ou uma forma de onda lida de um arquivo WAVE. Embora o Granulab não permita controlar os parâmetros através de um controlador MIDI, o sistema

disponibiliza uma função que salva a configuração atual dos parâmetros como *presets*, que podem ser carregados posteriormente. E ainda, uma segunda janela da interface apresenta o desenho de um quadrado onde é possível associar um *preset* a cada um dos quatro cantos. Ao mover o mouse pelo interior deste quadrado é feita uma interpolação (combinação convexa) dos valores dos parâmetros de cada *preset* associado ao quadrado cujo resultado é aplicado à síntese granular em tempo real.

Embora em ambos os sistemas, **Granulator** e **Granulab**, a síntese ocorra em tempo real, estes sistemas sintetizam apenas um *fluxo* de grãos. Atualmente a capacidade de síntese de áudio em tempo real dos computadores portáteis permite sintetizar diversos *fluxos* simultaneamente levando a sonoridades compostas de estruturas granulares bastante complexas. Além disto, as interfaces apresentadas por estes aplicativos não são amigáveis e tampouco intuitivas.

Em uma direção similar aos sistemas anteriores, mas incorporando a síntese a diversos *fluxos* de grãos, temos o sistema **Kenaxis** (<http://www.kenaxis.com/>), exibido na Figura 8. Os múltiplos *fluxos* de grãos são agrupados de acordo com o tipo de síntese de granular que realizam e são chamados de *Klangs*, *Basuneras* e *Granulators*. Este é um sistema muito completo e versátil com uma grande quantidade de parâmetros e uma quantidade ainda maior de menus e opções de normalizar, convoluir e transformar o material sonoro. É o sistema, dos analisados aqui, que apresenta melhor acabamento como software comercializável e a maior “biblioteca” de síntese granular e que também apresenta um extenso suporte a controladores externos (MIDI e OSC). Porém, da mesma forma que os sistemas analisados anteriormente, sua interface é um tanto desencorajadora. O entendimento de cada um dos diversos parâmetros disponíveis requer a leitura de um longo manual de instruções e da realização de muitos experimentos, minimizando seu valor quanto ao critério de interface amigável e intuitiva.

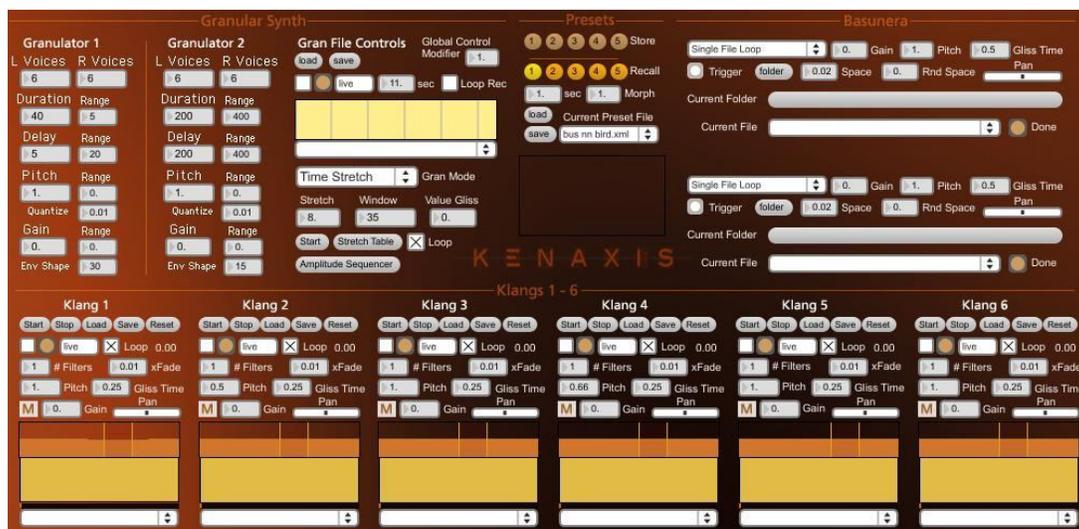


Figura 8 Kenaxis, uma orquestra completa

O próximo sistema analisado é um plug-in para um ambiente composicional denominado **Algorithmic Composition Environment**, ou apenas **AL**. Este aplicativo é um seqüenciador onde o usuário organiza eventos (ou objetos) no tempo e depois executa a peça montada em tempo diferido, ou seja, o sistema não opera em tempo real. O conteúdo destes objetos pode ser um arquivo de som já existente ou então pode ser criado através de diversos plug-ins de composição algorítmica. Embora o foco deste sistema não seja a Síntese Granular, o mesmo oferece um plug-in denominado **Schroedinger Equation with Radial Symmetry** (FISCHMAN, 2003). Através do uso da famosa *Equação de Schrodinger*, da Mecânica Quântica, que descreve como o estado quântico de um sistema físico evolui ao longo do tempo, é gerado um objeto contendo uma nuvem de grãos. O usuário pode configurar dezenas de parâmetros que especificam o tamanho, o conteúdo e a distribuição temporal dos grãos. O objeto pode então ser posicionado temporalmente entre outros objetos no AL e executado (destacando novamente que a síntese granular não ocorre em tempo real).

O último sistema analisado, **Chaosynth**, é uma aplicação de um modelo computacional chamado Autômato Celular, e foi desenvolvido por Eduardo Reck Miranda (MIRANDA, 1995). Assim como o sistema anterior, trata-se de uma interessante aplicação de um modelo de mais alto nível controlando os parâmetros da síntese granular. O

Chaosynth também apresenta um enorme conjunto de parâmetros configuráveis para se obter as mais diversas texturas e sons granulares, como podemos notar pela interface exibida na Figura 9.

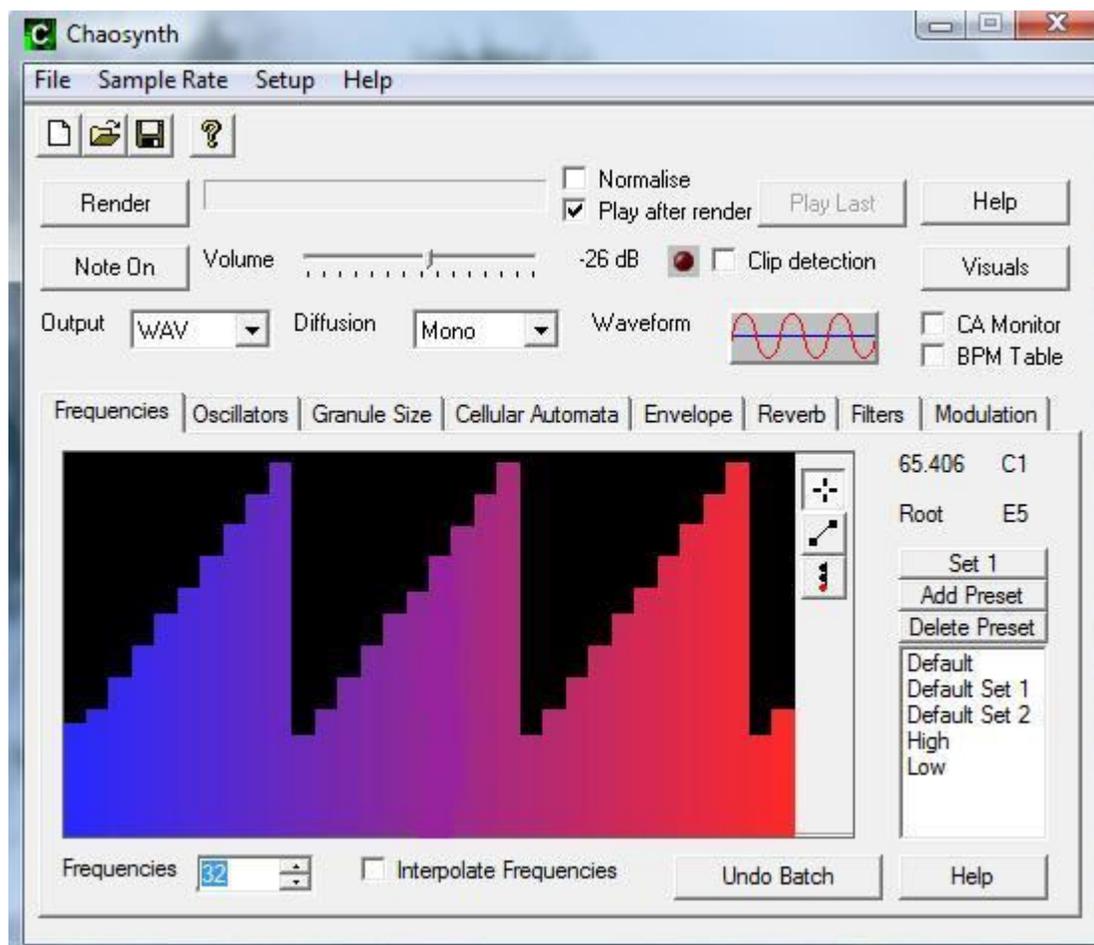


Figura 9 Chaosynth, aplicação de autômato celular em síntese granular

Os resultados obtidos na análise dos dois últimos sistemas descritos, o **AL** e o **Chaosynth**, são semelhantes. Embora os dois sistemas sejam capazes de gerar uma grande diversidade de texturas, nuvens e outros sons granulares controlando a síntese por meio de um modelo de alto nível, a apresentação deste modelo ao usuário é muito complexa e torna a interface pouco intuitiva. Entender como cada parâmetro influencia o resultado é uma

tarefa exploratória demasiadamente longa, lembrando ainda que, no caso do AL, a síntese não ocorre em tempo real.

2.5 Motivação

Uma dificuldade comum apresentada pelos métodos de síntese sonora é como realçar a saída sonora dando a ela vivacidade similar à dos sons naturais, ou seja, como reproduzir a complexa evolução temporal do espectro em detalhe. A Síntese Granular se apresenta como método de síntese que enfatiza as peculiaridades que ocorrem em nível micro-temporal. Neste método, centenas e até milhares de grãos são necessários para sintetizar sons com a duração da ordem de alguns segundos. Este método apresenta alto custo de processamento computacional, principalmente em sistemas de controle em tempo real, além do que seria extremamente tedioso ou até mesmo inviável especificar parâmetros para cada grão.

Com a inviabilidade da especificação direta dos parâmetros sonoros para cada grão é natural a idéia de que esta tarefa possa ser realizada por um processo automatizado, no qual somente controles de alto nível sejam atribuídos ao usuário (compositor) e a especificação (valor) de cada parâmetro de controle de baixo nível seja gerada de maneira automática por um procedimento algorítmico dedicado. Nos sistemas analisados na seção anterior encontramos dois exemplos de modelos matemáticos ou algorítmicos controlando a síntese de grãos de som: um autômato celular e uma equação da mecânica quântica. Nesta pesquisa queremos obter um sistema onde o controle dos parâmetros da síntese granular seja feito pelo usuário de uma maneira simples com uma interface amigável. Esta interface foi implementada como uma generalização do espaço Tempo-Frequência de Gabor, o qual será detalhado mais á frente. Além disto, o sistema utiliza um modelo conhecido como Algoritmos Genéticos para controlar os parâmetros da síntese de som que também se mostrou muito adequado.

Algoritmos Genéticos (*Genetic Algorithm* - GA) são programas de computador cuja modelagem é inspirada na teoria da evolução das espécies de Darwin (WIRT, 1994). Holland foi pioneiro no desenvolvimento deste modelo computacional e na experimentação

com algoritmos genéticos (HOLLAND, 1992). Aplicações musicais de algoritmos genéticos têm provado seu sucesso em sistemas tais como o VoxPopuli (MORONI, 1999), um software para composição algorítmica que roda sobre o protocolo MIDI, e EESynth (FORNARI, 2007), um software para síntese de sons baseado em GA que combina formas de onda evolutivamente. Ambos os sistemas foram desenvolvidos por pesquisadores do NICS – UNICAMP. A teoria básica dos algoritmos genéticos, necessária para desenvolvimento de nosso aplicativo EVOGrain, será apresentada com maior detalhe na seção 3.2. Neste sistema esperamos obter uma textura dinamicamente rica (estruturas sonoras em constante evolução), gerada pelo Algoritmo Genético através dos operadores de cruzamento, mutação e avaliação (*fitness*). Além de experimentar o modelo do algoritmo genético no controle da síntese granular, o sistema deve apresentar as qualidades, não obstante evitando as limitações, encontradas nos sistemas avaliados anteriormente. Assim estabelecemos as seguintes diretrizes de projeto:

a) a interface deve ser amigável e intuitiva, levando a alta usabilidade tanto por músicos com pouca experiência no uso de computadores, como também por músicos com pouca experiência em música eletroacústica a até mesmo sem conhecimentos prévios acerca de síntese granular;

b) o tempo de aprendizado inicial deve ser bem curto, porém sem implicar em limitações ao sistema, ou seja, o sistema deve permitir que a prática continuada resulte em melhoria considerável no uso;

c) opção de salvar e carregar *presets* contendo informações sobre a configuração dos parâmetros e levando a uma forma de partitura alternativa.

d) útil ao compositor de música eletroacústica e eletrônica;

e) opção de salvar arquivos WAVE com a síntese realizada;

f) deve operar em tempo real (com baixa latência);

g) deve ser integrável com outros sistemas;

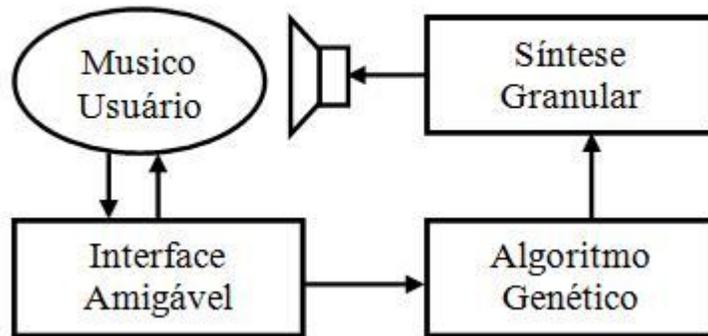


Figura 10 Fluxograma de operação do EVOGrain

Foi dentro desta perspectiva, e inspirados no modelo de Gabor, que desenvolvemos **EVOGrain**, nosso sistema para Síntese Granular em tempo real. A Figura 10 exibe o fluxograma de operação do sistema proposto: o usuário comunica-se com o sistema em uma interface intuitiva e amigável que controla um algoritmo genético em constante execução. Por sua vez, a cada ciclo do fluxograma o resultado do processo de evolução do algoritmo genético atualiza os parâmetros da síntese granular. O retorno ao usuário (*realimentação*) se dá tanto pelo som sintetizado pelo sistema, como graficamente pela interface. No próximo capítulo apresentamos os detalhes do sistema **EVOGrain**.

3 O SISTEMA EVOGRAIN

EVOGrain é um sistema de síntese granular em tempo real que controla simultaneamente dois *fluxos* de grãos através de uma interface gráfica simples e intuitiva. O controle dinâmico da síntese ocorre com a interação do usuário com uma seção da interface constituída de três janelas (Figura 11), cada uma delas interpretada como um espaço real bidimensional (R^2). Tudo que precisa ser feito para o sistema iniciar a síntese é clicar no botão “Play” e então usar o mouse para desenhar retângulos vermelhos e azuis nos três espaços disponíveis (vermelhos com o botão esquerdo e azuis com o botão direito). Os retângulos desenhados pelo usuário são utilizados como alvos para direcionar a evolução de uma seqüência de gerações de indivíduos através do algoritmo genético. O algoritmo genético por sua vez controla os parâmetros da síntese granular. Os detalhes serão apresentados no decorrer deste capítulo.



Figura 11 Interface do sistema EVOGrain v2.0.

A interface também disponibiliza outros campos para configurar funções diversas do sistema como: nome do arquivo WAVE de saída, configurações OSC e outros que também veremos mais adiante. O EVOGrain é dividido em três módulos principais onde diversos processos são executados paralelamente. Os módulos são descritos nas seções a seguir: Máquina de Síntese Granular em Tempo Real, Algoritmo Genético e Interface.

3.1 Máquina de Síntese Granular em Tempo Real

No núcleo do sistema é executado o sintetizador em tempo real que é encarregado de controlar desde funcionalidades básicas como acessar recursos da placa de som até a geração, armazenamento e execução de amostras de áudio. O sintetizador em tempo real trabalha atualmente com fluxos independentes de grãos, cada um com taxa de amostragem de 44.100Hz, 16 bits de profundidade e dois canais.

A arquitetura da máquina de síntese granular é construída sobre o paradigma do produtor e consumidor. O processo produtor é responsável por calcular as amostras e armazená-los em um *buffer*. O processo consumidor aguarda até que o *buffer* esteja completo e então os envia à placa de som do computador. A melhor situação, para obter uma latência nula, seria enviar cada amostra calculada diretamente para a placa de som, mas devido às oscilações na velocidade de processamento e com placa de som consumindo amostras numa velocidade maior que a velocidade de enchimento do *buffer* isto resultaria em cliques e interrupções sonoras. Após alguns experimentos foi definida a latência de 500ms como um equilíbrio satisfatório entre qualidade sonora e o controle em tempo real resolvendo o problema acima.

Cada fluxo é controlado por seis pares de parâmetros dinâmicos que podem ser atualizados a qualquer momento pelo usuário através de desenhos de retângulos na interface. As coordenadas destes retângulos são associadas aos parâmetros dinâmicos definidos em intervalos, ou seja, com um valor inicial e um valor final. Cada novo grão sintetizado dentro de uma família de grãos será criado com seus seis parâmetros sorteados com distribuição estatística uniforme dentro dos intervalos definidos pelos retângulos desenhados na interface. Os parâmetros dinâmicos são:

-*Intervalo de Freqüência*: a freqüência da forma de onda do grão. Se a forma de onda estiver sendo lida de uma *wavetable* este parâmetro determina a velocidade de leitura da tabela;

-*Intervalo de Duração*: a duração de cada grão;

-*Intervalo de Densidade de Fluxo*: a velocidade com que novos grãos serão criados, medida em grãos por segundo (GPS);

-*Intervalo de Offset*: Caso esteja sendo utilizada a granulação de uma forma de onda, este parâmetro determina a posição de início de leitura do próximo grão.

-*Intervalo de Distância Espacial*: simula a distância do alto-falante ao ouvinte. Afeta o volume sonoro (*loudness*) do grão;

-*Intervalo Panorâmico*: determina o posicionamento estereofônico do grão.

Cada *fluxo* também é controlado por parâmetros estáticos que determinam a função envoltória ou o “janelamento” de cada grão, bem como sua forma de onda. Estes parâmetros devem ser especificados antes da máquina de síntese granular ser iniciada e não podem ser alterados em tempo real. A Figura 12 mostra as opções disponíveis para envoltória dos grãos na versão 2.0 do **EVOGrain**:

-*Gaussian*: Leve ataque e decaimento;

-*Fat Gaussian*: Semelhante à gaussiana, mas com largura maior;

-*Expodec*: exponencial decrescente, efeito altamente percussivo;

-*Rexpodec*: exponencial;

-*Thin Gaussian*: semelhante à gaussiana, mas um pouco mais percussivo;

-*No Envelope*: sem envelope.

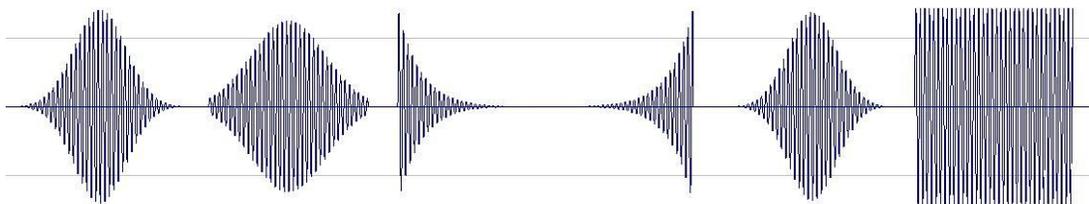


Figura 12 Envoltórias no sistema EVOGrain

As possibilidades para a forma de onda dos grãos são listadas a seguir:

-*Senoid*: uma onda senoidal;

-*Glisson*: um glissando sintetizado a partir de senóides dentro do intervalo definido;

-*Addsenoid*: Síntese aditiva com senóides. Permite especificar para cada parcial sua amplitude e sua frequência relativa à fundamental que é gerada dentro do intervalo de frequência associado a um retângulo desenhado na janela Tempo-Frequencia da interface;

-*Square*: uma onda quadrada;

-*Random*: ruído branco;

-*Granulator*: permite que o usuário selecione um arquivo de audio digital em formato *wave* estéreo. Esta opção abre possibilidades para uso de infinitos formatos de onda. Arquivos *wave* de grande duração são “fatiados” em grãos de duração especificada enquanto que arquivos muito curtos, menores que os grãos definidos pelo retângulo alvo da janela Tempo-Frequencia, são repetidos em loop até preencher a duração total do grão.

O processo produtor está constantemente sendo executado sob controle dos seis parâmetros descritos anteriormente. De acordo com a densidade do *fluxo* e com o tamanho de cada grão pode ocorrer a sobreposição (*overlapping*) de grãos. O Quadro 1 exhibe o pseudocódigo simplificado do algoritmo executado pelo processo produtor para obtenção cada amostra de áudio:

- Loop “enquanto estiver executando”

- Para “cada fluxo de grãos”

- Se foram definidos novos valores para o intervalo de densidade de fluxo:

- Sorteie uniformemente um valor único para a densidade de grãos por segundo dentro do intervalo definido

- Com base na densidade de grãos por segundo verifica se é o momento de criar um novo grão, em caso afirmativo:

- Sorteie (distribuição uniforme) valores para a frequência, tamanho do grão, offset, distância e pan dentro dos intervalos definidos pelos parâmetros

- cria o novo grão e inclui no depósito de grãos ativos
- Sorteia uniformemente um valor único para a densidade de grãos por segundo dentro do intervalo definido, este valor é a densidade de fluxo utilizada para a criação do próximo grão
- Fim do loop para “cada fluxo de grãos”**
- Inicializa a “amostra resultante” com valor nulo
- Para “cada grão presente no depósito de grãos ativos”**
- Recupera o valor da amostra para a forma de onda do grão neste momento atual
- Recupera o valor da envoltória do grão neste momento atual
- Acrescenta à amostra resultante o valor da operação $\text{valor da amostra} \times \text{valor da envoltória} \times \text{pan}$ (esquerdo e direito)
- Incrementa o ponteiro de momento atual
- Se o momento atual é maior que a duração do grão então retira o grão do depósito de grãos ativos
- Fim do loop para “cada grão presente no depósito de grãos ativos”**
- Aplica uma função de compressão à amostra resultante para este não exceder o valor de 1.0 (amplitude máxima)
- Inclui a amostra resultante no *buffer*
- Fim do loop “enquanto estiver executando”**

Quadro 1 Pseudocódigo do processo produtor de grãos

O processo consumidor é um algoritmo muito simples (Quadro 2):

- Aguarda até que o *buffer* de amostras esteja completo
- Obtém o *buffer* com amostrar para 500 ms
- Envia *buffer* para a placa de som do sistema

Quadro 2 Pseudocódigo do processo consumidor de áudio

A próxima seção apresenta o algoritmo genético e explica como este recebe informação a partir da interface e computa os parâmetros que são enviados para a máquina de síntese granular em tempo real.

3.2 Algoritmos Genéticos

Mencionamos anteriormente que o **EVOGrain** é um sistema para síntese granular que incorpora em sua estrutura um algoritmo genético, um programa de computador modelado na teoria da evolução das espécies de Charles Darwin (WIRT, 1994). Um algoritmo genético é fundamentalmente um programa que busca uma solução ótima para determinado problema em um espaço de parâmetros muito grande. Ao invés de testar todas as combinações de parâmetros que poderiam solucionar o problema, este programa parte de um conjunto reduzido de combinações aleatórias, que então “evolui” em direção a solução esperada. A Figura 13 ilustra as etapas realizadas pelo algoritmo para calcular ciclos de vida e criar novas gerações de uma população (como gerações de uma família).

Neste modelo o conjunto universo de todas as possíveis soluções é descrito por um conjunto finito de parâmetros que podem ser considerados como coordenadas de um vetor em um espaço vetorial que denominamos *Espaço de Parâmetros*. Um *indivíduo* é definido simplesmente como um vetor deste espaço de parâmetros. Nossos indivíduos são desprovidos de qualquer outra estrutura que não a sua carga genética e daí podemos, sem perda de generalidade, identificá-lo com seu *genótipo* (o DNA do indivíduo) e que configura uma solução em potencial. Um conjunto destes indivíduos é chamado de *população*, e pode ser escolhida de tamanho (quantidade de indivíduos) constante ou variável. Como ocorre na seleção natural darwiniana, o objetivo da natureza do sistema é obter, em cada geração, *indivíduos* cada vez mais bem adaptados ao meio ambiente. Para avaliar esta habilidade é definida uma função conhecida como *função de adequação* (*fitness function*). A função de adequação é uma função definida no Espaço de Parâmetros que recebe como argumento o *genótipo* de um indivíduo e retorna um valor real normalizado (entre zero e um) e que reflete a sua adequação ao meio ambiente. Este método de seleção descrito acima é conhecido como *Target-Driven Genetic Algorithm* (guiado por alvo). A cada ciclo de vida, isto é, a cada *loop* do algoritmo genético todos os indivíduos de uma população são avaliados e então separados em subconjuntos de sobreviventes (que serão mantidos no próximo ciclo de vida) e de não sobreviventes (que são descartados).

Para continuar o ciclo de vida são definidos os chamados operadores genéticos de *cruzamento* e *mutação*. O operador de *cruzamento* é uma função que cria um novo indivíduo (*embrião*) a partir dos parâmetros de outros indivíduos pertencentes ao subconjunto de sobreviventes. O ciclo de vida tem continuidade criando-se um novo subconjunto de embriões e aplicando a eles o operador de *mutação*. Para cada embrião é verificada a probabilidade de ocorrer mutação, se o resultado indicar que o embrião deve passar pela mutação, então uma parte de seu genótipo (DNA) é substituída por uma combinação aleatória. Após aplicar este operador o subconjunto de embriões é adicionado ao subconjunto de sobreviventes e então é obtida a nova geração de uma população. É interessante notar que o operador genético de cruzamento pode dirigir o sistema em um processo convergente longe da solução ótima definida pelo alvo. O operador de mutação tem então o papel crucial de gerar variabilidade dentro do espaço de parâmetros de maneira que indivíduos com novas características apareçam em populações posteriores e com isto garantindo que a solução ótima será atingida em tempo finito.

