Visualização de similaridades em bases de dados de música

Jorge Henrique Piazentin Ono

Visualização de similaridades em bases de dados de música

Jorge Henrique Piazentin Ono

Orientador: ***Prof. Dr. Luis Gustavo Nonato***

Monografia apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, para o Exame de Qualificação, como parte dos requisitos para obtenção do título de Mestre em Ciências - Ciências de Computação e Matemática Computacional.

USP – São Carlos
Fevereiro de 2014

Resumo

Coleções de músicas estão amplamente disponíveis na internet e, graças ao crescimento na capacidade de armazenamento e velocidade de transmissão de dados, usuários podem ter acesso a uma quantidade ilimitada de composições. Isso levou a uma maior necessidade de organizar, recuperar e processar esse tipo de dado de modo automático. Visualização de informação é uma área de pesquisa que permite a análise de grandes conjuntos de dados e, por isso, é uma ferramenta muito valiosa para a exploração de bibliotecas musicais. O objetivo deste projeto de mestrado é desenvolver novas técnicas visuais para a análise de grandes coleções de áudio. Neste documento, diversos mecanismos de extração de características e visualização de músicas são descritos e a metodologia para o desenvolvimento deste projeto é proposta.

Abstract

Music collections are widely available on the internet and, leveraged by the increasing storage and bandwidth capability, users can currently access a multitude of songs. This leads to a growing demand towards automated methods for organizing, retrieving and processing this kind of data. Information visualization is a research area that allows the analysis of large data sets, thus, it is a valuable tool for the exploration of music libraries. The goal of this master's project is to develop novel visual techniques for the analysis of large audio collections. In this document, several music feature extraction and visualization techniques are described and a methodology for the development of this project is proposed.

Sumário

1 Introdução **1**
1.1 Contexto e motivação 1
1.2 Organização da monografia 2

2 Extração de características de músicas **3**
2.1 Pré-processamento 4
2.1.1 Redução de canais de áudio 4
2.1.2 Remoção CC 4
2.1.3 Normalização 5
2.1.4 Redução da taxa de amostragem 5
2.1.5 Quantização 6
2.2 Análise espectral 6
2.3 Características de forma espectral 8
2.3.1 Centroide espectral 8
2.3.2 Atenuação espectral 8
2.3.3 Fluxo espectral 9
2.3.4 *Mel-Frequency Cepstral Coefficients* 9
2.4 Análise tonal 11
2.4.1 Percepção de *chroma* 11
2.4.2 *Pitch Class Profile* 12
2.4.3 *Harmonic Pitch Class Profile* 13
2.5 Métricas de similaridade 14
2.6 Considerações finais 18

3 Visualização de músicas **19**
3.1 Visualização de música individual 19
3.2 Visualização de coleções de músicas 23
3.3 Considerações finais 29

4 Proposta de trabalho **31**
4.1 Considerações iniciais 31
4.2 Metodologia 32
4.2.1 Extração de características 32
4.2.2 Visualização de coleções de músicas com características de *chroma* . 32
4.2.3 Visualização de similaridade entre músicas 35
4.3 Resultados preliminares 41
4.4 Atividades e cronograma previsto 43

Lista de Figuras

2.1 Processamento do sinal em janelas de tamanho $\mathcal{H}$ e intervalo $\mathcal{K}$, adaptado de (Lerch, 2012). 7
2.2 Espectrograma da música Abracadabra (Base Covers80 (Ellis, 2007)) . . . 7
2.3 Banco de filtros na escala normal e na escala Mel. Figura adaptada de (Han et al., 2006) 10
2.4 MFCC da música Abracadabra, interpretada por Steve Miller Band (base de dados Covers80 (Ellis, 2007)). 10
2.5 Uma oitava do teclado de um piano (Lerch, 2012). 11
2.6 Visualização da percepção de tom. Os eixos X e Y simbolizam as *chromas* e o eixo Z, as frequências. Figura adaptada de (Lerch, 2012). 12
2.7 HPCP da música Abracadabra, interpretada por Steve Miller Band (base de dados Covers80 (Ellis, 2007)). 14
2.8 Execução do algoritmo de DTW. (A) Séries C e Q. (B) Matriz de distâncias. (c) Alinhamento das duas séries com o caminho mínimo (Keogh et al., 2005). 15
2.9 *Cross Recurrence Plot* da música "*Day Tripper*"da banda The Beatles com (a) uma música cover interpretada por Ocean Colour Scene e (b) "*I've got a crush on you*" por Frank Sinatra. Parâmetros são $m = 9$ e $\tau = 1$. Figura adaptada de (Serrà et al., 2009). 16
2.10 Matriz Q da música "Daytripper" interpretada por The Beatles e Ocean Colour Scene. $\gamma_0 = 3$ e $\gamma_e = 7$ (Serrà et al., 2009). 17

3.1 *Piano roll* da música "*With You Friends*", de Skrillex 20
3.2 Visualização gerada pela *Music Annotation Machine* para a música *The Rite of Spring*, de Stravinsky. 20

3.3 Partitura condensada da composição "Clarinet Quintet A-major, K.V. 581", por W. A. Mozart (Hiraga et al., 2002). 21
3.4 Shape of Song da música *All The Small Things*, por Blink 182 (Wattenberg, 2002). 22
3.5 Visualização *Infinite Jukebox* da música *Lights*, por Ellie Goulding (Lamere, 2012). 22
3.6 Exploração de coleções de músicas com metadados (Torrens et al., 2004). . 24
3.7 Base de músicas representada com *MusicNodes* (Dalhuijsen et al., 2010). . 25
3.8 Representação de um conjunto de músicas com *Islands of Music* (Pampalk, 2001). 26
3.9 Exemplo de execução da ferramenta *MusicBox* (Lillie, 2008). 27
3.10 Organização de uma base de dados da banda The Beatles (Muelder et al., 2010). 28
3.11 Sistema de criação de Playlists com a projeção multidimensional PLP (Paulovich et al., 2011). 29

4.1 *Pipeline* da visualização de base de músicas proposta. 35
4.2 CRP de "*Daytripper*", interpretada por Beatles e Cheap Trick (Base de dados Covers80 (Ellis, 2007)). 36
4.3 Filtragem das diagonais mais longas da música "*Daytripper*", interpretada por Beatles e Cheap Trick (Base de dados Covers80 (Ellis, 2007)). 37
4.4 Transformação da matriz CRP em uma matriz de adjacência 37
4.5 Visualização do CRP da música "*Daytripper*", interpretada por The Beatles e Cheap Trick (Base de dados Covers80 (Ellis, 2007)). Limiar = 75 janelas e elemento estruturante quadrado de tamanho 7. 38
4.6 Representação em regiões de similaridade da música "*Daytripper*", interpretada por Beatles e Cheap Trick (Base de dados Covers80 (Ellis, 2007)). 39
4.7 Visualização de Cordas do relacionamento entre um grupo de entidades (Bostock, 2012). 40
4.8 Representação em regiões com *layout* circular da música "*Daytripper*", interpretada por Beatles e Cheap Trick (Base de dados Covers80 (Ellis, 2007)). 40
4.9 Projeção da base de dados DB50 com *Classical Multidimensional Scaling*. . 41
4.10 Representação em regiões com *layout* circular da música "Abracadabra", interpretada por Steve Miller Band e Sugar Ray (Base de dados Covers80 (Ellis, 2007)). 42

4.11 Representação em regiões com *layout* circular da música “Abracadabra”, Sugar Ray, com “*Daytripper*”, The Beatles (Base de dados Covers80 (Ellis, 2007)). 43

Lista de Siglas

CC	Corrente Contínua
CRP	*Cross Recurrence Plot*
DFT	*Discrete Fourier Transform*
DTW	*Dynamic Time Warping*
HPCP	*Harmonic Pitch Class Profile*
LAMP	*Local Affine Multidimensional Projection*
MAP	*Mean of Average Precisions*
MDS	*Multidimensional Scaling*
MFCC	*Mel-Frequency Cepstral Coefficients*
MIDI	*Musical Instrument Digital Interface*
MIR	*Music Information Retrieval*
MIREX	*Music Information Retrieval Evaluation eXchange*
PCA	*Principal Component Analysis*
PCP	*Pitch Class Profile*
RQA	*Recurrence Quantification Analysis*
SOM	*Self-Organizing Map*
STFT	*Short Time Fourier Transform*
SVM	*Support Vector Machine*

Capítulo

1

Introdução

O grupo de Processamento Visual e Geométrico do ICMC-USP tem desenvolvido um conjunto de ferramentas para visualização e exploração de diferentes tipos de bases de dados. Entretanto, no caso de base de dados de música, as ferramentas existentes são pouco eficientes, acarretando em várias limitações. O projeto de mestrado aqui proposto visa implementar descritores de sinal próprios para a análise de composições musicais, que darão subsídio para a concepção de uma nova ferramenta de exploração visual de bases de dados de música, disponibilizando recursos de navegação, organização e mineração visual, os quais são difíceis de serem encontradas nos *softwares* existentes.

1.1 Contexto e motivação

Visualização computacional vem deixando o papel de coadjuvante e surge como área de pesquisa básica onde são desenvolvidas metodologias integradas para análise, interação e mineração de grandes conjuntos de dados multimodais e distribuídos. Tal fato pode ser comprovado tomando como base os avanços alcançados pelas ferramentas de visualização que estão sendo empregadas na análise e exploração de grandes coleções de documentos (Cui et al., 2011) e na representação visual de redes dinâmicas (Hadlak et al., 2011; Withall et al., 2007). Existem setores, porém, onde as ferramentas de visualização ainda não alcançaram um nível de desenvolvimento satisfatório, como é o caso da exploração visual de grandes bases de dados de música.

Coleções de músicas estão amplamente disponíveis na internet. Em 6 de fevereiro de 2013, a Apple anunciou que seus usuários já compraram vinte e cinco bilhões de músicas da iTunes Store (Rodriguez, 2013). A crescente capacidade de armazenamento e transmissão de músicas criou uma dificuldade na identificação e exploração de bases de dados musicais. Visando a criação de um novo paradigma para exploração visual desses conjuntos de dados, o grupo de processamento visual e geométrico do ICMC-USP desenvolveu recentemente um protótipo de sistema que permite ao usuário manipular visualmente conjuntos de músicas a fim de criar *playlists* e classificar, interativamente, um determinado conjunto de músicas (Paulovich et al., 2011).

A principal contribuição deste trabalho é a criação de novos paradigmas de visualização de dados de música no contexto de exploração e mineração visual de grandes bases de dados. Ao contrário de trabalhos anteriores, que utilizam descritores desenvolvidos para o reconhecimento de voz, serão utilizadas características específicas para o domínio de aplicação de música, especificamente, características de timbre, ritmo e *chroma*.

1.2 Organização da monografia

O restante desta monografia está estruturada da seguinte maneira:

- No Capítulo 2, serão descritos os principais algoritmos relacionados à análise de conteúdo de áudio: pré-processamento do sinal, extração de características e métricas de similaridade de músicas;
- No Capítulo 3, os principais trabalhos relacionados à visualização de músicas serão apresentados;
- Por fim, no Capítulo 4, discute-se com mais detalhes a proposta deste trabalho e a metodologia a ser adotada.

Capítulo

2

Extração de características de músicas

Music Information Retrieval (MIR) é uma linha de pesquisa interdisciplinar que trata da descrição automática, entendimento, pesquisa, recuperação e organização de conteúdos musicais (Orio, 2006). Ela é uma área geral, que abrange o processamento de sinais de áudio (baseado em conteúdo), a análise de formatos de música simbólicos, como o MIDI (*Musical Instrument Digital Interface*), e metadados, por exemplo, o título, a letra e o compositor de uma música (Lerch, 2012).

O MIR baseado em conteúdo está diretamente relacionado com a análise de conteúdo de áudio, uma área mais específica que trata da extração de informações de sinais de áudio, por exemplo, gravações de músicas armazenadas em formato digital. A base de qualquer tipo de sistema de análise de áudio é a extração de características, ou seja, o cálculo de uma representação numérica compacta que pode representar um segmento de áudio (Tzanetakis et al., 2002).

Arquivos MIDI são essencialmente similares à partituras, descrevendo o início, duração, volume e instrumento de cada nota na composição (Tzanetakis et al., 2003). Entretanto, a maioria das músicas disponíveis ao público estão no formato de sinal de áudio e, mesmo quando metadados descrevendo autor, gênero e álbum estão disponíveis, não há garantias de que são condizentes com a música. Por isso, a fundamentação de extração de características é uma etapa importante para o desenvolvimento de sistemas MIR. Neste capítulo, os principais algoritmos de extração de características de áudio serão apresentados e discutidos.

2.1 Pré-processamento

De acordo com Lerch (2012), o sinal de áudio é frequentemente pré-processado antes da etapa de extração de características, reduzindo-se assim a quantidade de dados a serem analisados. Os algoritmos para pré-processamento podem ser agrupados em duas categorias:

- Algoritmos para tempo real, em que são conhecidas apenas as amostras atuais e as anteriores do sinal;
- Algoritmos *offline*, em que todas as amostras são conhecidas no tempo do processamento.

A seguir serão apresentados brevemente os principais algoritmos de pré-processamento.

2.1.1 Redução de canais de áudio

Com a popularização dos equipamentos eletrônicos com suporte à áudio multicanais, existem situações em que o número de canais do arquivo ou das caixas de som são limitados. Quando isso ocorre, é necessário aumentar (*up-mixing*) ou reduzir (*down-mixing*) o número de canais para se adequar ao agente limitante. Um estudo detalhado sobre essas operações pode ser encontrado em Bai et al. (2007).

No contexto da análise de conteúdo de áudio, muitas vezes a informação de interesse pode ser representada por um único canal. A forma mais simples de *down-mixing* é a média aritmética dos canais, definida pela equação 2.1, em que C é o número de canais de áudio. Também é possível utilizar uma média ponderada dos sinais, dando, por exemplo, pesos maiores a canais *surround* [1] e pesos menores a canais frontais (Lerch, 2012).

$$x(i) = \frac{1}{C} \sum_{c=0}^{C-1} x_c(i) \tag{2.1}$$

2.1.2 Remoção CC

Um deslocamento de corrente contínua (CC), mais conhecido como *DC offset*, ocorre quando a média aritmética do sinal é muito diferente de zero, o que pode ser prejudicial

[1] Sistema em que mais de dois canais de áudio são gravados. Alto-falantes são posicionados ao redor do ouvinte para melhorar a percepção do som. O cinema é um exemplo de sistema *surround*, em que são utilizados o sinal estereo (esquerdo L e direito R), um canal do meio M e dois sinais adicionais, L_B e R_B (Zolzer, 2008).

na etapa de extração de características (Lerch, 2012). Deseja-se, portanto, remover esse deslocamento do sinal.

