

SoMo: Um Instrumento Musical baseado em Movimento – Características Estéticas e Técnicas e potenciais Usos

Tiago Brizolara da Rosa¹, Marcos Goularte Moritz

¹Departamento de Música – Universidade Estadual de Maringá (UEM)

tiago.brizolara@gmail.com, marcosmoritz@gmail.com

Abstract. *This paper describes the conception, development and current characteristics of a movement-based musical instrument developed using the free open-source tools Pure Data and Processing. The instrument can be assembled using an usual personal computer and even with a low cost video camera (we use PSEye®) we attained a responsiveness of up to 100 frames analysed per second and real-time audio generated. The system was tested in Linux Ubuntu and Windows 7 operating systems. Along with its low cost, the project is open-source, turning it into more than a musical instrument, a potential tool for education and experimentation.*

Resumo. *Este artigo descreve a concepção, desenvolvimento e atuais características de um instrumento musical baseado em movimento desenvolvido utilizando as ferramentas livres de código aberto Pure Data e Processing. O instrumento pode ser montado usando um computador pessoal comum e mesmo com uma câmera de vídeo de baixo custo (nós usamos a PSEye®) obtivemos uma responsividade de até 100 quadros analisados por segundo e áudio gerado em tempo real. O arranjo foi testado nos sistemas operacionais Linux Ubuntu e Windows 7. Além de seu baixo custo, o projeto é de código-fonte aberto, tornando-o mais que um instrumento musical, uma potencial ferramenta educacional e experimental.*

1. Introdução

Este artigo descreve as características técnicas e estéticas do estado atual de um instrumento musical digital baseado em movimento, desenvolvido de forma independente, com código aberto hospedado em <https://bitbucket.org/mmoritz/somo>. Iniciamos apresentando como nossa concepção dialoga com o conceito de instrumento musical.

1.1 Do conceito de Instrumento Musical

O conceito de instrumento musical vem há mais de cem anos sendo revisitado. Em 1901 acontecia a primeira audição pública do Telharmonium, instrumento eletromecânico capaz de mimetizar diversos instrumentos tradicionais, produzir sons jamais ouvidos e de transmitir sua música ao vivo através de linha telefônica - e também de criar oscilações na rede elétrica e interferências nas conversas telefônicas (Brend, 2012). Logo no começo do século XX, o movimento futurista aborda recursos eletrônicos e de outras tecnologias como parte de sua estética, trazendo como instrumentos musicais materiais que até então jamais seriam assim denominados - como turbinas de avião e buzinas. Em meados do séc. XX temos na França a *Musique Concrète*, concebida por

Pierre Schaeffer, com a teorização de objeto sonoro passando pela experiência que o compositor batiza de *acusmática*, onde importa somente o som e não suas fontes. A música concreta é elaborada manipulando a própria fita magnética na qual há sons gravados. Na Alemanha, a *Elektronische Musik* se propõe a produzir música por meio exclusivamente eletrônico¹, explorando sons até então não disponíveis aos músicos - ao menos não de forma controlada.

No seu *Traité des Objets Musicaux* (Schaeffer 1961, p. 40), Schaeffer, pai da Música Concreta, coloca que “todo dispositivo que nos permite obter uma variada coleção de objetos sonoros ... mantendo no seu cerne a permanência de uma causa é um instrumento musical, no sentido tradicional de uma experiência comum a todas as civilizações” (tradução livre). Ruviaro (2012) propõe uma extensão da definição de Schaeffer para trabalhar o conceito nas *Lorks* (*Laptop Orchestras*), onde comumente há uma fricção entre a expectativa de uma experiência não-acusmática (os músicos estão no palco “executando” seus laptops) e a experiência de fato onde à plateia não se apresenta com clareza uma relação entre os sons e a presença dos instrumentistas com seus laptops e falantes (esse tipo de experiência com música computacional é precisamente a motivação do *survey* conduzido por Magnusson and Hurtado (2008) com praticantes de instrumentos digitais e acústicos). Para Ruviaro, os três conceitos a seguir devem ser levados em conta ao pensarmos em instrumento musical:

- Presença: a aparência física do instrumento e como o instrumentista se posiciona em relação a ele;
- Movimento: gestos e movimentos necessários para executar o instrumento; gestos e movimentos que são idiomáticos mesmo sem gerar som; uma máxima relativa aos instrumentos musicais tradicionais: “quando o movimento pára, o som pára ou começa a extinguir-se”;
- História: um instrumento faz parte de uma ou mais culturas. Há contextos onde ele costuma estar, repertório típico e forma típica de execução para cada contexto.

