



DYNAMIC TIME WARPING PARA CHORD FOLLOWING

Aluno: Bruno Tavares de Melo Gomes
Orientador: Giordano Ribeiro Eulalio Cabral

Recife, Junho de 2018

Universidade Federal de Pernambuco
Centro de Informática
Graduação em Ciência da Computação

DYNAMIC TIME WARPING PARA CHORD FOLLOWING

Aluno: Bruno Tavares de Melo Gomes
Orientador: Giordano Ribeiro Eulalio Cabral

Trabalho de Graduação apresentado no
Centro de Informática da Universidade
Federal de Pernambuco como requisito
parcial para a obtenção do grau de
Bacharel em Ciência da Computação.

Recife, Junho de 2018

*There is a gap between what can
be proved and what can be done.*

Sumário

Resumo	4
Abstract	5
Introdução	6
Page Turners e Score Following	6
Alinhamento de Acordes e Audio Chord Estimation	6
Estado da arte	8
Score Following	8
Automatic Chord Estimation (ACE)	8
Decisões de Projeto	9
Testes com DTW Convencional	10
Dynamic Time Warping	10
Dados	10
Input	10
Modificações na Abordagem de Araújo	11
Output	11
Metodologia de comparação	11
Resultados	11
Testes com Online DTW Convencional	12
O Online DTW	12
A Janela de Alinhamento	12
Métodos de Alinhamento	12
Método Arriscado	12
Método Conservador	12
Método Seguro	13
Metodologia de Testes	13
Abordagem com Parâmetros Padrão	13
Abordagens com Mudança de Parâmetros	15
Testes com $C = 2$ e MaxRunCount influenciado	18
Conclusão e Pesquisa Futura	20
DTW Online com Expansão apenas através do tempo	20
DTW Online com ACE Anterior	20
DTW Online com C Enviesado	20
DTW Online com MaxRunCount Enviesado	21
DTW Online com MaxRunCount e C Enviesados	21
Referências	22

Resumo

Ao aprender a tocar instrumentos musicais, uma prática comum é acompanhar cifras de músicas, amplamente disponíveis em sites a exemplo do Cifra Club¹. Cifras mostram em que parte da música os acordes devem ser tocados. Sendo um acorde um conjunto de duas ou mais notas tocadas ao mesmo tempo, a maneira exata como eles serão executados fica livre à interpretação do leitor.

Visto que é uma prática comum utilizar cifras para aprender a tocar, torna-se interessante um sistema que auxilie o estudante no processo. É proposta uma aplicação capaz de dar feedback (mostrar quando um acorde é tocado) e de mostrar na tela a parte da música que está sendo tocada no momento (eliminando a necessidade de parar de tocar para mudar de página).

O objetivo deste trabalho é apresentar uma solução para o problema de sincronizar o que está sendo tocado em tempo real e uma cifra musical conhecida. Chamamos este problema de Chord Following. Existe um problema intimamente atrelado ao proposto, conhecido como Score Following. Este problema, principalmente para sons com mais de uma nota tocada ao mesmo tempo, é conhecido na computação como um problema difícil.

Recentemente foram publicadas técnicas capazes de resolver o problema de Score Following com precisão superior a 90%². Esse trabalho irá tentar utilizar essas técnicas para resolver o problema proposto. Foi escolhido trabalhar com o violão, devido à sua popularidade no Brasil. Contudo, ao resolver o problema para o violão esperamos resolvê-lo também para outros instrumentos, visto que o violão torna o problema mais difícil do que guitarra e piano, por exemplo.

Abstract

A common practice when learning to play a musical instrument is to follow chords of the songs. These are broadly available in chords websites such as Cifra Club¹. Chords show in which part of the music the musician should play a given chord. Being a chord a set of one or more notes played at once, the exact manner that they will be executed depends on the interpretation of the reader.

As the use of chords is a common practice when learning how to play, a system that is able to help the student in the process becomes interesting. This work proposes an application capable of giving feedback (show when a chord is being played) and show on the screen the part of the song that is being played at the moment (eliminating the need to stop playing to scroll through pages).

The objective of this work is to present a solution to the problem of synchronizing what is being played in real time and its known chords. We call this problem Chord Following. There is a known problem that is intrinsically related to the proposed, named Score Following. This problem, especially for sounds where more than one note is played at once is known in computing as a hard problem.

Recently there were publications of techniques capable of solving the Score Following problem with a precision of over 90%². This work will attempt to use these techniques to solve the proposed problem. It was chosen to focus on acoustic guitars, because of its popularity in Brazil. Nevertheless, solving the problem for this instrument is harder than solving for Electric Guitar and Piano, for example. By solving it to acoustic guitar we expect to solve it for other instruments too.

Introdução

Este trabalho tem o objetivo de tentar criar uma solução para o problema aqui chamado de Chord Following. Este problema será dado como resolvido se for criado um sistema capaz de, dada ao algoritmo a cifra da música a ser tocada, reconhecer sem latência muito alta o acorde sendo tocado ao vivo num dado momento. A latência deve ser mencionada pois um acorde é um conjunto de notas e é muito provável que uma solução precise levar em conta mais de uma janela de informação para chegar a uma conclusão confiável.

Não foi encontrado na literatura nenhum trabalho que trata deste problema em específico. Existem, porém, alguns trabalhos com objetivos similares.

Page Turners e Score Following

Foram encontrados autores que trabalharam com Page Turners [PTurn1][PTurn2], que visavam trocar as páginas de uma representação simbólica da performance enquanto ela está sendo tocada ao vivo. A diferença é que ao invés de receber os acordes como um dos inputs, eles utilizam sons criados a partir da representação MIDI da música.

O nome do problema tratado pelos Page Turners encontrados é o Score Following. Existe um extenso estudo sobre a área. Todo ano, autores enviam várias soluções ao MIREX [MScr17] tentando obter melhores soluções. As soluções para este problema podem ter várias aplicações. Alguns exemplos de aplicação são sistemas de troca automática de páginas, adicionar instrumentos para acompanhar um solista automaticamente e controle automático de efeitos sincronizados com a música.

