Relatório Final
MAC0215

Felipe Silva Felix
Orientador: Marcelo Gomes de Queiroz

28 de Novembro de 2016

Conteúdo

1 Introdução

1.1 Descrição do problema

Audio thumbnail ou sumarização de áudio é um problema inserido na área de recuperação de informação musical (*Musical Information Retrieval - MIR*) [11]. O problema consiste em encontrar o segmento de áudio mais repetitivo numa gravação musical. O problema de *audio thumbnail* pode ser visto, também, como uma subtarefa da análise de estrutura musicais que é um tópico de pesquisa dentro de *MIR* que tem como principal objetivo dividir uma gravação musical em segmentos temporais correspondentes às suas partes musicais.

Uma motivação para a construção de *thumbnails* de áudio encontra-se no acesso a grandes coleções de músicas. Num sistema de busca e recuperação de gravações, o usuário deveria poder escutar as gravações de forma *resumida*, então, com base nesse *resumo* o usuário poderia decidir escutar a gravação inteira ou passar para a próxima gravação. Num sistema de busca por fotos, por exemplo, a busca pode ser facilitada utilizando versões menores, com amostras diminuídas da imagem original: essas versões são conhecidas como *thumbnail* da imagem original. Para gravações musicais, procura-se um excerto reduzido que seja útil para identificar a gravação original. Na música popular, por exemplo, um bom resumo de uma gravação seria um refrão que se repete em grande parte da música. Na música clássica, um bom *thumbnail* poderia capturar um tema ou seção que se repete muitas vezes durante a gravação.

1.2 Metodologia

A metodologia adotada nessa pesquisa pode ser descrita sucintamente através das seguintes tarefas:

- Estudo dos conceitos teóricos fundamentais necessários [11];
- Estudo dos artigos selecionados [4, 6, 12];
- Implementação das técnicas descritas;
- Avaliação e comparação das técnicas.

Durante o semestre nosso estudo não se limitou apenas às técnicas de *Audio Thumbnail*. Além dos conceitos necessários de computação musical para entender as técnicas, estudamos campos diversos em *Music Information Retrieval*, participando como ouvinte da disciplina de pós-graduação MAC6917 (Tópicos em Computação Sonora e Musical), que utilizou o livro *Fundamentals of Music Processing* [11] como livro-texto. Também participamos de reuniões individuais e coletivas no grupo de pesquisas em Computação Musical do IME/USP.

O detalhamento técnico dessas tarefas e o cronograma com sua distribuição no semestre serão detalhados nas seções seguintes.

2 Técnicas para sumarização de áudio estudadas

2.1 Conceitos fundamentais

As três técnicas estudadas para *Audio thumbnail* são baseadas em análise de similaridade entre trechos de um mesmo arquivo de áudio. Para tal, utilizam uma estrutura chamada “Matriz de auto-similaridade", que definiremos aqui por ser comum às três técnicas.

Inicialmente, transformamos a gravação musical em uma sequência:

$$X \coloneqq (x_1, x_2, ..., x_N) \tag{1}$$

de vetores característicos $x_n \in \mathcal{F}, 1 \leq n \leq N$, onde $\mathcal{F}$ denota um espaço de características adequado e n é o índice do segmento temporal de onde foi obtido o vetor característico x_n. Então, com base em uma medida de similaridade $s : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$, definimos a *matriz de auto-similaridade* $\mathcal{S} \in \mathbb{R}^{N \times N}$ por $\mathcal{S}(n, m) \coloneqq s(x_n, x_m)$, para $1 \leq n, m \leq N$. Note que $\mathcal{S}(n, m)$ mede a similaridade entre dois momentos distintos n e m de um mesmo sinal sonoro. Esta definição é geral o bastante para contemplar diversos tipos de espaço de características e medidas de similaridade; os algoritmos

estudados adotam o espaço de vetores de *croma*, como veremos nas próximas seções, e medem a similaridade através do produto interno, como discutimos a seguir.