1.2 Concepção Inicial do SoMo

A ideia básica do SoMo (das palavras som e movimento, e recordando *soma* - corpo, em grego) é a de um instrumento cuja interface é o vídeo, capturando o executante. Foi tentadora a ideia de utilizar o Microsoft Kinect®: com ele poderíamos mapear as 20 *joints* fornecidas pelo sensor (cabeça, mãos, quadris, pés, etc) a parâmetros de um sintetizador. Não enveredamos por esse caminho por causa da baixa taxa de dados que esse sistema fornece: 30 pacotes de dados por segundo (ou quadros por segundo, abreviados doravante por FPS - *frames per second*), o que só permitiria fineza na dimensão temporal com movimentos lentos. Segundo Menezes (2004), seções 3.4.4, 3.4.5], podemos dizer que o limiar de reconhecimento dos sons por parte do ouvido é da ordem de 50 ms - para o que 30 FPS bastariam -, mas objetos sonoros de mesma proveniência instrumental possuindo variação de altura entre si podem ser distinguidos com apenas 15 ou 20 ms - muito menos que a duração de um quadro a 30 FPS, que é de

¹ O que não excluiu a criação de obras mistas por parte desses compositores. Por exemplo, a peça *Mixtur*, de Karlheinz Stockhausen, 1964, foi composta para orquestra, três osciladores senoidais e 2 moduladores em anel.

33,3 ms. Assim, consideramos 30 FPS uma taxa desconfortável por estar ao redor de limitações do ouvido. Além disso, na seção 3 falamos sobre como as altas taxas de FPS conseguidas na captura e processamento do vídeo foram vitais não só para o áudio, mas para a experiência de imersão do executante com o uso do *feedback* visual (a decisão pelo *feedback* foi feita durante o desenvolvimento e é abordada nas seções 2.1.5 e 3.3).

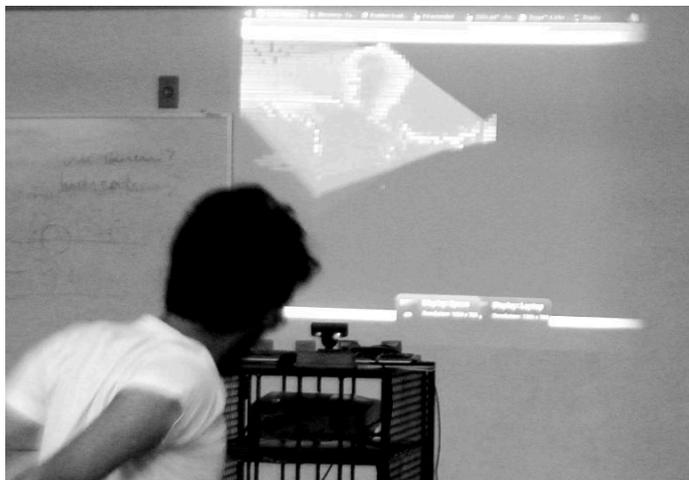


Figura 1. SoMo em uso. A projeção representa as alterações no vídeo em relação ao *frame* anterior e serve como *feedback* para o executante. Ao centro, sobre a mesa, encontra-se a câmera PSEye®. Os alto-falantes não aparecem na imagem.

Ao eleger o vídeo como interface há as desvantagens de não termos a abstração das partes do corpo do executante, mas pudemos nos valer de outras possibilidades abertas: podemos usar objetos; nos afastar ou aproximar da câmera à vontade (enquanto o Kinect tem uma cobertura pequena de distâncias possíveis da pessoa ao sensor); ou mostrar qualquer porção do corpo no vídeo ao invés do corpo todo.

Podemos dizer que as quatro seguintes decisões estéticas - não totalmente independentes - foram as chaves para guiar a construção do instrumento:

1. *O instrumento não deve ser um simples sonificador de dados*, como os instrumentos tradicionais também não são. Até mesmo quando tocados de modo alheio ao fazer artístico, como por robôs ou quando um instrumento é excitado acidentalmente (numa queda, ao ser atingido por algo...), percebemos que podem gerar uma gama muito grande e interessante de sons. Dizer que um piano é um “sonificador de dedos” pode até ter rigor de um ponto de vista estritamente mecânico, mas faz pouco sentido ao falarmos de música;
2. *O som deve ser produzido pelo movimento* (mudanças no vídeo) e não somente pela presença (estado instantâneo do vídeo). Essa decisão encontra eco na característica de movimento dos instrumentos musicais levantada na seção 1.1. Uma leve diferença é que no SoMo, quando não há movimentos no vídeo ainda resta um som em baixa intensidade;
3. *O som deve ser sintetizado* ao invés de serem trabalhadas sonoridades tradicionalmente consideradas musicais. Porque almejamos uma experiência onde o executante sente estar extraindo material sonoro do seu movimento e não

realizando uma experiência de manipulação/transformação de certos sons ao seu movimento ou efetuando um *DJing*²;

4. Esta decisão está aparentada à primeira. *A experiência de tocar o instrumento não deve se esgotar facilmente*, deve existir uma profundidade. Não há problema que o instrumento não seja fácil de dominar inicialmente - ninguém sai fazendo musica num violoncelo na primeira tentativa.

