Ana Corina Martins Calhau

PESQUISA DE OBRAS MUSICAIS POR CANTO

Dissertação no âmbito do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores, ramo de especialização em Telecomunicações, orientada pelo Professor Doutor Fernando Santos Perdigão e Dirceu Silva apresentada à Faculdade de Ciências e Tecnologia da Universidade de Coimbra no Departamento de Engenharia Eletrotécnica e de Computadores.

Setembro de 2021

UNIVERSIDADE D
COIMBRA

Faculdade de Ciências e Tecnologia da Universidade de Coimbra

PESQUISA DE OBRAS MUSICAIS POR CANTO

Ana Corina Martins Calhau

Dissertação no âmbito do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores, ramo de especialização em Telecomunicações, orientada pelo Professor Doutor Fernando Santos Perdigão e Dirceu Silva, apresentada à Faculdade de Ciências e Tecnologia da Universidade de Coimbra no Departamento de Engenharia Eletrotécnica e de Computadores

Setembro de 2021

Lista de Abreviaturas e Siglas

BD	Base de Dados
CRP	Cross Recurrence Plot
Csm	Cross Similarity Matrix
dB	Decibel
DFT	Discrete Fourier Transform
DTW	Dynamic Time Warping
FFT	Fast Fourier Transform
FI	Frequência Instantânea
Hz	Hertz
ID	Idenficador
IOACAS	Institute Of Acoustics, Chinese Academy Of Sciences
IOI	Inter-Onset-Interval
MIDI	Musical Instrument Digital Interface
MIR	Music Information Retrieval
MIREX	Music Information Retrieval Evaluation eXchange
MRR	Mean Reciprocal Rank
ms	Milissegundos
MTG	Music Tecnology Group
QBH	Query by Humming
SHS	Sub Harmonic Summation
STFT	Short Term Fourier Transformx

Agradecimentos

Quero deixar um sincero agradecimento a quem, de uma forma ou de outra, fez com que a realização deste trabalho se tornasse possível.

Ao Professor Doutor Fernando Manuel dos Santos Perdigão pela disponibilidade, apoio e sobretudo pela confiança depositada em mim e ao Dirceu Silva por ser incansável e estar sempre pronto para me ajudar e esclarecer as minhas dúvidas. Ainda um agradecimento ao IT-Instituto de Telecomunicações por me ter acolhido e propocionado a realização deste projeto.

À minha Mãe e ao meu Pai pelo apoio, paciência, constante preocupação e valores transmitidos não só ao longo da realização deste projeto mas também durante toda a minha vida.

Ao meu irmão David, pela ajuda na escolha do curso, e pela confiança que sempre me transmitiu de que era possível.

Ao meu namorado, César, por todo o amor, paciência, ajuda e incentivo que sempre me deu desde o início.

A todos os meus amigos pelo companheirismo e por todos os momentos partilhados, principalmente à Ana Margarida e ao Rui que foram um apoio fundamental nestes últimos anos.

Por fim, aos meus avós Almerinda e António e ao meu tio Zé, obrigada por todo o amor e boas memórias partilhadas.

Resumo

A identificação de músicas por canto, conhecida em inglês por *Query by Humming*, consiste em extrair a melodia a partir do sinal de canto ("humming") e comparar a melodia extraída comum à base de dados de melodias conhecidas. É já há muitos anos um desafio na área de investigação em música, sendo os maiores problemas a comparação de um som monofónico com sons de uma base de dados polifónica, o que torna a extração de melodia pouco eficaz com a presença dos vários instrumentos musicais e a extração da base de dados (BD).

Esta dissertação apresenta um algoritmo de procura por canto que tem por base uma BD construída e testada pela universidade espanhola Pompeu Fabra, denominada MTG, criada pelo *Music Tecnology Group*.

A base de dados MTG, disponibilizada para uso em trabalhos de investigação, foi gravada por 17 pessoas, homens e mulheres, com diferentes níveis de experiência musical, podendo cantar com ou sem letra um excerto de músicas que conhecessem de entre algumas que que lhes foram apresentadas numa lista, sem limite de tempo nem restrições de que parte da música iriam cantar. A BD de referência foi criada por nós, usando as informações apresentadas pela MTG, onde sabíamos quais eram músicas de referência mas não tínhamos acesso aos seus ficheiros aúdio devido a direitos autorais.

A seguinte dissertação faz então um resumo das abordagens a ser consideradas nomeadamente a implementação de DTW - *Dynamic Time Warping*, muito utilizada nesta área, através do algoritmo Qmax que também é usado neste projeto.

Palavras-chave: Procura por canto, Base de Dados MTG, *Dynamic Time Warping*, Qmax.

Abstract

The Query by Humming, consists in extracting the melody from the singing signal ("humming") and comparing the common extracted melody to the database of known melodies. It has been a challenge in the area of music research for many years, the biggest problems being the comparison of a monophonic music with polyphonic music database, which makes melody extraction ineffective with the presence of various musical instruments and the extraction of the database (DB).

This dissertation presents a singing search algorithm based on a DB built and tested by the Spanish University Pompeu Fabra, called MTG as it was created by the *Music Technology Group*.

The MTG database, available for use in research work, was recorded by 17 people, men and women, with different levels of musical experience, who could sing with or without lyrics an excerpt of songs they knew from among a few that were selected from a list, with no time limit or restrictions on what part of the song they would sing. The reference database was created by us, using the information presented by MTG, where we knew which were reference songs but we didn't have access to their audio files due to copyright.

The following dissertation then summarizes the approaches to be considered, namely the implementation of DTW - Dynamic Time Warping, widely used in this area, through the Qmax algoritthm that is also used in this project.

Keywords: Query by humming, Database MTG, Dynamic Time Warping, Qmax.

Conteúdo

Capítulo 1

Introdução

1.1 Pesquisa de obras musicais por canto - "Query by humming"

A pesquisa de músicas por canto ("Query by humming"ou QBH) é um método que tem como objetivo a identificação de uma música a partir de uma entrada cantada, assobiada ou trauteada, usando uma grande base de dados de músicas conhecidas para a sua identificação. Por exemplo, a melodia "Parabéns a Você"tem uma sequência de notas e respetivas durações que é bem conhecida por qualquer pessoa, que a consegue reproduzir cantando-a ou trauteando-a. Um sistema de QBH deve ser capaz de tomar o áudio desta realização da melodia, convertê-la para um formato comum a todas as melodias presentes na base de dados, de forma a fazer a comparação com cada música da base de dados e assim identificar qual foi a música que essa pessoa cantou.

A ideia subjacente a um sistema QBH é que as pessoas conseguem apreender a linha melódica de uma música que ouvem e conseguem recordá-la mais facilmente que o nome da música ou de quem a realiza. Assim, pretende-se que o sistema QBH identifique a música a partir do registo da pessoa a trautear ou a cantar essa música.

Basicamente, o método QBH deteta os tons da melodia cantada e converte-os numa sequência de frequências fundamentais que se podem associar a notas musicais. Esta sequência de notas é codificada com atributos de uma melodia, essencialmente notas musicais. A base de dados de músicas tem a mesma codificação. O passo seguinte consiste em comparar essa melodia com as existentes na base de dados de músicas conhecidas. Assim que há uma correspondência esta música é identificada.

No diagrama da figura 1.1 podemos ver um esquema generalista do processo de QBH:

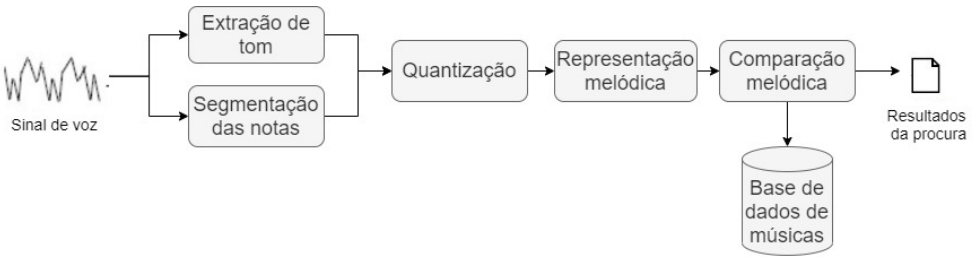


Figura 1.1: Diagrama de blocos construído com base nos artigos [1] e [2]

1.2 Dificuldades do QBH

A partir da descrição anterior pode parecer que o problema a resolver com o QBH é fácil de atingir. Contudo, este problema mostra-se muito complicado e de difícil generalização. As dificuldades podem definir-se em três níveis:

1. Imprecisão na realização da melodia (query);
2. Dificuldade na extração do tom da voz humana com precisão para associação às notas da melodia;
3. Dificuldade na comparação com uma grande base de dados (pesquisa rápida).

Estas dificuldades são pormenorizadas de seguida.

1.2.1 Realização da melodia ("query")

A reprodução humana de melodias é, quase sempre, imperfeita. Uma pessoa a cantar pode estar no tom errado, cantar mais lento ou mais rápido do que na música original ou até cantar demasiadas notas ou notas a menos, entre outros problemas.

De seguida podemos ver alguns dos aspectos que foram identificados pelo sistema de software CubyHum [1].

- **A pessoa canta qualquer parte da canção:** uma melodia repetitiva pode fazer com que a música fique "na cabeça"das pessoas, por vezes apenas uma parte;
- **A pessoa canta no tom errado:** as pessoas escolhem um tom aleatório para começar a cantar, acreditando-se que estas só acertaram no tom das suas músicas preferidas;
- **A pessoa canta num ritmo global razoavelmente parecido:** neste estudo, as pessoas sabiam ou sentiram, por já terem ouvido as músicas anteriormente, qual o andamento correto e foram capazes de se aproximar deste andamento com razoável precisão;
- **A pessoa canta poucas notas ou notas a mais:** neste estudo, as pessoas cantavam apenas as notas que se lembravam, adicionando ou "comendo"algumas, por vezes;
- **A pessoa canta intervalos errados ou confunde uns com outros:** neste estudo, as pessoas cantaram cerca de 59% dos intervalos corretamente, embora houvesse diferenças devido à experiência de canto, familiaridade musical e exposição recente da música;
- **A pessoa canta o contorno com uma precisão razoável:** neste estudo, a maioria das pessoas sabia quando subir e quando descer de tom ao cantar, sendo a taxa de sucesso de 80%;
- **Uma pessoa com experiência em canto, canta melhor em alguns aspectos do que as pessoas sem experiência:** neste estudo, a experiência ou ausência dela no mundo musical não ajuda a que se tenha mais sucesso na deteção, no entanto, cantores experientes reproduziram intervalos mais corretos e cantaram num melhor "timing";

- **A pessoa canta melodias conhecidas melhor do que outras menos conhecidas:** neste estudo, melodias menos conhecidas foram reproduzidas com menos notas e mais intervalos errados do que melodias conhecidas. Além disso, todos os participantes melhoraram o canto dos intervalos quando ouviram recentemente a melodia.

Assim, o problema pode residir mesmo antes da deteção de tom na realização da melodia, se a sequência de notas a comparar com as da base de dados é muito diferente da sequência exata da música pretendida.

1.2.2 Deteção de tom da voz humana

O conceito de tom ("pitch") está associado à altura de um tom musical, classificando-o em relação a outro como mais alto ou mais baixo. Em termos físicos, o tom da voz humana vai corresponder à frequência fundamental do sinal da voz cantada ou da fala, embora não exista uma correlação perfeita entre o tom e a frequência do som, pois o primeiro é influenciado por vários fatores, como por exemplo a intensidade. No contexto do processamento do sinal de voz o conceito de tom está associado à medida da frequência fundamental do sinal que se assume periódico, mas embebido em ruído. Assim sendo, são precisos algoritmos de determinação de tom para a voz cantada, essencialmente os mesmos métodos usados para a determinação do tom da fala.

Normalmente, uma voz masculina apresenta um tom perto dos 110 Hz, uma voz feminina perto dos 220 Hz e a de uma criança perto dos 300 Hz podendo variar cerca de uma oitava. Considerando a extensão vocal, o canto humano está entre os 80 Hz (baixo, voz mais grave) e os 1400 Hz (soprano, voz mais aguda). Se formos analisar o assobio, este está entre 700 Hz e 2800 Hz, sendo que o assobio mais grave, normalmente está perto (em frequência) da nota mais aguda que conseguimos cantar.

Um dos primeiros problemas a resolver num sistema QBH é a deteção do tom da voz cantada no "query". Trata-se de um problema genérico de deteção de tom na fala, embora na voz cantada o ritmo de variação do tom é um pouco menor que no caso da fala e o problema seja o da identificação da nota musical ou das suas variações dinâmicas ao longo do tempo. De facto, quase nunca se reproduz uma melodia usando as notas de referência. Contudo, o que é importante é a sequência relativa das notas e não o seu valor absoluto.

A quantização de notas está relacionada com a afinação na escala musical ocidental. Esta quantização transforma a entrada cantada em notas de uma partitura, e a partir daí a melodia pode ser transformada, por exemplo, para formato MIDI permitindo uma melhor avaliação auditiva.

Diversas técnicas de extração de tom irão ser discutidas no estado-da-arte deste problema.

1.2.3 Pesquisa rápida de melodia

Este problema refere-se à comparação aproximada de uma melodia que foi cantada ("query") com todas as melodias contidas na base de dados de melodias. O tempo de pesquisa irá ser proporcional ao número de melodias existentes na base de dados, e se esta for muito grande, esta pesquisa pode ser demasiado demorada, afetando o comportamento em tempo real que se pretende.

