

# Processamento visual de informações musicais em tempo real

Pedro Henrique Ventura Rodrigues de Almeida<sup>1</sup>, José Honório Glanzmann<sup>1</sup>

<sup>1</sup> Núcleo de Informática – Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais (IFSUDESTEMG)

Rua Bernardo Mascarenhas, 1283 - Fábrica – 36.080-001 – Juiz de Fora – MG – Brasil  
phventurarodrigues@gmail.com, jose.honorio@ifsudestemg.edu.br

**Abstract.** *This article proposes the development of software capable of generating audiovisual content suitable for musical presentations in real time and low cost. To achieve this goal, a research was carried out on known bases related to musical visualization, computer graphics and sound capture. Subsequently, researches on methods, media, languages and libraries were developed for the development of software for code interpretation and understanding of sound signals. Throughout the work a study was carried out regarding recent computational technologies for audiovisual and graphic development; algorithms and methods for reading and interpreting real-time sound to obtain interactive visual effects for use in live musical presentations.*

Key words: music visualization, sound analysis, processing.

**Resumo.** *Este artigo propõe o desenvolvimento de um software capaz de gerar conteúdo audiovisual próprio para apresentações musicais em tempo real e de baixo custo. Para realizar tal objetivo, realizou-se uma pesquisa em bases conhecidas relacionadas a visualização musical, computação gráfica e captação sonora. Posteriormente foram desenvolvidas pesquisas sobre métodos, meios, linguagens e bibliotecas, para o desenvolvimento do software adjunto a códigos para interpretação e entendimento de sinais sonoros. Ao longo do trabalho foi realizado um estudo referente à tecnologias computacionais recentes para desenvolvimento audiovisual e gráfico; especificidades sonoras; algoritmos e métodos para leitura e interpretação do som em tempo real para obtenção de efeitos visuais interativos para serem utilizados em apresentações musicais ao vivo.*

Palavras-chave: visualização musical, análise sonora, processing.

## 1. Introdução

### 1.1 Contextualização

A humanidade possui a necessidade de explorar, descobrir e se expressar. Durante milênios a arte foi evoluindo, desde pinturas feitas com sangue à pedra, até a arte digital. Com o advento da tecnologia digital, surgiram diversas ferramentas para criação artística, em consequência, novas formas de arte.

Aplicações de tecnologia audiovisual em espetáculos ao vivo são encontradas cada vez mais, em grandes shows, como pode ser percebido em espetáculos recentes, como o do cantor Roger Waters, em São Paulo, em sua turnê “*Us+Them*”. O espetáculo contou com mesclagem de efeitos em telões e câmeras de filmagem, efeitos com lasers, luzes e drones.



**Figura 1. Show US+Them - São Paulo Capital, Brasil (09/10/2018)  
Foto: Stephanie Hahne/TMDQA!**

Outro exemplo do uso de tecnologia digital em apresentações musicais são os *Vocaloids* (Kenmochi, 2010). Tratam-se de cantores sintetizados por software desenvolvido pela Yamaha capazes de gerar digitalmente voz e música. Este tipo de recurso tecnológico vem ganhando grande força nos últimos anos, principalmente na Ásia. Espetáculos, como da vocaloid “Hatsune Miku”, contaram com a presença da mesma, projetada em um holograma 3D, ao som de uma banda ao fundo (CONNER, 2013). A *vocaloide* obtém aceitação, inclusive de artistas já consagrados, como Lady Gaga, que teve seu show aberto pela mesma, em sua digressão “The ARTPOP Ball” em 2014, entre 06 de maio e 03 de junho (NBCNews, 2014).



**Figura 2. Show da Hatsune Miku - Holograma 3D.  
Extraída de NBCNews em (01/11/2018)**

Além das apresentações musicais, percebe-se o uso da tecnologia digital em programas de televisão, espetáculos circenses, cênicos, de dança, dentre outros.

Neste contexto, o presente trabalho tem por objetivo, criar um novo conteúdo audiovisual, com o intuito de ser utilizado em apresentações musicais/artísticas em tempo real e com baixo custo. Tendo como o público alvo do produto gerado, pequenos e micro produtores de apresentações musicais, que não possuem condições de obter equipamentos sofisticados para geração de efeitos ou contratar artistas/empresas especializados.

## **2. Aportes Teórico**

Para a compreensão do trabalho, três conceitos precisam ser definidos previamente, Som, Forma e Programação. Os mesmos serão descrito a seguir.

### **2.1 O Som**

Segundo Paula Filho (2013), em seu livro “Multimídia: Conceitos e Aplicações”, o som captado pelo ser humano trata-se de flutuações periódicas da pressão da atmosfera normal, que convertem em sinais elétricos ao cérebro, restrito a uma faixa em que o ser humano consegue captar (entre 20 Hz e os 20.000 Hz). Contudo, atualmente, dispositivos eletrônicos conseguem captar, armazenar, transmitir e sintetizar além da faixa audível pelo homem, demais frequências, captando ou sintetizando suas vibrações sonoras, com microfones e alto-falantes.

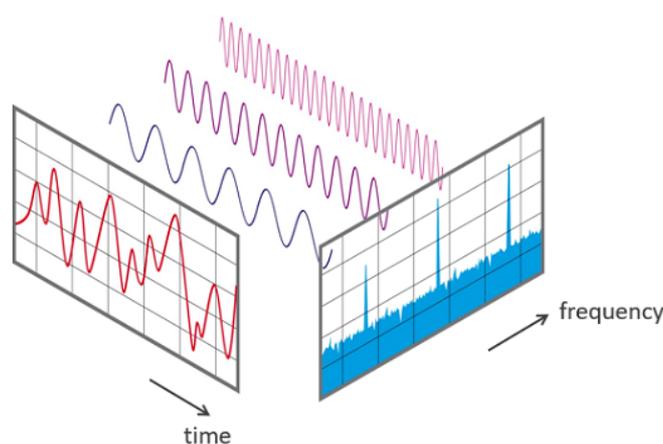
O autor cita alguns parâmetros Perceptuais do Som;

- Intensidade: capacidade de distinguir sons fortes e fracos;
- Altura: capacidade de distinguir sons graves de agudos;
- Timbre: capacidade de distinguir sons de intensidade e altura iguais mas gerados por instrumentos distintos, associado a fase.