Os detalhes da descrição do som do SoMo estão na seção 2.2.

2. Materiais e Métodos

O arranjo da plataforma atual de testes é: câmera Playstation PSEye® (o preço de uma nova varia entre R\$ 60,00 e R\$ 100,00) no modo de 100 FPS; um notebook com processador Intel Core i3 2.20 GHz, 3 GB de RAM e sistema operacional Linux - tivemos dificuldades em fazer a câmera PSEye funcionar com o Java no Windows - ; uma placa de áudio PreSonus AudioBox 22VSL. Os softwares utilizados são todos de código-fonte aberto. Para a captura, processamento e geração de vídeo fizemos um programa em Java utilizando as bibliotecas do Processing 2.0³, com o ambiente de desenvolvimento integrado Eclipse. A cada quadro do vídeo passamos os dados da sua análise para um *patch*⁴ da plataforma de programação de fluxo de dados Pure Data Extended⁵ (Pd, em abreviatura).

2.1 Processamento do vídeo

O processamento do vídeo tem dois objetivos: 1) detectar movimentos, analisá-los e traduzir essa análise na forma de quantidades geométricas (doravante denominadas *descritores*) a serem enviados como parâmetros para o sintetizador de áudio; 2) fornecer *feedback* para o instrumentista acerca de seu próprio movimento e trajetória.

Ambas as abordagens devem ser otimizadas, visto que busca-se a análise na mesma taxa de atualização da câmera, que pode alcançar 100 FPS. As principais etapas lógicas estão resumidas na Tabela 1.

Tabela 1. Algoritmos de processamento de vídeo

Algoritmo	Objetivo
Análise dos Setores	Detectar movimento; Atenuar menores variações, inclusive ruído
Clusterização (opcional)	Reconhecer agrupamentos isolados de setores ativos (chamados de <i>clusters</i>).
Envoltória Convexa	Definir um polígono que englobe os setores de um <i>cluster</i> e criar um objeto que definiremos como <i>corpo</i> com função de complementar o feedback visual e disponibilizar propriedades geométricas associadas a um <i>cluster</i> .

² O trabalho do DJ (disc-jóquei).

³ <http://processing.org>

⁴ Os programas feitos com o Pure Data costumam ser denominados *patches*.

⁵ <https://puredata.info/downloads/pd-extended>

2.1.1 O Setor de Análise

Ambos os requisitos - detecção de movimento e geração de *feedback* - são alcançados através da implementação de uma abstração que chamamos de **setor de análise** de vídeo e implementamos através da classe *AnalysisSector*, dividindo a tela em uma grade de setores retangulares (Figura 2). Cada setor é uma instância desta classe, contendo atributos que definem:

- A **posição** do setor;
- O **tamanho** em pixels na horizontal e na vertical;
- A **amostra** da análise: quantidade de pixels a serem analisados no setor (definimos o padrão como $\frac{1}{4}$ dos pixels abarcados pelo setor), como esquematizado na Figura 3;
- A **cor extraída** do setor.

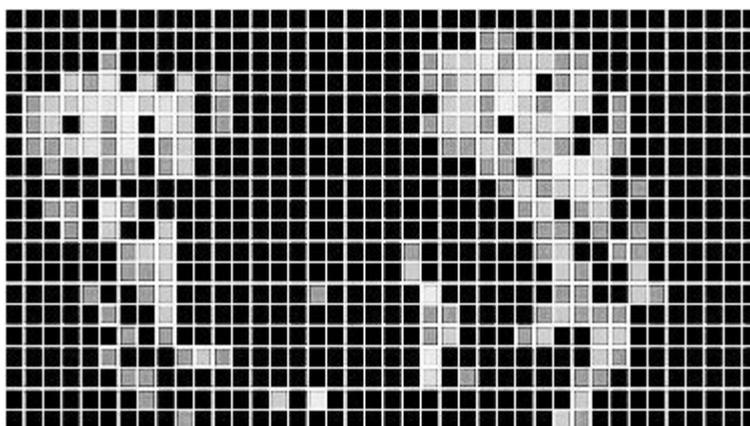


Figura 2. Os setores não pretos representam porções da imagem onde houve movimento (as linhas brancas são apenas para fins de ressaltar os setores nesta imagem e o polígono representando a envoltoria convexa está omitido). A estratégia de divisão em setores minimiza o efeito de ruídos e economiza processamento.