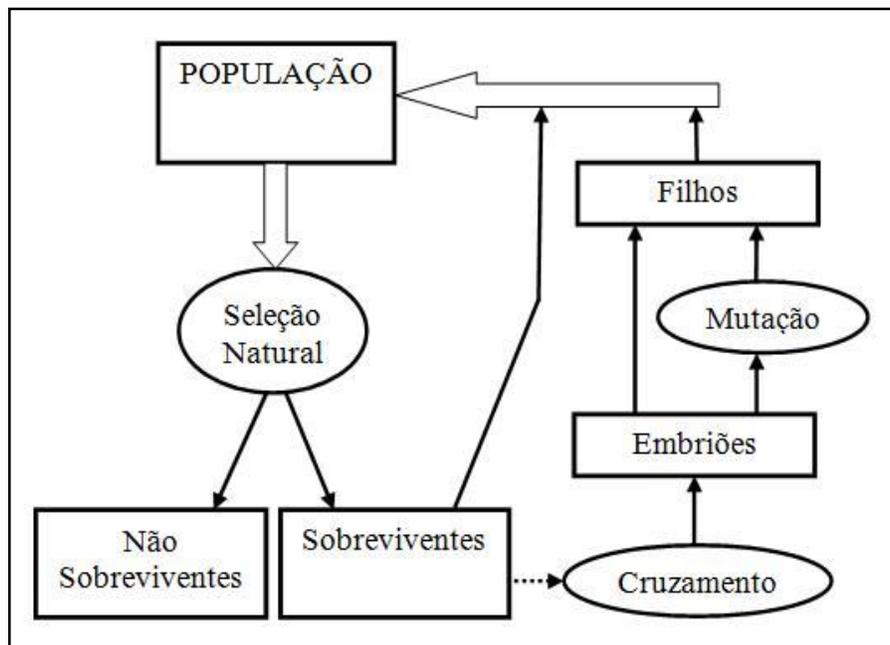


Figura 13 Modelagem Computacional de Algoritmos Genéticos

O comportamento esperado é que com o decorrer de diversos ciclos de vida os indivíduos da população se aproximem cada vez mais da solução ideal. O algoritmo deixa de ser executado quando certo número de ciclos de vida é atingido ou quando é encontrado um indivíduo com um valor de adequação muito alto, indicando que se trata de uma solução boa o suficiente mesmo não sendo a melhor possível. Em uma terceira abordagem, ao invés de buscar uma solução final, o algoritmo seleciona um indivíduo a cada ciclo de vida que é considerado como “solução parcial” do problema. São os parâmetros deste indivíduo que serão usados para controlar o módulo de síntese granular no caso do **EVOGrain**. Desta forma o usuário pode acompanhar visualmente na interface e acusticamente o processo evolutivo realizado pelo algoritmo genético.

No **EVOGrain** foi implementado um Algoritmo Genético configurável, interativo e guiado por alvo, onde o usuário pode determinar um novo indivíduo (retângulo) alvo a qualquer instante desenhando-o com o mouse. Antes de iniciar a execução do algoritmo genético o usuário pode clicar no botão “Reset EVOLution” (Figura 11) e configurar o tamanho da população (número de indivíduos de cada geração), o número de sobreviventes (especifica quantos indivíduos da população atual serão mantidos para a próxima geração), taxa de mutação (porcentagem de embriões que passarão pelo processo de mutação) e o passo da evolução (quanto tempo o algoritmo espera antes de iniciar cada ciclo genético, em milissegundos). Os valores padrão para esta configuração são: tamanho da população = 1000, população de sobreviventes = 500, taxa de mutação = 3% e passo da evolução = 100 ms.

Em qualquer um dos três subespaços bidimensionais do Espaço de Parâmetros, o genótipo de um indivíduo é definido como um conjunto ordenado de quatro valores inteiros entre 0 e 1000, isto é, $R = (t_1, f_1, t_2, f_2)$. Para estabelecer um processo evolutivo, um indivíduo alvo deve ser especificado. O indivíduo alvo é denotado como $R^* = (t_1^*, f_1^*, t_2^*, f_2^*)$. A representação gráfica para um único indivíduo (correspondente ao um *fenótipo* ou à maneira como um gene se manifesta) é um retângulo com lados verticais e horizontais em um espaço bidimensional como mostrado na Figura 14.

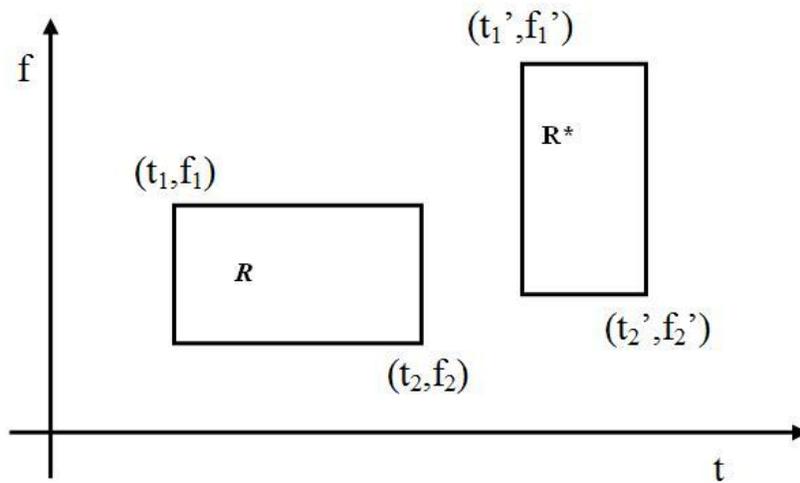


Figura 14 Dois indivíduos (retângulos), R e R^* , em um espaço cartesiano R^2

Uma população, isto é, um conjunto destes retângulos, é inicialmente construído sorteando aleatoriamente valores entre $[0-1000]$ para os quatro valores de todos os indivíduos. Agora é definida uma função de distância entre os indivíduos R e R^* (retângulos no espaço bidimensional):

$$d(R,R^*) = \max\{|t_1 - t_1'|, |f_1 - f_1'|, |t_2 - t_2'|, |f_2 - f_2'|\} \quad (3)$$

Esta fórmula é bem conhecida como *norma do máximo* na teoria dos espaços métricos (BRYANT, 1985). Esta função de distância é uma métrica muito simples e faz sentido, pois todos os parâmetros estão no intervalo entre $[0,1000]$. A seguir é definimos a função de adequação (*fitness function*) $f(R,R')$ como:

$$f(R,R^*) = \frac{1}{1+d(R,R^*)} \quad (4)$$

Observe que a função de adequação está normalizada, isto é, $0 \leq f(R,R') \leq 1$. Observe que 0 é apenas um valor teórico, pois como os subespaços bidimensionais são limitados, a distância máxima possível $d(R, R^*)$ é finita. Esta função também satisfaz o requerimento $d(R,R')=0 \Rightarrow f(R,R')=1$ (adaptabilidade total).

A adequação de cada indivíduo é avaliada e a população é dividida em sobrevivente e não sobreviventes conforme um critério muito simples: os n indivíduos com adequação mais alta são incluídos no grupo de sobreviventes, onde n é o tamanho da população de sobreviventes configurada pelo usuário. O próximo passo é aplicar os operadores de cruzamento sobre os sobreviventes para obter “filhos”, e então aplicar o operador de mutação aos “recém nascidos”. Em nosso modelo, o tamanho da população é mantido constante em todas as gerações. Isto significa que a cada ciclo genético alguns indivíduos serão descartados e novos indivíduos deverão ser criados. Claramente, o número de indivíduos que deverão ser criados é igual ao tamanho da população menos o numero de sobreviventes. Indivíduos são criados através do cruzamento dentro do grupo de sobreviventes. O operador genético de cruzamento seleciona aleatoriamente dois indivíduos dentro do grupo de sobreviventes e cria um novo individuo cujos quatro valores de seu genótipo são iguais a média dos valores dos “pais”, ou seja:

$$\text{Pai 1} = (t_1, f_1, t_2, f_2) \quad (5a)$$

$$\text{Pai 2} = (t_1', f_1', t_2', f_2') \quad (5b)$$

$$\text{Embrião} = ((t_1+t_1')/2, (f_1+f_1')/2, (t_2+t_2')/2, (f_2+f_2')/2) \quad (5c)$$

Claramente, há um número infinito de possibilidades para a função de cruzamento. Em princípio, diferentes combinações da função de cruzamento com a função de adequação e a mutação podem levar à dinâmicas bastante complexas no Espaço de Gabor Estendido (Espaço de Parâmetros). Esta nossa escolha foi motivada pela sua simplicidade matemática e por sua fácil implementação computacional.

Após o cruzamento, o operador de mutação é aplicado. Para cada individuo recém nascido, um valor real é sorteado uniformemente entre $[0,1]$. Se este valor for menor que a taxa de mutação definida pelo usuário, então a mutação é aplicada. A mutação propriamente dita escolhe um dos quatro valores do individuo e o substitui por um valor aleatório inteiro entre $[0,1000]$. Finalmente uma nova geração está completa, o processo então pára e aguarda um curto período de tempo conforme determinado pelo parâmetro *passo da evolução* antes de iniciar um novo ciclo genético.

Uma das características importantes do **EVOGrain** é que, ao invés de apresentar somente a solução final, seu *output* é um fluxo contínuo de som, em tempo real, resultante do processo evolutivo. A cada ciclo genético, logo após determinar quais indivíduos serão mantidos na próxima geração, o algoritmo seleciona um sobrevivente e usa o seu genótipo para modificar os parâmetros da síntese granular (denominamos aqui este indivíduo de “**o escolhido**”). Ao escolher um dos indivíduos em cada geração, calculada de acordo com regras pré-definidas, seus parâmetros são utilizados para controlar a síntese sonora. Assim o que se ouve é o indivíduo escolhido de cada geração calculada pelo algoritmo. De maneira muito sucinta podemos então dizer que nesta abordagem nós podemos ouvir a evolução temporal (em tempo real) da textura sonora produzida pela síntese granular.

Numa abordagem inicial, o algoritmo selecionava o melhor indivíduo da população de sobreviventes para ter seu genótipo enviado para o módulo de som. Porém, após alguns testes, verificamos que mesmo com poucos ciclos genéticos, a operação de mutação resultava num indivíduo muito semelhante ao alvo, que era então o melhor sobrevivente durante diversas gerações subseqüentes. Ao invés de se obter um processo contínuo em direção ao alvo, o resultado era um grande salto nos parâmetros da síntese, seguido de diversas gerações sem alteração nos parâmetros. O que acontece é que o sistema evolutivo, neste caso, faz com que o processo convirja para o alvo rapidamente, resultando em texturas estáticas.

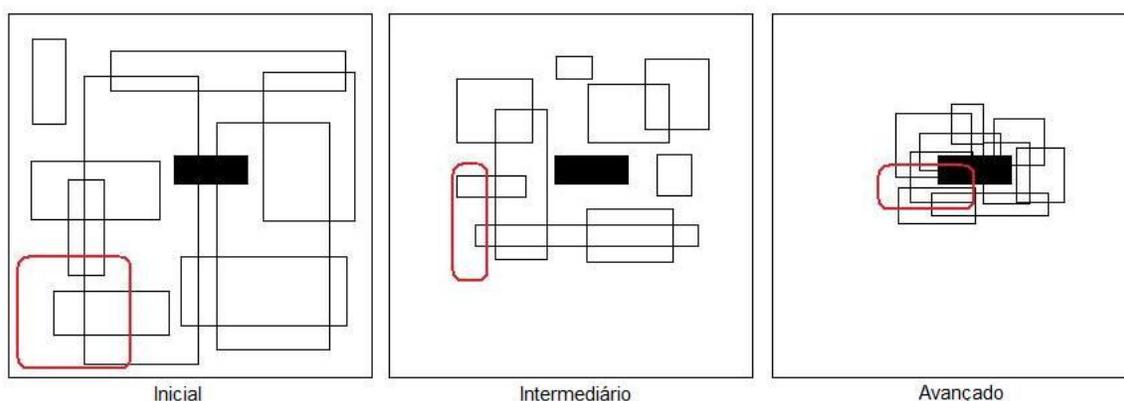


Figura 15 Estágios da evolução

Experimentamos então selecionar o indivíduo sobrevivente com a pior adequação. Isto implica numa convergência mais lenta do processo, de maneira que o usuário possa “ouvir” realmente a evolução das texturas sonoras resultantes ao longo do tempo. A Figura 15 ilustra o processo de evolução em três estágios: inicial, intermediário e avançado. O retângulo preenchido é o retângulo alvo, os retângulos vazados são os indivíduos sobreviventes de uma determinada geração e o retângulo vazado com bordas arredondadas é o indivíduo escolhido para controlar o módulo de síntese de som. Podemos notar que, a cada nova geração, a fronteira da população de sobreviventes cerca cada vez mais o indivíduo alvo até o momento em que todos os indivíduos da população de sobreviventes são muito semelhantes ao alvo.

3.3 Interface Intuitiva e Amigável

EVOGrain apresenta uma interface amigável que pode ser muito útil ao músico eletrônico e/ou eletroacústico. Conforme descrevemos na seção 3.1, a máquina de síntese granular em tempo real do **EVOGrain** pode gerar e controlar simultaneamente diversos *fluxos* de grãos sonoros. Cada um destes *fluxos* é controlado por seis pares de parâmetros que especificam a frequência, duração, distância, panorâmica, densidade e offset da síntese. Podemos dizer então que o algoritmo de síntese opera em um espaço de parâmetros hexadimensional que pode ser representado como o conjunto de três espaços bidimensionais conforme a Figura 16. O espaço maior à esquerda é o Espaço de Gabor (*Gabor Space*) original e controla a duração e a frequência de onda do interior do grão. Este espaço é também chamado de Espaço Tempo-Frequência. O espaço à direita e na parte superior denominamos simplesmente de Espacialização (*Spatialization*) e controla a distância do grão ao ouvinte e sua posição panorâmica. O espaço na parte inferior direita controla a densidade (fluxo) de grãos por segundo e o *offset* de leitura da forma de onda e o denominamos como Espaço de Fluxo (*Streaming Space*). Finalmente denominamos o produto cartesiano destes três espaços bidimensionais como **Espaço de Gabor Estendido** (*Extended Gabor Space - EGS*).

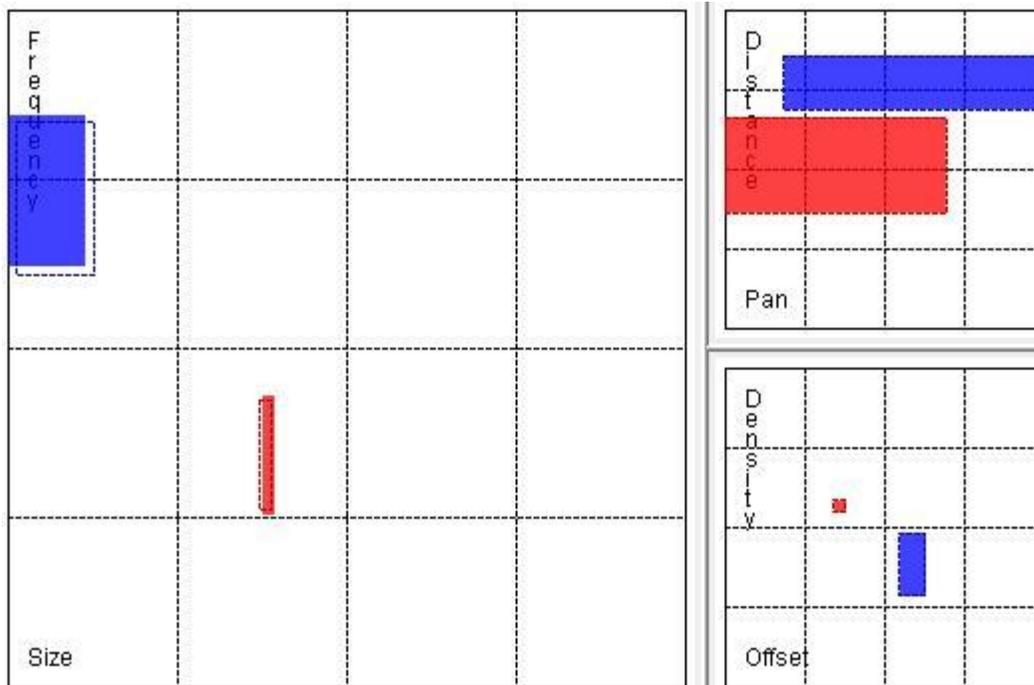


Figura 16 Interface EGS para o EVOGrain

Nesta versão do **EVOGrain** o usuário pode controlar simultaneamente dois fluxos de grãos desenhando retângulos nos três espaços do EGS com o mouse. Os retângulos podem ser de duas cores, o botão esquerdo do mouse desenha retângulos vermelhos, e o botão direito do mouse desenha retângulos azuis. Cada uma destas cores indexa um fluxo de grãos independente. Estes retângulos definem o indivíduo alvo que vai guiar o processo de evolução do algoritmo genético. Em cada espaço de parâmetros R^2 são executados dois algoritmos genéticos (um para o retângulo vermelho e outro para o retângulo azul), ou seja, o sistema executa seis algoritmos genéticos simultaneamente, ou equivalentemente, três algoritmos genéticos para cada uma das duas vozes (vermelha e azul).

Enquanto o algoritmo genético esta evoluindo, o individuo escolhido de cada geração é exibido nos três espaços da interface como retângulos pontilhados e sem preenchimento que se movimentam convergindo para o seu alvo. Sempre que um novo retângulo tracejado é desenhado em qualquer um destes três espaços, os parâmetros da Máquina de Síntese Granular são atualizados imediatamente. As coordenadas dos retângulos pontilhados, ou seja, as coordenadas do individuo escolhido de cada geração,

são lidas como parâmetros que controlam a síntese granular em tempo real. A interface torna-se interessante porque o usuário pode explorar com mais eficácia as posições dos retângulos alvo dentro do **EGS** segundo a realimentação que o binômio visualização/audição traz para ele. No processo como um todo, o usuário ouve o fluxo sonoro evoluindo enquanto os retângulos pontilhados convergem visualmente para os alvos.

Como casos limite, não somente retângulos podem ser desenhados, mas também linhas horizontais e verticais e, até mesmo, pontos. Pontos são retângulos degenerados, onde o valor inicial como para o valor final dos parâmetros controlados pelo eixo vertical e pelo eixo horizontal coincidem. Linhas horizontais resultam em um valor único para o parâmetro controlado pelo eixo vertical e uma faixa de valores para o parâmetro controlado pelo eixo horizontal. Linhas verticais se comportam de maneira análoga. Isto é muito importante caso o compositor esteja manipulando um parâmetro como densidade, onde um único valor implica em um fluxo síncrono de grãos, enquanto que um intervalo implica em uma densidade variável e, conseqüentemente, em um fluxo assíncrono de grãos. Por exemplo, ao desenhar um ponto no Espaço de Gabor para o *fluxo* azul, todos os grãos desta voz terão a mesma freqüência e a mesma duração. Em outro exemplo semelhante, se for desenhada uma linha vertical, implica que todos os grãos terão a mesma duração, mas a freqüência de cada grão será sorteada uniformemente dentro do intervalo definido pela linha.

EVOGrain permite ainda que o usuário armazene configurações de retângulos, especificamente, as coordenadas dos retângulos alvo. Isto pode ser usado como um auxílio em *performances*, bem como em composição. A qualquer momento um *preset* pode ser salvo para uso posterior com nomes como “Introdução”, “Começo da parte A”, “Seção de Improviso”, “Explosão Granular”, “Nuvens Agudas”, “Coda”, e assim por diante. Para salvar um *preset* com a configuração atual de retângulos alvo, basta clicar no botão “*save*” no canto superior direito da interface e informar um nome para o arquivo. Nesta mesma seção da interface é exibida uma listagem de *presets*. Com os botões “*add*”, “*remove*” e “*clear*” é possível incluir e excluir *presets* desta listagem. Para ativar um *preset*, o usuário deve clicar duas vezes sobre o nome exibido na lista e novos retângulos são desenhados

conforme as coordenadas do *preset*, e então o processo evolutivo é imediatamente direcionado para estes alvos.

EVOGrain também permite salvar um arquivo de áudio em formato “.au” contendo toda a síntese realizada em uma única sessão (a partir do momento do “*play*” até o “*stop*” do sistema). Para acionar esta funcionalidade o usuário deve marcar a caixa de diálogo “*save audio*” e informar um nome para o arquivo de áudio. Esta é uma funcionalidade muito importante, pois permite que o EVOGrain seja utilizado para gerar material sonoro de natureza granular para uso em composições trabalhadas em tempo diferido.

3.4 Integração com outros sistemas através de mensagens OSC

Com o objetivo de permitir a integração com outros sistemas digitais de controle de sons, um novo módulo foi integrado ao **EVOGrain**, sendo este habilitado com as funções de receber, interpretar e enviar mensagens OSC. *Open Sound Control* (OSC) é um protocolo para comunicação entre computadores, sintetizadores de som e outros dispositivos multimídia, otimizado para redes de comunicação modernas (WRIGHT, 1997). Uma mensagem OSC é composta de um nome, seguido de um ou mais parâmetros como, por exemplo, “\sinteseAditiva\frequencias 100Hz 300Hz 550Hz” ou “\MP3Player\Volume 50”. É um protocolo muito prático, de grande aceitação e que vem substituindo o uso do protocolo MIDI.

O sistema **EVOGrain** está preparado para receber mensagens OSC com as coordenadas de retângulos alvo e assim é criada uma alternativa que substitui o uso do mouse. As mensagens recebidas para o **EVOGrain** podem ser provenientes de um dispositivo de entrada (controlador gestual) ou de qualquer outro tipo de sistema capaz de enviar mensagens OSC. O Quadro 3 a seguir especifica as mensagens OSC de entrada previstas pelo sistema EVOGrain. As mensagens cujo nome se inicia com “\red” controlam a posição dos retângulos alvo vermelhos. Enquanto que as mensagens iniciadas com “\blue” determinam a posição dos retângulos azuis.

| Mensagem | Descrição |
|--|---|
| <code>\red\frequency\initial (float)</code> <code>\red\frequency\final (float)</code> <code>\red\size\initial (float)</code> <code>\red\size\final (float)</code> | Coordenadas para o retângulo alvo no Espaço de Gabor para a voz vermelha. |
| <code>\red\amplitude\initial (float)</code> <code>\red\amplitude\final (float)</code> <code>\red\pan\initial (float)</code> <code>\red\pan\final (float)</code> | Coordenadas para o retângulo alvo no quadro de Espacialização para a voz vermelha. |
| <code>\red\density\initial (float)</code> <code>\red\density\final (float)</code> <code>\red\offset\initial (float)</code> <code>\red\offset\final (float)</code> | Coordenadas para o retângulo alvo no espaço de Controle do Fluxo para a voz vermelha. |
| <code>\blue\frequency\initial (float)</code> <code>\blue\frequency\final (float)</code> <code>\blue\size\initial (float)</code> <code>\blue\size\final (float)</code> | Coordenadas para o retângulo alvo no Espaço de Gabor para a voz azul. |
| <code>\blue\amplitude\initial (float)</code> <code>\blue\amplitude\final (float)</code> <code>\blue\pan\initial (float)</code> <code>\blue\pan\final (float)</code> | Coordenadas para o retângulo alvo no quadro de Espacialização para a voz azul. |
| <code>\blue\density\initial (float)</code> <code>\blue\density\final (float)</code> <code>\blue\offset\initial (float)</code> <code>\blue\offset\final (float)</code> | Coordenadas para o retângulo alvo no espaço de Controle do Fluxo para a voz azul. |

Quadro 3 Mensagens OSC processadas pelo EVOGrain

Destaca-se que para todas estas mensagens o valor do argumento representa a coordenada do retângulo alvo e não o valor absoluto do parâmetro, isto é, ao utilizar, por exemplo, a mensagem “`\red\frequency\initial (float)`” o valor do argumento não é a frequência em Hertz, mas as coordenadas do retângulo alvo. Todas as coordenadas nos três espaços de parâmetros bi-dimensionais devem ser especificadas entre 0 e 1000. Para habilitar o sistema para o recebimento de mensagens OSC o usuário deve clicar na caixa de seleção “*Listen OSC*” e configurar a porta onde as mensagens serão recebidas.

O sistema ainda apresenta a possibilidade de enviar uma mensagem OSC de saída contendo a informação sobre os grãos de som gerados pelo módulo de síntese sonora. A mensagem contém as informações sobre o índice do fluxo de grãos (zero para voz vermelha e um para voz azul), frequência do grão em Hertz, tamanho do grão em milésimos de

segundo, amplitude do grão (entre 0 e 1), posição panorâmica do grão (valores contínuos desde -1, para canal esquerdo até 1 para o canal direito), densidade do fluxo de grãos no momento em que este grão foi criado e *offset*. Por meio desta mensagem é possível integrar o **EVOGrain** com outros sistemas de áudio, criar uma interface alternativa que reflete o estado de cada fluxo de grão, controlar parâmetros de aplicações de vídeo como velocidade da reprodução ou transição entre duas imagens e até mesmo realizar síntese de gráficos como veremos na seção 4.3. EVOGrain envia mensagem OSC (saída) com o formato: “\grain (fluxo) (frequency) (grainsize) (amplitude) (pan) (density) (offset)”.

Para ativar esta funcionalidade o usuário deve clicar na caixa de seleção “Send OSC” e configurar o endereço e porta de destino destas mensagens. Sempre que é criado um novo grão em qualquer um dos fluxos (voz vermelha ou voz azul) o sistema envia uma mensagem OSC para o destino com o formato exibido acima.