Apesar do extenso estudo sobre o Score Following, ele depende de uma representação MIDI da música para ser resolvido. Não é fácil obter a representação MIDI de músicas. Já acordes, encontrados em cifras, são amplamente disponíveis em websites a exemplo do CifraClub [CifClub]. Ao resolver o Chord Following, será possível fazer o acompanhamento de uma performance em tempo real apenas com os acordes de uma música.

Alinhamento de Acordes e Audio Chord Estimation

Existe uma Tese de Graduação anterior [dVSRA16] que trabalhou com o alinhamento da lista de acordes de uma música a uma performance da mesma. Ele é capaz de, dada a lista de acordes da música e a performance completa, inferir em que momento eles começaram a ser tocados.

Esta Tese utiliza técnicas adquiridas de soluções para o Audio Chord Estimation (ACE). O Alinhamento é, na verdade, uma simplificação do problema. O ACE em sua forma pura recebe apenas o áudio como input, enquanto o alinhamento recebe, além do áudio, uma lista de acordes. Este problema também é um problema extensamente estudado. Assim como a do Score Following, sua página do MIREX também recebe várias submissões anualmente [MACE17].

Em relação ao Chord Following, as soluções tanto do Alinhamento, quanto do ACE não podem ser utilizadas diretamente. Ambos recebem o áudio todo como entrada. No problema deste trabalho, o áudio será recebido em tempo real, fazendo com que várias técnicas do ACE não possam ser utilizadas

Para o Chord Following, será utilizada a simplificação do ACE, o Alinhamento, assim como na Tese anterior [dVSRA16].

Estado da arte

Dado que existem várias semelhanças entre o Chord Following e os problemas de Audio Chord Estimation (ACE) e Score Following, foi escolhido tratar o problema como uma mescla dos dois.

Nesta seção serão analisados os métodos mais promissores para resolução de ambas as tarefas. Como todo problema, existem outras maneiras de se resolver, mas o foco desta seção será só naquelas que se provaram mais eficazes.

Score Following

O desafio de Score Following no MIREX é o de alinhar áudio, recebido em tempo real, a uma representação MIDI dada.

Atualmente é possível resolver o Score Following com precisão de acima de 90% e latência média de 0ms [MScr17]. Um resultado impressionante. Para tal, os autores obtêm a aproximação de como as notas midi iriam soar, quando tocadas em conjunto e alinham esta aproximação com o áudio tocado em tempo real.

Para realizar o alinhamento é utilizada uma variação em tempo real do Dynamic Time Warping [OnDTW1] com utilização de Beta Divergence para calcular a dissimilaridade entre as performances. Para obter a aproximação de como as notas iriam soar eles utilizam métodos de fatorização espectral baseados em Fatorização de Matrizes Não-Negativas [ScrSA1].

Automatic Chord Estimation (ACE)

A melhor acurácia dentre os estimadores pertence ao Harmony Progression Analyser, criado por Ni, Rodriguez e Bie, como submissão para o MIREX 2011 [MACE11] [HPA1]. Com sua abordagem HP_Chan eles conseguem uma taxa de sobreposição de acima de 97% com os acordes reais da música. Um resultado excelente.

Para atingir tal resultado eles utilizam em conjunto técnicas de estimação de Tonalidade, Acordes e Batidas do áudio, mostrando que os vários desafios estão interligados e melhorar soluções para desafios em uma área ajuda na solução de outras. Detecção de Tonalidade e de Batidas, por serem em si problemas complexos, não serão abordados no trabalho atual.

Do ponto de vista de Estimação de Acordes, eles utilizam Chroma e HMM para efetuar a tarefa. Como feature eles obtêm dois Chromagramas, um para graves (de 55Hz até 207.65Hz) que visa estimar batidas e um para agudos (de 220Hz até 1661.2Hz). Para realizar o alinhamento eles utilizam HMM (para modelar os estados) e Viterbi para encontrar o melhor alinhamento.

Decisões de Projeto

Está sendo trabalhado com um problema um pouco diferente dos já conhecidos. Não é possível apenas seguir uma abordagem já publicada e esperar uma acurácia similar. Foi escolhido abordar o problema com o foco em testes, visando entender quão bem algumas das abordagens de outros problemas se adaptam ao problema proposto.

Nas melhores soluções para o Score Following, foi utilizado o Dynamic Time Warping (DTW) para inferir qual o melhor alinhamento. Existem, portanto, motivos para acreditar que o DTW é o melhor caminho para resolver o problema proposto.

Apesar de aparentar ser promissora, a utilização do Dynamic Time Warping para o alinhamento de acordes pode ser problemática. Normalmente no Score Following o DTW é utilizado para efetuar o alinhamento de duas sequências relativamente similares no tamanho. Nessas abordagens, o áudio que está sendo tocado pode ter o tempo modificado um pouco (normalmente se utiliza até 3x de diferença no tempo como margem [OnDTW1]). Quando se utilizam acordes para alinhamento, entretanto, a diferença de tamanho entre as sequências tende a ser muito maior.

Como exemplo dessa discrepância tem-se a Música Misery dos Beatles. A quantidade de janelas obtidas da análise do Spotify da música é de 364 enquanto existem apenas 51 acordes. São aproximadamente 8 vezes mais janelas do que acordes, uma margem já superior à margem de 3 vezes dada pelos autores de score following. É imprescindível que sejam efetuados testes da acurácia do online DTW nestes casos.

Foi escolhido trabalhar inicialmente com análises de áudio recebidos do Spotify [SpoAD] ao invés de processar o áudio real. Nestas análises, uma das features disponíveis é o Chroma. Tal decisão foi feita por permitir fazer testes mais ágeis.

Foram criados scripts para obter a análise de diversas músicas, tornando possível a execução de testes de eficácia com vários dados sem muito esforço. Se fosse utilizado o áudio real, seria necessário procurar na internet exemplos de cada música, coisa que é dificilmente factível por script. Além da simplicidade de obter vários exemplos do Spotify, o processamento de features localmente também seria um ponto de incerteza em relação a implementações anteriores. Poderiam ser introduzidos erros na implementação e isso teria que ser testado.