A **cor extraída** de um setor é a média das cores dos pixels da amostra do setor. Essa média ajuda a suavizar variações de um *frame* para outro, evitando falsos positivos advindos de ruído e movimentações sutis no vídeo. Se de um *frame* para o próximo essa média varia além de uma tolerância previamente fixada, determinamos que houve movimento naquele setor e ele passa a estar no estado **ativo**.

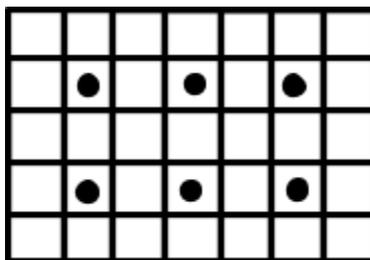


Figura 3. Representação esquemática de um setor de análise de 7 x 5 pixels. Apenas os pixels indicados com um círculo terão suas cores contabilizadas na média que confere a cor extraída do setor.

O fato da amostra ser menor que a quantidade de pixels abarcados pelo setor (ao redor de $\frac{1}{4}$) confere economia de processamento.

Na configuração atual a grade de setores tem 80 colunas e 48 linhas.

2.1.2 Corpos

Utilizando os setores ativos mais externos construímos a envoltória convexa (o conjunto de todas as combinações convexas de envoltórias formadas pelos centros dos setores ativos), que chamaremos simplesmente de **corpo** (Figura 4). A representação do corpo como envoltória convexa favorece a extração de propriedades geométricas, faz parte da estratégia - em fase de avaliação - de identificar vários corpos e é utilizada no *feedback* visual como um polígono com transparência, como podemos ver nas Figuras 1, 4 e 5.

2.1.3 Separação em Clusters (opcional)

Para que pudéssemos identificar movimento de partes isoladas da imagem, i.e., vários corpos, elaboramos um algoritmo capaz de interpretar se a nuvem dos setores ativos pertence ou não à uma única fonte de movimento, na forma de uma versão do algoritmo de classificação *K-Means* que, dado um número de agrupamentos (*clusters*), associa cada setor a um *cluster*. Neste algoritmo, um número k de centros c_i , $i \in [1, k]$ são posicionados arbitrariamente e cada setor é classificado como pertencente ao *cluster* cujo centro está mais próximo de sua posição. Os centros são recalculados a cada iteração pela média da posição do setores associados à cada *cluster*. O algoritmo converge quando nenhum setor muda de *cluster* em duas iterações consecutivas.

O resultado deste algoritmo depende da posição inicial dos centros, que é aleatória, e o algoritmo precisa rodar diversas vezes por *frame* para garantir um resultado confiável.

Cada *cluster* será matéria prima para construção de um corpo. A Figura 4 mostra um exemplo com reconhecimento de 2 clusters. No entanto, na configuração padrão do SoMo, não há clusterização - consideramos sempre que há um único corpo.

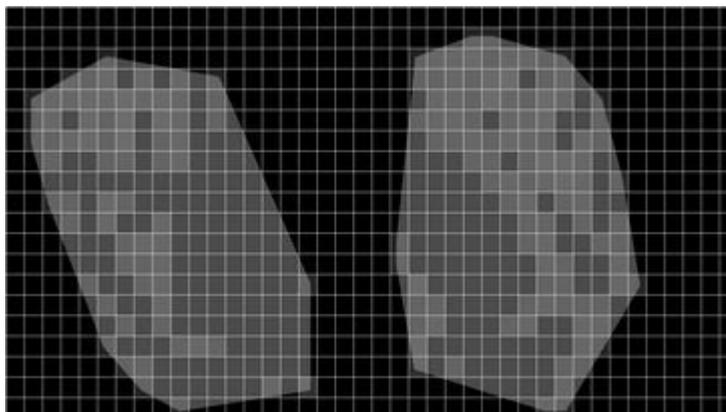


Figura 4. Grupos de setores são classificados como pertencentes ao *cluster* cujo centro se localiza mais próximo dele. Os pontos em cinza claro pertencem ao *cluster* 1, os em cinza escuro ao *cluster* 2. Os demais pontos são setores inativos. Nas configurações padrão do SoMo essa clusterização não é utilizada - definimos apenas um corpo.