A comparação tem de ser aproximada, dada a imprecisão do "query". A técnica usada na grande maioria dos casos é a do DTW – "Dynamic Time Warping", ou alinhamento temporal dinâmico, onde o tempo pode ser encolhido ou esticado de forma a que a correspondência entre a sequência de teste ("query") e de referência (da base de dados) seja a melhor possível. Trata-se de

uma técnica de programação dinâmica onde se pretende obter um caminho ótimo entre o par de padrões de pesquisa.

Nos gráficos da figura 1.2 podemos ver um exemplo da comparação entre as notas do ficheiro cantado e o as notas da música que se encontra na base de dados [3] [4].

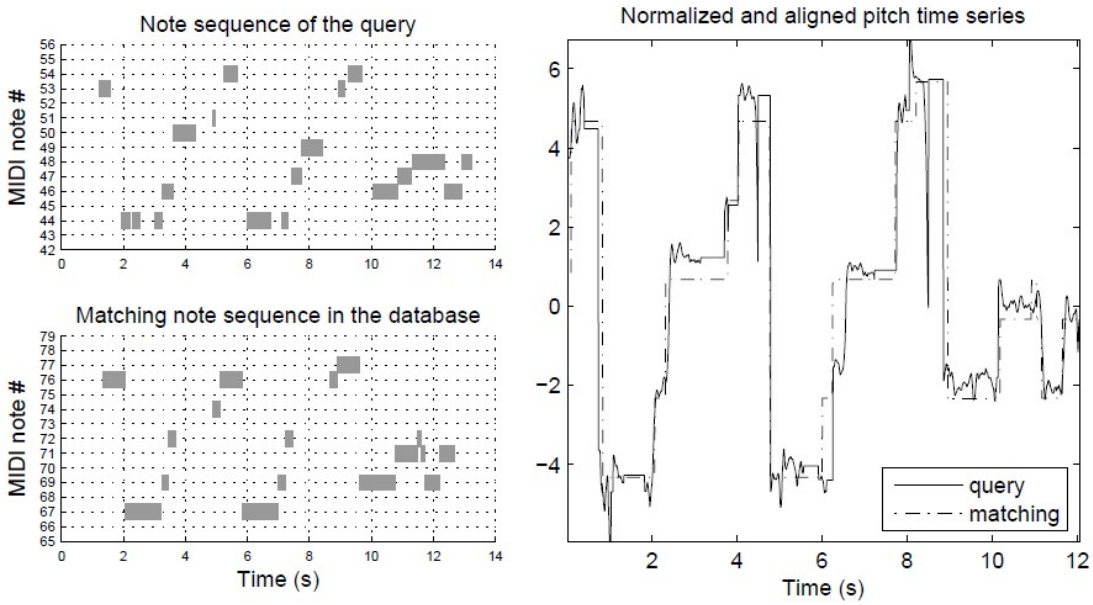


Figura 1.2: Exemplo retirado do artigo [2]

Na figura 1.2 podemos ver um exemplo do que é feito com o método de extração de melodia, mais à frente iremos abordar o Qmax, que é o método utilizado nesta dissertação para comparação dos queries com as músicas de referência.

Existe ainda uma outra dificuldade, que se prende com a construção da base de dados de referência, e que tem a ver com o facto de existirem muitos ficheiros MIDI de músicas, mas que são polifónicas, i.e., com vários instrumentos a tocar ao mesmo tempo, por vezes sem voz (caso de músicas instrumentais), mais à frente iremos abordar novamente este assunto.

Nos capítulos seguintes vamos poder ver quais são os estudos que já foram realizados na área de procura de músicas, como foi adquirida a base de dados utilizada nesta dissertação, qual o algoritmo de QBH que foi utilizado e como funciona e será ainda descrito todo o trabalho efetuado.

Capítulo 2

Estado da Arte

2.1 Extração de tom

Para estimar o tom da fala, que se aplica também ao canto, existe uma grande variedade de algoritmos propostos. Um destes métodos é o SHS ("sub-harmonic summation") [1].

Podemos realçar dois tipos de métodos: métodos temporais e métodos baseados na frequência.

Os métodos temporais extraem o tom a partir da autocorrelação de termo curto dos sinais quase periódicos. A sequência de autocorrelação tende a ser também periódica com o mesmo período, pelo que o tom fundamental corresponde ao 2^{o} pico da sequência de autocorrelação.

Nos métodos baseados na frequência, usa-se o espetro de termo curto do sinal. O tom extrai-se usando vários picos espetrais que devem acontecer em múltiplos da frequência fundamental. O SHS é um desses métodos que soma os picos espetrais em oitavas.

2.2 Segmentação das notas

A segmentação das notas é baseada em algoritmos usuais de processamento de sinal padrão [1]. De maneira a tentar obter uma melhor desempenho do sistema QBH, os utilizadores devem cantar monossílabos (por exemplo, lá/lá/lá), no entanto, ao pedir aos utilizadores para o fazerem estamos a limitá-los na forma como se expressam.

Assume-se que os inícios das notas são caraterizados por um aumento abrupto de energia ao longo de um alcance amplo de frequência. A nota sustentada tem uma forma espectral relativamente estável que representa os formantes da vogal usada. Embora os desvios de nota possam ser identificados até certo ponto pela queda de energia especialmente nas frequências mais altas, estes não são definidos com clareza.

O autor em [1] associa várias técnicas de processamento de sinal padrão à deteção da ascensão e queda de energia em diferentes frequências para segmentar o sinal em tempos de início e deslocamento de nota. Primeiro, o método de energia de curto prazo é usado para detectar partes silenciosas no sinal de canto. Posteriormente, pelo que é proposto em [1], cada parte não silenciosa é fornecida ao outros três métodos em sucessão. O autor associa um sinal digital *x(n)*, para $n = 0, ..., N$ que é pré-enfatizado pelo filtro $1 - 0,95z^{-1}$ para produzir um aumento de 32 dB na magnitude espectral, segmentado em tramas. Esta técnica não é genérica, e neste caso não foi utilizado por nós.

Na figura 2.1 podemos verificar a diferença entre estes métodos, falando um pouco melhor de cada um de seguida.

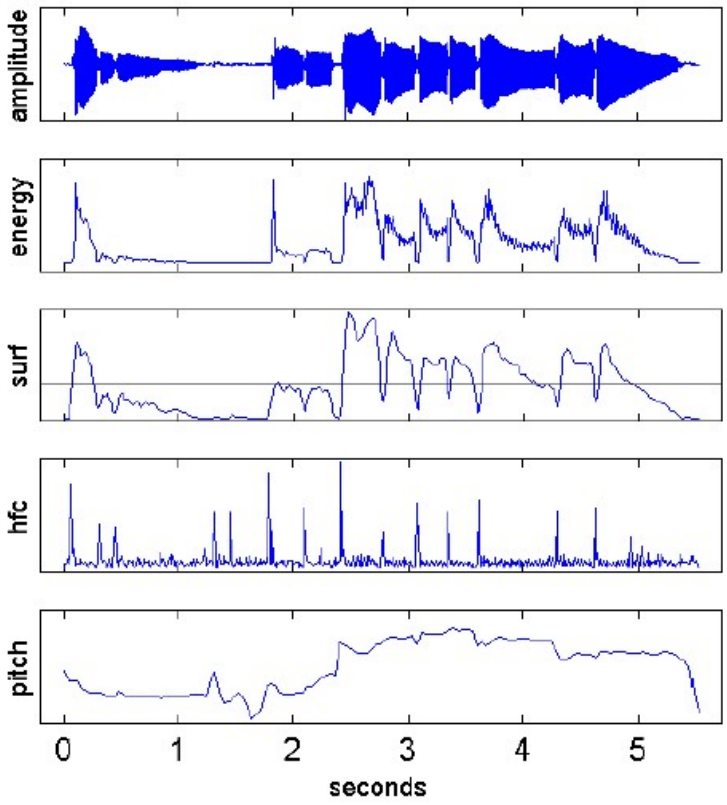


Figura 2.1: Exemplo retirado do artigo [1]

2.2.1 Método de energia de curto prazo

O método de energia de curto prazo é um método direto para detectar inícios e deslocamentos de notas e distinguir o canto do silêncio (pausas). Para isso, em [1] o sinal é segmentado em tramas de 10 ms.

Em primeiro lugar, o procedimento procura que o início da primeira nota seja detetado. Se a energia de curto prazo exceder o limite de início, é detetado o início de uma nota. Se, após o início da nota ter sido detetado, a energia de curto prazo cai abaixo do limite de deslocamento, é detetado o deslocamento da nota.

2.2.2 Método de surf

O algoritmo de início de detecção do Surf foi adotado a partir das técnicas de Schloss [5]. O sinal é passado por um filtro passa-alto de primeira ordem e segmentado em tramas de 20 ms, com

deslocamento de tramas de 10 ms. As tramas são usadas para calcular uma suavização envolvente da amplitude do sinal que representa as maiores frequências.

A inclinação da envolvente é encontrada por um procedimento de ajuste polinomial [6]. Para isso, a sequência da envolvente é equipada com uma segunda ordem polinomial sobre um segmento de M pontos da sequência. Denotamos uma pequena sequência da envolvente finita em torno da amostra τ por $\{y(\tau + t)\}_{t=-M}^{M}$ onde M especifica a largura do polinómio de interpolação. Este pequeno segmento é ajustado por uma segunda ordem polinomial $a + bt + ct^2$ minimizando alguns erros de ajuste.

Uma aproximação polinomial da inclinação em t é então dada por:

$$\frac{\partial y(\tau + t)}{\partial t} \cong b = \sum_{t=-M}^{M} \frac{t.y(\tau + t)}{\sum_{t=-M}^{M} t^2} \tag{2.1}$$

Já que o início de uma nota é caracterizado por um aumento abrupto de frequências do sinal, olhamos para cruzamentos de zeros positivos do contorno para encontrar esses tempos de início.

2.2.3 Método de conteúdo de alta frequência

O método de conteúdo de alta frequência foi usado em [7]. Este tem como objetivo revelar ambas as mudanças em energia geral e em energia concentrada em frequências mais altas.

Uma DFT de M pontos é usada para produzir $X_k(m)$. A energia de curto prazo E_k na trama k é calculada como a soma da magnitude quadrada de cada segmento FFT,

$$E_k = \sum_{m=m_0}^{M/2+1} |X_k(m)|^2 \tag{2.2}$$

onde m_0 denota o índice da DFT mais baixo que é tomado em consideração. Consideram-se apenas índices da DFT que caem dentro de uma banda de frequência de 400 Hz ou superior. O conteúdo de frequência mais alta de curto prazo H_k na trama k é uma versão ponderada de E_k, linearmente polarizado para as frequências mais altas,

$$H_k = \sum_{m=m_0}^{M/2+1} m.|X_k(m)|^2 \tag{2.3}$$

Uma função de detecção é calculada de maneira a que combine cada par de tramas consecutivas; é o produto do aumento da energia de alta frequência entre duas tramas e a corrente mais alta normalizada do conteúdo de frequência,

$$D_k = \frac{H_k}{H_{k-1}}.\frac{H_k}{E_k} \tag{2.4}$$

A primeira proporção representa o aumento no conteúdo de alta frequência. A segunda proporção representa o conteúdo de alta frequência normalizado para o quadro atual k. O seu produto tem um pico proeminente no aumento abrupto do conteúdo de energia de alta frequência. Se ultrapassar um dado limite, dizemos que a função de detecção encontrou um início.

2.3 Quantização

A quantização é a divisão do tom e do tempo em etapas, sendo que estas etapas servem para verificar onde houve alteração de tom e quanto tempo passou. A quantização de tempo não é utilizada em [1] pois esta não é utilizada na comparação melódica, sendo substituída pelo intervalo entre inícios (IOI, inter-onset-interval) que é a diferença de tempo entre dois inícios de nota adjacentes.

Para chegar a um tom musical discreto, ou seja um tom central, é necessário um contorno de inclinação dentro de um determinado intervalo de tempo. Este tom central deve ser o igual ao tom que o utilizador pretendia produzir. Embora se sabia que o tom discreto detetado em torno do tom é o da média, este não deve ser utilizado porque é sensível a erros de oitava ou estimativas de desvio de tom. Assim sendo utiliza-se o tom central.

O tom médio entre o início e o deslocamento de uma nota é utilizado para quantizar o valor do tom musical para cada nota utilizando uma escala musical afinada a 440Hz. O tom musical é representado em categorias ao longo das escalas. Essas categorias são medidas relativamente à frequência. Para calcular o tom musical discreto, p, de um tom médio, f, usamos a razão de frequência $R = f/8.176$ e

$$p = [12 \log_2 R + 0.5] \tag{2.5}$$

Assim sendo, o tom musical é representado por um valor inteiro. Então esse valor é adicionado a uma matriz, sendo que o seu valor terá que estar entre 0 e 127 no caso do MIDI para codificar o tom musical.

O utilizador ao cantar, obviamente vai ter alguns desvios da afinação universal (440Hz). Estes tendem a diminuir ou aumentar o tom devido a grandes intervalos, fadiga ou até imprecisões no controlo muscular. Para compensar isso o padrão de afinação é adaptado para o canto em curso. Assim sendo, para a primeira nota cantada assume-se que o utilizador começa na afinação correta, para as notas seguintes o tom musical é calculado e a diferença entre o tom cantado e o tom musical original é calculada.