Se o processamento *offline* é viável, subtrai-se a média aritmética do sinal X_{CC} de tamanho X de todas as amostras, ilustrado na equação 2.2 (Lerch, 2012).

$$x(i) = x_{CC}(i) - \frac{1}{X}\sum_{i=0}^{X-1} x_{CC}(i) \tag{2.2}$$

Em um sistema de tempo real, não é possível calcular a média aritmética de todo o sinal de áudio. Uma maneira de remover o deslocamento CC é por meio de um filtro passa alta, por exemplo, o descrito na equação 2.3, em que α é o parâmetro do filtro passa-baixas que atenua o impacto do diferenciador nas frequências mais altas.

$$x(i) = (1-\alpha).(x_{CC}(i) - x_{CC}(i-1)) + \alpha.x(i-1) \tag{2.3}$$

2.1.3 Normalização

A normalização de um sinal *offline* é realizada dividindo-se todas as amostras pelo módulo da maior amostra no sinal (ver equação 2.4, onde x_s é o sinal original) (Lerch, 2012).

$$x(i) = \frac{x_s(i)}{\max\limits_{\forall}(|x_s(i)|)} \tag{2.4}$$

De acordo com Lerch (2012), a normalização de sinais em tempo real é uma tarefa difícil. Pode-se utilizar algoritmos de controle de ganho automático, ou compressores e limitadores, monitorando características instantâneas do sinal.

2.1.4 Redução da taxa de amostragem

Taxas de amostragens comumente encontradas em sinais de áudio e voz são: 8kHz para telefonia, 32kHz para rádio digital e 44.1kHz para gravações de CD (Müller, 2007). O *down-sampling* modifica a quantidade de amostras de f_S para uma taxa mais baixa f_d em um fator l. A taxa de amostragem f_d será igual a $\frac{f_S}{l}$. Uma maneira simples de realizar o *down-sampling* é mostrado na equação 2.5.

$$x_d(i) = x(l \cdot i) \tag{2.5}$$

2.1.5 Quantização

Durante a amostragem do áudio, as amplitudes do sinal são quantizadas, ou seja, arredondadas para valores de amplitude pré-definidos. Usualmente, a quantidade de amplitudes que podem ser representadas ($\mathcal{M}$) é uma potência de 2, portanto representa-se computacionalmente o valor por meio de um código binário de tamanho w, calculado pela equação 2.6 (Lerch, 2012).

$$w = \log_2(\mathcal{M}) \tag{2.6}$$

Valores comuns para w são 16, 24 e 32 bits.

2.2 Análise espectral

De acordo com Chen et al. (2010), a forma de onda de uma música no domínio do tempo não apresenta muitas informações sobre o seu conteúdo. A distribuição de frequências de um sinal carrega informações mais relevantes e pode ser calculada por meio da análise do seu espectro de frequências. Para uma sequência $\{x(n), x \in \mathbb{R}, n \in \mathbb{Z}\}$, a transformada de Fourier decompõe o sinal em uma soma ponderada de componentes senoidais, dados pela equação 2.7.

$$X(\omega) = \sum_{n=-\inf}^{\inf} x(n)e^{-i\omega n} \tag{2.7}$$

Uma variante da transformada de Fourier muito utilizada para capturar transições em música é a *Short Time Fourier Transform* (STFT), uma transformada tempo-frequência. A STFT realiza a transformada de Fourier em pequenas subseções do sinal e, nessas subseções, considera-se que ele é estacionário (pode ser caracterizado por sua média e desvio padrão). Divide-se a música em quadros sequenciais (possivelmente com sobreposição), multiplica-se cada quadro por uma função *window* e analisa-se o sinal resultante com a transformada de Fourier. Desta forma, obtêm-se um gráfico chamado espectrograma, cujos eixos representam *tempo* x *frequência*, e a cor, a magnitude das frequências. A transformada é dada pela equação 2.8, em que k representa os índices da frequência, l representa o tempo e w é a função de janelamento utilizada (Chen et al., 2010).

$$x(k,l) = \sum_{m=0}^{N-1} w(l-m)x(m)e^{-j\omega_k m}, l = 0, 1, ... \tag{2.8}$$

O processamento do sinal é, portanto, baseado em blocos. Divide-se o sinal em janelas consecutivas de tamanho $\mathcal{K}$ a uma distância $\mathcal{H}$ do começo de um bloco até o próximo bloco. A figura 2.1 ilustra esse processo.

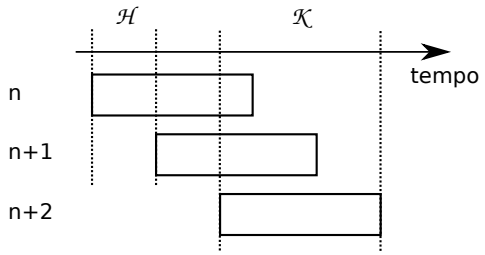


Figura 2.1: Processamento do sinal em janelas de tamanho $\mathcal{H}$ e intervalo $\mathcal{K}$, adaptado de (Lerch, 2012).

A figura 2.2 apresenta o espectrograma da música Abracadabra, interpretada por Steve Miller Band (base de dados Covers80 (Ellis, 2007)), com uma $\mathcal{H} = 512$, $\mathcal{K} = 512$ e função *window Blackman-Harris*.

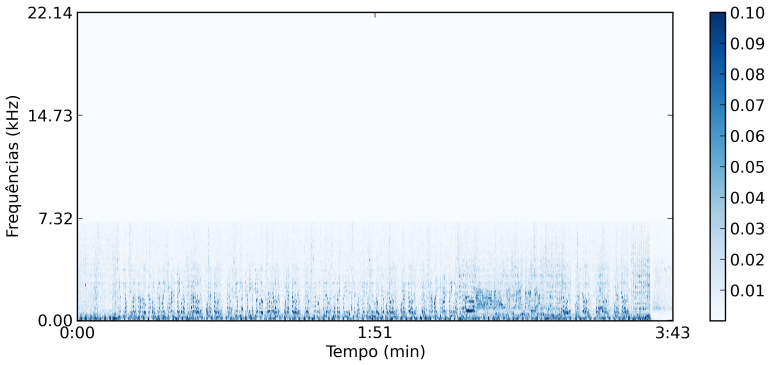


Figura 2.2: Espectrograma da música Abracadabra (Base Covers80 (Ellis, 2007))

A magnitude $|X(k)|$ e a fase $\phi(k)$ do espectro de frequências são dados pelas equações 2.9 e 2.10 (Gómez, 2006).

$$|X(k)| = \sqrt{Re(X_k)^2 + Im(X_k)^2} \tag{2.9}$$

$$\phi(X(k)) = \arctan \frac{Re(X_k)}{Im(X_k)} \tag{2.10}$$

2.3 Características de forma espectral

As características tipicamente utilizadas para representar um sinal de áudio musical estão relacionadas ao timbre do sinal. Grande parte dos algoritmos para extrair essas características são oriundas de técnicas de reconhecimento de voz. Elas são baseadas na STFT e são calculadas para cada janela de som. O timbre pode ser descrito como a "cor" do som, sua qualidade ou textura (Tzanetakis et al., 2002; Lerch, 2012).

2.3.1 Centroide espectral

Centroide espectral é uma medida que representa o centro de gravidade da magnitude do espectro da STFT. Pode ser calculado por meio da equação 2.11, onde $M_t(n)$ é a magnitude da transformada de Fourier no quadro t e frequência n (Tzanetakis et al., 2002).

$$C_t = \frac{\sum_{n=1}^{N} M_t[n] * n}{\sum_{n=1}^{N} M_t[n]} \tag{2.11}$$

O centroide é uma medida de forma espectral, em que valores mais altos correspondem à frequências maiores (Tzanetakis et al., 2002).

2.3.2 Atenuação espectral

A atenuação espectral é definida como a frequência R_t em que uma porcentagem P da distribuição de magnitude está concentrada. Pode ser calculada com a equação 2.12. Valores comuns para P são 0.85 e 0.95(Tzanetakis et al., 2002; Lerch, 2012).

$$\sum_{n=1}^{R_t} M_t[n] = P * \sum_{n=1}^{N} M_t[n] \tag{2.12}$$

A atenuação espectral pode ser normalizada, dividindo-se o valor resultante por $K/2-1$. Valores baixos indicam componentes insignificantes em frequências mais altas, portanto, baixa largura de banda (Lerch, 2012).

2.3.3 Fluxo espectral

O fluxo espectral é definido como a diferença ao quadrado entre as magnitudes normalizadas de sucessivas distribuições espectrais (equação 2.13). Essa é uma medida da quantidade de variações na forma espectral (Tzanetakis et al., 2002).

$$F_t = \sum_{n=1}^{N} (N_t[n] - N_{t-1}[n])^2 \tag{2.13}$$

Valores baixos de fluxo espectral indicam um estado estacionário no sinal ou valores de entrada muito pequenos. Há picos no fluxo espectral quando há mudança de tom ou no início de uma nova nota (Lerch, 2012).

2.3.4 Mel-Frequency Cepstral Coefficients

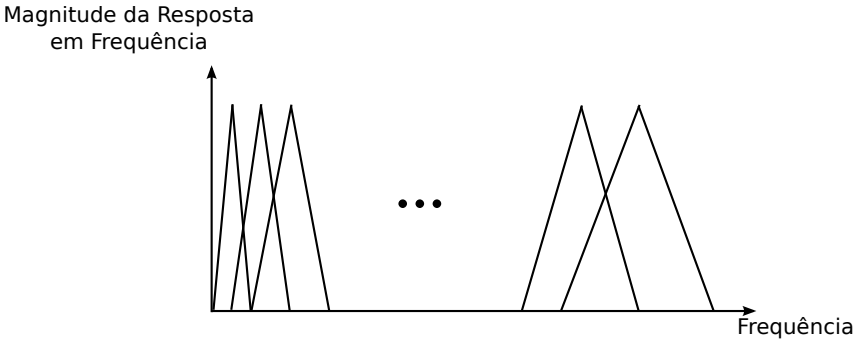
A percepção do som pelo homem é logarítmica. Essa não linearidade é atrelada à resolução de frequências perceptíveis pela cóclea humana (Lerch, 2012). Stevens et al. (1937) propuseram a escala Mel, um modelo para aproximar a percepção sonora, desenvolvida por meio de experimentos psicológicos. O modelo de conversão de frequências para a escala Mel elaborado por O'Shaughnessy (1987) é dado na equação 2.14.

$$m(f) = 1127 \cdot \log_{10}\left(1 + \frac{f}{700Hz}\right) \tag{2.14}$$

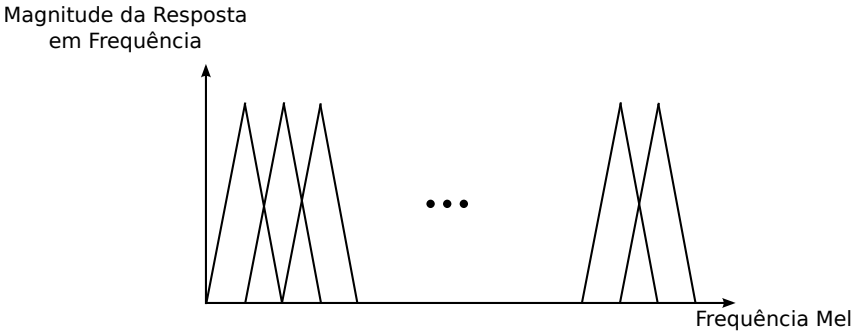
Mel-Frequency Cepstral Coefficients (MFCC) são características que tentam incorporar propriedades do sistema auditivo humano na representação do sinal. Essa técnica foi criada no contexto de reconhecimento automático de voz por Hunt et al. (1980) e fornece uma representação mais compacta para o áudio em comparação com a STFT.

Primeiramente, calcula-se a STFT do sinal de áudio. Em seguida, o espectro de cada janela é filtrado com um banco de filtros passa-banda triangulares, igualmente espaçados na escala Mel. A figura 2.3 ilustra o banco de filtros na escala normal (figura 2.3a) e na escala Mel (figura 2.3b). Os MFCCs são obtidos por meio da equação 2.15, em que K é o número de filtros no banco de filtros, S_k é a potência de saída do k^{esimo} filtro e n é o índice do *bin* MFCC (Han et al., 2006).

$$c_n = \sum_{k=1}^{K} (\log S_k) \cos[n(k - 0.5)\frac{\pi}{K}] \tag{2.15}$$



(a) Escala Normal



(b) Escala Mel

Figura 2.3: Banco de filtros na escala normal e na escala Mel. Figura adaptada de (Han et al., 2006)

A figura 2.4 apresenta o MFCC da música Abracadabra, com $\mathcal{H} = 1024$, $\mathcal{K} = 2048$ e função *window Blackman-Harris*.

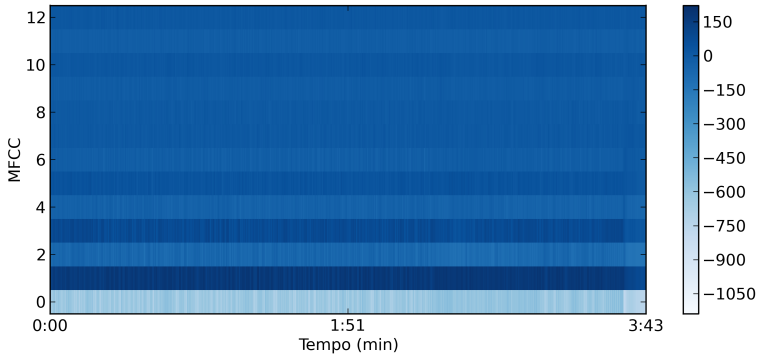


Figura 2.4: MFCC da música Abracadabra, interpretada por Steve Miller Band (base de dados Covers80 (Ellis, 2007)).

2.4 Análise tonal

Aspectos tonais são muito importantes para a análise de músicas. O tom (*pitch*) está diretamente relacionado com a melodia, a harmonia e a assinatura tonal de uma composição musical. A percepção humana de tom está relacionada com a frequência de um sinal, sendo que frequências mais altas levam a percepção de tons mais altos (Lerch, 2012).

Instrumentos musicais produzem sons que podem ser aproximados por uma combinação linear de componentes senoidais de frequências f_0, $2f_0$, $3f_0$, ..., nf_0, em que a frequência fundamental f_0 determina como o tom será percebido.

Os tons (ou notas) são nomeados usando as primeiras 7 letras do alfabeto. A figura 2.5 ilustra um exemplo de um teclado de piano, com as notas anotadas sobre as teclas. Em um piano, o C central é chamado de C4. O primeiro (mais a esquerda) C é o C1 e o último C (a direita) é o C8. Uma oitava é o intervalo de um C até o próximo C (8 teclas brancas) (Kostka et al., 1995).

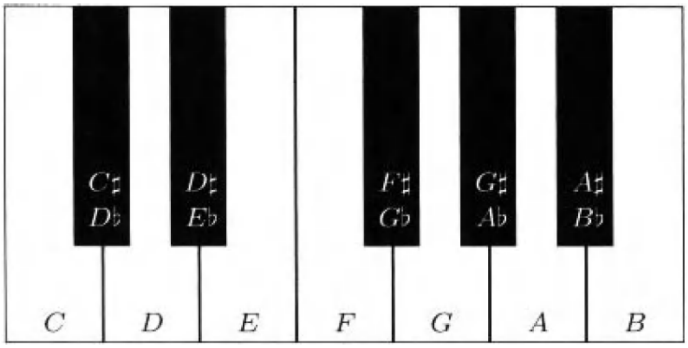


Figura 2.5: Uma oitava do teclado de um piano (Lerch, 2012).