3.5 Arquitetura Final do Sistema

Nesta seção apresentamos brevemente a arquitetura do sistema **EVOGrain v.2.0** a qual é mostrada na Figura 17 com certo detalhe. Do ponto de vista do músico ou usuário, o sistema é controlado com o mouse, tanto ao desenhar retângulos alvos no EGS como ao carregar *presets* de retângulos. A realimentação ao usuário ocorre de forma visual e em tempo real através dos retângulos tracejados desenhados pelo sistema que se movem nos seus respectivos espaços bidimensionais, mostrando a sua evolução e aproximação do retângulo alvo. A realimentação também é apresentada ao usuário/compositor como um fluxo sonoro de grãos emitido pelos alto-falantes estereofônicos com latência de 500ms.

O *input* do usuário, isto é, retângulos alvos, são recebidos pela interface e passados para o Algoritmo Genético. A adaptabilidade dos indivíduos do próximo ciclo genético será calculada com base nos novos retângulos desenhados. Assim que uma nova população de sobreviventes é calculada em qualquer um dos subespaços (R^2) do Espaço de Gabor Estendido (6-dimensional), o algoritmo determina o indivíduo escolhido e envia suas coordenadas de volta à interface que então desenha os retângulos tracejados na janela. Todo novo retângulo escolhido pelo Algoritmo Genético também é enviado à Máquina de Síntese

Granular em Tempo Real na forma de parâmetros que controlam o fluxo de grãos. Os novos grãos sintetizados pelo sistema respondem imediatamente aos novos parâmetros, mas devido ao tamanho do *buffer* seu efeito só pode ser percebido após meio segundo (latência).

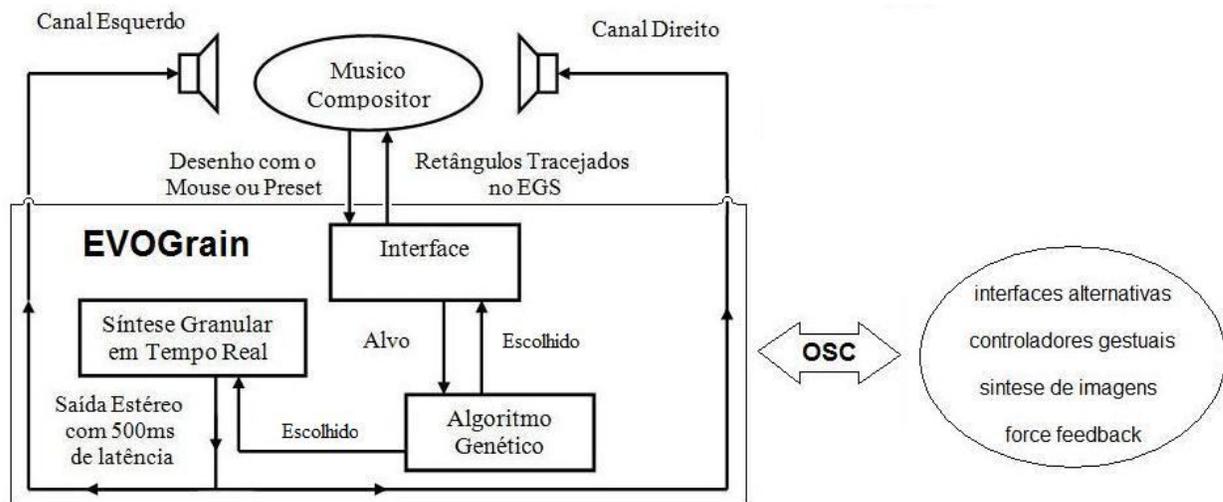


Figura 17 Arquitetura Geral do sistema EVOGrain

O **EVOGrain** também permite outras formas de interação e controle bem como diferentes modos de *realimentação* ao ser integrado com outros sistemas através de mensagens OSC. Em termos de entrada de dados (*input*) é possível, por exemplo, projetar uma interface gráfica alternativa ou conectar um controlador equipado com acelerômetros e desenhar os retângulos alvo com movimentos realizados com o dispositivo (como veremos no Capítulo 5). A mensagem de saída também proporciona as mais diversas formas de integração, como por exemplo, para alimentar controladores habilitados com *active tactile feedback* (controladores equipados com dispositivos que respondem as ações tácteis do usuário), ou para controlar a execução de um vídeo ou a síntese de imagens (conforme Seção 4.3 abaixo).

4 SISTEMAS DERIVADOS DO EVOGRAIN

Neste capítulo e no capítulo seguinte descreveremos os resultados obtidos no nosso estágio de estudos e pesquisa financiado pelo Programa de Líderes Emergentes nas Américas (*Emerging Leaders in the Americas Program – ELAP*), um programa de intercâmbio do governo canadense para estudantes e pesquisadores da América Latina. Este estágio foi realizado entre Outubro de 2009 e Abril de 2010 no *Input Devices for Music Interaction Laboratory (IDMIL)*, Music and Technology Area, Schulich School of Music, McGill University em Montreal, Canadá, sob supervisão do Prof. Dr. Marcelo M. Wanderley. O IDMIL é um laboratório que lida com projetos relacionados aos tópicos de *Human-Computer Interaction (HCI)*, design de instrumentos e interfaces para expressão musical, obtenção e análise de dados sobre movimentos, desenvolvimento de sensores e controladores gestuais.

O estudo de novos métodos de geração de som, linguagens musicais e interfaces gestuais para musica computacional têm forte impacto sobre o desenvolvimento de novos produtos, serviços e conteúdos associados à tecnologia digital. Investimentos em resultados desta área multidisciplinar estão no cerne das estratégias econômicas atuais que envolvem o entretenimento (jogos de computador, sistemas de som) e das indústrias de comunicação (telefones celulares). Este estágio propiciou a participação em atividades desenvolvidas em um dos laboratórios que está na vanguarda da pesquisa em música computacional no Canadá sendo, portanto, uma oportunidade única para obter uma experiência em primeira mão sobre os desenvolvimentos atuais desta are de pesquisa no Canadá. Tanto quanto pudemos sentir, para a McGill University e, em especial para o grupo do IDMIL, a oportunidade de trabalhar em estreita colaboração com o NICS foi muito bem-vinda. Sendo assim o projeto do estágio foi submetido com o objetivo encontrar um DMI (*Digital Musical Instrument*) adequado para controlar a composição de “auditory scenes” através de gestos mapeados para o espaço de parâmetros definido anteriormente como EGS (Espaço de Gabor Estendido).

O primeiro passo neste processo foi reformular **EVOGrain** dividindo-o em três sistemas independentes e torná-los compatíveis com os DMIs e Softwares disponíveis no

IDMIL. Assim, a Máquina de Síntese Granular em Tempo Real do EVOGrain foi convertida da linguagem Java para a linguagem C++ obtendo maior desempenho e flexibilidade, este sistema foi nomeado **GranularStreamer**. O Algoritmo Genético do EVOGrain foi transformado em um plug-in chamado **G. A. Mapper** que pode ser conectado a qualquer dispositivo de entrada e sistema de síntese de som. Buscando uma interface alternativa para o sistema, desenvolvemos também o **GranularDrawer**, um sistema que usa a linguagem de processamento gráfico Processing v1.0 para criar desenhos a partir das informações sobre os grãos de som sintetizados. No estágio atual estes desenhos dão apenas uma idéia visual do processo. No entanto há várias maneiras de tornar o GranularDrawer uma interface interativa. Por exemplo, poderíamos ativar parte de um desenho com um click de mouse e este dispararia uma mensagem OSC para o Granular Streamer emitir uma seqüência de grãos.

No próximo capítulo apresentamos a segunda parte das atividades realizadas durante o estágio. Trata-se da experimentação de DMIs disponíveis no IDMIL, entre eles o **SpaceNavigator**, um hyper-mouse tri-dimensional, o **T-Stick**, um instrumento digital em forma de bastão equipado com acelerômetros e outros sensores, e o **PC-1600x**, um controlador constituído por diversos *sliders*. Finalmente foi desenvolvido o protótipo de um novo DMI denominado **RedController** com sensores e componentes eletrônicos como acelerômetros, giroscópios, sensores de pressão e switches.

4.1 GranularStreamer

O **Granular Streamer** tem sua origem na Máquina de Síntese Granular em Tempo Real do sistema EVOGrain. Para obter melhor desempenho a linguagem de programação Java, uma linguagem interpretada, foi substituída pela linguagem C++, uma linguagem compilada. Outras modificações também foram feitas visando o aperfeiçoamento do desempenho como melhor utilização da memória do sistema, otimização do ciclo de tempo real e, principalmente, o uso de *wavetables* com 1024 pontos para todas as formas de onda e envoltórias de volume disponíveis no sistema. O **GranularStreamer** utiliza as bibliotecas

externas RTAudio, um conjunto de classes de entrada e saída de áudio, e OSCPack para a codificação de funções de recebimento e envio de mensagens OSC.

Após as modificações acima é possível sintetizar simultaneamente muitos fluxos de grãos com qualidade de áudio razoável com limitações impostas apenas pelos recursos do sistema operacional e hardware. Para executar o **GranularStreamer** o usuário deve digitar uma linha de comando passando como argumento o nome de um arquivo texto que contém as configurações do sistema. A seguir (Quadro 4) é apresentado um exemplo do arquivo de configuração seguido da explicação de cada um dos parâmetros.

| | |
|-----------------------|----------------------------------|
| OSCLISTENINGPORT | [7500] |
| OSSENDGRAIN | [TRUE] or FALSE |
| OSCDESTINATIONADDRESS | [local host] |
| OSCDESTINATIONPORT | [7900] |
| SAVEOUTPUT | [FALSE] or TRUE |
| OUTPUTFILENAME | [out put . au] |
| FLUXOS | [3] |
| ENVELOPE(0) | [GAUSSIAN;] |
| WAVE(0) | [SINE;] |
| MAXGRAINS(0) | [4] |
| ENVELOPE(1) | [EXPODEC;] |
| WAVE(1) | [SQUARE;] |
| MAXGRAINS(1) | [1] |
| ENVELOPE(2) | [TRAPEZOID;] |
| WAVE(2) | [FILE; .. \sample. wav; 1000;] |
| MAXGRAINS(2) | [10] |

Quadro 4 Arquivo de configuração do GranularStreamer

OSC LISTENING PORT especifica a porta do sistema para onde as mensagens de controle do GranularStreamer serão enviadas. OSC SEND GRAIN indica se o sistema deve enviar mensagens OSC com as informações de cada grão de som sintetizado, em caso afirmativo os parâmetros OSC DESTINATION ADDRESS e OSC DESTINATION PORT configuram o endereço de destino das mensagens. SAVE OUTPUT indica se o sistema deve salvar um arquivo de áudio, em caso afirmativo o parâmetro OUTPUT FILENAME define o nome do arquivo. STREAMS especifica o número total de fluxos que serão sintetizados. Para cada um destes fluxos outros três parâmetros devem ser especificados. ENVELOPE configura a envoltória de volume, WAVE configura a forma de onde e MAX

GRAINS indica o número máximo de grãos simultâneos ou sobrepostos (*overlapping*) que o fluxo poderá ter.

Quanto à envoltória de volume, as opções existentes no `GranularStreamer` são GAUSSIAN, EXPODEC e REXPODEC (também disponíveis no **EVOGrain**) e também TRIANGLE, TRAPEZIO e ADSR conforme Figura 18.

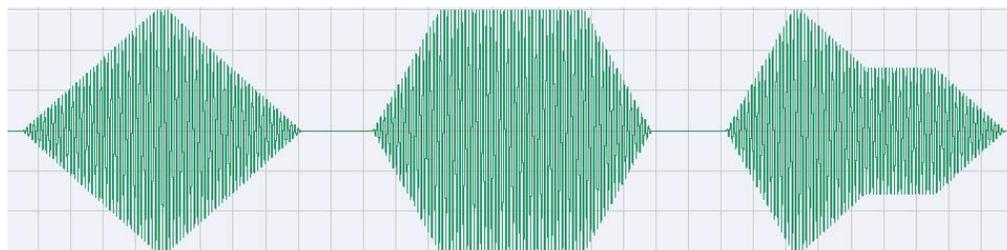


Figura 18 Envoltória triangular, trapezoidal e ADSR

Para as formas de onda são disponibilizados os tipos SINE, GLISSON, RANDOM e GRANULATION, também presentes no **EVOGrain**, e mais os tipos SQUARE (uma onda quadrada), SAWTOOTH (uma onda dente de serra) e FILE (um arquivo do tipo wave). A princípio os tipos GRANULATION e FILE podem parecer iguais, pois ambos utilizam um arquivo do tipo wave, porém a forma de leitura das amostras de áudio deste arquivo é diferente em cada um destes tipos. Quando a opção GRANULATION é utilizada, a posição inicial de leitura das amostras do arquivo é sempre incrementada de acordo com o parâmetro offset, ou seja, a posição inicial de leitura sempre avança dentro do arquivo resultando na granulação da fonte sonora. Já no caso da opção FILE, a posição inicial de leitura é sempre fixa para todos os grãos de acordo com o parâmetro offset, ou seja, a opção FILE se comporta como uma wavetable tradicional onde o parâmetro offset é a fase que determina a posição inicial de leitura.

O **GranularStreamer** não tem interface gráfica, trata-se apenas do módulo responsável pela síntese de som. Após iniciar o sistema com um arquivo de configuração conforme descrito anteriormente o controle dos parâmetros da síntese para cada fluxo é feito pelo mesmo conjunto de seis pares de parâmetros dinâmicos definidos para o sistema EVOGrain. Os valores para estes parâmetros são alterados pelo recebimento de mensagens

OSC. Desta forma o GranularStreamer pode ser facilmente integrado a qualquer dispositivo de entrada ou sistema de som capaz de enviar mensagens OSC como, por exemplo, PureData ou MAX/MSP. O Quadro 5 a seguir define as mensagens OSC previstas pelo sistema.

| Mensagem | Descrição |
|--|--|
| /raw/setStream float | Ajusta o fluxo atual para este valor. O fluxo atual aponta para o fluxo que receberá os valores das mensagens de parâmetros. |
| /raw/nextStream /raw/previousStream | Ajusta o fluxo atual para o fluxo seguinte ou anterior de maneira cíclica. |
| /raw/iFreq float /raw/fFreq float | Ajusta os valores de frequência inicial e final para o fluxo atual em Hertz |
| /raw/iGrainSize float /raw/fGrainSize float | Ajusta os valores de tamanho do grão inicial e final para o fluxo atual em milésimos de segundo |
| /raw/iAmp float /raw/fAmp float | Ajusta os valores de amplitude inicial e final para o fluxo atual entre 0.0 e 1.0 |
| /raw/iPan float /raw/fPan float | Ajusta os valores de pan inicial e final para o fluxo atual entre -1.0 (esquerda) e 1.0 (direita) |
| /raw/iDensity float /raw/fDensity float | Ajusta os valores de densidade inicial e final para o fluxo atual em grãos por segundo |
| /raw/iOffset float /raw/fOffset float | Ajusta os valores de offset inicial e final para o fluxo atual. Este parâmetro tem efeito somente para as formas de onda GRANULATION e FILE. No caso da forma de onda FILE o valor 0.0 resulta na leitura a partir do início do arquivo, o valor 0.5 a partir do meio do arquivo. Para a opção GRANULATION o valor 1.0 resulta na leitura em velocidade normal, 0.5 leitura em metade da velocidade e 2.0 no dobro da velocidade normal. |

Quadro 5 Mensagens OSC recebidas pelo GranularStreamer

Para facilitar ainda mais o controle da síntese, o **GranularStreamer** pode ser controlado por outros conjuntos de mensagens OSC equivalentes ao conjunto “raw” (Quadro 5). Por exemplo, o conjunto denominado “single” é constituído por mensagens com o formato “\single\tGrainSize float” e fazem com que o valor inicial e final de um parâmetro sejam atualizados com o mesmo valor. Existem ainda os conjuntos de mensagens que chamamos de “add” que adiciona um valor, positivo ou negativo, ao valor

atual de algum parâmetro; e o conjunto “*width*” que controla cada parâmetro por uma dupla composta de valor central e largura do intervalo.

Além destas mensagens de entrada, da mesma forma que acontece no sistema **EVOGrain**, sempre que um novo grão é criado o sistema envia uma mensagem OSC com os parâmetros do grão para um endereço de destino definido no arquivo de configuração, e desta forma é possível integrar o **GranularStreamer** com outros aplicativos de áudio ou até mesmo gráficos como visto na Seção 3.4.

4.2 Genetic Algorithm Mapper

Na segunda parte do nosso estágio na McGill University criamos um plug-in para aplicar algoritmos genéticos a um número arbitrário de parâmetros de entrada e de saída. Trata-se da conversão do módulo genético do sistema **EVOGrain**, o qual foi então chamado de **Genetic Algorithm Mapper (G. A. Mapper)** cuja interface é mostrada na Figura 19. De forma semelhante ao módulo genético do **EVOGrain**, o **G.A.Mapper** executa constantemente o processo de evolução genética e atualiza os parâmetros de saída sempre que uma nova geração é calculada. Com o uso deste plug-in é possível mapear qualquer interface ou dispositivo de entrada para um sistema de síntese sonora (seja síntese granular ou não).

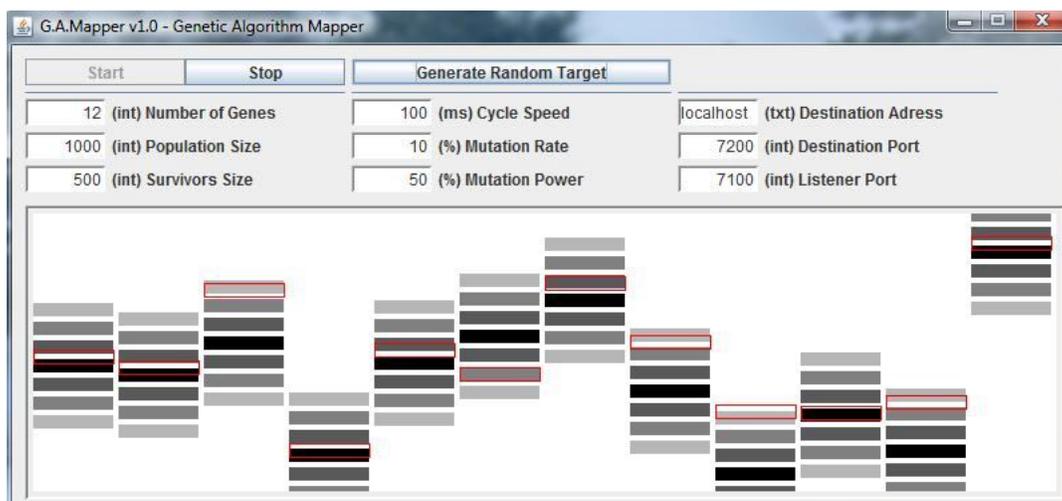


Figura 19 Genetic Algorithm Mapper

De maneira geral, o algoritmo genético executado pelo **G.A.Mapper** é muito semelhante ao algoritmo genético do **EVOGrain**. Enquanto que no **EVOGrain** o cromossomo é um vetor de 12 entradas, duas entradas (2 genes) para cada um dos 6 parâmetros do Espaço de Gabor Estendido, no **G.A.Mapper** o número de genes pode ser arbitrariamente escolhido pelo usuário no campo “number of genes” da interface. Cada indivíduo da população é definido por um cromossomo formado por uma quantidade de genes, onde cada gene corresponde a uma dimensão no espaço de parâmetros definido pelo usuário. Cada gene fornece um valor numérico de ponto flutuante (*float* ou valor decimal) entre 0 e 1. Se temos, por exemplo, um espaço de parâmetros composto de apenas de *pitch* e amplitude (duas dimensões), teremos indivíduos definidos matematicamente com um par de valores reais entre zero e um (dois genes).

No início da execução é criada uma população inicial com certo número de indivíduos com valores aleatórios entre 0 e 1 (com distribuição uniforme) para cada um de seus genes. Também é definido um indivíduo alvo cujo genótipo é determinado pelo dispositivo de entrada conectado ao **G. A. Mapper**. Cada um dos retângulos pretos na interface do **G.A.Mapper** representa um gene do indivíduo alvo e a coleção inteira dos retângulos pretos é seu cromossomo. Da mesma forma que acontece no **EVOGrain** uma função de adequação (fitness) pode ser definida em relação ao cromossomo alvo gerando um processo evolutivo.

Assim apresentamos acima o espaço de evolução do **G.A.Mapper**. Agora precisamos definir a dinâmica de evolução através de uma função de adaptação. Definimos então $D(i,t)$ como uma função que mede a distancia entre um indivíduo da população e o indivíduo alvo e $F(i,t)$ como a função de adequação. Na nossa abordagem definimos sempre uma função de adequação normalizada, isto é, $F(i,t)$ assume valores apenas no intervalo fechado $[0,1]$. Denotamos inicialmente $gene(i,g)$ o valor numérico do g -ésimo gene do i -ésimo indivíduo da população que está sendo avaliada e $gene(t,g)$ o valor numérico do g -ésimo gene do indivíduo alvo (ou target) “ t ”. Denotando também:

G = número de genes que compõe o cromossomo;

S = tamanho da população de sobreviventes;

a distância entre o i -ésimo indivíduo e o alvo “ t ” é definida por

$$D(i, t) = \sum_{g=1}^G | gene(i, g) - gene(t, g) | \quad (6)$$

e a função de adequação definimos por:

$$F(i, t) = \frac{(G - D(i, t))}{G} = 1 - \frac{D(i, t)}{G} \quad (7)$$

Esta função de adequação claramente leva em conta o tamanho arbitrário G do cromossomo. Se $D(i, t) = 0$, menor distância possível, obtemos $F(i, t) = 1$, perfeita adequação e se $D(i, t) = 1$, máxima distância possível temos $F(i, t) = 0$, a pior adequação possível.

Após avaliar a adaptabilidade de todos os indivíduos da população é formada a população dos sobreviventes com os S indivíduos com maior valor de adequação. Dentro desta população de sobreviventes, o indivíduo com a “pior” adequação é escolhido para ter seu genótipo aplicado à saída do **G.A.Mapper**. Ou seja, da mesma forma que é feito no **EVOGrain**, o pior sobrevivente é escolhido para alimentar tanto a interface gráfica (representado como retângulos vermelhos vazados), quanto os parâmetros do algoritmo de síntese. Lembramos que a escolha do “pior” sobrevivente é simplesmente para que o processo evolutivo não convirja muito rapidamente e possa fornecer um *fluxo* de sons que um ser humano possa acompanhar com certa facilidade.

O próximo passo realizado pelo algoritmo genético é gerar novos indivíduos para substituir aqueles que não foram incluídos no grupo de sobreviventes. Para cada novo indivíduo a ser gerado são sorteados dois indivíduos da população de sobreviventes com distribuição uniforme para serem seus pais. Um novo indivíduo é gerado a partir destes dois pais onde cada gene é copiado aleatoriamente de um pai ou de outro com igual probabilidade (distribuição homogênea).

A operação de mutação é aplicada da seguinte maneira. Definimos primeira mente uma Taxa de Mutação que irá indicar o percentual de indivíduos sobre os quais a mutação

irá incidir. Um segundo parâmetro denominado *Potência de Mutação* (Mutation Power) indica o percentual de genes que serão aleatoriamente substituídos, sempre com valores no intervalo [0,1].

Para realizar a verificação de um indivíduo, um número aleatório entre [0,1] é atribuído automaticamente a ele. Se este número for menor ou igual à *Taxa de Mutação* (Mutation Rate, na Interface Gráfica) o *Operador de Mutação* será aplicado a ele, caso o contrário será incorporado à próxima geração sem nenhuma modificação. Para indivíduo que será mutado o sistema escolhe aleatoriamente (distribuição uniforme) um subconjunto de seus genes. O tamanho deste subconjunto é dado pelo parâmetro MutationPower. Cada gene deste subconjunto recebe um novo valor, também aleatório, no intervalo [0,1].

Finalmente está calculada uma nova geração de indivíduos. O algoritmo aguarda um curto período de tempo configurado pelo usuário, denominado CycleSpeed na interface do G.A.Mapper e repete os passos para calcular a geração seguinte. Assim como no **EVOGrain**, o que se deseja obter aqui não é uma solução final mas um processo de evolução em direção a determinado alvo (Target-Driven Genetic Algorithm). Como mencionado acima, durante o processo de evolução os parâmetros da síntese são atualizados permitindo que o usuário ouça a evolução do som no decorrer do tempo.

Na Interface Gráfica, o **G. A. Mapper** apresenta o cromossomo dos indivíduos como uma seqüência horizontal de genes. Apenas o *indivíduo alvo* e o *indivíduo escolhido* são exibidos na interface, sendo que os genes do indivíduo alvo são representados pelos retângulos pretos sólidos e os do indivíduo escolhido são retângulos vermelhos vazados, que aparecem somente quando o processo evolutivo é ativado (play). Nesta versão, os retângulos acinzentados são meramente um enfeite gráfico, mas pensamos que numa versão posterior eles possam representar subestruturas do genes. Tanto o *indivíduo alvo* como o *indivíduo escolhido* são atualizados na interface em tempo real, o que permite que a *realimentação* gráfico seja utilizado para auxiliar o controle do sistema. Na Figura 20 vemos dois exemplos da representação de indivíduos típicos na Interface Gráfica. Na parte superior vemos um indivíduo formado por quarenta genes, e na parte inferior o sistema é executado com um cromossomo comportando oitenta genes.