2.4 Representação melódica

Uma representação melódica em [1] é a sequência de distâncias entre duas notas subsequentes expressas em semitons.

Em particular, numa sequência melódica $S = s_1 s_2 ... s_N$ compreendendo tons absolutos esta é transformado numa sequência $S' = .(s_2 - s_1)(s_3 - s_2)...(s_N - s_{N-1}$ onde o ponto '.' representa um elemento inicial especial, pois o intervalo não está associado à primeira nota de uma melodia.

Com a finalidade de usar uma escala diatónica estruturada, os intervalos são agrupados. Isto tem que ser elaborado sem conhecimento sobre a tónica, porque não se sabe qual é a chave da melodia, excepto para o uníssono, onde todos os intervalos são categorizados em grupos de dois intervalos (1 a 2 semitons). Todos os intervalos maiores que 6 semitons são agrupados numa categoria para intervalos ascendentes e descendentes.

Conforme mostrado na tabela seguinte, as categorias de intervalo são representadas pelo inteiros -4,-3,...,3,4. O elemento inicial especial '.' é mantido. A representação da melodia resultante é um contorno de 9 etapas e assemelha-se ao modelo *Dowling* de como as melodias são consideradas

armazenado na memória humana [8]. As informações temporais são mantidas por armazenar o valor real do intervalo IOI de cada nota a cada elemento de contorno de 9 etapas correspondentes.

nome do intervalo	**tamanho do intervalo**	**código inteiro**
desc. quinta maior e diminuta	<-6 st	-4
desc. perfeita/quarta aumentada	-5 ou -6 st	-3
desc. menor/terceira maior	-3 ou -4 st	-2
desc. menor/segunda maior	-1 ou -2 st	-1
uníssoma	0 st	0
asc. menor/segunda maior	1 ou 2 st	1
asc. menor/terceira maior	3 ou 4 st	2
asc. perfeita/quarta aumentada	5 ou 6 st	3
asc. quinta maior e diminuta	>6 st	4

Tabela 2.1: Representação melódica em 9 passos (st = semitom) [1]

A invariância para o tempo global é estabelecida pelo cálculo das razões de dois IOI, mas isso é feito na comparação melódica.

2.5 Comparação melódica

A comparação de uma melodia com as melodias da base de dados depende da forma como foi quantizada a melodia.

No primeiro caso, quando utilizamos a melodia do sinal sonoro de entrada, a melodia é definida como uma sequência de notas musicais. Pode considerar-se, então, que uma melodia é uma sequência de frequências, neste caso "chromas"[1].

O termo "chroma"ou "chromagrama"relaciona-se com as doze classes de notas diferentes. Os "perfis de classe de pitch", são uma ferramenta poderosa para analisar música cujos tons podem ser categorizado, geralmente em doze categorias, e cuja afinação se aproxima da escala temperada.

A pesquisa na base de dados terá de ser aproximada uma vez que o "query"é impreciso na sequência de símbolos. O algoritmo a usar é idêntico ao algoritmo de edição de "strings", onde se pretende obter uma distância[2] para medir a diferença entre as duas sequências (distância de Levenshtein). A melodia da base de dados com menor distância será a escolhida, no caso de se pretender indicar apenas uma solução. A alternativa é a de indicar as N melodias com menor distância (hipóteses N-best).

A 2ª alternativa consiste em definir uma melodia como uma sequência de tons a um dado ritmo fixo [9]. Neste caso a duração das notas fica implícita na sequência fornecida e corresponde ao número de vezes que uma dada frequência varia menos que 1/2 tom musical. O algoritmo a usar neste caso é o usual DTW.

A 3ª alternativa consiste em modelar estocasticamente a melodia com um processo de Markov não observável [10]. Cada estado do modelo corresponde a uma nota e a transição de estados à

[1]mais informações em `https://en.wikipedia.org/wiki/Chroma_feature`
[2]mais informações em `https://en.wikipedia.org/wiki/Edit_distance`

mudança para uma outra nota musical. O algoritmo a usar neste caso é o algoritmo de Viterbi, que define um caminho ótimo entre a sequência de "query"e os estados do modelo.

2.5.1 Resultados do MIREX 2020 para QBH

Os resultados do MIREX 2020 para extração de melodia de áudio, são aqui apresentados para servirem de referência de maneira a perceber-se que valores tem sido obtidos em outros trabalhos da área. Estes resultados pertencem às seguintes base de dados[3]:

- ADC04 Dataset
- MIREX05 Dataset
- INDIAN08 Dataset
- MIREX09 0dB Dataset
- MIREX09 -5dB Dataset
- MIREX09 +5dB Dataset
- ORCHSET15 Dataset

Algorithm	**Overall Accuracy**	**Raw Pitch Accuracy**	**Raw Chroma Accuracy**	**Voicing Recall Rate**	**Voicing False-Alarm Rate**
WJH2	0.3296	0.3848	0.4284	0.9346	0.8119
AH1	0.7501	0.7890	0.8025	0.8191	0.1832
HZ4	0.8240	0.7966	0.8418	0.9560	0.2348
KD1	0.8153	0.8322	0.8373	0.8686	0.1690
HL1	0.5952	0.6401	0.7730	0.7210	0.1589

Tabela 2.2: Resultados MIREX 2020: Audio Melody Extraction - ADC04 Dataset

[3]mais informações em `https://www.music-ir.org/mirex/wiki/2020:MIREX2020_Results`

Algorithm	Overall Accuracy	Raw Pitch Accuracy	Raw Chroma Accuracy	Voicing Recall Rate	Voicing False-Alarm Rate
WJH2	0.3274	0.4484	0.4975	0.9416	0.9379
AH1	0.6513	0.7156	0.7411	0.7894	0.2979
HZ4	0.6091	0.5978	0.6606	0.8241	0.3088
KD1	0.6764	0.7318	0.7631	0.7899	0.2985
HL1	0.5816	0.5987	0.7062	0.7683	0.2691

Tabela 2.3: Resultados MIREX 2020: Audio Melody Extraction - MIREX05 Dataset

Algorithm	Overall Accuracy	Raw Pitch Accuracy	Raw Chroma Accuracy	Voicing Recall Rate	Voicing False-Alarm Rate
WJH2	0.4853	0.5601	0.5683	0.8845	0.8634
AH1	0.6613	0.7322	0.7368	0.7125	0.1579
HZ4	0.6736	0.6735	0.6878	0.7429	0.2072
KD1	0.7140	0.7838	0.7876	0.8491	0.5008
HL1	0.6599	0.7083	0.7244	0.7646	0.1842

Tabela 2.4: Resultados MIREX 2020: Audio Melody Extraction - INDIAN08 Dataset

Algorithm	Overall Accuracy	Raw Pitch Accuracy	Raw Chroma Accuracy	Voicing Recall Rate	Voicing False-Alarm Rate
WJH2	0.5000	0.7538	0.7628	0.9903	0.9580
AH1	0.8093	0.8268	0.8280	0.9037	0.1583
HZ4	0.7216	0.7832	0.8010	0.9204	0.3839
KD1	0.6401	0.7387	0.7596	0.8334	0.4254
HL1	0.8366	0.8677	0.8700	0.9165	0.1517

Tabela 2.5: Resultados MIREX 2020: Audio Melody Extraction - MIREX09 0dB Dataset

Algorithm	Overall Accuracy	Raw Pitch Accuracy	Raw Chroma Accuracy	Voicing Recall Rate	Voicing False-Alarm Rate
WJH2	0.4388	0.6624	0.6792	0.9903	0.9580
AH1	0.6933	0.7342	0.7378	0.8572	0.2624
HZ4	0.5574	0.6311	0.6666	0.9087	0.5699
KD1	0.4853	0.5515	0.5942	0.7599	0.5091
HL1	0.7278	0.7736	0.7829	0.8889	0.2639

Tabela 2.6: Resultados MIREX 2020: Audio Melody Extraction - MIREX09 -5dB Dataset

Algorithm	Overall Accuracy	Raw Pitch Accuracy	Raw Chroma Accuracy	Voicing Recall Rate	Voicing False-Alarm Rate
WJH2	0.5190	0.7824	0.7889	0.9903	0.9580
AH1	0.8683	0.8776	0.8785	0.9107	0.0844
HZ4	0.8156	0.8392	0.8467	0.9211	0.2169
KD1	0.7568	0.8539	0.8626	0.8892	0.3350
HL1	0.8792	0.8997	0.9012	0.9186	0.0847

Tabela 2.7: Resultados MIREX 2020: Audio Melody Extraction - MIREX09 +5dB Dataset

Algorithm	Overall Accuracy	Raw Pitch Accuracy	Raw Chroma Accuracy	Voicing Recall Rate	Voicing False-Alarm Rate
WJH2	0.0531	0.0548	0.1572	0.9437	0.9625
AH1	0.4083	0.5025	0.7291	0.7638	0.6011
HZ4	0.1981	0.1779	0.4214	0.7565	0.5750
KD1	0.4685	0.5178	0.6652	0.7850	0.6337
HL1	0.1414	0.1397	0.5070	0.5579	0.3585

Tabela 2.8: Resultados MIREX 2020: Audio Melody Extraction - ORCHSET Dataset

As tabelas apresentadas representam os resultados mais recentes nesta área de investigação, podendo assim, mais à frente, compará-los com os que foram obtidos por nós neste trabalho.

Capítulo 3

Base de dados

3.1 Escolha da Base de Dados

Existem múltiplos trabalhos na área da pesquisa em base de dados de músicas, no entanto existe alguma dificuldade em aceder a essas base de dados livremente, pois normalmente não são disponibilizadas. Dada a complexidade da preparação de uma base de dados musical de raiz, optamos então por utilizar algo já existente. Assim sendo, depois de alguma procura encontrámos apenas disponíveis de forma gratuita três base de dados,cujos nomes foram atribuídos com o mesma nomenclatura de quem as produziu sendo que estas pertencem aos seguintes grupos e/ou institutos e são assim denominadas como:

- *Institute Of Acoustics, Chinese Academy Of Sciences* (IOACAS)
- *Music Information Retrieval* (MIR)
- *Music Tecnology Group* (MTG)

sendo que as duas primeiras foram encontradas através da página do MIREX 2021 -*Music Information Retrieval Evaluation eXchange*. O MIREX foi um concurso criado em 2005, durante a ISMIR - *International Conference on Music Information Retrieval*, tendo como objetivo a comparação de algoritmos e sistemas de recuperação de informações musicais do ano presente.

A base de dados IOACAS é composta por 759 queries cantadas e 298 arquivos MIDI monofónicos (apenas com 0 e 1) de referência. Não é garantido que no inicio das queries haja música. A base de dados MIR de *Roger Jang* tem na sua composição 4431 queries cantadas e 48 arquivos MIDI. Todas as comparações são feitas desde o ínicio das músicas de referência. A base de dados MTG foi disponibilizada online pela Universidade Pompeu Fabra em Barcelona.

3.1.1 MTG

Esta base de dados foi a escolhida para ser utilizada no nosso trabalho porque tínhamos acesso a 118 queries e aos metadados de 81 obras de referência. Não são disponibilizados os ficheiros de áudio das obras de referência por questões de direitos autorais mas temos ainda acesso aos resultados obtidos em várias experiências pela Universidade Pompeu Fabra no artigo [11].

Os documentos foram os seguintes:

- **Collection_Full.csv**

  Este ficheiro contém metadados acerca da coleção completa de músicas de referência que foram utilizadas para a avaliação feita nas experiências descritas no artigo mencionado [12]. O ficheiro tem os metadados acerca de 2125 músicas e para cada uma tem as seguintes informações:

  - **Song ID:** este identificador é arbitrário e foi atribuído a cada uma das gravações, o ID é único, pelo que diferente gravações da mesma música tem IDs diferentes.
  - **Title:** contém o título da música, sendo que gravações da mesma música podem ter títulos diferentes caso sejam cantadas em idiomas diferentes.
  - **Artist:** contém o artista que fez a performance da música.
  - **Original Artist:** contém o compositor da música.
  - **Canonical:** este campo é preenchido com um 'SIM' caso está seja a versão canonica da música, caso contrário o campo fica vazio.
  - **Class label:** esta etiqueta é única para todas as gravações da mesma música, sendo que é idêntica para todas as gravações que são consideradas versões da mesma música.

- **Collection_Canonicals.csv**

  Este ficheiro contém metadados acerca da coleção completa de músicas canónicas que foram utilizadas para a avaliação feita nas experiências descritas no artigo acima mencionado. Ao contrário do ficheiro anterior, este contém apenas uma versão para cada música, ou seja este ficheiro é uma versão reduzida do anterior. Neste documento estão descritas 481 músicas, sendo que as informações incluídas no mesmo são as mesmas do ficheiro anterior. Os valores do ID da música neste ficheiro são iguais aos valores do ID da música do ficheiro anterior.

- **Queries.csv**

  Este ficheiro contém metadados acerca da coleção completa das queries que foram gravadas por várias pessoas para a avaliação feita nas experiências descritas no artigo acima mencionado. O ficheiro tem os metadados acerca de 118 queries e para cada uma tem as seguintes informações:

  - **Filename:** contém o nome do ficheiro de áudio.
  - **Query ID:** contém o identificador (ID) da query.
  - **Song ID:** contém o ID da música de referência correspondente a cada query.
  - **Original Artist:** contém o compositor da música de referência correspodente a cada query.
  - **Class label:** esta etiqueta é única para todas as gravações da mesma música, sendo que é idêntica para todas as gravações (seja query ou música de referência) que são consideradas versões da mesma música.