Para poder registrar o som no computador, é necessário que o dispositivo de entrada converta tais variações em uma representação analógica e, posteriormente, em uma em um sinal digital, sendo valores discretos no tempo e amplitude (Martins, 2008).

Dois conceitos essenciais para o entendimento deste é o de frequência e amplitude, uma vez que os produtos deste trabalho foram gerados a partir da análise da amplitude sobre a frequência em relação ao tempo. Para McLeod (2003), frequência trata-se da quantidade de ciclos dados por uma onda por segundo, enquanto a amplitude é a distância da crista da onda ou vale em relação a um eixo. Tais informações podem vir a ser obtidas pela transformada de Fourier, citada mais a frente.

Para um melhor entendimento, conforme pode ser visto na Figura 3, a amplitude sobre a frequência pode ser vista como a intensidade, ou volume do áudio, enquanto sobre o tempo, trata-se da variação sonora, podendo ser interpretado como um pulso ou ritmo. A frequência em si, pode ser utilizada para distinguir um som mais agudo, de maior frequência, de um mais grave, menor frequência (Fernandes, 2002).



**Figura 3. Amplitude em uma visão 3D sobre Tempo e Frequência.**  
Phonical, CC BY-SA 4.0

## 2.2. A Forma (Imagem 2D ou 3D)

A visão humana é capaz de captar a luz e transformá-la em estímulos, para que possamos perceber cores, tons, luminosidade e profundidade. A imagem então, pode ser representada computacionalmente de forma similar a sonora, através de processamento de sinais representados bi ou tri-dimensionalmente, através de dispositivos de saídas que emitem luz, como um monitor (Martins, 2008).

Diferente de impressoras 3D ou canhões de laser, o monitor de um computador convencional ou *datashow* não conseguem por sua limitação física/tecnológica, gerar resultados tridimensionais reais. Através da computação gráfica, o computador consegue processar imagens capazes de gerar a sensação de que há um objeto modelado bi ou tri dimensionalmente.

Logo, assim como a música, um conjunto sequencial de imagens (animação ou vídeo) possui uma relação direta com o tempo, sendo possível realizar desde simples associações como cor com dados obtidos de faixas de uma determinada frequência, até associações com formas geométricas mais complexas em plano tridimensional (Krawczyk, 2012). Assim, é possível proporcionar ao espectador uma sensação de sincronia e associação da forma visual com o som escutado, em tempo real.

### 2.3. Computação (Programação)

A programação de computadores sempre esteve presente nas artes, principalmente na expressão de fatos, acontecimentos ou criação de novas ideias. Um único objeto artístico pode ser considerado um meio de armazenar e expressar um conjunto de dados (quando não compreendidos) e/ou informações (quando o consumidor compreende o contexto).

O poder de processamento de computadores atuais e o uso de sistemas digitais, propiciam o crescimento, e conseqüentemente sua aceitação para uso nas artes visuais e sonoras, sendo adotados por artistas, como uma ferramenta para a composição musical, captação, gravação, edição e síntese de sonora (Reas & Fry, 2014).

Na computação gráfica, o computador consegue gerar imagens 2D como *bitmaps* (matrizes de pixels) e vetoriais (geradas por equações matemáticas/geométricas com alfa<sup>1</sup>) e vetoriais. No formato de mapa de pixel, elas são exibidas em camadas, com sobreposições e alphas (nota de rodapé). Em imagens 3D, o computador renderiza diversas imagens 2D, capazes de simular perspectiva e profundidade (Martins, 2008). Além disso, graças a cálculos matemáticos e algébricos, o computador consegue simular ambientes que respeitam as leis da Física, utilizado principalmente em *engines* gráficas feitas para desenvolvimento de jogos, trazendo mais realidade em efeitos de movimentação da imagem/objeto (Silva, 2007).

Sendo assim, a computação é uma grande ferramenta para criação de conteúdo audiovisual e visualização musical, uma vez que, com ela, se é capaz de processar entradas sonoras, tratar os dados obtidos e, adjunto com a computação gráfica, ser capaz de gerar animações sincronizadas com o som.

### 2.4. Interseção das áreas

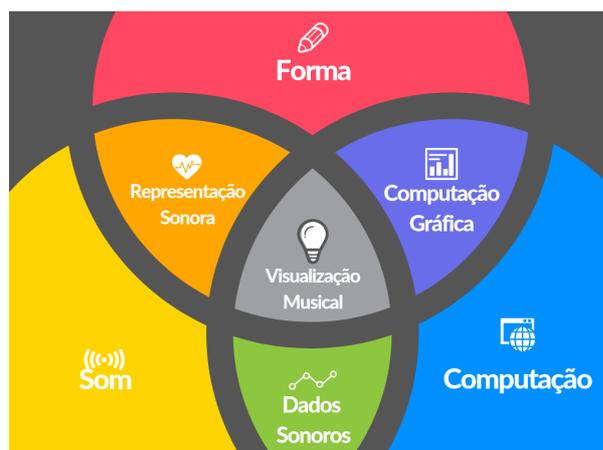
Após o estudo das áreas citadas anteriormente no desenvolvimento deste trabalho, percebe-se a relação direta que estas possuem e suas interseções;

- Som e Forma: capacidade de visualizar o som;
- Som e Lógica: capacidade de obter dados reais do som;
- Programação e Forma: capacidade de transformar dados matemáticos computacionais em formas geométricas visíveis no computador;
- Som, Forma e Programação: capacidade de gerar uma forma geométrica a partir de dados obtidos da análise sonora. A união das três gera um produto criativo e de cunho artístico, no caso a visualização musical.

