

Universidade Federal de Minas Gerais

Escola de Música

Luiz Gustavo Vianna Batista

Associação de Imagens a um Sistema Musical Interativo

Minas Gerais – Brasil

Outubro de 2011

Luiz Gustavo Vianna Batista

Associação de Imagens a um Sistema Musical Interativo

Dissertação apresentada ao Curso de Mestrado da Escola de Música da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Mestre em Música.

Linha de pesquisa: Sonologia

Orientador: Prof. Dr. Sérgio Freire Garcia

Belo Horizonte
Escola de Música da UFMG
2011

Resumo

Desenvolver um sistema musical interativo associado a imagens, objetivo desta pesquisa, ofereceu uma oportunidade única de manipular mecanismos, ideias e conceitos relacionados ao uso e processamento "ao vivo" de áudio e vídeo, em uma experiência que resultou na peça-protótipo "*Teste n°1*". Buscar um sistema pequeno e eficiente, para ser utilizado principalmente por um músico e seu computador pessoal, envolveu uma série de posturas e questionamentos, descritos sucintamente na introdução, ao nos depararmos com a delicada associação entre linguagens, com a conceituação de "sistemas musicais interativos" e com decisões de escolhas de softwares.

Dois capítulos tratam dos principais estágios conceituais envolvidos: a "captação" de dados e seu posterior "processamento". Articulando-os, (separando e unindo, ao mesmo tempo) temos um *Intermezzo* abordando o pragmatismo das dimensões do tempo e da escrita, heranças da música ocidental incontornáveis em nosso trabalho.

Abstract

The development of an interactive music system accompanied by images, purpose of this work, offered a unique opportunity to the manipulation of algorithms, ideas and concepts related to the use of audio and video processing in live performances, which resulted in an experiment called "*Teste n°1*". The achievement of a compact and efficient system, to be used mainly by a sole musician and his/her personal computer, was accompanied by a series of personal decisions, succinctly dealt with in the Introduction, facing the delicate issue of the association of languages, the conceptualization of interactive music systems, and the choice of the softwares to be used.

Two chapters deal with the two main processes in this experiment: "data sensing/acquisition", and its eventual "processing". An *Intermezzo*, simultaneously splitting and integrating both chapters, comes into play, discussing the pragmatism inherent to the notions of time and music-writing, traces of the western music heritage unavoidable in this work.

Sumário

Introdução:	Pág.
I) Associações e Linguagens.....	6
II) Projetando Imagens.....	8
III) Sistemas Musicais Interativos.....	13
IV) Performance Musical com Computador: Max e Live.....	18
1º capítulo: Captação e Análise (Sensing) – O 1º estágio de sistemas interativos.	
1.1 Interfaces e Transdutores.....	26
1.2 Alguns Descritores de Áudio.....	29
1.3 Ferramentas Midi, <i>Wii mote</i> e Software Osculator.....	34
1.4 Captação de Imagens e Produção de Matrizes.....	38
[Intermezzo] - Entre dois estágios: a dimensão do tempo e da escrita.	
I – Pragmatismo em Sistemas Musicais Contemporâneos.....	43
II – Nenhuma Novidade no <i>Front</i>	46
III – Um Roteiro Musical.....	48
2º capítulo: Processamento – O 2º e último estágio de um sistema.	
Um estudo de caso: "Teste n°1"	
2.1 Construindo um Sistema Visual.....	57
2.2 Contando as Seções e Acionando <i>Presets</i>	65
2.3 Um Divertimento para Teclados.....	76
2.4 Manipulando Processamentos.....	81
Conclusão: Uma sinergia possível.	
Um trabalho em Aberto.....	97
Bibliografia.....	103

Introdução

I - Associações e Linguagens.

No início desta pesquisa, que pretendia ilustrar, de certa forma, um sistema musical interativo, tínhamos apenas algumas imagens (pinturas realizadas durante meu curso de Belas Artes, alguns anos antes) e a certeza de que podíamos manipulá-las em ambiente propício, em um único computador pessoal, junto ao software Max/Msp/Jitter.

A ideia de “*Projetar Imagens em Sistemas Musicais Interativos*” nosso tema central, discutido por partes ainda nesta introdução, indicava ser necessário criar as relações dentro de um “sistema próprio”, junto a um pequeno instrumental, para poder então refletir sobre ele. Na ampla vastidão das experiências contemporâneas nestes campos (apresentações com sons e imagens), acreditamos ter conseguido oferecer um significativo exemplo das possibilidades de atuação “ao vivo,” obtido com modestos equipamentos.

A partir de meados desta pesquisa, depois de terminados os primeiros estudos e experiências (com tantas possibilidades e “ferramentas” que nos eram oferecidas), poderia abordar a curiosa autonomia e integração, geradas sob a ótica das três principais linguagens aqui envolvidas, tentando descrevê-las:

- **Imagens**, no contexto aqui apresentado de projeção em tela, adquirem um significado simbólico muito próximo à linguagem cinematográfica, e conseqüente influência de sua herança e tradição de quase um século. Variações (movimentação) de fundo sobre um morro estático (figura) em primeiro plano, um facho mixado posteriormente, zoom, enquadramentos e “movimentos de câmera”, além de uma certa narrativa, no desenvolvimento destas imagens, são alguns recursos nitidamente buscados desta aproximação. Outra influência importante (presente até nas pinturas originais) é a de uma certa “estética dos vídeo-games”. Na intenção de também fazer parte do discurso, esta mais nova integrante (a interatividade dos games) deve ter uma participação ao mesmo tempo discreta e significativa, a fim de não alterar gratuitamente ou exageradamente a progressão, também narrativa e pré-concebida, das imagens. Neste roteiro visual, onde não há, no entanto, filme algum, as “pinturas” continuam estranhamente íntegras, e nossa

projeção refletirá também apenas uma espécie de “cenário digital”, junto a um músico e o ambiente ao redor, com seu percurso diretamente vinculado às seções de uma apresentação musical.

- **A Música**, decorrente deste sistema instrumental formado por um músico e seu computador pessoal, (ilustrada abstratamente por imagens que certamente também se associariam a vários outros tipos de sonoridades e estilos) utiliza basicamente mecanismos de *loopings*, já bastante tradicionais desde os primórdios da música eletroacústica. A performance do software Live, incorporado ao nosso sistema e responsável pela gravação e reprodução das seções de nosso discurso sonoro, influencia estilisticamente a peça, porém não mais do que um determinado grupo instrumental influencia uma composição. Esta orquestra de um homem só, realizada por *loopings* e processamento de sinais de áudio e Midi, oferece, contudo, novas possibilidades de explorações harmônicas e contrapontísticas, ao permitir todo tipo de conjugação de sons, cada qual inerte em seu próprio eixo temporal, com liberdade de gerenciamento semelhante ao experimentado por *DJs* (*Disc-jóqueis*). Adicione-se ainda, aqui, a adrenalina, e as amplas variações de interpretação e liberdade de ação presentes nas performances “ao vivo,” junto a instrumentos tradicionais.

- **A Programação** de sistemas em softwares como o Max (ou Pd) permite acesso a mecanismos antes restritos aos profissionais das ciências da computação ou engenharia de áudio. Este ambiente gráfico “aberto” possibilita a criação de conexões, cálculos e sequências lógicas e formais de maneira direta e simplificada. Prazer idêntico aos processos de criação artística (sonora e visual) são experimentados ao construir algoritmos, ou utilizar equipamentos que respondam às demandas por um desempenho específico, ou por mais eficiência e simplicidade. Encontrar, entre tantas opções, não só a melhor, como também um sentido, a esta intervenção. Buscar a “inteligência” nos processos, e não no comportamento de máquinas e sistemas, sempre escravizados pela inevitável automação de seus mecanismos.

II - Projetando Imagens

Embora a música seja uma linguagem artística autônoma e independente, as apresentações ou performances musicais sempre se valeram de recursos visuais, seja através da figura de um intérprete e suas gestualidades, ou de cenários, iluminação, ou ambiente específico. A música também sempre se associou a outras linguagens artísticas, como a dança, o teatro ou, mais recentemente, ao cinema. Eventos musicais populares contam, atualmente, com a presença de diversos tipos de iluminação, projeções e cenários digitais. De fato, esta disponibilidade de recursos (especialmente a da última década) atinge toda a sociedade, com uma ampla variedade de equipamentos de áudio e vídeo se tornando cada vez mais baratos, na rápida substituição por modelos mais recentes (e, ao menos quanto à sua capacidade de processamento, mais eficientes). A quantidade de ferramentas de atuação multimídia não suprem, de toda forma, carências ou deficiências específicas de linguagem, ou excessos de superficialidade na relação entre os diversos elementos envolvidos. Estas possibilidades de atuação permitem, no entanto, a um indivíduo e seu computador pessoal, atuarem de maneira integrada e autônoma, e programar processos sonoros e visuais simultâneos. O presente trabalho, mais do que apenas acrescentar imagens às performances musicais, pretende desenvolver e manipulá-las sob o comando de um músico e seu discurso sonoro, encarando um dos principais desafios contemporâneos no relacionamento com a tecnologia digital: o de melhorar e aumentar a interatividade.

Neste início de séc. XXI a expansão da capacidade de processamento e de memória de hardwares continua. Junto ao desenvolvimento das máquinas vemos o aprimoramento dos softwares, que passam a incorporar novas funções, como é o caso do principal programa utilizado neste projeto (Max), que em 2008 adotou definitivamente o processamento de matrizes de imagens, algo talvez inimaginável quando foi inicialmente desenvolvido, e executava funções bem mais modestas.

É mesmo uma época muito feliz, para todos aqueles que se empenham com arte eletrônica, e que iniciaram seus trabalhos nos caros, lentos e pesados

equipamentos da era analógica. Complexos processamentos de áudio e vídeo se realizam em simples e compactos *Laptops*, e os aparelhos parecem caminhar em um constante aumento de desempenho e diminuição de tamanho. Imagens e música (e tudo mais que possa ser digitalizado) podem ser armazenados e baixados através das longas linhas de comunicação e centros de memória (*cloud computing*) criadas pela internet. Ela também é hoje uma ferramenta essencial, aos computadores e às pessoas, com seu imenso acervo de informações, serviços e usuários. Alguns dos objetos (uma das ferramentas básicas com as quais se constroem programações no Max) operados nesta pesquisa vieram não diretamente do software escolhido e de seu fornecedor, mas de artistas e pesquisadores que os dispuseram livremente via comunidades ou sites pessoais na web¹.

Nesta época da reprodutibilidade máxima de imagens e música (compartilhamento de dados pela web), e do amplo acesso à informação (wikipédia, redes sociais, etc.), o ambiente ideal de criação continuará a desafiar o candidato com a máxima liberdade possível de ação, e que é, no plano bidimensional, representado por uma página em branco. Uma programação para Max é preenchida sobre uma tela de computador vazia. Um compositor também é desafiado a preenchê-la, assim como um pintor ou desenhista.

As transformações que atingiram radicalmente a produção, reprodução e fruição das artes, a partir do séc. XIX, também alteraram as poéticas da pintura e da música. A criação das máquinas fotográficas, por exemplo, retirou dos pintores e desenhistas o privilégio da produção de imagens, ao mesmo tempo que tornou possível, e de certa forma os forçou, a buscar novos e abstratos simbolismos e expressões (abstracionismo, cubismo, etc.), libertando-os da mera reprodução mais ou menos exata da natureza.

As primeiras tentativas bem sucedidas para capturar imagens foram feitas na França, durante a década de 1830. O famoso Daguerreótipo das ruas de Paris (e de seu ilustre morador que engraxava os sapatos), feito por Louis J.M.Daguerre é de 1839. Aparentemente o desafio era mais voltado à descoberta de soluções químicas, uma vez que se tratava de registrar as imagens projetadas em uma

1 É o caso do objeto max “smooth” e do próprio “Fiddle”, explicado no 1º capítulo.

rudimentar, e há séculos conhecida, “caixa escura” (com um pequeno orifício, por onde a imagem se projetava na superfície oposta). Os primeiros experimentos foram feitos com chapas metálicas, revestidas de várias soluções de prata. Foram substituídas por chapas de vidro, que dominaram a 2ª metade do séc. XIX. Em 1890 a Kodak incorporou o modelo com rolo de filme, e em 1907 aparece o *autocrome*, capaz de registrar diferentes cores.²

Os objetivos seguintes, de ordem física e mecânica, se voltaram então para a captação e reprodução do movimento da imagem . Em 28 de dezembro de 1895, no Salão *Grand Café*, em Paris, os irmãos Lumière fizeram a apresentação pública, junto a seu invento, o cinematógrafo, do pequenino filme *L'Arrivée d'un Train à La Ciotat*.

O séc. XX viu surgir o fascínio do cinema e toda a sua influência sobre a invenção da vida moderna³. Emergência do cinema, da fotografia, do cartaz. Da publicidade, do desejo do consumidor, do marketing. Esta avalanche de imagens ajudou a consolidar a identidade visual e cultural da sociedade de consumo, principalmente a ocidental. Neste início de século XXI esta influência das “imagens-propagandas” sobre a vida pós-moderna continua ativa, e amplamente pulverizada através de celulares, *Ipods*, *P.C.s*, monitores (na cozinha, no carro, etc.). “Telas” ocupam um lugar significativo, na casa e na vida da maioria das pessoas, principalmente a TV⁴. Este estudo não abarca tais aspectos desta recente influência. Talvez seja mesmo seu reflexo direto e inevitável. Ao propor projetar imagens processadas juntamente a um sistema musical, talvez esteja partindo desta mesma necessidade que parece invadir até mesmo os palcos dos concertos eletroacústicos e de música eletrônica, onde o público, desprovido de estímulos visuais ou gestuais, parece ficar pouco à vontade.

São até demasiado comuns os aplicativos conhecidos como “*players*”, ou tocadores de mídias de áudio e vídeo, que acompanham esta insatisfação visual fornecendo aos arquivos apenas de áudio, como os MP3, por exemplo, formas

2 Data de comercialização do autocromo (Fonte: Wikipédia)

3 Parafrazeando o título da coletânea de ensaios “O cinema e a invenção da vida moderna”.

4 Waldemar W. Setzer (www.ime.usp.br/~vwsetzer) é aqui a principal referência, com seu posicionamento crítico às Tvs comerciais e à mistificação de computadores. Sua influência se encontra junto à minha postura “pragmática”, assumida abertamente ao longo deste trabalho. Ressalto apenas minha gratidão por seus textos agudos e entrevistas inesquecíveis (no Roda Viva, por exemplo).

projetadas por diferentes tipos de desenho gráfico, em uma contínua e hipnotizante sequência de imagens. Contudo, além dos anseios do público e consumidor, esta invasão das salas de concertos com imagens projetadas em telas também tem suas explicações práticas. A primeira é a disposição de auditórios e salas de teatro como local para concertos (é natural que as pessoas estejam sentadas de frente para um palco, e não se privem da visão). A segunda, e mais convincente, é de que também há um anseio legítimo por renovação e uso de diferentes linguagens. Muito mais do que telas com vídeos ou estética promocional, é possível projetar cenários com desenhos ou qualquer tipo de imagens. E representar visualmente a atividade de algum processamento sonoro ou seção musical, por exemplo, alterando esta imagem estática e fazendo com que, de alguma forma, ela adquira movimento, ou a represente sob a forma de alterações de brilho, saturação, contraste, inversão de matiz, etc.

Diferente do cinema, onde a música é apenas um dos componentes sonoros (junto à fala, e aos sons ambientes), e talvez mais próximo aos modernos “*vídeo-clips*”, onde as imagens apenas acompanham o discurso e ritmos sonoros, nossa associação a pinturas, tais como as escolhidas, nos remetem mais diretamente ao cinema de animação (desenho animado). Esta natureza sintética e estática da pintura, ou do desenho, os tornam ideais a receber um processamento que os transforme ou crie movimentos a partir de conexões com parâmetros sonoros. Conexões arbitrárias, posto que não existem relações artísticas ou estéticas exatas entre imagens e sons, tons/timbres e cores, ou entre efeitos visuais e efeitos sonoros. Relações psicofísicas ou psicoterapêuticas não nos interessam, por enquanto, mas sim estratégias de utilização (ao vivo) de processamento de efeitos visuais em imagens, dentro de uma produção musical (sonora).

Estas estratégias, contudo, vão além de conectar estruturas sonoras a efeitos visuais. Uma relação exagerada de dependência criaria um ambiente semelhante ao proporcionado em jogos eletrônicos, com suas relações diretas de causa e efeito, metas e objetivos. Neste sentido, um discurso musical melhor se associaria à tradição herdada do cinema, onde desde Eisenstein e Pudovkin se reconhece a necessidade de um contraponto (com sua independência de ritmos próprios) entre imagem e música (som). Além disto, é claro, uma cuidadosa

“montagem” deste contraponto.

O resultado visual desta associação, portanto, deverá se servir da vasta poética incorporada à tradição cinematográfica (principalmente a do cinema de animação), acrescidas de alguma relação direta de causa e efeito presente na interatividade dos *games*. Como origem, o curioso e ancestral meio da pintura (têmpera e óleo sobre papel e tela de algodão, fixados em chassis de madeira).



Dois quadros, de autoria própria, com técnicas e dimensões diferentes (o primeiro, têmpera sobre cartão 30x30cm, e o segundo, óleo sobre tela 130x130cm), fotografados digitalmente e utilizados como ponto de partida deste projeto.

III- Sistemas Musicais Interativos

Largamente empregado em exposições de 'artes digitais' ou 'novas mídias', o termo "interatividade" sugere a participação, seja da plateia ou de um intérprete específico, de maneira a interferir nos resultados de um sistema. Concebido principalmente em ambientes digitais (embora seja possível conceber arte interativa analógica ou mecânica, dada sua abrangência) o termo, quando se refere a "Sistemas Musicais," tornou-se mais restritivo e específico a partir da década de 1990.

R.Rowe, em seu pioneiro livro "*Interactive Music Systems: Machine Listening and Composing*", de 1993, o definiu como uma cadeia de processos em três estágios: - captação, processamento e resposta.

Sensing - processing - response

Logo na primeira página de seu livro, porém, publicou seu conceito tão criticado posteriormente, por incluir imprecisamente o termo '*musical input*':

"Interactive computer music systems are those whose behaviour changes in response to musical input. Such responsiveness allows these systems to participate in live performances, of both notated and improvised music"

Rowe também estruturou uma classificação primária, baseada na distinção entre os principais mecanismos e estruturas de funcionamento utilizados pelos principais Sistemas então existentes:

1- Gerenciamento de Tempo (*timing*): *Score-driven* ou *Performance-driven programs*.

2- Método de processamento: *Transformative, Generative* ou *Sequenced*.

3- Comportamento de resposta: *Instrument paradigm* ou *Systems following a player*.

J.Drummond, em seu artigo “*Understanding Interactive Systems*,” (2009 revista *Organised Sound* **14** (2): 129-133) é quem melhor aponta as recentes evoluções no conceito, a partir das definições iniciais de Rowe. Questionando o conceito impreciso de “*musical input*”, cita Paine (2002): “*The Rowe definition is founded on pre-existing musical practice (...) it does not derive from the inherent qualities of the nature of engagement such an 'interactive system' may offer*”. E Jordà (2005): “*How should an input be, in order to be musical enough?*”. Sua abordagem se estende à noção de controle compartilhado, composição interativa entre performer, plateia e instrumento, ou mesmo entre máquina e máquina, misturando as regras de atuação tradicional entre composição, interpretação e instrumento. Descreve também as três metáforas de Chadabe (2005), onde a interatividade (em tempo real) de Sistemas Musicais podia se comparar a:

1- *Sailing a boat on a windy day and through stormy seas.*

2- *The net complexity or the conversational model.*

3- *The powerful gesture expander.*

Drummond cita também Bongers (2000: 128) para quem a interação com um sistema envolve *feedback*, que ajuda o usuário a articular o controle. E Miranda e Wanderley (2006 : 3) que ressaltam um relevante aspecto estrutural, dentro do estágio descrito como processamento (*processing*): as conexões e mapeamento (*mapping*) dos gestos e sensores (*inputs*) aos softwares, algoritmos e demais processos⁵.

Os conceitos de mapeamento e *feedback* são importantes, pois ajudam a desmembrar a sobrecarregada etapa descrita como “processamento”. Computadores são imensas redes de mapeamentos e conexões, e sistemas

⁵ Agradeço ao Professor Marcelo Wanderley por seu curso “Seminário Multidisciplinar: interação humano-computador em situação musical”, em junho de 2011, na UFMG.

gráficos as utilizam de maneira semelhante aos da era analógica, conectando e desconectando cabos (só que utilizando linhas na tela de um monitor) numa sequência clara de operações com dados e sinais, que indicam e percorrem um caminho.

Destaque desnecessário, ou ao menos inerte, envolve a etapa descrita como resposta (*'response'*). O termo se refere, naturalmente, à produção sonora decorrente das operações anteriores, sendo, portanto, geralmente uma simples conversão digital-analógica. Cumpre-nos ressaltar que esta resposta é sempre uma “reação” deste sistema. Computadores e seus aplicativos reagem sempre da maneira imposta por seus algoritmos, em uma programação que é rígida e formalizada em processos lógicos e matemáticos. Respostas, aqui, são sempre reações provenientes de operações simples ou extremamente complexas, mas sempre reações.

Neste sentido, as metáforas de Chadabe são sugestivas, porém desde que não as tornemos exageradamente ingênuas. Sistemas musicais interativos se comparam a uma aventura de “veleiro” em mares agitados, onde estamos sempre corrigindo nossa orientação em função da direção pouco predizível do vento. Ou à uma conversa, onde dizemos ou respondemos coisas, numa influência mais ou menos recíproca. Metáforas são, contudo, e por definição, genéricas, e se aplicariam também a qualquer relação entre a concepção imaginada e sua concretização “física”. Nesta relação com materiais artísticos, ou mesmo com equipamentos e sistemas elétricos, por exemplo, percebemos a dificuldade de materialização de conceitos abstratos artísticos ou matemáticos por meio da realidade física e tecnológica dos materiais constituintes. Problemas com erros, instabilidade ou variações dentro de um sistema de computadores não se comparam a experiências “reais” e seus problemas, com a natureza “viva”. Abrir uma porta (real, não como metáfora), desconhecida, pode ser efetivamente uma “aventura” incrível. E exigir o mais alto grau de interação.

Simular modelos de conversa baseando-se na troca de informações é algo relativamente simples. O problema são os critérios e filtros de seleção destas informações, ou seja, seus algoritmos. Computadores simplesmente cumprem as funções para as quais são programados, em sua linguagem binária que opera

procedimentos rigorosamente lógicos e formais. Há uma beleza intrínseca e inerente nestes processos, com seu rigor matemático, sua estrutura e arquitetura internas, e às vezes na própria simplicidade dos resultados conseguidos. Beleza e méritos que são substituídos pelo discurso de uma suposta “inteligência artificial das máquinas e sistemas do futuro”. Por mais complexos que seja o conjunto hardware-software, operem eles por complexos sistemas que se realimentem infinitamente de dados ou recebam e reajam a uma vasta rede de interferências, haverá sempre a mesma subordinação a regras pré-concebidas. Para além da superficialidade da comparação homem/máquina, natural/artificial, desconhecido/calculado, ou ainda auto-determinação/programação, o fato é que a operacionalidade com cálculos e o desempenho com seu processamento incorporaram os circuitos integrados a boa parte de nossa atividade rotineira. Suas funções, ou dos modernos *micro-processadores*, seguem uma mesma característica que nortearam a criação das primeiras máquinas: - a automação.

Neste longo caminho, em busca de precisão e controle, temos também a lenta evolução das características dos instrumentos musicais. Em outro sentido, uma partitura também sugere certa “automatização,” ou dela precisa, ao pretender desenvolver um discurso conjunto e articulado sob o domínio do tempo. Não é possível pará-lo para experiências ou questionamentos, durante uma performance musical. Além disto, em uma apresentação em auditório, temos que seguir muitas vezes uma certa ordem prática, e há limitações de meios e recursos que não existem em uma música processada em estúdio. Enfim, as coisas tem que se apresentar, de certo modo, “prontas”.

Na presente pesquisa, na qual desenvolvem-se simultaneamente dois discursos, um sonoro e outro visual, há um enorme desafio em dispor e preparar previamente estes processos, quer semi-automáticos, ou de interferência direta, de uma maneira convincente e efetiva. A forma de se lidar com as seções e sub-seções de uma peça, preservada a liberdade e autonomia de execução do intérprete, sempre envolverá a decisão de qual mecanismo usar na passagem e controle do tempo. Acioná-lo manualmente, (o equivalente a apertar um botão) e disparar coordenadas - *presets* - é uma ação bem simples, quando desempenhada em ambiente digital, onde não espelha a presença física de cabos e conexões.

Desempenho com eficiência garantida, perdendo, contudo, qualidade na relação com a interatividade.

Neste sentido, sensores, estes mecanismos de interferência ou comunicação com o sistema, não se limitam a gatilhos (*triggers* ou *toggles*), disparados com um dedo ou por um pedal, informando eletricamente 1 ou 0 . Há um enorme interesse industrial na criação e desenvolvimento de interfaces e dispositivos capazes de reconhecer os mais diversos gestos e formas de expressão do ser humano. Sistemas de reconhecimento e síntese de voz, de rostos ou sorrisos, movimentos corporais captados em 3 dimensões, são alguns exemplos. As empresas de jogos eletrônicos são grandes investidoras e interessadas nos mecanismos de sensoriamento gestual, e no aprimoramento desta interface entre as pessoas e seus aparelhos. Novos controles tem sido popularmente comercializados, um dos quais veio se integrar a nosso sistema: um pequeno “*joystick*” (pág.83), com transmissão *bluetooth*, equipado com acelerômetros. Apresentam ainda inconveniências à nossa adaptação musical, mas apontam para a imensa aplicação que os novos sensores alcançarão, seja junto a tradicionais instrumentos musicais, seja como um próprio (e novo) instrumento. Precisarão, contudo, primeiro se justificar perante o discurso musical, e junto aos demais gestos e ações desempenhados no palco ou em performance.

IV– Performance Musical com Computador: Max e Live.

Terminada a 1ª década do séc. XXI, um rápido olhar sobre o panorama de eventos musicais ao redor do mundo mostraria, evidentemente, a mesma complexidade cultural que distingue cada uma das diferentes civilizações que aqui habitam. Novos meios de comunicação criaram o termo “aldeia global”, ao menos quanto à rede de transmissão de dados que interligam continentes por satélites e cabos de fibras óticas, e conectam pessoas, organizações, empresas e governos. Não é pequena, de qualquer forma, a parcela da humanidade que permanece com seus seculares hábitos, costumes e tradições isentos de influências diretas desta recente tecnologia digital. Sociedades reclusas, ou excluídas pela pobreza, intolerância religiosa e autoritarismo político. No entanto, a música coabita praticamente todos os povos, com seu sentido simbólico, afetivo, imersa nos ritos e tradições destas culturas e pessoas. Naturalmente, ao tocarmos um instrumento acústico não dependemos de computadores. O que eles permitem é aumentar significativamente a complexidade dos fatores envolvidos em todos os processos de produção musical, atuando em uma simples nota, ou em toda uma peça, a partir de um instrumento, ou qualquer outra fonte sonora.

Na imensa colcha de retalhos cultural e tecnológica em que vivemos, a comparação com uma metrópole (uma com origem medieval e europeia, como a música ocidental) nos mostraria prédios moderníssimos erguidos ao lado de construções antigas, de muitas épocas distintas. Relíquias, monumentos, e ruínas. E como as cidades, os homens também se apresentam cheios de anacronismos e contradições.

Ao mesmo tempo que preenchem nossa ânsia por renovação de linguagem e novas realizações, as novidades tecnológicas proporcionam recursos e carregam possibilidades, inclusive a de ser usada (deliberadamente ou não) para colidir com aspectos culturais e sensíveis, envolvidos na experiência musical.

lazzetta (1998), ao se referir às novas relações de produção musical, comenta:

“(…)A manipulação de gravadores, instrumentos eletrônicos e computadores rompe a materialidade sonora. Sons eletrônicos não estão conectados a nenhum

tipo de relação gestual ou corpórea com os aparelhos que os produzem. Ao mesmo tempo que isso representa uma extrema abertura em termos de liberdade de criação sonora, há também uma perda substancial de uma série de conexões entre o evento sonoro e o universo em que ele ocorre. E a perda destas conexões implica na perda de dimensões simbólicas e de significação. Ou seja, ao mesmo tempo que as novas tecnologias sonoras abrem caminho para novos e complexos universos estéticos, existe o risco de um empobrecimento semiótico desse universo. Ao se desviar de uma prática corporal para processos ligados à pesquisa intelectual, a composição musical se vê obrigada a suprir a perda de toda carga simbólica dos gestos instrumentais por meio de recursos formais.”

Estas parecem ser as dificuldades inerentes às experiências mais radicais envolvendo a criação de novos instrumentos, desvinculados dos gestos e conexões sonoras tradicionais. A maior parte das produções musicais contemporâneas, no entanto, ainda partem do alto nível de dedicação e comprometimento que músicos de orquestra dedicam a seus instrumentos, do alto desempenho sonoro que eles alcançam, e do simbolismo comum de suas performances.

O duro golpe na forte relação do instrumento e sua fonte sonora, e na secular tríade “compositor-intérprete-ouvinte” ocorreu já no final do séc.XIX, historicamente delimitado com a invenção e comercialização dos fonógrafos e gramofones (a indústria percebeu bem a sua potencialidade, com a criação da *Gramophone Company*, de Londres, em 1897). A multiplicação do acesso às informações musicais inauguradas pelos aparelhos e seus discos levaram também à necessidade de questionamentos e revisão de conceitos sobre os sons e formas musicais.