Utilizando o produto interno como medida de similaridade ($s(x_n, x_m) = |x_n^T x_m|$), no caso dos vetores normalizados em relação à norma Euclidiana, teremos que $\mathcal{S}(x_n, x_m) \in [0, 1]$ e $\mathcal{S}(x_n, x_n) = 1$ para todo $m, n \in [1, N]$. Note que $s(x_n, x_m) = |\cos(\theta_{n,m})|$ onde $\theta_{n,m}$ é o ângulo entre os vetores x_n e x_m, razão pela qual essa medida também é conhecida como "distância" do cosseno. Em geral, padrões recorrentes no espaço de características podem ser visualizados como blocos de alto valor na matriz. De forma similar, se os vetores de características contiverem sequências repetitivas, notaremos diagonais secundárias paralelas à diagonal principal com valores altos.

2.2 O algoritmo de Bartsch e Wakefield

O algoritmo de Bartsch e Wakefield é descrito no artigo *Audio thumbnailing of popular music using chroma-based representations* [4]. Grande parte desse artigo é reservada para tratar do conceito de *croma*, que se baseia no atributo cíclico das percepções de altura de uma nota, também conhecido como equivalência de oitavas. Na figura abaixo percebe-se que conforme a altura aumenta, por exemplo de D1 para D2, sua posição se move ao longo da hélice, girando *cromaticamente*[1] por todas as classes de altura (D,D#,E, etc) até retornar à classe de altura inicial (D) um ciclo (uma oitava) acima do ponto de partida.

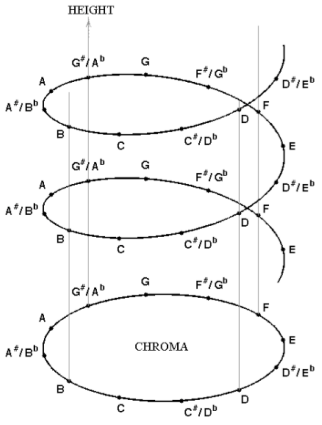

Figura 1: Ilustração da hélice de percepção de altura. A dimensão vertical é a altura da nota, enquanto a dimensão angular é o croma.

Um *vetor de croma* particiona a energia espectral de um segmento sonoro em 12 valores que representam a energia acumulada em cada uma das classes de altura; por exemplo, a partir da referência A4 = 440 Hz obtemos o valor do croma A somando os valores de energia nas frequências A0=27.5, A1=55, A2=110, A3=220, A4=440, A5=880, A6=1760, A7=3520, A8=7040 e A9=14080 Hz (todos os As no espectro audível entre 20 e 20000 Hz). A partir dessa característica, constrói-se o *cromagrama*, que nada mais é que uma sequência de vetores de croma distribuidos em relação ao tempo (vide Figura 2 abaixo).

Os autores então definem uma matriz S de auto-similaridade como descrito anteriormente com a característica croma. Essa matriz é então rotacionada para que as diagonais fiquem orientadas verticalmente. Isso pode ser representado por essa equação:

$$T_{i,j} = \sum_k \mathcal{S}_{i+k,i+j+k} w(k) \tag{2}$$

onde $w(k)$ é uma função de janelamento. O elemento $T_{i,j}$ indica a similaridade entre as características do segmento que começa no i-ésimo quadro com o segmento que começa no (i+j)-ésimo quadro.

A seleção do *thumbnail* é dada pelo valor máximo em T de acordo com algumas restrições, associadas a um atraso mínimo e a um limite no tempo inicial da seleção, para evitar o *fade-out* presente em algumas gravações.

[1] o termo se refere à escala de 12 semitons por oitava, chamada de escala cromática.