2.1.4. Análise do movimento

A cada *frame* são extraídos os seguintes descritores do *corpo* (ver também Figura 7):

- n. Quantidade de setores ativos;
- c. Coordenada horizontal do centróide dos setores ativos;
- w. Cobertura horizontal compreendida pelos setores modificados no último *frame*;
- $\check{y}1$. Distância vertical do limite superior do corpo ao centro da tela;
- $\check{y}2$. Distância vertical do limite inferior do corpo ao centro da tela;

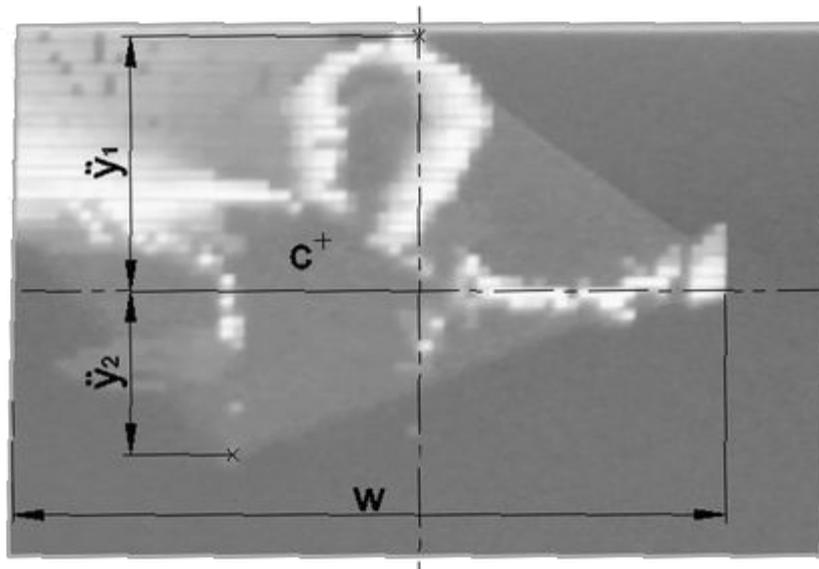


Figura 5. O vídeo de *feedback* consiste dos setores que acusaram modificação do *frame* anterior para o atual, mais um polígono representando a envoltória convexa do *corpo* definido por esses setores. Dessa envoltória são extraídos os seguintes descritores, para serem mapeados depois nos parâmetros do sintetizador: cobertura horizontal w , centróide c , extremos verticais $\check{y}1$ e $\check{y}2$ e quantidade de centróides modificados n (este não representado na imagem).

2.1.5. Feedback

O vídeo de *feedback* fica voltado para o executante e é exibido a 100 FPS. Como mostrado nas Figuras 1, 2, 4, 5, 6, apenas os setores ativos são exibidos no vídeo, sob fundo preto. Os setores que estavam ativos nos quadros anteriores também aparecem, com um grau de transparência proporcional a seu tempo de existência, gerando uma clara representação da trajetória dos movimentos (Figura 6).

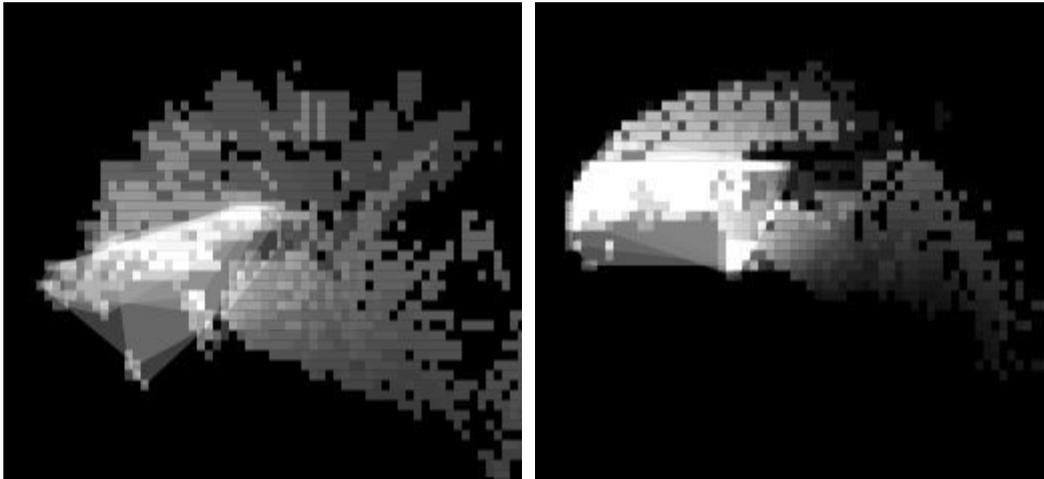


Figura 6. O *feedback* de vídeo inclui setores ativos de quadros anteriores, conferindo a noção de trajetória. Na figura temos um movimento rápido de braço. Esquerda: *feedback* de vídeo gerado a 60 FPS. Direita: *feedback* de vídeo a 100 FPS. Em ambos os casos a captura de vídeo está a 100 FPS.