2.4.1 Percepção de chroma

Sons cujas frequências têm uma razão de potência de 2 (f_0, $2f_0$, $4f_0$, $8f_0$, ...) são percebidos como similares. Esse fenômeno é conhecido como percepção de *chroma*. A figura 2.6 ilustra esse fenômeno como um gráfico em hélice. A frequência cresce monotonicamente no eixo Z (Lerch, 2012). Pontos com as mesmas coordenadas em X e Y compartilham de uma razão de frequência em potência de 2, ou seja, compartilham a mesma classe de notas (*pitch class*). Na música ocidental, as classes de notas são: C, $C\#/Db$, D, $D\#/Eb$, E, F, $F\#/Gb$, G, $G\#/Ab$, A, $A\#/Ab$, B. O termo classe de nota é utilizado para agrupar todas as notas que tem o mesmo tom, exceto pela diferença de uma ou mais oitavas (Kostka et al., 1995).

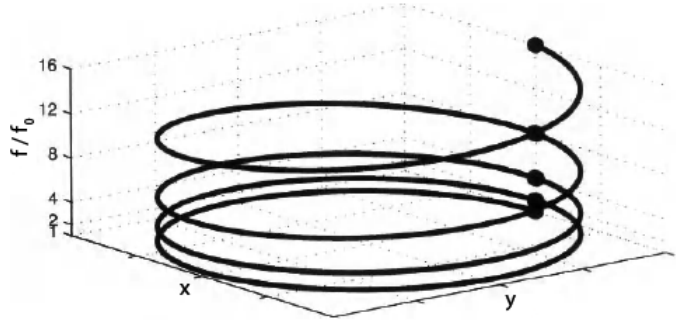


Figura 2.6: Visualização da percepção de tom. Os eixos X e Y simbolizam as *chromas* e o eixo Z, as frequências. Figura adaptada de (Lerch, 2012).

2.4.2 Pitch Class Profile

O algoritmo *Pitch Class Profile* (PCP) foi desenvolvido por Fujishima (1999) com o intuito de reconhecer acordes musicais. Neste processo, calcula-se a transformada discreta de Fourier (DFT) de janelas do áudio (*Short Time Fourier Transform*). Em seguida, deriva-se o PCP como sendo um vetor de doze dimensões que representa as intensidades de doze classes de semitons em cada janela. Cada vetor PCP é calculado por meio da fórmula 2.16, sendo M definida pela equação 2.17 e X a STFT em uma janela (Fujishima, 1999).

$$PCP(p) = \sum_{l|M(l)=p} ||X(l)||^2 \tag{2.16}$$

$$M(l) = \begin{cases} -1 & l = 0 \\ round(12log_2((f_s.\frac{1}{N})/f_{ref}))mod12 & l = 1, 2, ..., N/2 - 1 \end{cases} \tag{2.17}$$

$$X(l) = \sum_{n=0}^{N-1} e^{-2\pi ikn/N} x(n) \tag{2.18}$$

A tabela M mapeia o espectro resultante da DFT para o espectro dos doze vetores que representam os semitons. A constante de frequência de referência (tom em que os instrumentos foram afinados) é dada por f_{ref}, cujo valor padrão é 440 Hz, correspondente à nota A4.

2.4.3 Harmonic Pitch Class Profile

Harmonic Pitch Class Profile (HPCP) é uma *feature* de distribuição de tons baseado no PCP. O algoritmo para calcular o HPCP consiste em:

1. Dividir o sinal em fragmentos e aplicar a STFT;
2. Normalizar o espectro;
3. Calcular o conjunto de máximos locais;
4. Selecionar os máximos locais que têm frequências entre 40 e 5000 Hz;
5. Calcular o vetor HPCP para cada fragmento.

O vetor HPCP é definido com a equação 2.19.

$$HPCP(n) = \sum_{i=1}^{nPeaks} w(n, f_i) \cdot a_i^2, n = 1...size \tag{2.19}$$

$$w(n, f_i) = \begin{cases} cos^2\left(\frac{\pi}{2} \cdot \frac{d}{0,5 \cdot l}\right) & , |d| \leq 0,5 \cdot l \\ 0 & , |d| > 0,5 \cdot l \end{cases} \tag{2.20}$$

a_i é a magnitude e f_i é a frequência do pico número i. $nPeaks$ é o número de picos espectrais considerados e n é o *bin* HPCP. $size$ é o tamanho do vetor HPCP (12, 24, 36, ...) e $w(n, f_i)$, dado pela equação 2.20, é o peso associado à frequência f_i com o HPCP *bin* n. f_n é a frequência central do *bin* n (equação 2.21) e d é a distância em semitons entre o pico da frequência f_i e a frequência central f_n (equação 2.22).

$$f_n = f_{ref} \cdot 2^{\frac{n}{size}}, n = 1...size \tag{2.21}$$

$$d = 12 \cdot log_2 \frac{f_1}{f_n} + 12 \cdot m \tag{2.22}$$

Um pico é definido como um local de máximo na magnitude do espectro, onde a frequência está em um determinado intervalo e a magnitude é maior que um determinado limiar. Os picos são detectados com a abordagem proposta por Serra (1997).

A figura 2.7 apresenta o HPCP da música Abracadabra, com $\mathcal{H} = 1024$, $\mathcal{K} = 2048$ e função *window Blackman-Harris*.

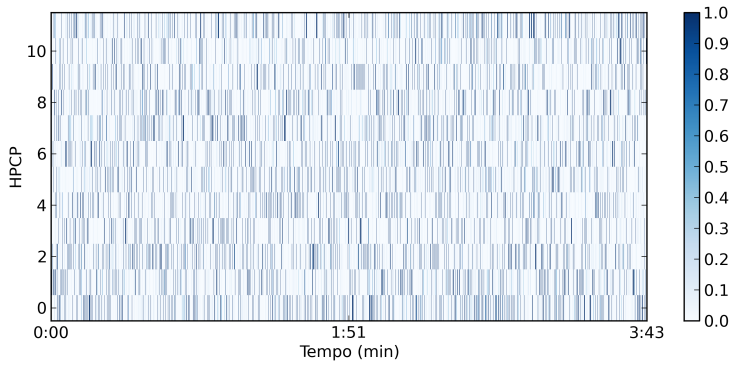


Figura 2.7: HPCP da música Abracadabra, interpretada por Steve Miller Band (base de dados Covers80 (Ellis, 2007)).

2.5 Métricas de similaridade

Dadas as *features* de um sinal de áudio, pode-se definir métricas de similaridade para comparar músicas. Tais métricas são importantes para a identificação de versões *cover* de músicas (versões alternativas da mesma música previamente gravada).

O *Dynamic Time Warped* (DTW) (Keogh et al., 2005) é uma técnica que permite encontrar similaridades e definir uma métrica de distância em séries temporais. Dadas duas séries, Q e C, $|Q| = n$ e $|C| = m$ (equação 2.23), cria-se uma matriz D, n x m, onde o elemento $d_{i,j}$ é a distância entre o elemento q_i e c_j.

$$\begin{aligned} Q &= q_1, q_2, q_3, ..., q_n \\ C &= c_1, c_2, c_3, ..., c_m \end{aligned} \tag{2.23}$$

A partir da matriz D, o algoritmo procura por um caminho mínimo, definido pela equação 2.24. Este caminho de deformação (*warping*) W é um conjunto de elementos da matriz D, $w_k = (i, j)_k$ que respeite as condições:

- Comece em $w_{1,1}$ e termine em $w_{m,n}$;
- Percorra apenas índices adjacentes;
- Percorra espaçamentos iguais no tempo.

$$DTW(Q, C) = min \sqrt{\sum_{k=1}^{K} w_k} \tag{2.24}$$

A figura 2.8 ilustra a execução do algoritmo DTW.

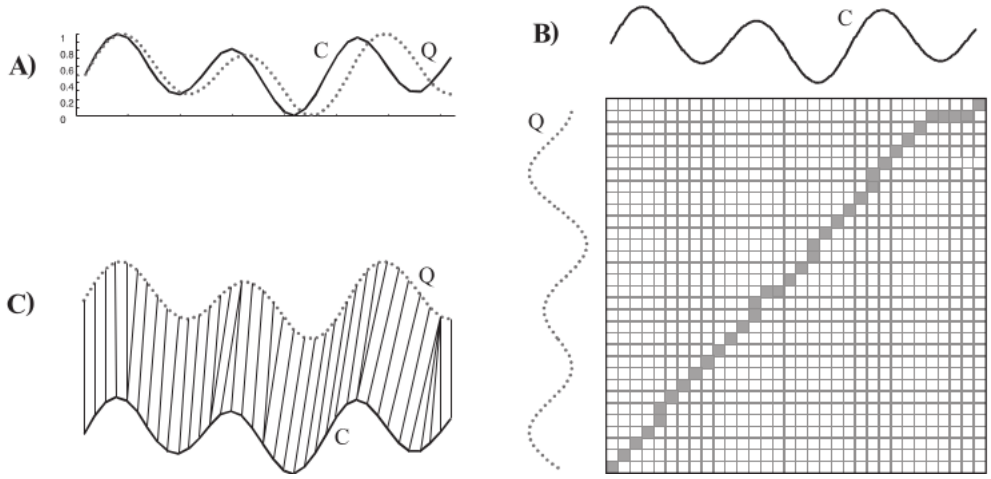


Figura 2.8: Execução do algoritmo de DTW. (A) Séries C e Q. (B) Matriz de distâncias. (c) Alinhamento das duas séries com o caminho mínimo (Keogh et al., 2005).

A medida $DTW(Q, C)$ pode ser utilizada para comparar dois sinais de áudio. Entretanto, é uma métrica computacionalmente muito custosa, tendo complexidade $O(nm)$. Como músicas são sinais muito grandes, sua aplicabilidade é muito limitada.

O trabalho de Serrà et al. (2009) propõe o uso de *Cross Recurrence Plots* (CRPs) para identificar partes de músicas que são semelhantes. Primeiramente, o trabalho extrai a característica de *chroma harmonic pitch class profile* (HPCP) (Gómez, 2006) do sinal de áudio com N_x^* janelas. Isso resulta em uma série temporal HPCP de $H = 12$ variáveis. Em seguida, é feito um *state space embedding* da série temporal da forma $\mathbf{x} = \{\mathbf{x_i}\}$, para $i = 1, ..., N_x$, $N_x = N_x^* - (m-1) * \tau$.

$$x_i = (x_{1,i}, x_{1,i+\tau}, ..., x_{1,i+(m-1)\tau}, x_{2,i}, x_{2,i+\tau}, ..., x_{2,i+(m-1)\tau}, ..., x_{H,i}, x_{H,i+\tau}, ..., x_{H,i+(m-1)\tau}) \tag{2.25}$$

Os autores estimaram os valores ótimos para m e τ para o reconhecimento de músicas *cover*. Valores no intervalo $7 \leq (m-1)\tau \leq 17$ estão em um intervalo ótimo e, no artigo, são utilizados as constantes $m = 10$, $\tau = 1$.

Recurrence Plot (RP) é uma ferramenta utilizada para identificar características similares de estados de um sistema em tempos diferentes. O RP é uma matriz quadrada preenchida com zeros e uns, que indicam se há ou não recorrência (o estado no tempo i é similar ao estado do tempo j (Eckmann et al., 1995)). A diagonal principal de um RP é, portanto, composta por uns.

CRPs são construídos da mesma maneira que RPs, mas cada eixo corresponde a uma música diferente e a matriz resultante não é quadrada. Para analisar as dependências entre duas músicas, x e y, computamos o CRP como na equação 2.26.

$$R_{i,j} = \Theta(\epsilon_i^x - ||\mathbf{x_i} - \mathbf{y_j}||)\Theta(\epsilon_j^y - ||\mathbf{x_i} - \mathbf{y_j}||) \tag{2.26}$$

em que $\Theta(\cdot)$ é a função degrau tipo Heaviside ($\Theta(v) = 0$ se $v < 0$ e $\Theta(v) = 1$ caso contrário) e ϵ_i^x e ϵ_i^y são limiares de distâncias. Em geral, pares de músicas diferentes não exibem nenhum padrão evidente e pares de músicas *cover* apresentam estrutura de linhas longas. A figura 2.9 ilustra o CRP da música "*Day Tripper*"(The Beatles) com (a) uma música cover e (b) com "*I've got a crush on you*". Observa-se que quando as músicas são covers (a), forma-se diagonais longas no CRP e caso contrário, tais padrões não se formam.

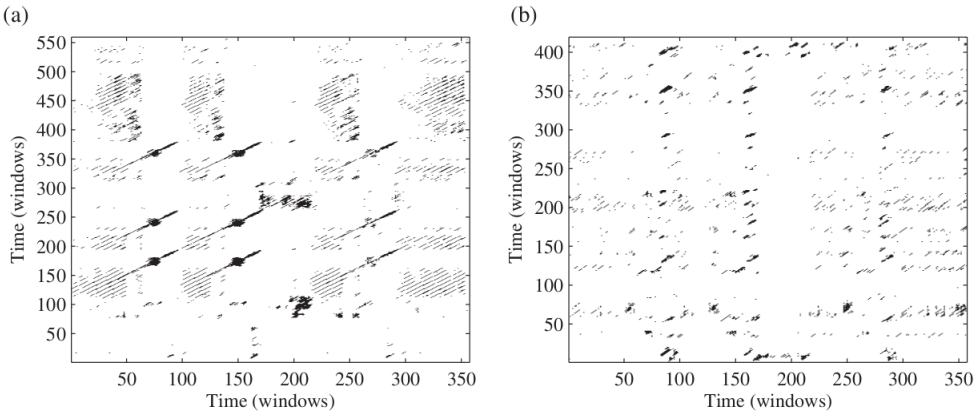


Figura 2.9: *Cross Recurrence Plot* da música "*Day Tripper*"da banda The Beatles com (a) uma música cover interpretada por Ocean Colour Scene e (b) "*I've got a crush on you*" por Frank Sinatra. Parâmetros são $m = 9$ e $\tau = 1$. Figura adaptada de (Serrà et al., 2009).

Dado o CRP de duas músicas, o trabalho de Serrà et al. (2009) propõe o uso da maior distância das diagonais formadas na matriz como uma *feature* de entrada em um *support vector machine* (SVM) para identificar *covers* e não *covers*. A medida Q_{max} é definida como o maior comprimento das diagonais na matriz CRP, considerando possíveis variações de tempo da música (que correspondem a curvaturas nos traços) e na melodia (pequenas rupturas). Para o cálculo do Q_{max}, define-se primeiramente a matriz Q: $Q_{1,j} = Q_{2,j} = Q_{i,1} = Q_{i,2} = 0$ para $i = 1..., N_x$ e $j = 1, ..., N_y$ e aplica-se a equação 2.27 recursivamente. Q_{max} corresponde ao valor máximo da matriz Q.

$$Q_{i,j} = \begin{cases} max\{Q_{i-1,j-1}, Q_{i-2,j-1}, Q_{i-1,j-2}\} + 1 & , R_{i,j} = 1, \\ max\{0, Q_{i-1,j-1} - \gamma(R_{i-1,j-1}), Q_{i-2,j-1} - \gamma(R_{i-2,j-1}), Q_{i-1,j-2} - \gamma(R_{i-1,j-2})\} & , R_{i,j} = 0\} \end{cases} \tag{2.27}$$

Para $i = 3, ..., N_x$ e $j = 3, ..., N_y$, com γ definido pela equação 2.28

$$\gamma(z) = \begin{cases} \gamma_0 & z = 1 \\ \gamma_e & z = 0 \end{cases} \tag{2.28}$$

γ_0 e γ_e são penalidades para irregularidades nas diagonais da matriz.