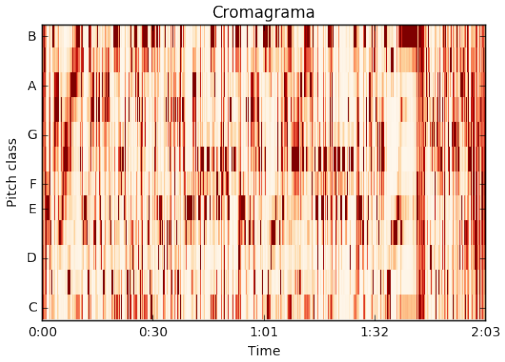

Figura 2: Exemplo de cromagrama para a música: *"La vie en rose"*

2.3 O algoritmo de Cooper e Foote

O algoritmo de Cooper e Foote é descrito no artigo *Automatic Music Summarization via Similarity Analysis* [6]. Dado $X \coloneqq (x_1, x_2, ..., x_N)$ uma sequência de vetores característicos, definimos a matriz de auto-similaridade $\mathcal{S} \in \mathbb{R}^{N \times N}$:

$$\mathcal{S}(n, m) \coloneqq s(x_n, x_m), 1 \leq n, m \leq N \tag{3}$$

Dado um segmento $\alpha = [q, r], 1 \leq q < r \leq N$, definimos a similaridade média do segmento:

$$\bar{\mathcal{S}}(q, r) = \frac{1}{N(r-q)} \sum_{m=q}^{r} \sum_{n=1}^{N} \mathcal{S}(m, n) \tag{4}$$

Uma interpretação simples de $\bar{\mathcal{S}}(q, r)$ corresponde à média das colunas da matriz de auto-similaridade sobre o intervalo $q, \ldots, r$. Então, intervalos com alta similaridade terão maior média $\bar{\mathcal{S}}(q, r)$.

Os autores então definem $\mathcal{Q}_L(i)$ como:

$$\mathcal{Q}_L(i) = \bar{\mathcal{S}}(i, i + L), i = 1, \cdots, N - L. \tag{5}$$

O melhor ponto inicial para o excerto para os autores é o tempo q_L^* que maximiza $\mathcal{Q}_L(i)$:

$$q_L^* = \underset{1 \leq i \leq N-L}{\arg\max}\ \mathcal{Q}_L(i), \tag{6}$$

e o melhor *thumbnail* é o excerto de tamanho L que inicia em q_L^* e termina em $q_L^* + L$.

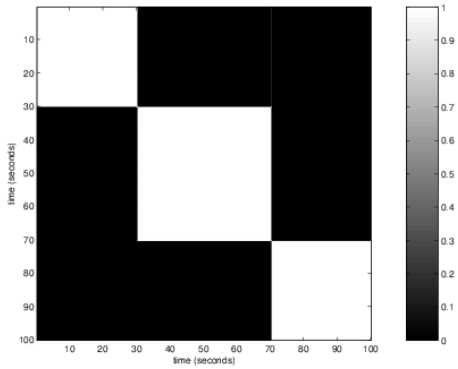

Figura 3: Exemplo de matriz de auto-similaridade para um sinal sintetizado formado por 30 segundos de uma senoide de frequência 1 kHz, 40 segundos de 500 Hz, e 30 segundos de 2 kHz.

A figura 3 traz uma matriz de similaridade contruída a partir de um sinal sintetizado contendo 30 segundos de uma senoide de frequência 1 kHz, 40 segundos de uma senoide de 500 Hz, e 30

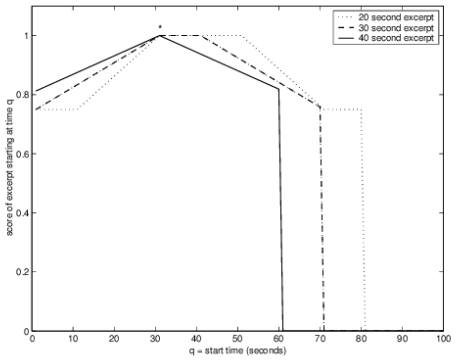

Figura 4: $Q_L(i)$ computado para a matriz de auto-similaridade acima com L = 20, 30 e 40 segundos.

segundos de uma senoide de 2 kHz. para *thumbnails* com tamanhos L = 20, 30 e 40 segundos. Todos mostram um pico em $q^* = 30$, que é o instante inicial para a senoide de 500 Hz (que tem duração de 40s, e que seria a escolha trivial para um sumário do sinal sintetizado).