3.2 Descrição da Base de Dados MTG

Todas as músicas de referência e queries da base de dados MTG foram utilizadas no formato wav, com frequência de 44100Hz. Apenas os áudios dos queries são disponibilizados.

As gravações das queries foram feitas por 17 pessoas, sendo este grupo constituído por 9 mulheres e 8 homens, cuja experiência musical ia de nula a músicos amadores. Todos os indivíduos presentes nas gravações das queries da base de dados MTG deram autorização para que este conjunto de dados fosse tornado público. Numa primeira fase os participantes receberam uma lista de músicas onde tinham que seleccionar quais conheciam e cantar uma parte destas, sendo que podiam cantar qualquer passagem e sem restrição de duração, com ou sem letra, podendo então cantar ou trautear.

Os indivíduos gravaram sem terem escutado as músicas originais e sem qualquer tipo de acompanhamento ou tom de referência, cantando apenas de memória, de maneira a não influenciar as gravações e consequentemente os resultados. Podiam cantar num tom diferente do original sem que isso afetasse o desempenho do algoritmo. Com o objetivo de simular um cenário de QBH realista, todas as gravações foram feitas através de um microfone básico de um computador portátil sem nenhum tipo de processamento. A duração média de gravação foi de 26.8 segundos.

Para além das gravações dos queries, também estavam incluídos três arquivos de metadados, como já foi mencionado anteriormente. Embora fosse dado acesso a todos os queries, não são disponibilizadas as músicas de referência que estavam listadas nestes arquivos (apenas os nomes e autores das músicas) por estarem protegidos pela lei dos direitos de autor. Sendo assim foi necessária a obtenção desses ficheiros através da transferência do aúdio dos vídeos correspondentes no youtube. Para tal foi utilizado o programa Youtube-DL, que é um gerenciador de download de código aberto para vídeo e audio do Youtube e de mais de algumas centenas de sites de vídeo. As músicas de referência que nos foram apresentadas como utilizadas pela BD MTG eram 81, no entanto acrescentamos além dessas 81 originais cerca de dois a três "covers"por cada música, acrescentando assim 179 covers resultando num total de 260 músicas de referência na nossa BD.

Capítulo 4

Algoritmo de QBH

4.1 Algoritmo de extração de melodia

A extração de melodia foi conseguida através de quatro blocos, segundo o descrito no artigo [13]. São eles a extração sinusoidal, o cálculo da função de saliência, a criação de contornos de inclinação e a seleção de melodia.

4.1.1 Extração sinusoidal

A extração sinusoidal é um processo que foi dividido em três etapas: filtragem, transformação espectral e correção da frequência/amplitude sinusoidal.

O sinal de aúdio é inicialmente filtrado de forma a simular as curvas de igual "loudness"[1].A descrição deste filtro é indicado em [13]. Depois de efetuada a filtragem, é aplicada a Transformada de "Fourier"de curto termo (STFT - *Short Term Fourier Transform*), que é dada pela seguinte equação:

$$X_l(k) = \sum_{n=0}^{M-1} w(n).x(n + lH)e^{-j\frac{2\pi}{N}kn}, \tag{4.1}$$

$$l= 1:\text{Nf e k} =0,1,...,\text{N-1}$$

Os valores que foram usados foram os seguintes: a função da janela "Hann", w(n) com um tamanho de 46.4 ms, salto de 2.9 ms, ou seja com dados amostrados f_S = 44.1 kHz temos M = 2048, N = 8192 e H = 128. O "hop-size"de 128 corresponde a uma taxa de 344.5 tramas por segundo, significando que iremos ter cerca de 345 estimativas da frequência da melodia por segundo.

O tamanho relativo do hop size é escolhido de maneira a que fique mais fácil o rastreamento F0, assim este torna-se mais preciso durante a criação de contornos de pitch.

[1] ver mais informações em `https://en.wikipedia.org/wiki/Equal-loudness_contour`

A decisão de usar a STFT, que tem uma resolução fixa de tempo-frequência, em vez de outro tipo de transformada de multi-resolução foi tomada pelos autores de [13] depois de algumas comparações. O uso de uma transformada de multi-resolução tem como benefícios uma maior resolução de frequência em baixas frequências, onde há agrupamentos de pico muito próximos, sendo estacionários ao longo do tempo, e uma maior resolução de tempo para as altas frequências, onde podemos encontrar picos que irá ser modulados muito rapidamente ao longo do tempo (exemplo: harmónicos vibrato).

Os picos espetrais são selecionados pela localização dos máximos locais do espetro de magnitude $|X_l(k)|$, considerando apenas uma frame de $X_l(k)$.

Correção de frequência/amplitude: os picos espectrais tem a sua localização limitada pelos "bins"da DFT, sendo que isto pode resultar num erro na estimativa da frequência de pico em frequências baixas. Para proceder à correção de frequência/amplitude temos que usar o espectro de fase $\phi_l(k)$ para calcular a frequência instantânea (FI) e a amplitude de pico, que são mais fidedignas a fornecer uma estimativa da verdadeira frequência e amplitude de pico.

Para correção de frequência/amplitude os autores compararam dois métodos, o de interpolação parabólica [14] e o de frequência instantânea usando o método de fase de "vocoder"[15]. Chegou-se à conclusão que ambos os métodos fornecem uma grande melhoria na precisão da frequência quando comparados com o uso dos "bins"da DFT, no entanto o método baseado em fase tem um desempenho ligeiramente melhor e por isso foi o escolhido. Mais pormenores podem ser obtidos em [13].

4.1.2 Cálculo da função de saliência

O passo seguinte do algoritmo descrito em [13], e usado neste trabalho, é a determinação de apenas uma frequência (da melodia) dado o espectro polifónico, contendo, por exemplo, acordes com várias notas em simultâneo. O principio usado neste algoritmo é a "saliência"do tom num conjunto de harmónicos fundamentais.

Para construir uma função de saliência são usados os picos espectrais extraídos. Estes são os candidatos F0 para a melodia principal. Para calcular a saliência o sistema usado foi baseado na soma harmónica do artigo [16], sendo que a saliência de uma determinada frequência resulta da soma das energias dos harmónicos daquela frequência. No entanto, ao contrário do que é apresentado no artigo [16], os picos espectrais só são usados na soma de maneira a descartar valores espectrais não fiáveis. Pode-se aplicar a correção de frequência através dos picos, o que nos permite melhorar a precisão da frequência da função de saliência.

O cálculo da função de saliência foi feito com combinações diferentes de parâmetros e os picos de saliência que daí resultaram foram avaliados através de métricas que foram escolhidas de maneira a estimar a predominância da melodia comparando com outros elementos de pico presentes na função de saliência. Sendo assim os valores ótimos que foram determinados para os parâmetros de saliência foram os seguintes: $\alpha = 0,8, \beta = 1, \gamma = 40$ e $N_h = 20$. Sendo que no MIREX 2010 os valores utilizados foram 0.8, 2, 40 e 8 respectivamente, mas foram utilizados em experiências realizadas antes da optimização de parâmetro em [17].

4.1.3 Caracterização de Contorno de Pitch

Depois da criação dos contornos, o maior desafio é a determinação dos contornos pertencem ou não à melodia. Para que isto seja possível é definido um conjunto de características de contorno utilizadas para melhorar a seleção dos contornos da melodia. Os recursos usados como base no contorno do tom foram o comprimento e saliência. No entanto, conseguimos estender esse conjunto se evitar-mos a quantização de contornos em notas, introduzindo características extraídas da trajetória de pitch do contorno, como o desvio de tom e a presença de vibrato.

As características calculadas para cada contorno são as seguintes:

- **Pitch médio** $C_{\overline{p}}$: altura média do tom do contorno.
- **Desvio de tom** $C_{\sigma p}$: desvio padrão do tom do contorno.
- **Contorno médio saliência** $C_{\overline{s}}$: saliência média de todos os picos que constituem o contorno.
- **Saliência total do contorno** $C_{\sum s}$: soma da saliência de todos os picos que constituem o contorno.
- **Desvio de saliência do contorno** $C_{\sigma s}$: desvio padrão da saliência de todos os picos que constituem o contorno.
- **Comprimento** C_l: comprimento do contorno.
- **Presença de vibrato** C_v: prensença ou não de vibrato (verdadeiro/falso).

O vibrato é detetado pelo sistema usando um método baseado em [18], onde é aplicada a FFT à trajetória de inclinação do contorno e é verificado se existem ou não um pico que se destaque na faixa de frequência esperada (5-8 Hz para o vibrato humano).

Para uma melhor percepção, na figura 4.1 podemos ver exemplos de contornos criados excertos de diferentes géneros musicais.

A dispersão de contornos não melódicos pode se dever ao filtro de volume igual e ao filtro de pico de saliência que já foram referidos anteriormente. Pela observação da figura podemos propor algumas características de contorno de modo a diferenciar a melodia do resto dos contornos, como vibrato, maior variação de tom, contornos mais longos, uma faixa de tom de frequência média e maior saliência.

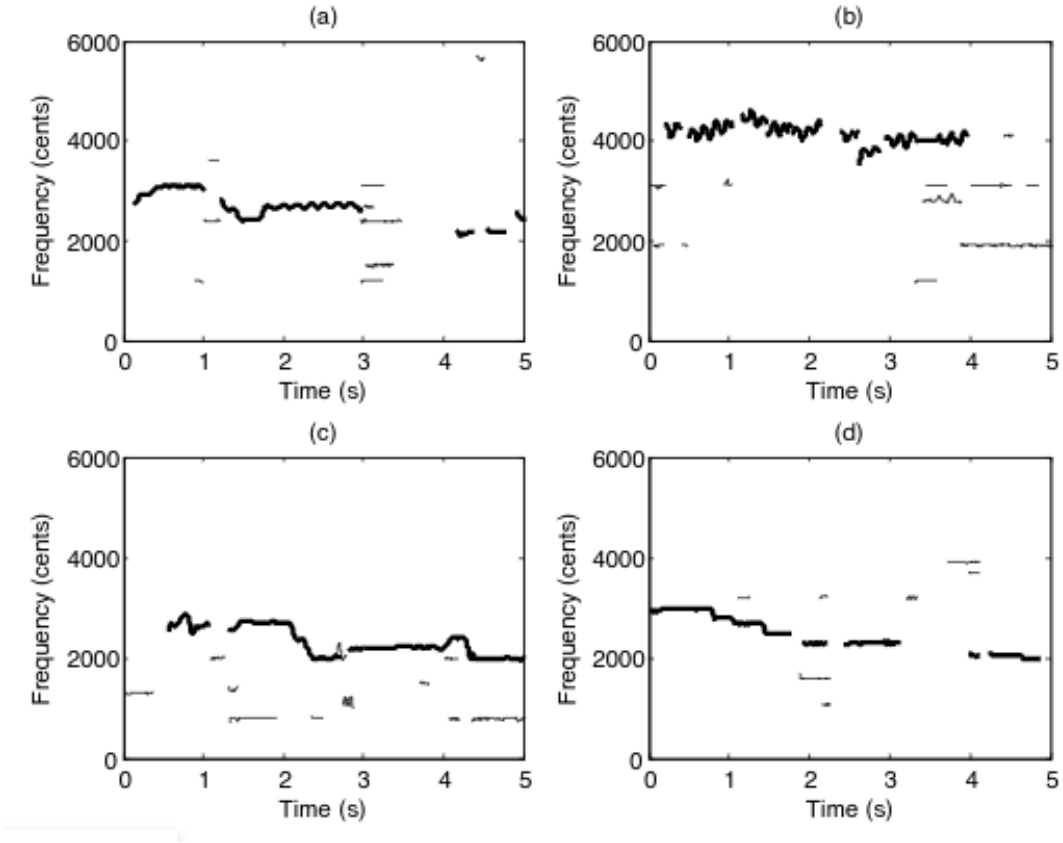


Figura 4.1: Figura retirada do artigo [13], contendo contornos de pitch gerados a partir de excertos de (a) jazz vocal, (b) ópera, (c) pop e (d) jazz instrumental. Os contornos da melodia são destacados em negrito.

De maneira a confirmar as observações foram calculadas as distribuições de recursos para contornos de melodia e não melodia. Podemos ver que em grande parte dos excertos a melodia é composta por voz humana. As distribuições são apresentadas na figura 4.2.

Nos gráficos (c), (d) e (e) da figura 4.2, os valores das características foram normalizados pelo valor médio das caraterísticas de cada excerto. Podemos verificar que 95% de todos os contornos em que o vibrato foi detetado eram contornos de melodia. Forma consideradas várias caraterísticas de contorno porque os instrumentos de acompanhamento não serão sempre selecionados como melodia mesmo que contenham uma certa caraterística melódica.