Surge aí a possibilidade nova de se analisar objetivamente as estruturas internas do som, em seus aspectos físicos e acústicos. O passo seguinte ocorreu naturalmente, na abrangência tecnológica do uso da eletricidade. Já não mais se distingue bem a figura do compositor e do técnico de som, e o intérprete é ainda mais anulado. Em meados do século XX nasce a música concreta, através dos estudos de Pierre Schaeffer para a Radiodiffusion-Television Française (RTF), onde se estabeleceu um estúdio de música. Surge também outro estúdio similar,

na cidade de Colônia, Alemanha, onde Herbert Eimert desenvolveria seu conceito de *elektronische Musik*.

S.Freire (2004) descreve o quanto nossa experiência auditiva e musical deve à prática de gravações e à presença de microfones e alto-falantes, estes buracos por onde passam sons reais ou imaginários:

“o alto-falante funciona muitas vezes como uma janela para micromundos acústicos explorados por microfones, através da qual podem ser escutados detalhes de ressonâncias, vibrações, atritos não acessíveis à audição normal. De um ponto de vista mais metafórico, ele também empresta sua voz a outros fenômenos – invisíveis – de natureza não mecânica, sejam circuitos eletroeletrônicos geradores de sons, sejam vibrações elétricas do corpo humano, sejam resultados de algoritmos que se desenvolvem dentro de microprocessadores.”

A partir da década de 1960, o mundo conheceria a chamada revolução digital, que converte e re-converte informações em uma lista de dados numéricos, podendo representar qualquer tipo de signo, sons, imagens, processos, etc. Computadores permitem tratar, transformar, armazenar e reproduzir estes dados com precisão, sempre de acordo com coordenadas claras de manipulação (em linguagem específica de programação).

Já na década de 1980 não apenas os computadores se tornaram acessíveis e populares como isto se estendeu também à produção e comercialização de novos aparelhos e instrumentos musicais, especialmente teclados/sintetizadores.

É dessa época um dos protocolos de comunicação ainda hoje em utilização (uma raridade para os padrões digitais), seja entre sintetizadores ou módulos de som, seja entre diferentes softwares: o Midi⁶. Esta rústica interface, voltada sobretudo à representação dos mecanismos simples de acionamento de notas e disposição de um teclado musical, representou uma base comum de comunicação entre vários modelos e marcas diferentes, o que até hoje traz benefícios, num mundo onde a pluralidade de empresas faz com que disputem mercado de maneira individualizada e com interesses próprios de seu modelo ou padrão.

6 Musical Instrument Digital Interface.

Foi também na década de 1980 que M. Puckette considerou o nascimento do software Max, moldado por músicos, pesquisadores e *performances* no IRCAM, durante o “*excited*” período de 1985 – 1990. Em seu artigo “*Max at Seventeen*”, de 2002 (Computer Music Journal 26/4, p31-43), Puckette descreve minuciosamente o desenvolvimento do “paradigma” Max, formado pelo Max/Msp, e o Pd (Pure Data, versão em software aberto para sistemas operacionais Linux). Esclarece a história e os personagens responsáveis por sua formação até a versão moderna, como Barry Vercoe e, claro, Max Mathews (de onde a homenagem do nome “Max”) e seu programa RTSKED.

Puckette aponta ainda que o Max é mais orientado aos processos do que às informações, que sua expressividade provém da facilidade de interconexão e intercomunicação, e simplificada disposição gráfica:

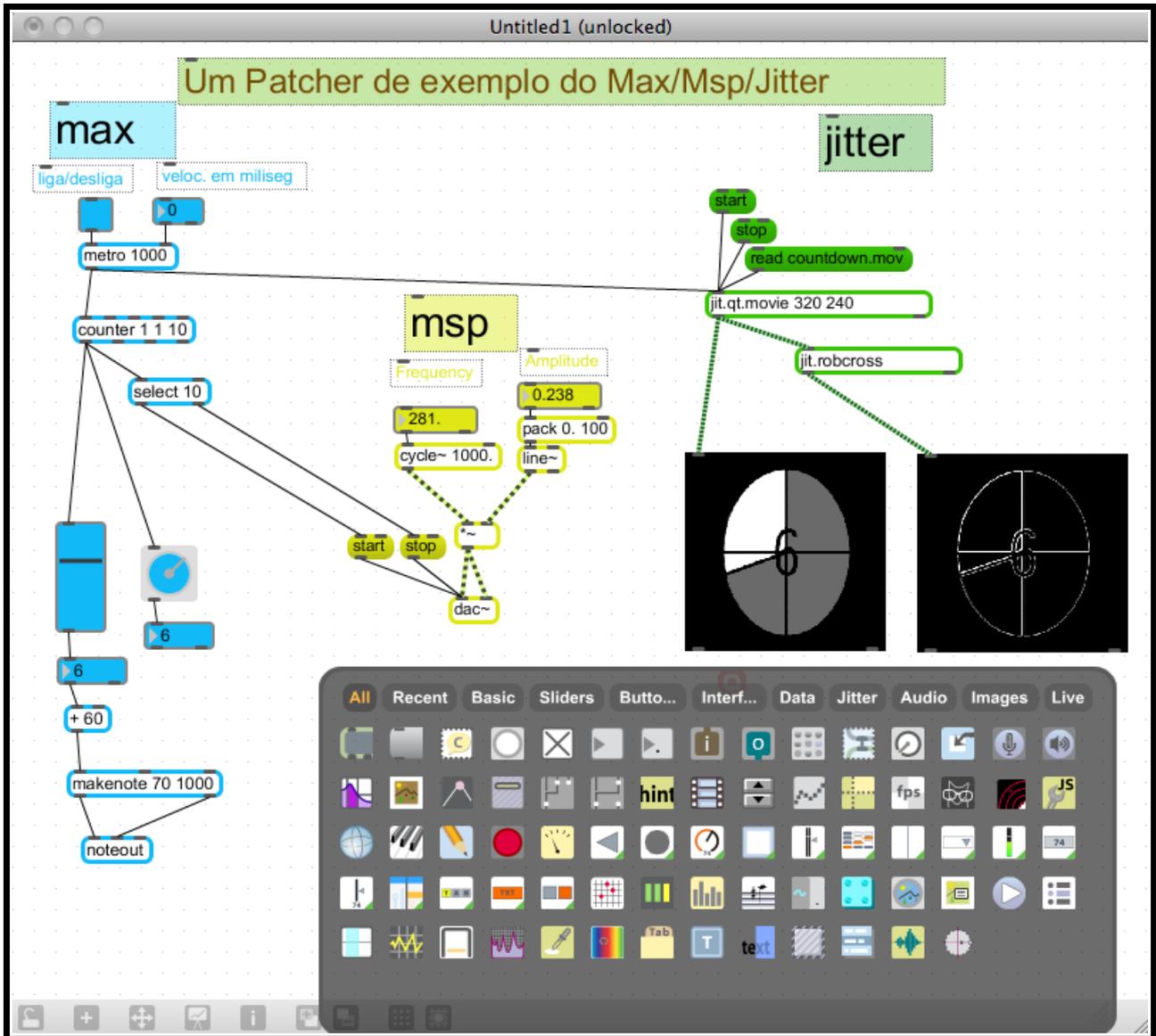
“There is no concept of scoping or namespaces in Max; all symbols and their bindings live in one flat space. This decision was made to remove a layer of complexity that didn't seem to be strictly necessary in the context of computer music production, in order to make Max as accessible as possible to people who aren't professional computer programmers”.

Sobre a neutralidade estilística que a “página em branco” inicial do Max propicia, ele ressalta o caráter singular da escrita, essencial junto à música ocidental europeia:

“On starting Max, the user sees nothing but a blank page -no staves, time or key signatures, not even a notion of “note”, and certainly none of instrumental “voice” or “sequence”. Even this blank page carries stylistic and cultural freight in at least one interesting respect: the whole idea of incorporating paper in the music-making endeavor was an innovation of Western Art Music, for which many other musics have no use at all. Musical practices which don't rely on paper may in some cases have much less use for computers than have the more Westernized ones”.

Aprender e se familiarizar um pouco com a linguagem simplificada de programação do Max irá requerer a leitura e estudo dos exemplos de seus manuais, especialmente aqueles designados como “Tutoriais”. A seguir, uma breve ilustração de um *patch* (uma página de programação), envolvendo alguns procedimentos simples manipulados no Max/Msp/Jitter, ou simplesmente Max, e

associados às três categorias de dados processados. (O Msp cuida especificamente dos sinais de áudio, e seus objetos possuem o símbolo “~” junto ao final do nome. O Jitter processa, por sua vez, os sinais de vídeo, e seus objetos iniciam-se com a abreviação “jit.”.)



Conexões e objetos tipicamente Max (à esquerda), Msp (no centro) e jitter (à direita). Em baixo, emoldurados por uma tela cinza, estão os recursos disponíveis - os sete primeiros se referem a objetos, mensagens, comentários, *bangs*, *toggles*, e *displays* para numerais inteiros e fracionados (*integer* e *floating*)

Ao contextualizar novamente o Max, aos 26, diria que ele ainda é o mais simples e eficiente “ambiente” sobre o qual a interatividade pode ser amplamente trabalhada (com sons e imagens). Sozinho, ele seria capaz de desempenhar todas

as operações e funções deste projeto (inclusive receber e decodificar as informações *bluetooth* do *wiimote*, operação para a qual utilizamos um programa específico, o Osculator).

No momento em que estudava uma disposição (no Max) para o áudio e *loopings*, e ao comprar uma “placa de som externa”, necessária a um bom rendimento com o sinal original de áudio, adquiri, junto a este pequeno hardware, uma licença de uma versão reduzida (“*lite*”) de um determinado software: o Live, da Ableton.

Diferentemente do Max, a tela inicial e *design* do Live são pré-configurados. Porém exprimem as funções elementares das operações “ao vivo” com áudio, e são representadas pela interface gráfica de uma tradicional “mesa de som”, com canais independentes e controle de mixagens. Na modalidade “*session view*,” utilizada por este projeto (o Live possui ainda uma visualização no sistema “*arrangement view*”) ele permite a gravação e reprodução de “clips” com performance suficiente para ser operado em apresentações ao vivo. Além dos canais de áudio, cada um estocando seu próprio conjunto de clips, dispõe também de canais para recebimento de mensagens Midi (e que também armazenam *loopings*).

Processadores de sinais de áudio (compressores, filtros, *delays*, etc.) podem ser dispostos em cadeia, sobre cada canal, assim como efeitos (acordes, arpejos) e instrumentos (piano, bateria) nos canais Midi. Notas e controladores Midi (de 0 a 127) também podem ser designados para disparar a maior parte dos comandos dispostos nesta “mesa de áudio” (a interface gráfica do Live).

A eficiência em processar os *clips* em *loopings*, sejam pequenos ou longos trechos, e reproduzi-los (um por canal), faz do Live (um software criado para este fim e comercializado pela empresa Alemã Ableton desde 2001), como o próprio nome indica, uma referência para a performance de música “ao vivo”⁷.

7 Começam a aparecer estudos acadêmicos sobre o Live, como no artigo “Deleuze e o Ableton Live: A criação-performance (...) para músicos digitais”, de A.Michailowsky



8 canais para sinais de áudio e Midi, na principal tela do software Live (*session view*). Cada canal conta com 6 gavetas (*slots*) onde se armazenam as gravações (*clips*). Só um destes *slots* pode ser tocado por vez, dentro de cada canal. A maioria dos comandos apresentados nesta tela se comunicam por mensagens Midi com o programa principal, Max.

Os dois programas (Max e Live), funcionando então paralela e independentemente (e dividindo uma única tela de nosso computador), recebem os sinais de áudio via placa de som externa, e mensagens Midi do teclado controlador (conectados via USB).

A aposta na união do Max com o Live se caracterizou pelo lançamento de um *Plug-in* específico, comercializado e chamado de “Max for Live”. Não chegamos a utilizá-lo neste projeto, embora seja um desdobramento natural, e uma atualização necessária.

1º Capítulo

Captação e Análise (*Sensing*):

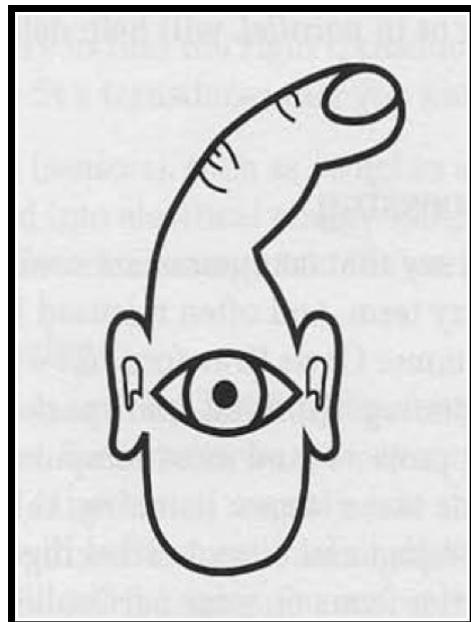
O 1º Estágio de Sistemas Interativos.

1.1 Interfaces e transdutores

O sucesso da *GUI* (*graphical user interface*), desenvolvido pela Apple e copiado pela Microsoft e Linux, simplificou a comunicação entre as pessoas e os computadores, tornando suas operações análogas às ações executadas em um escritório, por exemplo. Abrir e fechar janelas, arrastar e transferir arquivos em pastas, expô-los no *desktop* ou jogá-los na lixeira são ações rotineiras simples e amplamente reconhecidas por qualquer pessoa.

No livro *Physical Computing – Sensing and Controlling the Physical World with Computers* (2004), D.O'Sullivan e T.Igoe nos lembram como a maioria das pessoas descreve um computador : “*screen, Keyboard, and mouse*”. Da mesma maneira, escrevem os autores, somos reconhecidos pelos computadores, como alguém que aciona teclas, enxerga-lhe por um monitor bidimensional e é capaz de escutar sons estereofônicos. Ou seja, uma criatura com um olho, um dedo, e dois ouvidos.

A criatura que aperta teclas, enxerga um monitor bidimensional e escuta sons estereofônicos, na ilustração de D.O'Sullivan e T.Igoe.



Computadores pessoais ainda não dispõem de interfaces mais sofisticadas, capazes de se comunicar com todo o nosso corpo, e detectar outros gestos corporais. Mais do que o equivalente a uma Inteligência Artificial (*AI*, em inglês), que busca imitações aos comportamentos humanos, os autores propõem o termo Amplificação da Inteligência (*IA*), na busca por tornar interessante e possível

capturar e converter outras expressões pessoais em mensagens claras e que signifiquem benefício ao usuário, simplificação desta comunicação, ou ainda algum suporte. “E se uma pessoa quisesse comandar algo, se expressando com os pés, por exemplo?”

Interface é um amplo conceito, definido como os meios, sejam físicos ou lógicos, de comunicação (e utilização de dados) entre sistemas. Entretanto, os primeiros mecanismos de leitura de dados (de natureza física) são chamados “transdutores”, e convertem diferentes formas de energia em variáveis elétricas, transformadas posteriormente em representações digitais. Teclados de computadores traduzem energia mecânica em um comando elétrico simples – Liga/desliga. (I/O). Um sensor *CCD* ou *C-mos* reage à luz focada pela lente de uma câmera digital, e o caminho contrário é feito quando novamente decodificamos sinais elétricos para representações gráficas através da luz de um projetor ou monitor de vídeo. Microfones e alto-falantes representam transdutores equivalentes, no campo sonoro-musical.

Os seguintes sensores (captadores de energias físicas) foram utilizados neste trabalho:

Ação mecânica = contato simples (botões, pedal *sustain*) e sensores de pressão do teclado *Midi*.

Ondas sonoras = captador piezoelétrico de contato (embaixo do cavalete do violino).

Luz = sensor *CCD*, de filtro primário, de uma câmera fotográfica digital.

Movimentos (inclinação de planos) = acelerômetros do *wiimote*.

Sensores também se estendem a vários outros tipos de dispositivos desenvolvidos em pesquisas das mais diversas naturezas, como a bélica, a biomecânica, ou a indústria dos games. Suas informações, variações elétricas decodificadas por circuitos eletrônicos, convertidas nos parâmetros definidos por um protocolo de comunicação comum, e então compartilhadas (digitalmente), são utilizadas, ou como arquivos de dados, ou com funções de alterar e executar tarefas.

Ao serem captados “ao vivo”, estes dados precisam se encaixar nas condições em que os computadores (seus sistemas operacionais e programas) operam, seus mecanismos de tempo e manuseio de banco de dados (internos) conhecidos por “*handling and Scheduling*”, e que são responsáveis pelas funções estruturais, dentro destes programas e sistemas operacionais. O protocolo Midi, durante as décadas de 1980 e 1990 era a principal interface, junto a estes dispositivos. Rowe (1993 / pág.9) afirmou, na ocasião:

“Two basic pillars of interactive systems are MIDI handling and scheduling. (...) Music is a temporal art, and any computer program dealing with music must have sophisticated facilities for representing time and for scheduling processes to occur at particular points in time”.

Se a interface Midi ainda hoje representa esta “sofisticada facilidade,” cálculos e análise (descrição) dos sinais de áudio também hoje se apresentam “sofisticadamente fáceis”, para um bom computador pessoal. Programas como o Max, embora funcionem perfeitamente bem junto ao Midi, possuem seu próprio e elaborado sistema de “*Scheduling*,” imprescindível para um bom desempenho em tempo real. Ao tornarem-se simples “objetos” Max, os descritores de áudio também passam a ser incorporados aos Sistemas Interativos mais modestos (como este), e portanto a requerer maior atenção de músicos e programadores.

Em seu artigo “*Is there life after MIDI?*” (1994 ICMC), M.Puckette chamou a atenção para as duas partes envolvidas na discussão: a multidão de sintetizadores hedonistas disparados por um só homem, e a intransigência dos puristas de precisão matemática e caros equipamentos. A vida depois do Midi continua como antes do Midi, afirma Puckette, com todas as dificuldades formais e estéticas de se fazer música.

1.2 Alguns Descritores de áudio

Em 1998, graças também ao avanço na capacidade de processamento dos computadores pessoais, Miller S. Puckette, T. Apel e D. Zicarelli lançaram duas versões de descritores, ou dois objetos (unidades básicas de programação), para softwares Max/Msp ou PureData, capazes de pleitear, para o sistema de áudio, uma posição mais central e importante, inclusive junto aos controles e comandos interativos.

O primeiro destes objetos, utilizado neste trabalho, uma ferramenta de análise de áudio em tempo real chamado *Fiddle~*, é um eficiente (tanto quanto possível, pois não existem mecanismos perfeitos) 'analisador discreto de espectro', capaz de obter as frequências e amplitudes dos parciais constituintes de um som, e calcular a sua frequência fundamental.

Uma característica importante destes objetos (e de todos os demais objetos max), estrategicamente buscada por seus autores, é a simplificação de seus algoritmos, que são de fácil (re)codificação (para outras linguagens ou ambientes) e de comportamento predizível e aceitável. São, portanto, objetos “de baixa tecnologia”, e que ocupam espaços mínimos de processamento, nos recentes modelos de computadores pessoais.

Foto de Miller Puckette, do Department of Music University of California, San Diego no endereço eletrônico(2011)
<http://crca.ucsd.edu/~msp>



Mais do que uma simples referência a uma “rabeça” (e ao violino elétrico de Miller Puckette), o *Fiddle~* funciona excepcionalmente bem em um instrumento similar. Sua performance padrão (a 1024 *samplers*) é capaz de detectar uma frequência mínima de 108 Hz, ou um pouco abaixo da nota Midi 45 (uma terça menor abaixo do Dó grave da viola).

Seus argumentos obrigatórios, ou configurações de cálculo, a serem indicados pelo usuário, são:

1º) a quantidade de amostra de análise; 128 – 2048 (1024 *samples* é o recomendado)

2º) O número máximo de polifonia – notas – para tentar encontrar; 1 – 3 (recomenda-se apenas 1)

3º) A quantidade de picos de amplitude, no espectro a considerar; 1 – 100 (indica-se 20).

4º) O número de picos de saída, se algum (recomenda-se 3).

As seguintes conexões de saída (*outlets*) são oferecidas, e divididas pelo autor em *cooked* e *raw*. As saídas "cozidas" se assemelham aos dados de um instrumento Midi, com valores de nota e amplitude nos momentos de detecção de ataques. Já os valores "crus" são fornecidos continuamente, dentro de uma faixa dinâmica pré-estabelecida. São os seguintes:

1º) número da nota (de referência Midi, acrescidos de 6 casas decimais - *discrete pitch*) e amplitude em dB (*cooked*)

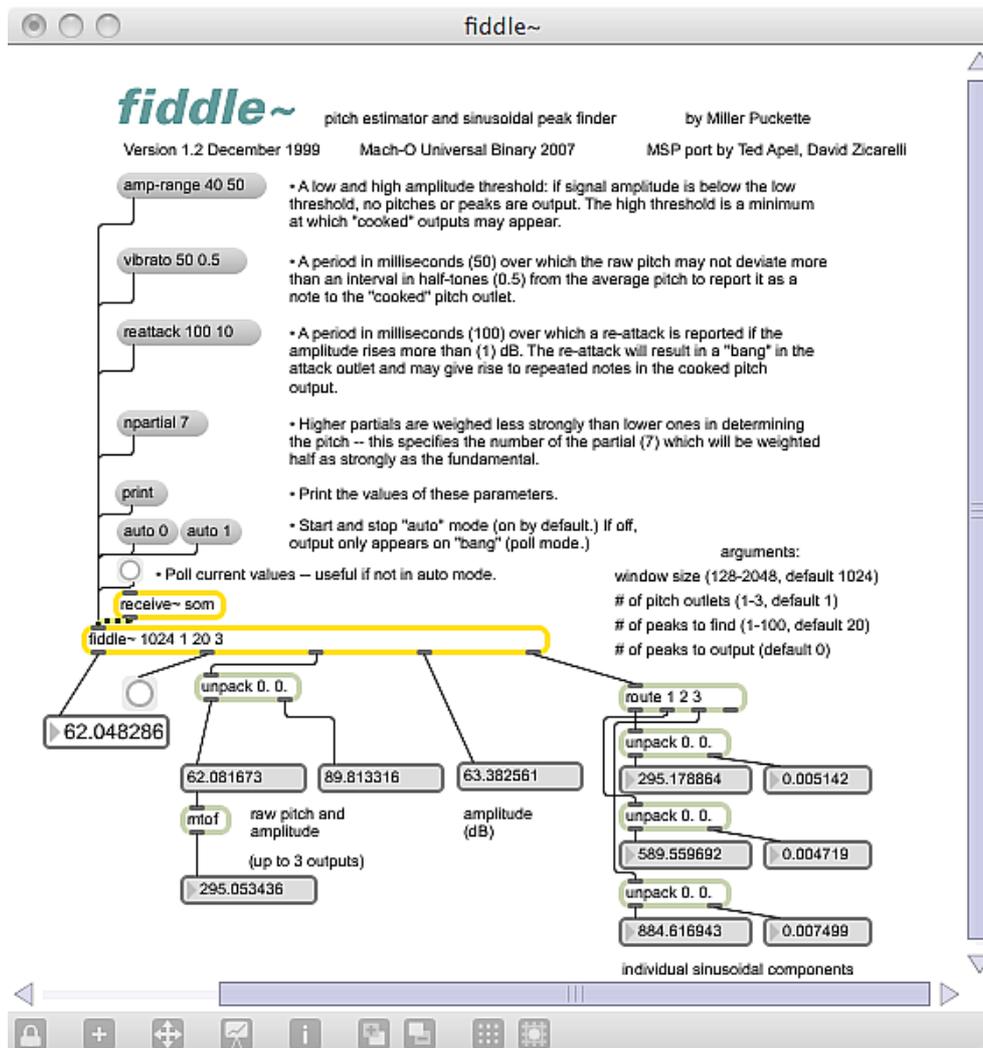
2º) ataques de notas detectadas (dispara um *bang*) - *cooked*

3º) Uma ou mais vozes de frequência contínua e respectivas amplitudes RMS

4º) A amplitude geral (em dB)

5º) Sequência de mensagens detectando picos senoidais.

Argumentos opcionais indicam ainda a extensão de amplitude (*amp-range*), detecção, ou não, de pequenas variações de frequência (vibrato), números de parciais calculados (*npartial*), além de funções de impressão de informações (*print*), e modo contínuo ou intermitente (*auto 0*).



O *fiddle~* (*pitch estimator and sinusoidal peak finder*), versão 1.2, com as mensagens reconhecíveis e conexão direta de entrada com os sinais de áudio, além de suas 5 saídas de dados (*outlets*)

Ao trabalharmos com programações gráficas no Max, temos condições de vislumbrar um pouco o funcionamento algorítmico, ou lógico e formal, dos computadores e seus programas, além de sua imensa capacidade de cálculo. Isso permite uma considerável aproximação junto a áreas inacessíveis ao músico padrão, como a engenharia elétrica ou ciências da computação.

Contudo, um simples objeto Max/msp/jitter carrega considerável carga de desenvolvimento formal, elaboração, e raciocínio matemático. No *fiddle~*, por exemplo, os dois principais problemas, envolvidos na análise discreta de espectro, se referem a obter as frequências e amplitudes dos constituintes de um som, e depois calcular a sua fundamental. M.Puckette, T.Apel, e D.Zicarelli, em artigo publicado em 1998 "*Real-time audio analysis tools for Pd and MSP*" descrevem

um pouco como ele foi construído, expondo suas severas e difíceis equações. Dizem partir das ideias já formuladas por A. M. Noll (1969) e publicadas com o sugestivo título “*Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate*”. E de L.R.Rabiner (1978) “*Digital processing of speech signals*”.

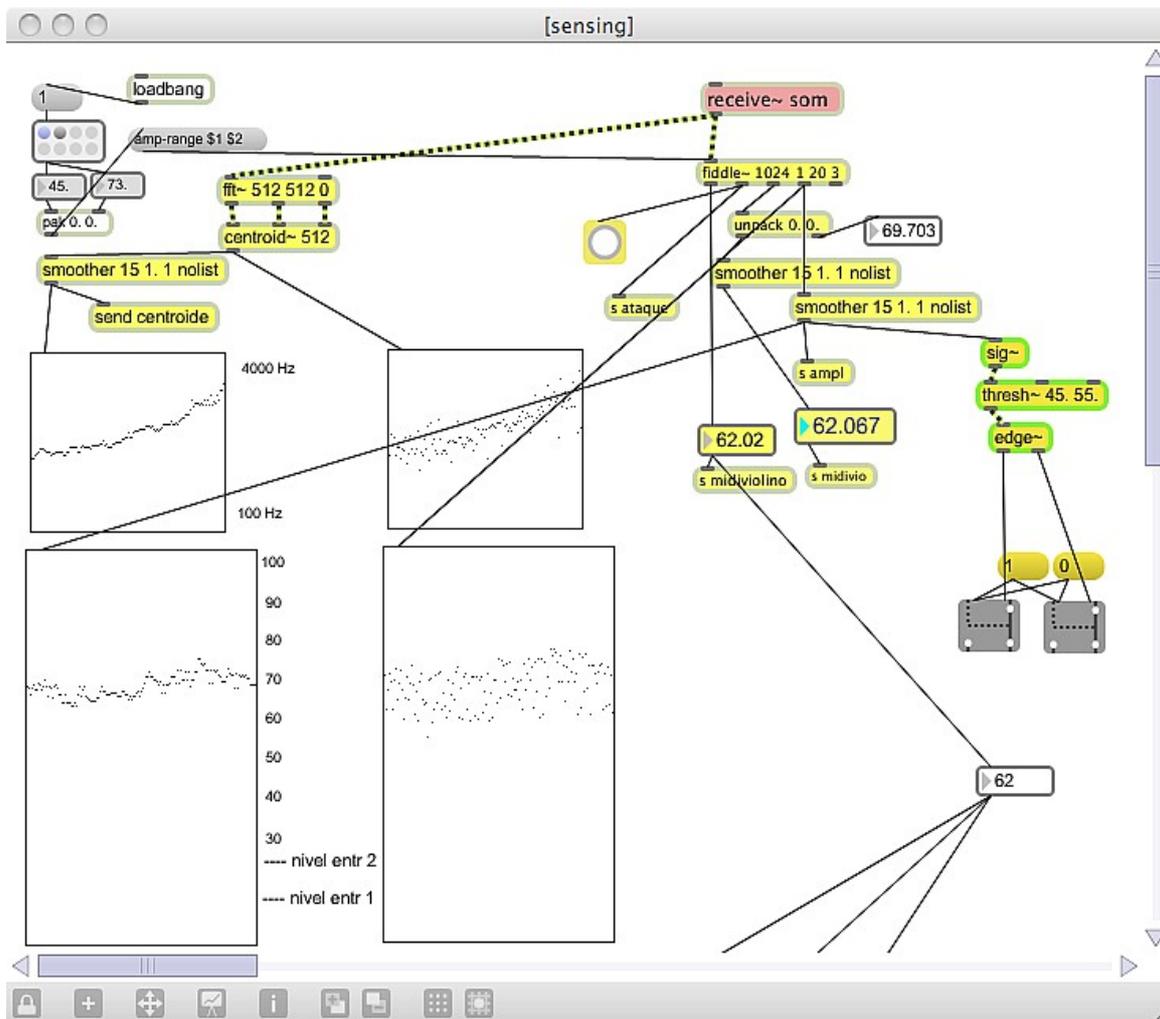
O segundo objeto lançado em 1998 juntamente ao *fiddle~*, chamado *Bonk~*, se destina a sons que não se prestam a decomposições senoidais, e gera informações úteis na detecção de ataques e modelos previamente armazenados, através do envelope global do espectro sonoro, mais eficiente que os clássicos “*envelope follower*” existentes até então. Com ênfase na velocidade de análise (sua janela, ou cálculo, se realiza sobre 256 amostras ou 5,8 milissegundos a 44.100hz), sua mais satisfatória utilização é na detecção de ataques de instrumentos de percussão. Oferece, portanto, uma rápida e eficiente comparação (escolha) a partir dos modelos armazenados.

