Pedro Rodrigues Nacione Pedruzzi
Ricardo A. Redder Jr.

Reconhecimento e Busca Adaptativos de Padrões Musicais

São Paulo – SP

Dezembro / 2008

Pedro Rodrigues Nacione Pedruzzi
Ricardo A. Redder Jr.

Reconhecimento e Busca Adaptativos de Padrões Musicais

Dissertação apresentada à Comissão de Graduação em Engenharia da Computação da Escola Politécnica da Universidade de São Paulo para a obtenção da graduação no curso de Engenharia da Computação.

Orientador:
Prof. Doutor João José Neto

Graduação em Engenharia de Computação
Departamento de Computação e Sistemas Digitais
Escola Politécnica da Universidade de São Paulo

São Paulo – SP

Dezembro / 2008

"A música escondida não tem valor."

Aulo Gélio

Resumo

Este trabalho propõe um método inétido de comparação de trechos monofônicos de músicas baseado em técnicas adaptativas, especificamente, utilizando um autômato adaptativo. Para avaliação deste método, um protótipo de um sistema de busca musical baseado em conteúdo é proposto, abrangendo a detecção e extração de notas, a comparação inexata de melodias e a busca.

A arquitetura deste protótipo é apresentada, assim como o detalhamento de cada componente, mostrando o fluxo de informação e apresentando os resultados de cada fase do processo. Um foco maior é dado ao componente de comparação adaptativo, porém, o trabalho não deixa de apresentar um protótipo que valide os conceitos apresentados, além de permitir a avaliação do desempenho do processo construído como um todo.

Abstract

This work proposes a novel method to compare monophonic music excerpts, based on adaptive techniques, specifically, using an adptative automaton. For the evaluation of this method, a prototype of a content-based music search engine is presented, comprehending pitch detection and note segmentation from audio samples, inexact melody matching and search.

The architecture of this prototype is presented, as well as the details of its components, showing the data flow and presenting the results of each phase of the process. A major focus is given to the adaptive comparison component, nevertheless, the work still presents a prototype which may validate the proposed concepts, besides that, it enables the evaluation of the process performance as whole.

Sumário

Lista de Figuras

Lista de Tabelas

Introdução p. 10

1 Motivação p. 11

2 Objetivo p. 14

3 Histórico e Resenha Bibliográfica p. 16

3.1 Histórico dos sistemas de busca . . . . . p. 16

3.2 Histórico de técnicas de estimação de frequência . . . . . p. 17

3.3 Histórico de técnicas de reconhecimento de sinais de áudio . . . . . p. 20

4 Conceitos p. 22

4.1 Processamento de sinais . . . . . p. 22

4.1.1 Transformada de Fourier . . . . . p. 22

4.2 Método dos mínimos quadrados . . . . . p. 23

4.3 Adaptatividade . . . . . p. 23

5 Técnicas e procedimentos usados p. 26

5.1 Especificação do sistema . . . . . p. 26

5.1.1 Descrição . . . . . p. 26

5.1.2 Uso do sistema . . . . . p. 26

5.1.3 Requisitos funcionais e Premissas . . . . . p. 27
5.1.3.1 Entrada de dados . . . . . p. 27
5.1.3.2 Tipo de busca . . . . . p. 27
5.1.3.3 Espaço de busca . . . . . p. 28
5.1.3.4 Resposta do sistema . . . . . p. 28
5.1.4 Arquitetura do Sistema . . . . . p. 28
5.1.4.1 Fluxo de informação do sistema . . . . . p. 28
5.1.4.2 Diagrama de componentes . . . . . p. 29
5.1.5 Escopo . . . . . p. 30
5.2 Extração de notas . . . . . p. 31
5.2.1 Aquisição do sinal . . . . . p. 31
5.2.2 Extração do espectrograma . . . . . p. 32
5.2.3 Filtro de intensidade . . . . . p. 32
5.2.4 Extração de picos de frequência . . . . . p. 33
5.2.5 Identificação das notas e Geração do modelo musical . . . . . p. 33
5.2.6 Sintetização de áudio . . . . . p. 34
5.3 Representação de notas e melodias . . . . . p. 35
5.4 Proximidade de melodias . . . . . p. 36
5.5 Comparação numérica . . . . . p. 36
5.6 Busca inexata com autômato adaptativo . . . . . p. 38
5.7 Notas como símbolos . . . . . p. 39
5.7.1 Quantização absoluta das alturas das notas . . . . . p. 41
5.7.2 Construção do autômato adaptativo . . . . . p. 42
5.7.3 Critério para avaliar a semelhança entre melodias . . . . . p. 47
5.7.4 Busca . . . . . p. 47