2.4 O algoritmo de Müller, Grosche e Jiang

O algoritmo de Müller, Grosche e Jiang é descrito no artigo *A Segment-Based Fitness Measure for Capturing Repetitive Structures of Music Recordings* [12]. A contribuição principal desse artigo é apresentar uma medida de aptidão $\varphi(\alpha)$ que atribui a cada segmento α de áudio um valor $\varphi(\alpha)$ que captura sua *precisão* (o quão similar o dado segmento é em relação a segmentos repetidos) e sua *cobertura* (o quanto da gravação musical é coberto pelo segmento e suas repetições).

Notações e definições

Dado $\mathcal{S} \in \Re^{NxN}$ uma matriz de similaridade, definimos o seguinte:

- Um *segmento* α é definido como um subconjunto $\alpha = [s : t] \subseteq [1 : N]$.
- Uma tupla $p = (n, m) \in [1 : N]^2$ é chamada uma *célula* de $\mathcal{S}$.
- O valor $\mathcal{S}(n, m)$ é chamado *pontuação* da célula p.
- Um **caminho** de tamanho L é uma sequência $\pi = (p_1, ..., p_L)$ de células $p_l = (n_l, m_l)$ para $l \in [1 : L]$ tal que $p_{l+1} - p_l \in \Sigma$, onde Σ corresponde a um conjunto de passos admissíveis, por exemplo: $\Sigma = \{(1, 1), (1, 2), (2, 1)\}$.
- A **pontuação** $\mu(\pi)$ do caminho π é definida como

$$\mu(\pi) = \sum_{l=1}^{L} \mathcal{S}(n_l, m_l). \tag{7}$$

Expandindo a definição de caminhos, definimos uma **Família de Caminhos** sobre α como um conjunto $\mathcal{P} \coloneqq \{\pi_1, \pi_2, \cdots, \pi_K\}$ tal que:

- as projeções π_k^2 dos caminhos π_k na segunda componente são todas iguais a α, $\forall k \in [1 : K]$.
- a coleção de projeções $\{\pi_k^1 | k \in [1 : K]\}$ na primeira componente satisfaz $\pi_i^1 \cap \pi_j^1 = \emptyset$, $\forall i \neq j$.
- Pontuação da **família de caminhos** $\mathcal{P}$ sobre α:

$$\mu(\mathcal{P}) = \sum_{k=1}^{K} \mu(\pi_k). \tag{8}$$

Seja $\mathcal{P}^*$ a família de caminhos sobre α que tem pontuação máxima: $\mathcal{P}^* \coloneqq \arg\max_{\mathcal{P}} \mu(\mathcal{P})$. Definimos então a pontuação do segmento α como:

$$\mu(\alpha) \coloneqq \mu(\mathcal{P}^*). \tag{9}$$

O cálculo da família $\mathcal{P}^*$ de caminhos que maximiza $\mu(\mathcal{P})$ para um α fixado pode ser feito de forma eficiente com uma pequena variação no algoritmo de DTW [11].

Lembremos também que desejamos indicar na medida φ o quanto o segmento α cobre a gravação, de modo que segmentos que cobrem uma grande parte da gravação terão φ maior do que os segmentos que cobrem uma pequena parte da gravação. Então, definimos a cobertura γ para uma família de segmentos $\mathcal{P}$ como:

$$\gamma(\mathcal{P}) \coloneqq \cup_{k\in[1:K]}\pi_k^1 \subseteq [1:N]. \tag{10}$$

Da equação (5) definimos a medida de cobertura normalizada:

$$\bar{\gamma}(\mathcal{P}) \coloneqq \frac{|\gamma(\mathcal{P})| - |\alpha|}{N}. \tag{11}$$

Enfim, chegamos a definição da medida desejada:

$$\varphi(\mathcal{P}) \coloneqq 2 \cdot \frac{\bar{\mu}(\mathcal{P}) \cdot \bar{\gamma}(\mathcal{P})}{\bar{\gamma}(\mathcal{P}) + \bar{\mu}(\mathcal{P})}; \tag{12}$$

Essa medida combina a cobertura e a pontuação da família de caminhos $\mathcal{P}$ através de uma média harmônica das medidas isoladas.