Se não tivemos aqui oportunidade de experimentar o *Bonk~*, utilizamos contudo um terceiro objeto, chamado *Centroid~*, e que é operado junto a um *FFT~* (um objeto Max de análise por “Transformada rápida de Fourier”). O *Centroid~* se destina, então, a ponderar a relação entre os componentes espectrais de um sinal e suas amplitudes. Ele busca o resultado da média espectral ponderada pelas amplitudes, ou seja, uma espécie de “centro de gravidade” espectral. A versão utilizada (em nosso sistema) foi desenvolvida em 1998 por T. Apel, J. Puterbaugh e D. Zicarelli. Ele analisa as informações de determinada “nota” e acusa as flutuações de deslocamento de seus harmônicos constituintes.

Associados, (*centroid~* e *fft~*) possuem uma conexão de entrada (*inlet*) para o sinal de áudio, e uma saída (*outlet*) com valores delimitados entre 0 e 5 KHz.

Um exemplo, a seguir, de utilização do *fiddle~* e *centroid~*, como controladores de operações e tempo, no ambiente Max/Msp/Jitter. Uma nota Ré (IIIª corda) é tocada (com arco e em trêmolo) por um violino, e sua frequência é mostrada no *display* em baixo, à direita (pelo padrão de notas Midi). Os valores *RMS* (o parâmetro usado para disparar as seções musicais) são estabilizados pelo objeto *Smoother* (expostos em gráficos), e podem acionar os contadores, ao passar por um *thresh~* e um *edge~*, que definem os valores do nível de energia

(RMS) necessário para detecção de início e fim da respectiva seção. O *Centroid~* também analisa esta nota, enquanto ela se desloca em direção a um *Sul ponticello*.



Exemplo de uma nota em trêmolo, Re3, ao violino, (equivalente a nota padrão Midi 62), se dirigindo da região próxima ao espelho para o cavalete (de *sul tasto* a *sul ponticello*, ao longo de uns 3 segundos aproximadamente). Os gráficos da esquerda tem seus valores estabilizados por objetos “*smoother 15 1. 1 nolist*”.

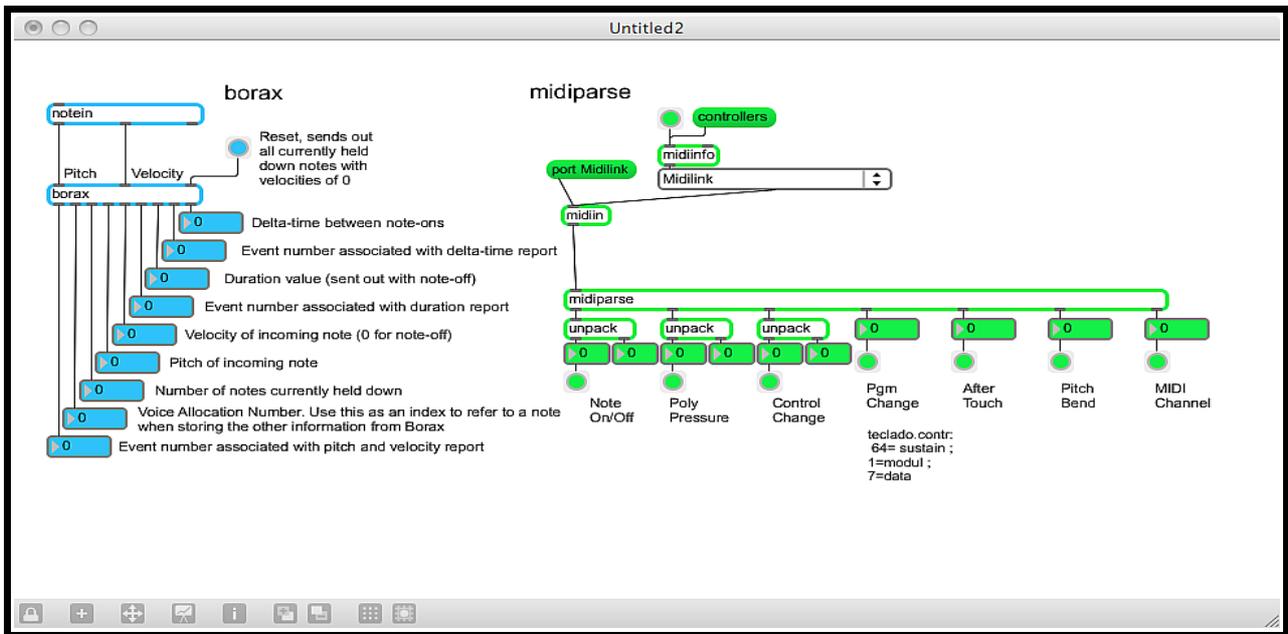
1.3 Ferramentas Midi, *wimotes* e software Osculator

O protocolo Midi, principalmente junto aos instrumentos do qual surgiu (um órgão sensitivo ou um piano insensível), ainda é uma simplificada e conhecida linguagem comum, uma interface utilizada não só em teclados e sintetizadores, mas entre programas e computadores.

Sendo uma interface adaptada de outra robusta interface (teclados musicais temperados já contam com mais de trezentos anos, e produzem música polifônica e de forma sintética), a utilização desta linguagem comum se estende dos instrumentos de brinquedo aos mais sofisticados, como em um piano digital *Disklavier*, com suas teclas que se movem “sozinhas” e atraem a atenção para o intérprete ausente. Estes mecanismos e dados Midi representam, talvez, uma maneira simples (portanto ancestral e difícil de ser apagada) de se fazer música, com sons manipulados e disparados pelas mãos, ou percutindo teclas com os dedos.

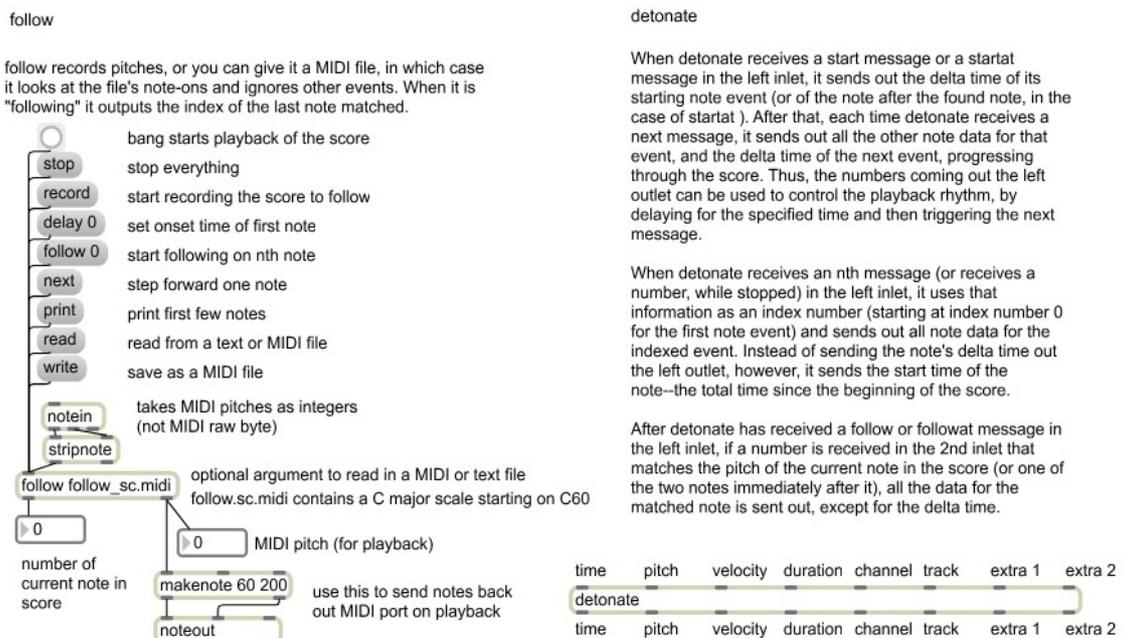
Um teclado controlador Midi (que atualmente se conecta aos computadores via cabos USB) é capaz de desempenhar relativamente bem as funções de transmissão do desempenho de um intérprete, apesar da defasagem tecnológica do seu protocolo e de sua baixa velocidade de transmissão. Seus dados, porém, são fáceis de analisar, interpretar e manipular.

Alguns objetos Max se destinam a fornecer descrições primárias desta performance, como o “*borax*”. Entre as informações fornecidas está, por exemplo, o do número de notas pressionadas (*hold down*). Ou ainda objetos de amplo reconhecimento de mensagens Midi, como o “*midiparse*”, que recebe as informações gerais (*raw*) e as decodifica nas várias categorias existentes. (*cc, note, pitch bend, velocity, after touch...*)



O objeto *borax* é utilizado para analisar a performance Midi executada em teclados, enquanto o *midiparse* reconhece e separa os diversos tipos de mensagens.

O *Follow* é um objeto tradicionalmente usado para acompanhar intérpretes, ou seja, um “seguidor de partitura”. Ele realiza o reconhecimento da sequência prévia de notas (Midi) e informa sua posição na cadeia de eventos (numerada). Outro objeto Max similar, porém mais abrangente, é o *detonate*.



Dois objetos seguidores de partitura (Midi), com seus *inlets* e *outlets*.

Nem mesmo o uso de *follows*, ou outros seguidores de eventos automáticos, estão isentos de uma certa “justificativa” quanto a sua utilização, pois é preciso lembrar que há sempre um forte concorrente, menos falível, simples e sem dúvida mais constrangedor: - o uso de mecanismos simples de acionamentos, como “botões” ou pedais, diretamente pelo intérprete, dentro do seu conjunto de ações.

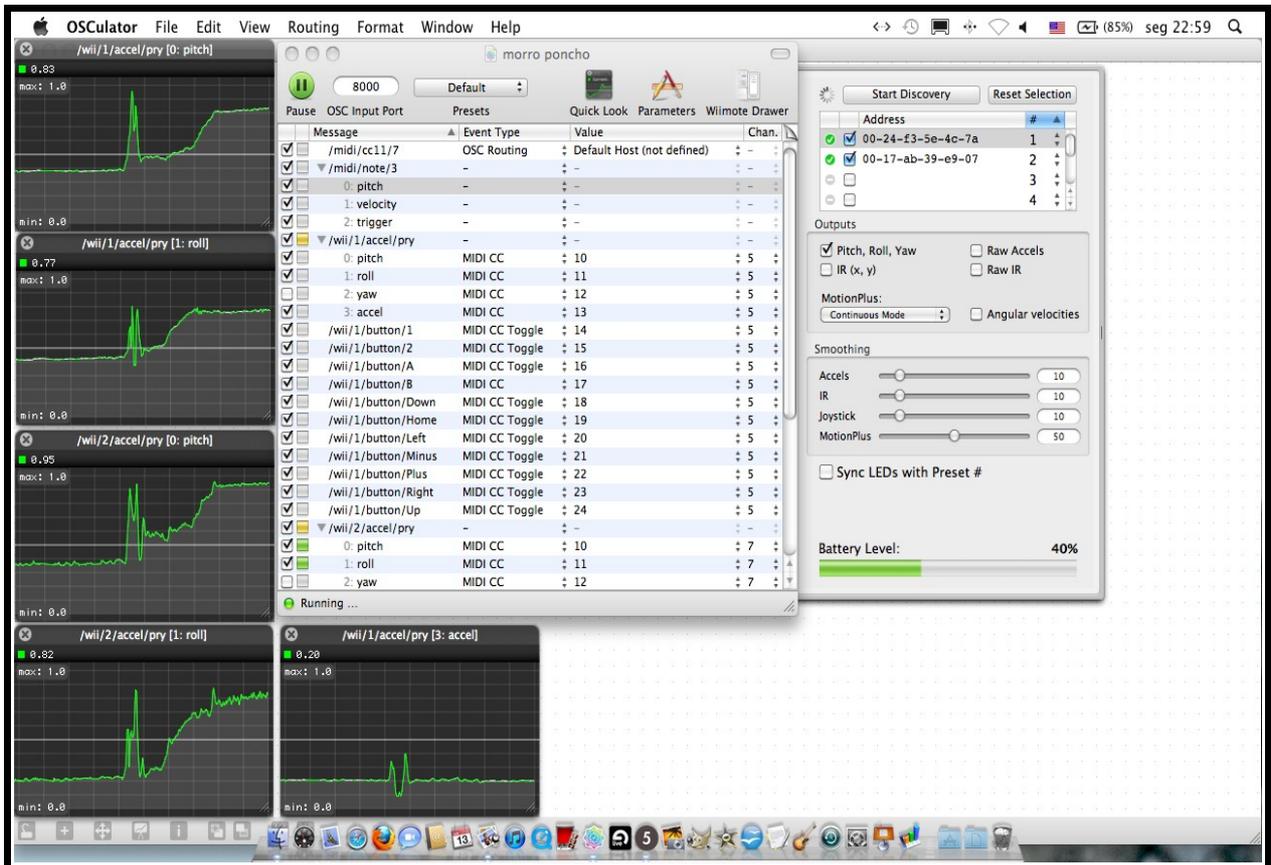
No caso do uso de botões e sensores (mais elegantes que os do teclado ou mouse de um computador, e talvez menos que equipamentos mais sofisticados, porém bem mais caros) dispomos de *joysticks Wiimote*, de transmissão *bluetooth* (radio-frequência de curto alcance) do aparelho de videogames da Nintendo Wii. Capaz de transmitir sinais contínuos de variações em seus eixos e aceleração, e contando também com vários botões simples de contato, ele é aqui decodificado por um software chamado Osculator. Embora o sinal *bluetooth* possa ser recebido e decodificado via programação Max, o software aqui utilizado é específico, de baixo custo e eficiente. Retransmite, mapeando, as informações do *joystick* para todo o computador, convertidas (ou não) em informações Midi: - *cc* (*control change*, variando de 0 a 127), ou *cc/togle* (0 ou 127).

Ao ligar um *notebook*, e reconhecer a transmissão de um *wiimote*, podemos, por exemplo, disparar ou manipular notas Midi a partir de relações primárias com gestos e movimentos (neste caso captados apenas os exercidos sobre o aparelho). Seja na utilização como um novo instrumento - baquetas eletrônicas que disparam sons ao serem percutidas no ar, acionando clips através de suas teclas, ou controlando o processamento de sinais do Live (por exemplo) pela inclinação de seus planos, o *wiimote* é uma robusta manete para videogames, acessível (de baixo custo) e de comercialização popular. E, mais recentemente, eficiente também como controlador junto à música por computador e demais atividades performáticas.

A utilização de acelerômetros acoplados ao corpo de instrumentos deve ganhar maior interesse, no futuro, principalmente se lembrarmos que são componentes pequenos e configuráveis através da montagem de interfaces físicas relativamente simples (cuja mais conhecida se chama “Arduíno”)⁸. Se o tamanho de um *wiimote* é inviável em outras utilizações que não como uma “manete

⁸ Ver nota na página 99.

controladora”, não devemos negar a potencialidade de seus mecanismos vitais (os acelerômetros e transmissores *bluetooth*), ou mesmo abrir mão de adquirir equipamentos mais caros. Por que não captar as variações de plano detectadas nos acelerômetros fixados ao corpo de instrumentos tradicionais, por exemplo ?



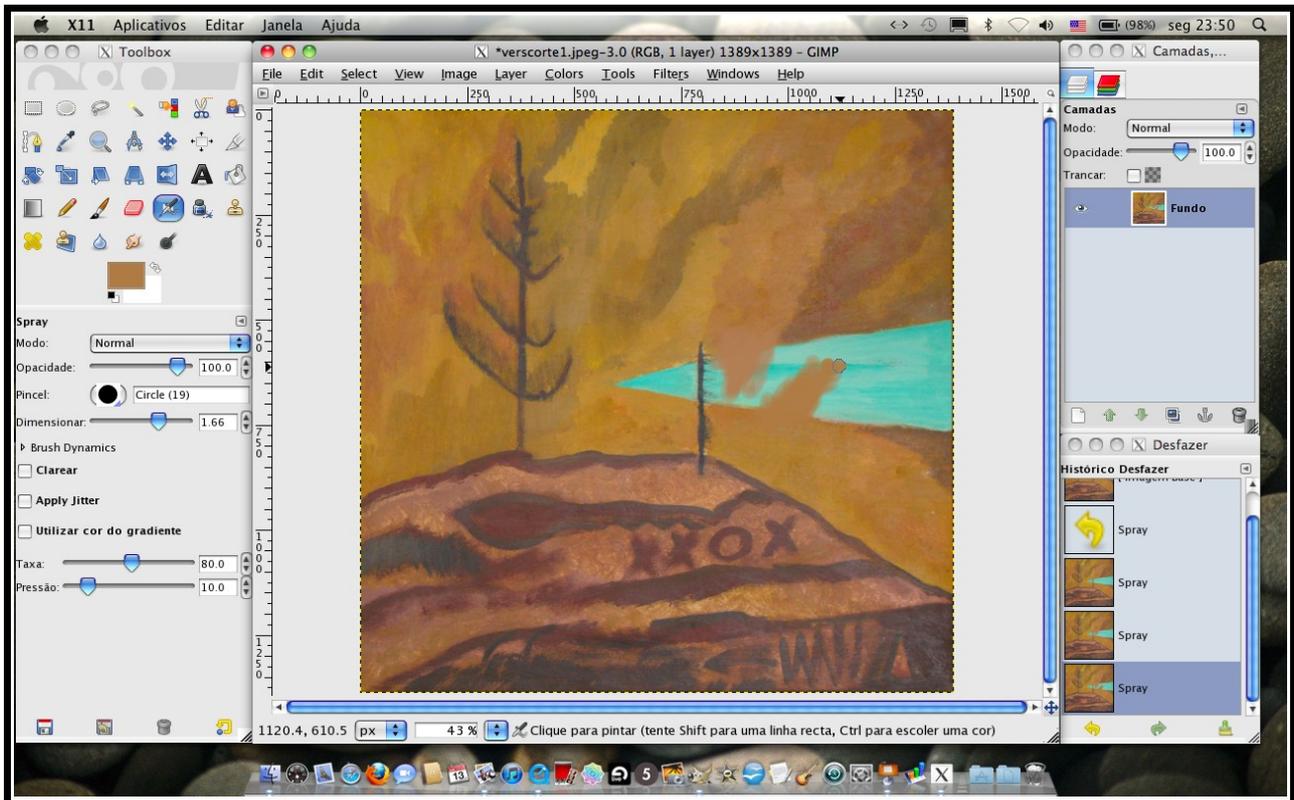
Tela do software Osculator. Os 5 gráficos escuros à esquerda descrevem as alterações nos acelerômetros. A janela do meio é a grade de “*mapping*” dos comandos, com sua conversão em mensagens midi (tipo de *cc.nº./canal*). À direita, janela de reconhecimento e conexões *bluetooth* com as manetes *wiimote*, tipo de saída de sinais e taxas de “*smoothing*” (este último responsável por regular a sensibilidade e reação, torná-las mais lentas ou rápidas e agitadas)

1.4 Captação de Imagens e Produção de Matrizes.

Fotografar uma imagem se compara diretamente ao ato (e suas consequências) de se gravar um som. Escolhe-se um tipo de equipamento (câmera e lentes, ou modelos de microfones e gravadores), um lugar e posição onde fixá-los, e como balancear seus constituintes (fontes de luz/som) e suas equalizações (capacidade de captação destes dados).

Se em nosso sistema a captação musical será ativa, recebendo informações ao longo do processamento, o mesmo não podemos dizer das imagens, que não foram “produzidas” em tempo real. No entanto, respeitando a analogia com sistemas musicais, o estágio de captação aqui não se encerra com o *click* fotográfico da câmera. Utilizando programas conhecidos como Processadores de Imagens (cujas versão mais famosa é o “Photoshop”, da Adobe) estes softwares são sinônimos de apagar coisas e melhorar a aparência das pessoas nas fotos oficiais. Com eles é possível, então, criar variações, limpar elementos ou incluir outros, em manipulações digitais análogas aos modos artesanais de trabalhos com colagens, tintas, e demais materiais gráficos. Funcionam como um ambiente de desenho tradicional, com ferramentas como pincéis, aerógrafos e canetas operados por mouse ou interfaces mais apropriadas (*Tablet/pen*). Particularmente, utilizamos uma versão de um software livre, chamado GIMP, disponibilizado pela Sourceforge.net

Refazer a imagem original traz a possibilidade de selecionar e alterar apenas algumas partes, criando variações que nos permitem criar a percepção de movimento, quando apresentadas em sequência, em um processo conhecido (cinematograficamente) como “*stop-motion*”. Para isso, excluímos o facho verde da pintura e depois criamos outras 15 versões diferentes, alterando apenas os desenhos das manchas no céu, mantidas as tonalidades originais mas orientadas, uma parte à direita, outra à esquerda, e outras ainda com padrão mais neutro.



Tela de manipulação gráfica (processamento de imagens) do software GIMP. No lado esquerdo temos uma caixa de ferramentas (com seletores como pincéis, aerógrafos, “dedos”, borrachas, etc.). O lado direito possui, além do histórico das ações, no canto inferior, a estratégica função de controle de “camadas”, em cima. Embora a tela principal apresente a imagem inteira sendo trabalhada (ampliação a 43%), isto ocorre, na verdade, em dimensões de escala bem maiores (superiores a 200%), a fim de buscar maior resolução e qualidade.

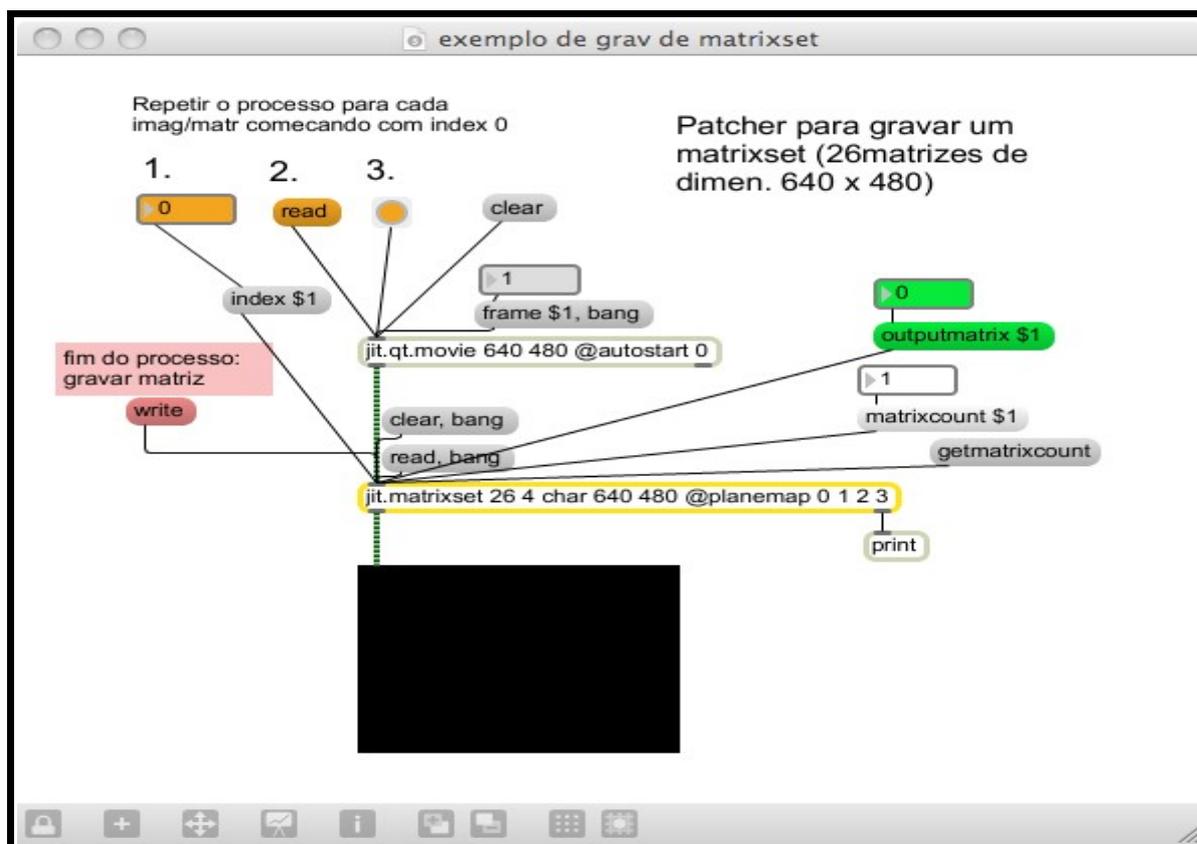
Em nosso caso, alternamos estas 16 versões de modo a intercalar as mais neutras (a, b, c, d, e) com os grupos direcionados à direita e esquerda, e também gravá-las isoladamente no final de nossa sequência, a fim de poder também dispor separadamente deste grupo.

Matrixset A : (total de 26 matrizes)

0= c2 ; 1= c ; 2= BB ; 3= b ; 4= b4 ; 5= d ; 6= b5 ; 7=b3 ;8 = c; 9= b2 ;
 10 = d ; 11= d2 ; 12 = a ; 13= a4 ; 14= b ; 15= a2 ; 16=e ; 17=a3 ; 18= c ; 19= a4 ;
 20= e2 ; 21= a ; 22= b ; 23= c ; 24= d ; 25= e

O objeto básico de manipulação de imagens no Max/Jitter, responsável por configurar e armazenar estes dados em sua disposição gráfica bidimensional, é o *jit.matrix*. Para se criar uma matriz convencional - "*jit.matrix 4 char*", as imagens de fotografias (.JPG) são transformadas em um formato específico (.JXF), onde mantêm seus quatro planos - ARGB (sendo A destinado à transparência – *Alfa* ; R ao valor de vermelho – *Red* ; G para verde - *green* ; e B para Azul – *blue*) . Cada um destes planos recebe valores de 0 a 255 para cada pixel, fundindo assim sua cor original a partir das cores (de luzes) básicas representadas pela paleta ARGB. Uma matriz de imagem é capaz, portanto, de operar sobre cada pixel, e sobre seus três valores de cor (além do canal alfa, destinado a informar a transparência desta imagem).

Já o objeto *jit.matrixset* comporta um conjunto de matrizes (*set*), tornando possível armazenar várias delas, na sequência desejada, e acioná-las livremente na cadeia de processamentos (em *output*).



Patch criado para formatar as *jit.matrixset* utilizadas neste projeto. Não existem tutoriais (ao menos que conheçamos) sobre sua utilização específica, embora elas possam ser naturalmente auferidas a partir de *jit.matrix* convencionais.

[Intermezzo] -

Entre dois estágios: a dimensão do tempo e da escrita.

I -Pragmatismo em Sistemas Musicais Contemporâneos

Muitos são os caminhos, e muitos os sinais elétricos até se transformarem em música. Em discussões envolvendo Sistemas Musicais Interativos, muitas são também as definições, os conceitos, e o número de estágios. Satisfaço-me com apenas dois, “Captação” e “Processamento”, desde que, entre eles, eu possa ressaltar um terceiro estágio, ou melhor, um conceito: o da escrita, sobre o domínio do tempo, da música proposta ao computador.

Falar em estágios separados como “processamento” e “resposta” talvez indique a falsa ideia de que há algo além do processamento, que uma simples conversão digital-analógica. E não há. Mesmo em nosso caso específico, com produção de sons e imagens, o que temos como “respostas” são simplesmente *outputs* de áudio e vídeo. Prefiro chamar este terceiro estágio pelo nome claro de “resultado”, preservando melhor sua relação orgânica com a origem e percurso do processo. Assim como procedemos neste nosso *Intermezzo*, cuja localização e importância de seu tema não se encontram em nenhum lugar separado e específico do processo, mas em toda a sua cadeia de eventos, o resultado, ou se preferir, “a resposta”, deveria estar presente como parte dos objetivos, desde o início deste percurso.

Winkler (1998 :6) lembra-se de incluir o estágio da “*computer composition*”, algo que considero fundamental para se definir a passagem e acompanhamento do tempo em uma apresentação musical, uma questão de se informar ao sistema “o quê” e “quando” executar suas ações.

As maneiras de se administrar o tempo (e, portanto, acionar as respectivas ações de uma peça) foram definidas inicialmente por Rowe (1993), e mantidas relativamente inquestionadas desde então, como pertencentes a dois paradigmas básicos : “*Score-driven*” ou “*Performance-driven*”.

Score-driven são aqueles em que o software se utiliza de informações previamente fornecidas para confrontá-las com os dados do intérprete, que o computador então recebe, seleciona, e produz outros eventos, expandidos hoje a qualquer tipo de categoria, pois Rowe, em 1993, ainda falava em receber “*music*” e notas Midi por seus *inputs*.

Performance-driven programs, não possuindo dados ou expectativas de realizações de nenhum evento prévio, mas usando parâmetros expandidos (densidade e regularidade, por exemplo) seriam sistemas capazes de interagir com uma variedade maior de *inputs* do intérprete.

As concepções acima pressupõem a existência (ou não) de uma partitura entendida em seu sentido tradicional. Para um Sistema musical contemporâneo (como este, realizado de 2009 a 2011), cujo leque de opções de “*sensing*” é bem maior que o da época do livro de Rowe, lidar com o tempo musical passa a ser parte integrante da programação, incorporando características tanto da escrita prévia (*score*) como da interpretação (*performance*).

A simples existência de um software já implica na escrita, ou na existência de dados prévios, ao sistema. Programas com interfaces gráficas, como o MAX, são propostos sobre caminhos claros para cada *bang* e fluxo de dados. O que mais é isto, dentro de um sistema de computadores, senão oferecer a ele uma partitura digital?

O conceito de se “seguir uma partitura (tradicional)” ainda traz outra discussão importante. “*Score following is, however, more reactive than interactive*”, nos diz Drummond (2009:127), citando ainda Jordà (2005:85) : “*score-followers constitute a perfect example for intelligent but zero interactive music systems*”.