A baixo é possível visualizar uma imagem simbolizando a intercessão das áreas anteriormente citadas.

---

<sup>1</sup> Alfa é um processo de combinar a imagem à frente com um fundo, trazendo a sensação de transparência (Karasinsk, 2009).



**Figura 4. Simbologia da interseção de “Som, Imagem,Computação”.**

**Fonte: O autor.**

### **3. Metodologia**

#### **3.1. Problema (Motivação)**

O problema a que este trabalho se propõe a resolver é a de suprir a geração de recursos visuais gráficos, a partir da performance musical ao-vivo e com baixo custo. Neste sentido, teve-se como meta o desenvolvimento de um software capaz de gerar efeitos visuais, em tempo real, com computadores pessoais. Utilizando linguagens de programação disponíveis no mercado, constatou-se a seguinte questão: identificar uma linguagem de programação capaz de gerar um produto de software que funcione em computadores pessoais e que fosse capaz de captar áudio em tempo real, analisar e processá-lo adequadamente. Além disso, o software deveria possuir uma boa usabilidade para o usuário final.

#### **3.2. Materiais e Métodos**

Inicialmente, optou-se por realizar uma revisão da literatura através de bases conhecidas, como Periódicos Capes e Google Acadêmico, com o intuito de obter informações relacionados a trabalhos na área, informações técnicas e buscar por linguagens, tecnologias e ferramentas de desenvolvimento. A busca foi feita utilizando os termos: “music visualization” OR “visualização musical” no título e em seu conteúdo (“graphic computer”) OR (“computação gráfica”) OR “multimídia” OR “sound processing”.

A mesma retornou 49 resultados pelo Portal de Periódicos Capes, 95 pelo Google Acadêmico. Foi realizada a leitura dos resumos dos trabalhos cujo o título tivesse relacionado ao tema. Após a leitura dos mesmos, foram selecionados 19 para uma leitura completa, sendo selecionado 7 trabalhos como material de referência e auxílio ao desenvolvimento do mesmo. Estas fontes mostraram-se mais relevantes à discussão.

Na revisão da literatura foram encontrados diversos trabalhos que realizam visualização musical, como por exemplo:

- “Exploring the Visualization of Music” (Krawczyk,2012) e “Real time music visualization: a study in the visual extension of music” (Bain, 2008), que plotam imagens (animações) baseadas em notas captadas por entrada MIDI<sup>2</sup>,
- “Visualização de música à distância de um gesto” de Gomes (2014) que processa música em tempo real e pré gravada utilizando *Three.js* e *Web Audio API*;
- “Visualização computacional de música com suporte à discriminação de elementos de teoria musical” (Catadeira, 2015) com foco em visualização de notas, também utilizando de MIDI.
- “Audible Images: um sistema para síntese de imagens controladas por áudio”, onde é desenvolvido um sistema que realiza análise de áudio via microfone, arquivo pré gravado ou MIDI, gerando acompanhamentos visuais da entrada musical.

Posteriormente ao estudo dos trabalhos encontrados, optou-se pela linguagem Processing 3, por se tratar de uma linguagem com foco para desenvolvimento de trabalhos de artes visuais e de fácil aprendizado, sendo multiplataforma (Windows, Linux e Mac OS) baseada em Java, e também por possuir bibliotecas para processamento de áudio.

A medida que foi-se elaborando o trabalho, novas pesquisas foram necessárias, além de pesquisar por termos como “Fast Fourier Transform” OR “Transformada Rápida de Fourier”, “sound analysis” OR “análise sonora”, além de buscas por “bibliotecas de processamento de som para Processing 3”.

### 3.2.1 Processing 3.0

Processing (Reas & Fry, 2014) é uma linguagem *open source*, com foco em desenvolvimento para artes visuais. Possui IDE própria, e pode ser obtida diretamente pelo site da sua mantenedora (Processing Foundation). Trata-se originalmente de uma linguagem baseada em Java, mas atualmente também possui versões baseadas em Python e Javascript. A mesma possui integração com OpenGL<sup>3</sup>. Neste trabalho optou-se por utilizar a versão em Java, por atender os requisitos levantados para o desenvolvimento do software.

O Processing 3.0 é de fácil aprendizagem, uma vez que foi criada para ensinar fundamentos de programação computacional (Reas & Fry, 2014), e tem por foco acolher pessoas da comunidade artísticas, muitas das quais, inexperientes em programação. Para programadores que possuem certa experiência com programação, principalmente com uma boa noção de conceitos básicos como tipos de variáveis, matriz, laços e controladores de fluxo, funções e orientação a objeto, possui uma curva de aprendizado interessante.

---

<sup>2</sup> MIDI é um sistema digital de comunicação de dados em tempo real, do tipo serial assíncrono, que possibilita comunicação entre instrumentos musicais, baterias eletrônicas, microcomputadores e outros equipamentos microprocessados utilizados em aplicações musicais.

<sup>3</sup> Open GL - Ambiente de desenvolvimento de aplicativos gráficos em 2D e 3D, capaz de renderizar texturas, efeitos especiais e afins, por ser multiplataforma, executa em um grande leque de sistemas operacionais e dispositivos (opengl.org, 2018).

Outro fator interessante da mesma, é o fato de possuir uma comunidade muito ativa e colaborativa, podendo facilmente obter e compartilhar códigos, ao mesmo tempo que diversos usuários auxiliam na resolução de problemas e ensinam como resolvê-los. A própria Processing Foundation, junto com seus criadores, possui material gratuito para ensino da linguagem, junto a vídeos disponibilizados no YouTube, com teorias e algoritmos básicos e conceitos avançados, além de funções específicas da linguagem.