6 Resultados p. 49

6.1 Rotinas para manipulação de melodias e arquivos MIDI . . . . . p. 49
6.2 Repertório musical e construção do repositório . . . . . p. 50
6.3 Processo de extração de notas . . . . . p. 51
6.4 Comparação numérica . . . . . p. 52
6.5 Quantização das alturas . . . . . p. 52
6.6 Comparação com autômato adaptativo . . . . . p. 54
6.7 Comparação de melodias e busca . . . . . p. 55

7 Análise e crítica p. 58
7.1 Quantização de notas . . . . . p. 58
7.2 Autômato adaptativo . . . . . p. 59

8 Melhorias e Trabalhos futuros p. 60

9 Contribuições p. 62

Referências p. 63

Lista de Figuras

1 Influência da frequência fundamental na taxa de cruzamento pelo zero . p. 18

2 Primeiro exemplo de autômato . . . . . p. 24

3 Configuração do autômato após primeira ação adaptativa . . . . . p. 24

4 Configuração do autômato após segunda ação adaptativa . . . . . p. 25

5 Caso de uso típico do sistema . . . . . p. 27

6 Fluxo de informação do sistema . . . . . p. 29

7 Diagrama de componentes do sistema . . . . . p. 30

8 Sinal de áudio de um assobio . . . . . p. 32

9 Espectrograma do sinal correspondente a um assobio . . . . . p. 33

10 Espectrograma após aplicação do filtro de intensidade . . . . . p. 33

11 Espectrograma após extração dos picos de frequência . . . . . p. 34

12 Alturas quantizadas pelo método relativo . . . . . p. 40

13 Comparação dos dois métodos de quantização de alturas . . . . . p. 42

14 Configuração inicial do autômato adaptativo . . . . . p. 43

15 Configuração do autômato após ação adaptativa 1 . . . . . p. 43

16 Configuração do autômato após ação adaptativa 2 . . . . . p. 44

17 Emparelhamento de durações na comparação nota a nota . . . . . p. 53

18 Quantização absoluta aplicada a uma melodia com perturbações aleatórias p. 53

19 Análise de acerto da busca . . . . . p. 57

Lista de Tabelas

1 Notas extraídas a partir do espectrograma . . . . . p. 34

2 Resultados do teste auditivo . . . . . p. 52

3 Comparação dos métodos de quantização . . . . . p. 54

4 Efetividade do autômato com perturbações . . . . . p. 54

Introdução

Sistemas de busca de texto já se tornaram parte do cotidiano das pessoas por todo o mundo, evoluíram de protótipos desenvolvidos em laboratórios de universidades até tornarem-se enormes e complexos sistemas comerciais. Apesar disso, estes sistemas ainda se apóiam fortemente sobre algoritmos de busca textual.

Com a popularização cada vez maior de conteúdos multimídia, surge o interesse por extender a idéia de sistemas de busca para esta área de informação. Dentre as diversas pesquisas que abordam o tema, uma particularmente interesse é a busca de áudio baseada em conteúdo. O presente trabalho aborda tal tema, com um foco especial em um método de comparação de conteúdos musicais monofônicos, baseado em técnicas adaptativas. O método de comparação desenvolvido termina por servir de base para a construção de um protótipo de um sistema de busca musical, envolvendo captura de áudio, identificação de notas, comparação e busca.

Em suma, o protótipo desenvolvido se propõe a identificar uma música reproduzida por um usuário através de um assobio. Após o detalhamento do protótipo, testes com o sistema são apresentados, assim como seus resultados, a fim de permitir uma avaliação inicial do método desenvolvido.

Por fim, é desenvolvida uma análise do trabalho, apontando limitações, pontos de melhoria, contribuições, extensões e trabalhos futuros. Assim, espera-se contribuir a esta crescente área de pesquisa que é a recuperação de informação em conteúdos multímidia, a partir da utilização de uma técnica adaptativa. Ao mesmo tempo, pretende-se contribuir também ao estudo da adaptatividade, apresentando uma interessante aplicação prática do conceito.

1 Motivação

Os sistemas de busca atuais evoluíram rapidamente desde suas primeiras versões, e hoje se tornaram parte fundamental do cotidiano de grande parte da população. Para muitos profissionais é difícil imaginar um dia de trabalho em que não se utilize algum mecanismo de busca. Pode-se citar como exemplo destes sistemas: Google.com ™, Yahoo.com ™ ou Live Search ™.

Porém tais sistemas de busca em geral baseiam-se sobre os mesmos princípios e métodos de busca, em sua maioria aplicados ao domínio textual. Mesmo alguns sistemas que, por exemplo, efetuam buscas por imagens ou vídeos continuam em sua essência baseando-se nos mesmos princípios, já que se apóiam na categorização textual do conteúdo.