A *envoltória convexa* do eventual *corpo* detectado é exibida como um polígono vermelho com transparência. Portanto, sua presença indica que o movimento está influenciando o áudio (omitimos sua presença na Figura 2).

2.2. Geração do Áudio

O áudio é gerado por um sintetizador desenvolvido na forma de um *patch* de *Pure Data*. Os valores dos descritores extraídos do vídeo são enviados continuamente do aplicativo Java para o *patch* via protocolo OSC (*Open Sound Control*) e cada um deles passa por uma função de calibração e é então roteado para um parâmetro diferente do sintetizador.

Indo ao encontro das decisões estéticas apresentadas na Introdução, a sonoridade buscada trabalha com os limites da sensação de altura definida, em um *patch* consistindo em:

- *OsSo* ("oscilador solitário"). Um oscilador senoidal que muda de frequência dezenas de vezes por segundo, sempre para uma frequência aleatoriamente escolhida dentro de certo intervalo. Seus parâmetros:
 - Amplitude A ;
 - Frequência central F ;
 - Intervalo ou janela de frequências W ;
 - Período de mudança de frequência T
- *Distorter* (opcional). Um nível de valor D adicionado ao sinal final, no sentido de saturar o sinal, distorcendo o áudio;
- *Uivo* (opcional). Um oscilador senoidal com frequência igual a $F/2$ (metade da frequência central do *OsSo*).

O som é espacializado em estéreo de forma coerente com a posição horizontal do centroide c . Quando não há movimentos no vídeo ainda resta um som em baixa intensidade.

Os parâmetros estão mapeados como indicado na Tabela 2:

Tabela 2. Mapeamento dos descritores do vídeo para os parâmetros do sintetizador (width e height são as dimensões do vídeo)

Parâmetro do sintetizador	Intervalo	Descritor do vídeo	Intervalo
Amplitude A	{0,1.2}	n . Quantidade de setores ativos no último <i>frame</i>	{0, 80x40} ⁶
Frequência central F	{100, 460} Hz	c . Coordenada horizontal do centróide do corpo	{0, width}
Janela de frequências J	{2, 460} Hz	w . Cobertura horizontal compreendida pelos setores ativos no último frame	{2, width}
Período de mudança de frequência T	{10, 20} ms	$y1$. Distância vertical do último setor ativo mais superior ao centro da tela.	{height/2., 0}
Nível do Distorter D	{0, 1}	$y2$. Distância vertical do setor ativo mais inferior ao centro da tela.	{height/2., height}

Os baixos valores de T fazem com que mesmo com apenas um oscilador (caso do *uivo* desligado), grandes quantidades de frequências sonoras estejam presentes em curtos intervalos de tempo. Podemos pensar no OsSo como uma espécie de síntese de frequência modulada (FM) com a modulação sendo dada por um sinal aleatório com características dadas pela frequência central F e janela de frequências J - não é exatamente o caso porque no OsSo a senóide não muda de frequência antes de completar um ou mais ciclos.

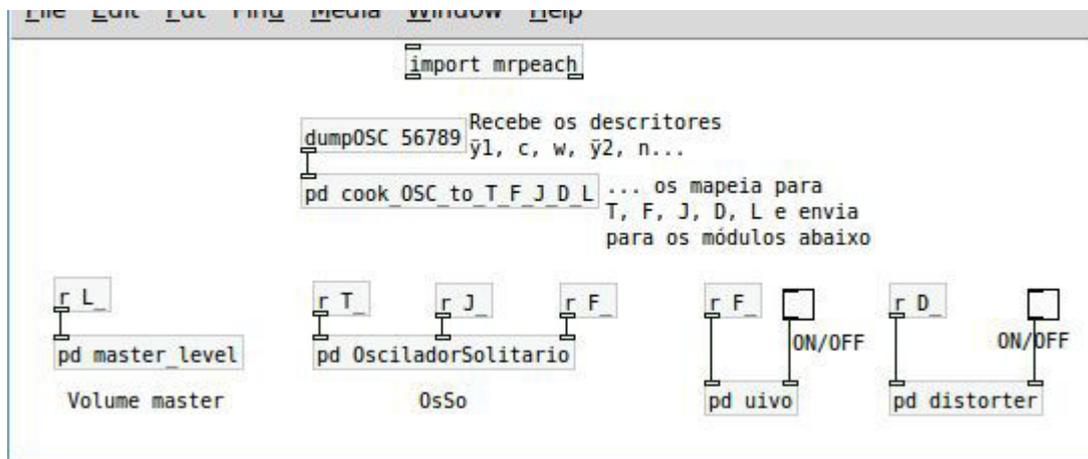


Figura 7. Patch do sintetizador, mostrando o mapeamento dos descritores do vídeo para os parâmetros do sintetizador, constituído pelos *sub-patches* [master_level], [OsciladorSolitario], [uivo], [distorter].