Me pergunto simplesmente o que vem a ser a interatividade, senão uma forma reativa do sistema. O que mais podemos esperar, fornecidos os comandos e instruções, além de respostas reativas (e que espera-se bem programadas) deste mesmo sistema?

Além disto, é preciso sempre reavaliar os resultados e seus reais significados, minimizados alguns deslumbramentos iniciais que estas descobertas e novidades nos causam. Para quê usar sofisticadas relações de interatividade quando a mesma ação pode ser desencadeada por um simples botão, junto ao conjunto das demais ações do intérprete? Tudo pode ser monitorado, analisado ou processado. No entanto, o que é realmente necessário?

Diferentes níveis de interatividade estão presentes em toda parte, desde o momento em que alguém conecta um amplificador ou PC à uma fonte de energia

elétrica. Seria um erro tentar definir conceitos tão amplos e delicados, como em *“interactive music systems”*, tentando restringir ou condicionar seus significados. M.Schafer e J.Cage se dispuseram, certa vez, a tentar uma definição para o termo “música”: Cage logo identificou o perigo e admitiu que deveríamos considerar os sons à nossa volta, estejam eles dentro ou fora das salas de concerto.⁹

⁹ O Ouvido Pensante, M.Schafer (Edit. Unesp), pág.120 : “Música é sons, sons à nossa volta, quer estejamos dentro ou fora de salas de concerto”

II – Nenhuma novidade no *Front*

Se ao fazer música, juntamente a um sistema, é necessário programar um computador, esta programação, por sua vez, só faz sentido se trabalhada junto às condições e coordenadas do compositor, e seu discurso formal. Antes de cobrarmos estas decisões específicas do campo “musical” ou estético, cabe ressaltar ainda algumas considerações acerca do artigo *“Loop-interpolation-random & gesture: déjà vu in computer-aided composition”*, de Silvio Ferraz e Leonardo Aldrovandi (2000 -revista *Organized Sound* 5 (2) :81-84).

Os autores, tentando responder à questão “o que faz sequências de sons se transformarem em música?”, abordam inicialmente as três formas, ou níveis, de escuta: -Textural, figurativa e simbólica, e de onde aparece a “expressividade”:

“Expression is born of a clash; it is not a separate entity. Whenever different discourse or language territories collide or enhance each other there is expression.”

Ferraz e Aldrovandi dizem que seria muito simples achar que a presença de um intérprete, e seus gestos (às vezes obsoletos e desvinculados de uma cadeia de significados) trariam a última solução aos problemas da música contemporânea, que são, em última análise, os problemas da criatividade contemporânea. O uso de *loops*, interpolações e jogos aleatórios de permutações, procedimentos tradicionais já reconhecidamente aceitos pela comunidade acadêmica, não garantem a transformação de seus resultados em “música”.

“Problems with eletroacoustic music, computer music, computer-aided composition and so on, are the very problems of contemporary creativity. By supressing the construction stages of sonic or musical objects and ascribing to software designers the assurance of musical acceptance, composers are receiving the key to enter a specific community and are risking their creative spirit “

Terminam com a proposta, ou impasse, para os “gestos”:

“So, in a non-pre-established situation it is important to note that the path ahead may be one where gestures are not defined; where gestures will be not an imitation of 'gesture', but a product of new ways of computer-composer-music interation.”

Estas são algumas das consequências de se enfrentar a música aqui proposta. Possuímos equipamentos extraordinários (se comparados aos de duas décadas atrás, por exemplo) mas ainda temos dificuldades em nos conectarmos simbólica ou espiritualmente a eles, e até mesmo à nova linguagem musical contemporânea, que eles evocam. Dificuldades naturais, frente ao desafio de controle de tantas variáveis e alternativas. Embora comuns, *loopings*, interpolações e processos aleatórios estão longe de terem seu desenvolvimento e exploração esgotados. Nem mesmo se trata de substituir a orgânica relação gestual entre os instrumentos tradicionais e as pessoas.

Existe ainda uma infinidade de propostas de produção de música e vídeo a partir de concepções mais científicas, com vínculos a processos biológicos, matemáticos ou de engenharia computacional. Nossa preocupação, no entanto, é construir um sistema confiável e eficiente, e que possa servir às pretensões artísticas (ainda que excêntricas ou demasiado subjetivas). Para um músico que terá sua interpretação gravada e imediatamente reproduzida, é conveniente deixá-lo concentrar-se apenas em sua performance, devendo o computador desempenhar suas ações de maneira automatizada. Este foi o sentido buscado na elaboração de uma obra experimental que ocupará nossas atenções daqui por diante, com suas 41 seções em três partes distintas. Nela, um músico é acompanhado de maneira semi-automática pelo sistema (que desempenha todo o processamento de áudio e vídeo sem necessidade de coordenação do intérprete), ao longo das duas primeiras partes, carregando-o com sons que ele produz em sua performance junto a seus instrumentos sonoros. Já na terceira e última parte, o intérprete passa a administrar este banco de sons, seu processamento de áudio, e a maior parte dos comandos de vídeo diretamente através de dois *joysticks wiimote*, em uma literal manipulação – algumas das funções estão associadas não a botões, exclusivamente, mas aos seus acelerômetros. Aumentar o volume, por exemplo, é alcançado girando o aparelho em um de seus eixos.

III - Um Roteiro Musical

Embora cercada das naturais limitações, que nos impedem de representar satisfatoriamente música e imagens em movimento, esta versão de “partitura visual”, ou “roteiro musical” ajudará a ilustrar, juntamente com as tabelas das pág. 72 a 75, as sequências envolvidas ao longo do tempo.

Nossa imprecisão na escrita musical, principalmente quanto à métrica e andamento, ajuda a consolidar a independência dos clips, com sua autonomia de tempo e expressividade (percorrendo seu percurso e existência enquanto frase musical, ainda que repetida mais tarde sobre a forma de *loop*). Tudo pode ser copiado, processado e disposto conjuntamente, cada qual mantendo seu isolamento temporal e sua heterogeneidade, frente ao restante do grupo (Inserimos, inclusive, um *clip*, na seção 40, onde será gravado uma amostra de transmissões de rádio). Esse não deixa de ser um desafio artístico contemporâneo, ou seja, o de encontrar “ordem” junto ao caos, ou ainda manipular conjuntos deliberadamente caóticos (o que é mais realista e possível), nesta pluralidade inevitável de formas e experiências musicais (e visuais) possíveis.

Em nossa moderna Instrumentação envolvendo um Sistema Musical Interativo, além dos instrumentos sonoros pessoais do intérprete, temos uma lista (tão reduzida quanto possível) de softwares e hardwares:

Instrumentação

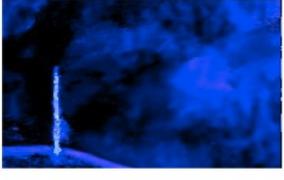
- um violino elétrico de 4 cordas
- Um teclado controlador MIDI
- Um rádio pequeno de pilhas.
* * *
- Um computador (*macbook*)
- Uma placa externa de áudio (*full-duplex*)
- Um projetor
- Dois *wimotes*
- Software Max/msp/jitter 5 (programação principal)
- Software Live 8 Lite (áudio, loops e processamento de sinais)
- Software Osculator (recepção e *mapping bluetooth* do wii)

Teste nº1

(1. Violino)



Fiddle~

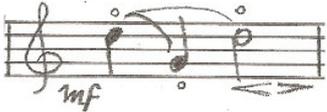


Fiddle~



Pedal

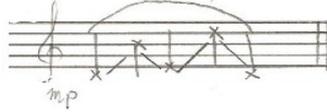
1



2



3





Canais Live
GRAY. REP. PED. FADE
Midi: 6 2 7 →
5 →
4 →
3 →
2 →
1 →



Pedal

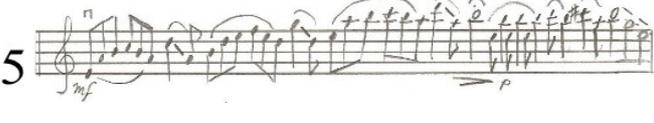


Pedal

4



5





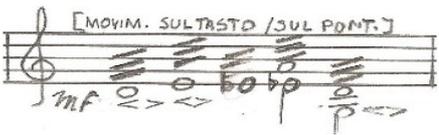


Pedal



Pedal

6



7



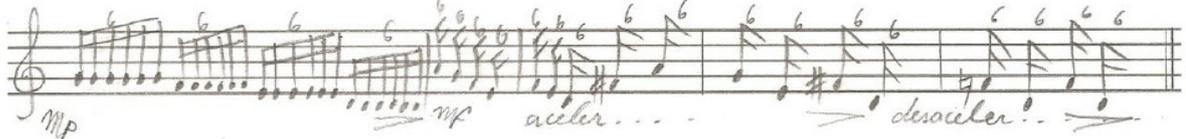




Pedal

Pedal

8





Pedal

Pedal



Pedal

Pedal

9



10



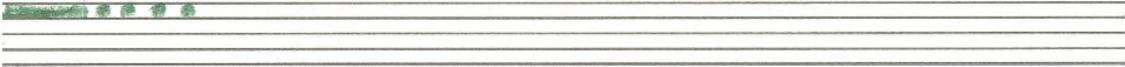


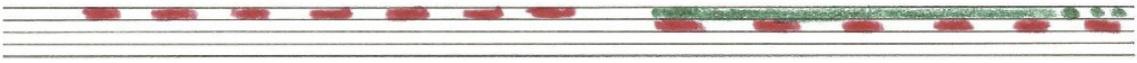
Fiddle ~

[Pedal: de 0 a 12]

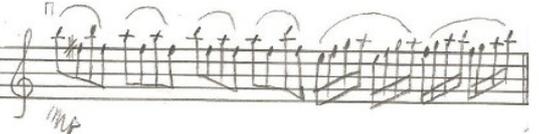
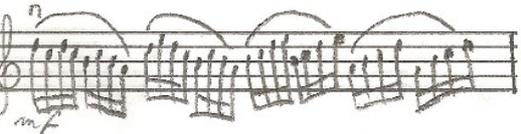
Wii(branco): "home"

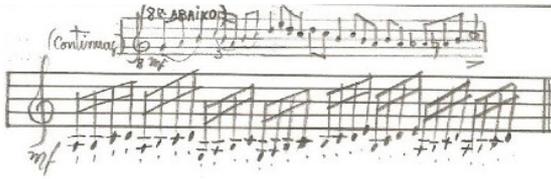
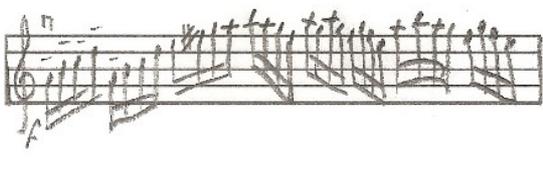
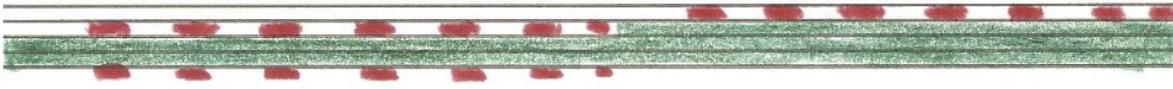
11 [1ª CADENZA]

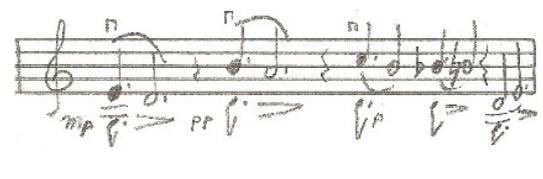


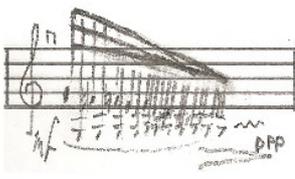
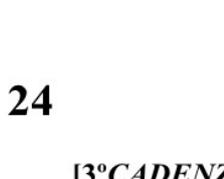
	
Pedal	Pedal
12 	13 
	

	
Pedal	Pedal
14 	15 
	

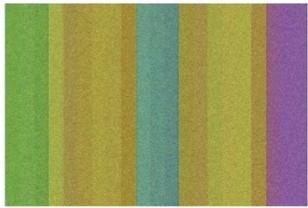
	
Pedal	Pedal
16 	17 
	

				
	Pedal	Pedal	Pedal	Pedal
18		19		
				

				
	Pedal	Pedal	Pedal	Pedal
20		21		
				

					
	Pedal	Pedal	Fiddle~ wii:h	Fiddle~ Fiddle~	
				Fiddle~ autom.	
22		23		24	
				25	
					

(2 - Teclado)



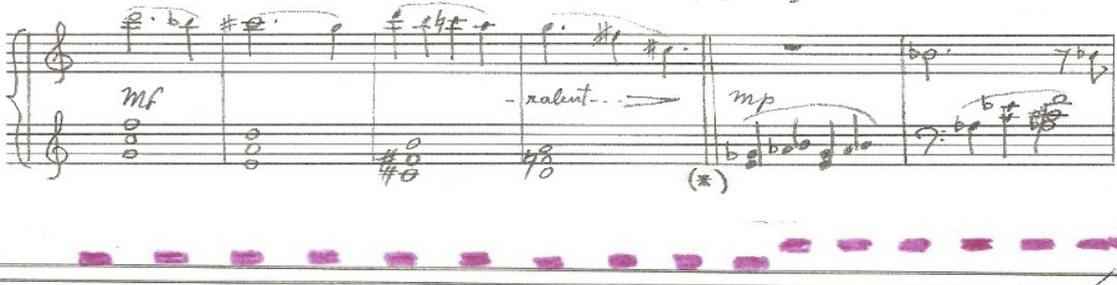
Pedal

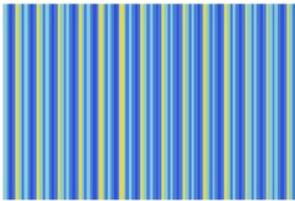


Pedal - Automat.

26

27

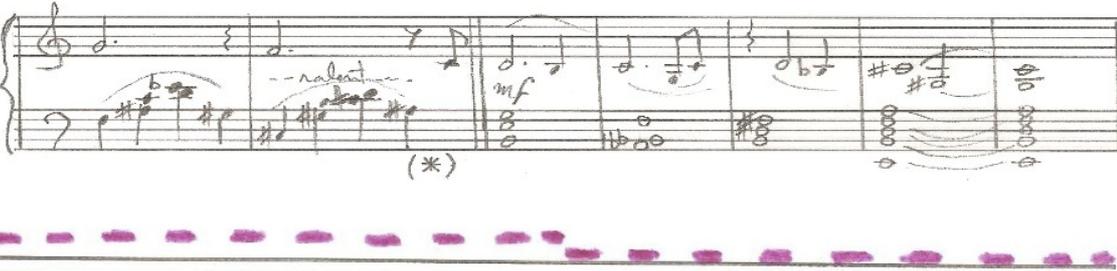


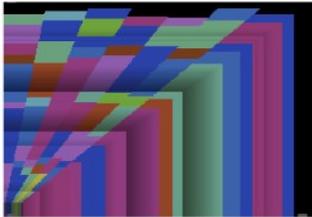


Pedal - Automat.

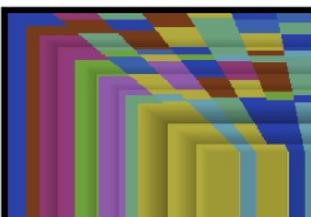
27

28





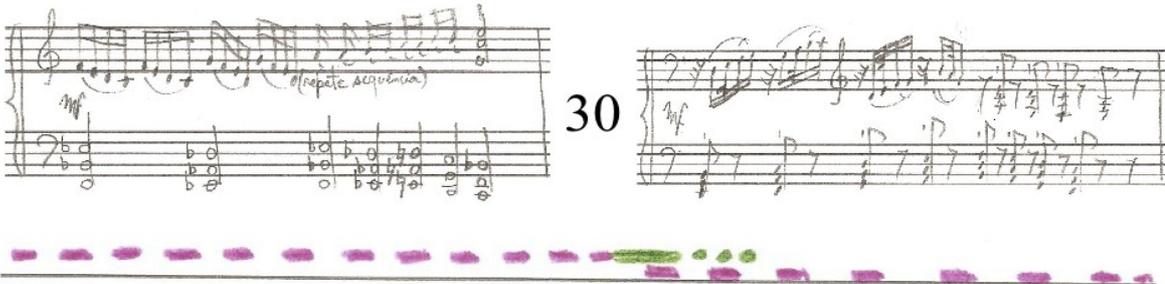
Pedal

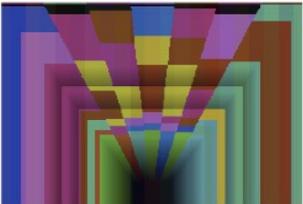
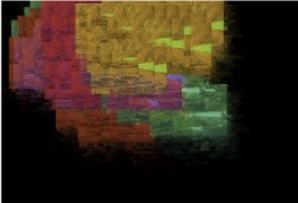


Pedal

29

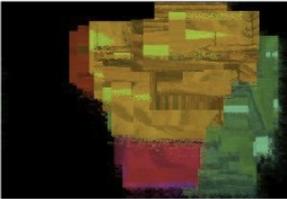
30



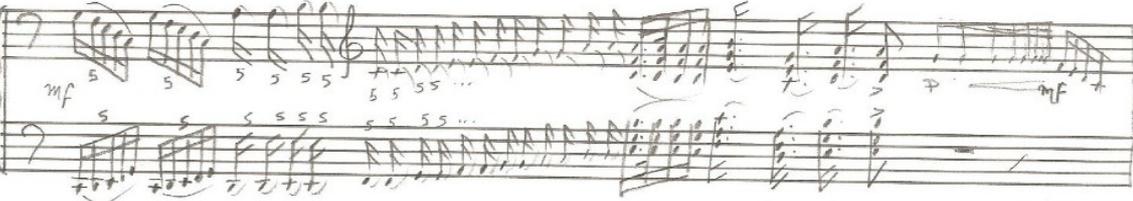



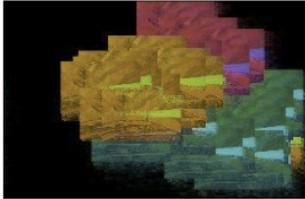
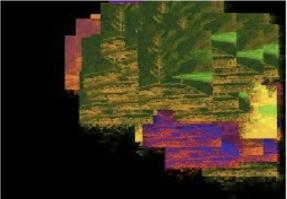
Pedal Pedal Pedal Pedal

31  32 

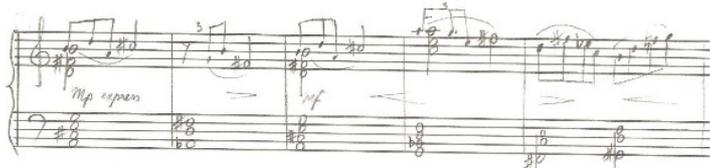



Pedal Pedal

33 

Pedal Pedal Pedal Pedal

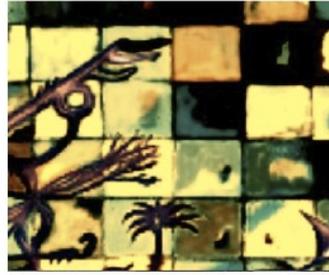
34  35 



(3 - Wiimotes)



Wii home autom.



Autom Wii home

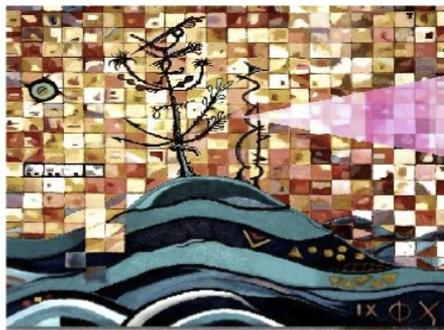


Wii home Wii home

36 Transição

37 {Sequência - I}

38 {Sequência - II}



Fiddle~ Wii home



Wii home Wii home

(Violino) : [4^o CADENZA]

39



{Microfone /rádio FM}

40

Gravação de aprox.30 segundos de amostras de gravações obtidas a partir de um rádio de pilhas FM



Wii home

Wii home

41

{Sequência – III } (wiimotes)

2º capítulo:

Processamento – O 2º e último estágio de um sistema.
Um estudo de caso: "*Teste nº1*"

2.1 Construindo um Sistema Visual.

Criar um acompanhamento visual a ser oferecido ao nosso intérprete musical, para ser projetado em parede ou tela, assemelhando-se a um vídeo de animação ou um simples cenário digital, não significa apresentar somente resultados artisticamente subjetivos. Os mecanismos apresentados neste capítulo, de qualquer forma, além de serem eficientes e genéricos o bastante para serem amplamente utilizados como base em outros trabalhos ou projetos, estão a disposição de qualquer músico que queira ocupar-se com imagens ou cenários, durante suas apresentações.

A programação visual (assim como os demais aspectos formais deste *Teste nº1*) está dividida em três partes. Sua estrutura se assemelha diretamente ao conhecido formato musical $A - B - A'$, onde ao mecanismo principal de projeção “A” se sucede um contraste “B” (um divertimento para teclados), e novamente retornamos aos dispositivos de “A”, embora um pouco alterados. A primeira parte, portanto, abarcando as primeiras 25 seções, terá um *patch* específico de programação chamado “*ControlimagA*”, e será abordado neste subcapítulo (2.1). A parte B, que vai da seção 26 à 36 será tratada no subcap. 2.3, e seu *patch* é o “*ControlimagB*”. O último, “*ControlimagC*” é uma ligeira variação do primeiro (A'), porém conectados com algoritmos presentes em outro *patch*, “*Sensing*”, onde se localizam os processamentos das informações dos *wiimotes*. A tela principal (*patch*) de programação, por onde se acessam os *patches* (secundários) acima, se encontra na pág. 102.

Não é tarefa das mais generosas ter que recortar estas telas de programação ao nosso formato padrão de páginas A4. Mas tentaremos seguir a ordem de ocorrência dos eventos, e apresentá-los da maneira mais “limpa” possível (excluindo o excesso de conexões, para simplificar a leitura). Uma gravação em vídeo (DVD), do primeiro teste envolvendo este sistema, acompanha este texto, e ajudará o leitor a compreender melhor os resultados até então alcançados¹⁰.

10 Ver nota da pág. 101. Performances mais recentes poderão ser acompanhadas pelo site oficinadoluiiz.wordpress.com

Processar imagens é uma tarefa que costuma requerer uma parcela significativa da capacidade (máxima) de processamento de um computador. As duas principais variáveis que irão influenciar esta utilização serão o tamanho desta imagem (e portanto a qualidade da resolução) e o número de imagens que serão disparadas por segundo, dentro da cadeia de cálculos (ou número de quadros por segundo), assim como ocorre com o processamento sonoro, com suas taxas de amostragem e resolução. Na busca pelo equilíbrio entre bom desempenho e boa resolução, somos conduzidos a um estudo mais minucioso dos algoritmos e processos envolvidos, e a obtermos, sempre que possível, simplicidade e refinamento neste funcionamento.

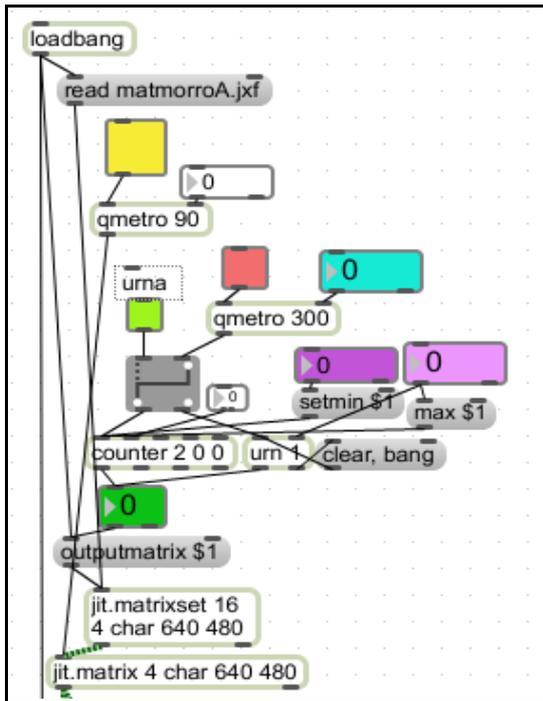
Quanto ao tamanho de nossas matrizes (imagens), percebemos que dificilmente conseguiríamos um bom desempenho com uma qualidade atualmente conhecida como “alta definição”, e nos contentamos com menos da metade disto (640 x 480) suficientes para uma projeção convencional VGA (lembrando ainda que a capacidade de processamento de nosso computador - modelo básico 2009 - será compartilhada por três programas: O Max (ocupando aproximadamente cerca de 57% deste limite), o Live (27%), e o Osculator(7%).

O mecanismo de projeção, responsável por disparar as imagens para processamento, primeiro componente do *controle de imagens A e C*, requereu que buscássemos sua máxima eficiência, tal como nos mostram os dois exemplos a seguir, ambos controlando os ritmos de alimentação de imagens e de troca de matrizes. No primeiro, dois *qmetro*¹¹ (o “q” à frente do *metro*, abreviação de metrônomo, serve para dar baixa prioridade aos seus “bangs”¹²) acabam se sobrepondo, criando pequenos acúmulos de sobrecarga, ou irregularidades, no ritmo da projeção (polirritmia). O segundo exemplo, além de gerenciar melhor o fluxo de troca de matrizes e alimentação do sistema, proporcionando um fluxo contínuo e estável de *bangs*, utiliza um único *qmetro*, e ainda opta pelo processo sequencial de contagem (ida e volta), capaz de controlar melhor as peculiaridades de nossa *matrixset*, e evitar possíveis repetições (ou sequências) oriundas de processos aleatórios.

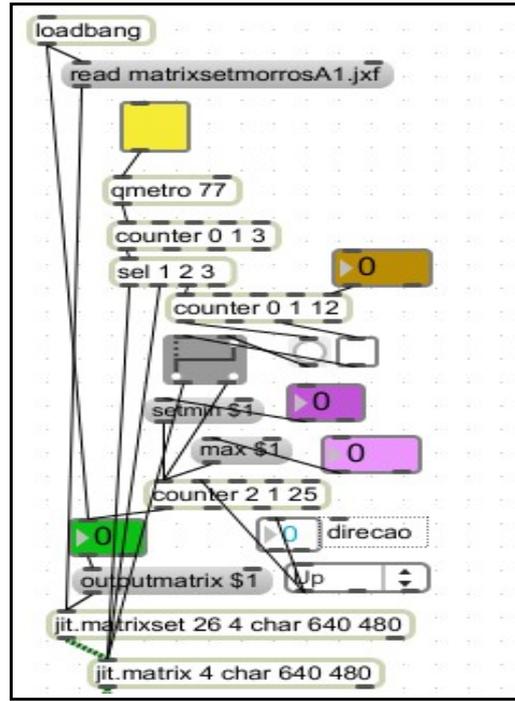
11 Os nomes dos objetos Max estarão sempre grafados em *itálico*.

12 *Bang* é uma espécie de ordem de “cumpra-se”, disparado ao longo da cadeia de processamento, e que exige uma execução segundo a função dos objetos envolvidos. É também parecido ao clássico comando computacional “ENTER”.

Exemplo 1



Exemplo 2



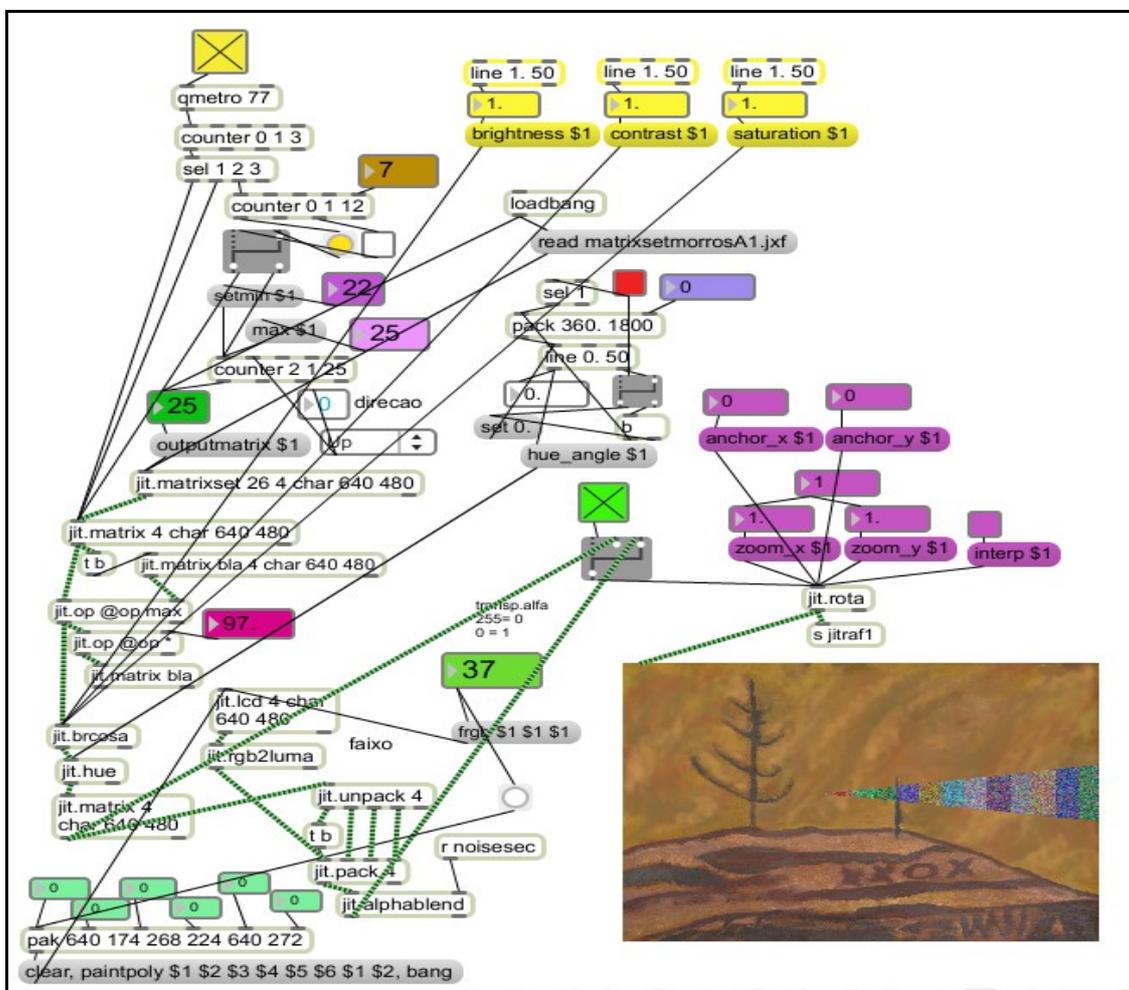
Dois mecanismos de projeção de imagens, com o melhor desempenho do segundo exemplo. Ainda poderíamos optar por outros modelos, como por exemplo construir pequenos vídeos e acioná-los através de um *jit.qt.movie*.