O Processing utiliza, por padrão, duas funções para funcionamento:

- **Setup():** onde são definidas configurações gráficas e sonoras, como tamanho da janela, desenho 2D ou 3D, entradas de áudio e afins.
- **Draw():** uma função de loop contínuo, onde tudo que é escrito nele é executado várias e várias vezes em círculos lineares.

### 3.2.2. Bibliotecas de processamento de áudio

Apesar do Processing 3.0 possuir inúmeros recursos, ela possui como foco as artes visuais, havendo poucos recursos nativos para processamento sonoro. Por isto, buscou-se bibliotecas de processamento de áudio já existentes, encontrando na comunidade algumas bibliotecas gratuitas com foco no processamento de áudio.

Após a análise das bibliotecas de áudio mais usadas pela comunidade da Processing e levando em consideração o número de resultados no Google e versões estáveis, foram selecionadas as bibliotecas *Beads* e *Minim*.

- **Beads:** segundo o site da desenvolvedora (Bown, 2018) “Beads é uma biblioteca de software escrita em Java para áudio em tempo real”. É um projeto open-source e desenvolvido em conjunto com a Universidade Monash, em Melbourne. A mesma possui tanto versão para uso de java normal, podendo ser utilizada em IDEs, como Eclipse e Netbeans, como versão própria para Processing, contando inclusive no banco de bibliotecas para *download* do próprio Processing.
- **Minim:** trata-se de uma biblioteca de áudio que utiliza a “JavaSound API”, com o intuito de ser uma biblioteca de áudio de uso simplificado para pessoas que desenvolvem via Processing. Apesar desse conceito, também é capaz de prover inúmeras funções e ferramentas para usuários mais avançados (code.compartmental.net, 2018).

Ambas possuem fácil implementação, documentação bem explicada e com bons exemplos. Constatou-se que a biblioteca Beads possui um grande leque de ferramentas, com inúmeras funções específicas em relação a Minim. Contudo, a **Minim** demonstrou-se mais interessante para usuários menos técnicos com relação a propriedades técnicas sonoras, possuindo funções mais intuitivas, sendo a objetividade o principal fator para escolha da mesma.

### 3.2.3 Análise sonora

Após estarem definidas a linguagem e biblioteca de processamento de áudio, surgiu a necessidade de estudar algoritmos de análise sonora. Após um estudo mais detalhado

sobre a documentação da biblioteca Minim, encontrou-se diversas ferramentas de análise por amostragem via Transformada Rápida de Fourier (Fast Fourier Transform - FFT), descrita mais à frente.

Após uma pesquisa sobre o algoritmo de FFT, junto às ferramentas Minim baseadas na transformada, a mesma se mostrou de grande valor, uma vez que consegue retornar inúmeros dados interessantes, como frequência e amplitude, que serviram de base para o desenvolvimento dos efeitos.

### 3.2.3.1. Fast Fourier Transform (FFT)

A Transformada Rápida de Fourier (Fast Fourier Transform) trata-se de um algoritmo capaz de calcular a Transformada Discreta de Fourier, junto a sua inversa. Segundo Cooley (1969), algoritmos de FFT, por serem capazes de obter amostras de sinais, e processá-las, permitiu (principalmente para época) que computadores conseguissem economizar centenas de horas de processamento. Graças a tal feito, utilizando de FFT, é possível que o computador processe a entrada de sinais em tempo real, tornando aplicável para o trabalho.

A transformada é capaz de pegar amostragens de ondas de cada frequência, em tempo hábil para que o computador possa armazená-las ou processá-las do formato analógico para digital, sem que ocorram perdas significativas. Neste trabalho, a transformada foi utilizada para que se pudesse pegar ondas sonoras, e as convertessem para o formato digital. Após a digitalização é possível analisá-las e utilizar os dados de entrada como valores para criação de efeitos visuais. Abaixo segue a fórmula da Transformada do Somatório de Fourier, ela é utilizada no desenvolvimento do software.

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) W_N^{ux} \quad (1)$$

**Fórmula 1. Fórmula do somatório de Fourier**

## 3.3 Desenvolvimento

### 3.3.1 Entrada de som

Inicialmente, foram utilizados arquivos pré gravados em formato MP3 e WAV, para estudo e compreensão dos dados a serem processados via algoritmos de FFT. Posteriormente, após *feedback* positivo, utilizou-se de captação sonora externa em tempo real.

O Java, por padrão, utiliza o dispositivo de áudio ativo no sistema (no caso do SO Windows), o que é utilizado pelo Processing 3. Com isso, é possível reconhecer quaisquer dispositivos ou interfaces de áudio que possua drivers para o sistema, desde de que o trate como um dispositivo de entrada/saída para o SO.

Sendo assim, viabilizou-se a extração de dados via conector Line in P10/P2, P2/P2, onde a ponta P2 era ligada na entrada padrão de microfone (conector de coloração rosa por padrão) do equipamento, ou na entrada de linha (conector azul por padrão). Foi

captado entrada de som também por equipamentos USB, como interface de áudio, webcams e microfones usb.

Os algoritmos da biblioteca Minim de FFT se mostraram com tempo de resposta satisfatório para a análise do material de entrada, não havendo delay significativo ou perda de informações, quando bem configurado.

### 3.3.2 Equipamento

Como o software resultante tem por pré-definição funcionar em microcomputadores adjunto a periféricos como *datashow*, placas e equipamentos de captura de áudio, optou-se por testar a ferramenta em máquinas de baixo custo.

Computadores utilizados para teste (Modelo, processador, memória ram e placa gráfica)

- Desktop: I5-6400 (2.70Ghz), 4Gb ram ddr4, Gtx 660TI;
- Notebook HP Pavilion 550 - I7 , 8Gb ram dd3, Intel HD Graphics;
- Notebook Asus X450CA - I5, 8Gb ram dd3, Intel HD Graphics.