Em suma, dado que existem dúvidas quanto ao coração do algoritmo (O Dynamic Time Warping) foi escolhido adiar o processamento de features, para obter mais agilidade nos testes iniciais.

Testes com DTW Convencional

Inicialmente, usando como referência trabalhos já feitos para alinhamento de acordes, será testada a eficácia do Dynamic Time Warping padrão (offline).

Este teste é essencial por vários motivos. Tem que ser descoberta a melhor acurácia possível de um DTW efetuando a tarefa, para que se saiba o quanto um erro é causado pela abordagem Online ou por fatores externos ao algoritmo. Além disso, se o DTW offline não tiver uma acurácia satisfatória, uma versão online dificilmente será superior.

Será usado como parâmetro de utilidade da abordagem proposta o Trabalho de Graduação de Araújo, 2016 [dVSRA16]. O objetivo é descobrir se o DTW é capaz de obter uma eficácia similar.

Dynamic Time Warping

Este trabalho não irá explicar em detalhes o funcionamento do Dynamic Time Warping, dado que já existem fontes mais completas a disposição [DTW1]. Em poucas palavras, o DTW vai encontrar o alinhamento de menor custo entre duas séries temporais.

Uma das séries temporais será formada por uma lista de Chromas teóricos de cada acorde. A outra será formada por uma lista de Chromas dos segmentos do Spotify (representando o áudio real). Os acordes da música serão representados pelas colunas da matriz (J). O áudio será representado pelas linhas (T).

Para encontrar o alinhamento com o menor custo, é necessário que seja definida uma métrica de custo para o alinhamento de elementos. Podem ser utilizados Distância Manhattan, Distância Euclidiana, Divergência Beta, entre outros. Nesta seção foi utilizada a Distância Euclidiana entre os Chromas. Chromas muito próximos aos tons teóricos de um acorde tem custo menor.

Dados

Inicialmente foi obtido o Ground Truth [BtlDt], marcações de acordes feitas por especialistas para várias músicas dos Beatles. A partir dos nomes das músicas foram obtidas suas análises do Spotify. A partir do Ground Truth foi obtida a lista de acordes com tons. Os tons de cada acorde foram computados pela biblioteca em python de Zámečník [ChLb].

Depois de cada uma dessas etapas, algumas faixas foram eliminadas do dataset final. Algumas músicas não foram encontradas no Spotify, outras resultaram em erro quando tentou-se obter os tons dos acordes. Ao final, sobraram 168 músicas.

Input

Serão usados os mesmos parâmetros de entrada nos dois algoritmos. Uma das entradas é uma lista de acordes retirada do Ground Truth e seus tons em ordem. Os tons são os índices do Chroma que teriam energia máxima num cenário teórico deste acorde. A outra é a análise de uma música recebida do Spotify [SpoAD].

Modificações na Abordagem de Araújo

Originalmente, a abordagem de Araújo recebia como input o id de uma música do spotify, obtia sua análise e procurava seus acordes no CifraClub [CifClub]. Ela era restrita a um subconjunto dos acordes, sendo incompatível com várias das músicas do dataset dos Beatles.

A abordagem de Araújo foi estendida para ser capaz de receber como input um conjunto de arquivos de texto. Cada iteração recebe uma lista de acordes com os respectivos tons e um arquivo de análise do Spotify.

Output

Ambos os algoritmos devem criar arquivos de texto para cada input. O formato final deve ser “{Tempo inicial} {Tempo final} {Acorde}”. Tal formato está de acordo com as anotações dos acordes no Dataset dos Beatles [BtlDt], que foi utilizado como Ground Truth.

O algoritmo de Araújo tem o output no formato “{Acorde} {Tempo Inicial} {Duração}”, que foi modificado facilmente para o formato final. Apenas foi necessário somar o tempo inicial e a duração para obter o tempo final.

Metodologia de comparação

Para testar a acurácia dos resultados dos algoritmos, foram utilizadas algumas das métricas utilizadas nos testes de Automatic Chord Estimation do MIREX. Os testes de ACE se preocupam não só com a precisão dos segmentos, mas também, com o tipo de acordes retornados. O teste completo poderia criar problemas para as implementações, visto que se acordes próximos tivessem o mesmo template de Chroma, dificilmente o algoritmo iria encontrar a transição correta. Nos testes deste trabalho foi utilizado apenas o Chroma dos acordes, ignorando detalhes como Root e Bass. Dois acordes diferentes com o mesmo Chroma são tratados pelos testes como apenas um elemento.

O framework utilizado para os testes, foi utilizado anteriormente para testar as submissões ao MIREX em 2013 [Eval13]. Ele está disponível no Github de um dos autores [PoMus]. Para obter os resultados esperados, o parâmetro “--chords ChromaFMeasure” foi escolhido.

Resultados

Ao fim dos testes ficou provado que o DTW é capaz de realizar o alinhamento de acordes de forma eficaz com os parâmetros escolhidos. Foram comparados os resultados obtidos ao rodar o DTW e o HMM implementado por Araújo sob o dataset. O de Araújo obteve uma média de 68% enquanto o DTW obteve uma média de 78.9%. Um ganho significativo de mais de 10% em eficácia.

Testes com Online DTW Convencional

Já disponíveis os resultados do alinhamento realizado com o DTW Offline, sabe-se que ele é capaz de realizar o alinhamento de acordes com uma faixa de áudio. Nesta etapa serão testadas diversas abordagens de uso do DTW para alinhamento em tempo real e seus resultados serão comparados com a versão Offline.

O Online DTW

A abordagem seguida foi a de Dixon, 2005 [OnDTW1]. Novamente este trabalho não explicará em detalhes o funcionamento dado que ele segue o que foi feito nesta referência. Alguns pontos, essenciais ao entendimento, serão discutidos.

A Janela de Alinhamento

Na versão Online do DTW, o algoritmo irá manter uma janela de tamanho C e irá realizar o alinhamento apenas entre as diagonais desta janela. A diagonal superior direita é o começo e a inferior esquerda é o fim. Esta janela tem como objetivo fazer com que o algoritmo não cresça exponencialmente de complexidade com a adição de novos elementos. Ela permite que o algoritmo seja executado em tempo real.