O segredo da eficiência do 2º exemplo sobre o 1º está no 3º *outlet* do objeto *counter 0 1 12*, que dispara um 'marcador' junto ao seu maior número (um *flag* que aciona um *switch*), direcionando-o desta forma à troca de matriz. Assim, mantemos um fluxo contínuo de imagens, controlado por um só *qmetro*. A velocidade de troca de matrizes fica restrita a uma faixa já testada, indo do máximo de 3 vezes do valor do *qmetro* ($77\text{ms} \times 3 = 231\text{ ms}$), à variação mínima definida a partir de $3N$. (231×2 ; 231×3 ; $231 \times 4 \dots$) . A velocidade de projeção que utilizamos, *qmetro* a 77, nos oferece um fluxo regular de 12,99 quadros por segundo.

Os mecanismos que controlam a parte A, bem como a parte C, podem ser descritos em sua ordem de ocorrência, iniciados no algoritmo de projeção de imagens e troca de matrizes acima ilustrado:

- 1 – Alimentação e troca de matrizes (*qmetro*, *counter* e *jit.matrixset*)
- 2 – Multiplicação e *feedback* (*jit.matrix "bla"* e *jit. Op : max* e *)
- 3 – Brilho, saturação e contraste; rotação de matiz de cores (*jit.brcosa*); (*jit.hue*)

- 4 – Mixagem por canal alfa (*jit.alphablend*, *jit.rgb2luma* e *jit.lcd*)
- 5 – Zoom e deslocamento de câmera (*jit.rota*)
- 6 – Mixagem e varreduras de cor (*Jit.Traffic*) - [em um *patch* específico .]

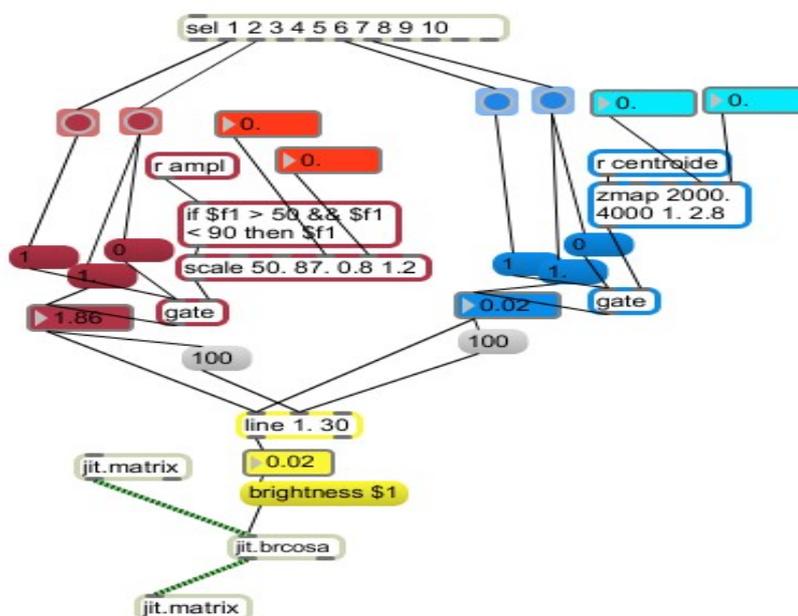


Sequência completa da cadeia de eventos de produção e processamento visual, desencadeada ao longo da primeira parte (controle de imagem A – seções 1 a 25).

A multiplicação de matrizes com realimentação (*feedback*), o primeiro efeito a atuar sobre as imagens produzidas pelo mecanismo de projeção, é um recurso usado para fundir as matrizes, em nosso caso as pequenas variações na textura do céu. Funciona (aqui, especificamente) em uma faixa de atuação efetiva (multiplicação junto ao *jit.op@op**) que vai de 0.90 a 1.00 , e possui uma analogia com o som *legato* (com seu coeficiente alto, acima de 0.98), e o *detaché*, (abaixo de 0.95), onde, assim como no som, as imagens se apresentam ou mais ligadas

ou mais articuladas.

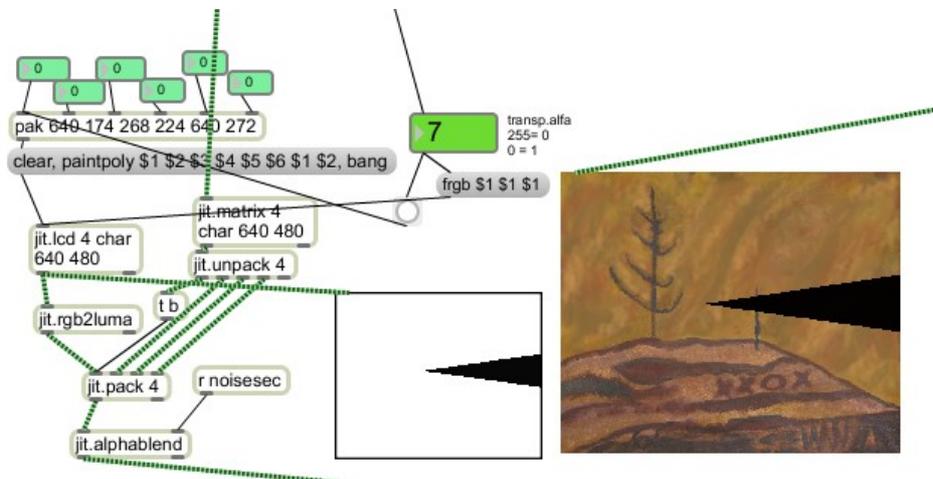
A etapa seguinte é constituída por objetos simples, *jit.brcosa*, responsável por alterações de brilho, contraste e saturação, e o *jit.hue*, que executa uma rotação no eixo de matiz de cor, preservando os valores de luminância . Eles podem ser acionados automaticamente, ou se conectar aos mecanismos de interatividade com os sinais de áudio (do *fiddle~* ,e variações do *Centroid~*), ou ainda ser manipulados pelas manetes *wiimote*, como nas últimas seções (parte C). Uma relação primária de causa e efeito musico-visual, por exemplo, é obtida através do resultado direto entre amplitude sonora e amplitude luminosa (muito som = mais luz ; pouco som = pouca luz).



Dois mecanismos de interação áudio/imagem. À esquerda,o sinal de recebimento(*r ampl*) do valor *RMS* (do 4ºoutlet do *fiddle~*) é facilmente convertido na escala apropriada (objetos *scale* ou *zmap*). À direita está o recebimento e conversão dos valores do *centroide*. Atuam diretamente (atenuadas por linhas de transição de 100 milisegundos) no *jit.brcosa*, informando seu valor para brilho (*brightness*), junto ao fluxo de processamento das matrizes.

A mixagem da imagem do “facho”, o 4º mecanismo, em seu formato e posicionamento triangular e específico, representou uma dificuldade encontrada ainda no início deste trabalho. Ao fugir do formato original retangular de uma tela,

tornou-se necessário configurar seu posicionamento e formato em uma mensagem de *“paintpoly”* junto a um objeto *jit.lcd*, (reconfigurado por um objeto *jit.rg2luma*) para finalmente servir de máscara para a mixagem com ruídos brancos (*r noisesec*) transmitidos pela seção B (*Patcher contrimag B*) ao *jit.alphablend*.

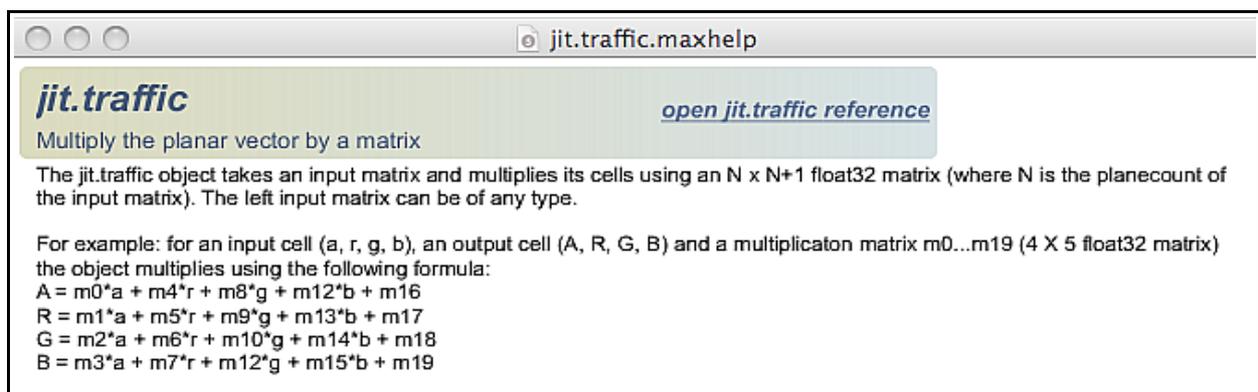


Mensagem de *paintpoly* direcionada ao *jit.lcd 4 char*, e demais sequências de formato e definição peculiar do facho. Podemos regulá-lo livremente, quanto a seu formato e posição, e também quanto a seu valor de transparência.

Os próximos efeitos, *zoom* e deslocamentos de *“anchor points”* (pontos de referência de enquadramento) são ações possíveis de serem realizadas por outros objetos, e mesmo através de mensagens diretas às matrizes (localização da imagem de origem, *“source dimension start/end”*, ou localização de saída *“destination dimension start/end”*, etc). Porém o objeto *jit.rota* executa mais satisfatoriamente tal função, com a opção de interpolação específica para seus *zooms* e deslocamentos. Quando no lugar certo e na hora certa, a interpolação se apresenta como um recurso extraordinário, capaz de camuflar as exageradas demarcações entre os *pixels*, em uma imagem severamente ampliada, por exemplo. Contudo, deve ser usada com parcimônia, seja porque podem *“embaçar”* desnecessariamente uma imagem, seja porque exigem sua cota razoável do uso de nosso processador.

A última etapa de todas as seções (A, B e C) ocorrerá em um *patch* específico, chamado *“Mixatraf”*. Nele, um objeto *jit.traffic* realiza suas operações numa função semelhante a desempenhada por um *mixer* de cores, operado por

uma sequência de 32 potênciômetros (*multislider*), divididos em 8 conjuntos de 4 multiplicadores ARGB (Alfa, Red, Green, Blue), cada conjunto agindo e influenciando o resultado dos demais por uma ampla faixa (de -5.0 a +5.0), tornando os resultados imprevisíveis e “varrendo” a imagem em todas as inúmeras diferenças de suas variáveis. Imperceptíveis marcas de manipulação digital das imagens durante o processamento junto ao GIMP (pág. 39), são facilmente detectáveis em algumas combinações do *jit.traffic*. A explicação de seus cálculos e funcionamento, ao contrário de sua utilização empírica, é um pouco mais complicada:



Todos os objetos max, msp e jitter possuem explicações e exemplos, além de um referencial mais completo (*Maxhelp* e *documentation:reference*)

Na concepção de nosso *patch Mixatraf* (na pág. seguinte), um *preset* é configurado para armazenar os arranjos (*settings*) de cada uma das seções da peça, além do tempo desta seção, que é utilizado para definir a transição (junto a um objeto *line*) rumo a cada nova combinação (a movimentação pode ser acompanhada no *multislider* inferior). Isto proporciona que cada uma das 41 seções da peça disponha de uma configuração “colorística” própria, e que possa se mixar à próxima em um tempo estipulado, ou realizar um *fade out*, ao fim de uma seção, por exemplo, ao receber uma mensagem 0. 0. 0...0 / 500. (todos os *sliders* em 0., em 500 milissegundos).

2.2 Contando seções e acionando *presets*.

O “sistema nervoso central” de nosso projeto talvez decepcione um pouco pela sua simplicidade. Mas observemos também que ele é o resultado de um princípio que busca oferecer a eficiência e praticidade de um sistema “ao vivo”, e o máximo de conforto e precisão, a um intérprete musical.

A limitação aparente de um sistema repartido em seções é a linearidade de sua narrativa e isolamento de suas partes. Esta é, realmente, a opção aqui escolhida, embora não precisasse necessariamente ser adotada. Ela, no entanto, torna possível controlar melhor não só a narrativa e o contraponto entre som e imagens, o que não desprezamos, como permite gerenciar também todo o sistema - ligar e desligar mecanismos específicos, por exemplo, poupando cálculos ao nosso já sobrecarregado processador “*dual core*”. Mas aqui não se trata de equipamentos. É que na verdade desconheço outra forma que pudesse se enquadrar neste projeto (e que de fato me convencesse a adotá-lo), que não a linearidade e organização rígida de suas seções.

Este centro de controle, então, nada mais é do que um contador de seções (dois, exatamente, um para indicar o início, e o outro para indicar o fim destas seções). São acionados principalmente por meio de um pedal (Midi *sustain*, do teclado controlador – *ctlin* 64). O sinal de áudio produzido pelo violino, através da saída de amplitude *RMS* do *fiddle*~ (trabalhado por alguns objetos, como o *thresh*~ e o *edge*~) também pode acionar estes contadores, embora optamos quase sempre pela utilização do pedal, mais confortável, do ponto de vista gestual, e capaz de responder frações de segundo antes do início do ataque do som, o que é importante ao gravarmos estes trechos musicais no Live, durante a maior parte das seções, com precisão no ataque e no término das frases.

Ao longo destas contagens, o sistema se encarrega de configurar a si próprio, abrindo e fechando interruptores (*switchs*) que receberão as próximas mensagens.

Há ainda um pequeno inconveniente, observado antes da conversão analógica/digital. Por ser um sensor elétrico de contato simples (passível, portanto,

de seus contatos “ricochetearem”), pedais (ou dispositivos de contato simples) podem disparar mais de um *bang*, ao serem acionados (embora isto não seja muito frequente). Para evitar esta redundância, filtramos este possível excesso de disparos utilizando um objeto chamado *onebang*, que só autorizará novo disparo 700 milissegundos depois (pelo valor definido no objeto *delay*, a ele relacionado, ou através de um *bang* diretamente em seu 2º *inlet*). (Exemplo na pág. 67)

Grupos de *onebangs* são acionados também diretamente pelo *follow* (um objeto Max “seguidor” de notas Midi), cuja seleção prévia está definida para a 23ª, 53ª, e 77ª notas de sua sequência pré-gravada (ao longo das seções 26, 27 e 28). Quando tocadas pelo tecladista abrirá janelas para que um *bang* específico possa ser disparado pelo pedal (que ao longo do restante da seção poderá ser utilizado livremente apenas como pedal *sustain* de um teclado temperado).

Todos os *patches* deste projeto se articulam a partir desta contagem, de início e fim de cada seção, e a partir daí executam suas ações específicas, sejam de caráter imediato, ou produzidos ao longo do tempo (através de objetos *line* ou *delay*, por exemplo).

Ao dispararmos a mensagem inicial de “ligar” (*r ligar*), os comutadores do recebimento de informações pelo *fiddle~* são abertos, preparando o sistema para contagem do início e fim da primeira seção, exercida por dois objetos “*counter 0 1 41*”, de cor azul para indicar os objetos vinculados ao início das seções, e em cor lilás para indicar a contagem de fim de seções. Objetos de seleção acusam a ocorrência dos eventos e preparam os próximos. Ao disparar o fim da seção 1, por exemplo, o *fiddle~* naturalmente interrompe seus próprios comutadores e libera os que controlam o acionamento do pedal *sustain (ctlin 64 1)*, e seu contador auxiliar, em verde, localizado na parte de cima do *patch* ilustrado a seguir.

Estes contadores auxiliares, em verde (há ainda um outro, destinado a direcionar o botão *home* do *wii* branco), de “1 a 2”, se destinam a direcionar os comandos para a correta destinação: 1= início da seção ; 2= fim da seção.

Temos ainda um contador de tempo, informando a duração de cada seção, e conexões que organizam e articulam objetos simples como seletores, interruptores e objetos de comunicação (*send* e *receive*, abreviados em *s* e *r*).

Ainda neste mesmo *patch* (*Sensing*) as sequências de contagem das seções disparam duas listas com dados a serem transformados em notas Midi pelo objeto *noteout* do Max (transmitidos via porta “c” *from max 1*). Estas informações (notas e controles Midi) são usadas como protocolo de comunicação, previamente associadas (e armazenadas) às funções principais de comando (em tela) do Live, responsável pelo funcionamento das gravações e reprodução dos clips (*loopings*).

Panel Live

canais	1	2	3	4	5	6	7	8	9Aux	10master
selec Audio	10	20	30	40	50	60	70	80	90	100
stop	17	27	37	47	57	67	77	87		107
arm bt	18	28	38	48	58	68	78	88		
rec/play						tocar todos				
1 slot	11	21	31	41	51	61	71	81		101
2 slot	12	22	32	42	52	62	72	82		102
3 slot	13	23	33	43	53	63	73	83		103
4 slot	14	24	34	44	54	64	74	84		104
5 slot	15	25	35	45	55	65	75	85		105
6 slot	16	26	36	46	56	66	76	86		106

Secoes, nos respect. slots e canais.

	1	2	3	4	5	6	7	8	9	10
1	22	5	4	3	2	26	27	40	x	x
2	18	6	7	8	9	28	29	-	x	x
3	6	15	14	13	12	30	31	-	x	x
4	21	16	17	18	19	32	33	-	x	x
5	2	22	21	10	20	34	35	-	x	x
6	-	-	-	-	-	-	-	-	x	x

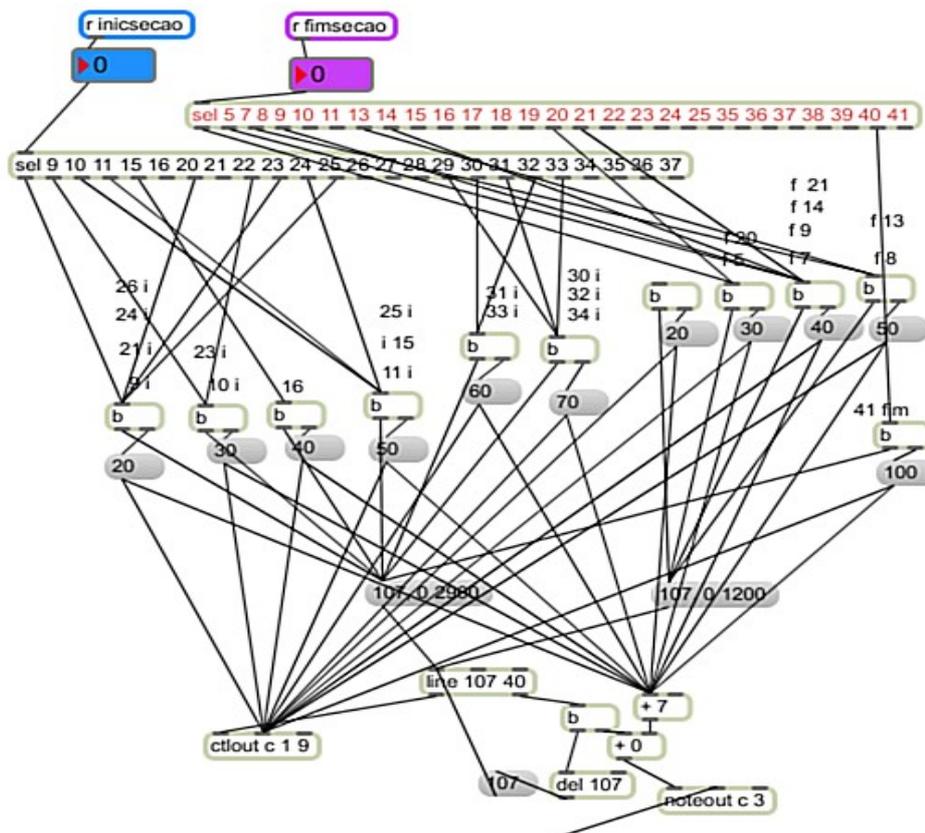
Listas de notas Midi (em cima) a serem transmitidas ao Live, selecionadas a partir das contagens de início e fim de seções. Os comandos (notas Midi) relacionados aos *slots* de gravação, *stop* e *armbutton* podem ser simulados no quadro logo a seguir. Por último a disposição de gravação das seções, nos respectivos canais e *slots* (em baixo). A seção 2, por exemplo, ocupará o *slot* 15, junto ao primeiro canal de áudio, e o *slot* 51, junto ao quinto canal. Um trecho gravado em dois lugares diferentes (e em canais separados), para serem reproduzidos juntos.

Um pequeno exemplo de leitura, a ser feita através da lista de mensagens dos objetos *coll* (coleção), disparando notas Midi (pelo Max) que reconhecidas pelo Live irão acionar suas funções de comando (em sua tela principal de *session view*):

- fim da seção 1, disparar nota Midi 58 (que é a função de “*armbutton*” do canal 5 do live, que prepara este canal para ser gravado).
- Início da seção 2, disparar nota 15 e 51 (5º *slot* do canal 1 e início de gravação; 1º *slot* do canal 5 com respectivo início de gravação).
- Fim da seção 2, notas 17 (encerra a gravação do 1º canal -*slot* 15), 58 e 48 (58 desarma o *armbutton* do canal 5. Neste caso, encerra-se a gravação e imediatamente inicia-se a reprodução do 1º *slot*. A nota 48 aciona o *armbutton* do canal 4).
- Início da seção 3, disparar nota 41 (1º *slot* do canal 4 e início de gravação).

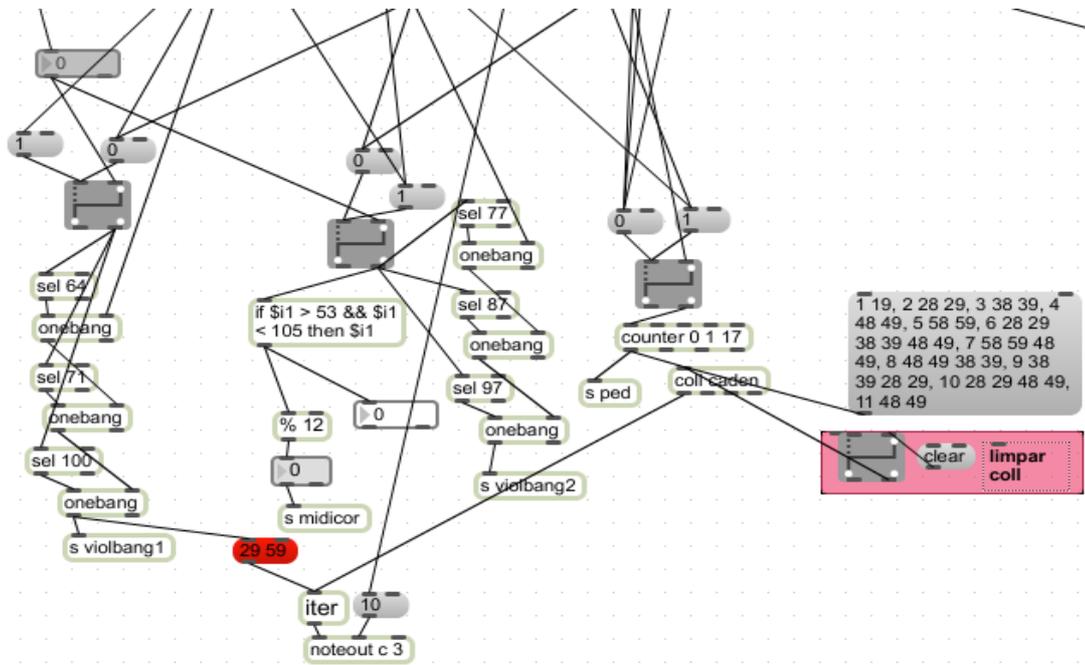
O Max, portanto, define a seção, e envia os comandos (pelo protocolo Midi) ao Live, que por sua vez realiza todo o processamento sonoro (gravação e reprodução de áudio). Na transposição da localização dos principais comandos de tela do Live, dispostos logo abaixo destas listas, temos os números das respectivas notas Midi vinculadas aos comandos, para uma melhor visualização, assim como a disposição de gravação das seções (na parte inferior do exemplo). À exceção das seções 1, 11, 23, 24, 25, 36, 37, 38 e 41, todas as demais serão gravadas em *slots* (as gavetas localizadas sob cada canal de áudio ou Midi do Live).

Adicionalmente a estes comandos de ligar e desligar os “*slots*” de gravação e reprodução, acrescentamos também um pequeno dispositivo para se criar um desligamento suave destas reproduções, criando primeiro um “*fade out*” em seu controle de volume (um objeto *line 107 40*) de 2,9 segundos para aqueles ocorridos no final das seções, e de 1,2 seg. para os que se situarem no início, e só depois a respectiva *noteout* Midi de *stop* será emitida (além de restabelecer o valor normal – 107 – ao controle de volume).



Há controladores (*cc*, ou “*control change*”) e “notas” Midi, indo em geral de 0 a 127. Ambos foram organizados e numerados de 10 em 10 (até 100) para indicar o respectivo canal (nota) ou o principal controle deste canal (*cc*), o potenciômetro de volume. O *cc* 20, por exemplo, altera o volume do 2º canal de áudio, enquanto uma nota Midi 27 (20 = seleção do 2º canal, + 7) aciona o botão de *stop* deste canal, disparado pelo *bang* emitido ao final da “linha” produzida pelo objeto *line 107 40*.

Contar e definir as seções torna possível acionar mecanismos de interação em momentos específicos. *Selectors* vinculados à seção 23, recebendo os sinais do 3º *outlet* do *fiddle~* (notas Midi), dispararão *bangs* ao terem suas notas específicas tocadas pelo violino. O Pedal *sustain* também abrirá um contador extra, no início da seção 11, a fim de disparar uma sequência de acionamentos e desligamentos de efeitos e *armbuttons* de canais do live. Por último, o *switch* aberto na seção 39, definindo as notas (padrão Midi) a serem tocadas (em sequência 77, 87 e 97) para o disparo do facho (*s violbang2*) com a mesma nota sendo enviada à nossa coleção (B) de cores (pág.76) para se mixarem ao facho (*s midicor*), conforme a disposição do próximo exemplo.



Três interruptores são abertos (mensagem 1) e fechados (0) nos devidos instantes (início ou fim de seções), e liberam os *onebangs* e contadores a partir dos comandos e seleção definidos.

A seguir uma tabela das seções abordando os comandos de contagem de seções, *presets* do *jit.traffic* (*traffic 0* = escuro total ; OR = imagem original) e sua transição em segundos, *zoom* (acima e abaixo de 1.0) e *anchor points* x e y (além do tempo do evento em segundos), número das matrizes utilizadas e a velocidade de troca destas matrizes (1 é a velocidade mais rápida), fator de multiplicação (*feedback*), e por último os mecanismos de interação com o áudio e outras observações.

Seus dados se referem à programação utilizada junto ao registro em vídeo da 1ª apresentação “teste”, anexado a esta pesquisa, e necessitam, portanto, de inúmeras correções (atualizações) e ajustes mais finos.¹³

13 Servem como uma planilha inicial para futuras alterações. Ver nota da pág.101 .

1ª parte: violino

Seção		Contag	Traffic (seg)	Zoom/anch x-y (segundos)	matriz	Vel 1/3n	Feed	Outros
0	ligar	Fiddle~0	0	2.48 / 532-192	21-25	8	1.	Início RMS(1.05/0.95 -brilho)
1	iníc	Fiddle~1	1 (0,5)					
	fim	Fiddle~0	0 (0,5)					
2	iníc	Pedal 1	2 (1,0)					
	fim	Pedal 1						
3	iníc	Pedal 1	3 (2,0)					
	fim	Pedal 1						
4	iníc	Pedal 1	4 (3,0)			6		
	fim	Pedal 1						
5	iníc	Pedal 1	5 (3,0)	2.48 a 1.00 (14,0)				
	fim	Pedal 1						Fim RMS
6	iníc	Pedal 1	6 (2,0)			3		Início Centroid 0.7a1.97 cont,sat.
	fim	Pedal 1						Fim Centroid
7	iníc	Pedal 1	7 (7,5)		10-20			
	fim	Pedal 1						
8	iníc	Pedal 1	8 (2,5)			2	0.96	
	fim	Pedal 1						
9	iníc	Pedal 1	9 (11,5)		0 -10		1.	
	fim	Pedal 1						
10	iníc	Pedal 1	10 (7,0)					Inic. RMS 1.07 a 0.95
	fim	Pedal 1				5		
11	iníc	Fiddle~1	11(60,0)			3		CADENZA - Pedal : de 1 a 11 efeitos Live(canal1 a 5). RMS: del 45s=0.95 a 1.07
	fim	Wii“vai”	0 (0,5)					RMS fim.