Dispositivos de captura de áudio

- Cabos P10/P2, P2/P2;
- Interface “Guitar Link” (USB);
- Interface “Behringer UMC22”;
- Webcam Logitech C270 (Uso do microfone, USB);
- Dispositivos de entrada de áudio nativa dos notebooks.

Todas as máquinas conseguiram executar o software de forma satisfatória e sem perdas, além do funcionamento correto de todos os dispositivos de entrada utilizado.

### 3.3.3 Desenvolvimento do Código de Análise e Efeitos

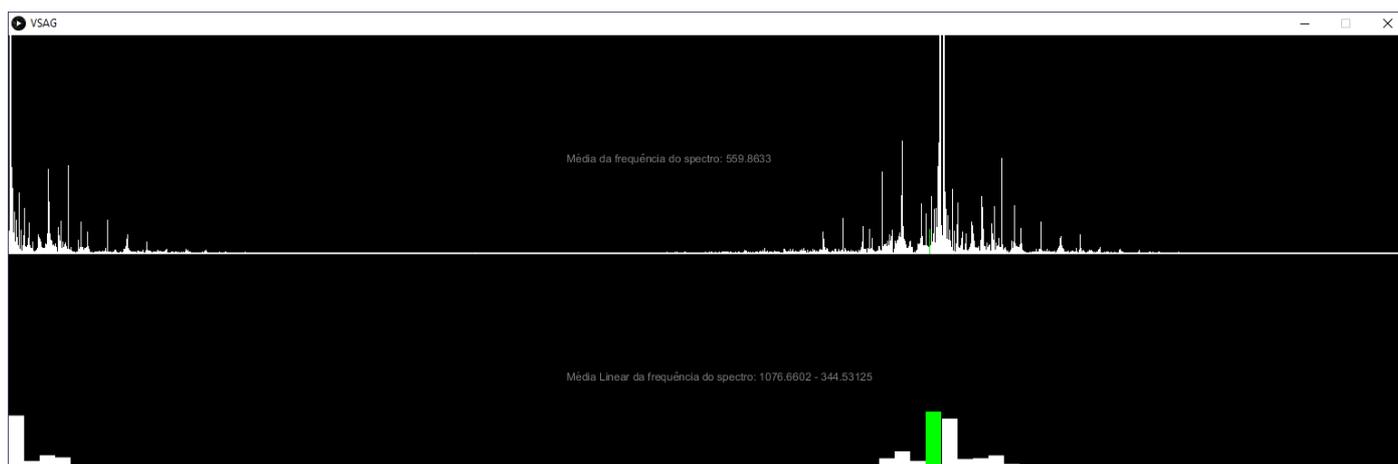
Em um primeiro momento, foi desenvolvido um código para fazer o estudo de como é feita análise do sinal de entrada pela linguagem com a biblioteca Minim para compreensão dos valores obtidos com a transformada discreta de Fourier. O mesmo teve como base de um código exemplo da própria biblioteca Minim, obtido em sua documentação.

Como padrão, o conceito de buffer nos diz que se trata de um espaço físico de uma memória que será alocado para armazenamento temporário de um determinado dado, antes de ser processado ou afim. Com isso, a biblioteca Minim armazena a entrada de áudio em um buffer, onde o áudio pode ser de padrão MONO ou STEREO, em caso de STEREO, armazena o lado direito e esquerdo.

O código realiza a leitura de um áudio em tempo real advindo de um dispositivo de entrada, ou MP3. O sinal é discretizado de 3 formas; entrada esquerda do áudio, entrada direita e uma junção de ambas (mixer), sendo exibido a amplitude em barras brancas que aumentam sobre o eixo Y e frequência estática ao eixo X. Cada discretização do áudio é demonstrado pela amplitude em cada frequência e por média da amplitude em um intervalo de frequência, que no caso é dada como divisão total da frequência. Exemplo,

uma frequência de 3000hz é dividida em 30 partes, e cada exibição mostra uma faixa de 100hz de diferença.

Uma funcionalidade foi desenvolvida para que seja exibida a frequência de uma amplitude. Esse recurso é acionado com o passar do mouse sobre a mesma. Com isso, a barra de amplitude, branca por *preset*, altera sua cor, destacando a faixa selecionada.



**Figura 5. Resultante gráfica para visualização de amplitude sobre frequência.**  
**Fonte: O autor.**

A partir disso começou o desenvolvimento de aplicações simples com efeitos baseados no tamanho da amplitude em determinadas frequências.

Um primeiro efeito foi criado com cubos, onde a frequência total é dividida em cinco partes, e cada parte com um respectivo cubo. A coloração dos cubos atenua de acordo com a amplitude, aumentando à tonalidade da cor.



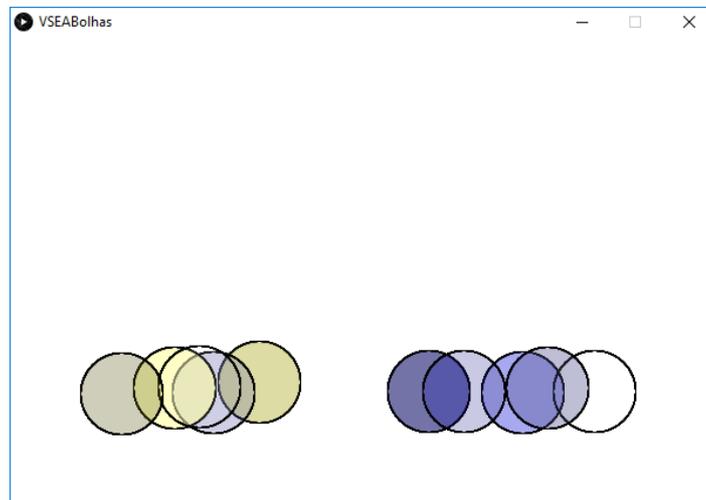
**Figura 6. Cubos que mudam de cor e tonalidade de acordo com a amplitude em cada  $\frac{1}{5}$  da frequência.**  
**Fonte: O autor.**

Segundo Frederic Patin (2003), a habilidade auditiva humana consegue determinar o ritmo musical através da detecção de uma sucessão de batidas “pseudo-periódicas”, ou seja, após determinados períodos de sons monótonos (sinais elétricos

contínuos), quando uma grande quantidade de energia (em relação a monótona) for transportada e interpretada, nosso cérebro compreende a variação elétrica (pico) como uma “batida”.