Ao instrumentista diminuir a janela de frequências J através da ocupação de menos espaço horizontal na tela, obterá um som parecido com um vibrato errático em torno de uma nota, e pode ser trabalhoso manter uma pequena porção do vídeo ativa sem deslocar-se horizontalmente. Ao aumentar J , o som ganha características de um borbulhar, agregando um pouco de ruído quando a janela é demasiadamente aberta, o que pode ser amenizado aumentando T se o instrumentista atingir porções superiores do vídeo. O *uivo* ligado fornece a possibilidade de construir linhas melódicas sem o esmero

⁶ 80x40 são as dimensões da grade de setores.

de manter uma pequena janela J - além, de obviamente, mudar a qualidade (timbre) do som. O *Distorter*, mais pronunciado quando o instrumentista alcança porções inferiores do vídeo, é um recurso timbrístico.

3. Resultados

3.1. Implementação

Do lado técnico, conseguimos com o arranjo de baixo custo descrito na seção 2.1. utilizar a câmera PSEye a 100 FPS (não conseguimos acessar os modos de captura de 125 FPS); executar o aplicativo Java também a 100 FPS - isto é, depois de bastante otimização conseguimos fazer a análise completa de cada *frame* antes de receber o próximo - e atualizar o sintetizador no Pd de acordo com os parâmetros em tempo real. Testes sem uma placa de áudio dedicada também não mostraram prejuízo da experiência.

3.2. Uso

O parâmetro de período de mudança de frequência T , mapeado pelo descritor de distância vertical superior y/l , pouco influenciou na experiência. E se aumentarmos o intervalo de T 's possíveis, a sonoridade muda demasiadamente - o ouvido passa a tentar acompanhar cada variação de frequência -, de um modo que nos parece perturbar a coerência da experiência. Portanto, consideramos o parâmetro T e seu mapeamento como não acabados.

Algumas características que ressaltamos dos testes preliminares:

Imersão - Nos testes preliminares, executantes advindos de experiências distintas - música, artes visuais, artes cênicas, ciências exatas, entre outras - avaliaram o uso do SoMo como de grande grau de imersão e estimulador da exploração.

Funcionamento - os princípios do funcionamento do instrumento não foram óbvios aos participantes, embora facilmente explicáveis aos músicos. Aos não músicos bastou uma explicação como: “imagine que você tem um monte de macaquinhos aos seus dois lados, espalhados sobre as teclas de um piano e tocando-as freneticamente. Quanto mais espaço você ocupar na horizontal do vídeo, mais espalhados no piano estarão os macaquinhos, gerando sons então por uma grande faixa desde graves até agudos. Quanto menos espaço horizontal você ocupa, mais os macaquinhos estarão junto a você, tocando, assim, um pequeno intervalo de teclas. Já seus movimentos na vertical mudam a sonoridade do instrumento”.

Percussão - O caráter imediato da resposta sonora e visual do sistema possibilitou o uso do instrumento como percussão. Mesmo questões de variação de sonoridade como as relatadas por Silva et al (2013) como difíceis de conseguir com interfaces digitais, podem ser alcançadas com variação de origem e sentido do gesto, forma do objeto móvel (mão aberta ou fechada, braço, objeto, proximidade com a câmera, etc), presença ou não de *Clipper* e *Uivo* e ainda a presença de mais partes móveis, para abertura da janela de frequência W . Tudo isso (exceto a presença/ausência do *Distorter* e *Uivo*) ainda pode ser variado durante o gesto. Novamente, acreditamos

que as estratégias que adotamos tiveram sucesso em conseguir a *experiência com profundidade* que foi norteadora do projeto.

Formas de uso - Até o momento, diversas formas de uso foram observadas. Em uma delas, um instrumentista se posiciona num extremo lateral e o outro, no extremo oposto e tocam o instrumento alternadamente quando querem uma janela pequena e juntos quando querem uma janela (bastante) grande⁷. Em outra, um instrumentista *toca* tríades alternando movimentos de uma mão, cabeça e a outra mão. Com *Uivo* e *Distorter* ligados, o abrir e fechar de uma mão próxima à câmera, deslocada à esquerda, soa como um coração pulsante.

Gestuais típicos - Gestos típicos foram o mover rápido de dedos para obter corpos bem localizados e o abaixar de um braço ou os dois ou o corpo todo com braços abertos, gerando sonoridades tipicamente encorpadas e delineadas no tempo.

3.3. Feedback visual, imersão

Se nos remetermos ao Theremin, um dos instrumentos eletrônicos mais antigos e de maior sucesso e longevidade, podemos vir a pensar que num instrumento todo *feedback* (exceto o sonoro, claro) pode ser dispensado.