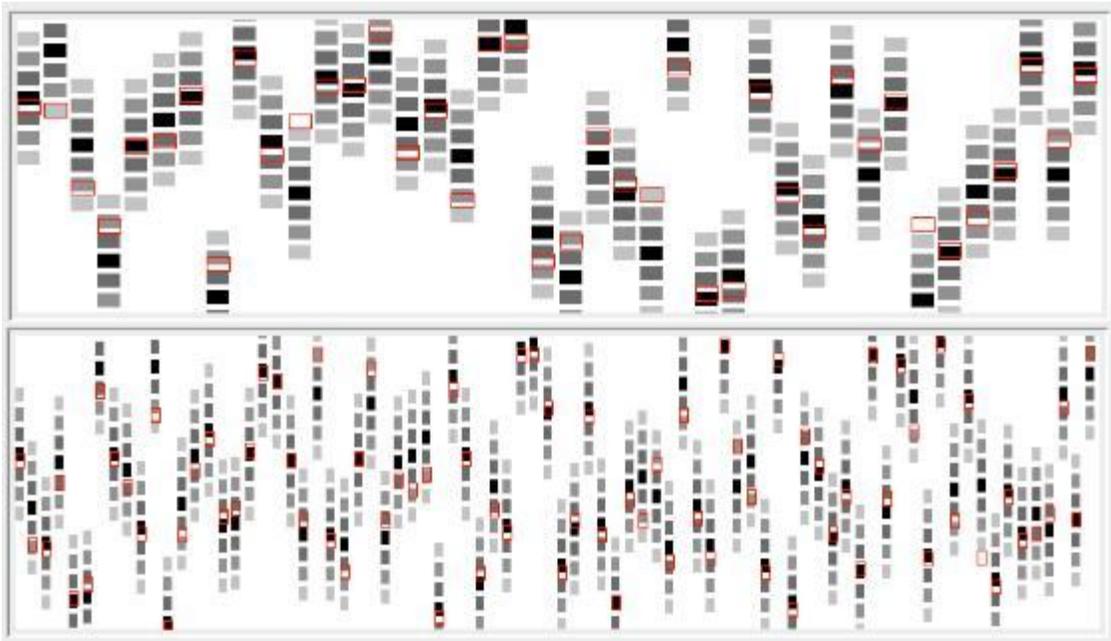


Figura 20 Dois cromossomos: 40 genes (em cima) e 80 genes (em baixo)

Por não se tratar de uma aplicação que requer grande poder de processamento em tempo real foi escolhida a linguagem de programação JAVA e assim é obtida a portabilidade que permite executar o **Genetic Algorithm Mapper** em qualquer plataforma (Windows, MacOS ou Linux). Antes de iniciar a execução do sistema o usuário deve configurar alguns parâmetros de comunicação como porta de origem e endereço e porta de destino das mensagens OSC, bem como parâmetros de configuração do algoritmo genético conforme definições a seguir:

-“*Number of genes*”: determina o número de genes que vai formar o cromossomo de cada indivíduo da população. É também o número de entradas que o plug-in vai receber por meio de mensagens OSC e o mesmo número de saídas;

-“*Population size*”: a quantidade de indivíduos em cada geração calculada pelo algoritmo genético;

-“*Survivor Size*”: a quantidade de indivíduos da geração atual que serão mantidos na próxima geração. Os demais indivíduos perecerão sendo substituídos por novos indivíduos;

-“*Cycle speed*”: é o tempo, medido em milésimos de segundo, que o sistema aguarda para o cálculo de cada geração;

-“*Mutation rate*”: é o percentual de indivíduos em que a operação de mutação será aplicada. Esta operação é aplicada somente para os indivíduos recém nascidos, ou seja, aqueles que estão substituindo indivíduos da geração anterior que não sobreviveram até a geração atual;

-“*Mutation power*”: quando um indivíduo recém nascido é selecionado para passar pelo processo de mutação, este índice indica a quantidade de genes de seu cromossomo que será alterada.

| Mensagem | Descrição |
|---|--|
| \inGeneNumber\0 (float) \inGeneNumber\1 (float) ... \inGeneNumber\n (float) | Mensagem de entrada para o valor de cada gene do indivíduo alvo. O valor do argumento deve estar entre zero e hum. |
| \outGeneNumber\0 (float) \outGeneNumber\1 (float) ... \outGeneNumber\n (float) | Mensagem de saída para o valor de cada gene do indivíduo escolhido da população de sobreviventes. O valor do argumento vai estar entre zero e hum. |

Quadro 6 Mensagens OSC de entrada e saída do G.A.Mapper

Conforme citado anteriormente, o **G.A.Mapper** pode ser utilizado com um número arbitrário de entradas e saídas para conectar um dispositivo de entrada a um algoritmo de síntese. Toda comunicação com o sistema de origem (por exemplo, um controlador) e com o sistema de destino é feita por meio de mensagens OSC (definidas na Quadro 6) que definem o valor de cada gene do indivíduo alvo e do indivíduo escolhido.

4.3 GranularDrawer

Inicialmente esta aplicação foi desenvolvida para facilitar a visualização do funcionamento do **Granular Streamer** permitindo a correção de erros no sistema já que o mesmo não possui interface gráfica nem *realimentação*, porém, após um breve uso, mostrou-se como uma idéia simples e poderosa para integração de síntese granular de

áudio, gráficos, vídeos e imagens. Nesta dissertação vamos no restringir a descrever o funcionamento de umas das aplicações do aplicativo **GranularDrawer**, que será utilizada na demonstração de diversos exemplos de síntese granular. Claramente há inúmeras possibilidades de integração do GranularDrawer com outros softwares através da troca de mensagens OSC.

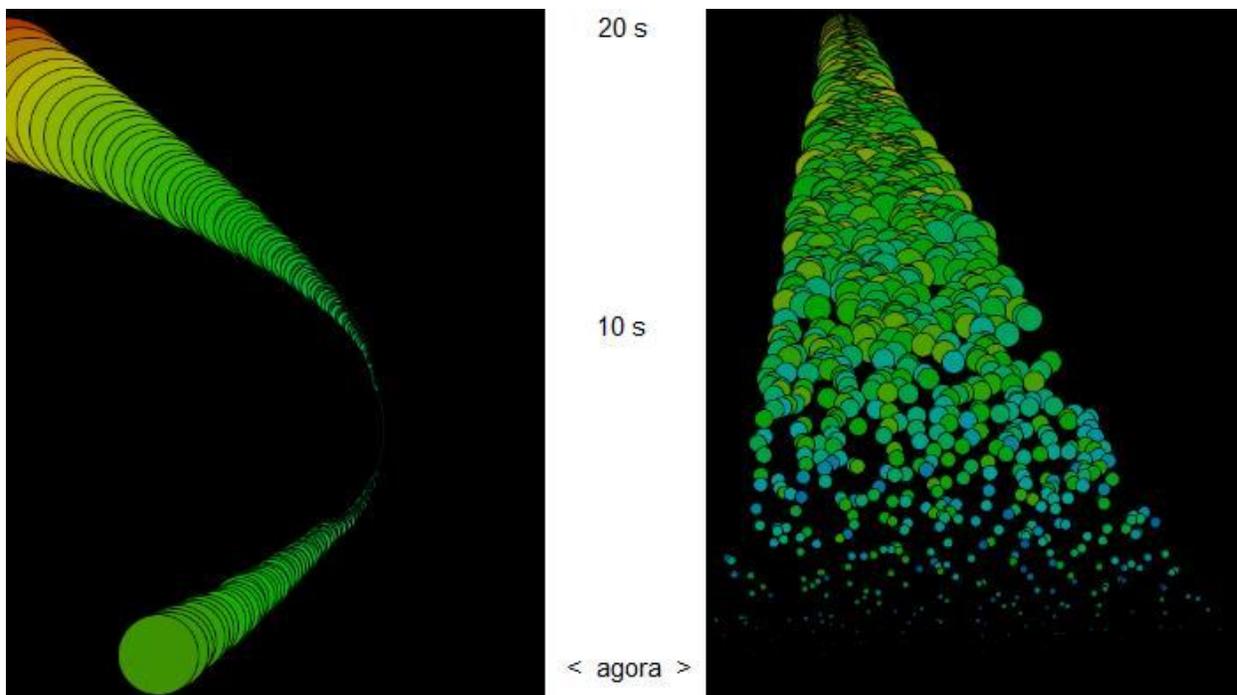


Figura 21 Dois exemplos de desenhos feitos pelo GranularDrawer

Em termos técnicos o **GranularDrawer** é uma aplicação gráfica que utiliza a linguagem *Processing 1.0* para desenhar objetos na tela do computador. Ao integrar o *Processing 1.0* com o *OSCPack*, uma biblioteca Java, foi possível receber mensagens OSC para controlar o processo de criação dos gráficos. Desta forma foi desenvolvida esta aplicação que recebe as mensagens de saída do **GranularStreamer** com os dados de cada grão criado pela síntese de áudio e desenha círculos de diferentes tamanhos, cores e posições na parte inferior da janela, que são então deslocadas para cima com o passar do tempo (*scroll*). A Figura 21 apresenta dois exemplos de imagens geradas por este sistema.

Para cada grão sintetizado é desenhado um círculo o qual é desenhado de acordo com o mapeamento de parâmetros descrito a seguir:

-Cor: a cor de cada círculo é mapeada a partir da frequência da forma de onda de cada grão, sendo que baixas frequências (de 20 Hz a 200 Hz) são representadas pelas cores do vermelho ao amarelo, frequências médias (de 200 Hz a 4 KHz) vão do amarelo ao azul, e frequência altas (de 4 KHz a 10 KHz) vão do azul para o roxo;

-Brilho: além de utilizar as cores para representar as frequências, o brilho de cada círculo também é alterado de acordo a amplitude de cada grão. Formalmente o valor numérico do brilho é dado pela raiz quadrada da amplitude. Este mapeamento se mostrou satisfatório do ponto de vista perceptual. Claramente outras possibilidades de mapeamento podem ser propostas. Assim, grãos com cuja forma de onda tem uma amplitude maior ficam com as cores mais vivas, enquanto grãos com forma de onda de menor amplitude são mais apagados tendendo ao preto;

-Tamanho: o tamanho de cada círculo tem relação direta com o tamanho de cada grão de som. O diâmetro medido em pixel corresponde à metade do tamanho do grão em milésimos de segundo, com um mínimo de 2 pixels e máximo de 200 pixel. Claramente, esta simples relação linear pode ser modificada com funções mais complexas;

-Posição Horizontal: a posição horizontal do centro de cada círculo é determina pela posição panorâmica (esquerda e direita de cada grão). Novamente na escala do Pan usamos um mapeamento linear;

-Quantidade: a densidade do fluxo de grãos é mapeado na quantidade círculos que são gerados por segundo na interface gráfica. Assim a densidade do fluxo também deixa dinamicamente a sua marca no desenho gerado. Devemos observar aqui que quando a quantidade de grãos por segundo ultrapassa as capacidades gráficas da máquina resulta que os grãos excedentes sejam ignorados neste mapeamento, isto é temos algo como um efeito de saturação.

O **GranularDrawer** permite visualizar o resultado gráfico da alteração nos valores dos parâmetros que controlam os processos de transformação do espectro sonoro num intervalo de tempo que, no nosso caso, foi fixado em 20 segundos. Por exemplo, na parte

esquerda da Figura 21 exibida anteriormente podemos visualizar um processo com 20 segundos de duração que se inicia com uma baixa densidade de grãos de tamanho grande posicionados no canal esquerdo. Conforme o fluxo se torna mais denso o tamanho dos grãos diminui e o posicionamento fica mais centralizado. Neste exemplo, o processo tem continuidade levando os parâmetros aos seus estados iniciais.

Na mesma figura, do lado direito, é exibido o exemplo de um processo, também com 20 segundos de duração onde é formada uma nuvem de grãos. Os grãos, que inicialmente são emitidos em uma única posição panorâmica, passam a ser espalhados cada vez mais para as laterais. À medida que a densidade aumenta, os grãos tornam-se menores e largura de banda é deslocada de uma região de frequências médias (círculos amarelos e verdes) para frequências mais agudas (círculos verdes e azuis).

5 CONTROLADORES GESTUAIS E INSTRUMENTOS MÚSICAIS DIGITAIS

Neste capítulo são introduzidos alguns conceitos sobre controladores gestuais e DMIs fundamentais para a compreensão das seções subsequentes deste capítulo onde são descritos experimentos nos quais foram usados controladores gestuais para síntese granular. Descrevemos também o desenvolvimento de um novo controlador de nossa autoria.

Nos primeiros instrumentos musicais inventados, suas interfaces e mecanismos de produção sonora estão presentes fisicamente nos próprios instrumentos de forma integrada e inseparável. Nestes instrumentos tradicionais, sejam de cordas, sopro ou percussão, são os próprios instrumentos que, ao serem excitados, produzem seus respectivos sons característicos. Porém, com o avanço da tecnologia ocorre também a possibilidade de construir instrumentos nos quais a interface se distancia cada vez mais dos mecanismos de produção do som. Órgãos antigos são os primeiros instrumentos onde a produção de som e a interface apresentam independência, permitindo que o instrumentista manipule chaves que alteram a extensão ou o timbre do instrumento (MALLOCH, 2008).

Com a evolução da música computacional e das ciências da computação a síntese de som em tempo real tornou-se uma realidade com, inclusive, computadores de custo razoavelmente acessível. Hoje é comum a apresentação musical com síntese em tempo real. Por outro lado, a pesquisa em sensores eletrônicos para a captação de movimentos humanos também se encontra hoje em um estágio avançado. Ao unir estas duas áreas de pesquisa tornou-se possível a invenção de novos instrumentos musicais criando inúmeras possibilidades tanto na síntese de som quanto nas formas de se interagir com um instrumento musical (WANDERLEY, 2004).

Neste momento é necessário definir os conceitos de controladores gestuais e de instrumentos musicais digitais (DMIs). Um *Controlador Gestual* é um conjunto de sensores montados sobre um dispositivo eletrônico que apenas capta os movimentos realizados por um ou mais usuários e os converte em sinais que podem então ser processados pelo computador e usados como input em vários aplicativos. Um Instrumento Musical Digital é um sistema constituído de três partes: o controlador gestual, que é um tipo de interface física com o usuário, o algoritmo de síntese sonora e um conjunto de mapeamentos onde

são feitas as ligações entre os sinais disponibilizados pelo controlador com os parâmetros de entrada de um algoritmo, que no nosso caso é um algoritmo de síntese granular.

Em DMIs, como não existe uma relação física entre o gesto do instrumentista e os mecanismos de produção do som, é necessário que o instrumentista, compositor ou projetista faça as ligações, ou seja, mapeamentos entre o controlador gestual e algoritmo de síntese ou qualquer outro aplicativo musical, mais adequados para sua obra ou performance. Se por um lado isso possibilita que um mesmo controlador gestual seja utilizado de diversas maneiras resultando em uma grande flexibilidade no projeto de DMIs, por outro lado a tarefa de estabelecer as ligações pode se tornar trabalhosa e desencorajadora (RYAN, 1991), pois resultados satisfatórios são difíceis de conseguir.

O sucesso na concepção de um DMI depende muito da capacidade de estabelecer conexões adequadas com o conjunto de mapeamentos (também denominado *camada de mapeamento*). Para realizar tal tarefa devem ser levadas em consideração algumas particularidades no uso de dispositivos de entrada para controle em tempo real de sistemas multi-paramétricos (HUNT, 2000), como é o caso de DMIs:

- Não existe uma ordem fixa no diálogo entre o computador e o usuário;
- Não existe um conjunto limitado de opções (como opções em um menu), mas uma série de controles contínuos;
- Há uma resposta imediata aos movimentos do usuário;
- O mecanismo de controle é um dispositivo físico e multi-paramétrico que o usuário precisa de um tempo de aprendizado até que suas ações se tornem automáticas;
- A prática melhora a habilidade do usuário com o dispositivo;
- Uma vez que o usuário se torne familiar com o dispositivo, ele é capaz de realizar outras ações concomitantes como acontece, por exemplo, ao dirigir e conversar ao mesmo tempo.

A classificação tradicional de instrumentos acústicos, seja pelo som produzido (idiofones, membranofones, cordofones e aerofones), pelo timbre (metais, madeiras, cordas e outros) ou ainda pelo tipo de interface (teclados, percussão, paleta) se mostra incompleta

para classificar a grande variedade de gestos humanos e os diferentes contextos de interação que podem ocorrer com o uso de dispositivos de entrada digitais. A grande maioria dos DMIs existentes podem ser classificados em: *imitações de instrumento*, quando o DMI procura reproduzir todos os detalhes mecânicos e acústicos de um instrumento conhecido; *inspirados em instrumentos* nos quais embora o formato do DMI seja parecido com um instrumento acústico, o mesmo não tem objetivo de ser uma cópia digital e nem de reproduzir o seu som; *instrumentos estendidos*, uma fusão de instrumentos acústicos com sensores eletrônicos, e finalmente, *controladores alternativos*, cujo projeto não tenta reproduzir nenhum instrumento acústico (WANDERLEY, 2004).



Figura 22 T. Opie e o Poseidon (esquerda) e S. Goto e o SuperPolm (direita)

A Figura 22 mostra dois DMIs que realizam síntese granular. Do lado esquerdo vemos o Poseidon, um controlador alternativo projetado para controlar um sistema de síntese granular onde o usuário atua movimentando dez *sliders* que controlam os parâmetros do algoritmo de síntese granular (OPIE, 2003). Em sua primeira abordagem, Timothy T. Opie ligou cada um dos *sliders* diretamente a um único parâmetro da síntese, a saber: duração dos grãos e variação, densidade do fluxo e variação, frequência dos grãos e variação, amplitude e variação, posição panorâmica e variação. Na mesma figura, do lado direito, vemos Suguru Goto tocando o SuperPolm, um DMI inspirado em um violino

controlando síntese granular (GOTO, 1999). O instrumento é equipado com um sensor de toque no lugar do braço do violino, um painel com oito botões próximo ao queixo do instrumentista e um arco especial (sem crina) que possibilita captar informações sobre pressão, duração e velocidade das arcadas.

As próximas seções descrevem experimentos realizados com três controladores gestuais diferentes e os resultados obtidos. Cada um destes dispositivos foi mapeado de múltiplas maneiras com os sistemas produzidos no decorrer desta pesquisa de mestrado. O capítulo termina com a descrição do projeto e construção de um novo controlador gestual e algumas possíveis aplicações em síntese granular.

5.1 Experimentos com SpaceNavigator

Este experimento foi realizado conectando o controlador SpaceNavigator (Figura 23) ao GranularStreamer . O SpaceNavigator é um hyper-mouse tri-dimensional com uma espécie de manche e dois botões. Sua base fica fixa sobre a mesa e o manche permite seis graus de liberdade (seis movimentos primitivos do usuário), a saber, translação nos planos x, y, e z, e rotação nos eixos x, y e z.



Figura 23 O super mouse 3D Space Navigator

Nosso primeiro experimento foi conectar cada um dos seis movimentos do controlador a cada par de parâmetros do GranularStreamer, mas com o valor inicial e valor final sendo iguais, ou seja o intervalo do parâmetro tem tamanho zero, com um único valor. Este é o mais simples mapeamento que pode ser feito com este controlador. Foram experimentadas diversas permutações com base na conexão exibida no Quadro 7 a seguir:

| | |
|--------------|-----------------|
| Translação x | Frequência |
| Translação y | Tamanho do grão |
| Translação z | Amplitude |
| Rotação x | Densidade |
| Rotação y | Pan |
| Rotação z | Offset |

Quadro 7 O Primeiro mapeamento realizado

Em seguida foram realizados novos mapeamentos fixando-se os valores de alguns parâmetros do GranularStreamer como, por exemplo, duração dos grãos ou amplitude, e com isso obtivemos um maior controle sobre os demais parâmetros em menor número, permitindo o uso de intervalos não-nulos para os parâmetros. Desta forma foram obtidas estruturas granulares mais interessantes. Destaca-se o fato de que, ao se relacionar mais de um parâmetro da síntese a um mesmo movimento do hyper-mouse, estamos fazendo algo análogo ao que acontece na parametrização das chamadas *grainlets*, tipo particular de grão onde os valores dos seus parâmetros definidores são interdependentes. Um exemplo de um mapeamento do tipo descrito acima é mostrado no Quadro 8 abaixo.

| | |
|--------------|--|
| Translação x | Frequência central |
| Translação y | Banda de frequência |
| Translação z | Densidade |
| Rotação x | Amplitude central |
| Rotação y | Varição da amplitude |
| Rotação z | Overlap: tamanho do grão em função da densidade do fluxo |

Quadro 8 Conexões mais complexas

Estudamos também as limitações deste tipo de sistema. Com estes primeiros experimentos comprovamos que a camada de mapeamento é realmente um ponto crucial do sistema. As conexões foram efetuadas com base em uma estratégia ingênua de mapeamento, que após um curto período de tempo adquire um aspecto lúdico e não convida ao aprendizado contínuo. Wessel e Wright estabelecem que um DMI deve proporcionar um aprendizado inicial simples, mas ao mesmo tempo deve permitir o desenvolvimento de uma expressividade musical a qual é importante tanto do ponto de vista da criação como da performance. Para tanto, é proposto o uso de metáforas explícitas que possam ser incorporadas facilmente ao gesto e este, por sua vez, mapeado para máquina de síntese (WESSEL, 2002). Além destas limitações com o mapeamento, o dispositivo também apresenta algumas outras dificuldades, ao menos da forma como foi utilizado, como por exemplo, a de manipular um dos movimentos sem causar consideráveis oscilações em outros, uma espécie de instabilidade paramétrica.

5.2 Experimentos com T-Stick

O *T-Stick* é um controlador gestual desenvolvido por J. Malloch no IDMIL, Área de Música e Tecnologia da McGill University (MALLOCH, 2008). É um controlador equipado com acelerômetros, sensores de pressão, de toque e alguns outros sensores opcionais (Figura 24). O conceito do T-Stick é construir um instrumento capaz de sentir tudo que está acontecendo em seu corpo como, por exemplo, quão forte o instrumento está sendo tocado, quanto de sua área está coberta, se está sendo chacoalhado, torcido, inclinado, etc.

O T-Stick oferece uma vasta gama de possibilidades para controlar o GranularStreamer . Nos mapeamentos experimentados ele foi utilizado tanto como um instrumento para ser chacoalhado, quanto como um bastão para disparar processos de sons granulares. No primeiro experimento, o parâmetro *shake* do bastão é mapeado simultaneamente na amplitude dos grãos e na densidade de um fluxo assíncrono. Os grãos são espalhados por uma banda de frequência que se desloca para o grave e para o agudo conforme a inclinação horizontal do bastão.



Figura 24 O T-Stick

Na segunda abordagem o bastão tem a função de disparar processos sonoros em analogia a um regente de orquestra. Controladores alternativos, como o T-Stick, possibilitam o uso de um vocabulário gestual diferente daquele da manipulação tradicional de instrumentos musicais. Desta forma, podem propiciar um uso inicial mais simplificado sendo porém necessário algum tempo de aprendizagem para dominar este novo vocabulário (WAISWISZ, 1985). Assim, foi definido um gesto a ser realizado com o T-Stick como segue. Segurando o T-Stick com as duas mãos em uma extremidade, simular um golpe em direção ao chão. O movimento pode ser feito em uma linha reta para baixo, com um leve movimento circular no final ou com um movimento circular completo no final. Através dos dados do acelerômetro de três eixos equipado no controlador são obtidos dois parâmetros que são usados para acionar os processos granulares. *Hit Power*, é a intensidade do golpe que determina a duração total do processo, e *Hit Spin* é a quantidade de movimento circular realizado no final do movimento que seleciona um tipo de processo diferente. Posteriormente, veremos como o novo controlador projetado, o RedController, foi utilizado de maneira semelhante para disparar processos sonoros.

5.3 Experimentos com PC-1600x

O PC-1600x é um controlador MIDI, comercialmente difundido, composto principalmente por um painel de *sliders*. É essencialmente muito similar ao Poseidon, o instrumento desenvolvido por Tim T. Opie, um DMI descrito no início deste capítulo, pois oferece os mesmos tipos de controle. Para o PC-1600x utilizamos uma arquitetura diferente daquela dos experimentos anteriores (Figura 25), aqui o controlador foi conectado ao **G.A.Mapper** o plug-in que teve origem no módulo genético do **EVOGrain**. Após processar cada ciclo genético, o G.A.Mapper atualiza os parâmetros de um fluxo de grãos do **GranularStreamer**.

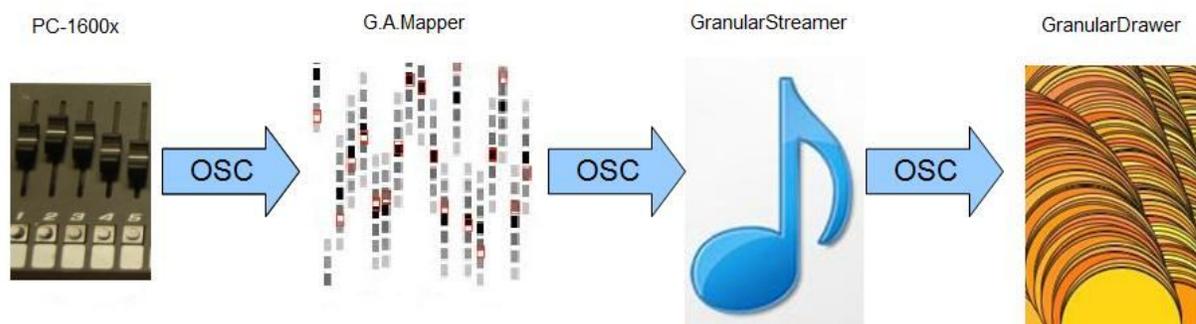


Figura 25 Arquitetura geral de um DMI com o mapeamento feito pelo G.A.Mapper

Também foram realizados testes utilizando o G.A.Mapper e os controladores apresentados anteriormente, porém eles se mostraram um tanto inadequados nesta interação. O motivo é que os valores dos parâmetros de saída destes dispositivos estão sempre variando impedindo o estabelecimento de um alvo fixo pelo G.A Mapper. Estas limitações podem ser contornadas restringindo-se os movimentos ou criando-se novos vocabulários gestuais compatíveis com os processos genéticos convergentes do G.A.Mapper. Diferentemente, o PC-1600x, por seu caráter estático, favorece o estabelecimento de um alvo fixo de forma semelhante a que é feita no EVOGrain, e permite então que o algoritmo genético evolua durante um período de tempo razoável em direção ao alvo.