Seção		Contag	Traffic (seg)	Zoom/anch x-y (segundos)	matriz	Vel 1/3n	Feed	Outros
12	iníc	Pedal 1	12 (0,5)	2.71/150-0, 186 (9)				[Interp on]
	fim	Pedal 1	0 (0,5)					
13	iníc	Pedal 1	13 (0,5)	2.85/495, 6(7)-469				
	fim	Pedal 1	0 (0,5)					
14	iníc	Pedal 1	14 (0,5)	5.2, 2.37(7)/196-145				
	fim	Pedal 1						
15	iníc	Pedal 1	15 (1,5)	2.37 a 1. (7)				
	fim	Pedal 1						(interp off)
16	iníc	Pedal 1	16 (0,5)		0 -20	2		iníc.jithue.vel3970
	fim	Pedal 1						
17	iníc	Pedal 1	17 (0.5)					hue vel 1970
	fim	Pedal 1						
18	iníc	Pedal 1	18 (1,5)			1	0.96	hue vel 970
	fim	Pedal 1						
19	iníc	Pedal 1	19 (2,5)		10-20	2		Hue vel 3970
	fim	Pedal 1						fim do jithue
20	iníc	Pedal 1	20 (10))			3		
	fim	Pedal 1						
21	iníc	Pedal 1	21 (9)			5		
	fim	Pedal 1						
22	iníc	Pedal 1	22 (7)			6		
	fim	Pedal 1						Liga B(1)
23	iníc	Fiddle~1	23 (2,5)					Disparo facho: nota midi 100 facho a 180 Interaç mixerA / RMS
	fim	Wii-hom						Facho a 217
24	iníc	Fiddle~1	24 (7,9)					Facho a 107
	fim	Fiddle~0						Facho a 217
25	iníc	Fiddle~1	25 OR (7,9)					Zoom em 12 seg (5+7)
	fim	Autom.						Mixagem p B [del 3 seg e deslig Ae troca jitter] Liga B(2)

Facho 255= transparente; 0=opaco.

2ª parte: teclado

Seção	Contag	Traffic (seg)	Zoom/anch x-y(seg.)	Xfade 1e2 (seg)	Rote mod	Outros	
26	iníc	Pedal 1	OR (1,7)	1.,0.07 (19,7s) / 0 480	0.5 (0,5) - 0.7 (0,5)	4 fold	
	fim	Pedal 1					
27	iníc	Autom.	27(7)l	1. (19,7s)	0.5 - 0.9 (0,7)	3 clip	(iníc.com del 170)
	fim	Pedal 1					
28	iníc	Autom.	OR (7)	1. a 0.07 (19,7s) / 320 480	1. (0,5) - 0.9	2	(iníc.com del 170)
	fim	Pedal 1					
29	iníc	Pedal 1	29 (7)	1., 0.07 (15,7s) / 0 480	1. - 1. (0.7s)	0	
	fim	Pedal 1					
30	iníc	Pedal 1	OR (7)	1.,0.07 (9,7) 640/480			
	fim	Pedal 1					
31	iníc	Pedal 1	31	1., 0.07 (19,7) 320/240			
	fim	Pedal 1			- 0.7 (0,97)		Liga B(3) Liga Metros posic aleat.
Seção	Contag	Traffic (seg)	Posic.Aleat .	X fade 3		Outros	
32	iníc	Pedal 1	OR (7)	userstdim	0,97 (0,2)		Desliga B(2) (Xfade1=0,5) (Xfade2= 0 (0,2))
	fim	Pedal 1			0,79 (2,0)		
33	iníc	Pedal 1	33 (7)	usersrcdim	0,97 (0,2)		
	fim	Pedal 1			0,79 (2,0)		
34	iníc	Pedal 1	OR (7)		0,97 (0,2)		
	fim	Pedal 1			0,79 (2,0)		
35	iníc	Pedal 135 1,5	35 (7)		0,97 (0,2)		
	fim	Pedal 1			0,45 (2,0)		

3ª parte: Wiimotes

Seção		Contag	Traffic (seg)	Vel	Feed	Outros
36	iníc	Wii home	OR (4,9)			Rotação e esticamento em 13 seg. Deslig metro aleator.
	fim	Autom.	0 (0,5)			Jit.traffic: de 2 p/3 em 2,7seg. Mensag. brcosa 0.33
37	iníc	Autom	37	4	0.95	Deslig B (1 e 3) inicio c/ delay 970
	fim	Wii home				
38	iníc	Wii home	38	2	0.9	
	fim	Wii home				Liga B91)
39	iníc	Fiddle~	39	1	0.85	Cadenza 2 - (notas disp.77,97,87) Pedal efeitos – abre midicor- 97 (1,9s)_facho RMS= xfade 1
	fim	Wii home				Fim midicor e RMS (fach:207 mix 0.2 - 1,95)
40	iníc	Wii home	40	2	0.9	Facho 107 (2,7)
	fim	Wii home				Facho 197 (1,9)
41	iníc	Wii home	41	3	0.95	Deslig de facho e B com del 2011.
	fim	Wii home				

2.3 Um divertimento para teclado

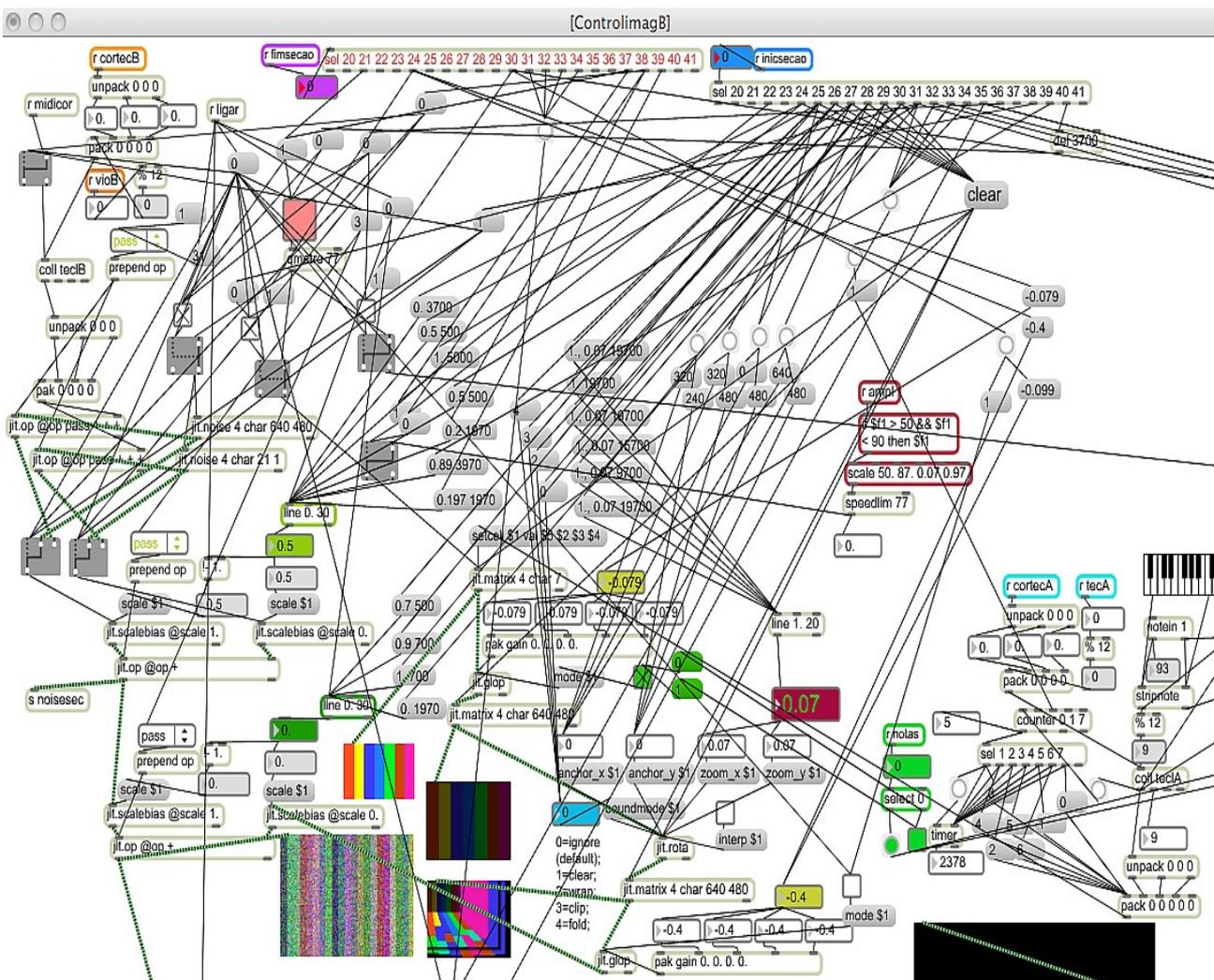
Pensado como um contraste (parte B), em um convencional formato A B A', a inserção de uma transição mais abstrata, originária de cores arbitrariamente relacionadas às notas de um teclado musical, acontece da seção 26 à 35, logo após o direcionamento da imagem ao interior do facho, produzido pelas matrizes de ruído branco (visual) neste mesmo *patch ControlimagB*. Essa ampliação exagerada, proporcionada pelo zoom da seção 25, faz com que a transição, das imagens do *patch ControlimagA* para as do *ControlimagB*, possa ser o mais suave possível. Efetuamos uma mixagem simples, seguida por troca de alimentação (permutação) para possibilitar a substituição e o desligamento dos mecanismos de imagens “A” (junto ao *patch mixatraf*, pág 64).

A origem dos processos envolvidos nesta parte B está na confecção de uma lista (que acabamos desdobrando em duas, caso se queira diferenciá-las, uma para o teclado junto às seções 26 a 35, e outra para o violino, junto à seção 39) que associe uma cor a cada uma das 12 notas em uma escala temperada de um teclado (sem referência ao posicionamento de 8^{as}). Isto pode ser indicado pelo intérprete, no *patch* principal. Uma vez definidas as cores de cada nota, elas passam a ser representadas graficamente, seja na forma de faixas verticais ou matizando matrizes, sempre que as respectivas teclas forem acionadas no teclado controlador Midi.



Lugar, junto ao *patch* principal *Morro P.(41)*, onde o intérprete pode escolher combinações de cores vinculadas às notas Midi, através de 2 listas TecA, para o teclado (seções 26 a 35) e VioB, para o violino (seção 39).

A primeira matriz sintetizada por nossas cores de notas terá apenas uma única linha e sete células (1 x 7, junto ao objeto *jit.matrix 4 char 7*), dispostas a partir do centro (células 3 4 2 5 1 6 0) e operados por objetos *counter*, *select* e *message setcell \$1 val \$5 \$2 \$3 \$4*, que reúne os valores de cores ARGB e informa a localização (de 0 a 6).



Parte do *patch ControlimagB*, responsável pelas matrizes de ruído branco e das faixas verticais coloridas produzidas pelo teclado. Embora de leitura difícil, optamos por demonstrá-lo aqui com todas as conexões, vinculadas diretamente às seções acionadas.

Atuando juntamente a dois objetos, *jit.glop* e *jit.rota*, podemos regular diferentes conjuntos para esta exposição de faixas coloridas, principalmente utilizando uma longa linha de tempo junto ao *zoom* do *jit.rota*, arrastando a imagem para o interior da tela e deixando impressa sua configuração anterior. O *jit.rota*, em

processos onde o *zoom* se projeta com coeficientes abaixo de 1.0 ,ou seja, a imagem original torna-se menor que o tamanho da tela, pode atuar de diferentes modos: *Ignore* , *Clear*, *Clip* e *Fold*.

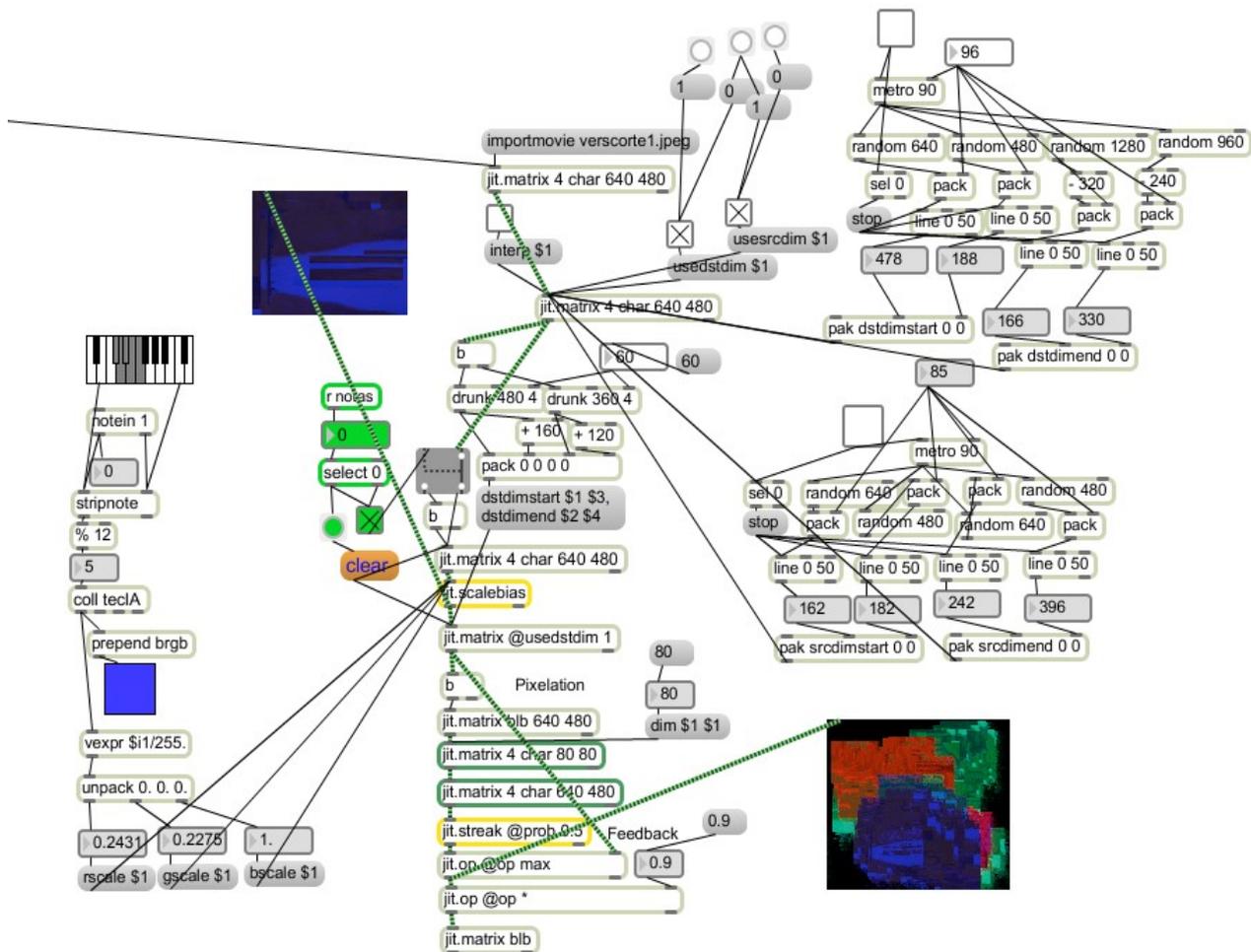
<code>boundmode</code>	<code>int</code>	Boundary handling mode (default = 0 (ignore)) 0 = ignore: Values that exceed the limits are ignored. 1 = clear: Values that exceed the limits are set to 0. 2 = wrap: Values that exceed the limits are wrapped around to the opposite limit with a modulo operation. (256 wraps to 0, 257 wraps to 1, and -1 wraps to 255, -2 wraps to 254, etc.) 3 = clip: Values are limited not to exceed min or max. (e.g. numbers greater than 255 are set to 255, and numbers less than 0 are set to 0.) 4 = fold: Values that exceed the limits are folded back in the opposite direction. (256 is folded back to 254, 257 is folded back to 253, and -1 is folded back to 1, -2 to 2, etc.)
------------------------	------------------	---

Referências Max, sobre os modos de “preenchimento” de tela, com o *zoom* situado abaixo de 1.

Diferentemente do *patch* anterior (*ControlimagA*), onde o fluxo de imagens corria sobre um percurso de caminhos e efeitos bem rígido, aqui dispomos de mecanismos independentes operados por mixagens simples, capazes, por exemplo, de fundir as duas matrizes de ruído branco (a primeira com dimensões de 640 x 480, e a segunda de 21 x 1, ou seja, 21 listas verticais) com a matriz inicial de 7 listas coloridas pelas notas do tecladista..

Ao efetuarmos trocas simples de matrizes por meio de comutadores podemos desligar o processamento anterior. Já as mixagens precisam ainda ser alimentadas por imagens junto a seu 1º *inlet*. Na troca ocorrida na seção 32 (início) uma parte dos mecanismos anteriores é desligada, mas mantivemos os responsáveis pelo fluxo da mixagem (os dois *jit.noise*). O novo mecanismo de nosso “divertimento” terá como origem, desta vez, uma matriz carregada com uma fotografia original (com o facho) do primeiro quadro. Em sua cadeia de processamentos, estão algoritmos simples de geração de combinações aleatórias para mensagens *dstdimstart*, *dstdimend*, *srcdimstart* e *srcdimend* responsáveis pelo posicionamento e dimensões das imagens recebidas pelo *jit.matrix*. Esse fluxo de imagens geradas por posicionamentos aleatórios (com a frequência de *bangs* de seus objetos *metro* a 77 milissegundos) é ainda matizado pelas cores associadas ao nosso teclado – *coll teclA* (através de um objeto *jit.scalebias*), e processado em *feedback* junto a um objeto de efeito visual chamado *jit.streak* (com

seu respectivo argumento, `@prob 0.5`). Processar em *feedback* significa nomear duas matrizes com um mesmo nome, e posicioná-las em locais distintos, junto ao início e fim de cada processo, por exemplo, de modo a se retroalimentarem. Antes de chegar ao *jit.streak*, e como parte deste processamento, as imagens são redimensionadas em uma matriz menor (*jit.matrix 4 char 80 80*) e por último multiplicadas pela taxa de *feedback* prescrita ao *jit.op @op**, onde conseguimos um efeito conhecido por “cauda de cometa”.

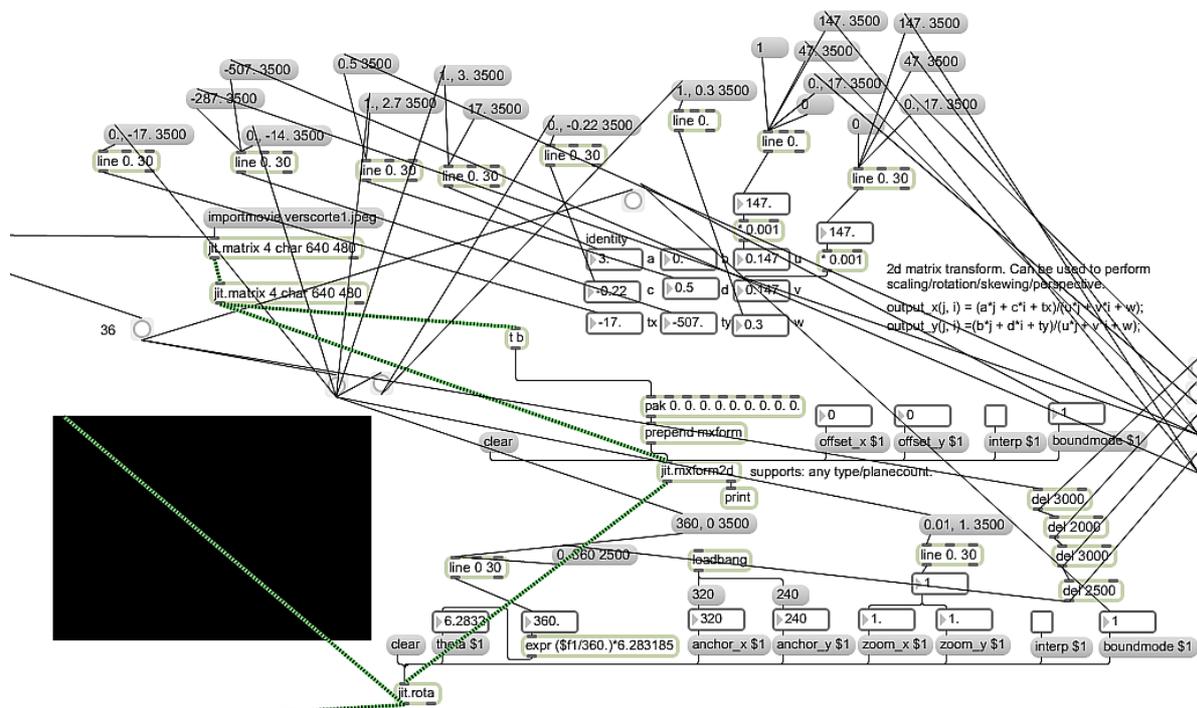


Parte do *patch ContrimagB* que define a matriz (original do quadro A) e seus efeitos, a partir de processos aleatórios (dimensões de destino e de fonte - *dstdim* e *srcdim* – das matrizes).

Tanto as operações com *feedback* como as de geração de posicionamento aleatórios utilizadas aqui são exemplos retirados de modelos descritos nos tutoriais do Max/jitter, sendo portanto mecanismos simples de regulagem e efeito sobre matrizes. Há uma categoria mais avançada, caracterizada por trabalhar com

projeções concebidas por desenho vetorial, através de uma linguagem padrão (*standart*) chamada de *Open GL* (*Open Graphics Library* – desenvolvida em 1992 pela *Silicon Graphics Inc.*), para a produção de desenhos gráficos computacionais em duas ou três dimensões. Embora o Max seja capaz de manusear os diversos objetos do grupo *GL* com bastante facilidade, e seus resultados sejam bem mais sofisticados, seu uso também deveria ser acompanhado de um criterioso estudo, algo pouco realista no espaço de tempo que dispomos para esta pesquisa. No entanto, o uso desta biblioteca visual de projeções de desenho gráfico é um dos inúmeros desdobramentos, a que este trabalho naturalmente nos encaminha.

A seção 36 marca o fim desta parte B, e funciona como transição principalmente das imagens, que executarão uma “rústica” movimentação através de um objeto *jit.rota* e do *jit.mxform2d*, responsáveis pelas transformações ocorridas com nossa matriz bidimensional (a mesma da seção anterior, com a imagem completa do primeiro quadro).



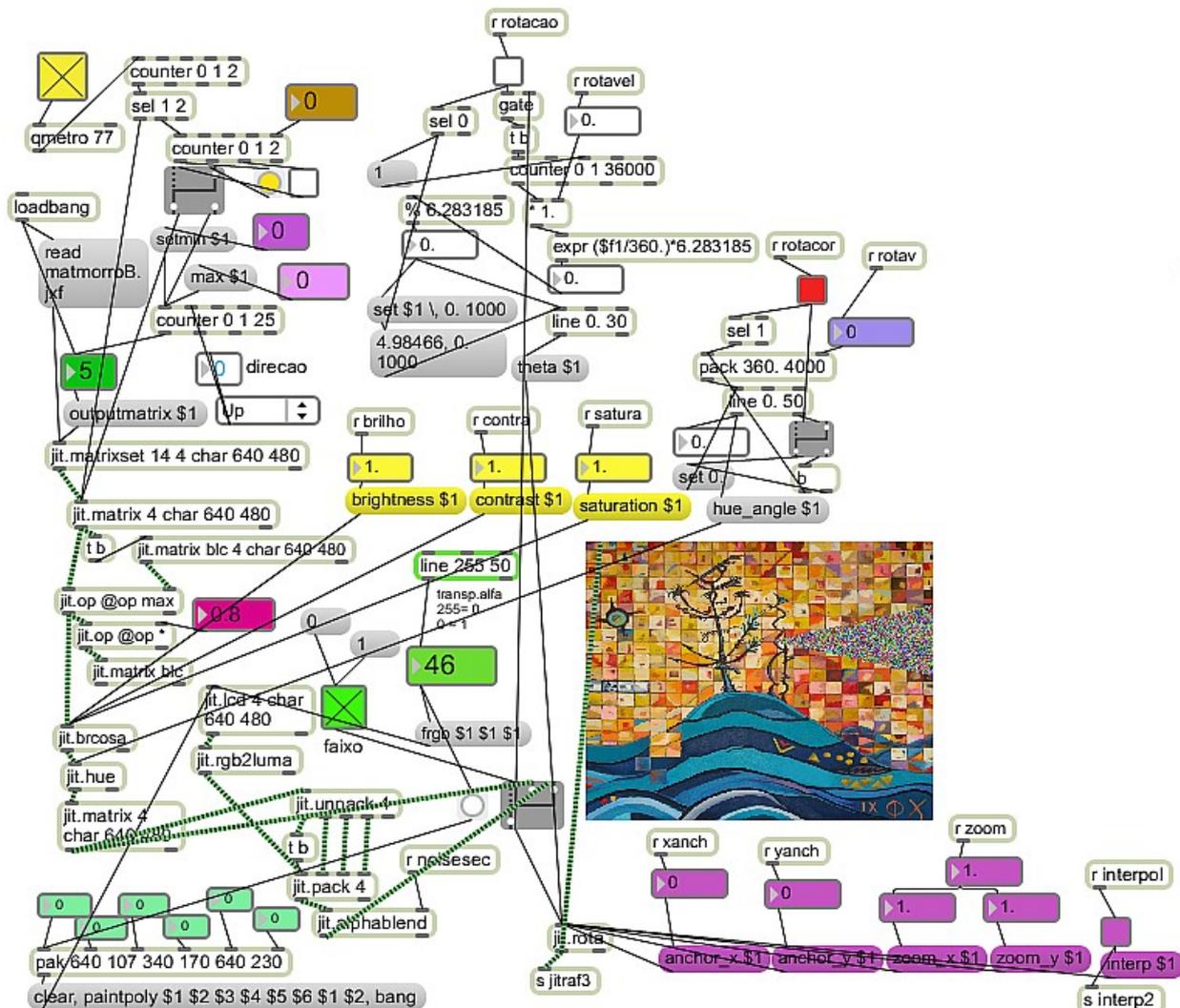
Vários objetos *delay* são usados para desencadear linhas (objetos *line*) que perfazem as movimentações e distorções ocorridas com a matriz original na seção 36.

2.4 Manipulando Processamentos

A partir da seção 36 (desencadeada por um comando do *wiimote* branco – a tecla *home*), depois de terem sido gravados cerca de 25 pequenos trechos de frases musicais, executadas inicialmente através de um violino (seções 1 a 25) e posteriormente através de um teclado (26 a 35), o intérprete é convidado então a manipular as duas manetes de videogames, *wiimotes* (um branco e outro preto), com alguns comandos (de vídeo) do Max, e (de áudio) do Live, vinculados a seus controles.

Não se trata, evidentemente, de uma atuação passiva, pois espera-se certo controle na coordenação de efeitos visuais e, principalmente, no fluxo sonoro, com uma administração idêntica à exercida por *Disq-Jóqueis (DJ)*. Um bom desempenho aqui certamente significará um bom desempenho musical, ou uma melhor qualidade sonora.

O sistema visual é praticamente o mesmo utilizado na primeira parte (A – seções 1 a 25), com pequenas adaptações. A primeira é no tempo mais rápido de troca de matrizes, que passa do máximo de 3 vezes do valor do *qmetro* a duas vezes este valor (objeto *counter 0 1 2*). Também incluímos um pequeno mecanismo de rotação de imagem (inexistente em A), junto ao nosso conhecido *jit.rota*. A programação prévia (por mensagens vinculadas aos seletores de seções) é bem menor, e vários objetos *receive* se conectam diretamente aos controles dos processamentos. A origem destas mensagens (objetos *send*) é o desfecho, talvez inevitável, deste protótipo de apresentação sonora/visual: a manipulação dos materiais expostos anteriormente e seus (melhores) processamentos possíveis. O foco de nossas atenções deverá ser, portanto, o mapeamento (*mapping*) construído a partir dos dados dos sensores de nossas manetes e dos seus algoritmos de controle. O *patch ContrimagC*, portanto, com a mesma estrutura apresentada no subcapítulo 2.1 (pág.57), ao invés de receber informações disparadas segundo as indicações de seções, conecta diretamente os valores produzidos pelas manetes *wiimotes* às mensagens de controle dos objetos *jitter*.

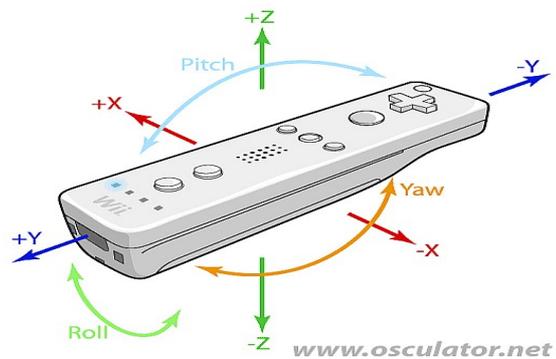


Estrutura com os mecanismos de processamento visual de “ControlimagC”. A velocidade máxima de trocas de matrizes é alterada para 144 milissegundos (2 x qmetro 77) . Vários objetos *receive* (r), conectados aos *wiimotes* comandam este processamento

A primeira regra a observar, junto às manetes do *wiimote* é a sua configuração anatômica já estabelecida. Na nossa interpretação, os botões 1 e 2 servem para configurar duas versões diferentes de todos os demais controles do aparelho, enquanto os botões A e B abrem interruptores de outras funções, relacionadas aos acelerômetros e seus indicadores de posicionamento.