A matriz Q da música "Day Tripper", interpretada por The Beatles e Ocean Colour Scene, é apresentada na figura 2.10.

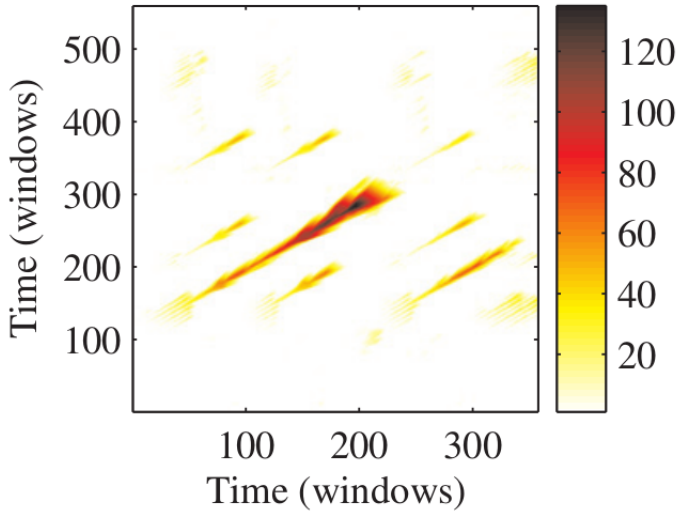


Figura 2.10: Matriz Q da música "Daytripper" interpretada por The Beatles e Ocean Colour Scene. $\gamma_0 = 3$ e $\gamma_e = 7$ (Serrà et al., 2009).

Dada uma música para busca de *covers*, todas as outras músicas da base serão avaliadas e a que resultar no maior Q_{max} será considerada uma *cover*. O trabalho conseguiu uma acurácia de 0.661 na competição MIREX (*Music Information Retrieval Evaluation eXchange*)[2] 2008 e ficou em segundo lugar na competição de identificação de músicas *cover*, perdendo apenas para uma variação desta técnica que utiliza aprendizado não supervisionado para classificar as músicas.

[2] `www.music-ir.org/mirex`

2.6 Considerações finais

Neste capítulo, o processo de extração de características de sinais de áudio foi descrito, desde o pré-processamento até a extração de *features* e a comparação de músicas.

O estado da arte em característica de *chroma*, HPCP, foi apresentado. Essa técnica será utilizada para gerar a principal característica para comparação de músicas do presente trabalho, pois apresenta os melhores resultados para caracterização e comparação de músicas.

Desde 2008, o melhor resultado de *Mean of Average Precisions* (MAP) (Manning et al., 2008) da competição MIREX para identificação de músicas *cover* é o de Serra et al. (2009), com $MAP = 0.75$. Este trabalho é um aprimoramento do algoritmo descrito em (Serrà et al., 2009), que adiciona uma etapa de pós-processamento à matriz de similaridades derivada do Q_{max} e identifica grupos de músicas *cover* com técnicas de aprendizado não supervisionado.

A visualização de informações é uma importante ferramenta para análise, exploração e entendimento dos dados. No próximo capítulo, serão descritos os principais métodos de visualização de músicas.

Capítulo

3

Visualização de músicas

A visualização de informações é uma ferramenta que permite a apresentação e a rápida interpretação de uma grande quantidade de dados. Além disso, possibilita a identificação de padrões e problemas no método de coleta de dados (Ware, 2013). No contexto de visualização de músicas, pode-se agrupar os métodos existentes com relação à quantidade de dados analisada: visualização de música individual e visualização de coleção de músicas. Neste capítulo, serão apresentados os principais trabalhos nessas duas vertentes.

3.1 Visualização de música individual

A visualização de composições musicais tem como objetivo facilitar a sua interpretação, oferecendo informações sobre diversas propriedades, por exemplo, tons, notas, acordes, harmonia e contexto.

A visualização *Piano Roll* é derivada dos rolos perfurados de programação de pianolas (pianos com um dispositivo que executa automaticamente a músicas do rolo de programação). Nesta representação, as notas são marcadas num plano 2D, onde o eixo Y representa o tom (*pitch*) da nota e o eixo X, o tempo. A figura 3.1 apresenta a música "With You Friends", de Skrillex, com a representação de *piano roll*. O software utilizado para gerar a visualização é o MIDITrail (Yknk, 2012).

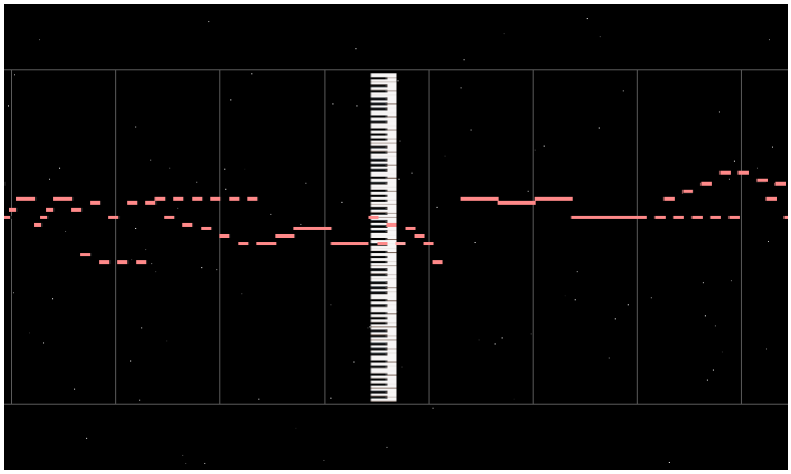


Figura 3.1: *Piano roll* da música "*With You Friends*", de Skrillex

No *piano roll*, pode-se visualizar as notas que serão tocadas, progressões e padrões no tempo. Entretanto, cabe ao usuário a interpretação da interação entre as notas e o seu contexto na música como um todo.

Malinowski (2013) desenvolveu uma variação para o *piano roll* original chamada *Music Annotation Machine*, em que cada faixa (instrumento) da música MIDI é representada por uma forma geométrica diferente e a cor indica a tonalidade (*chroma*). A figura 3.2 apresenta a visualização gerada para a música *The Rite of Spring*, de Stravinsky. Para tornar a visualização mais agradável, foram utilizados efeitos de brilho, indicando as notas tocadas no momento.

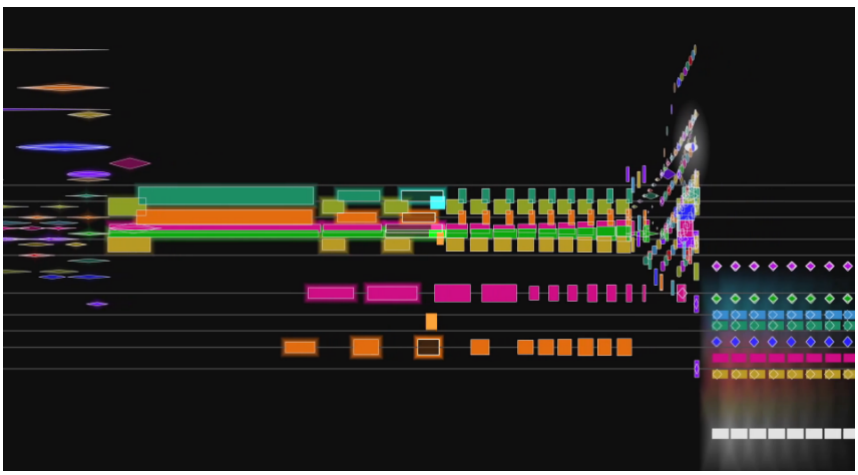


Figura 3.2: Visualização gerada pela *Music Annotation Machine* para a música *The Rite of Spring*, de Stravinsky.

Algumas técnicas foram desenvolvidas para permitir a visão de uma música como um todo. Essas visualizações respeitam o princípio de foco + contexto (definido por Ware

(2013) como o problema de encontrar e explorar detalhes em um contexto maior) e evitam exibições variantes no tempo.

Hiraga et al. (2002) utiliza a técnica *fish eye* (Furnas, 1986) para criar uma notação reduzida de partituras. A representação utiliza barras verticais e variações de intensidade de cinza para transmitir uma ideia geral da composição. A figura 3.3 ilustra a composição "Clarinet Quintet A-major, K.V. 581", por W. A. Mozart, com a partitura condensada. A armadura da clave (parte inicial) é preservada para prover informações de clave e tonalidade da partitura.

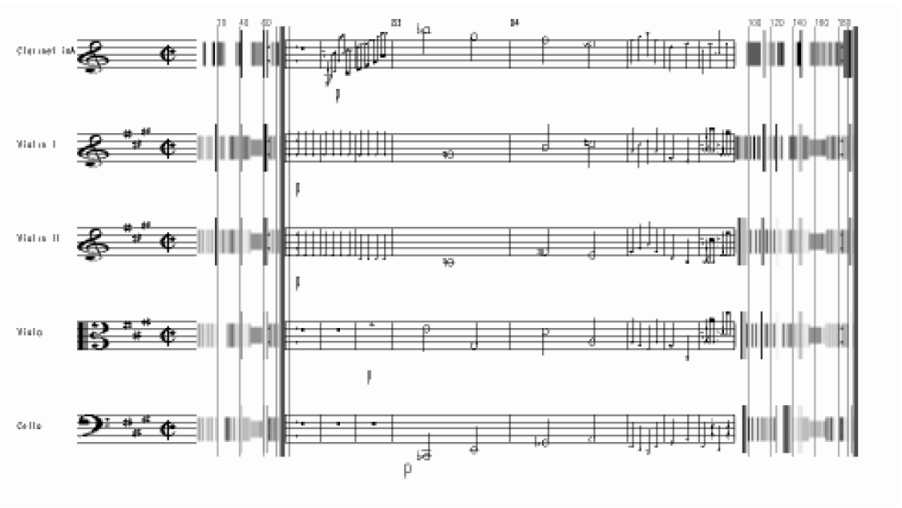


Figura 3.3: Partitura condensada da composição "Clarinet Quintet A-major, K.V. 581", por W. A. Mozart (Hiraga et al., 2002).

A visualização *Shape of Song* (Wattenberg, 2002) busca por estruturas que se repetem em músicas MIDI, definindo um limiar de aceitação para correspondências quase perfeitas. A tarefa é realizada buscando-se sequências diretas de notas que aparecem em outras regiões, de forma similar a uma comparação de strings. Seções que se repetem são ligadas por um semi-círculo. A figura 3.4 mostra uma execução do *software* para a música *All The Small Things*, da banda Blink 182.

Figura 3.4: Shape of Song da música *All The Small Things*, por Blink 182 (Wattenberg, 2002).

A ferramenta *Infinite Jukebox* (Lamere, 2012) foi criada para visualizar estruturas de repetição dentro de um sinal de áudio. Ela permite que o usuário crie representações de músicas de uma base de dados pré-definida ou envie músicas para o sistema *online*. A API *Echonest* (Echonest, 2013) é utilizada para segmentar o sinal em tempos (batidas) e extrair o tom, timbre e altura de cada um dos segmentos de áudio. Assim como em *Shape of Song*, busca-se por segmentos que se repetem e representa-se essa estrutura como um grafo com *layout* circular. O aplicativo permite que se crie uma versão "infinita" da música por meio de um percurso aleatório nos caminhos descritos no grafo. As cores dos nós representam o timbre da música na seção. A figura 3.5 apresenta essa visualização.

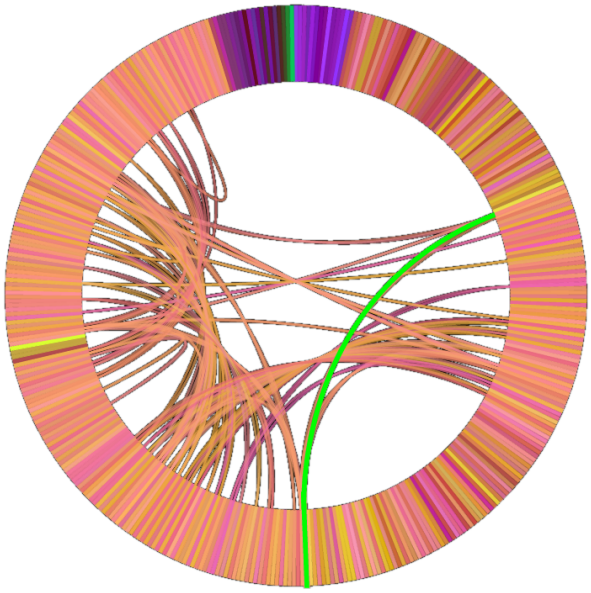


Figura 3.5: Visualização *Infinite Jukebox* da música *Lights*, por Ellie Goulding (Lamere, 2012).

A tabela 3.1 apresenta uma comparação das técnicas de visualização de composições musicais. As técnicas são classificadas de acordo com o tipo de dado que processam e quais informações são apresentadas na visualização: notas, *chroma*, harmonia (combinação de notas) e contexto.

Tabela 3.1: Comparação de técnicas de visualização de composições musicais

Técnica	Tipo de dado	Notas	*Chroma*	Harmonia	Contexto
Piano Roll	Partitura digital	✓		✓	
Music Anotation Machine	Partitura digital	✓	✓	✓	
Hiraga et al. (2002)	Partitura digital	✓		✓	✓
Shape of Song	Partitura digital				✓
Infinite Jukebox	Sinal de áudio				✓

3.2 Visualização de coleções de músicas

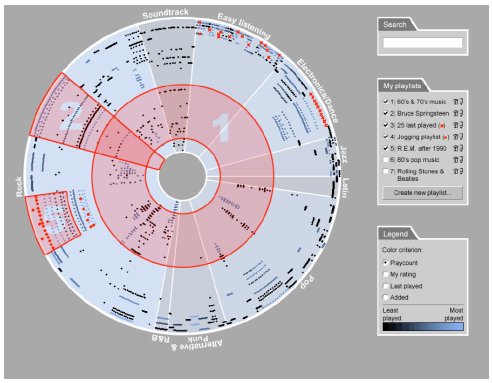
Diversos trabalhos foram desenvolvidos no contexto de visualização de coleções de músicas, com o objetivo de facilitar a exploração de grandes bibliotecas de áudio. As técnicas apresentadas a seguir mapeiam o conjunto de músicas para o espaço visual e permitem a interação do usuário, a identificação de grupos e, em sua maioria, a criação de *playlists*.

O trabalho de Torrens et al. (2004) propõe a exploração de coleções de músicas por meio de metadados. Três visualizações foram desenvolvidas: disco (figura 3.6a), retângulo (figura 3.6b) e *treemap* (figura 3.6c). As *features* utilizadas provém de *tags* das músicas, que contém informações sobre o artista, compositor, ano, álbum, gênero, quantidade de reproduções e classificação.