5.4 O protótipo RedController

Motivado pelos experimentos descritos nas seções anteriores foi projetado um novo controlador gestual denominado **RedController**. Este dispositivo deveria ser flexível para poder ser utilizado na criação de DMIs inspirados em instrumentos acústicos, e também como um controlador alternativo. O objetivo foi adquirir os conhecimentos necessários para montar controladores com sensores capazes de reproduzir a forma de interação dos diversos experimentos mencionados anteriormente. Por questão de completude, e para se ter uma idéia da complexidade do instrumento, listamos abaixo o equipamento necessário para montagem de um protótipo:

- uma placa controladora Arduino Mini;
- uma porta de comunicação serial USB;
- um acelerômetro de três eixos;
- um giroscópio de dois eixos;
- três botões sensíveis a pressão com cerca de 1 cm de diâmetro;
- quatro botões convencionais;
- placa de montagem, fios e resistores;
- cola quente, fita adesiva, fita isolante, pequenos pedaços de espuma;
- recipiente de plástico transparente com extremidades de borracha vermelha;
- cabo USB.

A placa controladora *Arduino* implementa o conceito de *hardware* aberto, um microprocessador programável que disponibiliza diversos pinos de entrada ou saída onde podem ser conectados circuitos elétricos ou eletrônicos. Os botões convencionais atuam como interruptores em um circuito elétrico que pode assumir dois estados, ligado ou desligado, isto é, controlam apenas um estado binário (pressionado, não pressionado) e são conectados a um pino de entrada digital da placa *Arduino*. Os demais sensores são conectados a pinos analógicos (entradas ligadas a conversores analógico-digitais). Em geral os sensores são circuitos elétricos onde um dos componentes é uma resistência variável e

cuja voltagem final está compreendida dentro de um intervalo previsto. A Figura 26 exibe o circuito do **RedController** montado sobre uma placa de protótipo, e a Figura 27 exibe o dispositivo completo montado em seu invólucro plástico.

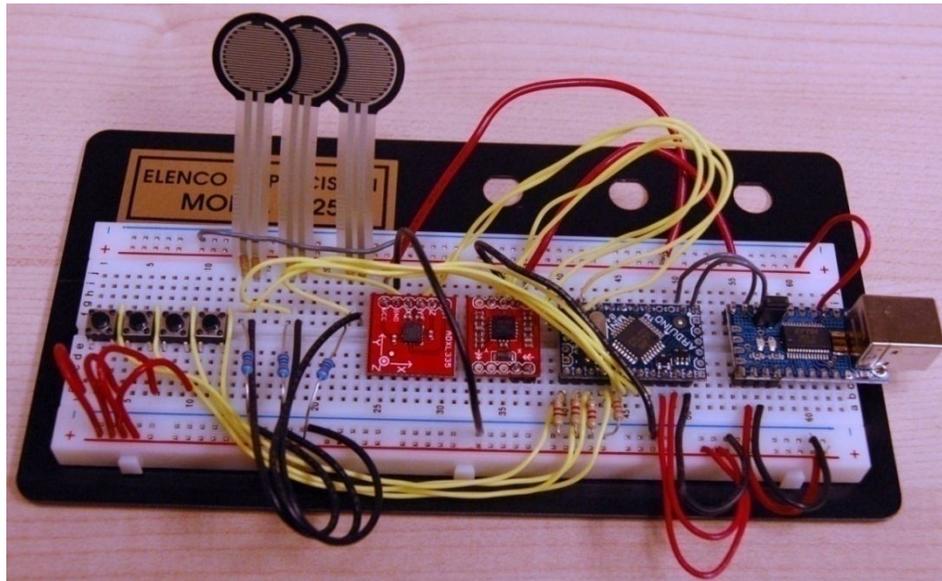


Figura 26 Circuito do RedController montado sobre uma placa de testes

Os sensores de pressão, para apresentar um exemplo, são construídos basicamente por dois filmes feitos de material condutor, separados por um filme de material não condutor e de espessura reduzida. Ao pressionar o sensor, os dois filmes feitos de material condutor se aproximam causando a diminuição da resistência total do material. Já os sensores de aceleração e de velocidade angular foram adquiridos como circuitos eletrônicos em *hardware aberto*. Existem diversas soluções na microeletrônica para a implementação de acelerômetros e giroscópios, bem como outros sensores de mesma natureza (<http://www.sensorwiki.org>, último acesso em 05/07/2010). Esses circuitos eletrônicos disponibilizam pinos de saída cujo valor da voltagem oscila conforme a aceleração ou velocidade angular em torno de um eixo específico. A placa controladora é facilmente conectada à placa que contém a porta USB através de dois pinos digitais específicos chamados *tx* e *rx*.

Para ler os sensores conectados à placa controladora é necessário escrever um pequeno programa que é carregado e executado diretamente na placa controladora chamado de *firmware*. Neste programa são escritas instruções para a leitura de cada um dos sensores conectados aos pinos da placa controladora. Após obter os valores, os parâmetros de saída são enviados para o *driver* pela porta serial USB em uma seqüência de *bytes*. O *driver* é um programa executado do outro lado da conexão USB, ou seja, do lado do computador, e é responsável por solicitar e receber informações do *firmware* com uma taxa de amostragem em torno de 200 Hz, uma frequência bem adequada para DMIs (WANDERLEY, 2004). O loop do *firmware* é executado diversas vezes para cada leitura solicitada pelo do *driver* com esta taxa de amostragem. O *driver* também decodifica os valores dos sensores e envia os resultados para um endereço de destino em mensagens OSC.



Figura 27 RedController, o primeiro protótipo

As informações enviadas pelo *driver* são divididas em três camadas. A primeira camada de parâmetros fornece os valores dos sensores no exato momento da leitura, ou seja, são parâmetros simples e onde o único tratamento feito é a normalização dos valores no intervalo de zero e mil. A segunda camada de parâmetros fornece novos valores que são calculados com base nos parâmetros da primeira camada, ou que apresentem algum outro tratamento. A terceira camada é semelhante à segunda, porém os cálculos são efetuados no *driver* ao invés de estarem no *firmware*. A tabela a seguir exhibe a relação dos parâmetros de saída do **RedController**.

| Primeira camada de parâmetros | Descrição |
|---|---|
| Aceleração X Aceleração Y Aceleração Z | Valores dos sensores do acelerômetro calibrados de 0 a 1000. Na ausência de aceleração o valor é em torno de 500. |
| Velocidade Angular X Velocidade Angular Y | Valores dos sensores do giroscópio, com mesmo tratamento do acelerômetro. |
| Sensor de Pressão 1 Sensor de Pressão 2 Sensor de Pressão 3 | Valores dos três botões equipados com sensor de pressão. O valor aumenta de zero a mil de acordo com a pressão. |
| Botão 1 Botão 2 Botão 3 Botão 4 | Valor dos switches, 0 para aberto, e 1 para pressionado. |
| Segunda Camada de Parâmetros | Descrição |
| Delta Aceleração X Delta Aceleração Y Delta Aceleração Z | Diferença absoluta entre o valor atual e o valor imediatamente anterior da aceleração em cada eixo, acompanhada de um sinal que indica se o delta é positivo ou negativo. |
| Total Delta Aceleração | Soma do delta absoluto de aceleração nos três eixos. Este parâmetro recebe o nome de <i>shake</i> . |
| Jab | É o valor do total delta aceleração quando o mesmo é mais alto que um determinado limiar, indicando que houve um movimento brusco com o objeto, é caracterizado um <i>jab</i> . Outro <i>jab</i> não pode ser detectado antes do término de um curto período de supressão. |
| Delta Velocidade Angular X Delta Velocidade Angular Y | Mesmo conceito que delta aceleração. |
| Total Delta Velocidade Angular | Mesmo tratamento que delta total aceleração. O gesto de girar o instrumento em torno de seu eixo perpendicular é chamado de <i>spin</i> . |
| Delta Sensor de Pressão 1 Delta Sensor de Pressão 2 Delta Sensor de Pressão 3 | Mesmo conceito que delta aceleração. Produzido pelo gesto de tamborilar (<i>tapping</i>) sobre os sensores de pressão. |
| Terceira Camada de Parâmetros | Descrição |
| Suavização Aceleração X Suavização Aceleração Y Suavização Aceleração Z | A suavização de um sensor qualquer é feita somando-se os últimos n valores dos sensores e calculando-se a média. Assim, mudanças bruscas na aceleração são diluídas atenuando o contorno da função. Utilizando o acelerômetro desta forma é obtido um sensor de inclinação (<i>tilt</i>). |

| | |
|---------------|---|
| Cinética | É a soma de todos os deltas aceleração e velocidade angular, suavizados em um período longo de quatro segundo. Informa de certa maneira de a quantidade total de alterações de movimento do objeto nos últimos instantes. |
| Hit | É idêntico ao <i>jab</i> , mas com o limiar de ativação extremamente alto. A única forma de provocar um delta aceleração tão alto é dando um tapa no fundo do instrumento com a mão livre ou então batendo o mesmo em uma mesa ou parede. |
| Pressão Total | É a soma da pressão total exercida sobre o objeto (some do valor dos três botões de pressão). |

Quadro 9 Os parâmetros de saída do RedController

No próximo capítulo serão apresentados diversos resultados obtidos em experimentos onde utilizamos o RedController como um controlador gestual de diversas maneiras diferentes, tanto para interagir com o sistema **EVOGrain**, como o **G.A.Mapper** e **GranularStreamer** .

6 RESULTADOS

É sempre desejável, uma vez obtidos resultados sonoros satisfatórios do ponto de vista experimental, comparar estes com os de outra autoria e assim ter uma idéia de até que ponto chegamos em termos de originalidade e aperfeiçoamento técnico-artístico. Na seção 2.4 fizemos uma breve análise de peças musicais de dois expoentes da síntese granular, Curtis Roads e Barry Truax e assim foi possível identificar algumas formas recorrentes de organização dos grãos. Das peças de Curtis Roads destacam-se duas formas que denominamos *palavras granulares* e *estruturas clássicas*, enquanto que *texturas e processos* são exemplificados pela obra de Barry Truax. Neste capítulo utilizamos estes termos como referencial para avaliar os resultados obtidos com os sistemas **EVOGrain** e **GranularStreamer** em diversas configurações.

6.1 Resultados Obtidos com o Sistema EVOGrain

EVOGrain implementa uma solução interessante e exclusiva que permite ao usuário explorar as inúmeras possibilidades de síntese granular no EGS (Extended Gabor Space) de maneira intuitiva. Quando alvos são desenhados pelo usuário, o algoritmo genético calcula diversas gerações por segundo e atualiza os parâmetros da síntese sonora em tempo real (latência de 500 ms). Essa resposta imediata permite que o usuário localize facilmente regiões do espaço de parâmetros nas quais os intervalos definidos pelos alvos resultem em estruturas granulares clássicas. Foi realizada uma exploração sistemática, porém não exaustiva, das possibilidades do sistema. Diversos efeitos sonoros, timbres ricos e texturas podem ser sintetizados pelo EVOGrain.

A Figura 28 exibe o desenhos de alvos onde o *fluxo* vermelho realiza síntese com densidade constante de 50 grãos por segundo e onde todos os grãos têm a mesma duração e frequência de 28 ms e 200 Hz; o *fluxo* azul forma uma nuvem assíncrona de grãos de diferentes tamanhos (de 23 a 73 ms), espalhados por uma banda de frequência (de 1300 e 5650 Hz) com densidade de 7 a 27 grãos por segundo. O **Exemplo 01 do CD-ROM** é uma gravação da síntese realizada pelo sistema EVOGrain com estes alvos e com seis

configurações diferentes para envoltória e forma de onda: *Gaussian* e *Sine*, *Expodec* e *Sine*, *Expodec* e *Square*, *Rexpodec* e *Add Sine*, *Thin Gaussian* e *Random*, *Fat Gaussian* e *Glisson*.

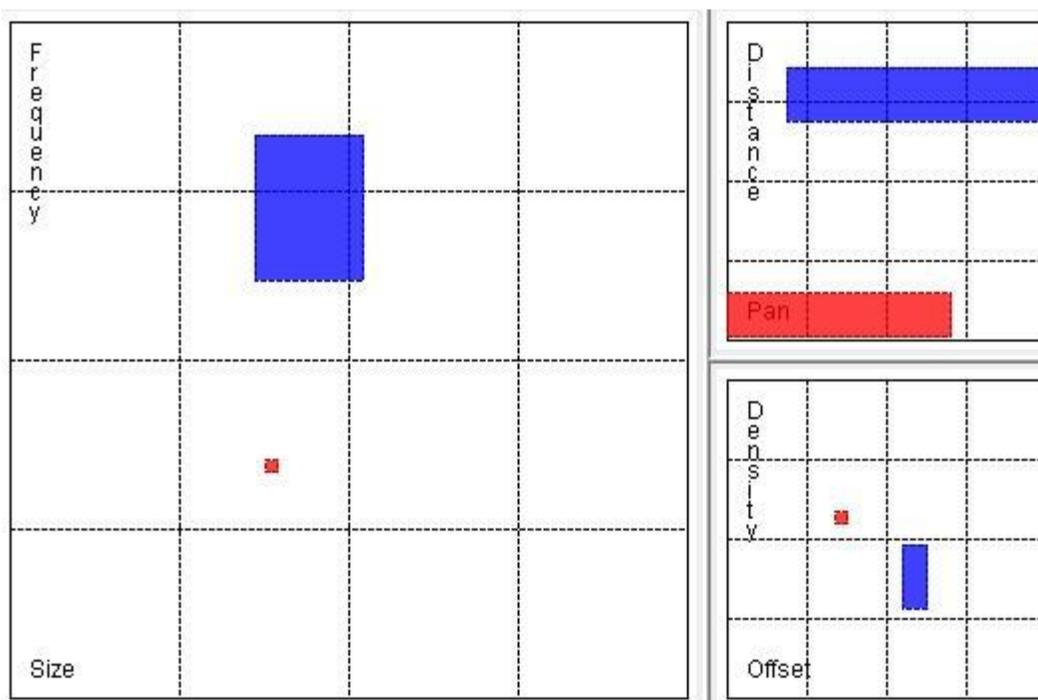


Figura 28 Um *fluxo* síncrono (vermelho) e uma nuvem assíncrona (azul)

Além de facilitar a localização de estruturas clássicas como demonstrado no exemplo anterior, a interface gráfica permite ao usuário conduzir (ou “reger”) as camadas pelo Espaço de Gabor Estendido controlando processos de transformação de texturas sonoras com uma dinâmica bastante rica. Por exemplo, a partir do estado inicial representado pela Figura 28 realizamos dois processos de transformação das texturas. Primeiramente a textura azul tem o tamanho dos grãos reduzidos para uma faixa de 1 a 3 ms e, conseqüentemente, torna-se mais rarefeita, pois os espaços de silêncio entre os grãos são aumentados. Logo em seguida realizamos um processo de dilatação da textura vermelha no qual os grãos que estavam concentrados em uma única freqüência passam a ser espalhados por uma banda mais larga de 120 Hz a 560 Hz. Este processo está gravado no **Exemplo 02 do CD-ROM**, e o estado final do sistema é exibido na Figura abaixo.

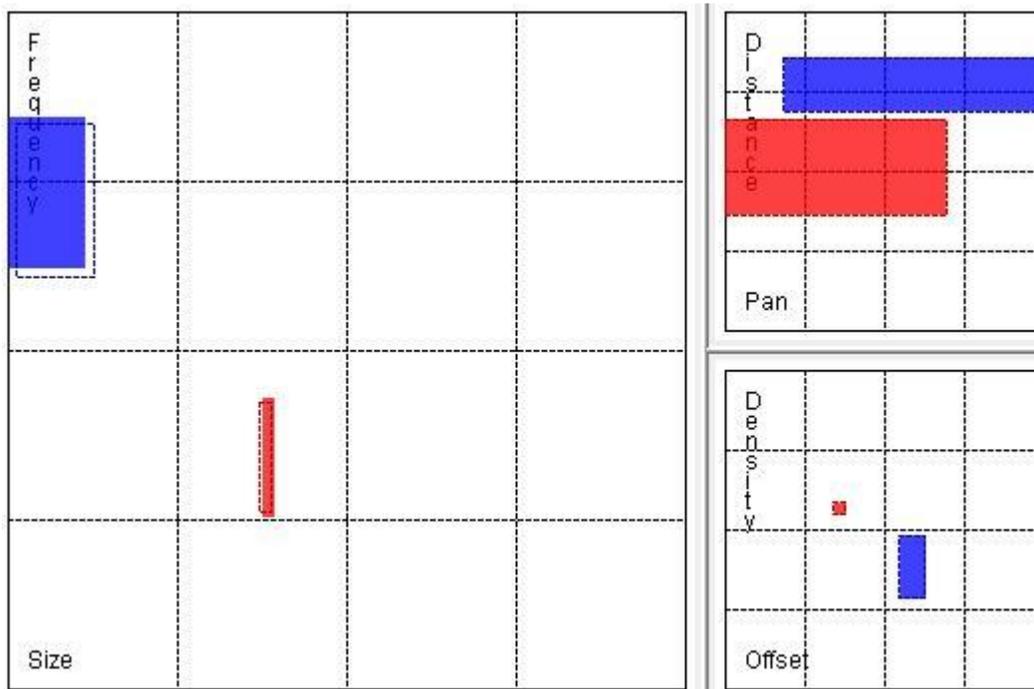


Figura 29 Estado final dos processos de rarefação e dilatação

A funcionalidade de salvar *presets* com configurações de retângulos alvos se mostrou muito útil para a elaboração de processos mais ricos e que podem ser utilizados, por exemplo, em composição em tempo diferido. No próximo exemplo utilizamos quatro conjuntos de alvos para compor um processo de transformação cíclico, isto é, as configurações são carregadas em ordem e, após o último conjunto, voltamos ao primeiro (Figura 30) em um loop. O **Exemplo 03 do CD-ROM** contém a síntese de dois ciclos deste processo sendo que o *fluxo* vermelho tem grãos com envoltória *Expodec* e o azul com *Gaussian*. Nota-se que quando os *fluxos* estão próximos à configuração “4” as texturas convergem para uma mesma região do espaço de parâmetros e tendem a se fundir acusticamente.

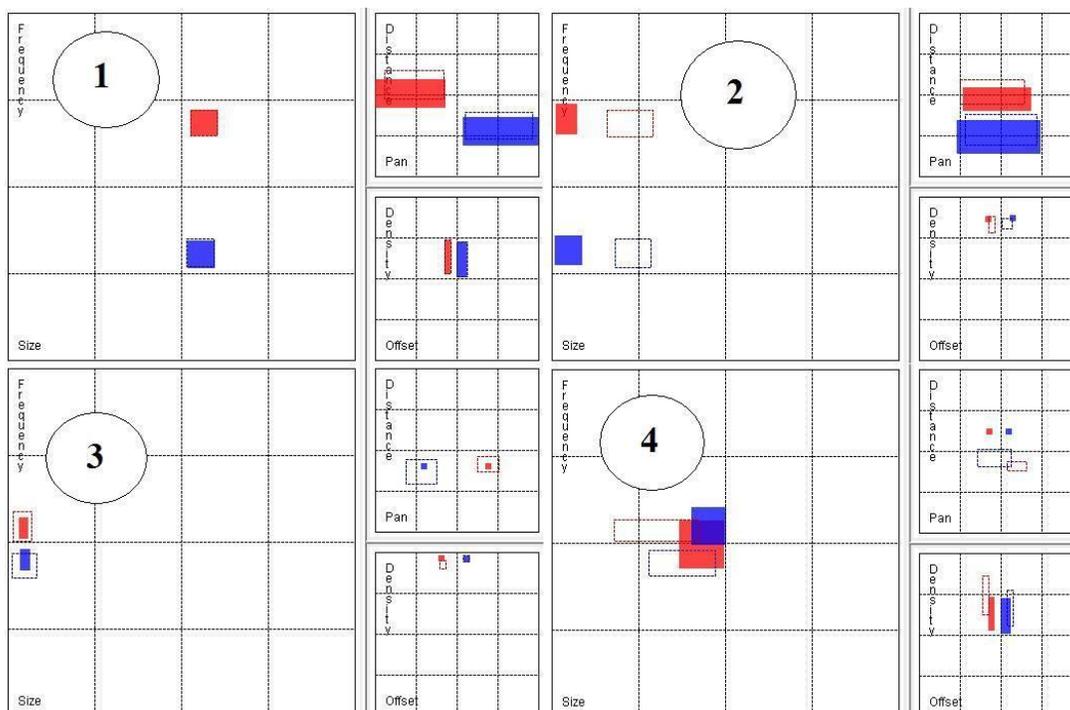


Figura 30 Quatro *presets* do EVOGrain

Com estes breves exemplos podemos fazer algumas observações sobre a condução de processos no EGS do ponto de vista da Espectromorfologia de Smalley (SMALLEY, 1997). Primeiramente precisamos relacionar as ações dentro do EGS com a taxonomia de Smalley. Diremos que temos um *processo de transformação unidirecional* quando um novo alvo é simplesmente transladado dentro de um subespaço do EGS, ou seja, quando o alvo é redesenhado com as mesmas dimensões do anterior, porém com coordenadas diferentes. Um *processo de transformação multidirecional* quando mais de uma das dimensões do alvo é modificada em um ou mais subespaços do EGS como, por exemplo, a contração ou dilatação dos intervalos dos parâmetros associados. Assim, uma alteração na dimensão horizontal e/ou vertical do alvo implica em processos multidirecionais classificados por Smalley como aglomeração, dissipação, contração e dilatação e outros. Evidentemente que os alvos desenhados podem conter simultaneamente alterações tanto na posição como no tamanho em relação ao alvo anterior. Verificamos experimentalmente que quanto mais semelhante for a posição e o tamanho dos alvos de dois *fluxos* diferentes, melhor será a

percepção de fusão entre as texturas sonoras sendo daí mais fácil o controle da convergência dos processos.

Outra funcionalidade muito importante do sistema é a capacidade de salvar arquivos de áudio de uma sessão em tempo real os quais podem, posteriormente, ser utilizados em composição em tempo diferido com o auxílio de um seqüenciador. O **Exemplo 04 do CD-ROM** apresenta um breve estudo no qual utilizamos um seqüenciador para organizar diversos arquivos de áudio gerados pelo EVOGrain. O estudo inicia-se com a entrada de uma nuvem de grãos espalhados por uma largura de banda que estreita até soar uma única frequência; outras três “vozes” realizam o mesmo processo formando um “acorde”. Após esboçar o plano do estudo, gravamos diversas sessões conduzindo os *fluxos* vermelho e azul conforme um planejamento composicional ou “partitura”. Colocando o arquivo da primeira sessão em *playback*, realizamos uma segunda sessão para gravar as vozes três e quatro, e assim consecutivamente para obter a gravação das vozes cinco a oito (que só soam a partir do meio do estudo). O espectro sonoro deste estudo pode ser visto na Figura 31.

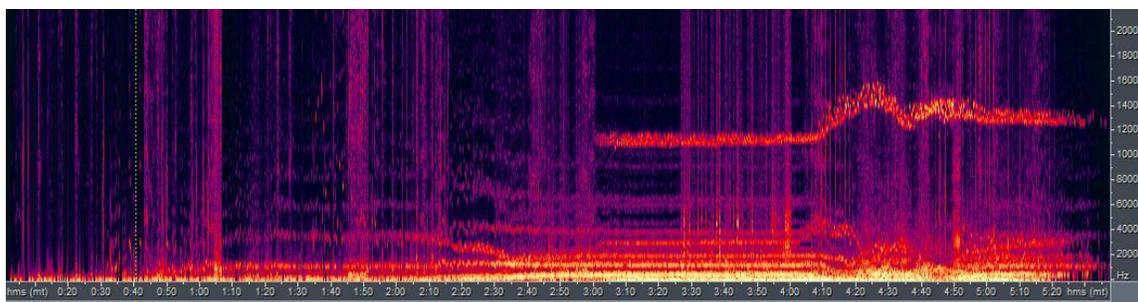


Figura 31 Espectro do áudio da Faixa 04 do CD

Além de realizar o procedimento acima para sobrepor camadas texturais, também utilizamos a função de salvar arquivos de áudio para obter objetos sonoros específico. Por exemplo, este estudo é inteiramente permeado pela presença de uma estrutura clássica da síntese granular chamada *pulsar*. Para gerar estes objetos sonoros gravamos quatro arquivos de áudio com pulsares de diferentes velocidades, e então recortamos estes arquivos em pequenos pedaços que foram arranjados ao longo do estudo. Por último, em uma sessão utilizamos a gravação de duas frases proferidas por Curtis Roads para obter exemplos da

técnica conhecida como granulação (onde os grãos são fragmentos de um arquivo de áudio) sendo aplicada pelo EVOGrain.

O último resultado observado no sistema **EVOGrain** se refere ao seu uso em conjunto com um controlador gestual. Como visto acima, o EVOGrain é uma sistema que realiza, por meio de um algoritmo genético, um processo gradual de transformação do som do estado atual para um estado alvo. Desta forma, quando um novo alvo é desenhado, é necessário deixar o sistema evoluir por alguns segundos para que seja evidenciado o processo de transformação do som. Não é um sistema projetado para ter resposta rápida quanto aos parâmetros da síntese, portanto, se novos alvos são desenhados muito freqüentemente não se obtém nem um processo de transformação do som e nem a resposta imediata aos parâmetros definidos pelo alvo. Conclui-se que o uso de um controlador gestual não deve resultar em uma atualização muito freqüente dos alvos do EVOGrain, ao invés disso deve apenas interferir em momentos específicos dependendo da intenção musical do usuário.