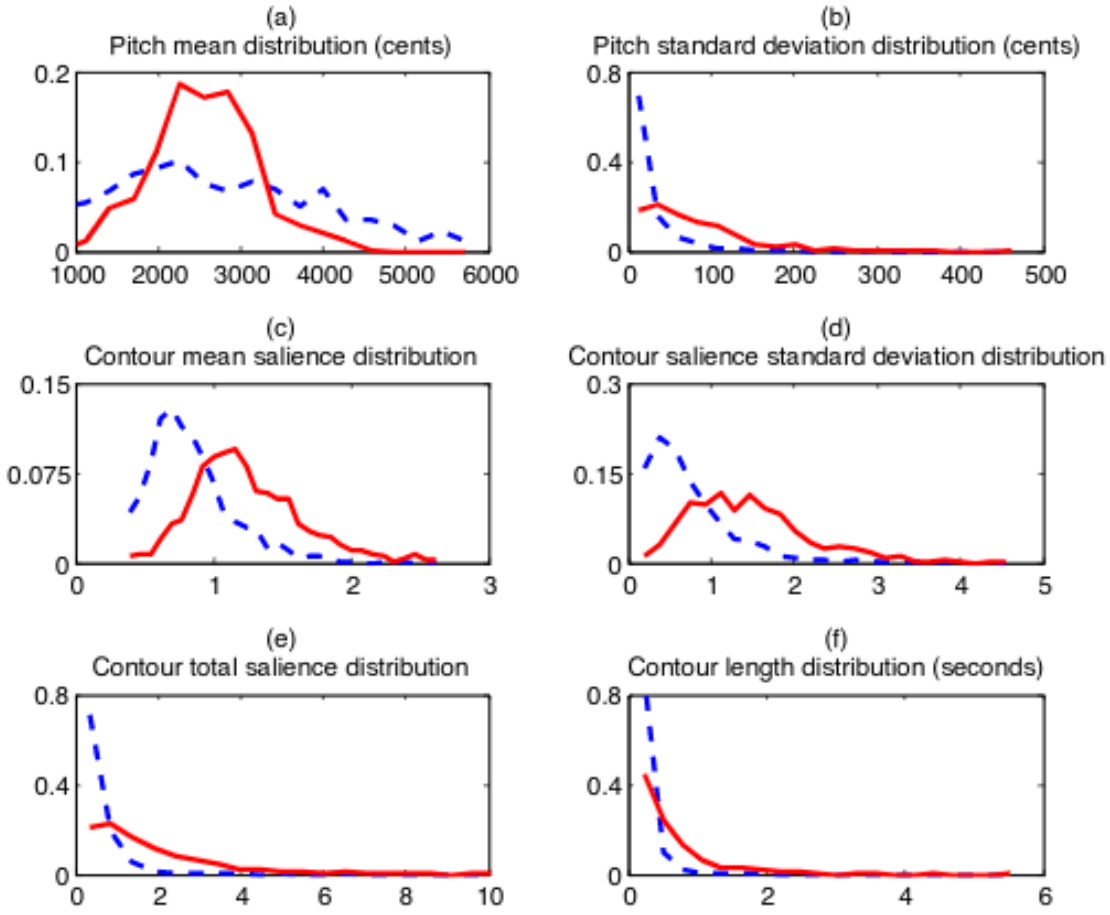


Figura 4.2: Figura retirada do artigo [13], contendo distribuições de características de contorno de inclinação: (a) Média de inclinação, (b) Padrão de desvio de inclinação, (c) Saliência média, (d) Padrão de desvio de saliência, (e) Saliência total, (f) Comprimento. A linha contínua representa a distribuição dos recursos de contorno da melodia, a linha tracejado representa a distribuição de recursos de contorno não melódicos.

4.1.4 Seleção de melodia

Após a criação dos contornos, nesta etapa em vez de selecionar os contornos de melodia colocaram esta tarefa como um problema de filtragem de contorno, sendo o objetivo filtrar todos os contornos não melódicos. Este processo é composto por três fases, sendo que nas duas primeiras fases as caraterísticas de contorno são usadas para filtrar os contornos não melódicos e, na etapa final a frequência da melodia em cada frame é seleccionada dos restantes contornos:

- **Deteção de voz:** é a fase que determina quando é que a melodia está ou não presente.

Podemos observar no gráfico (a) da figura 4.1, que a melodia está presente nos segundos 0-3 e 4-5, havendo ausência de melodia entre 3-4, ou seja onde os contornos não melódicos foram encontrados. De maneira a que esses contornos possam ser filtrados, pode-se aproveitar a distribuição da saliência média do contorno que é dada no gráfico (c) da figura 4.2.

Pode-se observar que ao definir um limite um pouco abaixo da saliência média do contorno médio de todos os contornos no excerto $C_{\overline{s}}$, se pode filtrar uma quantidade considerável de contornos não melódicos que não produzem efeito nos contornos da melodia. Definiram o limiar de vocalização τ_v com base na média de distribuição $\hat{Cs}$ e seu desvio padrão $\sigma_{C\hat{s}}$:

$$\tau_v = \hat{Cs} - \nu \cdot \sigma_{C\hat{s}} \tag{4.2}$$

em que o parâmetro ν vai determinar a leniência da filtragem, sendo que um valor mais alto dará mais falsos positivos e um valor baixo mais falsos negativos. Também foi comparado usando a saliência total do contorno, $C_{\sum s}$ mas não obtivemos melhores resultados do que o último, provavelmente devido à tendência da saliência total do contorno para contornos mais longos, o que não ajuda ao sucesso desta fase porque podemos estar a dispensar contornos curtos de melodia.

Já foi visto que se o sistema detetar vibrato num contorno, em principio este vai ser um contorno de melodia. Além disso, podemos ver no gráfico (b) da figura 4.2, que há uma queda repentina nos contorno não melódicos (acima de 20 cents e abaixo de 40 cents[2]), assim sendo a probabilidade de o contorno não pertencer à melodia é de 5%. Então, podemos usar esta informação para afinar o filtro de voz, fazendo com que os contornos onde há vibrato se tornem "imunes"(C_v = verdadeiro, $C_{\sigma p} > 40$), assegurando assim que os contornos de saliência baixos, com caraterísticas melódicas fortes não serão filtrados nesta fase.

- **Minimização de erros de oitava e valores atípicos de tom:** uma das principais fontes de erros nos sistemas de extração de melodia é a seleção de vários harmónicos da melodia correta em vez do F0 correto, sendo conhecido como um erro de oitava.

Quando é estudada uma frame isoladamente, a tarefa de tentar determinar se dois picos de saliência com distância de uma oitava foram produzidos por duas fontes separadas ou pela mesma fonte torna-se difícil. No entanto, se tivermos os contornos de altura podemos facilmente detetar a presença de oitavas duplicadas, uma vez que estas se manifestam como contornos com trajetórias idênticas com a distância de uma oitava entre elas.

Para comparação de trajetórias de contorno, é necessário efetuar o calculo da distância entre os valores de inclinação por frame para a região em que eles se sobrepõem e calcular a média sobre essa região. Se a distância média variar entre 1150 e 1250 cents os contornos são considerados de oitava duplicada. Na figura 4.2 (b) podemos observar nos segundos 3-4

[2]ver mais promenores em `https://en.wikipedia.org/wiki/Cent_(music)`

um exemplo de oitava duplicada, sendo que o contorno correto se encontra em 4000 cents e a duplicada em 2800 cents.

O método de minimização de erros de oitava proposto por [13] tira proveito da informação temporal de duas maneiras. Primeiro, tal como já foi mencionado, usa-se a criação de contornos de afinação para detetar oitavas duplicadas comparando com trajetórias de contorno e de seguida usa-se a relação entre contornos vizinhos para decidir qual das oitavas é a correta, sendo que este processo é baseado em duas suposições, que na maioria dos casos o contorno correto terá maior saliência do que a sua duplicação e que as melodias tendem a ter uma trajetória de pitch contínua, evitando grandes saltos, de acordo com os princípios de vozes [19]. De maneira a por em prática esses princípios foi calculada uma "média do tom da melodia"$\overline{P(t)}$, que é no fundo uma trajetória do tom que representa a evolução do tempo em grande escala do tom da melodia. Supostamente, quando temos uma oitava duplicada os contornos que se encontram antes e depois dessa oitava duplica irão puxar $\overline{P(t)}$ em direção à oitava correta, assim a oitava mais próximo de $\overline{P(t)}$ é selecionada como a oitava correta e a outra é descartada. Também se usa $\overline{P(t)}$ para remoção de valores atípicos.

A filtragem de "outliers"garante que não hajam grandes saltos na melodia e também pode filtrar contornos não sonoros que foram capturados por engano. A distância entre o contorno e $\overline{P(t)}$ é calculado pela média das distâncias por "frame". Este processo é sintetizado pelos seguintes passos:

1. Cálculo de $\overline{P(t)}$ em cada "frame"como a média ponderada da inclinação de todos os contornos presentes no quadro.
2. Suavização de $\overline{P(t)}$ usando um filtro médio deslizante de 5 segundos com um tamanho de salto de 1 "frame". Isso limita a taxa na qual a trajetória do tom da melodia pode mudar, garantindo a continuidade e evitando grandes saltos.
3. Deteção de pares de oitavas duplicadas e, para cada par, remoção do contorno mais distante de $\overline{P(t)}$.
4. Recomputação de $\overline{P(t)}$ usando os contornos restantes, seguindo as etapas 1-2.
5. Remoção dos "outliers"de afinação excluindo contornos a uma distância de mais de uma oitava de $\overline{P(t)}$.
6. Recomputação de $\overline{P(t)}$ usando os contornos restantes, seguindo as etapas 1-2.
7. Repetição das etapas 3 a 6 mais duas vezes, cada vez começa com todos os contornos que passam pelo estágio de detecção de abertura, mas usando a média de tom da melodia calculada mais recentemente $\overline{P(t)}$.
8. Passagem dos contornos restantes após a última iteração para o estágio final de seleção de melodia.

A média do tom $\overline{P(t)}$ que foi anteriormente calculada aproxima-se da trajetória verdadeira da melodia quando a contribuição de cada contorno é ponderada pela sua saliência total $C_{\sum s}$.

Isto faz com que a média para contornos salientes se apresente por um período mais longo de tempo, o que faz com que sejam mais propensos a pertencer à melodia.

Um exemplo de execução das etapas 1-6 é o fornecido na figura 4.3. No gráfico (a) podemos ver um conjunto de contornos e a respetiva média do tom da melodia modificada $\overline{P(t)}$. No gráfico (b) vemos oitavas duplicadas, sendo que a oitava mais distante da média é removida. Em seguida, no gráfico (c) podemos ver a média $\overline{P(t)}$ recalculada e os valores atípicos de pitch são detetados e eliminados. Por fim, no gráfico (d) podemos ver que $\overline{P(t)}$ já foi recalculado e é exibido no gráfico junto com os contornos restantes.

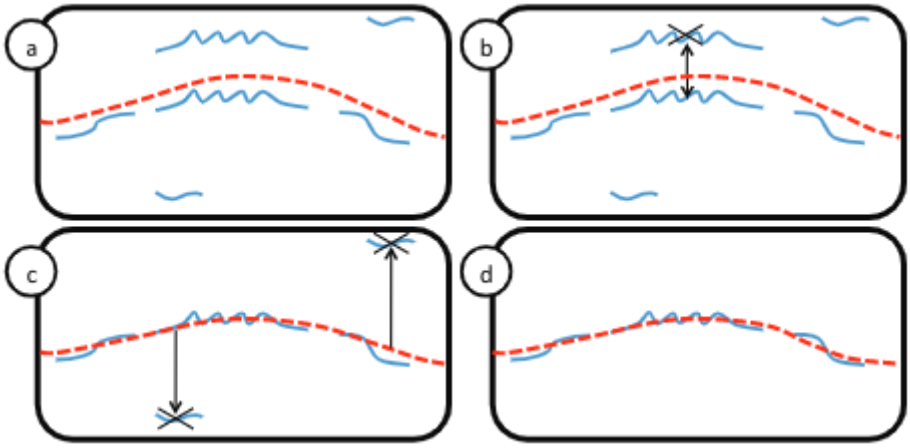


Figura 4.3: Figura retirada do artigo [13], removendo oitavas duplas e valores atípicos. (a) Etapas 1-2: a média do tom da melodia modificada pelas iniciais $\overline{P(t)}$ é calculada (linha vermelha tracejada). (b) Etapa 3: uma oitava duplicada é detectada e removida. (c) Etapas 4-5: P (t) é recalculado e dois valores discrepantes de pitch são removidos. (d) Etapa 6: $\overline{P(t)}$ é recalculado.

- **Seleção final da melodia:** nesta última fase o objetivo é seleccionar os picos que pertencem à melodia final dos contornos que sobram, sendo que cada pico representa um candidato F0.

Em outros sistemas esta etapa pode envolver rastreamento de pico o que torna todo este processo difícil, no entanto no nosso sistema já tivemos em consideração este problema pelo processo de criação, caracterização e filtragem dos contornos, ou seja, normalmente teremos apenas um pico para escolher, quando isto não acontece a melodia seleccionada como pico que pertence ao contorno é a que tem maior saliência total ($C_{\sum s}$). Se não houver nenhum contorno então a "frame"é surda. De maneira a avaliar o pitch bruto e a precisão do "chroma", é fornecida uma estimativa de F0 para "frames"não sonoros, selecionando o pico de contorno mais saliente que estava presente nessas "frames"antes da filtragem de contorno.

Na figura 4.4 podemos ver um exemplo do processo completo de extração de melodia de um excerto apresentado anteriormente no gráfico (a) da figura 4.1.