Em linhas gerais, o cérebro humano, assim como o computador consegue compreender variações elétricas, e transforma o pulso de uma variação superior em batida. Com isso, após compreender a linearidade da entrada sonora, ou seu crescimento regular, qualquer variação drástica positiva, pode ser interpretada como uma batida. A detecção de batidas pode ser utilizada para criação de efeitos com o intuito de associar a batida escutada, com movimentações visuais bruscas, criando assim um efeito associativo entre visual e sonoro.

Seguindo o conceito de batidas, criou-se um efeito onde “bolhas” (elipses em alpha) se movimentavam em torno do eixo Y quando uma batida fosse detectada (para cima), além de ampliar o raio da elipse e alterar a cor, enquanto oscila no eixo X de forma aleatória..



**Figura 7. Bolhas que sobem de acordo com batidas detectadas.**  
**Fonte: O autor.**

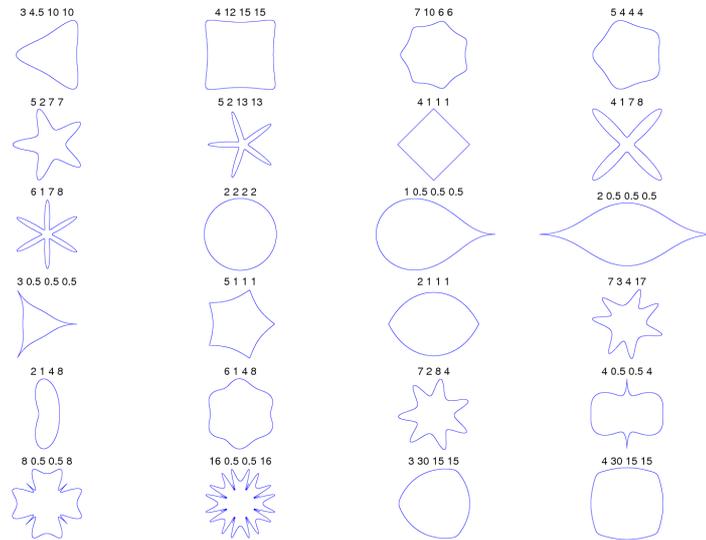
A super fórmula (Gielis, Joham. 2003) é uma equação matemática genérica capaz de descrever diversas formas complexas com múltiplas arestas e curvas que podem ser encontradas na natureza. A fórmula pode ser representada pela imagem a seguir.

$$\left( \left| \frac{\cos\left(\frac{m_1\varphi}{4}\right)}{a} \right|^{n_2} + \left| \frac{\sin\left(\frac{m_2\varphi}{4}\right)}{b} \right|^{n_3} \right)^{-\frac{1}{n_1}} \quad (2)$$

**Formula 2. Formula da Super Fórmula.**

Na fórmula, “a” e “b” tratam do tamanho da forma geométrica a ser gerada, posição dos vértices e curvatura entre eles, chamados de espinhos, “m” (“m1”=”m2”) a quantidade de vértices da forma, “n1” altera a suavidade entre os “espinhos”, “n2” e “n3” determinam a forma dos espinhos (Miller, Alexander. 2016). Alguns dos efeitos gerados

pela formula podem ser vistos na imagem a baixo. Os valores sobre as formas são respectivamente sobre “m”, ”n1”, ”n2”, ”n3”



**Figura 8. Plots da Super Fórmula com códigos respectivos a “m”, ”n1”, ”n2”, ”n3”.  
Fonte: Tiago Charters de Azevedo - Own work, CC BY 3.0.**

Utilizando da Super Fórmula, foi desenvolvido um código onde “m” pode ser alterado em múltiplo de dois com o clique do mouse (esquerdo para somar, direito para subtrair), em “n2” e “n3” é passado como valor a amplitude das dez primeiras frequências. É possível também alterar as cores pressionando teclas, como “r” para vermelho e “p” para lilaz. O valor tonal das cores varia de acordo com a amplitude da frequência em questão.

Trecho do código em Processing responsável por plotar a forma gerada pela Super Fórmula:

```
//desenhando a super fórmula
beginShape(POINTS); //ponits para criar efeito pontilhados
for (int i=0; i<10; i++) //percorrendo as 10 primeiras frequências {
  cor(i);
  for (float theta = 0; theta <= 2 * PI; theta +=0.01) {
    float rad = sFormula(theta,
    2, //a - X
    2, //b - Y
    m, //m - pontos de divisão do polígono
    1, //n1 suavidade das pontas
    sin(fft.getBand(i)/2)*0.5+0.5, //n2
    cos(fft.getBand(i)/2)*0.5+0.5 //n3
    );
    float x = rad * cos (theta)*50;
    float y = rad * sin (theta)*50;
    vertex (x, y); //desenhando
  }
}
```

```

}
endShape();
}
//superfórmula
float sFormula (float theta, float a, float b, float m, float n1, float n2, float n3) {
return pow(pow(abs(cos(m * theta/4.0)/a), n2) + pow(abs(sin(m * theta/4.0)/b), n3), -
1.0 / n1);}