No entanto, Magnusson e Hurtado (2008) mostram que a ausência de *feedback* tátil de vários instrumentos digitais é muito sentida. No nosso caso, verificamos que a implementação do *feedback* de vídeo, inicialmente apenas para análise, mudou radicalmente a experiência. Isso se deve em grande parte pelo instrumento não detectar movimentos mínimos, em razão de sua resolução e de um limiar programado. Tendo o *feedback*, ao começar a se movimentar, o instrumentista percebe o quanto e que partes de seu corpo estão se movendo, e quando o movimento atinge o limiar, a *envoltória convexa* do *corpo* gerado é exibida, deixando claro que o movimento passa a gerar e modificar o som.

O fato de nem todo movimento influenciar o instrumento propôs naturalmente aos instrumentistas uma estratégia de mover-se sorratamente até preparar-se para executar determinados movimentos. Consideramos que esse tipo de uso, bastante constante, foi um fator contribuinte no estabelecimento de períodos de grande imersão dos instrumentistas.

Como o *feedback* visual exhibe os rastros do movimento, avaliamos que para não prejudicar essa imersão, deveríamos ter grandes FPSs de vídeo exibido, o que nos levou a chegar aos 100 quadros por segundo - caso contrário, a trajetória aparece muito cortada, como uma imagem estroboscópica (ver Figura 6). Não fosse esse fator de imersão, poderíamos ter usado 60 FPS, que haviam se mostrado satisfatórios para controle do sintetizador e que já são uma alta taxa para vídeo.

4. Observações finais e Trabalhos futuros

A transformação em diferentes instrumentos pode ser feita facilmente roteando os descritores a qualquer outro sintetizador em *Pd* ou em outra plataforma que possa interpretar o protocolo OSC, ou seja, tratando o módulo de vídeo como uma interface.

⁷ Exemplo, entre 0:05 e 00:10s do vídeo https://www.youtube.com/watch?v=XWqqz_XgEHU#t=5

Acreditamos que em razão do baixo custo do arranjo, o SoMo tem grande potencial para o uso no contexto educacional (não necessariamente em nível superior).

Futuramente, pretendemos explorar possibilidades de outros descritores, como razão largura/altura do corpo, eixo principal, velocidade do centroide e mesmo análises mais complexas que possam dizer mais do que se passa com o corpo do instrumentista - por exemplo, utilizando visão computacional. Também aventamos a possibilidade de identificar e responder a gestuais.

O SoMo ainda é novo. Podemos, por fim, propor descompromissadamente formas de execução que ainda não experimentamos a contento, como:

- Lançar objetos - movimentos tipicamente rápidos de objetos lançados são possíveis graças à responsividade do SoMo. Com isso, consegue-se gestos que não seriam possíveis utilizando o corpo.
- Pêndulo - objetos pendurados podem ser excitados pelo executante. A característica rítmica desse tipo de dispositivo pode vir a ser esteticamente interessante.
- Ponto luminoso - apagar as luzes faz com que um objeto luminoso - como um telefone celular, um fósforo... - atue como um ponto voador, entidade mais difícil de obter com movimentos corporais
- Plataformas - ao atuar sobre uma plataforma, tem-se um renivelamento das transformações verticais de timbre

5. Agradecimentos

A Élder Sereni por testes iniciais e ideias, Dimitri Camorlinga e Fabio Yudi Yokomizu por testes, ideias e experimentações em montagens no curso de Artes Cênicas na Universidade Estadual de Santa Catarina-UDESC.

6. Referências

- Brend, M., *The Sound of Tomorrow: How Electronic Music Was Smuggled into the Mainstream*, Bloomsbury Publishing USA, 2012.
- Ruviaro, R. (2012) "From Schaeffer to *Lorks: An extended definition of Musical Instrument in the context of laptop orchestras", In: *Proceedings of the 1st Symposium on Laptops Ensembles and Orchestras*, Louisiana State University, Baton Rouge, Louisiana.
- Magnusson, T. and Hurtado, M. (2008) "The Phenomenology of Musical Instruments: A Survey", http://econtact.ca/10_4/magnusson_hurtado_survey.html (acessado em 12/09/2015).
- Menezes, F., *A Acústica Musical em Palavras e Sons*, Atelie Editorial, 2004.
- Schaeffer, P., *Traité des Objets Musicaux*. Traducción de Araceli Cabezon de Diego, Alianza Editorial, 2003.
- Silva et. al. (2013) "Avaliando Interfaces Gestuais Para Prática de Instrumentos Virtuais de Percussão", In: *Simpósio Brasileiro de Computação Musical*, Escola de Música de Brasília, Distrito Federal.