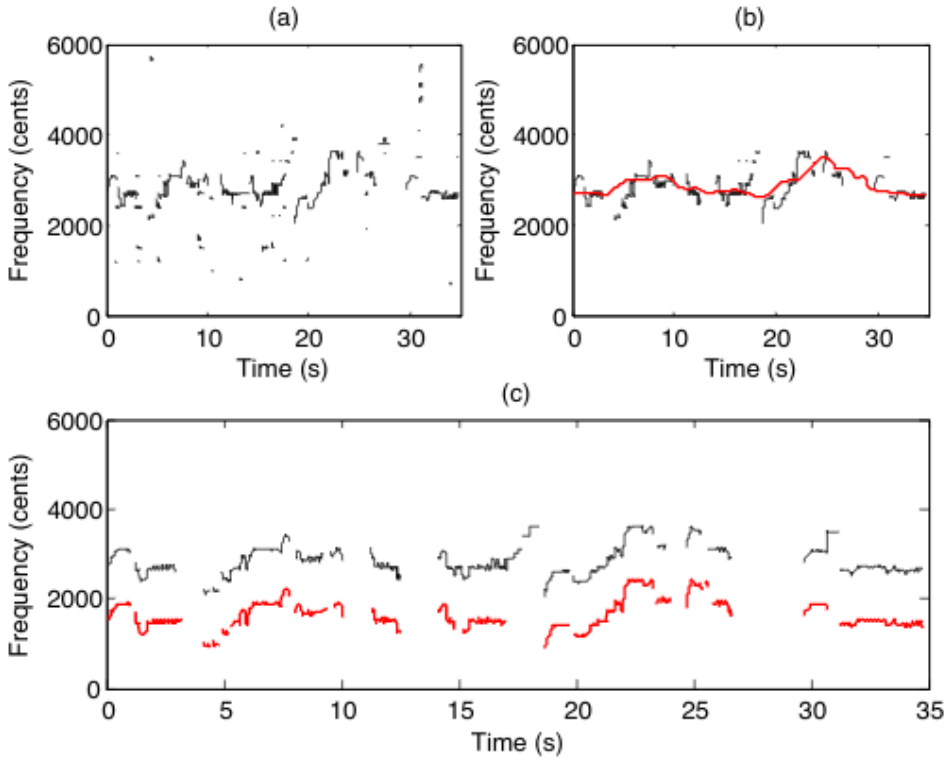


Figura 4.4: Figura retirada do artigo [13], contêm excerto do jazz vocal: (a) todos os contornos de altura criados, (b) contornos após a filtragem e a média do tom da melodia (linha vermelha grossa), (c) melodia extraída final (preto) e "ground truth"(vermelho grosso, descida uma oitava para maior clareza), que é um termo usado em vários campos para se referir a informações que são conhecidas como reais ou verdadeiras, fornecidas por observação direta e medição, em oposição às informações fornecidas por inferência.

4.2 Abstração de representação

Após a extração das sequências de melodia e linha de baixo (parte instrumental com as notas mais graves), devemos escolher uma representação adequada para calcular a similaridade musical. Sabemos que o algoritmo de correspondência pode lidar com a transposição, sendo que este processo é caraterizado pela alteração da altura da nota ou de um conjunto de notas, ou seja, quando se transpõe uma música muda-se a tonalidade em que esta se encontra, então podemos utilizar a representação extraída tal como se encontra de maneira a comparar as sequências F0 diretamente.

Para além da alteração do tom e do tempo de duração da melodia cantada, os músicos podem alterar a oitava cantada de maneira a se ajustar melhor à sua extensão vocal. O uso de efeitos expressivos (como glisando ou vibrato) vai variar em cada versão, assim sendo devemos ter como objetivo uma representação que se abstraia de informações e detalhes específicos da performance, enquanto mantém a progressão tonal médica básica. Desta maneira, definimos os seguinte tipos de

abstração de informações:

- **Abstração de semitons:** irá quantizar a informação de pitch em semitons, ajudando a remover efeitos expressivos locais.
- **Abstração de oitava:** irá mapear todas as informações de pitch numa única oitava, ajudando a resolver mudanças de oitava da melodia.
- **Abstração de intervalo:** irá substituir a informação de pitch absoluto pela diferença entre valores de pitch consecutivos.

Antes da aplicação de qualquer um dos tipos de abstração anteriormente indicados, todos os valores de frequência terão que ser convertidos numa escala de cents. De seguida, comparamos os diferentes graus de abstração aplicados às sequência de melodia: nulo, semitom, intervalo, intervalo + semitom e semitom + oitava. Para o nosso algoritmo usamos a abstração semitom + oitava para os descritores de melodia e linha de baixo.

O processo de abstração processa-se da seguinte maneira. Primeiro todos os valores de frequência são convertidos em cents. Em seguida, os valores de afinação são quantizados em semitons e mapeados numa única oitava. Após isto, reduz-se o comprimento da sequência, reduzindo cada 150 "frames"como um histograma de classe pitch, onde a contribuição de cada "frame"para o histograma é ponderada pela saliência da melodia naquela mesma "frame", determinada pelo algoritmo de extração de melodia. Isto faz com que se produza uma sequência curta onde cada "frame"é um vetor de 12 bins que representa a distribuição por classes do tom da melodia em cerca de meio segundo. Para a etapa de resumo, em primeiro lugar reduz-se o comprimento da sequência e, portanto reduz-se o tempo de cálculo do algoritmo de correspondência. De seguida, reduz-se a influência de mudanças de tom muito curtas. E por fim, o vetor de cada "frame"é normalizado pelo valor do seu bin mais alto.

4.3 Algoritmo de comparação de melodia

Existem vários algoritmos com a tarefa de comparar melodia, um deles é o DTW - *Dynamic time warping*, que serve para comparar e alinhar duas séries temporais. Este algoritmo é utilizado para alinhar qualquer tipo de dado que obedeça a uma ordem temporal, como vídeo, áudio ou imagens. Existem outros algoritmos que servem, neste caso, para comparar áudios através da similaridade entre eles, como é o caso do Qmax e do Dmax. Tratam-se de algoritmos DTW com penalizações. Os subcapítulos seguintes abordam estes algoritmos em pormenor.

4.3.1 QMax

A função Qmax tenta calcular o comprimento do segmento de tempo mais longo no qual dois excertos musicais contém padrões semelhantes com base no CRP ("Cross Recurrence Plot"). Um CRP, aqui denotado como **C**, é uma matriz de similaridade binária, em que o elemento $c_{p,q}$ é definido como "1"quando existe correspondência e "0"caso contrário. Para saber mais acerca do CRP consultar [20]. A matriz acumulativa (denotada como **O**), conta os elementos a "1"que são calculados com base em **C** (matriz de distâncias locais binarizada), de acordo com a equação 4.3:

$$o_{p,q} = \begin{cases} \max\{o_{p-1,q-1}, o_{p-2,q-1}, o_{p-1,q-2}\} + 1, & \text{se } c_{p,q} = 1 \\ \max\{0, o_{p-1,q-1} - \gamma(c_{p-1,q-1})\}, & \\ \mathrm{o}_{p-2,q-1} - \gamma(c_{p-2,q-1}), & \\ \mathrm{o}_{p-1,q-2} - \gamma(c_{p-1,q-2}), & \text{se } c_{p,q} = 0 \end{cases} \tag{4.3}$$

onde $\gamma(z)$,

$$\gamma(z) = \begin{cases} \gamma_o, & \text{se z = 1} \\ \gamma_e, & \text{se z = 0} \end{cases} \tag{4.4}$$

é a penalização aplicada se a similaridade local não for 1, e que apresenta dois valores (γ_o e γ_e) consoante o valor da distância local do caminho parcial até $c_{p,q}$ (ver figura 4.5a)). Assim sendo, o valor $Qmax$ é o máximo de elementos $o_{p,q}$ de **O**, assim sendo:

$$Q_{max} = \max(o_{p,q}) \tag{4.5}$$

Uma vez que o comprimento do excerto musical quase nunca tem o mesmo tamanho que a música de referência Q_{max} é dado como uma distância (e não como similaridade) através do seu inverso e normalizado com a raiz quadrado do nº de tramas de referência:

$$D^{Q_{max}}(i,l) = \sqrt{N_l}/Q_{max} \tag{4.6}$$

onde $D^{Q_{max}}(i,l)$ é a distância Q_{max} normalizada e N_l é o número de tramas incluídas do excerto musical em análise.

4.3.2 DMax

No algoritmo D_{max} a matriz acumulativa (denotada **O**) é calculada de outra maneira:

$$o_{p,q} = \begin{cases} \max\{\hat{o}_{p-1,q-1}, \hat{o}_{p-2,q-1} + c_{p-1,q}, \\ \quad \hat{o}_{p-1,q-2} + c_{p,q-1}, \\ \quad \hat{o}_{p-3,q-1} + c_{p-2,q} + c_{p-1,q}, \\ \quad \hat{o}_{p-1,q-3} + c_{p,q-2} + c_{p,q-1} + 1, & \text{se } c_{p,q} = 1 \\ \max\{0, \hat{o}_{p-1,q-1}, -\gamma(c_{p-1,q-1}), \\ \quad \hat{o}_{p-2,q-1} + c_{p-1,q} - \gamma(c_{p-2,q-1}), \\ \quad \hat{o}_{p-1,q-2} + c_{p,q-1} - \gamma(c_{p-1,q-2}), \\ \hat{o}_{p-3,q-1} + c_{p-2,q} + c_{p-1,q} - \gamma(c_{p-3,q-1}), \\ \hat{o}_{p-1,q-3} + c_{p,q-2} + c_{p,q-1} - \gamma(c_{p-1,q-3}), & \text{se } c_{p,q} = 0 \end{cases} \tag{4.7}$$

Assim sendo, a similaridade baseada em D_{max}, é chamada de $D^{D_{max}}(i,l)$ e é calculada da seguinte maneira:

$$D^{D_{max}}(i,l) = \sqrt{N_l}/D_{max} \tag{4.8}$$

onde N_l é o número de "frames"incluídos do excerto musical e D_{max} é calculado da seguinte maneira:

$$D_{max} = \max(\hat{o}_{p,q}) \tag{4.9}$$

Os caminhos possíveis para o Q_{max} e o D_{max} são mostrados na figura 4.5. Como se pode observar, a única diferença entre as duas é o o número de caminhos locais possíveis até chegar a um ponto (p,q) na matriz de similaridade acumulada. Um exemplo de aplicação do algoritmo *Qmax* pode ser visto na figura 5.5.

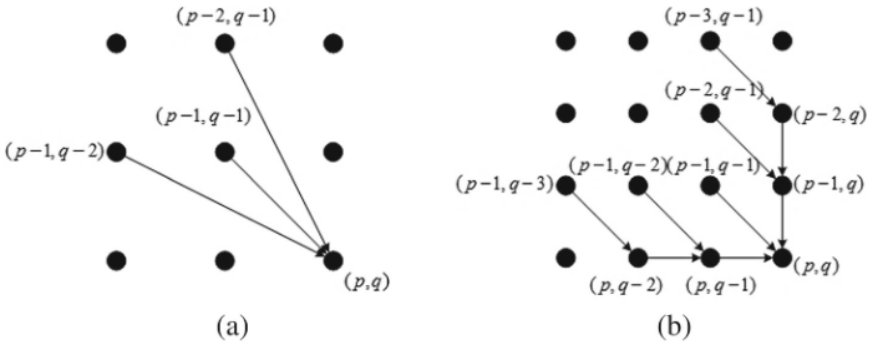


Figura 4.5: Retirado do artigo [21], contendo possíveis restrições de alinhamento em (a) Qmax e (b) Dmax

Capítulo 5

Descrição do Trabalho Efetuado

5.1 Implementação do Algoritmo QBH

Criação da BD de referência com covers

Como já foi referido anteriormente, a base de dados de referência não foi fornecida pelo MTG, por questões de direitos autorais, sendo apenas fornecido os nomes e autores das respetivas músicas. Foi efetuado o download das músicas através do programa *Youtube-DL*, começando pelas 81 músicas que nos foram fornecidas como sendo de referência. Numa segunda fase fizemos o download dos covers, acrescentando 179 covers, prefazendo um total de 260 músicas de referência.

Criação de queries cantados

As queries fornecidas pela BD MTG foram 118, no entanto gravamos ainda 4 queries com o intuito de poder testar o funcionamento do algoritmo em casos com presença de ruído ou com excertos musicais cantados fora de tom.

Uso dos algoritmos do software "Essentia"

O *sotfware Essentia* é uma biblioteca C++ de código aberto com ligações *Python* e *JavaScript* para análise de áudio e recuperação de informações musicais baseadas em áudio.

Duas das funções do *software Essentia* utilizadas no nosso código que mais relevância tiveram foram *ChromaCrossSimilarity* e *CoverSongSimilarity*[1].

Criação dos "scripts" *Python*

Para a testagem do algoritmo de QBH foram criados vários "scripts"de *Python*, através do *Jupyter Notebook*[2].

[1] consultar mais promenores em `https://essentia.upf.edu/reference/std_ChromaCrossSimilarity.html` e `https://essentia.upf.edu/reference/std_CoverSongSimilarity.html`

[2] Os *scripts* podem ser consultados através do seguinte link: `https://github.com/anacalhau/query_by_humming/tree/main/src`. Acesso em: 20 de agosto de 2021.