A janela se move em direção a diagonal superior direita global de acordo com uma heurística. O algoritmo tenta descobrir em que direção se encontra o melhor caminho, então ele move a janela nesta direção, eliminando elementos com menos chance de participar da solução.

Se o valor de C for pequeno, cada movimento do algoritmo terá uma chance maior de mover a janela num mal caminho, tornando a solução pior. Se o C for grande, o algoritmo tem mais coisas para computar e requer um tempo de bufferização maior. O tempo de bufferização existe pois o algoritmo só passa a alinhar as sequências quando a janela inicial está cheia.

Métodos de Alinhamento

Quando se utiliza um DTW Online, decisões tem que ser feitas em relação a quando o algoritmo irá retornar algum resultado de alinhamento. Existe mais de uma forma de retornar resultados, a decisão se resume a esperar mais para ter uma confiança maior ou receber um resultado impreciso rapidamente.

Método Arriscado

O Método Arriscado recebe o melhor elemento nas fronteiras da janela de alinhamento. Ele é o mais veloz possível e oferece robustez mínima. Neste trabalho não se utilizou este método, mas sua velocidade pode ser útil em um sistema onde o sistema está funcionando muito bem e o objetivo seja adquirir velocidade.

Método Conservador

No método conservador os resultados são retornados dentro da janela de alinhamento. Ele funciona da mesma forma que o DTW Offline, porém apenas dentro da

janela de alinhamento. Em cada iteração é calculado o melhor caminho de alinhamento entre o elemento (T,J) e o elemento (T-C, J-C).

É importante mencionar que em cada iteração o caminho percorrido pode ser diferente. No trabalho atual foram usados métodos de suavização que apenas retornam algum resultado após ele pertencer ao caminho ideal em S iterações. Quanto maior o S, mais tempo o algoritmo leva para retornar algo e mais certeza ele tem deste retorno.

Esta abordagem retorna os elementos com maior acurácia em relação à realidade. É a mais complexa e também a mais adaptável, dado que a filtragem de resultados errados pode ser efetuada de várias formas.

Método Seguro

O método seguro só retorna resultados quando eles são eliminados da janela de alinhamento. Ele é o mais lento. Só retorna resultados depois que a janela de alinhamento não os está levando mais em consideração.

Este método não foi encontrado na literatura, mas se mostrou útil para um sistema de troca de páginas. Ele retorna o último momento em que um elemento seria alinhado pelos outros métodos. Se ele gerar erros por retornar algo antes de ser tocado o erro é no modo que a janela está se movendo e não no método de alinhamento.

Metodologia de Testes

Inicialmente as abordagens foram testadas apenas na música Misery dos Beatles. O objetivo foi tentar encontrar parâmetros que funcionassem bem nessa música para, só então, expandir os testes para todo o Dataset. Se o algoritmo não funcionar bem para uma música, com certeza não funcionará para todas.

Um dos objetivos deste trabalho é criar um sistema capaz de passar as páginas de uma Cifra em tempo real. Para testar esta capacidade, um novo tipo de teste foi necessário. Nos testes offline, o único teste utilizado era o de alinhamento, que testa a qualidade do melhor caminho retornado pelo algoritmo. Ele compara o quão próxima a segmentação de acordes efetuada pelo DTW se assemelha à segmentação real.

Em tempo real, dado que existe um delay entre o acorde tocado e o alinhamento, mesmo que o alinhamento não seja perfeito, é possível passar as páginas de uma cifra caso o algoritmo apenas retorne os acordes depois de que eles tenham sido tocados. O novo teste verifica se os acordes são retornados depois de serem tocados no Ground Truth e mede quanto tempo se passou entre o acorde ser tocado e ele ser reconhecido.

O pré requisito de se utilizar uma solução para passar páginas de uma cifra é que ela nunca retorne que um acorde foi tocado, antes de ele ser tocado na realidade. Dado que existam várias soluções capazes de efetuar a tarefa, o algoritmo mais eficiente será o que tem o menor delay entre o acorde ser tocado e ele ser reconhecido. Se for possível reconhecer acordes com um delay irreconhecível, de menos de 200 milissegundos, o sistema será excelente.

Abordagem com Parâmetros Padrão

Na abordagem padrão da literatura, os autores utilizam $C = 500$ e $MaxRunCount = 3$. O $C = 500$ significa uma janela que precisa receber 500 janelas de áudio antes de se tornar online. É o buffer inicial. O $MaxRunCount$ serve para certificar que o algoritmo não vai

expandir a janela mais de 3 vezes em uma única direção. Ele mantém o algoritmo mais ou menos em direção à diagonal.

Nos dados utilizados, não existe nenhuma música com 500 acordes, portanto, o valor de C teve que ser diminuído. Tal diminuição implica em que os resultados do alinhamento serão menos confiáveis. Haverão menos elementos dentro do buffer para efetuar a comparação.