O mapeamento inicial da recepção das duas manetes ocorre junto ao programa Osculator (pág. 37), que converte informações OSC em Midi cc ou Midi cc *Toggle*. Ambos transmitem valores de 0 a 127, e a diferença entre eles é no número de bangs decorrentes do acionamento de botões: Enquanto um Midi cc

informa duas mensagens, uma de 127, enquanto o botão é pressionado, e 0 quando ele é solto, um *Midi cc Toggle* precisará ser acionado duas vezes vezes (uma para ser ativado - 127, e outra para ser desativado - 0). Em ambas as manetes, os botões 1, 2, *home*, *minus* e *plus* informarão *Midi cc Toggles*, enquanto os demais, transmitirão *Midi cc*. Os sensores dos acelerômetros irão transmitir continuamente por *cc* os valores de posicionamento (entre 0 e 127) referentes ao eixo *roll* e *pitch*, de cada aparelho. Ou, como no caso do valor “*accel*” (*Midi cc 13*) o valor escalar sobre a soma de seus três planos (X, Y e Z) . O *wiimote* branco é transmitido pelo canal *Midi 5*, e o preto através do canal *7*, ao *Max*, que por sua vez transmite mensagens ao *Live* utilizando o canal *3* e *9*.



Message	Event Type	Value	Chan.
▼ /wii/1/accel/pry	-	-	-
0: pitch	MIDI CC	10	5
1: roll	MIDI CC	11	5
2: yaw	-	-	-
3: accel	MIDI CC	13	5
/wii/1/button/1	MIDI CC Toggle	14	5
/wii/1/button/2	MIDI CC Toggle	15	5
/wii/1/button/A	MIDI CC	16	5
/wii/1/button/B	MIDI CC	17	5
/wii/1/button/Down	MIDI CC	18	5
/wii/1/button/Home	MIDI CC Toggle	19	5
/wii/1/button/Left	MIDI CC	20	5
/wii/1/button/Minus	MIDI CC Toggle	21	5
/wii/1/button/Plus	MIDI CC Toggle	22	5
/wii/1/button/Right	MIDI CC	23	5
/wii/1/button/Up	MIDI CC	24	5
▼ /wii/2/accel/pry	-	-	-
0: pitch	MIDI CC	10	7
1: roll	MIDI CC	11	7
2: yaw	-	-	-
3: accel	MIDI CC	13	7
/wii/2/button/1	MIDI CC Toggle	14	7
/wii/2/button/2	MIDI CC Toggle	15	7
/wii/2/button/A	MIDI CC	16	7
/wii/2/button/B	MIDI CC	17	7
/wii/2/button/Down	MIDI CC	18	7
/wii/2/button/Home	MIDI CC Toggle	19	7
/wii/2/button/Left	MIDI CC	20	7
/wii/2/button/Minus	MIDI CC Toggle	21	7
/wii/2/button/Plus	MIDI CC Toggle	22	7
/wii/2/button/Right	MIDI CC	23	7
/wii/2/button/Up	MIDI CC	24	7

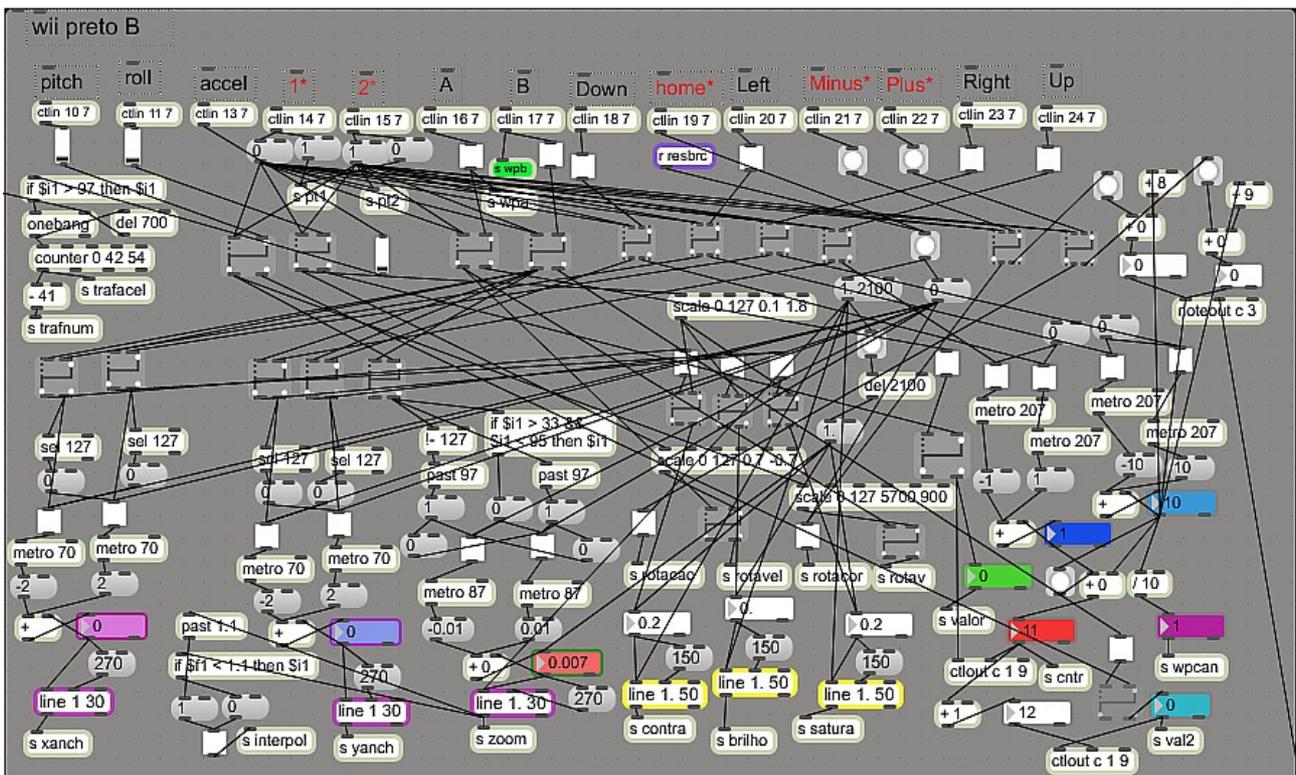
C...	Note/Control	Path	Name	Min	Max
9	CC 28	2-Audio Chamber...	Reson1 Tune	C-1	C5
9	CC 27	2-Audio Chamber...	Reson Mix	0.0 %	100 %
9	CC 26	2-Audio Chamber...	Pitch Mix	0.0 %	100 %
9	CC 25	2-Audio Chamber...	Pitch2	-36.0	12.0
9	CC 24	2-Audio Chamber...	Pitch1	-36.0	12.0
9	CC 23	2-Audio Mixer	X-Fade Assign		
9	CC 22	2-Audio Mixer	A-Reverb	-inf dB	0.0 dB
9	CC 21	2-Audio Mixer	Track Panning	50L	50R
9	CC 20	2-Audio Mixer	Track Volume	-inf dB	6.0 dB
9	CC 15	1 violino Guitar-D...	Reverb Level	-inf dB	6.0 dB
9	CC 14	1 violino Guitar-D...	Chorus Rate	0.00 ms	6.50 ms
9	CC 13	1 violino Mixer	X-Fade Assign		
9	CC 12	1 violino Mixer	A-Reverb	-inf dB	0.0 dB
9	CC 11	1 violino Mixer	Track Panning	50L	50R
9	CC 10	1 violino Mixer	Track Volume	-inf dB	6.0 dB
9	CC 6	Master	Song Tempo(fine)		
9	CC 5	Master Mixer	Song Tempo	20.00	999.00
9	CC 4	Clip	Loop Length		
9	CC 3	Clip	Loop Start		
9	CC 2	Clip	Volume		
9	CC 1	Clip	Transpose		
3	Note B6	Master	Stop Clips		
3	Note A#6	Master	Scene 6		
3	Note A6	Master	Scene 5		
3	Note G#6	Master	Scene 4		
3	Note G6	Master	Scene 3		
3	Note F#6	Master	Scene 2		
3	Note F6	Master	Scene 1		
3	Note E6	Master	Select Track		
3	Note D#6	A-Reverb Reverb	Device On		
3	Note E5	8 microfone Mixer	Arm		
3	Note D#5	8 microfone	Clip Stop		
3	Note D5	8 microfone	Slot 6		

Em cima o aparelho *wiimote*, em sua versão branca (de frente e detrás), seguido da ilustração de seus eixos e planos. Em baixo, à esquerda, o mapeamento ocorrido no *Osculator* (completo). No lado direito temos o mapeamento de recebimento de comandos *Midi* pelo *Live* (apenas uma parte dele).

Em sua 2ª configuração, nossos dois *wiimotes* selecionam os potenciômetros e demais controles de som da “mesa” do Live, porém é na 1ª disposição que distinguimos radicalmente as funções de cada manete: A branca irá selecionar os *slots* com gravações de *clips* e os acionar, enquanto a preta se encarregará exclusivamente dos controles de vídeo.

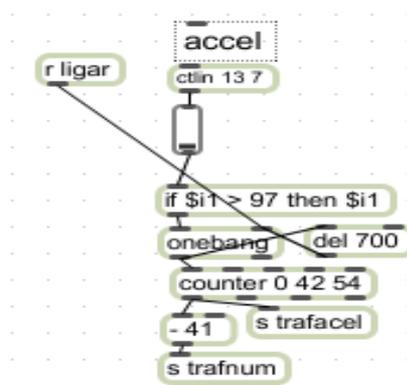
Observemos então as soluções alcançadas para cada caso, voltando ao *patch Sensing*, onde se localizam os algoritmos de conversões para os objetos Max/jitter da 3ª parte (*patch ControlimagC*), responsável pelo controle das imagens, e de transmissão de mensagens Midi, desta vez endereçadas ao Live para controle de suas funções.

O *Wiimote* preto:



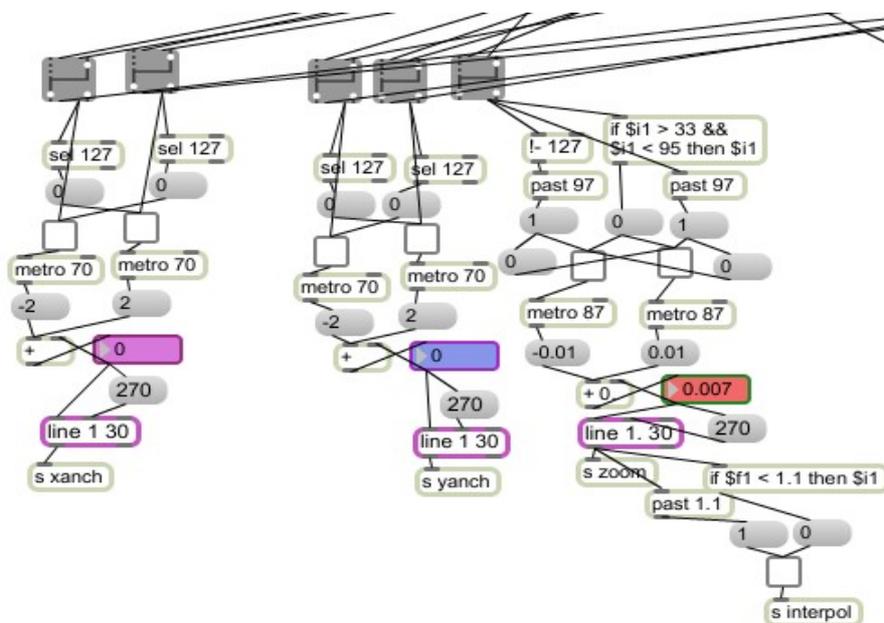
A programação envolvendo o *wiimote* preto começa com o recebimento pelos *Ctlin* (canal 7) das informações fornecidas pelo software Osculator, que mapeia o recebimento do sinal *bluetooth* (Osc) para informações Midi (0 a 127). A seguir, uma longa barreira de comutadores distribuem os sinais a partir de dois modos, selecionados pelas teclas 1 e 2.

Dois dispositivos estão isentos da seleção entre o modo 1 e 2 : o primeiro é a tecla *home*, que normaliza algumas funções do vídeo. O segundo está localizado no canto superior esquerdo, onde o “*ctlin 13 7*”, de aceleração global do *wii* preto, é conectado diretamente, sem passar pela barreira de *switchs* que direcionam os dois modos. É, porém, filtrado por uma expressão (objeto *if ...*) que pede que o impulso seja maior que 97, e um *onebang* (liberado por um *delay* de 700 miliseg, o que impede uma provável redundância). Esse bang resultante de um gesto brusco (maior que 97) acionará um contador conectado ao *preset* do *patch Mixatraf*, que por sua vez utilizará uma faixa adicional (de 42 a 54) destes *presets* para selecionar novos arranjos junto ao *jit.traffic*.



13 variações (de 42 a 54) junto ao *jit.traffic* podem ser acionadas por uma contundente “aceleração” junto à manete preta, independente da seleção de modo 1 ou 2.

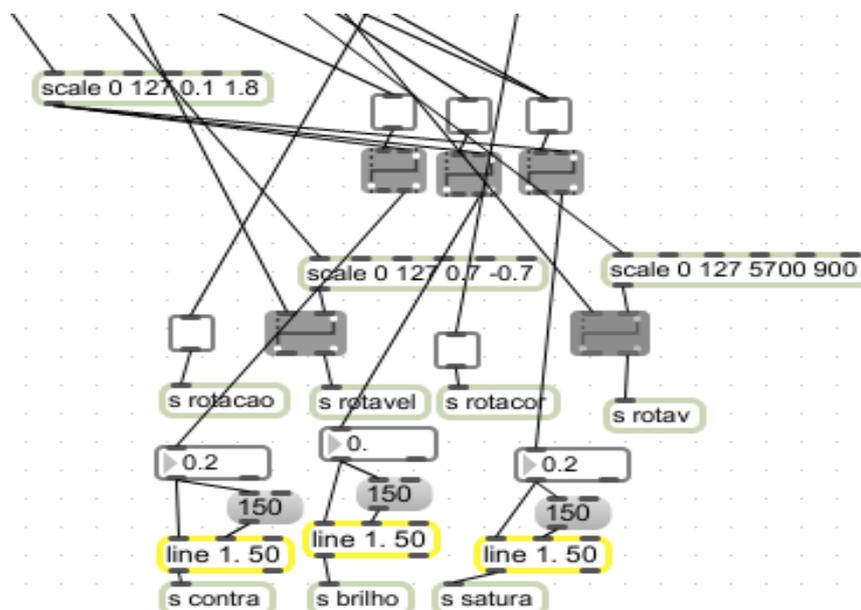
Os próximos mecanismos, no modo 1 da manete preta, designando os *anchor points* e *zoom* que serão transmitidos ao *jit.rota*, partem de acionamentos de *metros* que adicionam ou subtraem valores a partir de teclas *left* e *Right*, para o eixo X, e *Down* e *Up* para o eixo Y (eixos do plano bidimensional das imagens). O valor do *zoom* será definido pela rotação designada pelo eixo *pitch* do *wiimote* (*ctlin 10 7*), que também acionará *metros* a partir de seus extremos (maior que 97 ou seu inverso).



Enquadramentos (*anchor points* x e y), *zoom* e interpolação, a partir de contagens (adicionando ou subtraindo valores) efetuadas por *metros* (*metrônimos*).

Estes mecanismos contam ainda com um pequeno apêndice, destinado a informar ao *jit.rota* o momento certo da interpolação atuar - somente quando o *zoom* estiver acima de 1.1 . Deslocamentos e *zoom* precisam antes passar pelo acionamento de um interruptor (*switch*), aberto pela tecla B para que se distingua de outra interferência visual possível, exercida ainda na função 1 sempre que a tecla B não estiver atuando.

Essa intervenção ocorre ao pressionarmos diretamente as teclas *Down*, *Right* ou *left*, que sem serem direcionadas pela tecla B abrem, desta vez, novos interruptores, desta vez direcionando o controle *roll* (rotação de um dos eixos do *wii*, *ctlin 11 7*) para atuar diretamente nas respectivas funções de controle de brilho, contraste, e saturação.

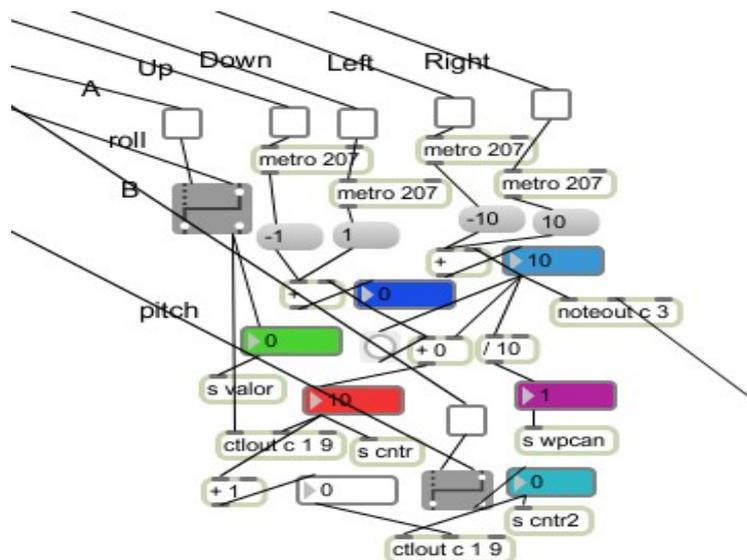


Conversões de dados Midi (0 a 127), para variáveis do *jit.brcosa* (de 0.1 a 1.8) a partir do eixo *roll*, velocidade de rotação do *jit.hue* – eixo *pitch* (de 5700 a 900), e de rotação de imagem, através do *jit.rota* -eixo *roll* (de 0.7 para movimentos à direita, a 0.7 para movimentos à esquerda).

Os botões *minus* e *plus* acionam diretamente dois mecanismos junto ao *jit.hue* e o *jit.rota*, a rotação de matizes de cor, e a rotação da imagem em si, dentro da tela. A velocidade de rotação também pode ser definida através dos valores informados pelo *roll* e *pitch* (reconfigurando sua escala, através dos objetos *scale*), e novamente liberando um *switch*, desta vez através do botão A. O *home* funciona como um normalizador geral, através de mensagens “1.” aos controles de zoom, brilho, contraste e saturação, e “0” aos de rotação de cor e demais metros.

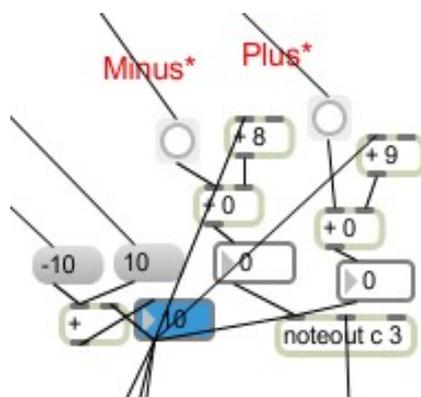
Ao acionarmos o botão “2” e seu respectivo segundo modo, de nosso *wiimote* preto, nos vemos novamente diante da transmissão de mensagens MIDI (principalmente os *cc - control change*) endereçadas ao Live, com os principais comandos ilustrados na pág. 92, para acionamento de efeitos e controle de mixagem de canais. *Left* e *Rigth* acionam notas de 10 a 100 (em saltos de 10), indicando, entre os 10 canais disponíveis no Live (6 de áudio, 2 para mensagens Midi do teclado controlador, 1 auxiliar e 1 *master*), qual será utilizado. *Up* e *Down*, por sua vez, adicionam uma contagem unitária, e definem, a partir da seleção do canal anterior, qual de seus controladores será aberto por um interruptor vinculado ao botão A, ou ainda o controlador seguinte (+1), aberto pelo botão B. Ambos

podem ser usados simultaneamente, e o primeiro passa a ser controlado pelo eixo *roll* (*ctlin11 7*), e o segundo pelo eixo *pitch* (*ctlin 10 7*), em ambos *wiimotes*.



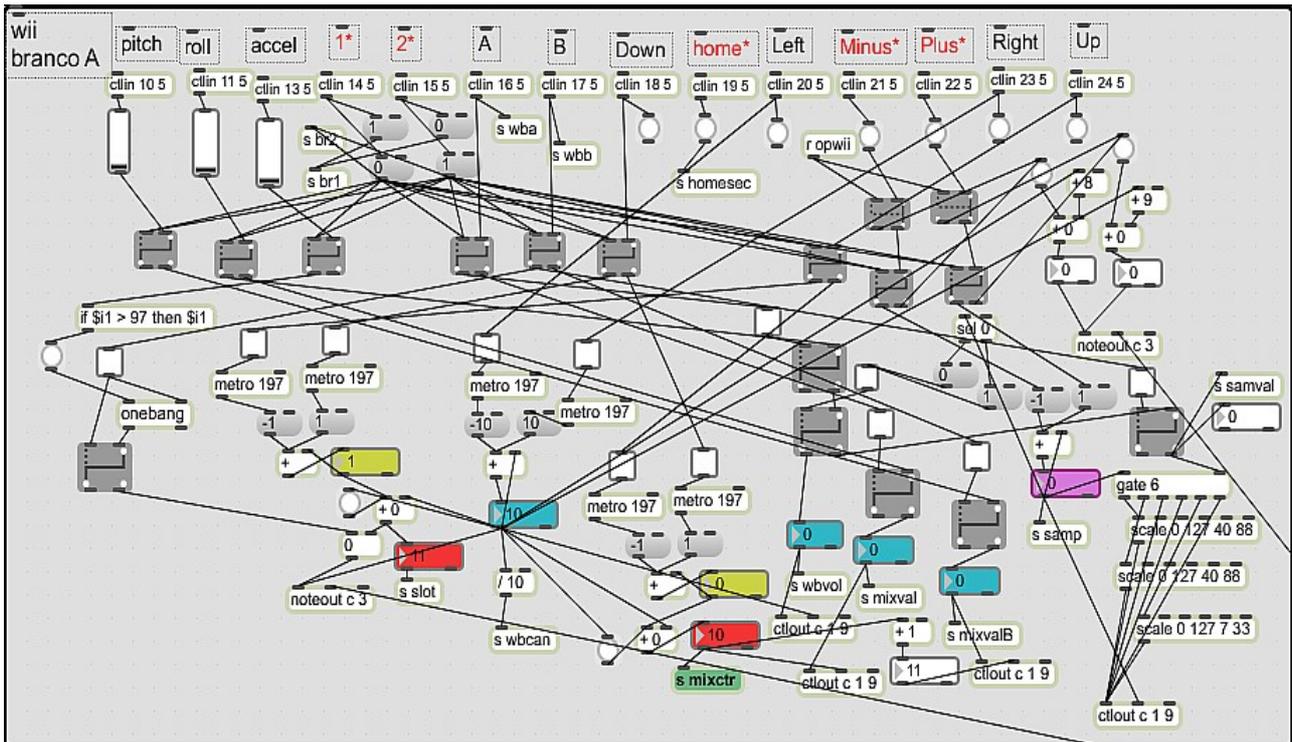
O 2º modo do *wii* preto define, assim como o 2º modo da manete branca, os controles da mesa de áudio do Live, volume, mixagem, etc., a partir do canal escolhido.

Notas Midi referentes aos comandos de *armbutton* e de ligar ou desligar os processadores de sinais de áudio (efeitos) para cada canal selecionado são acionados diretamente pelos botões *minus* (nºcanal + 8) para o *armbutton*, e *plus* para ligar e desligar os efeitos (nºcanal + 9).



No 2º modo, as teclas *minus* e *plus* do *wiimote* (branco e preto) acionam a respectiva tecla de *armbutton*, ou o comando de ligar e desligar dos efeitos de áudio ou Midi, sempre vinculados ao canal selecionado.

O Wiimote branco:

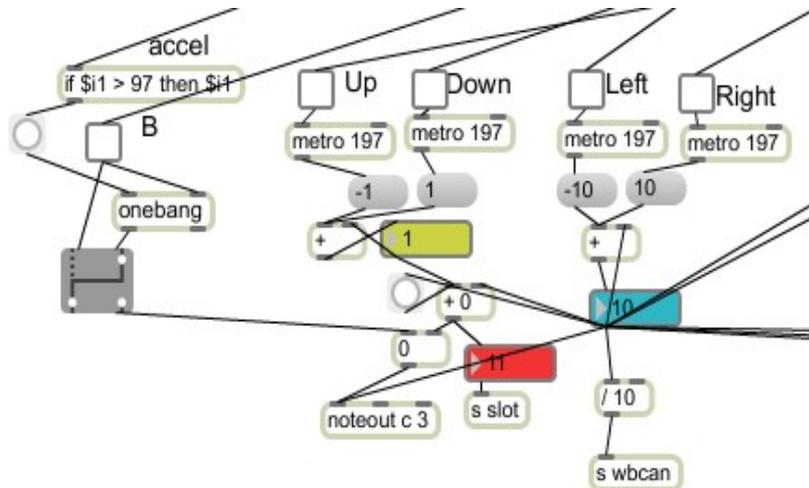


A entrada de dados Midi provenientes do Osculator ocorre agora pelo canal 5 (*ctlin N 5*). *Left* e *Right* sempre indicam o canal Live selecionado. Pode-se optar ou não pelo controle de amostras de áudio do *clip*, através de uma opção junto ao patch principal (pág.102).

A segunda configuração da manete branca é idêntica à segunda configuração da manete preta, o que permite que ambas atuem sobre a mesa de som (na tela) do live, cada qual agindo sobre um canal, ou ainda no mesmo, manipulando até quatro comandos, através dos eixos de *pitch* e *roll* (e teclas A e B) das duas manetes. Os comandos de seleção, *armbutton* e de acionamento dos efeitos de áudio continua o mesmo.

No entanto, a primeira configuração de nossa manete branca é peculiar. *Left*, *Right*, *up* e *Down* selecionam os *slots* que agora reproduzirão os *clips* já gravados. A ordem é a mesma da seleção dos *cc* e notas Midi : 10, 20, 30... indica o canal selecionado, enquanto 11, 12, 13.. indica a ordem, de cima para baixo, de seleção dos *slots* (as “gavetas” de arquivos de sons) de cada canal (neste caso o primeiro). Depois de selecioná-los, precisamos ainda acioná-los com um “golpe de aceleração maior que 97”, através da expressão “*if*” (objeto *max*) conectado às informações de aceleração global (*accel ctilin 13 5*). Além disto, devemos destravar

e liberar o *onebang* deste golpe com o botão B. Para cada golpe é preciso apertar (e segurar) novamente o botão B. Se apertarmos o botão A estaremos controlando diretamente o potenciômetro (*slider*) de volume, do canal que tivermos selecionado por último, através do eixo *roll*.



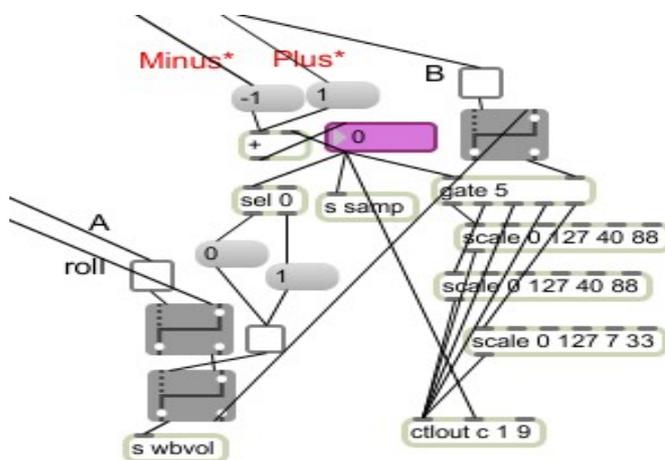
Mecanismo de seleção e reprodução de clips (*wii* branco modo1).

Os comandos *right* e *left*, de seleção de canal, estão isentos de definição de modo 1 ou 2, pois exercem sempre a mesma função. Além deles, o *home* desempenha função bem específica e importante: acionará os mecanismos de contagem de seções, a partir da seção 36 (poderá ser usado também no fim da 11 e 23).

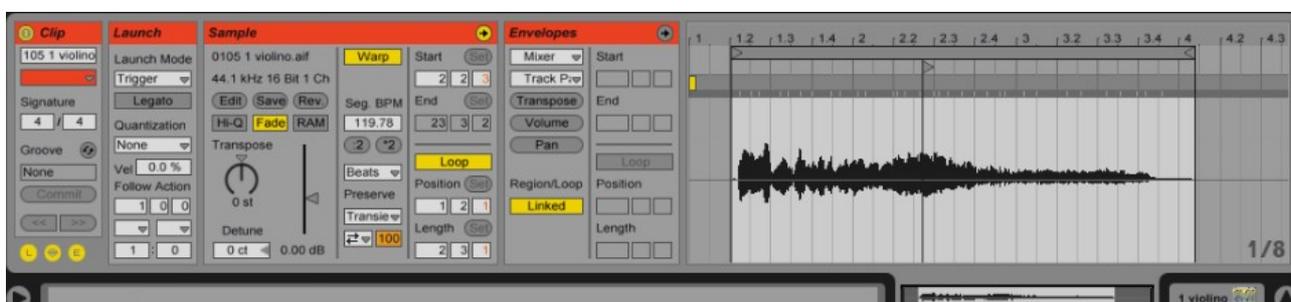
Por último os botões *minus* e *plus* poderão acionar (caso selecionado) uma categoria de controles que atuam diretamente na amostra de áudio do (último) clip, sendo eles:

- 1= transposição, em semitons;
- 2=volume de som da gravação;
- 3= início do loop;
- 4= fim do loop;
- 5 e 6 = tempo de reprodução geral (neste caso abrangendo todo o Live).

Estas alterações são manuseadas segundo as coordenadas *roll*, configuradas por alterações de escala ou não (por objetos *scale*), e vão necessitar de liberação dupla, dos botões A e B, para atuarem (o que fez com que precisássemos de mais um *switch*, a fim de não direcionar o *roll* para a função de volume).



Mecanismo de alteração direta nas amostras de áudio de cada clip, necessitando de liberação específica junto ao *patch* principal (pág.102), e acionamentos dos botões A e B do *wiimote* branco para agir.