```

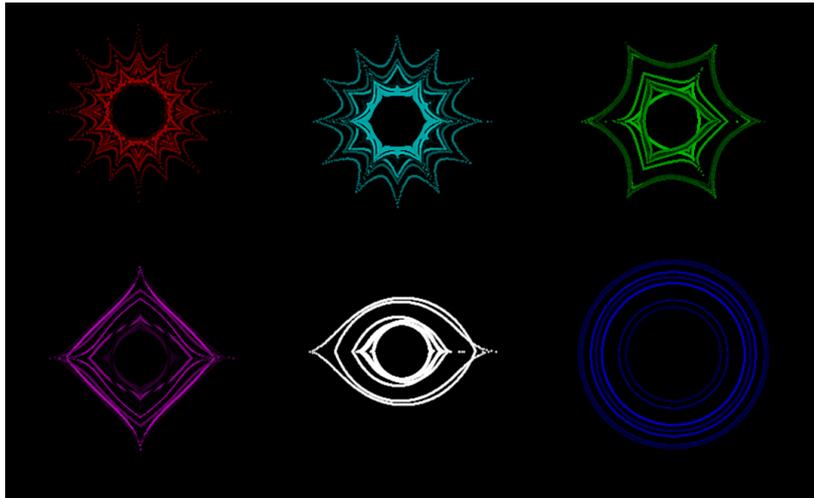


Figura 9. Efeitos gerados pelo software utilizando a Super Fórmula.

Fonte: O Autor

#### 4. Resultados

A transformada rápida de Fourier mostrou-se de grande valor para a visualização musical. Com ela foi possível extrair os dados necessários a partir de uma amostragem do som em tempo hábil para trabalhar com as informações geradas, tanto para entrada com dispositivos analógicos, como entrada de microfone p2 nativa dos computadores ou digitais via entrada usb, como microfones e interfaces de áudio.

Durante o desenvolvimento do código foram feitos vários testes para avaliar o tempo de resposta dos efeitos em relação a entrada do som (delay), uma vez que, resposta em tempo real é de grande importância para gerar a sensação de sincronia e associação imagem e som. Os testes apresentaram resultados satisfatório, com *delays* aceitáveis na maioria dos casos, exceto em dois, citados a seguir.

O primeiro caso é referente ao código desenvolvido com a Super Fórmula, onde acima de 50 formas a serem projetadas simultaneamente de um mesmo objeto, em uma das máquinas testadas, apresentou grande delay de processamento, tornando inviável o uso com esse valor ou superior. Optou-se por 10 formas simultâneas, por já apresentar um efeito interessante e satisfatório (utilizado na Figura 9). O segundo caso de delay registrado foi referente a saída de áudio. Ao ativar a saída de áudio para escutar o retorno do som de entrada, um delay acima de um segundo ocorria, tornando inviável a utilização da máquina como um equipamento de saída de áudio em simultâneo com a imagem.

Foram elaborados protótipos de efeitos em um aspecto bidimensional, com funções simples de serem alteradas, personalizadas e adaptadas às necessidades,

conforme apresentado no item 3.3.3. Todas com um tempo de resposta satisfatório e testadas nos dispositivos citados neste trabalho, sendo aptas para o uso.

## **5. Conclusão**

O objetivo principal proposto neste trabalho de construir efeitos visuais em tempo real gerados a partir de dados obtidos de uma entrada sonora, para apresentações modestas ao vivo foi alcançado. Com isso, se tornou visível a sincronia entre som e imagem (animação), mostrando como a tecnologia pode ser uma saída interessante para criação artística, uma vez que as ferramentas ampliam o leque de possibilidades para arte.

Para isso, foi necessário uma pesquisa e análise de ferramentas para o desenvolvimento da mesma. Com esse estudo, a linguagem Processing com a biblioteca Minim demonstrou-se uma ótima ferramenta para desenvolvimento, suprimindo as necessidades encontradas, sendo realmente de fácil aprendizagem, e capaz de fazer um levantamento de dados sonoros em tempo hábil para criação dos efeitos, principalmente com o algoritmo de FFT para leitura do som e formas geradas pela Super Fórmula.

Ao visualizar todo o processo de análise sonora até a criação artística, foi possível ver a complexidade em realizar a leitura de um sinal analógico e transformá-lo em digital, além da complexidade criativa em juntar computação gráfica e a matemática com os dados obtidos para geração dos efeitos.

Com isso, a experiência e informações obtidas no desenvolvimento deste trabalho, contribuiu consideravelmente para ampliar o conhecimento adquirido ao longo do curso de Bacharelado de Sistemas de Informação, além de compreender melhor o âmbito artístico-digital e análise de sinais, sendo possível visualizar a dificuldade para criação dos efeitos, que demonstra o porquê de baixa utilização por pequenos e médios artistas.

Outro ponto que merece destaque, é o fato de se tratar de um estudo direcionado da linguagem Processing 3, junto com a biblioteca Minim. Que por sua vez, demonstraram-se eficientes no processamento, atendendo aos requisitos inicialmente levantados...

Os códigos fonte desenvolvidos neste trabalho<sup>4</sup> podem ser obtidos na plataforma de hospedagem “Github”.

## **6. Trabalhos Futuros**

Para trabalhos futuros é possível sugerir mudanças capazes de agregar valor computacional e artístico ao projeto. Atualmente somente efeitos bidimensionais simples foram criados, sendo assim, a criação de efeitos mais elaborados e tridimensionais seriam de grande valor artístico e visual para as performances musicais ao-vivo.

Outra sugestão seria a criação de uma interface onde fosse possível importar efeitos previamente criados, inclusive de outros artistas/programadores. O usuário

---

<sup>4</sup> Código fonte disponível em: <[https://github.com/ArtFicer/TCC\\_IFSudesteMG\\_PedroV](https://github.com/ArtFicer/TCC_IFSudesteMG_PedroV)>

poderia selecionar quais gostaria de utilizar, sendo controladas por tempo ou botão para alteração entre elas, uma vez que as aqui desenvolvidas rodam em softwares individuais.