Os "scripts"criados foram os seguintes:

- **qbh**

Este "script"foi construído com o objectivo de poder verificar o MRR (*mean reciprocal rank*, descrito de seguida) e os valores dos Tops 1, 3, 5 e 10, onde se considera a posição em que a música de referência correspondente ao query, é apresentada numa lista ordenada por ordem decrescente contendo o número de similaridade. Os resultados dos Tops que foram obtidos pelo MTG são nos apresentados no artigo [11].

O MRR é uma medida estatística que tem como objetivo a avaliação de qualquer processo que produza uma lista de respostas possíveis para uma amostra,assim o MRR (que varia de 0 no pior caso a 1 no melhor caso) para N queries é definido como:

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{r_i} \tag{5.1}$$

onde r_i é a ordem ("rank") em que a referência correta aparece na consulta do query i e N é o número de queries.

Neste "script"começamos por carregar os ficheiros de referência, de seguida carregamos os arquivos que contém os metadados e fazemos o carregamento dos ficheiros que contêm as queries (N). Tanto quando se fez o carregamento das músicas de referência como das queries foram extraídos os "chromas", através de uma função que efetuou a redução dos "chromogramas".

```
def mean_reciprocal_rank(rs):
    """Score is reciprocal of the rank of the first relevant item

    First element is 'rank 1'.  Relevance is binary (nonzero is relevant).

    Example from http://en.wikipedia.org/wiki/Mean_reciprocal_rank
    >>> rs = [[0, 0, 1], [0, 1, 0], [1, 0, 0]]
    >>> mean_reciprocal_rank(rs)
    0.61111111111111105
    >>> rs = np.array([[0, 0, 0], [0, 1, 0], [1, 0, 0]])
    >>> mean_reciprocal_rank(rs)
    0.5
    >>> rs = [[0, 0, 0, 1], [1, 0, 0], [1, 0, 0]]
    >>> mean_reciprocal_rank(rs)
    0.75

    Args:
        rs: Iterator o1f relevance scores (list or numpy) in rank order
            (first element is the first item)

    Returns:
        Mean reciprocal rank
    """
    rs = (np.asarray(r).nonzero()[0] for r in rs)
    return np.mean([1. / (r[0] + 1) if r.size else 0. for r in rs])
```

Após isto fizemos as comparações através do *CoverSongSimilarity*, função do *software Essentia*, já mencionada anteriormente, que permite fazer comparações entre todos os queries e todas as músicas de referência. Esta função foi aplicada tanto às músicas de referência polifónicas como aos queries monofónicos. Depois disto ordenamos os valores da similaridade e procedemos ao cálculo do MRR e dos Tops que foram calculados da seguinte maneira:

```
relevance = []
for query in results:

    rel = [ref_dic[ref["ref"]]["songid"] == query_dic[query]["songid"] for ref in results[query]["res"]]
    if any(rel):
        relevance.append(np.array(rel).astype(int))
        results[query]["relevance"]=np.array(rel).astype(int)

MRR = mean_reciprocal_rank(relevance)
print(MRR)
```

```
0.20431167968191633
```

Top 1- 3 - 5 - 10

```
relevance = np.array(relevance)
tops = [0]*4
for k,t in enumerate([1,3,5,10]):
    tops[k] = sum(relevance[:,:t])/relevance.shape[0]

print(tops)
```

```
[0.14285714285714285, 0.21008403361344538, 0.226890756302521, 0.29411764705882354]
```

Na tabela 5.1 apresentamos os resultados que foram obtidos ao longo dos vários testes que elaboramos.

	1º teste	**2º teste**	**3º teste**	**4º teste**
Top 1	12.61%	14.29%	24.37%	31.09%
Top 3	21.85%	21.01%	35.29%	34.45%
Top 5	26.89%	22.69%	37.82%	35.29%
Top 10	32.77%	29.41%	46.22%	41.18%
MRR	0.205507	0.204345	0.314202	0.364914

Tabela 5.1: Resultados em percentagem do sistema de procura

O primeiro teste foi feito sem qualquer tipo de alteração ao código original, apenas replicamos o que foi feito pelo MTG de maneira a comparar os nossos resultados com o do artigo [11], podendo assim concluir que mesmo não utilizando as mesmas músicas de referência os resultados obtidos foram muito semelhantes.

No segundo teste tivemos em conta apenas o corte de ruído inicial e final das músicas o que originou uma ligeira melhoria, no entanto não muito significativa. Este processo foi feito com recurso ao *Audacity* onde foram analisadas cuidadosamente todas as queries, caso a caso. Os valores deste teste são os que aparecem na figura anterior.

No terceiro teste foram adicionadas à base de dados mais músicas de referência, sendo algumas delas covers das músicas já existentes de maneira a melhorar as probabilidades do sistema achar a música correspondente, podemos verificar que esta alteração trouxe uma melhoria de cerca de 10% no Top 1 e percentagens ainda maiores nos seguintes Tops.

Por fim, o 4º e último teste foi realizado após a normalização, podendo observar-se uma melhoria significativa no Top 1, mas algum decréscimo nos restantes Tops. O processo de normalização foi feito usando o quociente do valor máximo de similaridade pelo número de tramas.

- **melody_extraction**

Neste "script"podemos perceber o processo de extração de melodia com mais cuidado, através de vários gráficos. No gráfico 5.1 podemos ver com detalhe a extração de melodia com hz quantizados para um determinado exemplo. Podemos observar alguns erros na deteção de melodia mas que acabam por ser aceitáveis num todo, não comprometendo o objectivo final.

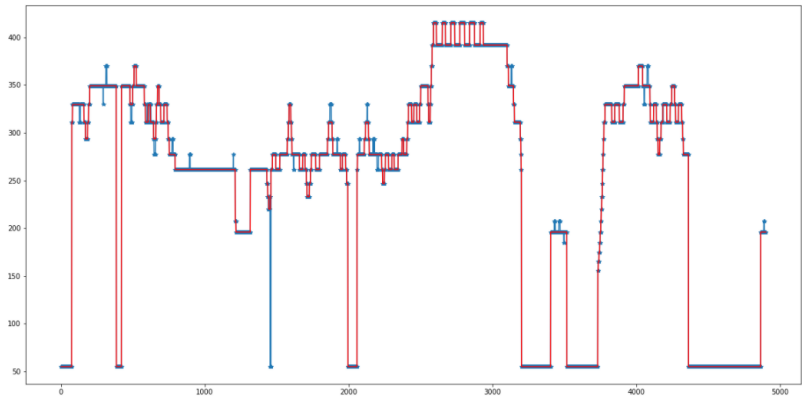


Figura 5.1: Plot com Hz quantizados, onde a linha azul representa os valores em cents quantizados e a linha vermelha representa os valores em cents quantizados e filtrados

Na linha a azul temos a representação dos valores sem a aplicação de qualquer filtro, a vermelho podemos ver a aplicação de um filtro de mediana numa matriz N-dimensional. A matriz será automaticamente preenchida com zeros.

De seguida, nas figuras 5.2 e 5.3 podemos ver os "chromogramas"que foram obtidos para o mesmo exemplo:

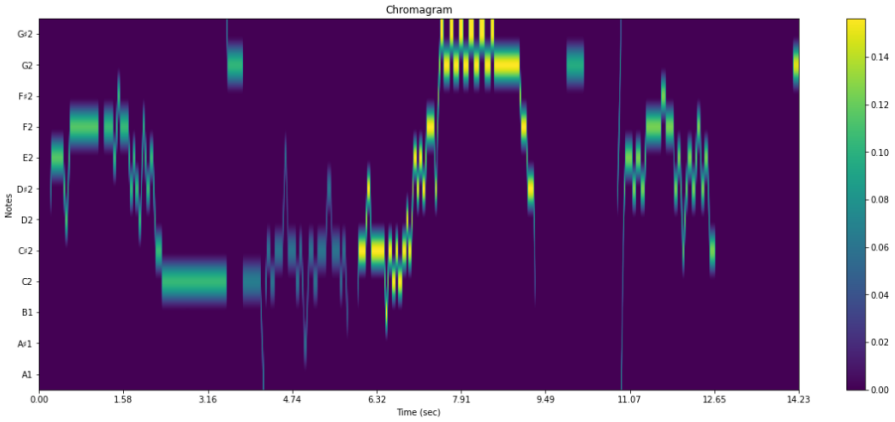


Figura 5.2: Chromograma extraído a partir de uma query de exemplo

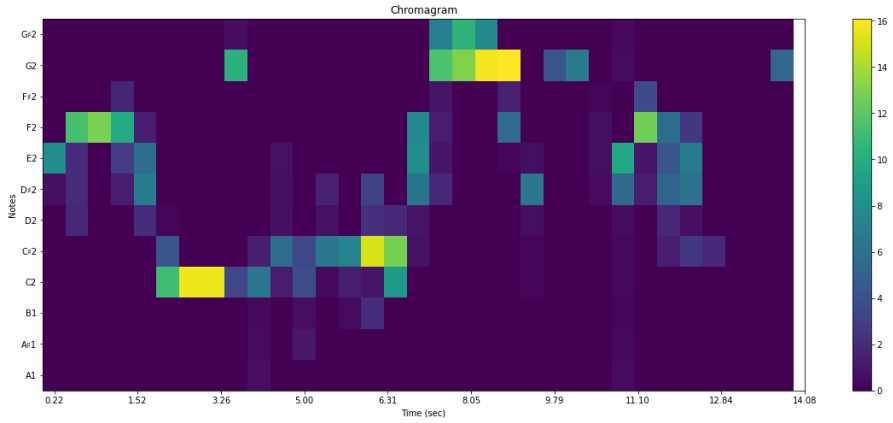


Figura 5.3: Chromograma reduzido extraído a partir de uma query de exemplo

Na figura 5.3 podemos ver um "chromograma"reduzido. Para obter este gráfico começamos por utilizar a função *arange* para as frequências, que retorna valores com espaçamento uniforme num determinado intervalo. De seguida fizemos as conversões de semitons para hertz e de hertz para cents. E por fim colocamos todos os cents que fossem menores que 1, a 0 de maneira a proceder à redução do "chromograma".

- **melody_extraction_stepbystep**

Neste script examinamos novamente o processo de extração de melodia mas passo a passo de maneira a que se torne ainda mais fácil a sua percepção.

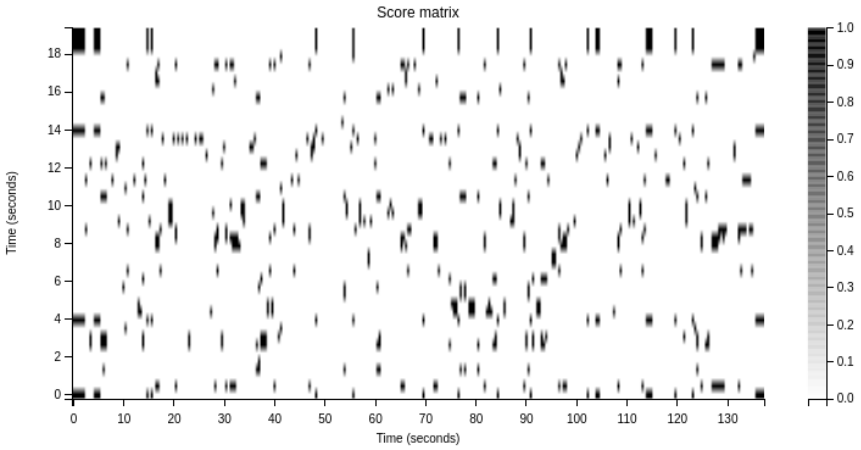


Figura 5.4: Score Matrix Csm

Na figura 5.4 podemos ver uma matriz binário (0's e 1's), onde está representada a *score matrix Csm (cross-similarity matrix).*

O Csm calcula uma matriz de similaridade cruzada euclidiana de duas sequências de recursos de tramas. Os valores de similaridade podem ser opcionalmente binarizados. Os parâmetros padrão para binarização são otimizados para identificação de música usando recursos de "chroma".

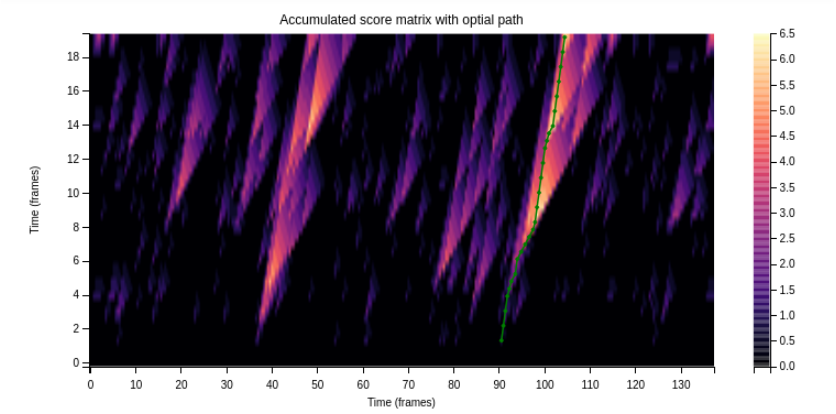


Figura 5.5: Matriz de acumulação do Csm

Na figura 5.5 podemos verificar onde há maior probabilidade de correspondência (representado a amarelo), podendo observar a existência de vários "picos"a amarelo, uns mais fortes que outros. A linha a verde representa o **backtracking**, ou seja as tramas em que houve correspondência.