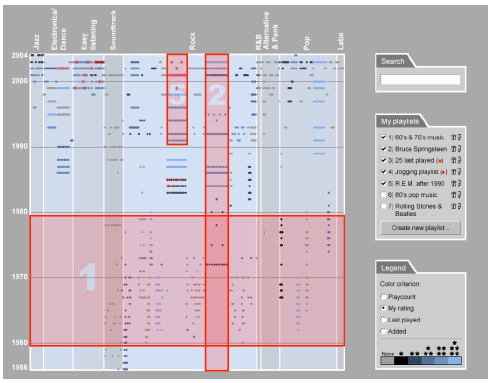
A visualização "disco" é baseada nos gráficos de disco, que possibilitam a percepção de porcentagem e proporção. O disco é dividido em diferentes setores que são associados aos gêneros da biblioteca. O seu tamanho é proporcional à quantidade de músicas do gênero. Cada setor é dividido em sub-setores, que representam os artistas no determinado gênero (Torrens et al., 2004). *Glyphs* para músicas são posicionados em seus respectivos setores e sub-setores.

As visualizações disco e retângulo têm basicamente as mesmas funcionalidades: permitem a exploração da coleção de músicas e a criação de *playlists*. A diferença está no formato de apresentação, que no retângulo consiste em colunas representando o gênero e linhas, os artistas.

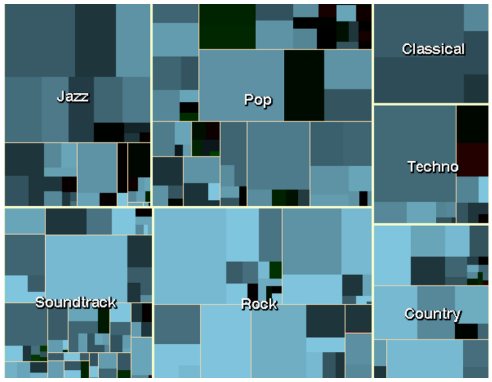
A representação *treemap* permite a visualização de três níveis de detalhe da biblioteca: gênero, sub-gênero artista. Entretanto, não permite a criação de *playlists*, pois músicas individuais não são apresentadas nessa visualização.



(a) Visualização por disco



(b) Visualização por retângulo



(c) Visualização *treemap*

Figura 3.6: Exploração de coleções de músicas com metadados (Torrens et al., 2004).

Outro trabalho que permite visualizar, explorar e gerenciar um conjunto de músicas baseado em *tags* é o *MusicNodes* (Dalhuijsen et al., 2010). A ferramenta possibilita que o usuário atribua uma porcentagem de semelhança a diferentes gêneros, por exemplo, 40% rock e 60% eletrônica, e crie novas *tags* para classificar as músicas.

Nesta representação, cada ponto simboliza um álbum, que é colorido de acordo com um mapeamento gênero-cor. Os álbuns são posicionados no espaço bidimensional por meio de um sistema baseado em forças: gêneros musicais atraem álbuns que pertencem a sua classe, enquanto cada álbum afasta um pouco seus vizinhos. O sistema permite que usuários façam seleção e exportem *playlists*, por meio de buscas textuais e seleções visuais.

A figura 3.7 apresenta uma base de músicas com a ferramenta MusicNodes.

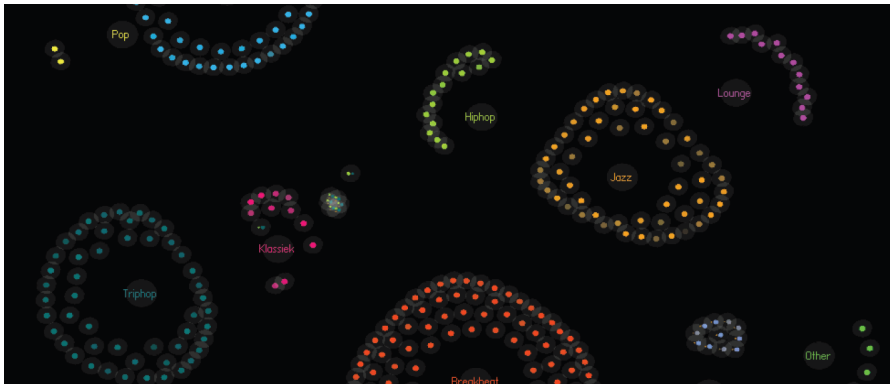


Figura 3.7: Base de músicas representada com *MusicNodes* (Dalhuijsen et al., 2010).

O trabalho de (Pampalk et al., 2002) propõe a visualização *Islands of Music*. Nela, não são considerados os metadados dos arquivos de áudio, mas sim o seu conteúdo. Cada música é avaliada por um *pipeline* de extração de características de ritmo que resulta em uma matriz de 20x60 coeficientes. As *features* do conjunto de músicas são levadas ao algoritmo *Self-Organizing Map* (SOM), que organiza a coleção de dados no espaço visual. A representação *Islands of Music* é gerada como um mapa de densidade dos pontos no espaço 2D. A figura 3.8 ilustra a execução desta técnica em um conjunto de setenta e sete músicas.

Figura 3.8: Representação de um conjunto de músicas com *Islands of Music* (Pampalk, 2001).

MusicBox (Lillie, 2008) é uma ferramenta que realiza o mapeamento de músicas mp3 no espaço bidimensional. Primeiramente, características são extraídas do sinal de áudio (duração da música, tempo, timbre e histograma de ritmo) e de descritores de fontes online (popularidade, humor e gênero). Essas características são então mapeadas para o espaço 2D por meio da técnica *Principal Component Analysis* (PCA). Esse trabalho também permite a customização do *layout* e a criação de *playlists*. A figura 3.9 apresenta um exemplo de execução do sistema.

Figura 3.9: Exemplo de execução da ferramenta *MusicBox* (Lillie, 2008).

Muelder et al. (2010) propõe uma técnica para visualizar bibliotecas de música com grafos, baseado na análise do conteúdo dos arquivos de áudio. O trabalho calcula a média e o desvio padrão dos MFCCs (*Mel Frequency Cepstrum Coefficients*) de cada música e usa essas características para medir a similaridade entre os sinais de áudio com a distância euclidiana. No grafo proposto, cada música é representada por um nó e as relações entre músicas, por arestas ponderadas. Os nós são posicionados no espaço bidimensional por meio de uma técnica de mapeamento descrita em (Muelder et al., 2008). O algoritmo, baseado em curvas de preenchimento de espaço, evita sobreposições e possibilita que as capas dos álbuns sejam utilizadas como *glyphs*. É estabelecida uma relação entre a transparência das arestas e o grau de similaridade entre músicas. A figura 3.10 apresenta um exemplo da execução desse algoritmo com uma coleção de músicas da banda The Beatles.

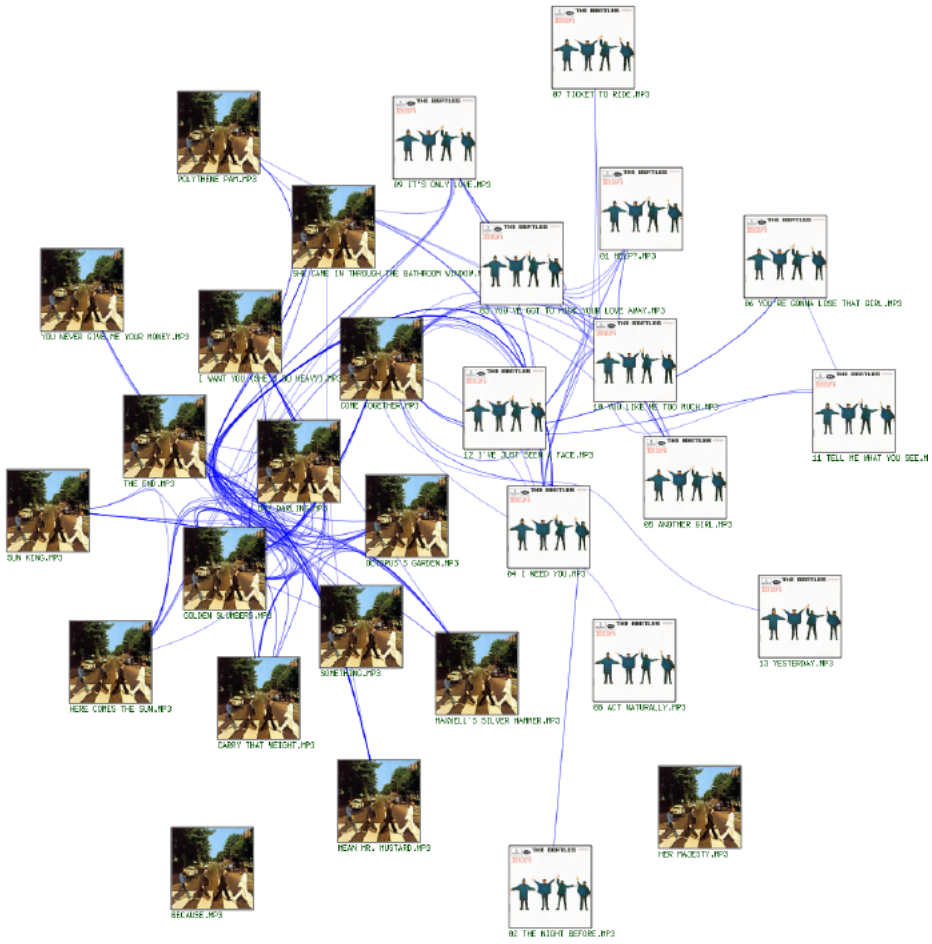


Figura 3.10: Organização de uma base de dados da banda The Beatles (Muelder et al., 2010).

O trabalho de Paulovich et al. (2011) propõe uma técnica de projeção multidimensional local denominada *Piecewise Laplacian-based Projection* (PLP) e a aplica no contexto de visualização de bases de música. Um vetor de 78 características é extraído para cada uma das 3.857 músicas da base de dados, por meio da ferramenta *JAudio* (McKay et al., 2005). Entre as características utilizadas, estão medidas estatísticas, histograma de batidas e análises espectrais. A figura 3.11 ilustra um exemplo de projeção de bases de música com a projeção PLP. Seleções podem ser feitas (na figura, em verde e rosa) para organização e criação de *playlists*.

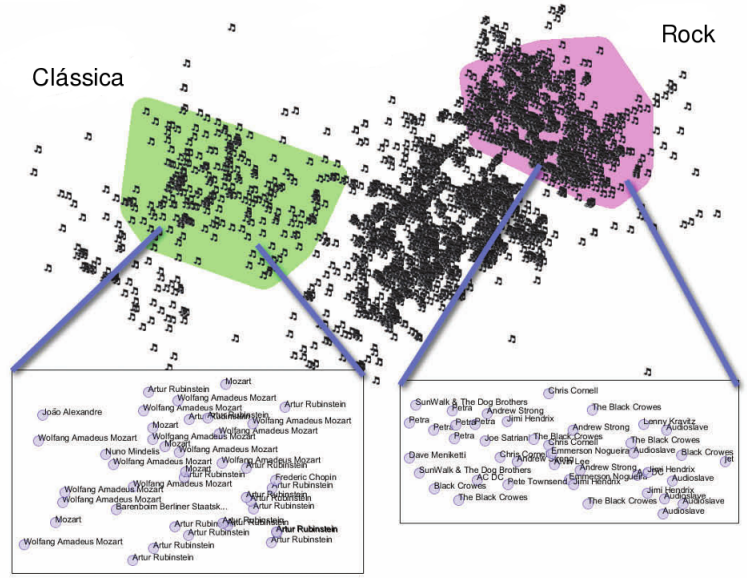


Figura 3.11: Sistema de criação de Playlists com a projeção multidimensional PLP (Paulovich et al., 2011).

A tabela 3.2 apresenta uma comparação de técnicas de visualização de coleções de músicas. As técnicas são classificadas quanto ao tipo de dado analisado, a técnica de visualização empregada e a informação apresentada: visualização de gênero ou informação de similaridade.

Tabela 3.2: Comparação de técnicas de visualização de coleções de música

Técnica	Tipo de dado	Tipo de visualização	Gênero	Similaridade
Torrens et al. (2004)	*Tags*	Subdivisão do Espaço	✓	
Music Nodes	*Tags* + Usuário	Projeção baseada em forças	✓	
Islands of Music	*Features* do Sinal	Mapa de Densidade		✓
MusicBox	*Features* do Sinal	Projeção PCA		✓
Muelder et. al (2010)	*Features* do Sinal	Grafos + Edge Bundle		✓
Paulovich et. al (2011)	*Features* do Sinal	Projeção PLP		✓

3.3 Considerações finais

Neste capítulo, foram descritos os principais trabalhos sobre visualização de composições musicais e de coleções de músicas.

Embora existam diversos trabalhos que permitem a exploração visual de características de músicas individuais e de bases de músicas, não existe nenhum método que ressalte visualmente quais são os trechos de duas ou mais músicas semelhantes.

No próximo capítulo, será proposta uma técnica que permita a identificação de trechos de músicas semelhantes, que resultará em uma visualização mais intuitiva do que os CRPs de Serrà et al. (2009).

Capítulo

4

Proposta de trabalho

4.1 Considerações iniciais

Neste documento, diversas técnicas de visualização de músicas foram investigadas. Pode-se agrupá-las em duas categorias: visualização de composições musicais, em que propriedades de uma única peça são estudadas, e visualização de coleções de músicas, cujo objetivo é apresentar de forma visual suas relações de similaridade, permitindo ao usuário organizar de forma visual sua base de dados e criar *playlists*. Neste projeto de mestrado, serão feitas contribuições nas duas linhas de exploração visual de músicas.

Embora seja possível identificar músicas semelhantes com os algoritmos de visualização apresentados, não foram encontradas representações que destacam em quais trechos (intervalos) essas similaridades ocorrem. Tal funcionalidade pode ser útil na identificação de músicas *cover*, identificação de plágio e mixagem (combinação) de músicas por DJs. Com base nesse contexto, a proposta deste projeto de mestrado é desenvolver novos paradigmas visuais para visualização e comparação de músicas, que permitam explorar bases de dados e ressaltar quais são as partes similares entre as diferentes músicas que a compõem.

O trabalho de Serrà et al. (2009) introduziu dois conceitos importantes para a área de recuperação de informações de música: o uso de *Cross Recurrence Plots* para analisar as dependências entre músicas e a medida *Qmax*, uma métrica de similaridade de séries temporais derivada dos CRPs.

Neste projeto de mestrado, serão utilizadas essas contribuições para derivar duas visualizações distintas e complementares de bases de dados de música:

1. Visualização de coleções de músicas por meio de projeções multidimensionais e métrica de distância *Qmax*;
2. Visualização das similaridades entre músicas com CRPs.

Na próxima seção, as duas visualizações serão detalhadas e as etapas para o seu desenvolvimento, descritas.

4.2 Metodologia

O desenvolvimento deste projeto de mestrado será dividido em três etapas, descritas nas subseções a seguir. A primeira etapa consiste na extração de características e as próximas duas são referentes às técnicas de visualização de dados.

4.2.1 Extração de características

Nesta etapa, os principais algoritmos de extração de características de sinais de áudio serão estudados e implementados, entre eles, características de forma espectral, MFCC (Hunt et al., 1980) e HPCP (Gómez, 2006). As matrizes CRP e as medidas *Qmax* entre músicas da base de dados serão calculadas.

Descritores de ritmo, por exemplo, tempo, detecção de início e *beat tracking* (Gouyon et al., 2005), serão implementados para complementar a informação de *chroma* na etapa de visualização de coleções de músicas.