Por fim, em nível técnico, apesar do algoritmo da Transformada Rápida de Fourier suprir as necessidades aqui apresentadas, tendo por esse motivo sido escolhido para o desenvolvimento, durante o estudo de bibliografias, foram citados em outros métodos/algoritmos de análise sonora em frequência e tempo, sendo um deles a Transformada Wavelet, sendo muito utilizada na eliminação de ruídos, compressão de dados, identificação de singularidades, separação de componentes no sinal, entre outros. A literatura demonstra uma boa contribuição do Wavelets na área, sendo interessante um estudo futuro para a aplicação do mesmo na geração de efeitos gráficos, a partir do áudio executado em tempo real.

## 7. Referências

- AZEVEDO, T. C. D. 2D superformula. Wikipedia. Disponível em: <<https://commons.wikimedia.org/wiki/File:Sf2d.png>>. Acesso em: 03 Novembro 2018.
- BAIN, M. N. Real time music visualization: a study in the visual extension of music. The Ohio State University, 2008.
- BOWN, O. Beads. Beads. Disponível em: <<http://www.beadsproject.net/>>. Acesso em: 11 Agosto 2018.
- CANTAREIRA, G. D. Visualização computacional de música com suporte à discriminação de elementos de teoria musical. Dissertação de Mestrado para Título em Mestre em Ciências de Computação e Matemática Computacional, São Carlos, São Paulo, Brasil, 2015.
- COOLEY, J. W. The Fast Fourier Transform and Its Applications. IEEE TRANSACTIONS ON EDUCATION, VOL. 12, NO. 1, 1969.
- FERNANDES, J. C. Acústica e Ruídos. Bauru – SP: UNESP - Câmpus de Bauru - Faculdade de Engenharia, 2002.
- FILHO, W. D. P. Multimídia: Conceitos e Aplicações. 2ª. ed. Rio de Janeiro, Brasil: LTC - Livros Técnicos e Científicos Editora Ltda, 2013.
- FRY, B.; REAS, C. Overview. Processing. Disponível em: <<https://processing.org/overview/>>. Acesso em: 11 Agosto 2018.
- GIELIS, J. A GENERIC GEOMETRIC TRANSFORMATION THAT UNIFIES A WIDE RANGE OF NATURAL AND ABSTRACT SHAPES. American Journal of Botany 90(3): 333–338, Antwerp, Belgium, 2003.
- GitHub. (26 de Novembro de 2018). About GitHub. Fonte: GitHub: <https://github.com/about>
- GOMES, J. T. S. Visualização de música à distância de um gesto. UNIVERSIDADE DE LISBOA Faculdade de Ciências Departamento de Informática defesa de MESTRADO EM INFORMÁTICA, Lisboa, Portugal, 2014.
- KARASINSKI, E. O que é o Canal Alfa e para que ele serve? Tecmundo, 2009. Disponível em: <<https://www.tecmundo.com.br/video-game-e-jogos/1358-o-que-e-o-canal-alfa-e-para-que-ele-serve-.htm>>. Acesso em: 2018 Novembro 05.
- KENMOCHI, H. VOCALOID and Hatsune Miku phenomenon in Japan. InterSinging 2010 - First Interdisciplinary Workshop on Singing Voice, Tokyo, Japan, 01-02 Outubro 2010.
- KRAWCZYK, R. J. Exploring the Visualization of Music. College of Architecture, Illinois Institute of Technology, Chicago, USA, 2012.

- MACHADO, N. Características de um Som. Disponível em: <[http://www.aulas-fisica-quimica.com/8f\\_06.html#](http://www.aulas-fisica-quimica.com/8f_06.html#)>. Acesso em: 11 Fevereiro 2018.
- MADEIRO, F. Introdução à Compressão de Sinais. REVISTA DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO, NÚMERO 1, Recife, PE, Outubro 2011.
- MARTINS, M. Z. Audible Images: um sistema para síntese de imagens controladas por áudio. DISSERTAÇÃO APRESENTADA AO INSTITUTO DE MATEMÁTICA E ESTATÍSTICA DA UNIVERSIDADE DE SÃO PAULO PARA OBTENÇÃO DO TÍTULO DE MESTRE EM CIÊNCIAS, São Paulo, Janeiro 2008.
- MCLEOD, P.; WYVILL, G. Visualization of Musical Pitch. Department of Computer Science, University of Otago, New Zealand, 2003.
- NASCIMENTO, M. O que é MP3, WMA, WAV, AIFF, MIDI – O formato de arquivos de áudio. Gravando em Casa, 2010. Disponível em: <<http://gravandoemcasa.com/2010/03/o-que-e-mp3-wma-wav-aiff-midi-arquivo-audio/>>. Acesso em: 2018 Novembro 05.
- NBC NEWS. Lady Gaga is Going on Tour with a Hologram. NBC News, 2014. Disponível em: <<https://www.nbcnews.com/tech/innovation/lady-gaga-going-tour-hologram-n83406>>. Acesso em: 01 Novembro 2018.
- PATIN, F. BEAT DETECTION ALGORITHMS, 2003.
- PHONICAL, C. B.-S. 4. 0. FFT-Time-Frequency-View. Wikipedia, 2017. Disponível em: <<https://commons.wikimedia.org/w/index.php?curid=64473578colocar>>. Acesso em: 31 Outubro 2018.
- QUARTZ, D. Minim. Codde Compartmental. Disponível em: <<http://code.compartmental.net/tools/minim/>>. Acesso em: 08 Agosto 2018.
- REAS, C.; BEN, F. Processing A Programming Handbook for Visual Designers and Artists. 2ª. ed. Cambridge, Massachusetts: The MIT Press, 2014.
- SILVA, R. E. D. ENSINO DE CIÊNCIA DA COMPUTAÇÃO ATRAVÉS DO DESENVOLVIMENTO DE JOGOS. VII Congresso Iberoamericano de Informática Educativa, Joinville/SC, 2007.