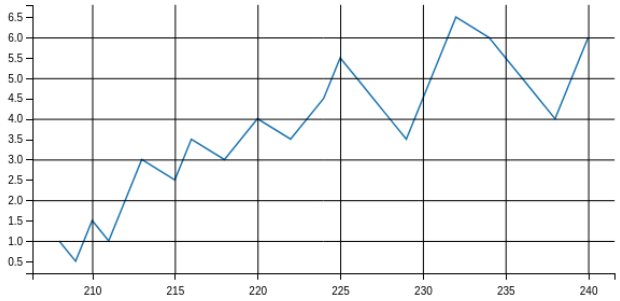


Figura 5.6: Gráfico representativo da similaridade em função do número de tramas

No gráfico da figura 5.6 podemos ver as penalizações que são atribuídas durante o caminho, sendo que tanto a *disOnset*, penalidade para o início da interrupção, como a *disExtension*, penalidade para extensão de interrupção, tem o valor de 0.5. São consideradas interrupções todos os valores da *Score Matrix Csm* que são apresentados a 0.

- **Qmax**

À semelhança do que já foi observado no "script"anterior, podemos observar o funcionamento do algoritmo Qmax, neste caso com um novo exemplo, observando ainda o seu *backtracking* (representado a vermelho). A diferença deste "script"para o anterior é que este foi construído para verificar o funcionamento do Qmax exclusivamente e o anterior tinha mais detalhes a serem analisados.

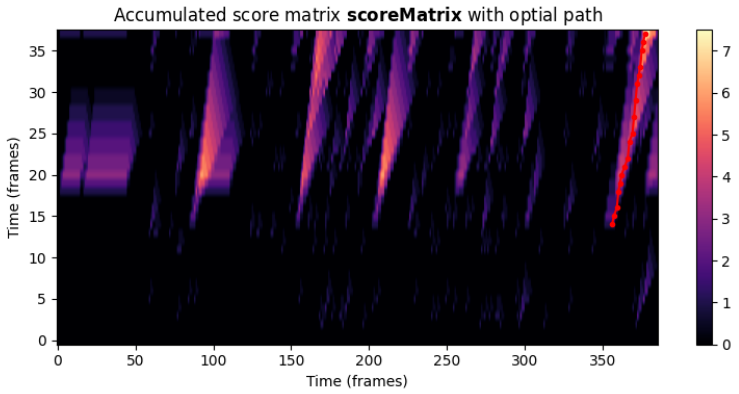


Figura 5.7: Backtracking do Qmax (representado a vermelho) na Matrix de acumulação do Csm

Na figura 5.7 podemos ver que foram encontrados quatro "picos"e apenas um deles acabou por ser escolhido como sendo o local onde foi encontrado o query. Esta escolha é feita pelo "pico"que tiver maior similaridade.

- **Looking_For_Music**

Neste "script"temos o algoritmo de procura de músicas com um interface gráfico simples mas intuitivo pronto a ser utilizado pelo utilizador. Este "script"foi construído com base no que observamos nos anteriores mas com o intuito de ser o único com a aplicação final do nosso trabalho.

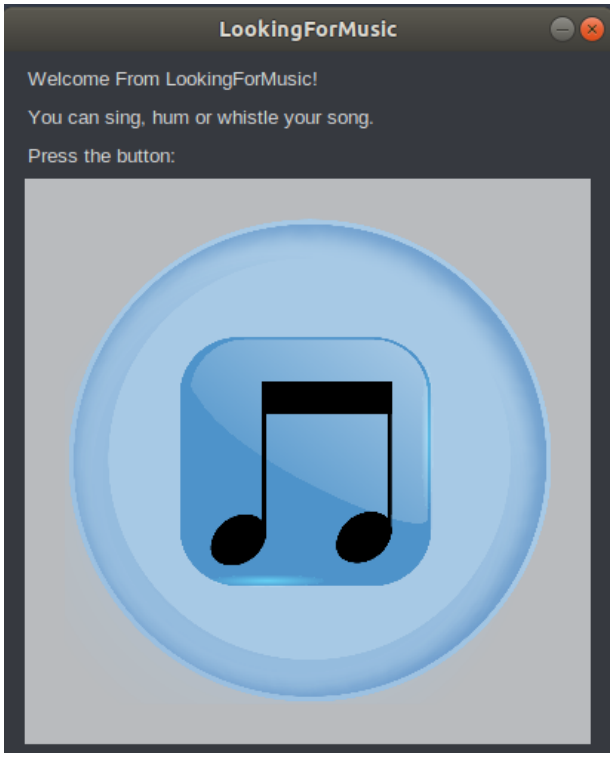
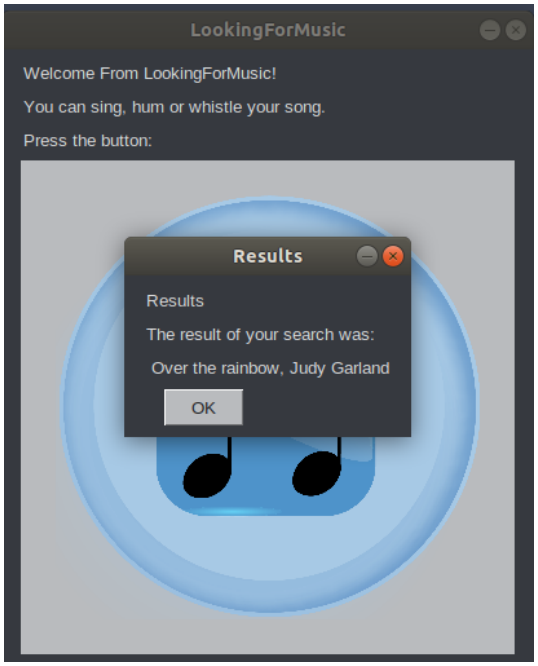
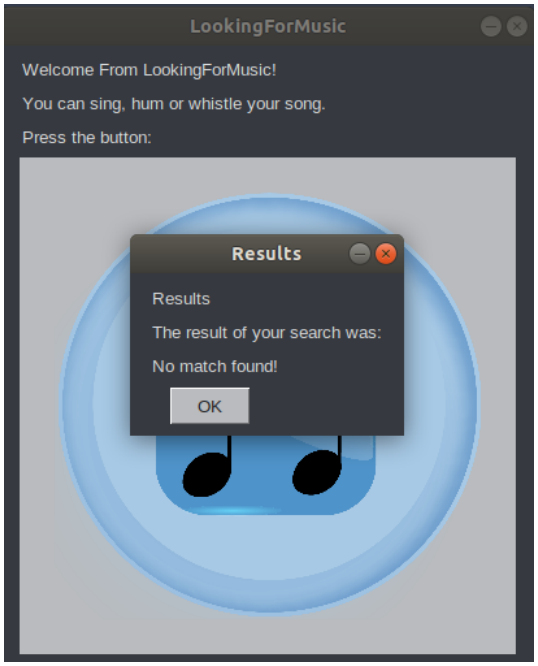


Figura 5.8: Interface Gráfico

De seguida o utilizador deverá carregar no botão azul e gravar o seu query, cantando, assobiando ou trauteando um excerto de uma canção. No caso de a música pertencer à base de dados e ser encontrada irá aparecer algo deste género:



caso contrário, se a música não for encontrada ou não pertencer à base de dados o resultado será o seguinte:



Uma procura bem sucedida é obtida caso a música de referência com o melhor resultado (maior similaridade) tenha o nível de superioridade desejado, i.e., não existe nenhuma outra música de referência com 90% ou mais do seu valor de similaridade. Podemos ver na figura 5.9 um exemplo de procura bem sucedida:

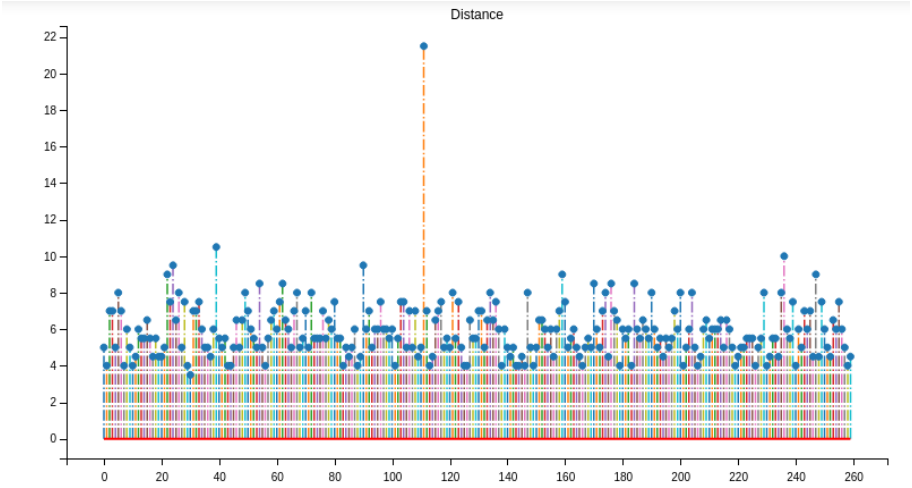


Figura 5.9: Exemplo de procura com valores de similaridade para todas as músicas existentes na base de dados de referência

Após a conclusão desta dissertação podemos dizer que os resultados obtidos ficaram um pouco aquém do esperado. No entanto se compararmos com os resultados de outras pesquisas da área podemos verificar que nos últimos anos não tem havido melhorias significativas nos resultados obtidos.

Capítulo 6

Conclusões

Neste trabalho foi desenvolvido um algoritmo de procura musical, através de um outro já desenvolvido de extração de melodia. O nosso objetivo foi de obter os melhores resultados possíveis nessa procura, tendo optimizado o código.

Até ao momento, o uso de diferentes representações musicais para calcular a similaridade de versões não tem tido grandes desenvolvimentos. Achamos que é uma área que tem deixado de ser explorada ao longo do tempo e que merece mais atenção.

Foram estudadas técnicas de procura por similaridade o que nos permitiu ter várias ferramentas à nossa disposição de maneira a tentar sempre utilizar as que nos trouxessem melhores resultados.

Embora ainda haja muito trabalho a ser feito nesta área, os resultados apresentados aqui servem como uma prova de conceito e esperamos levar ao desenvolvimento futuro de soluções QBH de alto desempenho totalmente automatizadas.

Bibliografia

[1] Steffen Pauws. Cubyhum: a fully operational "query by humming"system. *ISMIR 2002, 3rd International Conference on Music Information Retrieval, Paris, France, October 13-17, 2002, Proceedings*, 2002.

[2] M. Rocamora, Pablo Cancela, and A. Pardo. Query by humming: Automatically building the database from music recordings. *Pattern Recognit. Lett.*, 36:272–280, 2014.

[3] Chotirat Ann Ratanamahatana and Eamonn J. Keogh. Everything you know about dynamic time warping is wrong. 2004.

[4] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, M Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Data mining a trillion time series subsequences under dynamic time warping. *IJCAI International Joint Conference on Artificial Intelligence*, pages 3047–3051, 08 2013.

[5] Peter H Sellers. The theory and computation of evolutionary distances: Pattern recognition. *Journal of Algorithms*, 1(4):359 – 373, 1980.

[6] Lawrence R. Rabiner and B. H. Juang. Fundamentals of speech recognition / lawrence rabiner, biing-hwang juang. 1, 1993.

[7] Paul Masri and Andrew Bateman. Improved modeling of attack transients in music analysis-resynthesis. pages 100–103, 1996.

[8] W. J. Dowling. Assimilation and tonal structure: Comment on castellano, bharucha, and krumhansl. *Journal of Experimental Psychology*, pages 417–420, 1984.

[9] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, M Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Data mining a trillion time series subsequences under dynamic time warping. *IJCAI International Joint Conference on Artificial Intelligence*, pages 3047–3051, 08 2013.

[10] C. Lin, J. Ding, and C. Hu. Advanced query by humming system using diffused hidden markov model and tempo based dynamic programming. pages 1–7, 2016.

[11] Justin Salamon, Joan Serrà, and Emilia Gómez. Tonal representations for music retrieval: From version identification to query-by-humming. *International Journal of Multimedia Information Retrieval, special issue on Hybrid Music Information Retrieval*, 2013.

[12] Joan Serrà Julià. *Identification of versions of the same musical composition by processing audio descriptions*. PhD thesis, Universitat Pompeu Fabra, Barcelona, 2011. TDX link: http://hdl.handle.net/10803/22674.

[13] Justin Salamon and Emilia Gómez. Melody extraction from polyphonic music signalsusing pitch contour characteristics. 2012.

[14] X. Serra. Musical sound modeling with sinusoids plus noise. page 91–122, 1997.

[15] J. L. Flanagan and "Bell SystemsTechnical Journal vol. 45 pp. 1493–1509 1966. R. M. Golden, "Phase vocoder.

[16] A. Klapuri. Multiple fundamental frequency estimation by summingharmonic amplitudes. 216–221.

[17] Justin Salamon, Emilia Gómez, and Jordi Bonada. Sinusoid extraction and salience function design for predominant melody estimation. 2011.

[18] P. Herrera and J. Bonada. Vibrato extraction and parameterization in thespectral modeling synthesis framework. page 107–110, 1998.

[19] P. Herrera and J. Bonada. Tone and voice: A derivation of the rules of voice-leading from perceptual principles. page 1–64, 2001.

[20] Andrzejak RG (2009) Cross recurrence quantification for cover song identification. JPhys 11(9):093–017 Serra J, Serra X.

[21] Haidong Xiao Ning Chen, Wei Li. Fusing similary functions for cover song identification. 2018.