Para dar um exemplo específico, o controlador gestual **RedController** foi utilizado como dispositivo de entrada para o sistema **EVOGrain** de duas formas diferentes. Na primeira, o retângulo alvo tem suas coordenadas x e y determinadas pela inclinação do controlador e o tamanho dos lados horizontal e vertical determinados pela pressão nos sensores 1 e 2. Sempre que o controlador é movimentado, o alvo muda de posição, e para manter um mesmo alvo e permitir que o som evolua é necessário segurar o controlador no mesmo lugar no espaço. Na segunda abordagem, as coordenadas e o tamanho do alvo são determinados de maneira semelhante, porém o alvo é desenhado somente quando gesto *jab* é realizado com o controlador. Portanto, o *performer* realiza o gesto de um *jab* para desenhar um alvo e então pode realizar outras funções enquanto o EVOGrain se encarrega de processar a evolução da síntese em direção ao alvo.

6.2 Resultados Obtidos com o Sistemas Derivados do EVOGrain

Nesta seção são apresentados os resultados sonoros obtidos com o uso do **GranularStreamer** realizando a síntese granular ao invés do EVOGrain. Conforme

descrevemos na seção 4.1, o GranularStreamer é um sistema derivado do módulo de síntese sonora do EVOGrain e que realiza apenas a tarefa de controlar *fluxos* de grãos, não possui interface e todo controle é feito através de mensagens OSC. Ou seja, para utilizar este sistema é necessário utilizar um software que envie mensagens OSC com os valores desejados para os parâmetros da síntese. Este software pode ser uma interface gráfica ou o *driver* de um dispositivo físico como, por exemplo, o RedController.

A Figura 32 apresenta duas interfaces desenvolvidas em PD (*Pure Data*) para controlar os parâmetros da síntese granular, ambas com abordagens bem tradicionais. A primeira interface (à esquerda) é um painel com dois *sliders* para cada parâmetro, sendo um para o valor inicial e o outro para o valor final, que controla os parâmetros da síntese em tempo real. A segunda interface (à direita), também em PD, é constituída de uma área onde o usuário desenha, em tempo diferido, uma linha que descreve o valor de um parâmetro ao longo do tempo.

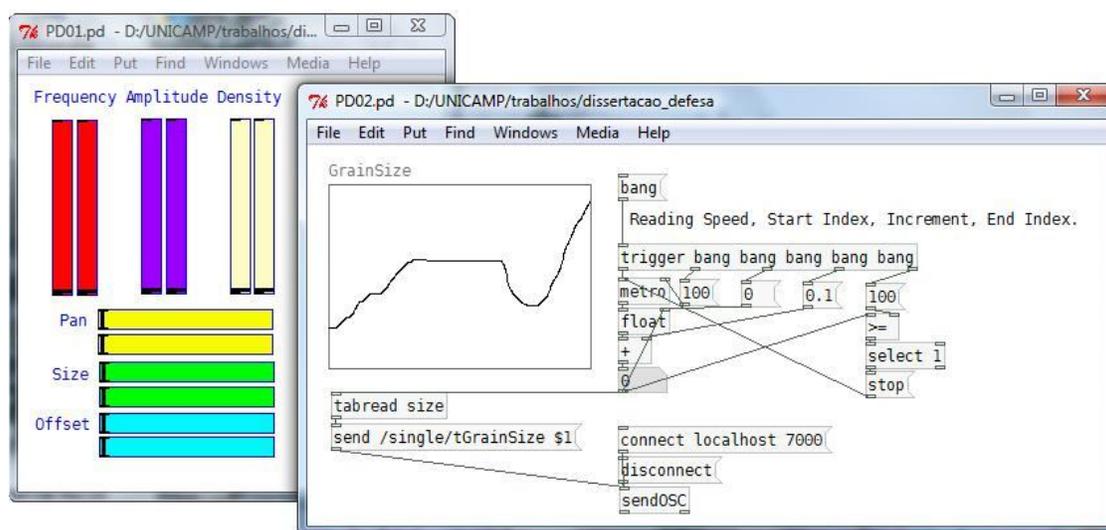


Figura 32 Dois *patches* em PD para controlar o GranularStreamer

Quando o GranularStreamer é controlado pelo **G.A.Mapper** (Genetic Algorithm Mapper), ou melhor, quando G.A.Mapper é utilizado como um módulo intermediário entre a interface de um sistema e o módulo de síntese de som, os resultados obtidos foram muito semelhantes aos resultados do EVOGrain. Este fato ocorre porque ambos os sistemas

apresentam a mesma arquitetura e forma de operação, isto é, por meio de uma interface (gráfica ou física), o usuário determina alvos que guiam um processo evolutivo em um espaço de parâmetros multidimensional e também foi utilizada a mesma distribuição aleatória (uniforme) para sortear valores dos parâmetros de cada grão. O resultado de tal arquitetura é uma sonoridade em constante transformação conforme apresentamos na seção anterior.

Não obstante, ao realizarmos experimentos com o **GranularStreamer** conectado diretamente a uma interface gráfica ou a um controlador, encontramos resultados bastante diversos daqueles encontrados com o EVOGrain. Primeiramente lembramos que os alvos do sistema EVOGrain são desenhados em função de suas coordenadas e tamanho, ou seja, o usuário não pensa em valores exatos de frequência, duração ou densidade de grãos por segundo, ao invés disto, os valores das coordenadas são proporcionais aos valores físicos das quantidades, e além disto o usuário desenha alvos intuitivamente conforme deseja obter texturas ou sonoridades, por exemplo, “graves” ou “agudas”, mais ou menos “granuladas” ou, ainda, “coloridas” ou “monocromáticas”. Já no GranularStreamer, os valores dos parâmetros da síntese são definidos em valores físicos exatos, por exemplo, a frequência é definida em Hertz, a duração dos grãos em milésimos de segundo, etc. Isto permite que sejam criados objetos granulares e processos de transformação com contornos deterministicamente bem definidos. Destaca-se também que o GranularStreamer pode sintetizar simultaneamente uma grande quantidade de *fluxos* de grãos com as mais diversas formas de onda e envoltória, e que estes *fluxos* podem ser operados de maneira independente possibilitando composições texturais em tempo real com grande diversidade de material sonoro. Resumidamente, por se tratar apenas de um módulo de síntese de sonora, desacoplado do algoritmo genético, o GranularStreamer abre novas possibilidades de controle para a síntese granular, tanto através de interfaces gráficas e controladores gestuais, como por aplicativos derivados de modelos computacionais de alto nível como, por exemplo, algoritmos genéticos, autômatos celulares, grafos, Cadeias de Markov, etc.

O **Exemplo 05 e 06 do CD-ROM** apresentam dois exemplos de síntese granular onde os parâmetros são controlados de maneira determinística para criar um processo muito bem delineado. Na Figura 21 da Seção 4.3 onde apresentamos o **GranularDrawer**, são

exibidas duas imagens (esquerda e direita), geradas pelo GranularDrawer, que representam respectivamente os exemplos sonoros cinco e seis. Na parte esquerda da figura (ouça a Faixa 05) é apresentado um processo recíproco onde a densidade de grãos por segundo (GPS) aumenta a partir do zero, atinge o pico de dois mil GPS e retorna ao zero. Ao mesmo tempo a frequência dos grãos também varia ascendente e descendente, e sua duração, inicialmente de 250 ms, é reduzida até 1 ms para em seguida voltar a 250 ms. Espacialmente pode-se notar que os grãos começam a ser emitidos apenas no canal esquerdo, então caminham em direção ao centro e então retornam à esquerda. Já a parte direita da mesma figura (ouça a Faixa 06) representa um processo onde uma nuvem de grãos é alterada uni-direcionalmente, isto é, a banda de frequência é cada vez mais alta, a duração dos grãos cada vez menor e, espacialmente, os grãos são espalhados cada vez mais para as laterais a partir de uma posição centralizada.

Ao utilizar o **RedController** para controlar os parâmetros da síntese do **GranularStreamer** podem ser obtidas diversas formas de interação musical. Para realizar o mapeamento entre os parâmetros de entrada e os parâmetros de saída foi necessário desenvolver um pequeno aplicativo que denominados **VirtualEnsemble**. Este aplicativo é constituído basicamente de três módulos: o primeiro módulo é responsável por receber todas as mensagens OSC que são enviadas pelo *driver* do RedController com o valor atual dos sensores como, por exemplo, aceleração nos planos x, y e z; o segundo módulo é responsável por enviar mensagens OSC para o GranularStreamer com os valores dos parâmetros da síntese para cada *fluxo* como, por exemplo, frequência e duração dos grãos; o terceiro módulo é onde são codificadas todas as funções que convertem os valores dos sensores do controlador em valores da síntese granular, ou seja, é módulo onde é feito o mapeamento propriamente dito.

Conforme mencionamos anteriormente, o mapeamento de um controlador ao algoritmo de síntese sonora é um ponto crucial na construção de Instrumentos Musicais Digitais (DMIs). Mesmo considerando um único controlador como, por exemplo, o **RedController**, e um único algoritmo de síntese como, por exemplo, o **GranularStreamer**, podemos realizar o mapeamento de inúmeras maneiras resultando em DMIs de natureza consideravelmente distinta tanto quanto ao gesto como a sonoridade

obtida (Figura 33). Foi então realizada uma exploração sistemática das formas de construção de vários DMIs conectando o protótipo **RedController** ao sistema de síntese granular chamado **GranularStreamer** .

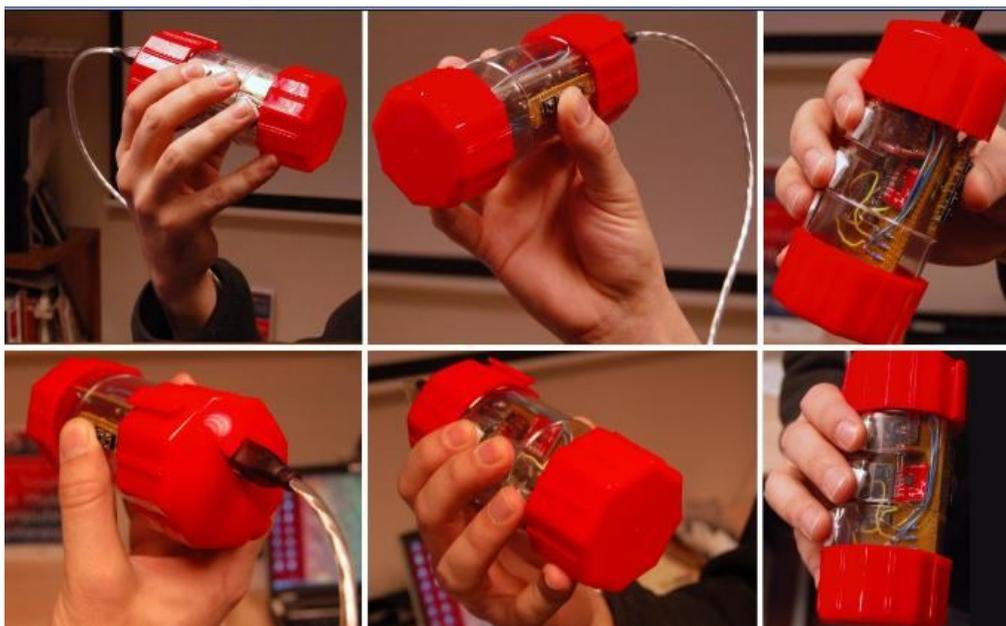


Figura 33 Diferentes mapeamentos levam a diferentes DMIs

A seguir é apresentada uma breve descrição de alguns DMIs de exemplo sem entrar nos detalhes particulares de cada mapeamento feito com o **VirtualEnsemble**:

-Hit Pulsar: no **Exemplo 07 do CD-ROM** apresentamos um exemplo sonoro produzido pelo instrumento granular que denominamos *Hit Pulsar*. Ao realizar um gesto *jab* com controlador, é disparado um pulsar de curta duração. Um pulsar é uma estrutura granular bem conhecida, fluxos de densidade baixa e constante que se comportam como pulsos sonoros regulares, com variações de frequência. O nome foi emprestado da astrofísica estelar por causa de estrelas que emitem pulsos de luz com uma regularidade espantosa. No caso do nosso Hit Pulsar, a duração total do pulsar varia de 4 a 20 segundos de acordo com a intensidade do *jab*. Outros parâmetros como posição panorâmica (*pan*), banda de frequência e densidade do fluxo de grãos do pulsar são determinados pela pressão

exercida nos botões do controlador na hora do golpe. O instrumento não se limita a disparar apenas um pulsar, pois, podemos criar diversos *fluxos* do **GranularStreamer** disparando pulsares que soam simultaneamente, cada um com sua densidade, frequência e posição particular, criando assim uma polirritmia de objetos sonoros pulsantes;

-*Roads's Words*: outro resultado interessante foi obtido com a criação de alguns objetos granulares de curta duração (entre 0,5 e 2 segundos) os quais chamamos de *sílabas granulares*. Para disparar cada um destes objetos o *performer* deve tamborilar sobre os botões de pressão do **RedController**. Cada um destes três botões dispara diversas sílabas diferentes de acordo com a inclinação do controlador. Este DMI foi batizado com o nome *Roads's Words*, pois, ao tamborilar rapidamente sobre os botões de pressão, são formadas as “palavras granulares” tipicamente utilizadas por Curtis Roads em suas composições. Uma breve *performance* com este DMI está gravada no **Exemplo 08 do CD-ROM** e o espectro de uma palavra com três segundos de duração é apresentado na figura abaixo;

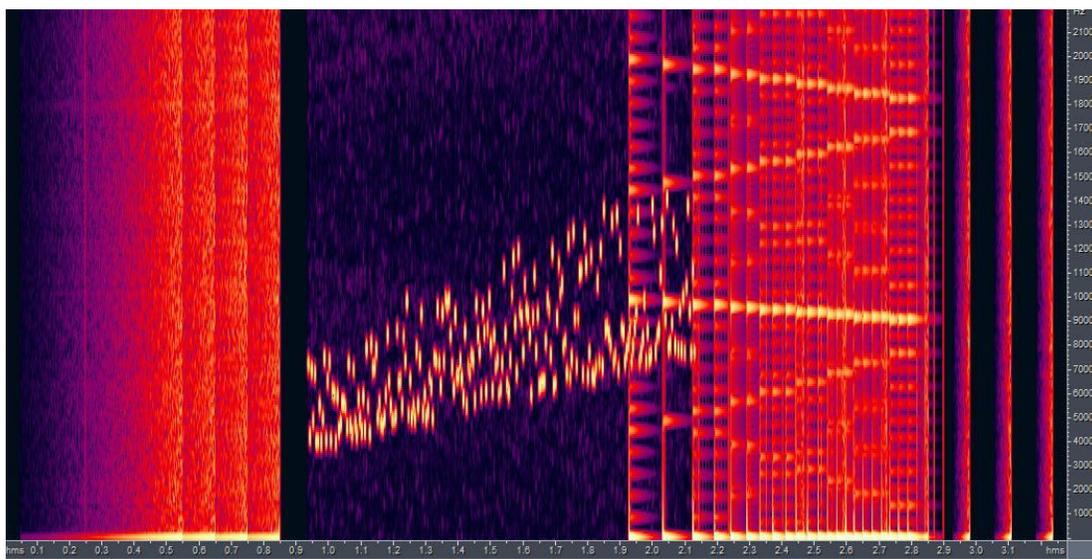


Figura 34 Uma palavra de três segundos gerada pelo DMI “Roads’s Words”

-*Streamer* : neste experimento procuramos utilizar o RedController como um *joystick* e de maneira muito semelhante ao SpaceNavigator. A propriedade de inclinação é utilizada para controlar a frequência central dos grãos e a posição panorâmica; os botões de

pressão controlam a densidade de grãos por segundo, e a largura de banda dos grãos; a amplitude dos grãos é fixa, enquanto que a duração é calculada com base na densidade. Um exemplo da síntese sonora resultante deste DMI está gravado no **Exemplo 09 do CD-ROM**;

-*Vocoder*: neste DMI cuja síntese está gravada no **Exemplo 10 do CD-ROM** o som é produzido realizando *jabs* no ar (movimentos bruscos e curtos). Sempre que um *jab* é realizado há um aumento súbito de amplitude, proporcional à força do movimento, e que diminui gradualmente em poucos segundos. Os grãos têm o seu conteúdo proveniente de um arquivo WAVE com a gravação de uma voz pronunciando as vogais “a”, “e”, “i”, “o”, “u”. Claramente outros fonemas podem ser usados e até mesmo arquivos com qualquer conteúdo sonoro. De forma semelhante ao *Streamer*, este DMI utiliza os parâmetros de inclinação e de pressão para navegar em um espaço de parâmetros que controla o ponto de leitura do arquivo WAVE, duração dos grãos e a densidade de grãos por segundo;

-*Shaker*: este DMI é um instrumento inspirado em um chocalho, mas que produz sons granulares cristalinos resultantes de grãos com envoltória EXPODEC e forma de onda senoidal em altas frequências (conforme o **Exemplo 11 do CD-ROM**). O som é produzido ao chacoalhar o dispositivo de três maneiras diferentes: frontalmente, lateralmente e girar o instrumento em seu eixo (*spin*). O movimento resultante controla a densidade de um fluxo assíncrono de grãos, enquanto a intensidade de cada movimento componente controla a frequência central da banda de frequência;

-*Cloud Mixer*: como o próprio nome nos leva a deduzir, este instrumento musical permite misturar nuvens granulares. Cada um dos três botões de pressão controla a densidade dos grãos de um *fluxo*. Cada um destes *fluxos* é configurado com envoltórias e formas de onda diferente. Combinando a pressão exercida sobre os botões do RedController é possível mixar nuvens e explorar as minúcias de seu conteúdo espectral particular (**Exemplo 12 do CD-ROM**). Outros parâmetros do controlador como *tilt*, *shake* e *spin* controlam outros parâmetros de forma idêntica para os três *fluxos*;

-*Throw*: neste DMI reproduzimos o segundo experimento realizado com o **T-Stick**. Aqui o usuário deve executar o gesto de arremessar o objeto. Deste gesto são capturados os parâmetros de força do arremesso, efeito *spin*, ângulo de elevação e pressão total sobre o

objeto. O parâmetro ‘ spin’ é utilizado para escolher entre quatro objetos sonoros que foram chamados de linha, parábola, círculo e espiral; enquanto os outros três parâmetros controlam a frequência, duração total e posição panorâmica dos objetos (duas gravações destes processos foram apresentadas nos **Exemplos 05 e 06 do CD-ROM**).

No decorrer dos experimentos realizados com o GranularStreamer foram construídos diversos outros objetos granulares e DMIs para síntese granular, porém os exemplos acima são suficientes para instanciar a seguinte afirmação: se por um lado o **GranularStreamer** não é um sistema amigável, pois não apresenta interface e requer programação para ser utilizado, e muito menos intuitivo, não prescindindo do espaço de parâmetros da síntese granular, por outro lado apresenta alto desempenho em tempo real e muita flexibilidade para controlar vários *fluxos* simultâneos e pode ser facilmente acoplado a qualquer outro sistema que suporte o protocolo OSC complementando assim as características do sistema EVOGrain.

7 CONSIDERAÇÕES FINAIS E PERSPECTIVAS

A idéia inicial deste trabalho foi estender o espaço bidimensional Tempo-Frequencia de Gabor para um espaço de seis dimensões descrito anteriormente, onde processos evolutivos, controlando fluxos de sons granulares, pudessem ser visualizados e controlados. O software EVOGrain, resultante desta pesquisa, apresenta uma interface gráfica bastante simples e intuitiva e uma grande versatilidade no controle dos parâmetros básicos do fluxo sonoro. Uma possibilidade bastante interessante para compositores é ter mais que dois fluxos de sons independentes. Isto, no entanto, levaria a uma maior latência do sistema para responder ao usuário em tempo real o qual foi verificado parcialmente com o desenvolvimento do GranularStreamer, que foi escrito em C++, em vez de JAVA, e no qual realizamos uma série de otimizações do código. A latência baixou de 500 ms para 50 ms. Claramente em máquinas com processadores muito rápidos esta latência pode chegar a ser praticamente nula.

Como vimos no capítulo 5, outros controladores gestuais foram previamente utilizados no controle da síntese granular como, por exemplo, o *Poseidon*. Porém o fato deste controlador ser equipado apenas com *sliders* torna-o muito limitado quanto às possibilidades de gestos que podem ser capturados. O RedController, por utilizar sensores de natureza distinta, em particular, os sensores de aceleração, permite que o sistema seja controlado pelo mais diversos gestos. Através dos experimentos realizados verificamos a flexibilidade do RedController ao ser utilizado na “confeção” de um instrumento musical digital.

Os resultados desta pesquisa dão ensejo a várias outras questões e desenvolvimentos adicionais importantes:

- 1) O Espaço de Gabor é um modelo natural para definir processos evolutivos para gerar fluxos de sons granulares em tempo real. Podemos aplicar o mesmo modelo no sentido de Roads (2001), onde os grãos são definidos de forma mais flexível do que os de Gabor (quanta de sons). Os espaços (e também a função de adaptabilidade) podem incluir mais variáveis como harmonicidade, rugosidade e brilho, emprestados da psicoacústica

(ROEDERER, 1998), como descritores de alto nível. Isto obviamente implica adicionar novas dimensões ao Espaço de Gabor Estendido e ainda com diferentes métricas e função de adequação (*fitness*).

2) A função de adequação, em princípio pode depender não somente do genótipo de um indivíduo como também de parâmetros externos definidos antes da execução do algoritmo. Em certo sentido isto poderia ser interpretado como influência de um ambiente externo, que em nosso sistema poderia ser codificada como mensagens OSC.

3) Claramente, há um número infinito de possibilidades para a função de cruzamento. Em princípio, a combinação desta função de cruzamento com a função de adequação e a mutação pode levar a dinâmicas bastante complexas no Espaço de Gabor Estendido (Espaço de Parâmetros). A nossa particular escolha foi motivada pela sua simplicidade matemática sendo de fácil implementação computacional.

4) Uma melhoria importante está relacionada com a distribuição de probabilidade utilizada para sortear os parâmetros de criação de um grão dentro de uma família. Somente a distribuição uniforme foi utilizada no **EVOGrain** v1.0. Notadamente, uma classe mais abrangente de distribuições de probabilidade pode ser incorporada ao sistema.

5) O uso de mensagens OSC podem ser mais abrangentes. Por meio destas mensagens é possível integrar o **EVOGrain** com outros sistemas de áudio, criar uma interface alternativa que reflita o estado de cada fluxo sonoro, controlar parâmetros de aplicações de vídeo, como a velocidade da reprodução ou transição entre duas imagens e até mesmo realizar síntese de gráficos como vimos na seção 4.3 com o **GranularDrawer**.

6) Novas funções de adequação (*fitness*) podem ser exploradas levando a novos tipos de convergência e conseqüentemente de diferentes fluxos granulares.

7) É possível definir famílias de grãos associadas a novas formas geométricas como triângulos, círculos, curvas e polígonos. Pensamos que estudos em artes visuais digitais podem levar a uma mudança completa da geometria da interface do EVOGrain e, além disto, permitindo a princípio, desenhos geométricos como, digamos, os de Kandinsky. Uma interface com uma plasticidade maior pode contribuir para um controle mais refinado e interessante do sistema. Por exemplo, a espacialização pode ser estendida do ambiente estereofônico aqui apresentado para um ambiente polifônico. Do ponto de vista da interface gráfica pode ser interessante trocar retângulos por regiões com geometria circular ou elipsoidal, onde o centro representa a posição do ouvinte ou ainda com a geometria de uma sala em particular.

8) No sentido de aproximar a interface das intenções do compositor ou sound designer, é interessante possibilitar a utilização de envoltórias e formas de onda desenhadas pelo usuário. Estas formas de onda particulares poderiam ser incorporadas a banco de formas de onda e envoltórias disponibilizadas ao usuário.

9) Vimos que taxa de mutação alta faz com que o processo de convergência para o indivíduo alvo seja bem mais rápida. Na atual implementação a mudança nesta taxa não pode ser feita em tempo real. Seria desejável que isto pudesse ser controlado pelo usuário através de uma interface gestual, levando, em tempo real, a mudanças de velocidade de convergência dos fluxos para seu respectivos indivíduos alvo.

10) Criação de novos mapeamentos entre o GranularDrawer e o GranularStreamer tornando as imagens mais elaboradas e indicativas do processo que está sendo visualizado e permitindo inclusive que o controle da síntese seja feito a partir da interação do usuário com as imagens do GranularDrawer. Novos mapeamentos podem ser pesquisados como, por exemplo, entre o brilho e a amplitude da forma de onda, entre a cor e a frequência da forma de onda, tamanho do círculo (ou outra forma geométrica) e duração do grão, e posição horizontal e Pan.

11) Nesta versão do G. A. Mapper, os retângulos acinzentados são meramente um efeito gráfico, mas pensamos que numa versão posterior eles possam representar subestruturas do genes, tais como cadeias de proteínas em genética molecular. O G.A.Mapper apresentaria então uma hierarquia de estruturas manipuláveis por operadores em seus respectivos níveis, ou seja, conceitualmente teríamos um sistema que pode ser classificado computacionalmente como top-down.

12) O G.A Mapper pode também ser utilizado coletivamente por uma número arbitrário de usuários, cada um deles controlando parte do cromossomo do indivíduo alvo. Isto possibilita vários tipos de interações musicais tais como: jogos, jam sessions, improvisos coletivos, dependendo apenas da comunicação entre estas sessões coletivas realizadas remotamente pelos usuários.

13) A partir do projeto do RedController podemos desenvolver uma família de controladores equipados com outros tipos de sensores como, por exemplo, um sensor de pressão barométrica acoplado a um bocal para capturar a intensidade do sopro do usuário. O uso de controladores gestuais diversos nos leva a investigar se existe uma classe de sensores mais adequados para o controle da síntese granular e com maior adaptabilidade ao gesto humano.

14) No caso particular do EVOGrain, o usuário poderia controlar o sistema utilizando um dispositivo em cada mão. Desta forma o RedController controlaria apenas o *fluxo* vermelho, enquanto que o “BlueController” estaria associado ao *fluxo* azul. Ambos dispositivos poderiam funcionar de forma idêntica (o mesmo mapeamento) ou de maneiras completamente distintas. Por exemplo, a densidade do *fluxo* vermelho seria controlada por movimentos de chacoalhar o RedController, enquanto a densidade da voz azul pode corresponder diretamente à pressão do ar do sopro do instrumentista.