4.2.2 Visualização de coleções de músicas com características de chroma

Os algoritmos de visualização de coleções de músicas apresentados neste trabalho utilizam diversas características (*tags*, dados fornecidos por usuários ou *features* baseadas no sinal de áudio) para comparar interpretações. Com base nesses dados, são propostas visualizações que mapeiam cada música para uma região no plano 2D de acordo com relações de similaridade (músicas similares são mapeadas em regiões mais próximas e músicas diferentes, em regiões mais distantes).

Na revisão realizada, entretanto, não foram encontradas visualizações que utilizam características de *chroma* para realizar o mapeamento. Isto é uma deficiência da área, pois diversos trabalhos de classificação automática de gênero utilizam características tonais para realizar a avaliação das peças, o que torna a visualização deste tipo de dados uma linha de pesquisa promissora. As principais características utilizadas por algoritmos de classificação de gênero são timbre, melodia/harmonia e ritmo (Scaringella et al., 2006).

O trabalho de Tzanetakis et al. (2003) utiliza histogramas de tom (vetor de 128 inteiros indexado pelo número da nota MIDI) e o classificador *k-nearest-neighbour* para identificar o gênero de músicas MIDI e dados de áudio (MP3). O artigo apresenta os resultados de classificação para uma base de 500 MIDIs pra treinamento e 500 MIDIs para teste, divididas 5 gêneros de tamanhos iguais: "eletrônica", "clássica", "jazz", "irlandesa"e "rock". Para avaliação, é feito o *10-fold cross-validation*, obtendo-se uma acurácia de $50 \pm 7\%$ (mais de duas vezes melhor que a classificação aleatória, 20%). O algoritmo também é testado com sinais de áudio, utilizando-se a análise multi-tom de Tolonen et al. (2000) para converter o sinal para *chroma*. Nessa abordagem, uma acurácia de $40 \pm 7\%$ é obtida. Este algoritmo possui problemas para identificar música "eletrônica", que se mistura muito com os outros gêneros. Os autores indicam que a combinação de características de ritmo com características tonais pode solucionar este problema.

Em um artigo mais recente, Anglade et al. (2009) apresenta um algoritmo de classificação e caracterização de gênero que analisa partituras digitais por meio de árvores de decisão. A classificação obteve 66% de acurácia em uma base de dados de três classes (clássica, jazz e popular) com 856 composições do programa *Band in a Box* [1]. Por meio deste sistema, pode-se gerar caracterizações de gêneros musicais do tipo:

> "*Some popular music pieces contain a chord root interval sequence of two consecutive ascending fifth directly followed by an ascending minor seventh.*"

Uma vantagem de se utilizar árvores de decisão para classificar o gênero de músicas é que o usuário pode verificar quais regras foram criadas. De acordo com (Anglade et al., 2009), usuários têm dificuldade em acreditar em sistemas de classificação de gênero do tipo "caixa-preta".

Com base nesse contexto, uma hipótese deste trabalho é que a projeção multidimensional com base em *features* de *chroma* e CRP, além de preservar relações de semelhança entre músicas, também formará grupos de músicas que se assemelham por gênero. Primeiramente, o vetor HPCP será calculado para toda a base. Em seguida, as músicas

[1] `http://www.bandinabox.com/bb.php?os=win&lang=pt`

serão comparadas par a par com a medida Q_{max} (Serrà et al., 2009), gerando assim uma matriz de dissimilaridades. As informações contidas nessa matriz serão combinadas com características de ritmo (medidas de periodicidades nas músicas) para obter uma descrição mais completa do sinal de áudio, como sugerido por Tzanetakis et al. (2002).

De posse desses dados, pode-se visualizá-los por meio de diversas projeções multidimensionais, por exemplo *Classical Multidimensional Scaling* (MDS), *Least Squares Projection* (LSP) (Paulovich et al., 2008) e *Local Affine Multidimensional Projection* (LAMP) (Joia et al., 2011). A técnica LAMP permite que o usuário interaja com com a projeção, modificando-a com o reposicionamento de poucos pontos de controle. Assim, ela será utilizada para representar os dados de música, de modo que o conhecimento do usuário seja inserido na visualização. Posicionados os elementos na tela, o usuário poderá interagir com a representação visual, realizando seleções, navegando por meio da técnica *fish eye* (Furnas, 1986) e criando *playlists*.

O aluno de mestrado também desenvolverá *glyphs* para representar características de músicas individuais na projeção multidimensional. Serão seguidas as diretrizes de projeto e técnicas descritas em (Borgo et al., 2012).

A figura 4.1 ilustra o *pipeline* desta visualização.

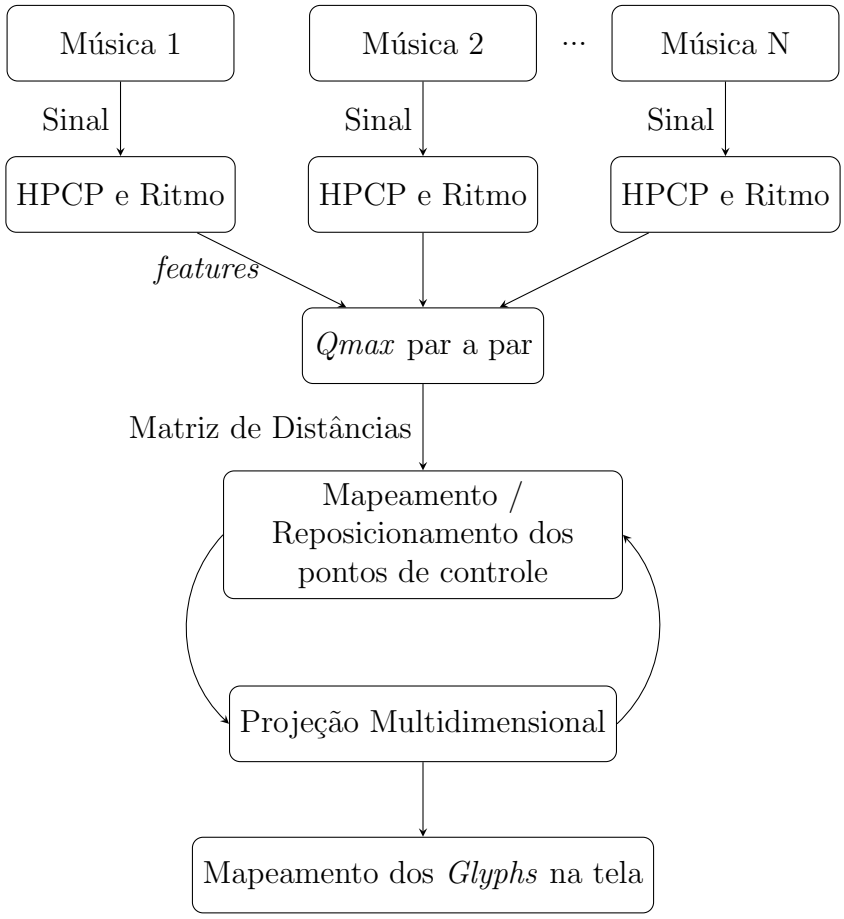


Figura 4.1: *Pipeline* da visualização de base de músicas proposta.

4.2.3 Visualização de similaridade entre músicas

A matriz CRP de dois sinais contém a informação dos trechos em que as duas músicas são semelhantes. A figura 4.2 apresenta a matriz CRP de duas versões da música "*Daytripper*", com os parâmetros ótimos para comparação de músicas de $m = 10$ e $\tau = 1$, encontrados por Serrà et al. (2009). As regiões em azul correspondem ao valor 0 (não há similaridade) e as regiões em branco indicam valor 1 (onde há similaridade).

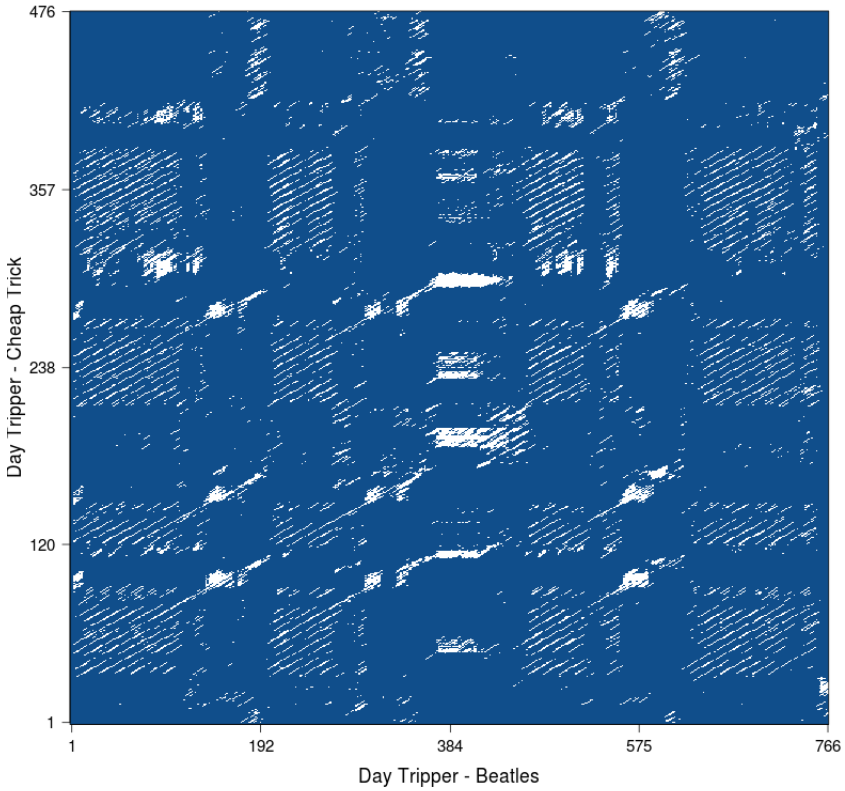
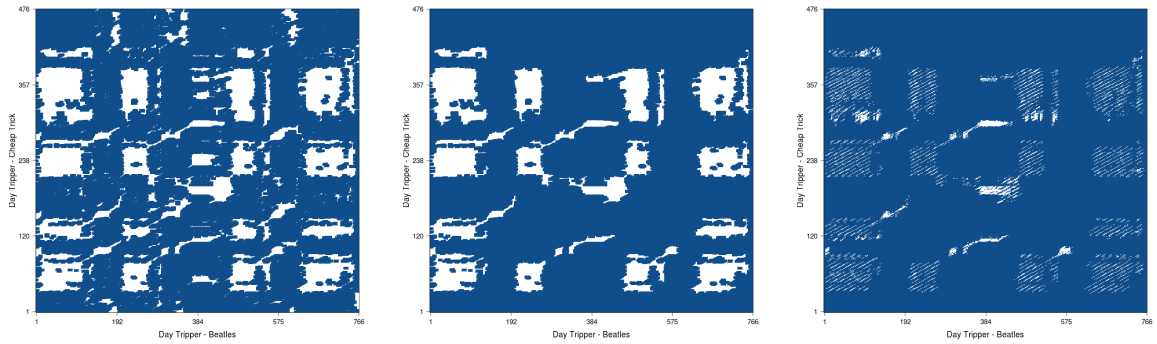


Figura 4.2: CRP de "*Daytripper*", interpretada por Beatles e Cheap Trick (Base de dados Covers80 (Ellis, 2007)).

Deseja-se encontrar as diagonais mais longas da matriz CRP, que correspondem aos trechos em que as maiores semelhanças ocorrem. Em (Serrà et al., 2009), encontra-se o tamanho da diagonal mais longa por meio da medida Q_{max}. Para encontrar todas as diagonais maiores que um limiar, primeiramente é aplicada uma operação de fechamento (dilatação seguida de erosão) (Gonzalez et al., 2008, Capítulo 10) na matriz CRP, agrupando as diagonais mais próximas. Em seguida, rotula-se os componentes conectados (Gonzalez et al., 2008, Capítulo 10) e calcula-se as caixas delimitadoras de cada componente conectado (ponto inferior esquerdo e superior direito). Por fim, elimina-se da CRP as diagonais que pertencerem aos componentes conectados cuja largura e altura da caixa delimitadora é menor do que um limiar, definido pelo usuário. A figura 4.3 ilustra as etapas da filtragem da matriz: em 4.3a, a matriz CRP é transformada com a operação morfológica de fechamento, com um elemento estruturante quadrado de tamanho 7. Em 4.3b, remove-se da matriz os grupos que possuem largura e altura menores do que um limiar, no caso, 75. Em 4.3c, obtêm-se as diagonais de interesse, multiplicando-se elemento a elemento a matriz CRP pela matriz da figura 4.3b.



(a) Fechamento do CRP **(b)** Grupos filtrados **(c)** Matriz CRP filtrada

Figura 4.3: Filtragem das diagonais mais longas da música "*Daytripper*", interpretada por Beatles e Cheap Trick (Base de dados Covers80 (Ellis, 2007)).

A matriz CRP filtrada pode ser considerada uma representação para o grafo que liga todos as janelas de *embedding* similares da música 1 com a música 2. Dada uma CRP $R_{n,m}$ (recorrência entre uma música com n janelas e outra com m janelas) a criação de uma matriz de adjacência que representa o grafo de semelhança das músicas é direta, sendo que a nova matriz $A_{n+m,n+m}$ será simétrica e terá uma cópia da matriz R no intervalo $A_{m:m+n,0:m}$ e uma cópia de $R(-x,y)^T$ (R refletida no eixo y e transposta) no intervalo $A_{0:m,m:m+n}$. A figura 4.4 ilustra essa operação.