Para medir a qualidade de um segmento α, consideramos a medida acima tomada em uma família de caminhos sobre α ótima em relação ao DTW:

$$\varphi(\alpha) \coloneqq \varphi(\mathcal{P}^*). \tag{13}$$

Então, teremos a seguinte definição para a solução do problema de *audio thumbnail*:

$$\alpha^* \coloneqq \arg\max_{\alpha} \varphi(\alpha) \tag{14}$$

3 Descrição das atividades

3.1 Reuniões semanais do grupo de Computação Musical no IME-USP

O aluno participou de todas as reuniões do grupo de computação musical do IME, que ocorrem semanalmente. Nas reuniões são tratados assuntos técnicos sobre a pesquisa de cada aluno do grupo, e periodicamente cada aluno faz uma apresentação mais longa sobre algum resultado ou artigo relacionado à sua pesquisa. O aluno apresentou o artigo *Automatic Music Summarization via Similarity Analysis* [6] para o grupo no dia 4 de outubro.

3.2 MAC6917 - *Topics in Sound and Music Computing: Music Information Retrieval*

O aluno também cursou a disciplina *Topics in Sound and Music Computing: Music Information Retrieval* ministrada pelo prof. Marcelo Queiroz. As aulas ocorreram do dia 26/7 ao dia 6/10, com 5h de aula por semana. O curso foi baseado no livro: *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*[11]. Semanalmente discutíamos em sala algum artigo selecionado dos artigos premiados na ISMIR (principal conferência de *MIR*).

A avaliação da disciplina consistiu em um seminário aberto para a comunidade IME sobre um dos artigos selecionados e na realização de um experimento também baseado num outro artigo selecionado. O aluno apresentou junto com sua dupla um seminário [2] baseado no artigo: *A Segment-Based Fitness Measure for Capturing Repetitive Structures of Music Recordings* [12]. E realizou o experimento baseado no artigo *Beyond NMF: Time-Domain Audio Source Separation without Phase Reconstruction* [13] sobre separação de fontes, um problema clássico de recuperação de informação musical.

O aluno também acompanhou o curso online *Audio Signal Processing for Music Applications*[1]. O curso é composto por aulas teóricas em que foram abordados temas clássicos em computação

musical como: transformada discreta de Fourier e suas propriedades, modelo senoidal e harmônico, entre outros. Houveram tarefas para entrega *online* que foram implementações em *Python* de conceitos apresentados nas aulas teóricas.

3.3 Cronograma

Semana 1. 15/08 - 19/08

- Estudo do capítulo 2 do livro [11]: "Fourier Analysis of Signals". Primeiras intuições sobre a transformada de Fourier.
- Leitura inicial do artigo *A Segment-Based Fitness Measure for Capturing Repetitive Structures of Music Recordings* [12].
- Nas aulas de MAC6917 foi discutido o capítulo 3 do livro [11] sobre sincronização musical, e foi apresentado o algoritmo de DTW (*Dynamic time warping*).
- Discussão do artigo: *Taste Space Versus the World: an Embedding Analysis of Listening Habits and Geography* [10].

Semana 2. 22/08 - 26/08

- Estudo do capítulo 4 do livro de Müller [11]: "Music Structure Analysis". Conceituação de parte musical, segmento musical e matriz de auto-similaridade. Técnicas de aprimoramento da matriz: *path smoothing*, *thresholding*.
- Discussão do artigo *A MIREX Meta-analysis of Hubness in Audio Music Similarity* [7] na aula de MAC6917.