Finalmente queremos enfatizar que as premissas as quais preconizamos para o desenvolvimento do sistema EVOGrain na seção 2.5 foram completamente satisfeitas. O

GranularStreamer é um aplicativo derivado daquele e com uma grande flexibilidade de interação com outros ambientes externos através do protocolo OSC, já bem estabelecido no meio musical. O RedController é um protótipo de um controlador gestual exemplificando as possibilidades de ambientes externos ao GranularStreamer . Esta tríade de aplicativos demonstra o potencial musical, do ponto de vista composicional e performático, do Espaço de Gabor Estendido, o universo sonoro da síntese granular.

REFERÊNCIAS¹

- BRYANT, V. *Metric Spaces: Iteration and Application*. Cambridge University Press, 1985.
- CLARKE, M. Composing at the intersection of time and frequency. *Organised Sound* 1(2). Cambridge University Press, 1996. pp.107-117.
- DISCIPIO, A. Composition by Exploration of Nonlinear Dynamical Systems. *Proceedings of the 1990 International Computer Music Conference*. San Francisco: International Computer Music Association, 1990. pp.324-327.
- FISCHMAN, R. Clouds, Pyramids, and Diamonds: Applying Schrödinger's Equation to Granular Synthesis and Compositional Structure. *Computer Music Journal* 27(2). 2003. pp.47-96.
- FORNARI, J.; MANZOLLI, J. ESSynth - Estudo e Desenvolvimento de um Sintetizador Evolutivo de Segmentos Sonoros. PosDoc research sponsored by FAPESP, process: 04/00499-6, 2007.
- GABOR, D. Theory of Communication. *JIEE*, Vol.93, Part 3, No.26. London: 1946. pp.429-457.
- GABOR, D. Acoustical Quanta and The Theory of the Hearing. *Nature* 159. 1947. pp.591-594.
- GABOR, D. Lectures on Communication theory. Technical Report 238. MIT, Research Laboratory of Eletronics, 1952.
- GOTO, S. The aesthetics and technological aspects of virtual musical instruments: The case of the SuperPolm MIDI violin. *Leonardo Music J.*, vol. 9. 1999. pp.115–120.
- HOLLAND, J. H. Genetic Algorithms *SCIETIFIC AMERICAN*, 1992. pp.66-72.
- HUNT, A.; KIRK, R. Mapping strategies for musical performance-trends in gestural control of music. *Trends in gestural control of music*. Paris: IRCAM, 2000.
- MALLOCH, J. W. A Consort of Gestural Musical Controllers: Design, Construction and Performance. Master Thesis, Music Technology Area, Schulich School of Music, McGill University, Montreal, Canada, 2008.

¹ Baseadas na norma NBR 6023, de 2002, da Associação Brasileira de Normas Técnicas (ABNT).

MALLOCH, J.; SINCLAIR, S.; WANDERLEY, M.M. A Network-based Framework for Collaborative Development and Performance of Digital Musical Instruments. Proceedings Of Computer Music Modeling and Retrieval 2007 Conference. Berlin: 2008.

MAIA JR, A.; MIRANDA, E.R. Granular synthesis of sound through Markov chains with fuzzy control. Annals of the International Computer Music Conference (ICMC). Barcelona: 2005.

MIRANDA, E.R. Granular Synthesis of Sounds by Mean of a Cellular Automaton. Leonardo, Vol.28, No.4. 1995. pp297-300.

MOORE, F. R. Elements of Computer Music. New Jersey: Prentice Hall, 1990.

MORONI, A. et al. VoxPopuli: evolutionary computation for music evolution. Proceedings of the AISB'99 Symposium on Creative Evolutionary Systems - CES'99. Edimburg: 1999.

OPIE, T. Creation of a real-time granular synthesis instrument for live performance. Master Thesis, Creative Industries: Music, Queensland University of Music, Brisbane, Australia, 2003.

REIGLE, R. Spectral Musics Old and New. Proceedings of the Istanbul Spectral Music Conference. Istanbul: Pan Yayincilik, 2008.

ROADS, C. Microsound. Cambridge: MIT Press, 2001.

ROEDERER, J. Introdução a física e psicofísica da música. São Paulo: EDUSP, 2002.

RYAN, J. Some remarks on musical instrument design at STEIM. Contemporary Music Review, vol. 6, no.1. 1991. pp.3–17.

SMALLEY, D. Spectromorphology: explaining sound-shapes. Organised Sound 2(2). Cambridge University Press, 1997. pp.107-126.

SMITH III, J. O. Viewpoints on the History of Digital Synthesis. Center for Computer Research in Music Acoustics, Department of Music, Stanford University, Stanford, 2002.

STOCKHAUSEN, K. ...How time passes.... Die Reihe vol.3. Vienna: Universal Edition, 1957.

THOMSON, P. Atoms and errors: towards a history and aesthetics of microsound. Organised Sound 9(2). Cambridge University Press, 2004. pp.207-218.

TRUAX, B. Real-time Granular Synthesis with Digital Signal Processing Computer. Computer Music Journal 12(2). 1988. pp.14-16.

TRUAX, B. Composing with Real-Time Granular Sound. *Perspectives of New Music*, Vol.28, No.2. 1990.

VAGGIONE, H. Articulating Microtime. *Computer Music Journal*, Vol.20, No.2. 1996. pp.33-38.

VALLE, A.; LOMBARDO, V. A two-level method to control granular synthesis. *Proceedings of the XIV colloquium on musical informatics*. Firenze: 2003.

WAISWISZ, M. The Hands: a Set of Remote MIDI Controllers. *Proc. ICMC*, 1985. pp313-318.

WANDERLEY, M. M.; DEPALLE, P. Gestural Control of Sound Synthesis. *Proceedings of the IEEE*, Vol.92, No.4. 2004.

WESSEL, D.; WRIGHT, M. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, vol. 26. 2002. pp.11–22.

WIRT, A. Notes on the simulation of evolution. *IEEE Transactions on Neural Networks*, Vol.5, No.1. 1994. pp.130-148.

WRIGHT, M.; FREED, A. Open SoundControl: A new protocol for communicating with sound synthesizers. *International Computer Music Conference Proceedings*. 1997.

XENAKIS, I. *Formalized Music: Thought and Mathematics in Composition*. Hillsdale: Pendragon Press, 2001.



Sociedade de Engenharia de Áudio Artigo de Congresso

Apresentado no 7º Congresso de Engenharia de Áudio
13ª Convenção Nacional da AES Brasil
26 a 28 de Maio de 2009, São Paulo, SP

Este artigo foi reproduzido do original final entregue pelo autor, sem edições, correções ou considerações feitas pelo comitê técnico. A AES Brasil não se responsabiliza pelo conteúdo. Outros artigos podem ser adquiridos através da Audio Engineering Society, 60 East 42nd Street, New York, New York 10165-2520, USA, www.aes.org. Informações sobre a seção Brasileira podem ser obtidas em www.aesbrasil.org. Todos os direitos são reservados. Não é permitida a reprodução total ou parcial deste artigo sem autorização expressa da AES Brasil.

Evolutionary Processes for Granular Synthesis in Extended Gabor Spaces

Fernando Falci^{1,2}, Adolfo Maia Jr.^{1,3}

¹Núcleo Interdisciplinar de Comunicação Sonora (NICS), UNICAMP

Cid.Universitária Zeferino Vaz, Campinas, SP, Brazil.

²Departamento de Música, Instituto de Artes, UNICAMP

Cid.Universitária Zeferino Vaz, Campinas, SP, Brazil.

³Departamento de Matemática Aplicada, IMECC, UNICAMP

Cid.Universitária Zeferino Vaz, Campinas, SP, Brazil.

fernando_ffs@yahoo.com.br, adolfo@nics.unicamp.br

ABSTRACT

This paper presents *EVOGrain*, a computer music system that generates and controls granular sounds in real time through a graphical evolutionary interface. We also introduce the *Extended Gabor Space (EGS)*, the theoretical framework for *EVOGrain*. EGS is composed by three two dimensional spaces, namely, *Gabor Space*, *Spatialization Space* and *Streaming Space*. An individual is a set of three rectangles evolving dynamically in EGS. Genetic Algorithms are defined in order to generate sequences of populations which are associated to granular sounds outputs. Finally, *EGS* is represented by a friendly graphical interface which can be very useful for electronic and electroacoustic musicians.

0 HISTORY AND TECHNOLOGY

0.1 Brief comments on the history of granular synthesis

Granular Synthesis (GS) is a sound synthesis method that focuses on the time domain, presented as an alternative to traditional methods like Fourier Additive Synthesis that focuses on the frequency domain. Sounds, timbres, textures and music are here obtained by the temporal sequencing of grains, tiny pieces of sound by the order of one to a hundred milliseconds. These grains are based in the concept of Gabor's Acoustical Quanta [1], the smallest audible particle of sound. As soon as this theory was published, many experiments about the limits of perception

of sound in time and frequency took place. These experiments explored the ambiguity effects present at the microtime level of sound like fusion of grains, masking of attacks, intensity perception and silence perception [2].

As a pre-digital history, granular synthesis follows the tendency stated by Webern's radical atomization of the music material [3], and continued at Darmstadt and Köln' schools of modernism that focused at the "point" as the smallest unit of music, as opposed to the "note". In 1957 a unified view of various time scales of music structure was proposed by Stockhausen in his landmark article "...how Time Passes..." [4]. In 1958, Xenakis presented his pioneer granular composition "Concrete P.H." made by the splicing and collage of tiny pieces of magnetic tape. He was the first to use the term *Microsound* to refer compositions

where the composer's control acts at the level of the acoustical quanta [5].

Although it is possible to make granular music with analogical resources, and even with acoustic or natural sounds (e.g. all strings playing pizzicato, maracas and shakers, waterfalls), microsound is more idiomatic and versatile to the digital domain, where many contributions have been made by Curtis Roads [2] (several systems for granular sequencing, granulation and granular analysis) and Barry Truax [6] (real-time control of granular synthesis and granulation). More recently, DiScipio [7] applied physics and mathematical models like fractals, dynamic systems and chaos theory to systems for generation and control of grains. In this direction also Maia and Miranda [8] developed a system in MATLAB named FuzzKov which use Fuzzy Sets and Markov Chains. Theory of graphs as a control structure has also been proposed by Valle [9]. The excellent book "Microsound", by Roads [2], provides a complete account of the history of granular synthesis up to 2000.

0.2 Evolutive computation in music

One common challenge for frequency-based synthesis of digital sound is how to enhance the sound output giving it the liveness of natural sounds. Natural sound presents a timbre evolution through time that is more easily synthesized through time based techniques. One way to obtain such characteristics is specifying a huge amount of data parameters to define and control streams of hundreds of grains. This is very expensive for processing, mainly in real time streaming. A way out from this state of affairs is achieved by using genetic algorithm to control those numerous parameters.

As well known genetic algorithms (GA) are computer programs inspired in the Darwin's evolution of the species theory [10]. In this model, it is necessary to define an individual as a possible solution and a population as a set of individuals. It is also necessary to define genetic operators like crossover, mutation and a fitness function that measures the potential of each individual to be the desired solution. As genetic operators are applied, new generations of individuals are created. The algorithm halts when a new population contains an individual that solves the problem or, alternatively, when certain number of generations has been reached.

Musical applications of genetic algorithms have been proved to be a successful approach as we can seen in VoxPopuli [11], a GA software for composition that runs over the MIDI protocol, and also in EESynth [12] a software that combines waveforms until a target waveform is obtained. Both these systems were developed by researchers from NICS/ UNICAMP.

In EVOGrain v1.0, we take a six dimensional space named Extended Gabor Space (EGS), where the

parameters associated to a family of grains (an individual) are defined inside three rectangles drawn in the EGS (Figure 1). In addition a target individual is specified by hand. Once the evolution process begins, instead of finding (and hearing) only the final solution, we can hear, in time, the winner individual of each generation calculated by the algorithm. In other words in this approach we can hear the timbre evolution. In next section we show details of our GS system.

1 EVOGRAIN V1.0

EVOGrain is a system, currently in development at NICS - UNICAMP, that generates and controls granular sounds in real time through a graphical interface, consisting of three windows, each one of them interpreted as a 2-dimensional real parameter space (R^2). Using the mouse, the user draws one rectangle in each R^2 window. These three rectangles define a target individual which will drive the evolution of the sequence of generations through a genetic algorithm. For each generation an individual has a set of six parameters which define its genotype, namely, *frequency band, grain duration, flux density interval, granulation offset, pan and spatial distance*. This parameters form a vector in a 6-dimensional space which is interpreted as the individual genotype. In this way this genotype coded as a six parameters vector is used to control the granular synthesis process in real time.

EVOGrain is divided in three main modules where many threads are running in parallel. We describe them in the following sections: Real Time Granular Synthesis Machine, The Genetic Algorithm and The Evolutive Interface.

1.1 Real Time Granular Synthesis Machine

At the core of the system runs the real time synthesizer which is in charge for basic functions like accessing the soundcard resources, and also for generating, storing and playing samples. The real time synthesizer is currently working with two independent streams of grains, each with sample rate of 44.100Hz, 16 bits depth and 2 channels.

The Real Time Granular Synthesis Machine architecture is built under the producer/consumer paradigm. The producer thread is responsible for calculating samples and storing them in a buffer. The consumer thread waits until the buffer is completed and then sends its samples to the computer soundcard. Each stream has its own producer/consumer threads as well as access to the soundcard output line. The best situation, in order to achieve no latency, would be to send each generated sample directly to the soundcard, but this causes buffer under run resulting in clicks and sound interruptions. After some experiments we found the latency of 500ms a good balance between liveness and sound quality.

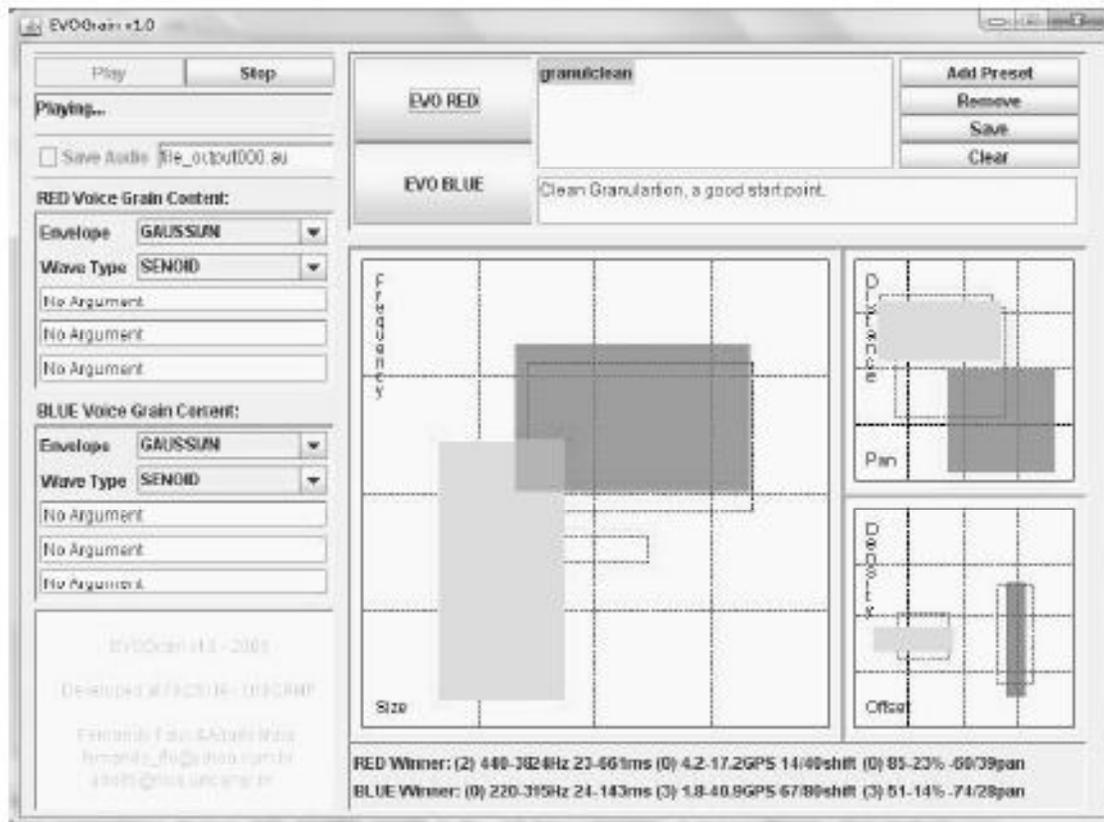


Figure 1 EVOGrain v1.0 interface

As mentioned above, each stream is controlled by six pairs of dynamic parameters that may be updated at any time. In section 2.3 we show that the dynamic parameters are indirectly controlled by the rectangles drawn by the user. The dynamic parameters are defined in intervals with an initial and a final value. Each new synthesized grain inside an individual (a family of grains) will be created with six parameter values sorted, with uniform statistical distribution, within the intervals defined by the rectangles drawn on the interface. The dynamic parameters are:

- *Frequency Interval*: the frequency of the waveform for the grain. If the waveform is being read from a wavetable, this parameter will determine the reading speed;
- *Duration Interval*: the time size of a single grain to choose within this interval;
- *Flux Density Interval*: the speed at which grains will be created, measured in grains per second;
- *Offset Interval*: this parameter is used only in case of granulation of a waveform. It determines the offset position to read the next sample of a wavetable;
- *Spatial Distance Interval*: the distance from the loudspeaker to the listener. It is going to affect the grain loudness;
- *Pan Interval*: the stereophonic positioning of the grain.

Each stream is also controlled by static parameters that determine grain envelope and grain content, they must be

set before playing the machine, so these parameters cannot be changed in real time. Figure 2 shows the possible options for grains envelope in this version of EVOGrain, from left to the right:

- *Gaussian*, a smooth attack and decay;
- *Fat Gaussian*, a wide bell shaped Gaussian;
- *Expoder*, decreasing exponential, a very percussive envelope;
- *Repodex*, exponential;
- *Thin Gaussian*, little bit more percussive than Gaussian envelope;
- *No Envelope* at all.



Figure 2 Grain Envelopes

The possible grain content is listed below:

- *Sinusoid*, one sinusoid wave;
- *Glisson*, one sinusoid glisson with specified interval. It can be upwards or downwards;
- *Addsinusoids*, additive synthesis with sinusoids. User can specify frequencies and amplitudes for all partials;

- *Square*, one square wave;
- *Random*, white noise;
- *Granulator*, allows the user to granulate a selected wave stereo file, this opens possibilities to a wide range of waveforms.

The producer thread is constantly running controlled by the six parameters described above. According to flux density and to grain size it may result in the overlapping of grains and consequently the formation of clouds in the sense of Xenakis [13]. In order to obtain each sample, the producer algorithm executes the following steps:

1. Check for new values for the six pairs of parameters;
2. Sort (uniform distribution) current values for frequency, grain size, flux density, offset, distance and panoramic within the range specified by the parameters, this parameters will be used to create the next grain;
3. According to flux density, verifies if it is time to create a new grain. In positive case, creates new grain with current values as specified above, and places it in an array of active grains;
4. For each active grain, get the next sample, apply envelope factor and calculate the stereo samples for the left and right channels according to specified pan;
5. For each active grain, if the end of the grain has been reached it is removed from the array of active grains;
6. Calculates the final sample by adding the current sample of all active grains for the left and right channels and includes it in the sample buffer.

The consumer thread is a very simple algorithm:

1. Wait while the buffer is filled with samples;
2. Get a completed 500ms long buffer;
3. Send the samples to the computer soundcard.

Next section presents the genetic algorithm and explains how it receives information from the user interface, and how it computes the parameters that are sent to the real time granular synthesis machine.

1.2 The Genetic Algorithm

In EVOGrain we have implemented a simple and configurable genetic algorithm. Before starting evolution machine, the user can set up the population size (number of individuals in each generation), number of survivors (determines how many individuals will be kept to the next generation and how many will be replaced), mutation rate (the chance for a new born individual pass through mutation, measured in percent) and the pace of evolution (how long to wait between genetic cycles, in milliseconds).

The part of genotype of an individual in this 2-dimensional component is here defined as an ordered set of four floating point numbers ranging from 0 to 1000, that is, $R = (t_1, f_1, t_2, f_2)$. Now, a target individual must be specified in order to construct an evolutive process. We denote it as $R^* = (t_1^*, f_1^*, t_2^*, f_2^*)$. The representation for

an individual is a rectangle with vertical and horizontal sides in a bi-dimensional space as show in Figure 3 below. EGS is a 6-dimensional space which in order to be described in a plane interface must be splitted in three 2-dimensional spaces, that is, EGS has a structure of $R^2 \times R^2 \times R^2$. Now we discuss some details of each single 2-dim space.

In our model, a population is a set of rectangles. The initial population is created with random values sorted in the interval [0-1000] for all floating point numbers of all individuals. In each genetic cycle the algorithm computes the fitness value of all individuals gets the winner and keeps some survivor individuals. The next step is to apply crossover on the survivors in order to generate children and to apply the mutation operator to the new born.

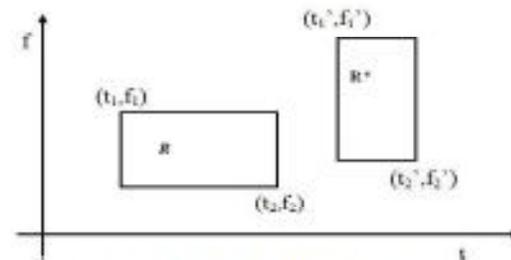


Figure 3 Individuals in time x frequency Gabor Space

Now we define a distance function between two individuals (rectangles in 2-dimensional Space) as:

$$d(R, R^*) = \max\{|t_1 - t_1'|, |f_1 - f_1'|, |t_2 - t_2'|, |f_2 - f_2'|\} \quad (1)$$

This is the well known norm of the maximum (or max norm) in the theory of metric spaces [14]. This distance function is a very simple metric and it makes sense here since all parameters are defined in the real interval [0, 1000]. Now we define our fitness function as:

$$f(R, R^*) = 1 / (1 + d(R, R^*)) \quad (2)$$

Observe that this fitness function is normalized, that is $0 \leq f(R, R^*) \leq 1$. Also it satisfies the requirement $d(R, R^*) = 0 \Rightarrow f(R, R^*) = 1$ (total fitness).

After calculating the fitness of all individuals, the algorithm will determine which of them is the winner, that is, is which is the one with the greatest fitness value. The winner parameters are immediately passed as parameters to the real time synthesis machine and to the evolutive interface. The algorithm will also select the best individuals to include in the group of survivors.

In our model, the population size is kept constant for all generations, so in each genetic cycle some individuals must be discharged and new individuals should be created (the number of individuals to be created is equal to population size less the number of survivors). Individuals are created by crossover between survivors only. Crossover takes place by choosing two survivor individuals and creating a new one whose values are the mean of the parents' values:

$$\text{Parent1} = (t_1, f_1, t_2, f_2) \quad (3)$$

$$\text{Parent2} = (t_1', f_1', t_2', f_2') \quad (4)$$

$$\text{Child} = ((t_1+t_1')/2, (f_1+f_1')/2, (t_2+t_2')/2, (f_2+f_2')/2) \quad (5)$$

After crossover, the mutation operator is applied. For each new born individual a floating point value is uniformly sorted in the interval [0,1]. If this value is smaller than the mutation rate, set by the user, then mutation takes place. The mutation operator chooses one of the four numbers from the genotype of the child and overwrites it by a random value in the interval [0,1000].

After mutation the new generation is completed, the algorithm then halts in order to wait a short time (around some hundreds of milliseconds) specified by the evolution pace before starting the next genetic cycle.

1.3 Evolutive Interface

EVOGrain v1.0 presents a friendly interface, which can be very useful for the electronic or electroacoustic musician. As Figure 1 shows, it has three different R^2 spaces (white squares in the interface) where the user can draw target rectangles with the mouse. The rectangles can be of two colors, mouse left button for red rectangles (light gray), mouse right button for blue rectangles (dark gray). These colors index two different streams of grains.

Since, in each R^2 parameter space runs two genetic algorithms, the system runs six independent genetic algorithms at once, or equivalently three for each stream (blue and red). While the genetic algorithm is evolving, the winner in each generation is shown in the interface spaces as a dashed rectangle which moves converging to its target.

The coordinates of the dashed rectangles, the winner of each generation, are read as parameters that controls the real time granular synthesis machine. EVOGrain v1.0 is currently synthesizing two streams of grains, one represented by the three red moving dashed rectangles and the other by the blue ones. These are what are being really played by the synthesizer. In the overall process, the user hears the sound stream evolving while the dashed rectangles visually converge to the target ones.

As mentioned above, EVOGrain interface is splitted in three R^2 spaces. The left R^2 is the original *Gabor Space* and controls grain duration and frequency, the right upper is the *Spatialization Space* (pan x distance). The right lower one controls grain sequencing and is named *Streaming Space* (density x offset). So, the Cartesian product of these three spaces we named *Extended Gabor Space*. Every time a new winner arrives in any of the three component spaces the stream parameters are updated.

1.4 Overall System Architecture

After presenting details of the three modules that compose EVOGrain v1.0, Figure 4 describes the overall system architecture. From the musician standpoint, the system is controlled with the mouse either by drawing target rectangles in EGS or through loading rectangles presets.

User visual feedback is shown at time as dashed rectangles in the three EGS spaces; feedback is also given to the composer as well to the audience as the sound stream of granular sounds that arrives in stereophonic speakers within 500ms latency. User input, that is, target rectangles, is received by the Evolutive interface and passed to the Genetic Algorithm.