Tela de visualização de cada *clip*, com características específicas e alternadamente à exposição dos processadores de áudio. O potenciômetro designado “*transpose*”, o *slider* ao lado (assinalando o volume sonoro) e o set “*position*” e “*length*”, em baixo da tecla Loop, podem ser comandados via *wiimote* branco (modo1).

Assim como os controles vinculados aos efeitos de áudio/Midi alocados em cada canal, cujas alterações são muitas vezes severas e difíceis de demarcar, as mudanças produzidas sobre os *clips* (amostras de áudio) são também de difícil visualização. Há sem dúvida o que melhorar, dentro desta programação, testar e

refinar melhor seu funcionamento, como é natural. Estes recursos adicionais, as alterações sobre as amostras de áudio (clips) e alguns controles de efeitos de áudio/midi, não precisarão, contudo, ser usados pelo intérprete.

A Segunda configuração de ambas as manetes terá, portanto, as seguintes funções de controle, dentro do Live:

cc - controle de mixagem e efeitos - canal 9										
cc volume	10	20	30	40	50	60	70	80	90	100
pan	11	21	31	41	51	61	71	81	91	101
send	12	22	32	42	52	62	72	82		
xfade-a/b	13	23	33	43	53	63	73	83	93	103
lig/des-efeitos	19	29	39	49	59	69	79	-	99	- (nota ch3)
1contr	14	24	34	44	54	64	74	---	94	---
2contr	15	25	35	45	55	65	75	---	---	---
3contr	---	26	36	46	56	66	76	---	---	---
4contr	---	27	37	47	57	67	77	---	---	---
5contr	---	28	---	---	58	68	78	---	---	---
6contr	---	29	---	---	59	69	79	---	---	---

A sequência horizontal é a da disposição de canais do Live (áudio, Midi, auxiliar e *master*), e a sequência vertical (de cima para baixo) é a de comandos em operação. Os quatro primeiros (10, 11, 12 e 13, por exemplo) estão disponíveis em todos os canais (exceto o auxiliar e *master* que não possuem *send*). A partir daí, ou seja, os seis restantes, irão manipular os controles relativos aos respectivos processadores de sinais de áudio (ao menos aqueles que nos interessam) colocados neste canal. A ativação dos efeitos de áudio ocorrerá por nota midi (e não *control change* – cc como os demais comandos).

1 áudio 2 áudio 3áudio 4áudio 5áudio 6 midi 7 midi 8áudio 9 Aux 10 master



Controle de mixagem do Live, onde no primeiro canal de áudio temos o cc

10 para o *slider* principal de volume, o cc 11 para o potenciômetro ao lado, informando o *Pan*, e o cc 12, que aciona o potenciômetro acima, abrindo o *send* ao nosso canal Aux (9). Já o cc 13 aciona a seleção do subgrupo A/B, em baixo, enquanto o cc 103, do *master*, à direita, controla a mixagem destes dois grupos.

	<i>Wiimote preto</i>	<i>Wiimote branco</i>
	<i>Acell</i> >97 = troca <i>jit.traffic</i> ; <i>Home</i> = normalizar imagens	<i>Left e Right</i> = seleção de canais Live (1 a10); <i>Home</i> = aciona a contagem das seções.
	Controle de imagens	Controle de áudio
Função 1	Tecla B: <i>zoom</i> (eixo <i>pitch</i>) posição: (<i>down, right, left, up</i>); Teclas Down (brilho), Left (contraste), right e up (saturação): valor (eixo <i>roll</i>); Tecla minus: rotação de matiz de cor (valor: tecla A +eixo <i>pitch</i>); Tecla plus: rotação da imagem (valor: tecla A + eixo <i>roll</i>)	Teclas Down e Up : selecionam,dentro do canal escolhido, um <i>clip</i> , em ordem crescente de cima para baixo (11,12,13...); Tecla A : Altera o volume (valor: <i>roll</i>) do respectivo canal Tecla B: Libera um <i>onebang</i> (do <i>accel</i> > 97) para acionamento e interrupção (<i>play/stop buttons</i>) de <i>clips</i> ;
		Opção 1*: Teclas minus e plus: <i>arbuton</i> e efeitos de áudio (o mesmo que na função 2)
		Opção 2*: Teclas minus e plus: percorrem (de 0 a 6) os controles sobre a amostra de áudio do último <i>clip</i> selecionado e velocidade global de execução do Live.(valor: teclas A + B e eixo <i>roll</i>)
	Controle de áudio	
Função 2	Teclas Left e Right : seleção de canais do Live (1 a 10), sobre o qual atuar. Teclas Down e up : seleção dos controladores disponíveis, de cima para baixo, dentro de cada canal do Live: 0 = volume, 1 = pan, 2 =send, 3 = mixer A/B; do 4 ao 9 temos os controladores (6 no máximo) dos efeitos de áudio, alocados no canal. Valor: teclaA+roll Tecla B abre o controlador seguinte (+1) e o valor é definido pelo eixo <i>pitch</i> . Tecla minus :liga/desliga <i>arbuton</i> do canal Tecla plus : liga/desliga os efeitos de áudio respectivos.	

(*A opção é feita junto ao *patch principal Morro P(41)*, última coluna (em cima) de *toggles* das informações do *wiimote branco*.)

Limitados a 6 efeitos de áudio, 4 efeitos Midi, e dois instrumentos (Midi), em razão de nossa versão reduzida do Live (Live Lite 8.2), o que também não faz muita diferença em razão de outra limitação, a da capacidade de processamento de nosso computador, utilizamos então os seguintes “*audio effects*” disponíveis,

divididos em dois tipos: no primeiro estão os *racks* (canais 1, 2 e 4), com seu conjunto de efeitos, sem, porém, maiores condições de editá-los. O segundo tipo são processadores de efeitos específicos (canais 3, 5 e 9Aux), e de maior possibilidade de edição. Logo abaixo estão os cc utilizados.

Canal 1 /áudio



cc14= Chorus rate;
cc15= Reverb Level.

Canal 2 /áudio



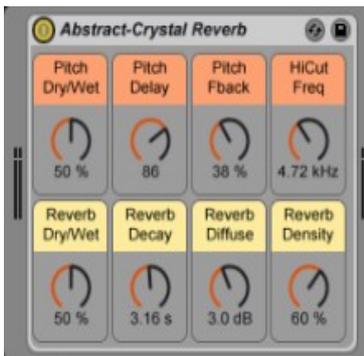
cc24= Pitch1; cc 25= Pitch2;
cc26= P.mix; cc27=R.mix;
cc 28=R1tune; cc29=R2tune.

Canal 3 /áudio



cc34= tempo; cc35= F;
cc 36= feedback;
cc37=Dry/wet

Canal 4 /áudio



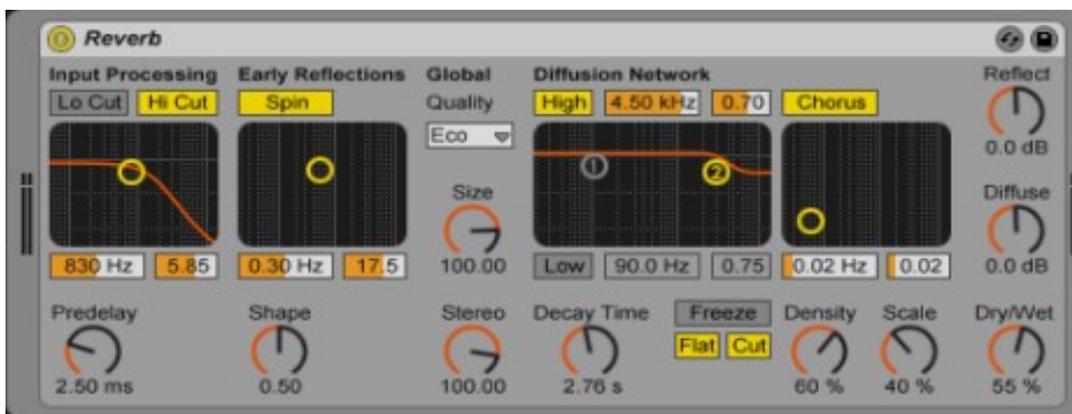
cc44=Pitch DW; cc45=P.Fback;
cc46=Rev.DW; cc 47=RevDecay

Canal 5 / áudio



cc 54 = spray; cc 5= Pitch; cc56= Rand Pitch ; cc57=feedb.
cc 58= Dry/wet, cc59= tempo.

Canal 9 Auxiliar



cc 44= dry/wet.

Os canais Midi, com sua sequência de processamentos (*signal chain*), comportam cada um dois efeitos e um instrumento. Enquanto os efeitos alteram apenas as notas e demais mensagens Midi, efetivamente são os instrumentos que os transformam (as notas e demais controles de nosso teclado controlador) em sinais de áudio para o respectivo canal.

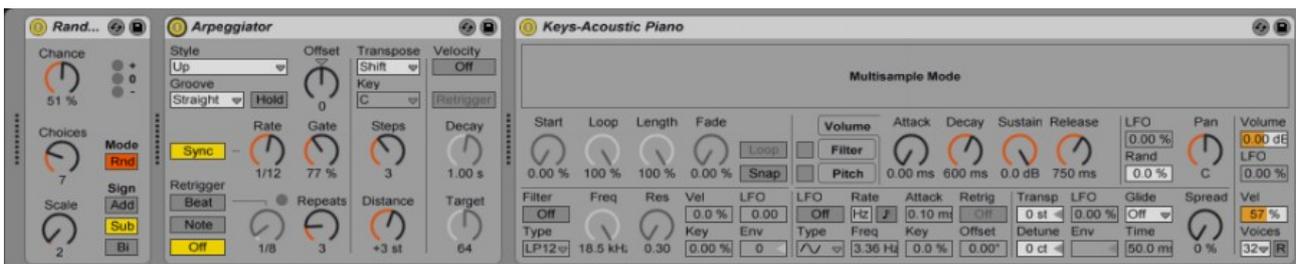
Canal 6 /midi



Random effect: cc 64=chance; cc65=sign mode;

Chord effect: cc 66= liga e desliga efeito; cc 67= *shift1*; cc 68=*shift2*; cc 69= *shift3*

Canal 7 /midi



Random effect: cc 74=chance; cc 75=sign mode;

Arpeggiator effect: cc 76= liga e desliga efeito; cc 77= *Rate*; cc 78=*Repeats*; cc 79= *Distance*.

Conclusão : Uma sinergia possível

Um trabalho em aberto

O uso de imagens em shows e apresentações musicais, popularizado pela revolução digital em curso, oferece um variado leque de utilizações, tais como a projeção de cenários ou a exibição de vídeos de qualquer natureza. Lidar com a complexidade das produções artísticas, multimídias ou não, (algo encarado às vezes com a excitação saudável da busca e do desafio, e às vezes com o desânimo e angústia causados pela mesma busca e desafio) é também saber lidar com a complexidade das ideias e da criatividade humana, algo de certa forma misterioso e imprevisível. Há cada vez mais tarefas específicas e especializadas, mas no entanto isto não impede que produções caseiras possam ser realizadas, e que qualquer um produza sua própria “arte digital”.

Além disso, um dos grandes contentamentos envolvidos nas criações artísticas é a percepção de autonomia criativa, de dominar um pequeno, mas orgânico, conjunto de fatores, bem diferente do mundo “exterior”, geralmente pouco flexível aos desejos pessoais. Computadores permitem, a um indivíduo, não só uma gerência e produção sonoro-visual, mas também a sua realização “ao vivo”. Não se trata nem de vangloriar ou propor o isolamento de artistas e pessoas, consequência de nossa sociedade e época. Apenas afirmamos, e este projeto o comprova, que não dependemos de muita coisa para nos aventurar em “arte digital”.

Utilizamos até aqui um sistema mínimo, dentro de um único “Laptop”. Ao recorrermos, no entanto, a mais de um computador, ligados por conexão de rede, (além, é claro, de somarmos a capacidade de processamento), poderemos utilizar também mais de um projetor de vídeo. Um palco com 2 ou 3 telões é potencialmente mais interessante que aquele com apenas 1. Projetores também podem ser utilizados como iluminação direta, e serem conduzidos e direcionados manualmente. Ou ainda em disposições variadas, com seus posicionamentos e conjuntos de imagens sendo produzidos em diferentes tipos de telas, utilizadas como suportes destas projeções, e que podem ser opacas ou semi-transparentes, por exemplo. Outro recurso de inquestionável sedução àqueles que se dedicam às imagens (e seu uso em sistemas interativos) são as câmeras digitais de vídeo.

Mesmo neste nosso reduzido sistema, estivemos próximos de utilizar a câmera instalada no próprio *notebook*.¹⁴

Dentre as apresentações artísticas envolvendo sistemas interativos que me chamaram a atenção, gostaria de citar rapidamente algumas. A primeira é “*Itinerário do Sal*”, espetáculo solo de Miguel Azguime (grupo *Misoensemble* de Portugal) apresentado em B.Horizonte (em 2010), com sofisticada instrumentação (2 ou 3 câmeras, 2 ou 3 projetores, 3 ou 4 computadores) , e que também utilizava o software Max (além de outros). Lembro-me sua maneira descontraída e alegre de abordar às minhas iniciantes inquietações: “*Como é meu gerenciamento de timing e paradigma de resposta? Simples, são cento e poucos bangs disparados por um pequeno sensor MIDI localizado próximo a meu pulso*”, respondeu aproximadamente, naquela época. (dispunha na verdade de um sistema amplo, com monitores de controle, interações com áudio/vídeo, etc).

A presença de mais intérpretes (ou performances) é a outra forma natural de expansão a que podemos recorrer. As apresentações do *Grupo Mesa de Luz*, por exemplo, (2010 – em B.Horizonte) ilustram este trabalho em equipe, com funções bem definidas: Um de seus integrantes (T. Seferin) executava pequenos trechos musicais que desencadeavam *loops* (utilizando o Live). Outro (H. do Vale) cuidava exclusivamente do processamento de vídeo e seu gerenciamento em software específico (Resolume). A terceira integrante, Marta Mencarini, executava performances visuais sobre uma mesa de vidro, captada por câmeras em vários ângulos, desfilando objetos e compondo uma narrativa cheia de delicadeza e poética. Sua experiência está retratada na dissertação “*Mesa de Luz: Colagem-Composição*”, defendida em 2010 junto a Universidade de Brasília (UnB).

Os últimos trabalhos com imagens e interatividade a serem considerados não os presenciei ainda, mas conheci-os pela TV, como no caso do espetáculo “*Olho*”, do grupo *Minik Mondó* (sesc TV - agosto de 2011). Utiliza várias combinações de captação e projeção, de maneira muito bem exploradas (os posicionamentos de câmera em relação aos dançarinos e às projeções), em uma bela sinergia com a dança contemporânea. De festivais como o Livecinema (RJ,

¹⁴ Testamos a câmera e alguns exemplos (praticamente prontos), compartilhados via WEB por comunidades ligadas ao Max/Jitter.

SP - 2011), (e sua divulgação pela internet) destacaria o Duo N-1 (G.Obici e A.Fenerich) em “Metaremix” e a obra “Conexões Dispersas, Dispersões Conexas”, de L. Campezzato, V. Kisil, J. Jaramilo e F. Iazzetta.

Com incrível rapidez, computadores, câmeras e projetores vem se tornando ferramentas comuns em espetáculos artísticos de qualquer natureza. Os exemplos acima representam experiências interativas bem sucedidas construídas a partir de aparatos tecnológicos relativamente simples, extraíndo o máximo de expressividade a partir de ferramentas e recursos tradicionais, como *loops* e processamentos de sinais de áudio e vídeo. São poderosos “efeitos”, subordinados à narrativa (ou qualquer outro objetivo) das linguagens artísticas envolvidas.

Se apresentamos nesta dissertação nosso “formato de sistema” junto a seus mecanismos de controle e interatividade, falta-nos comentar sobre o que ainda poderia ser feito para melhorá-lo. Ao utilizarmos o pedal *sustain* do teclado como controlador de contagem de seções, estamos reduzindo o conjunto de hardwares necessários, e restringindo a comunicação ao protocolo Midi. É uma simplificação útil até aqui, no entanto é preciso apontar que há interfaces mais interessantes, e de certa forma também disponíveis a iniciantes em eletrônica: a disposição personalizada de sensores e outros dispositivos a partir da montagem de placas (com circuitos integrados) “Arduino”¹⁵ ou similares.

O sistema que montamos (além da participação especial das peculiares imagens originárias de pinturas), possui, como é de se esperar, potencial musical (ou sonoro) ainda inexplorado. Não sei se deveria usar o termo criação ou montagem, de um sistema, mas o fato é que isto consumiu quase integralmente estes trabalhos, embora, agora, ao final deles, pareçam ser o “caminho direto e previsível” da utilização dos softwares escolhidos. Tudo parece “simples e previsível”, depois de ter sido feito. Contudo, nem mesmo os caminhos até aqui foram lineares. Dezenas de vezes versões diferentes foram feitas, refeitas e na maioria das vezes abandonadas.

De alcance e concepção talvez muito além da realidade desta pesquisa, a análise das características de contraponto e harmonia possíveis (ou praticados, ou apenas selecionados, posto que infinitos) junto a equipamentos e programas

15 (<http://arduino.cc>).

similares, (neste início de séc.XXI) partiria da necessidade de metodologias e nomenclaturas amplas, ora próximas ao universo das experiências eletroacústicas e acusmáticas, ora utilizando-se dos modelos e termos musicais tradicionais¹⁶. Nossa vocação (ou quem sabe de nossa época, uma vez que todo este instrumental é bem recente) não nos encaminhou à análise, mas sim à experimentação.

A ideia de harmonia e contraponto tradicionais sugere ainda um forte vínculo a uma pulsação comum, o que poderia ser obtido junto aos chamados “atuadores”¹⁷. Estes são pequenos transdutores capazes de enviar ao intérprete alguma mensagem física, como, por exemplo, informar a contagem de tempo por impulsos táteis. Embora não tenhamos percorrido este caminho (de uma pulsação definida), ela pode ser obtida, caso se queira, sobrepondo os *loops* de maneira sincronizada, como nas seções 12, 13 e 14, onde a primeira impõe sua pulsação às seguintes. Tratando livremente o fluxo temporal e melódico de cada *loop*, e criando massas sonoras mais ou menos estáticas, o que fizemos quase sempre, mesmo assim podemos distinguir pulsações individuais, bastando para isto que as frases musicais as contenham em si, e se articulem em torno delas.

É um erro encarar as notas musicais escritas na partitura da pág. 49 como fundações básicas e imutáveis sobre a qual se estrutura nossa peça experimental. São tão móveis e flutuantes como os demais mecanismos, e devem constantemente ser avaliados, movidos, testados e alterados. Também não devemos encarar a figura dos *clips* como sendo sempre uma célula mínima de construção musical. Podemos fazê-los longos, articulados e bastante complexos. Alguns *clips* em *loop*, assim como o enunciado em uma frase musical, facilmente desgastáveis em uma repetição persistente, são radicalmente transformados, ao receberem uma segunda parte contrastante (dentro do mesmo *clip*), o que faz dela, naturalmente, uma frase musical mais elaborada, como nas seções 5 ou 18.

Os resultados percebidos até aqui nos animam a almejar um futuro promissor, ao nosso até então protótipo de Sistema. Deixamos o estágio de estabilização e reparos de problemas para o de ajuste fino e gradual, e ainda não

16 “Contraponto de extratos sonoros”, utilizados por Debussy, por exemplo.

17 Atuadores (estes sensores “invertidos”) são particularmente possíveis (toda ordem deles) através de Arduinos.

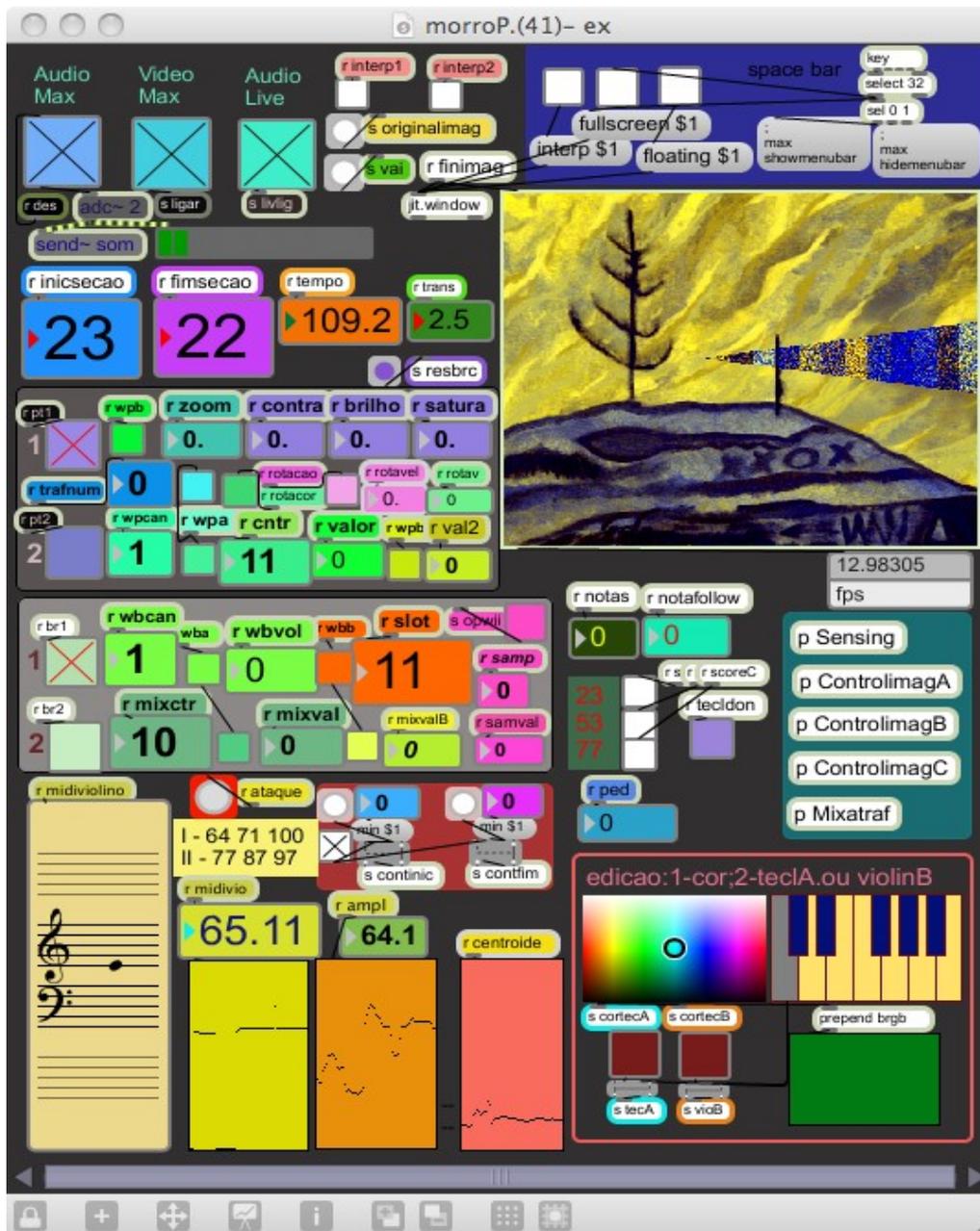
desistimos de experimentar ideias e alternativas¹⁸. Neste sentido, esta é ainda uma obra em aberto. Buscar esta sinergia, ou outros fatores expressivos quaisquer, responsáveis por fazer com que nos processos artísticos a soma de 2 + 2 dê pelo menos 5, parece uma tarefa sem fim. Neste longo caminho, que vai de “4 a 5”, também não costumam existir atalhos.

Por último, ressalto ainda um significativo aspecto, envolvido em nosso Sistema: ao estimular experiências interativas, o uso de imagens, processamentos diversos e toda ordem de recursos digitais, incentivamos consequentemente uma “disciplina intelectual” desvalorizada e atualmente pouco encorajadora: a composição musical.



Um músico, seu sistema pessoal e cenário-filme. Eu, em gravação-teste realizado em 23/09/2011, no Auditório da EMUFMG.

18 A versão anexada em DVD, sendo um primeiro teste deste sistema, e contando ainda com várias “improvisações” (*Cadenzas...*), deverá sofrer melhoramentos (em sua programação e composição) necessários e naturais, e se cristalizar em uma peça chamada “*Morro P.(41)*”. Performances mais recentes poderão ser acompanhadas pelo site <http://oficinadolui.wordpress.com> ; Meu endereço: oficinadolui@yahoo.com.br



Tela principal do projeto. Começando de cima, em sentido anti-horário, temos os controles da tela de projeção, os botões de acionamento do Max (áudio e vídeo) e (o *armbutton* do canal 1) do Live, seguidos dos contadores de seções (azul e rosa) e tempo, e dois painéis com as funções 1 e 2 do *wiimote* preto e branco (o *toggle* rosa - *s opwii* – realiza a seleção da opção 1 ou 2 das teclas *plus* e *minus* do *wii* branco). Em baixo, variações do *centroid~*, e notas Midi e amplitude, fornecidos pelo *fiddle~* (áudio do violino). Em cima destes há um pequeno painel (em vermelho) para acionarmos a contagem das seções manualmente. No canto inferior direito temos a edição do teclado colorido, e logo em cima mecanismos de monitoramento do *follow*, do teclado Midi e do pedal *sustain* (quando estiver acionando efeitos de áudio). Por fim, um quadro (em azul) onde se situam os 5 *patchers* (na verdade *subpatchers*), representados com as iniciais “p” analisados nesta dissertação.

Bibliografia

BATISTA, André. **Funções da música no cinema: contribuições para a elaboração de estratégias composicionais**. Dissertação de mestrado. EMUFG, 2007.

* * *

BONGERS, B. **Physical Interfaces in Eletronic Arts – Interaction Theory and Interfacing Techniques for Real-Time Performance**. In M.M.Wanderley and M. Battier (eds.) *Trends in Gestural Control of Music*. Paris: IRCAM – Centre Pompidou 2000.

* * *

CHADABE, J. **The Meaning of Iteration, a Public Talk Given at the Workshop in Interactive Systems in Performance (Wisp)**. *Proceedings of the 2005 HCSnet Conference*. Macquarie University, Sydney, Austrália 2005.

* * *

CHARNEY, Leo, SCHARTZ, Vanessa R.(org.). **O Cinema e a invenção da vida moderna**. São Paulo: Cosac&Naify, 2001.

* * *

COOK, Nicholas. **Analysing Musical Multimedia**. Oxford: Oxford University Press, 1998.

* * *

DRUMMOND, Jon. **Understanding Interactive Systems**. *Revista Organised Sound* **14** (2) : 124 - 133. UK: Cambridge University Press, 2009.

* * *

FERRAZ, Silvio; ALDROVANDI, Leonardo. **Loop-interpolation-random & gesture: déjà vu in computer-aided composition**. *Revista Organized Sound* **5** (2) : 81 – 84. UK: Cambridge University Press, 2000.

* * *

FREIRE GARCIA, Sérgio. **Alto-,Alter-,Auto-falantes: Concertos Eletroacústicos e o ao Vivo Musical.** Tese de Doutorado, PUC-SP, 2004.

* * *

GARCIA, Denise. **Modelos Perceptivos na Música Eletroacústica.** Tese de Doutorado, PUC-SP, 1998.

* * *

GUIMARÃES, Marta M. **Mesa de Luz: Colagem-Composição.** Dissertação de Mestrado, Brasília, UnB, 2010.

* * *

IAZZETTA, Fernando. **Sons de Silício : Corpos e Máquinas Fazendo Música.** Tese de Doutorado, PUC-SP, 1996.

* * *

JORDÀ, S. **Digital Lutherie: Crafting Musical Computers for New Music's Performance and Improvisation.** Phd dissertation, Universitat Pompeu Fabra, Barcelona, 2005.

* * *

MIRANDA, E.R. And WANDERLEY, M. **New Digital Musical Instruments : Control and Interaction Beyond the Keyboard.** Middleton, WI: A-R Editions, 2006.

* * *

MICHAÏLOWSKY, Alexei. **Deleuze e o Live: a criação-performance pela improvisação em um software para músicos digitais.** 1º Simpósio Brasileiro de Pós-graduandos em Música/UNIRIO, pág.423 a 431 , novembro de 2010.

* * *

MOORE, F. Richard. **The Disfunctions of MIDI** - Proceedings of the 1987 International Computer Music Conference. San Francisco: International Computer Music Association 256-262, 1987.

PAINE, G.. **Interactivity. Where to from Here?** Revista *Organised Sound* 7 (3): 295-304 UK: Cambridge University Press 2002.

* * *

PUCKETTE, Miller S.; APEL, Theodore; ZICARELLI, David. D.. **Real-time audio analysis tools Pd and Msp**, reprinted from Proceedings, ICMC, 1998
<http://crca.ucsd.edu/~msp>

* * *

PUCKETTE, Miller S. **Is there life after MIDI?** Reprinted from ICMC proceedings, p.2, 1994. <http://crca.ucsd.edu/~msp>

----- **Max at Seventeen.** Computer Music Journal 26 / 4, p.31-43, 2002.

* * *

ROWE, Robert. **Interactive Music Systems:Machine Listening and Composing.** Cambridge: MIT Press,1993.

* * *

SETZER, Valdemar W. **Textos** disponíveis em www.ime.usp.br/~vwsetzer

* * *

SCHAFFER M. **O Ouvido Pensante.** São Paulo: Ed.Unesp,1992.

* * *

SULLIVAN, Dan O'; IGOE, Tom. **Sensing and Controlling th Physical World with Computers.** Physical Computing, by Thonson Course Technology PTR, Boston, 2004.

* * *

WINKLER, Todd. **Composing Interactive Music: Techniques and Ideas Using Max.** Cambridge : MIT Press, 1998.