Semana 3. 29/08 - 03/09

- Estudo e preparação do seminário [2] sobre o artigo *A Segment-Based Fitness Measure for Capturing Repetitive Structures of Music Recordings* [12].
- Continuação do estudo do capítulo 4 do livro [11]. Conceitos estudados: *audio thumbnail, novelty based segmentation, precision, recall, F-measure*.
- Discussão do artigo: *Solving Misheard Lyric Search Queries Using a Probabilistic Model of Speech Sounds* [9].

Semana 4. 05/09 - 09/09

- Introdução ao conceito de NMF (*Nonnegativa Matrix Factorization*) para separação de fontes.
- Estudo do capítulo 5 do livro de Müller [11]: *Chord Recognition*.
- Discussão do artigo: *Musical Instrument Recognition in Polyphonic Audio Using Source-Filter Model for Sound Separation* [8].

Semana 5. 12/09 - 16/09

- Estudo do artigo: "Automatic Music Summarizantion via Similarity Analysis." - Matthew Cooper and Jonathan Foote.
- Apresentação do seminário [2] sobre o artigo: "A Segment-Based Fitness Measure for Capturing Repetitive Structures of Music Recordings" [12], como avaliação da matéria MAC6917: *Topics in Sound and Music Computing - Music Information Retrieval*.
- Discussão do artigo: *A Distributed Model For Multiple-Viewpoint Melodic Prediction*[5].

Semana 6. 18/09 - 24/09

- Início da implementação. Construção e visualização da matriz de auto-similaridade.
- Estudo do capítulo 6 do livro de Müller [11]: *Tempo and Beat Tracking*.

Semana 7. 26/09 - 30/09

- Início do curso: *Audio Signal Processing for Music Applications*. Primeira semana completa.
- Apresentação do artigo *Automatic Music Summarizantion via Similarity Analysis*[6], para o grupo de computação musical.
- Preparação do experimento para a matéria MAC6917.

Semana 8. 03/10 - 07/10

- Término da segunda semana do curso: *Audio Signal Processing for Music Applications*. Conceitos estudados: DFT, e implementação da DFT em python.
- Implementação da técnica apresentada por Matthew Cooper e Jonathan Foote no artigo "Automatic Music Summarizantion via Similarity Analysis". O progresso da implementação pode ser acompanhado aqui: `https://github.com/fsfelix/audio-thumbnailing`.
- Apresentação de experimento baseado no artigo *Beyond NMF: Time-Domain Audio Source Separation without Phase Reconstruction*[13], como avaliação da matéria MAC6917.

Semana 9. 10/10 - 14/10

- Término da terceira semana do curso: *Audio Signal Processing for Music Applications*. Conceitos estudados: Propriedades da DFT, FFT.
- Leitura do artigo *Audio thumbnailing of popular music using chroma-based representations*[4].

Semana 10. 17/10 - 21/10

- Leitura do artigo: *To catch a chorus: Using chroma-based representations for audio thumbnailing*[3].

Semana 11. 24/10 - 28/10

- Início da implementação da técnica descrita no artigo *A Segment-Based Fitness Measure for Capturing Repetitive Structures of Music Recordings*[12]. Problema para implementar o algoritmo para calcular a família de caminhos com score máximo (DTW modificado).

Semana 12. 31/10 - 04/11

- Continuação da implementação da técnica descrita no artigo *A Segment-Based Fitness Measure for Capturing Repetitive Structures of Music Recordings*[12]. O algoritmo para encontrar famílias de pontos com *score* máximo foi implementado.

Semana 13. 7/11 - 11/11

- Início da elaboração do pôster para a disciplina de MAC0215.
- Testes para a técnica de Müller [12], um *bug* foi encontrado, algumas famílias de caminho não possuem projeção de tamanho α no eixo x.

Semana 14. 14/11 - 18/11

- Continuação do pôster.
- Início do relatório para a matéria MAC0215.

Seminários assistidos

Durante o semestre, o aluno assistiu vários seminários de computação musical, especificamente na área de MIR.

- Sobre o Uso de Codificações Esparsas Relativas ao Tempo em Música (Arthur Tofani e Thilo Koch).
- Aprendizagem não-supervisionada de características esparsas para classificação escalável de áudio (Marcio Masaki Tomiyoshi e Roberto Piassi Passos Bodo).