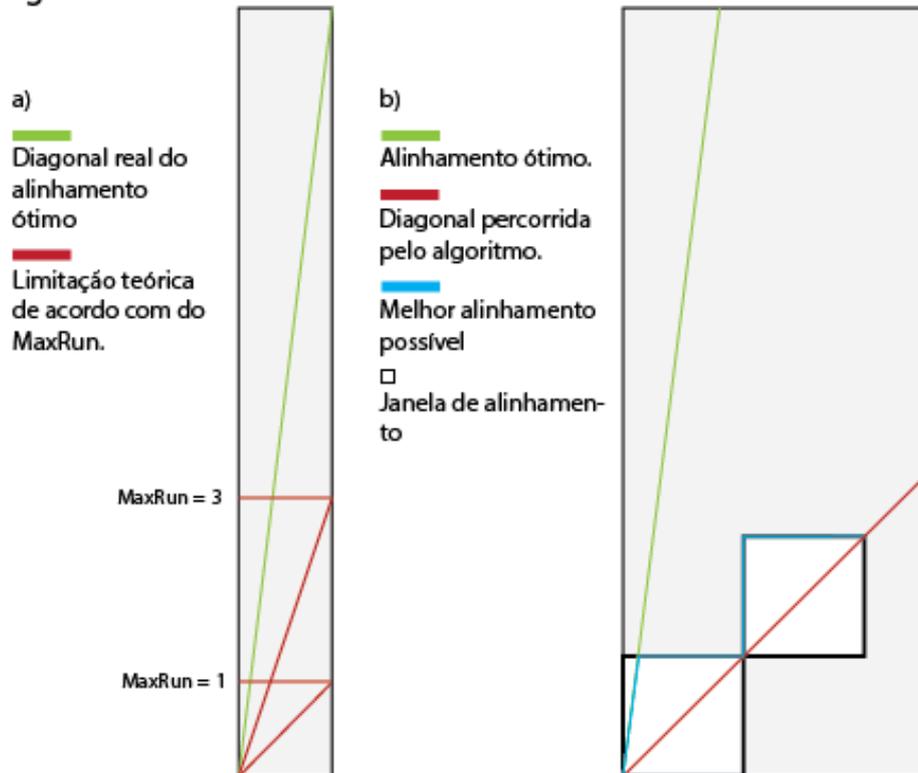
Inicialmente, foi usado $C = 5$ e $\text{maxRunCount} = 3$. Os resultados, tanto do teste de alinhamento quanto o de troca de páginas tiveram resultados muito fracos. Foi analisado o funcionamento do Online DTW com parâmetros padrão e foram **identificadas limitações na sua utilização para o alinhamento de séries temporais de tamanhos muito diferentes**. Existe uma limitação quanto à troca de páginas e uma quanto ao alinhamento da séries: ambos atrelados ao caminho percorrido pelo algoritmo online.

Quanto à limitação da utilização para troca de páginas, ela acontece devido à discrepância do tamanho das séries temporais. Será usada como exemplo a música de testes, que como dito no início deste texto, tem 364 segmentos, que representam janelas do áudio tocado, e 44 acordes. É uma proporção de aproximadamente 8 para 1.

O Online DTW Convencional, ao utilizar **MaxRunCount = 3**, **impede que o algoritmo siga expandindo em direção ao tempo mais de 3 vezes de uma vez**. Ele obriga o algoritmo a terminar sua janela de alinhamento em um máximo teórico de 132 ($3 \cdot 44$). O comportamento do algoritmo é mostrado na Figura 1a.

A segunda limitação, quanto ao alinhamento, se dá devido ao comportamento do algoritmo quando a primeira limitação acontece. O Online DTW, visando não se tornar mais complexo ao receber mais janelas de áudio, gera o alinhamento na janela contida entre os índices (T, J) e $(T-C, J-C)$. T é a linha atual e J a coluna atual. Devido a isto, quanto mais longe T e J , estejam do alinhamento ótimo, pior é o alinhamento. O comportamento é exemplificado na Figura 1b.

Figura 1



Abordagens com Mudança de Parâmetros

Os **parâmetros padrão não foram satisfatórios**, então foram testados os resultados do algoritmo com diversos valores para os parâmetros. Para simplificar os testes, escolheu-se testar, inicialmente, a capacidade do algoritmo de não alinhar todos os acordes antes de serem recebidos todos os segmentos do áudio. Os resultados deste teste indicam o melhor caminho a ser seguido, pois, como foi demonstrado na seção anterior, a qualidade de alinhamento nunca será boa se o algoritmo não for capaz de manter a janela de alinhamento no caminho do alinhamento ótimo. O caminho do alinhamento ótimo, recebido da versão offline do DTW sempre termina na diagonal superior direita, sendo na música Misery $T=363$ e $J=43$.

A métrica escolhida para medir a capacidade para tal foi o número do segmento em que o algoritmo alinha o último acorde. Tal métrica mostra em que momento a janela de alinhamento “colide” com o fim das colunas da tabela. Caso o caminho esteja correto, esta colisão deve ocorrer no fim dos segmentos, que na música de exemplo seria próximo ao índice 364.

Foi feito um loop que modifica os valores de $MAXRunCount$ e C . O range de valores do $MAXRunCount$ foi de $\{\text{Número de Segmentos} / \text{Número de Acordes}\}$ até 30. O valor de início foi escolhido, pois, de acordo com a limitação teórica, exemplificada na seção anterior, seria impossível obter um resultado ótimo com valores menores do parâmetro. O valor final foi escolhido, pois é grande o suficiente para analisar o comportamento dos resultados de acordo com seu crescimento. Já para o C , o range foi de 2 até $\{\text{Número de Acordes}\}$. O valor mínimo foi escolhido pois quando C é menor que 2 o algoritmo é obrigado a seguir a diagonal, dado que não existe escolha a ser feita. Já o valor final, um valor de C não pode ser maior que o número de acordes, pois não existiriam mais elementos na coluna.

Foram efetuados os testes para três músicas, Misery, Taxman e Come Together: todas dos Beatles. Caso fosse testada apenas uma música, os resultados obtidos poderiam ser apenas sorte, ou um tipo de otimização específica para a música. Comportamentos interessantes foram observados.