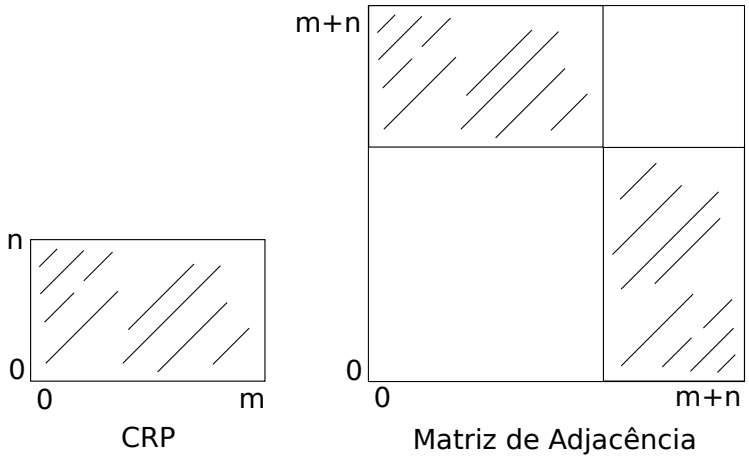
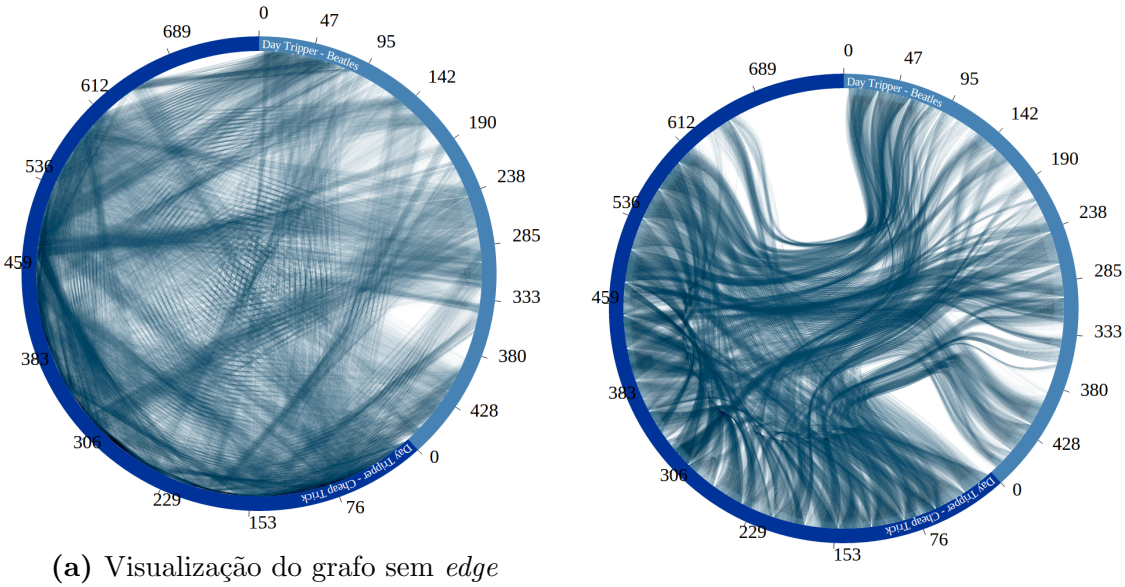


Figura 4.4: Transformação da matriz CRP em uma matriz de adjacência

A matriz de adjacência criada representa um grafo que conecta os trechos semelhantes de duas músicas. A informação temporal provê o posicionamento dos nós do grafo, que devem ser apresentados em sequência (figura 4.5a). Como as arestas ficam muito sobrepostas com sua representação em linhas retas, pode-se reduzir esse problema com um layout do tipo *hierarchical edge bundling* (Holten, 2006) (figura 4.5b). A hierarquia

criada consiste na divisão dos dados por músicas e, em seguida, na divisão das janelas em 30 grupos. A escala do gráfico está em janelas de *embedding*.



(a) Visualização do grafo sem *edge bundling*

(b) Visualização do grafo com *edge bundling*

Figura 4.5: Visualização do CRP da música "*Daytripper*", interpretada por The Beatles e Cheap Trick (Base de dados Covers80 (Ellis, 2007)). Limiar = 75 janelas e elemento estruturante quadrado de tamanho 7.

Percebe-se que mesmo com o *edge bundling*, ainda é difícil encontrar as regiões de similaridade entre os sinais. A única informação visível na figura 4.5b é que a interpretação de Cheap Trick (arco esquerdo, azul escuro) é muito diferente da interpretação de The Beatles após a janela 650.

Torna-se necessária uma nova simplificação na representação de grafos para que as relações de similaridade fiquem mais claras. Para fazer essa simplificação, utiliza-se a matriz dos grupos filtrados (figura 4.3b). Para cada grupo, verifica-se onde a similaridade começa e onde termina (canto inferior esquerdo e superior direito da caixa delimitadora). De posse dessa informação, pode-se gerar visualizações que representam com áreas as regiões semelhantes. A figura 4.6 ilustra uma representação linear das músicas e dos trechos semelhantes. Como ainda há muita sobreposição, o usuário pode escolher uma área de destaque. Na figura, a região de destaque é a ligação do início de Cheap Trick ao final de Beatles com opacidade = 100%.

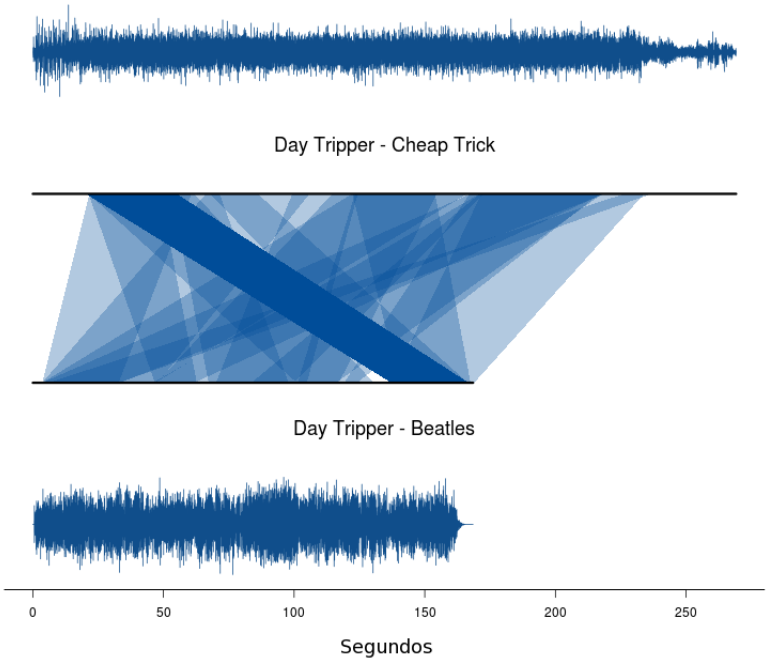


Figura 4.6: Representação em regiões de similaridade da música "*Daytripper*", interpretada por Beatles e Cheap Trick (Base de dados Covers80 (Ellis, 2007)).

Para amenizar o problema da sobreposição, um *layout* circular foi proposto, inspirado na visualização de cordas (figura 4.7) (Bostock, 2012) da biblioteca D^3 (Bostock et al., 2011). A visualização de cordas foi concebida para representar relacionamentos entre grupos de entidades, apresentando cordas (áreas) que ligam porcentagens de uma região com outra. Entretanto, ela possui duas desvantagens:

1. Não permite a sobreposição de cordas no local de origem e de destino (frequente na comparação de músicas);

2. Não apresenta onde ocorre o relacionamento. A posição das cordas é escolhida aleatoriamente.

Uma versão mais flexível do *layout* de cordas foi implementada. Nessa versão, o algoritmo recebe como parâmetro os segundos de início e fim de cada similaridade. Curvas de Bézier quadráticas são desenhadas entre os pontos de início das similaridades e entre os pontos de fim de similaridades, com um ponto de controle no centro do círculo. A figura 4.8 apresenta a visualização proposta com duas versões da música "Daytripper".

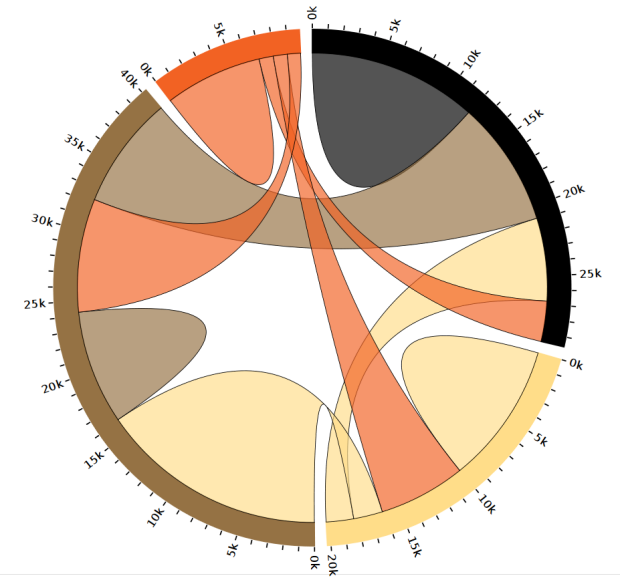


Figura 4.7: Visualização de Cordas do relacionamento entre um grupo de entidades (Bostock, 2012).

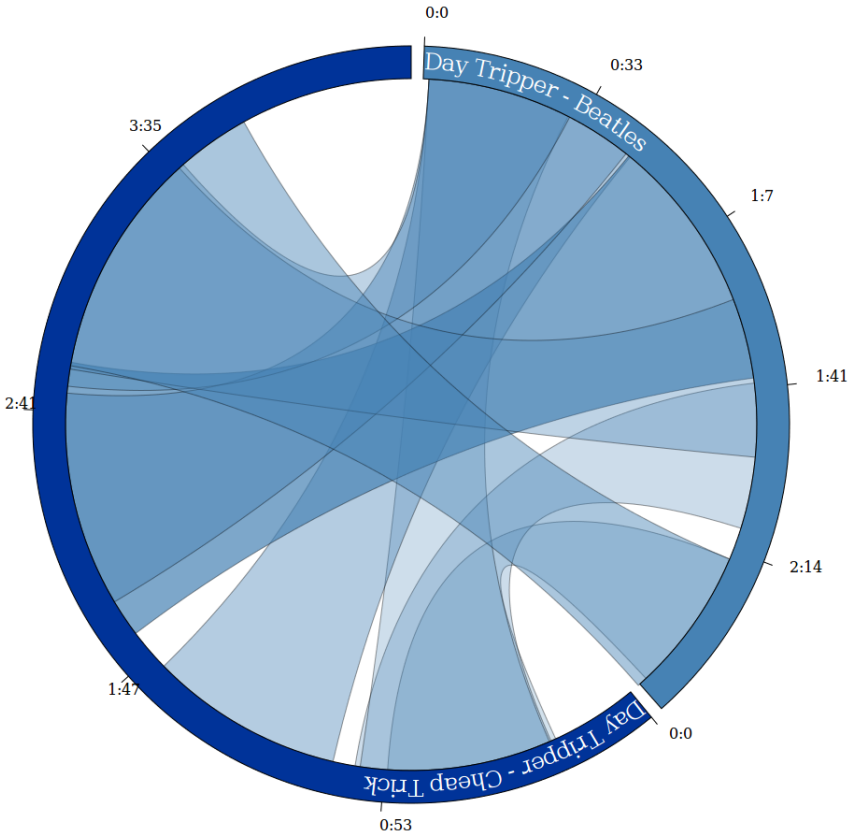


Figura 4.8: Representação em regiões com *layout* circular da música "*Daytripper*", interpretada por Beatles e Cheap Trick (Base de dados Covers80 (Ellis, 2007)).

A próxima etapa no desenvolvimento desta visualização é a adaptação do algoritmo para mostrar a similaridade de mais de duas músicas. Para evitar muitos cruzamentos de arestas na visualização, pode-se ordenar as músicas no layout circular de acordo com sua medida Q_{max}.

4.3 Resultados preliminares

Os algoritmos de extração de características PCP, MFCC e HPCP, e de comparação de séries temporais Q_{max}, descritos no capítulo 2, foram implementados na linguagem R (Team, 2013) com a biblioteca Essentia (Bogdanov et al., 2013).

Os algoritmos de projeção multidimensional Force Scheme (Tejada et al., 2003), LAMP (Joia et al., 2011) e MDS, necessários para a visualização de coleções de músicas, foram estudados e implementados na linguagem R. As características estão sendo extraídas da base de dados Covers80 (Ellis, 2007) e de uma coleção pessoal com 10 versões *cover* de 5 músicas (DB50).

A figura 4.9 apresenta a projeção MDS da base DB50, em que cada cor representa uma classe de músicas *cover*. Os parâmetros para o cálculo da medida Q_{max} estão sendo avaliados para melhorar o resultado da projeção (agrupar músicas *cover* e afastar músicas não *cover*).

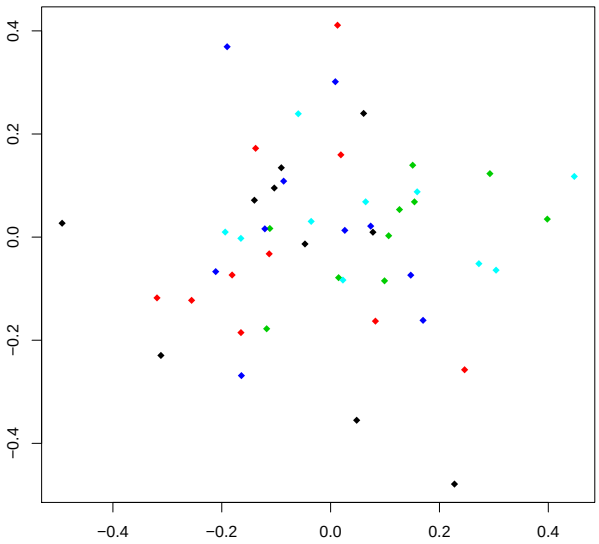


Figura 4.9: Projeção da base de dados DB50 com *Classical Multidimensional Scaling*.

As visualizações de similaridade entre pares de músicas foram implementadas com a linguagem Javascript e a biblioteca de visualização para internet D^3 (Bostock et al., 2011).

O algoritmo de visualização de similaridades foi testado com músicas *cover* e músicas não *cover*. Quando músicas não *cover* são visualizadas, fica claro que poucas partes entre elas são semelhantes.

A figura 4.10 apresenta a visualização de similaridades para duas versões da música "Abracadabra". O limiar utilizado para filtrar as diagonais desta visualização foi de 50 janelas. A figura 4.11 ilustra a mesma visualização para uma versão da música "Abracadabra" com uma versão da música "*Day Tripper*" e limiar de 40 janelas (com um limiar maior, não há regiões de similaridade para apresentar). Nesta visualização, há poucas semelhanças entre as duas músicas.

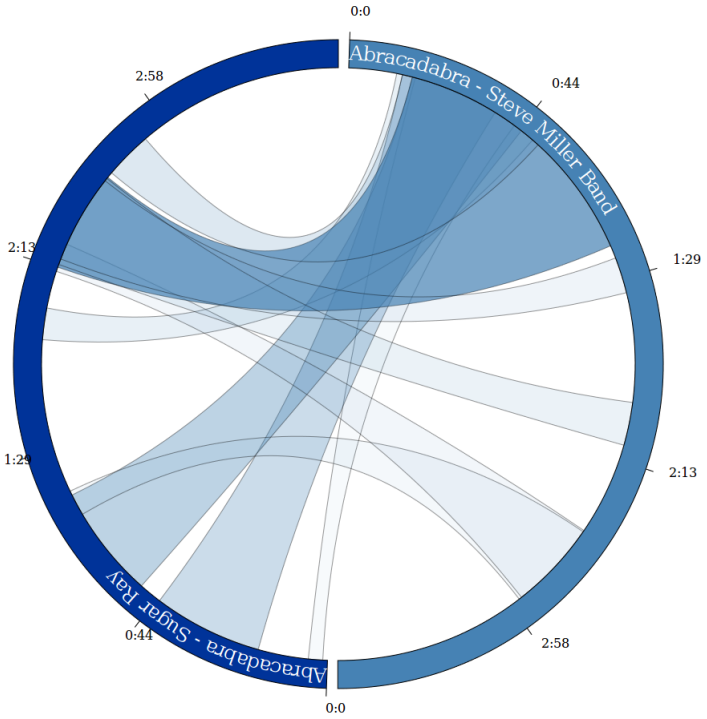


Figura 4.10: Representação em regiões com *layout* circular da música "Abracadabra", interpretada por Steve Miller Band e Sugar Ray (Base de dados Covers80 (Ellis, 2007)).