- Acompanhamento musical em tempo real utilizando múltiplas performances como referência (Alessandro Palmeira e Itai Soares).
- *Score Analyser*: Determinando o nível de dificuldade de uma partitura para o aprendizado on-line de instrumentos automaticamente (Guilherme Feulo e Guilherme Jun).
- Um modelo probabilístico simples para taggeamento de música (Rodrigo Borges e Shayenne Moura).

4 Resultados

Apesar de termos estudados três técnicas para o problema descrito, implementamos duas técnicas descritas aqui em 2.4 e 2.5. Tivemos dificuldades em implementar o algoritmo modificado de *DTW* para o cálculo das famílias de caminhos com *score* máximo, o que tomou bastante tempo e nos impediu de implementar a terceira técnica descrita em 2.3. Obtivemos resultados coerentes, com vantagem para a técnica desenvolvida por Müller *et al.*[12]. Não nos limitamos apenas no problema de *Audio Thumbnail*, como pode ser visto no cronograma, estudamos também tópicos centrais em computação musical e em MIR.

As implementações feitas durante o semestre podem ser vistas aqui: `https://github.com/fsfelix/audio-thumbnailing`.

5 Trabalho futuro

Desejamos ainda implementar mais uma técnica apresentada no artigo: *Audio thumbnailing of popular music using chroma-based representations* [4]. Também desejamos implementar o método de avaliação apresentado em [11] e então comparar o resultado das três técnicas para um maior número de gravações musicais.

Referências

[1] *Audio signal processing for music applications*, `https://www.coursera.org/learn/audio-signal-processing`, Acessado: 28-11-2016.

[2] *Seminário*, `http://compmus.ime.usp.br/pt-br/node/534`, Acessado: 23-11-2016.

[3] Mark A Bartsch and Gregory H Wakefield, *To catch a chorus: Using chroma-based representations for audio thumbnailing*, Applications of Signal Processing to Audio and Acoustics, 2001 IEEE Workshop on the, IEEE, 2001, pp. 15–18.

[4] ———, *Audio thumbnailing of popular music using chroma-based representations*, IEEE Transactions on multimedia **7** (2005), no. 1, 96–104.

[5] Srikanth Cherla, Tillman Weyde, Artur S d'Avila Garcez, and Marcus Pearce, *A distributed model for multiple-viewpoint melodic prediction.*, ISMIR, 2013, pp. 15–20.

[6] Matthew L Cooper and Jonathan Foote, *Automatic music summarization via similarity analysis.*, ISMIR, 2002.

[7] Arthur Flexer, Dominik Schnitzer, and Jan Schlüter, *A mirex meta-analysis of hubness in audio music similarity.*, ISMIR, 2012, pp. 175–180.

[8] Toni Heittola, Anssi Klapuri, and Tuomas Virtanen, *Musical instrument recognition in polyphonic audio using source-filter model for sound separation.*, ISMIR, 2009, pp. 327–332.

[9] Hussein Hirjee and Daniel G Brown, *Solving misheard lyric search queries using a probabilistic model of speech sounds.*, ISMIR, 2010, pp. 147–152.

[10] Joshua L Moore, Thorsten Joachims, and Douglas Turnbull, *Taste space versus the world: an embedding analysis of listening habits and geography.*, ISMIR, 2014, pp. 439–444.

[11] Meinard Müller, *Fundamentals of music processing: Audio, analysis, algorithms, applications*, Springer, 2015.

[12] Meinard Müller, Peter Grosche, and Nanzhu Jiang, *A segment-based fitness measure for capturing repetitive structures of music recordings.*, ISMIR, Citeseer, 2011, pp. 615–620.

[13] Kazuyoshi Yoshii, Ryota Tomioka, Daichi Mochihashi, and Masataka Goto, *Beyond nmf: Time-domain audio source separation without phase reconstruction.*, ISMIR, 2013, pp. 369–374.