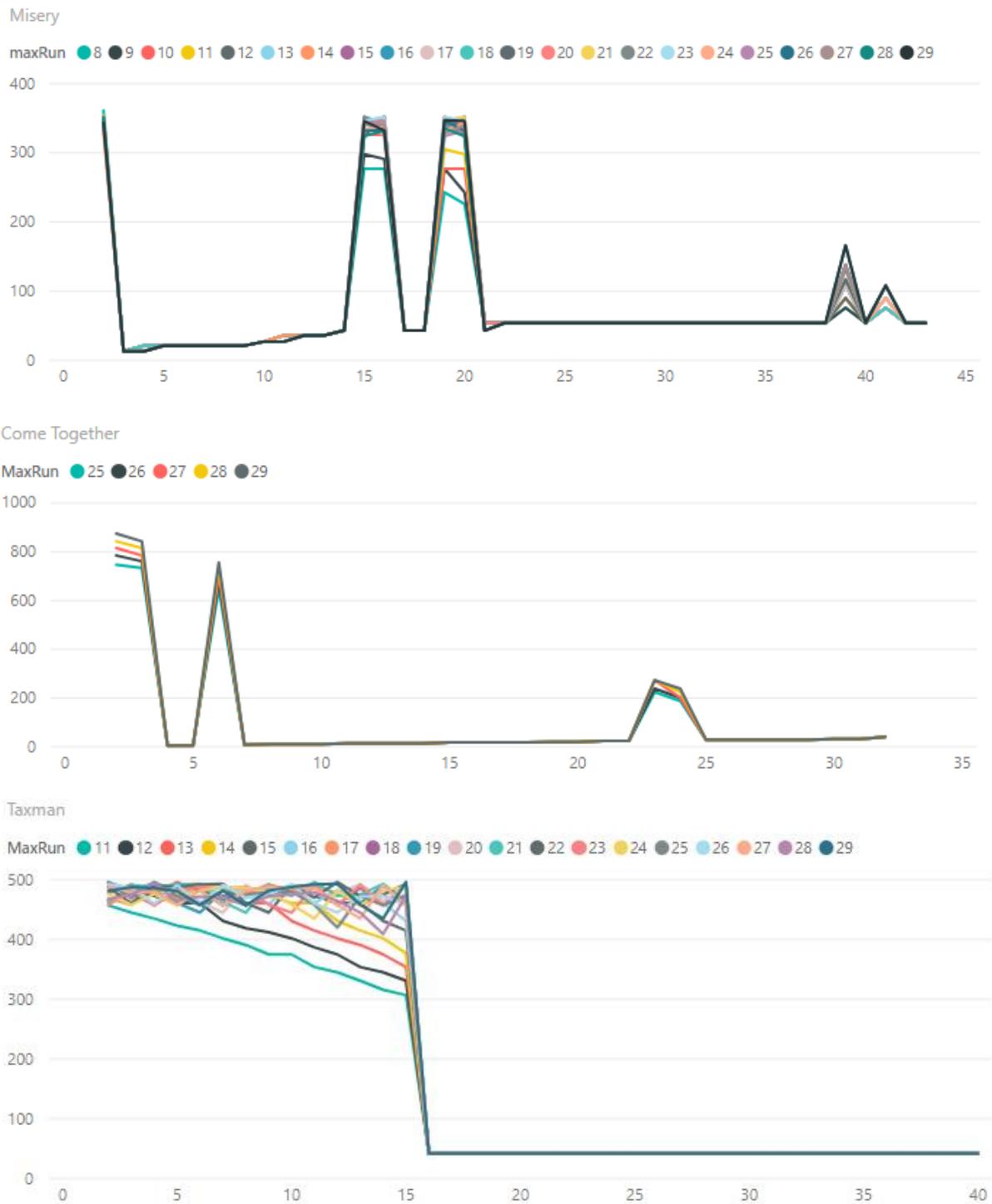
Foi percebido que o MaxRunCount, não pode ser muito alto, pois ele se comporta como uma faca de dois gumes. Ao mesmo tempo que ele deixa o algoritmo se expandir mais em direção ao tempo, ele também pode causar uma expansão mais rápida ainda em direção aos acordes, resultando em colisões com as colunas que mudam drasticamente de um valor de C para o próximo (**Figura 3**).

Foi encontrado um caso peculiar. Quando $C = 2$. O alinhamento sempre terminou próximo do lugar esperado. Em nenhuma das execuções com esta configuração o valor se expandiu rápido em direção aos acordes.

Em execuções onde $C \neq 2$, os melhores resultados vem em momentos sem muita lógica. Os gráficos da Figura 3 mostram alguns destes resultados. É possível que exista alguma relação, mas ela não é visível entre apenas três músicas. Dada a tendência das relações de ficarem mais fracas à medida que mais músicas são adicionadas, o autor deste trabalho escolheu não se estender muito.

Figura 3

Gráficos de resultados obtidos com diferentes valores de MaxRunCount e C. MaxRunCount é representado pela cor das linhas, C pelo eixo x e o último acorde alinhado pelo eixo y.



Testes com $C = 2$ e MaxRunCount influenciado

Sabe-se, devido aos testes da seção anterior, que existem valores de C e MaxRunCount capazes de alinhar o último acorde em uma posição aceitável. Estes parâmetros são $C=2$ e MaxRunCount maior ou igual que a razão entre o Número de Segmentos e o Número de Acordes. Nesta seção será testado se é possível criar um sistema de troca de páginas de cifras para a música Misery usando tais parâmetros.

É importante lembrar que os valores de MaxRunCount utilizados nesta abordagem não estariam disponíveis em uma situação real de acompanhamento. Ainda assim, dado que existe a possibilidade de inferir um valor aproximado de MaxRunCount em uma situação real, foi decidido testar a eficácia do algoritmo com estes parâmetros. As execuções do algoritmo nesta seção passam, então, a requerer um parâmetro de entrada a mais: o Número de Segmentos do áudio, que seria desconhecido em uma situação real.

Foram utilizados $C = 2$ e MaxRunCount = 8, 9 e 10. O alinhamento foi realizado da forma mais segura em relação ao delay. Só foram retornados acordes após os mesmos saírem da janela de alinhamento do Online DTW (método de alinhamento seguro). Visto que a janela só tem 2 elementos esta escolha não impacta muito nos resultados. Os resultados dos testes são demonstrados na Figura 4.