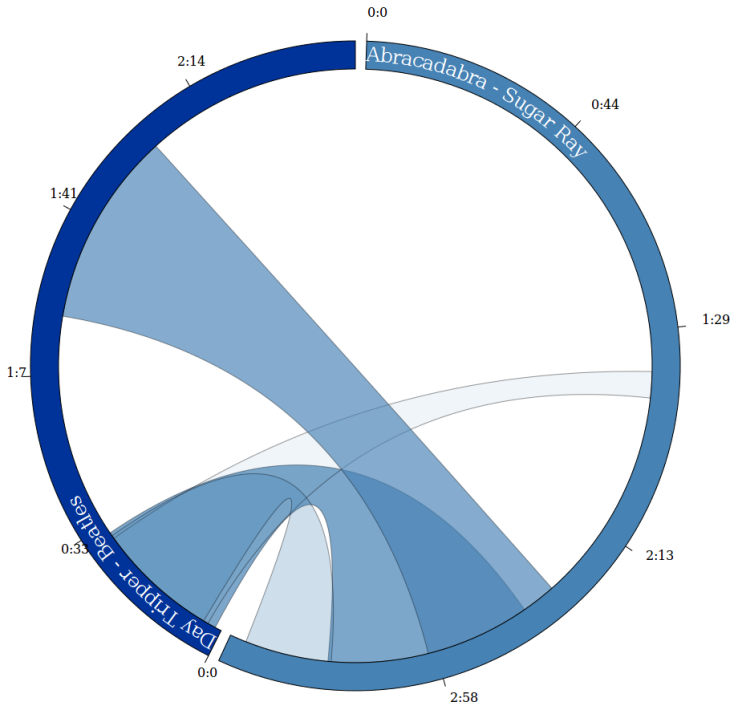


Figura 4.11: Representação em regiões com *layout* circular da música "Abracadabra", Sugar Ray, com "*Daytripper*", The Beatles (Base de dados Covers80 (Ellis, 2007)).

4.4 Atividades e cronograma previsto

As atividades principais previstas para o desenvolvimento do presente trabalho de mestrado são as seguintes:

1. Cumprimento dos créditos das disciplinas exigidos pelo programa;
2. Levantamento bibliográfico das técnicas de extração de características de áudio e de comparação de séries temporais;
3. Redação da monografia de qualificação;
4. Implementação e validação das técnicas de extração de características e de comparação de sinais;
5. Desenvolvimento da técnica de visualização de similaridades entre músicas;
6. Testes e validação da técnica proposta;
7. Escrita da dissertação;
8. Defesa do mestrado.

O cronograma de execução das atividades é apresentado na tabela 4.1.

Tabela 4.1: Cronograma

	2013		2014		2015
Atividade	1º S.	2º S.	1º S.	2º S.	1º S.
1					
2					
3					
4					
5					
6					
7					
8					

Bibliografia

Anglade, A., R. Ramirez e S. Dixon (2009). "First-order logic classification models of musical genres based on harmony". Em: *6th Sound and Music Computing Conference*, pp. 309–314. URL: https://www.eecs.qmul.ac.uk/~simond/pub/2009/Anglade2009-smc.pdf (acesso em 2014).

Bai, M. R. e G.-Y. Shih (2007). "Upmixing and Downmixing Two-channel Stereo Audio for Consumer Electronics". Em: *IEEE Transactions on Consumer Electronics* 53.3, pp. 1011–1019.

Bogdanov, D. et al. (2013). "Essentia: An audio analysis library for music information retrieval". Em: *Proceedings of ISMIR*. URL: http://mtg.upf.edu/system/files/publications/essentia_ismir_2013.pdf (acesso em 2014).

Borgo, R. et al. (2012). "Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications". Em: *Eurographics 2013-State of the Art Reports*. The Eurographics Association, pp. 39–63. URL: http://diglib.eg.org/EG/DL/conf/EG2013/stars/039-063.pdf.abstract.pdf;internal&action=action.digitallibrary.ShowPaperAbstract (acesso em 2014).

Bostock, M. (12 de novembro de 2012). *Chord Diagram*. MBostock's Blog. URL: http://bl.ocks.org/mbostock/4062006 (acesso em 2014).

Bostock, M., V. Ogievetsky e J. Heer (2011). "D3 Data-Driven Documents". Em: *Visualization and Computer Graphics, IEEE Transactions on* 17.12, pp. 2301–2309.

Chen, Q. et al. (2010). "Analysis of Music Representations of Vocal Performance Based on Spectrogram". Em: *2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*. 2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), pp. 1–4.

Cui, W. et al. (2011). "TextFlow: Towards Better Understanding of Evolving Topics in Text". Em: *IEEE Transactions on Visualization and Computer Graphics* 17.12, pp. 2412–2421.

Dalhuijsen, L. e L. van Velthoven (2010). "MusicalNodes: The Visual Music Library". Em: *Proceedings of the 2010 International Conference on Electronic Visualisation and the Arts*. EVA'10. Swinton, UK, UK: British Computer Society, pp. 232–236. URL: http://dl.acm.org/citation.cfm?id=2227180.2227214 (acesso em 13 de novembro de 2013).

Echonest, T. (2013). *The Echonest API*. URL: http://the.echonest.com/ (acesso em 10 de novembro de 2013).

Eckmann, J.-P., S. O. Kamphorst e D. Ruelle (setembro de 1995). "Recurrence Plots of Dynamical Systems". Em: *Turbulence, Strange Attractors And Chaos*. Vol. 16. WORLD SCIENTIFIC, pp. 441–445. URL: http://www.worldscientific.com/doi/abs/10.1142/9789812833709_0030 (acesso em 8 de novembro de 2013).

Ellis, D. P. W. (2007). *The "covers80"cover song data set*. URL: http://labrosa.ee.columbia.edu/projects/coversongs/covers80/. (acesso em 6 de novembro de 2013).

Fujishima, T. (1999). "Realtime Chord Recognition of Musical Sound : a System Using Common Lisp Music". Em: *Proc. ICMC, 1999*, pp. 464–467. URL: http://ci.nii.ac.jp/naid/10013545881/ (acesso em 12 de agosto de 2013).

Furnas, G. W. (1986). "Generalized fisheye views". Em: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '86. New York, NY, USA: ACM, pp. 16–23. URL: http://doi.acm.org/10.1145/22627.22342 (acesso em 11 de novembro de 2013).

Gonzalez, R. C. e R. E. Woods (2008). *Digital image processing*. Upper Saddle River, N.J.: Prentice Hall.

Gouyon, F. e S. Dixon (2005). "A review of automatic rhythm description systems". Em: *Computer music journal* 29.1, pp. 34–54. URL: http://www.mitpressjournals.org/doi/pdf/10.1162/comj.2005.29.1.34 (acesso em 2014).

Gómez, E. G. (2006). "Tonal description of music audio signals". Barcelona: Universitat Pompeu Fabra. URL: http://www.tdx.cat/TDX-0326107-170800/ (acesso em 12 de agosto de 2013).

Hadlak, S., H. Schulz e H. Schumann (2011). "In Situ Exploration of Large Dynamic Networks". Em: *IEEE Transactions on Visualization and Computer Graphics* 17.12, pp. 2334–2343.

Han, W. et al. (2006). "An efficient MFCC extraction method in speech recognition". Em: *2006 IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006.*

Proceedings. 2006 IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006. Proceedings, pp. 145–148.

Hiraga, R., F. Watanabe e I. Fujishiro (2002). "Music learning through visualization". Em: *Second International Conference on Web Delivering of Music, 2002. WEDELMUSIC 2002. Proceedings*. Second International Conference on Web Delivering of Music, 2002. WEDELMUSIC 2002. Proceedings, pp. 101–108.

Holten, D. (2006). "Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data". Em: *Visualization and Computer Graphics, IEEE Transactions on* 12.5, pp. 741–748. URL: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4015425` (acesso em 2014).

Hunt, M., M. Lennig e P. Mermelstein (1980). "Experiments in syllable-based recognition of continuous speech". Em: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '80*. Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '80. Vol. 5, pp. 880–883.

Joia, P. et al. (2011). "Local Affine Multidimensional Projection". Em: *IEEE Transactions on Visualization and Computer Graphics* 17.12, pp. 2563–2571.

Keogh, E. e C. A. Ratanamahatana (1 de março de 2005). "Exact indexing of dynamic time warping". Em: *Knowledge and Information Systems* 7.3, pp. 358–386. URL: `http://link.springer.com/article/10.1007/s10115-004-0154-9` (acesso em 9 de novembro de 2013).

Kostka, S. et al. (1995). *Tonal Harmony with an Introduction to Twentieth-Ce*. McGraw-Hill.

Lamere, P. (2012). *Infinite Jukebox*. Versão 1.0. URL: `http://labs.echonest.com/Uploader/index.html` (acesso em 8 de novembro de 2013).

Lerch, A. (2012). *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*. John Wiley & Sons. 248 pp.

Lillie, A. S. (2008). "MusicBox: Navigating the space of your music". Massachusetts Institute of Technology.

Malinowski, S. (2013). *The Music Animation Machine*. URL: `http://www.musanim.com/` (acesso em 7 de novembro de 2013).

Manning, C. D., P. Raghavan e H. Schütze (2008). *Introduction to information retrieval*. Vol. 1. Cambridge University Press Cambridge.

McKay, C., I. Fujinaga e P. Depalle (2005). "jAudio: A feature extraction library". Em: *Proceedings of the International Conference on Music Information Retrieval*, pp. 600–3. URL: `http://www.music.mcgill.ca/~cmckay/papers/musictech/jAudio_ISMIR_2005.pdf` (acesso em 6 de dezembro de 2013).

Muelder, C. e K.-L. Ma (2008). "Rapid Graph Layout Using Space Filling Curves". Em: *IEEE Transactions on Visualization and Computer Graphics* 14.6, pp. 1301–1308.

Muelder, C., T. Provan e K.-L. Ma (2010). "Content Based Graph Visualization of Audio Data for Music Library Navigation". Em: *2010 IEEE International Symposium on Multimedia (ISM)*. 2010 IEEE International Symposium on Multimedia (ISM), pp. 129–136.

Müller, M. (2007). *Information retrieval for music and motion*. Springer. 313 pp.

Orio, N. (2006). "Music Retrieval: A Tutorial and Review". Em: *Foundations and Trends in Information Retrieval* 1.1, pp. 1–96. URL: http://www.nowpublishers.com/product.aspx?product=INR&doi=1500000002 (acesso em 7 de novembro de 2013).

O'Shaughnessy, D. (1987). *Speech communication: human and machine*. Addison-Wesley series in electrical engineering. Reading, Mass: Addison-Wesley Pub. Co. 568 pp.

Pampalk, E. (2001). "Islands of music: Analysis, organization, and visualization of music archives". URL: http://www.ofai.at/~elias.pampalk/publications/pam_oegai03.pdf (acesso em 13 de novembro de 2013).

Pampalk, E., A. Rauber e D. Merkl (2002). "Content-based organization and visualization of music archives". Em: *Proceedings of the tenth ACM international conference on Multimedia*. MULTIMEDIA '02. New York, NY, USA: ACM, pp. 570–579. URL: http://doi.acm.org/10.1145/641007.641121 (acesso em 12 de agosto de 2013).

Paulovich, F. et al. (2008). "Least Square Projection: A Fast High-Precision Multidimensional Projection Technique and Its Application to Document Mapping". Em: *IEEE Transactions on Visualization and Computer Graphics* 14.3, pp. 564–575.

Paulovich, F. et al. (2011). "Piece wise Laplacian-based Projection for Interactive Data Exploration and Organization". Em: *Computer Graphics Forum* 30.3, pp. 1091–1100. URL: http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8659.2011.01958.x/abstract (acesso em 4 de dezembro de 2013).

Rodriguez, M. (2013). *iTunes Store bate novo recorde com 25 bilhões de músicas vendidas*. Apple. URL: http://www.apple.com/br/pr/library/2013/02/06iTunes-Store-Sets-New-Record-with-25-Billion-Songs-Sold.html (acesso em 12 de novembro de 2013).

Scaringella, N., G. Zoia e D. Mlynek (2006). "Automatic genre classification of music content: a survey". Em: *Signal Processing Magazine, IEEE* 23.2, pp. 133–141. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1598089 (acesso em 2014).

Serra, J. et al. (2009). "Unsupervised Detection of Cover Song Sets: Accuracy Improvement and Original Identification." Em: *ISMIR*, pp. 225–230. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.182.597&rep=rep1&type=pdf (acesso em 2014).

Serra, X. (1997). "Musical sound modeling with sinusoids plus noise". Em: *Musical signal processing*, pp. 91–122.

Serrà, J., X. Serra e R. G. Andrzejak (15 de setembro de 2009). "Cross recurrence quantification for cover song identification". Em: *New Journal of Physics* 11.9, pp. 1–20. URL: http://stacks.iop.org/1367-2630/11/i=9/a=093017?key=crossref.60913d34a4637672a628c7d815fa5d22 (acesso em 17 de outubro de 2013).

Stevens, S. S., J. Volkmann e E. B. Newman (1937). "A Scale for the Measurement of the Psychological Magnitude Pitch". Em: *The Journal of the Acoustical Society of America* 8.3, pp. 185–190. URL: http://scitation.aip.org/content/asa/journal/jasa/8/3/10.1121/1.1915893 (acesso em 13 de novembro de 2013).

Team, R. C. (2013). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. URL: http://www.R-project.org/.

Tejada, E., R. Minghim e L. G. Nonato (1 de dezembro de 2003). "On Improved Projection Techniques to Support Visual Exploration of Multi-Dimensional Data Sets". Em: *Information Visualization* 2.4, pp. 218–231. URL: http://ivi.sagepub.com/content/2/4/218 (acesso em 14 de agosto de 2013).

Tolonen, T. e M. Karjalainen (2000). "A computationally efficient multipitch analysis model". Em: *Speech and Audio Processing, IEEE Transactions on* 8.6, pp. 708–716. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=876309 (acesso em 2014).

Torrens, M., P. Hertzog e J. L. Arcos (2004). "Visualizing and Exploring Personal Music Libraries." Em: *ISMIR*. The International Society for Music Information Retrieval. URL: http://www.cs.fiu.edu/~lli003/Music/mv/5.pdf (acesso em 13 de novembro de 2013).

Tzanetakis, G. e P. Cook (julho de 2002). "Musical genre classification of audio signals". Em: *IEEE Transactions on Speech and Audio Processing* 10.5, pp. 293–302. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1021072 (acesso em 12 de agosto de 2013).

Tzanetakis, G., A. Ermolinskyi e P. Cook (1 de junho de 2003). "Pitch Histograms in Audio and Symbolic Music Information Retrieval". Em: *Journal of New Music Research* 32.2, pp. 143–152. URL: http://www.tandfonline.com/doi/abs/10.1076/jnmr.32.2.143.16743 (acesso em 2014).

Ware, C. (2013). *Information visualization: perception for design*. Third edition. Interactive technologies. Waltham, MA: Morgan Kaufmann. 512 pp.

Wattenberg, M. (2002). *Shape of Sound*. URL: http://www.turbulence.org/Works/song/ (acesso em 10 de novembro de 2013).

Withall, M., I. Phillips e D. Parish (2007). "Network visualisation: a review". Em: *IET Communications* 1.3, pp. 365–372.

Yknk (2012). *MIDITrail*. Versão 1.2.0. URL: `http://en.sourceforge.jp/projects/miditrail/` (acesso em 5 de novembro de 2013).

Zolzer, U. (2008). *Digital audio signal processing*. Chichester, U.K.: Wiley.