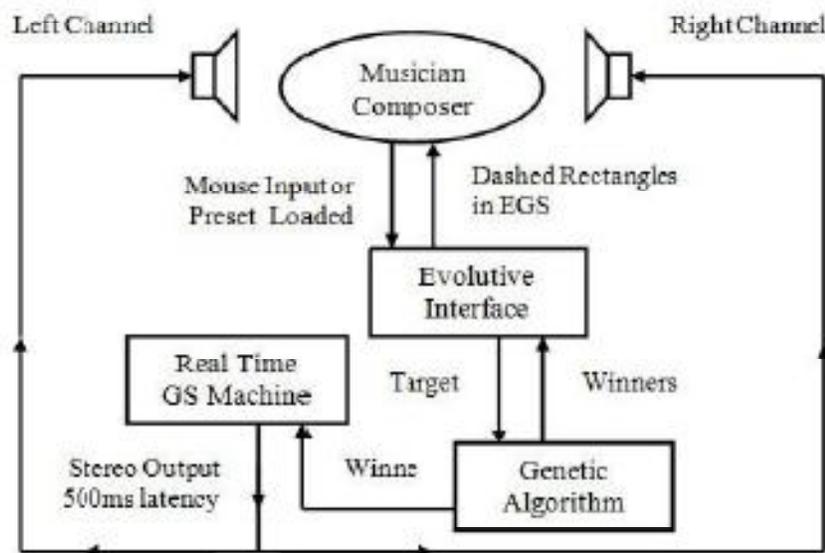


Figure 4 EVOGrain System Architecture

The fitness value of the individuals calculated in the next genetic cycle will be calculated considering the new user set targets. As soon as a new winner arrives in any R^2 space of the 6-dimensional Extended Gabor Space its coordinates are sent back to the Evolutive Interface Module that draws the dashed winner rectangle on the screen.

Every new winner in the Genetic Algorithm Module is also sent to the Real Time Granular Synthesis Machine as new parameter values that control one stream of grains. The new grains synthesized by the real time machine responds immediately to the new parameters, but due to the buffer size it can be heard only after half second.

It is worth to mention that sound dynamics is a direct result of grains spectral content and of the variability of the evolutive process, which, in turns, depends on the definition of the crossover and mutation operators. New definitions for these operators acting on the parameters of the system can lead to new set of sound streams.

2 PERFORMANCE

Microsound opened new horizons not only to sound synthesis but also to music composition. As well known, from the microsound time level, frequency and time are closely related. The idea is to explore this relationship, specially the twilight region between rhythm and pitch. Also, composing with microsound should give emphasis on the sound transformation process in time rather than on the establishment of rigid sound objects [15].

EVOGrain implements an interesting and exclusive solution to control the transformation process and gives life to the synthesized textures. When targets are set by the user, the evolving genetic algorithm periodically updates the synthesis parameters; the result is that the timbre is constantly evolving in time. Up to its latency, if new targets are draw on the EGS, the algorithm recalculate automatically the process of evolution.

The graphical interface presented in EVOGrain allows the user to conduct the texture on the Extended Gabor Space in a much intuited manner. We have made a systematic, but not exhaustive, exploration of EVOGrain possibilities. A lot of sound effects, rich timbres and textured sound streams come out from this system.

As limit cases, not only rectangles can be drawn but also points and horizontal or vertical lines. Points result in one single value for both rectangle parameters (e.g. frequency and grain size). Lines results in one single value parameter while the other one still runs randomly within the range draw by the user. This is especially important if you are manipulating density in order to obtain a synchronous or asynchronous stream (a single value for density, for example, results in a synchronous stream).

Another way is to think about frequency layers. Each stream will leave its own spectral blurring. For example, if the user draws a vertical line in the time x frequency space

(left area); a bandwidth is set as target. As grain frequency will be uniformly sorted within its range and after a short period of time, the whole bandwidth will be filled with grains sorted among all frequencies of the interval (vertical line).

It's worth mentioning that, to help live performance and composition process, EVOGrain allows the user to save and load a rectangle configuration preset. Anytime, the target rectangle configuration can be saved, for later use, with any name such as, for example, "Introduction", "Begin of Part A", "Bridge Section", "Cloud Explosion", "Coda", and so on.

3 FUTURE WORK

As EVOGrain is currently in version 1.0, many features can be improved. We would like to increase the number of streams to an arbitrary number of voices and, in order to achieve such goal the performance of the real time granular synthesis algorithm should be improved or even be converted to a compiled programming language (C, C++) instead of a interpreted one (JAVA).

Also, spatialization should be extended from this stereophonic environment to a multiphonic one. The square spatial drawing area should be replaced by a circle where the center is the listener position. It is possible also to implement grain envelopes which can be pre-defined or draw by the user.

The Gabor model is a natural space to define evolutive processes in order to generate dynamical granular sounds streams. Nevertheless we can apply the same model for microsounds in the sense of Roads [5], which are defined in a more flexible manner. In addition, spaces (and also the fitness function) can include more variables such as harmonicity, rugosity and brightness, borrowed from psychoacoustics [16], as a top level composing aim. This, of course, implies to new extensions of the Gabor Spaces to higher dimension, which can be endowed also with different metrics and fitness functions.

Another important improvement is related to probability distribution for sorting the parameters values for grains. Only uniform distribution was used in EVOGrain v1.0. Clearly, more general classes of probability distributions can be incorporated to the model and to the code.

We think is also possible to define family of grains associated visually with new geometrical forms such as triangles, circles or any other shapes. Further studies in visual arts could lead the user drawings closer to, say, a Kandinsky picture, which will be associated to more dynamical control of the sound parameters.

Finally, as mentioned above, at the beginning of Section 0.2, we intend our method can produce synthesized live sounds as those ones from nature. Of course, this is a subjective aspect and it implies that subjective tests must be performed. However, EVOGrain is a system yet under construction. So, we think that after implementation of the

new aspects above mentioned the system will be able to be submitted to subjective tests. We would like to thank an anonymous referee to call our attention to this point.

4 ACKNOWLEDGEMENTS

This research was partially supported by FAEPEX (Fundó de Apoio Ao Ensino, à Pesquisa e à Extensáo) of UNICAMP (University of Campinas). A. Maia Jr. thanks CNPq (National Council for Scientific and Technological Development) for a research grant in Computer Music.

5 REFERENCES

- [1] Gabor, D.; *Acoustical quanta and the theory of hearing*, Nature 159(4040), pp. 591-594 (1947).
- [2] Roads, C.; *Microsound*, MIT Press, Cambridge, MA (2001).
- [3] Thomson, P.; *Atoms and Errors: towards a history and aesthetics of microsound*, Organized Sound 9(2), pp. 207-218, Cambridge Press (2004).
- [4] Stockhausen, K.; *...How time passes...*, Die Reihe vol.5. Universal Edition, Vienna (1957).
- [5] Xenakis, I.; *Formalized Music: Thought and Mathematics in Composition*, (Harmonologia Series No.6). Pendragon Press, Hillsdale, NY (2001).
- [6] Truax, B.; *Real-time Granular Synthesis with Digital Signal Processing Computer*, Computer Music Journal 12(2), pp. 14-16 (1988).
- [7] DiScipio, A.; "Composition by Exploration of Nonlinear Dynamical Systems", Proceedings of the 1990 International Computer Music Conference. International Computer Music Association, San Francisco, pp. 324-327 (1990).
- [8] Maia Jr., A.; Miranda, E.R.; *Granular synthesis of sound through Markov chains with fuzzy control*, Annals of the International Computer Music Conference (ICMC), Barcelona (2005).
- [9] Valle, A.; Lombardo, V.; *A two-level method to control granular synthesis*, Proceedings of the XIV Colloquium on Musical Informatics, Firenze (2003).
- [10] Wirt, A.; *Notes on the simulation of evolution*, IEEE Transactions on Neural Networks, Vol.5, No.1, pp. 130-148. See also <http://www.aicresearch.com/research/notes.html> (1994). Last access in 24/02/2009.
- [11] Moronis, A., et al.; *VoxPopuli: evolutionary computation for music evolution*, Proceedings of the AISB'99 Symposium on Creative Evolutionary Systems - CES'99, Edinburg, Holland (1999).
- [12] Fornari, J.; Manzolli, J.; *ESSynth - Estudo e Desenvolvimento de um Sintetizador Evolutivo de Segmentos Sonoros*, PosDoc research sponsored by FAPESP (www.fapesp.br), process: 04/00499-6 (2004-2007).
- [13] Clark, M.; *Composing as the intersection of time and frequency*, Organized Sound 1(2), pp. 107-117. Cambridge University Press (1996).
- [14] Bryant, V.; *Metric Spaces: Iteration and Application*, Cambridge University Press (1985).

[15] Smalley, D.; *Spectromorphology: explaining sound-shapes*, Organized Sound 2(2), pp. 107-126. Cambridge University Press (1997).

[16] Roederer, R.; *Introdução à física e psicofísica da música*, Edusp, São Paulo, (1998).



Sociedade de Engenharia de Audio Artigo de Congresso

Apresentado no 8º Congresso de Engenharia de Audio
14ª Convenção Nacional da AES Brasil
4 a 8 de Maio de 2010, São Paulo, SP

Este artigo foi reproduzido do original final entregue pelo autor, sem edições, correções ou considerações feitas pelo comitê técnico. A AES Brasil não se responsabiliza pelo conteúdo. Outros artigos podem ser adquiridos através da Audio Engineering Society, 60 East 42nd Street, New York, New York 10153-2126, USA, www.aes.org. Informações sobre a seção Brasileira podem ser obtidas em www.aesbrasil.org. Todos os direitos são reservados. Não é permitida a reprodução total ou parcial deste artigo sem autorização expressa da AES Brasil.

An Input Device to Control Granular Sound Processes

Fernando Falci de Souza^{1,2}, Marcelo M. Wanderley⁴, Adolfo Maia Jr.^{1,3}

¹Núcleo Interdisciplinar de Comunicação Sonora (NICS)

²Departamento de Música (IA)

³Departamento de Matemática Aplicada (IMECC)

UNICAMP, Campinas, SP, Brazil.

⁴Input Devices and Music Interaction Laboratory - Music Technology Area

McGill University, Montreal, QC, Canada

fernando_ffs@yahoo.com.br, adolfo@nics.unicamp.br, marcelo.wanderley@mcgill.ca

ABSTRACT

In this paper we present the next state of the art of our previous work on the control of granular synthesis through evolutive processes occurring in Extended Gabor's Space. We describe the GranularStreamlet, a new version of EVOGrain's granular sounds synthesizer, and a suitable coupling to T-Stick, an input device which can control synthesis parameters in real time. This work is technologically oriented to achieve a Digital Musical Instrument which can be easily explored by a musician to trigger granular sound transformation processes according to Smalley's Spectromorphology through a gestural interface.

0 INTRODUCTION

Granular Synthesis (GS) is a sound synthesis method firstly based on the concept of Gabor's Acoustical Quanta [1], the least energetic particle of sound that can be detected by an auditory device, such as, for example, the human ear. With this method, sounds, timbres, textures and music are obtained by temporal sequencing of tiny particles of sound by the order of one to a hundred milliseconds, which is nowadays named grains of sound [2,3,4]. A commonplace challenge to design sound synthesis systems operating at the microtime level is to find appropriated algorithms and devices in order to control streams with hundreds of grains per second and to get high-level structures generated by sometimes a huge

amount of data. Recently we presented a prototype system [4], named EVOGrain which generates and controls streams of granular sounds in real time through a graphical interface, consisting of three windows, each one of them interpreted as a 2-dimensional real parameter space (R2). The three windows together encompass a 5-dimensional parameter space we named Extended Gabor Space. Using the mouse, the user draws a rectangle in each R2 window that defines a target in its respective two-dimensional space which will drive the evolution of a time-ordered set of generators. The winner individual of each generation, that is, the one closest to the target rectangle according to the metric (norm of the maximum) [5], defined in the extended

Gabor Space, is interpreted as control parameters of a stream of granular sounds.

This paper describes some recent efforts to find an alternative interface to EVOGrain focusing on gestural controllers. Although Granular Synthesis is a well known electronic music technique, its use in live performance is not widely spread, specially when controlled by digital instruments. The particularity of this sound material, that is, grains of sound, place its own difficulties on the development of an input device like the Poseidon [6,7], a digital controller that makes use of sliders and pressure pads to control granular synthesis.

This research is a collaboration of the Interdisciplinary Nucleus of Sound Communication (NICS), UNICAMP, Brazil, and the Input Devices and Music Interaction Laboratory (IDMIL) from McGill University, Canada. The following sections describe, shortly, the core of the granular synthesis system, some early experiments with commercial and academic input devices, and finally, propose the development of a new input device suitable to control granular sound transformation processes.

1 GRANULAR STREAMS AND PROCESSES

In this section we present the sound synthesis algorithm used in this paper, the GranularStreamer (a new version of EVOGrain, our Real Time Granular Synthesis Machine). It also features a new network layer that receives parameter updates in real time through Open Sound Control (OSC) messages. Next we discuss the expected spectromorphological behavior of the granular sounds that we intend to control in real time with an input device or gestural controller.

1.1 GranularStreamer

As its antecessor EVOGrain, the GranularStreamer is a granular synthesizer that generates multiple streams of grains in real time, currently working with sample rate of 44.100Hz, 16 bits depth and 2 channels. It uses the RtAudio library, a set of real-time audio i/o classes, to access the system's audio devices [8].

To achieve better performance the source code of the GranularStreamer was converted from JAVA to C++, and some fine tuning was made in the algorithm (e.g. better use of system memory, optimization of the real time loop, use of 1024-points wavetables with two-point linear interpolation for all wave and envelope types). Now it's possible to synthesize many simultaneous streams of grains limited only by the system's resources. The Real Time Granular Synthesis Machine architecture is built under the producer/consumer paradigm. The producer thread is responsible for calculating samples and storing them in a buffer. The consumer thread waits until the buffer is completed and then sends its samples to the computer soundboard.

To run GranularStreamer the user should type a command line that specifies the number of streams of grains, the maximum number of overlapping grains (shared for all streams) and the wave and envelope types for each stream. The currently available wave types are Sine, Square, Sawtooth and Granulation [2]. The latter allows the user to granulate a selected wave mono file, this opens possibilities to a wide range of waveforms. The available

envelope types are Gaussian, Exponential, Rectangular and None [2].

Each stream is controlled by six pairs of dynamic parameters (initial value and final value for each parameter) that may be updated at any time. Every new synthesized grain will be created with six parameter values sorted, with the uniform statistical distribution, within the intervals defined by each pair of parameter. To control the synthesis parameters in real-time we developed an OSC message listener using the `oscpack` library [9]. OSC is a communication protocol for messaging among computers, sound synthesizers and multimedia connected by computer networks or USB ports. In this way, GranularStreamer can be easily integrated to any input device or sound system capable of sending OSC messages as, for instance, PureData patches or the Mapping Tools described next section. A complete documentation of the OSC messages supported by the GranularStreamer can be found at <http://www.nics.unicamp.br/~fernando/granularstreamer.html>. The real time parameters that GranularStreamer currently accepts for each stream are listed below:

- Initial and Final Frequency: the frequency of the grain waveform in Hertz;

- Initial and Final Grainsize: the grain size in milliseconds;

- Initial and Final Density: the flux density of the stream in grains per second. If Initial and Final Density are equal, a synchronous stream is obtained, otherwise the density of the flux will be changed after the creation of every grain.

- Initial and Final Offset: if a wavefile is being used as waveform, the offset parameter is used to define the reading position within the wavefile.

- Initial and Final Amplitude: the amplitude of each grain.

- Initial and Final Pan: the panoramic position (left/right) for each grain;

In addition to the parameters above the system can be controlled by a secondary set of messages called "width parameters" where, instead of specifying an initial value and a final value for each, it specifies one center value and the width of the interval. For instance, to specify the frequency of a cloud of grains the system can be controlled by the parameters Center Frequency and Bandwidth or, in the case of Amplitude, Center Amplitude and Amplitude Variation.

1.2 Granular processes and spectromorphology

Microsound opened new horizons not only to sound synthesis but also to music composition. Composing with grains should give emphasis on the sound transformation process in time rather than on the establishment of rigid sound objects [10]. As mentioned above, the prototype system EVOGrain is presented as a graphical interface that allows the user to conduct the texture in the Extended Gabor Space in an intuitive manner and it doesn't require absolute control of the process. Here we are looking for a Digital Musical Instrument (DMI) with the same characteristics of EVOGrain that could be useful to trigger a wide variety of granular sound transformation processes like those present, for example, in the famous composition *Riverrun* of Barry Truax [11], where each stream leaves its

own spectral blurring. Motion and growth processes are very suitable to describe that variety of granular sounds transformations as stressed in the quote below by Smalley:

The metaphors of motion and growth are appropriate ways of considering a time-based art like electroacoustic music. Traditional concepts of rhythm are inadequate to describe the often dramatic contours of electroacoustic gesture and the internal motion of texture which are expressed through a great variety of spectromorphologies. Quite often listeners are reminded of motion and growth processes outside music and the terms selected are intended to evoke these kinds of connections. Since motion and growth have spectral contours, they are set in spectral space. Therefore their occupancy of spectral space and their spectral density will be important additional qualifiers" [12].

For those ends we describe a list of motion and growth processes according to Smalley's Spectromorphology [11] that will serve as reference to the evaluation of the proposed DMI:

- Linear: a linear ascendent or descendent progression;
- Parabolic: an ascendent function followed by an descendent function and vice-versa;
- Cyclic/Spiral: an oscillatory movement described by a circular or spiral trajectory.



Figure 1 Spectromorphological Processes

These processes describe the position occupied by a particle in a 2-dimensional trajectory in certain moment of time. Projecting the position of the particle to Cartesian coordinates X and Y, we can obtain a parameter variation in time that can be scaled to any parameter of the 6-dimensional Extend Gabor's Space (grain frequency, grain size, stream density, wave offset, amplitude and panoramic position). If we predefine the trajectories, the overall time of the process will be determined by the simulated speed of the particle. For instance, the X coordinate of the above parabolic trajectory (Fig.1) can be scaled to stream density resulting in a increasing density of grains per second, while the Y coordinate can be scaled to grain frequency resulting in a process of rising and then lowering the frequency of the grains. The X and Y coordinates can also be scaled to any other parameter like those in the set called "width parameters" resulting in processes of agglomeration, dissipation, dilation and contraction. For example, projecting the X coordinate of a cyclic trajectory to the parameter that controls the bandwidth of a cloud we obtain an oscillatory process of dissipation (while the grains are being scattered within an increasing bandwidth) and contraction (while the bandwidth decreases and the grains concentrates around the center frequency). So the class of curves above, can be mapped in any of the control parameters leading to complex sound streams mediated by suitable Input devices. Audio examples of these processes synthesized by the GranularStreamer can be found at <http://www.nics.unicamp.br/~fernando/aes2010.html>.

1.3 Mapping Tools

In this section we present Mapping Tools, a Max MSP patch that makes easy to connect source DMIs to sound parameters, followed by the T-Stick, a gestural music controller used in this paper to do preliminary experiments controlling granular processes with gestures.

Mapping Tools is a collaborative development system that allows mapping between controllers and sound parameters in a simple and fast way being especially useful when connecting a large group of controllers to sounds parameters or accelerating the exploration of a suitable controller to a new sound synthesis algorithm [13].

The Mapping Tools interface is divided into three sections. The left section shows the Sources, that is, all controllers connected to the system, e.g., SpaceNavigator, T-Stick, Keyboards, Joysticks, MIDI PADs and others. The right section shows the Destinations, that is, all sound generators connected to the system, e.g., any MAX MSP or PD patch, or our GranularStreamer. In the center of the interface, between the Sources and the Destinations are located the Connections. A connection is established between a source and a destination by click on the controller and on the sound system. After doing a connection, the mapper shows the details of the controller, that is, the list of all available sensors, and the list of all sound parameters. Then it is possible to connect each sensor to each sound parameter using an appropriate user-defined expression.

The Mapping Tools is in charge of receiving the controller OSC messages with sensor values, evaluating the conversion expression, and sending OSC messages with the sound parameter values to the sound system. Details on functioning of Mapping Tools and its documentation can be founded in the page <http://www.idmil.org/software/mappingtools>.

1.4 The T-Stick

The T-Stick is a gestural controller developed by J. Malloch at the IDMIL (Input Devices and Music Interaction), Music Technology Area in McGill University [11]. It is a DMI equipped with accelerometers, pressure sensors, touch sensors and other sensors. Its outputs determine how much or how strong the instrument is being touched, shaken, twisted, tilted, etc.



Figure 2 The T-Stick

The T-Stick offers a wide range of possibilities to control the GranularStreamer. It can be used like a

performer instrument. For example, mapping the 'shake' parameter simultaneously into stream density (grains per second) and amplitude we obtain a granular shaker, and it can also be used as a conductor instrument triggering sound transformation processes like those described in section 1.2. To achieve this we defined a gesture to be performed with the T-Stick that is, holding the T-Stick with both hands at one extremity, simulate a hit in direction of the ground. The hit can be done in a straight line downwards, with slight semi-circular bounce at the end or with a full circular movement at the end.

The stick is equipped with a 3 axis accelerometer at the free extremity of the stick, that is, the extremity opposed to the one being held. After extracting and processing the data received from the accelerometer we obtained two normalized parameters that are used to trigger the granular processes. The first, "Hit Power", is controlled by the intensity of the hit, and the second "Hit Spin" is determined by the amount of circular movement performed at the end of the hit.

These parameters are then sent to a middle layer named "trajectory layer" where we codified two fixed trajectories (linear and circular) with same length of 1000 units. The "Hit Spin" parameter is used to determine the trajectory; if "Hit Spin" < 0.5 the trajectory will be linear; if "Hit Spin" ≥ 0.5 the trajectory will be circular. The "Hit Power" parameter is used to control the speed of the process, so we scale "Hit Power" in such a way that; if "Hit Power" = 0 the complete linear or circular process will be 10 seconds long and if "Hit Power" = 1 the complete process will be one second long. The trajectory layer is then responsible to calculate the X and Y axis projections and sending this values as parameters updates to the GranularStream.

2 CONCLUSIONS AND PERSPECTIVES

In the experiments that we did in this paper we reduced the possibilities of the spectromorphological trajectories (linear and circular) and used only two gesture inputs (hit power and hit spin). Although we realized that the gesture proposed to be performed with the T-Stick is not very suitable because is hard to have a satisfactory control of power and spin, the sound processes obtained using the model proposed above strongly suggests that rich granular synthesis transformations can be triggered by a gesture performed with a DMI.

In order to extend the results obtained in this paper we begin the construction of a new DMI suitable for triggering processes of granular sounds transformation but with a simple and intuitive interface [15]. This device is an individual alternate controller metaphorically inspired in the shape of a baseball ball with greater dimensions (about 15 cm of diameter). The user is going to trigger the sound transformation processes by performing the gesture of throwing the ball toward, downward or upward; with different angles and spin effects. The DMI will be equipped with a 3-axis accelerometer, a gyroscope to sense spin effect and a pressure sensor. The user should also be provided with selectors to choose the active stream and the combination of synthesis parameters that are going to be updated by the "trajectory layer".

The "trajectory layer" itself can then be upgraded to compute multiple trajectories with variable shapes, lengths and speeds according to the parameters of the throw

performed with the DMI like throw power, elevation, spin effect and pressure increasing the variety and richness of the granular sound transformation processes.

3 ACKNOWLEDGEMENTS

This research was supported by ELAP (Emerging Leaders in the Americas Program), International Scholarships, Government of Canada. Fernando F. Souza thanks McGill University and IDMIL - Input Devices and Music Interaction Laboratory for financial support and for the pleasant research atmosphere.

This research was also partially supported by FAPESP (Fundo de Apoio Ao Ensino, à Pesquisa e Extensão) of UNICAMP (University of Campinas). A. Maia Jr. thanks CNPq (National Council for Scientific and Technological Development) for a research grant in Computer Music.

BIBLIOGRAPHY

- [1] Gabor, D.; Acoustical quanta and the theory of hearing, *Nature* 159(4040), pp. 591-594 (1947).
- [2] Roads, C.; *Microsound*, MIT Press, Cambridge, MA (2001).
- [3] Thomson, P.; Atoms and Errors: towards a history and aesthetics of microsound, *Organized Sound* 9(2), pp. 207-218, Cambridge Press (2004).
- [4] Souza, F.F.; Maia Jr., A.; Evolutionary Processes for Granular Synthesis in Extended Gabor Spaces, 7o Congresso de Engenharia de Audio, AES Brasil, São Paulo, SP (2009).
- [5] Bryant, V.; *Metric Spaces: Iteration and Application*, Cambridge University Press (1985).
- [6] Opie, T.; *Creation of a Real-Time Granular Synthesis Instrument for Live Performance*, Mater Thesis, Brisbane, Australia (2001).
- [7] Opie, T.; *Granular Synthesis: Experiments in Live Performance*, Proc. ACMA, Fitzroy, Australia (2002).
- [8] Copyright (c) 2001-2010 Gary P. Scavone. <http://www.music.mcgill.ca/~gary/rtaudio/index.html>
- [9] Copyright © 2004-2006 Ross Bencina. <http://www.audiounlch.com/~rossb/code/oscpack/>
- [10] Clark, M.; Composing at the intersection of time and frequency, *Organized Sound* 1(2), pp. 107-117, Cambridge University Press (1996).
- [11] Truax, B.; Real-time Granular Synthesis with Digital Signal Processing Computer, *Computer Music Journal* 12(2), pp. 14-16 (1988).
- [12] Smalley, D.; Spectromorphology: explaining sound-shapes, *Organized Sound* 2(2), pp. 107-126, Cambridge University Press (1997).
- [13] Malloch, J.; Sinclair, S.; Wanderley, M.M.; A Network-based Framework for Collaborative Development and Performance of Digital Musical Instruments. Proc. Of Computer Music Modeling and Retrieval 2007 Conference, Berlin (2008).
- [14] Malloch, J.; *A consort of Gestural Musical Controllers: Design, Construction and Performance*, M.A. Thesis, McGill University (2007).
- [15] Wanderley, M.M.; Depalle, P.; Gestural Control of Sound Synthesis. Proceedings of the IEEE, Vol.92, No.4 (2004).