Figura 4

EndGap - tempo entre o último acorde retornado e o fim da sequência de áudio

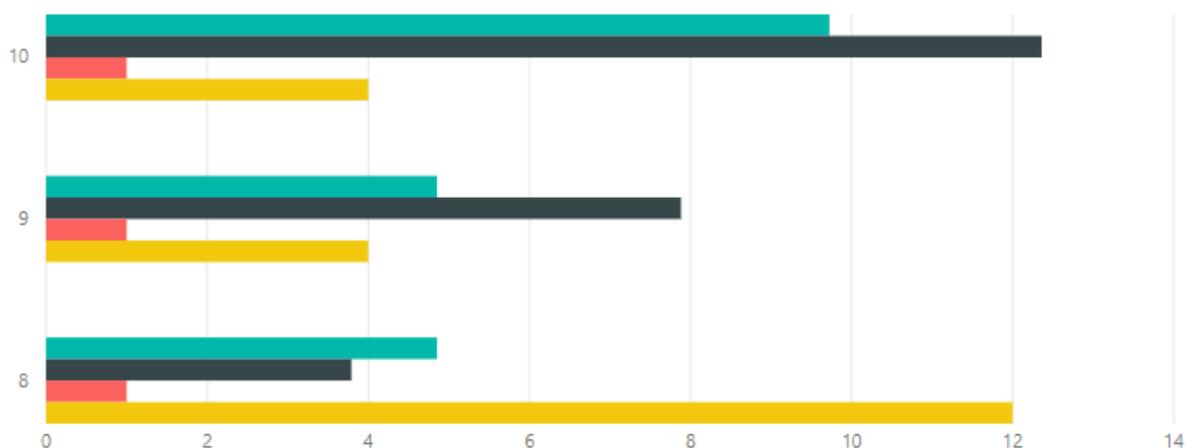
Avg Delay - tempo médio entre acorde ser tocado e ser reconhecido

MissCount - número de acordes não retornados pelo algoritmo

PreAlignCount - número de acordes retornados antes de serem tocados na realidade

Resultados de testes de Page Turning

● EndGap ● AvgDelay ● MissCount ● PreAlignCount



Fica evidente que nenhuma das configurações é utilizável para um sistema de troca de páginas. Todas elas alinham alguns acordes antes de serem tocados (PreAlignCount).

O PreAlignCount diminui com valores mais altos de MaxRunCount. Um caminho que, a primeira vista, poderia ser seguido é usar um MaxRunCount mais alto para obter menos erros em troca de um AvgDelay mais alto. É uma estratégia interessante para obter melhores resultados nos testes, mas que não seria útil em um aplicação real.

O comportamento mais relevante inferido dos testes é que o é que **o MaxRunCount está influenciando demais a solução**. Pode ser observado que o MaxRunCount **muda os resultados drasticamente de um valor para o próximo**. O resultado depende muito mais do MaxRunCount do que do resto dos parâmetros.

Em uma situação real o melhor MaxRunCount seria inferido, e diferentemente do comportamento observado, o algoritmo deveria ser robusto a tais mudanças. **A janela com apenas 2 elementos (C=2) não tem poder de inferência o suficiente para seguir uma solução aceitável**. Outras estratégias são necessárias para resolver o problema proposto.

Conclusão e Pesquisa Futura

Foi estudado a fundo o comportamento do Online Dynamic Time Warping Convencional na tarefa de Chord Following. Durante a pesquisa, foram descobertas diversas limitações e, ao mesmo tempo, vários caminhos alternativos.

Nesta seção serão discutidas algumas abordagens alternativas propostas pelo autor deste trabalho que podem ser capazes de resolver o problema de forma mais eficaz. Algumas das abordagens foram testadas informalmente. Prós e Contras, assim como impressões do autor durante os testes serão apresentados.

DTW Online com Expansão apenas através do tempo

Esta abordagem modifica o algoritmo de forma que ele apenas se expande em direção ao tempo. O parâmetro C se torna o número de acordes e a janela de alinhamento apenas se move em direção dos novos segmentos.

Existe apenas um pequeno número de acordes (Ex: 44 na música Misery) e é possível esperar que o algoritmo receba 44 segmentos antes de começar a efetuar o alinhamento.

Esta abordagem não tem a limitação teórica da janela de alinhamento estar fora da solução ótima, pois todas as soluções possíveis estarão incluídas no caminho percorrido. O que foi percebido nos testes informais é que uma solução baseada nesta técnica iria depender muito da filtragem de resultados errados. Tal fato se dá pois os valores de início e fim do alinhamento (os vértices da janela) vão mudar em todas as iterações, gerando vários resultados errados.

DTW Online com ACE Anterior

Seguindo autores de trabalhos anteriores em Score Following, criar sequências sem discrepâncias muito grandes no tamanho. Nesta abordagem, ao invés de utilizar acordes para realizar o alinhamento em tempo real, são utilizados elementos de uma performance anterior da música. É realizado o ACE de forma offline da performance anterior. Assim que um segmento da performance anterior onde o ACE identificou que existiu uma troca de acorde é atingido pelo DTW em tempo real, o sistema de Chord Following alerta a troca de acordes.

Esta abordagem é muito promissora. Ela seria capaz de utilizar soluções muito boas de ACE, a exemplo do Harmony Progression Analyzer e as limitações do Online DTW seriam eliminadas. Uma das modificações seria a necessidade de obter uma performance anterior da música a ser tocada.

DTW Online com C Enviesado

Nesta abordagem, a janela de alinhamento deixa de ser um quadrado e passa a ser um retângulo influenciado pela razão entre a quantidade de elementos entre sequências.

Um dos problemas durante os testes com o DTW Online Convencional foi que, mesmo influenciando o MaxRunCount com o número de segmentos do áudio ele não conseguiu um resultado aceitável. Esta abordagem obriga o algoritmo a se expandir mais em direção ao tempo. A cada movimento em direção aos acordes, o algoritmo faz R

movimentos em direção ao tempo, sendo R a razão entre o número de acordes e o número de segmentos do áudio.

Esta abordagem foi muito promissora nos testes informais. Usando $\text{MaxRunCount} = 2$ e $R = 8$, o sistema foi capaz de efetuar a tarefa de passar páginas da música Misery. Houveram 2 erros no início da música, mas foram causados enquanto a janela de alinhamento estava incompleta e foram ignorados.

Um dos possíveis problemas desta abordagem é que ela é muito truncada em relação aos movimentos da janela. Caso seja utilizado $\text{MaxRunCount} = 3$, por exemplo, o algoritmo gera um resultado muito ruim por fazer apenas algumas modificações erradas.

DTW Online com MaxRunCount Enviesado

Esta abordagem é similar à anterior, a diferença é que ao invés de enviesar a janela de alinhamento (o C), o MaxRunCount é Enviesado. Ao invés de utilizar apenas um valor de MaxRunCount são utilizados dois, MaxRunCount_T e MaxRunCount_J . O MaxRunCount_T (em direção o tempo) é R vezes o MaxRunCount_J (em direção aos acordes). R é a razão entre o número de segmentos e o número de acordes.

Não foram feitos testes com esta abordagem, mas ela pode resolver alguns dos problemas da abordagem anterior. Ao invés de obrigar o algoritmo a se expandir em direção do caminho ótimo, ela faz com que esta expansão seja bem mais provável. O caminho percorrido pela janela pode ser mais fluido. A abordagem anterior obriga o algoritmo a rodar R vezes mais em direção ao tempo em todas as iterações enquanto esta abordagem permite que ele faça uma escolha a cada momento.

DTW Online com MaxRunCount e C Enviesados

Esta abordagem une as duas abordagens anteriores, usando uma janela retangular e dois valores de MaxRunCount . Como benefício, ela tem elasticidade de balancear os prós e contras das duas.

Utilizar esta abordagem inicialmente iria ser arriscado, pois não iria se entender bem os efeitos de cada uma das mudanças no algoritmo. Ela pode gerar um resultado pior do que usar apenas uma das abordagens.

Referências

- [PTurn1] Arzt, A., Widmer, G. and Dixon, S. (2008). Automatic Page Turning for Musicians via Real-Time Machine Listening. [online] [Dl.acm.org](https://dl.acm.org/citation.cfm?id=1567337). Available at: <https://dl.acm.org/citation.cfm?id=1567337> [Accessed 10 Jun. 2018]
- [PTurn2] Arzt, A. (2007). Score Following with Dynamic Time Warping An Automatic Page-Turner. [online] [Cp.jku.at](http://www.cp.jku.at). Available at: http://www.cp.jku.at/research/papers/Arzt_Masterarbeit_2007.pdf [Accessed 10 Jun. 2018].
- [OnDTW1] Dixon, S. (2005). Live Tracking of Musical Performances Using On-line Time Warping. [online] [Semanticscholar.org](https://www.semanticscholar.org). Available at: <https://www.semanticscholar.org/paper/Live-Tracking-of-Musical-Performances-Using-On-line-Dixon/94956fc204b043c61afa739ec669162c24915165?tab=abstract> [Accessed 10 Jun. 2018].
- [MMB14] McVicar, M., Santos-Rodriguez, R., Ni, Y. and Bie, T. (2014). Automatic Chord Estimation from Audio: A Review of the State of the Art. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(2), pp.556-575.
- [MScr17] Music-ir.org. (2017). 2017:Real-time Audio to Score Alignment (a.k.a. Score Following) Results - MIREX Wiki. [online] Available at: [http://www.music-ir.org/mirex/wiki/2017:Real-time_Audio_to_Score_Alignment_\(a.k.a._Score_Following\)_Results](http://www.music-ir.org/mirex/wiki/2017:Real-time_Audio_to_Score_Alignment_(a.k.a._Score_Following)_Results) [Accessed 10 Jun. 2018].
- [MACE17] Music-ir.org. (2018). 2017:Audio Chord Estimation Results - MIREX Wiki. [online] Disponível em: http://www.music-ir.org/mirex/wiki/2017:Audio_Chord_Estimation_Results.
- [MACE11] Nema.lis.illinois.edu. (2012). MIREX 2011: Audio Chord Description. [online] Available at: http://nema.lis.illinois.edu/nema_out/mirex2011/results/ace/ [Accessed 10 Jun. 2018].
- [HPA1] Ni, Y., Mcvicar, M., Santos-Rodriguez, R. and De Bie, T. (2012). HARMONY PROGRESSION ANALYZER FOR MIREX 2011. [online] Music-ir.org. Available at: <http://www.music-ir.org/mirex/abstracts/2011/NMSD2.pdf> [Accessed 10 Jun. 2018].
- [ScrSA1] Alonso, Pedro & Cortina, Raquel & Menéndez-Canal, Jonatan & Rodríguez-Serrano, Francisco & Vera-Candeas, Pedro. (2015). A Real-Time Audio-to-Score Alignment Framework using Spectral Factorization and Online Time Warping.
- [DTW1] Müller, M. (2007). Information Retrieval for Music and Motion. Berlin, Heidelberg: Springer-VerlagBerlinHeidelberg, pp.69-84. Available at: https://www.springer.com/cda/content/document/cda_downloadaddocument/9783540740476-c1.pdf?SGWID=0-0-45-452103-p173751818

[OnDTW1] Dixon, Simon. (2005). Live tracking of musical performances using on-line time warping.

[PoMus] Pauwels, J. (n.d.). MusOOEvaluator. [online] GitHub. Available at: <https://github.com/jpauwels/MusOOEvaluator>.

[Eval13] Pauwels, J. and Peeters, G. (2013). Evaluating automatically estimated chord sequences. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing.

[SpoAD] Analyzer Documentation. [online] Available at: http://docs.echonest.com.s3-website-us-east-1.amazonaws.com/_static/AnalyzeDocumentation.pdf [Accessed 15 May 2018].

[dVSRA16] Pedro Tiago de V. S. R. Araújo. Transcrição e matching automático de uma música e sua cifra. Trabalho de Graduação, CIn - UFPE, 2016.

[LPA17] Arthur Lubambo Peixoto Accioly. Transcrição Automática de Acordes com Fontes Híbridas. Trabalho de Graduação, CIn - UFPE, 2017.

[CifClub] Cifra Club. (2018). [online] Available at: <https://www.cifraclub.com.br/> [Accessed 6 Jun. 2018].

[ChLb] Zámečník, B. chord-labels. [online] GitHub. Available at: <https://github.com/bzamecnik/chord-labels> [Accessed 6 Jun. 2018].

[BtIDt] Isophonics.net. Reference Annotations: The Beatles | Isophonics. [online] Available at: <http://www.isophonics.net/content/reference-annotations-beatles> [Accessed 6 Jun. 2018].