A popularização destes sistemas de busca mostrou as possibilidades de expansão e a importância que estes podem adquirir no cotidiano pessoal e profissional das pessoas. Porém suas limitações de contexto (restrição a textos apenas) imediatamente levantam a necessidade de novas técnicas com o fim de ampliar os domínios de aplicação destes sistemas.

Um dos domínios de extrema importância com relação a tal tema é o domínio de áudio, devido a fatores como sua popularidade no meio virtual, quantidade de conteúdo disponível, facilidade de produção de conteúdo, etc. Músicas em formatos digitais, tais como MP3, MIDI, WAV, etc, podem ser facilmente encontradas na internet e já se tornaram parte do cotidiano de grande parte da população. Além disso, a capacidade de reproduzir tais formatos torna-se cada vez mais um padrão nos aparelhos eletrônicos de áudio. Paralelamente aos conteúdos de áudio digitais, podem-se citar ainda conteúdos de áudio-visual, que recentemente adquiriram grande popularidade e, analogamente aos conteúdos de áudio, estão cada vez mais se tornando parte importante do ambiente virtual.

Uma situação típica e extremamente frequente, é aquela na qual um indivíduo é capaz de reproduzir apenas um trecho da melodia de uma música que deseja encontrar. Tal reprodução está inerentemente sujeita a variações de diversas naturezas com relação

à melodia original. O indivíduo pode reproduzir a melodia com divergências na altura (frequência fundamental de vibração) ou duração das notas musicais, ou pode ainda cometer erros com relação a própria melodia, como por exemplo, esquecendo uma nota ou inserindo uma nota inexistente. Este caso pode manifestar-se a partir de diversas situações do cotidiano, como ao tentar encontrar uma música ouvida em um filme, um programa de televisão ou rádio, etc.

Este tipo de problema é extremamente difícil para um indivíduo comum devido à falta de ferramentas que possam auxiliar tal busca, já que a grande quantidade de músicas existentes torna inviável uma varredura completa dos repositórios existentes, ou seja, ouvir todas as músicas, uma a uma. Além disso, as formas mais comuns de organização de repositórios musicais recaem sobre estilos musicais e nomes, o que não é o suficiente para endereçar, ou ajudar no problema apresentado, já que, em geral, tais informações não são suficientes para encontrar uma música a partir de um trecho de uma de suas melodias.

Atualmente, a forma mais comum de tentar encontrar uma solução para tal problema é utilizando-se a ajuda de um especialista (um vendendor de discos, por exemplo), que em alguns casos, é capaz de identificar o trecho reproduzido. O problema imediato com tal abordagem é o fato da mesma não ser escalável e de difícil utilização, além de enfrentar limitações impostas pela própria natureza do especialista, como dificuldade de lidar com grandes quantidades de músicas, diversidades de estilo, etc.

Assim, apesar da grande relevância de conteúdos multimídia e dos incontáveis esforços de pesquisa na área, diversas questões ainda permacenem em aberto. Poucos são os mecanismos de busca já em estágio maduro capazes de lidar com tais conteúdos. Dentre tais questões em aberto, uma de extrema relevância é a similaridade entre conteúdos de áudio. Onde, apesar das diversas técnicas já utilizadas, dificuldades no estabelecimento do significado da similaridade entre dois conteúdos ainda permanece. Além disso, o estabelecimento de um algoritmo de similaridade eficaz serve como um dos pilares para um bom sistema de busca, já que diversos métodos de busca se apóiam em tal medida.

Sistemas capazes de identificar músicas automaticamente a partir de uma amostra reduzida também teriam uma grande aplicação na detecção de plágios e usos não autorizados. A fiscalização de uso não autorizado de músicas ou trilhas musicais é particularmente difícil, pois após a sua vinculação dificilmente têm-se um rastro de sua utilização, e a identificação de um plágio é bastante subjetiva. Sistemas de identificação automática seriam capazes de fornecer fortes indícios de ambos os casos. Basta imaginar que um

sistema como esse poderia monitorar uma transmissão de rádio ou TV, verificando se há similaridade com um dado conteúdo. Uma alta similaridade poderia dar indício de um uso não autorizado ou um plágio deste conteúdo.

Estes fatos demonstram a demanda por avanços que possam auxiliar tal tipo de busca, ou seja, a necessidade por ferramentas ou algoritmos que permitam o avanço da área de recuperação de informação multimídia. E é justamente neste ponto que este trabalho se encaixa, na tentativa de propor mais uma ferramenta para comparação de conteúdos de áudio, em especial, conteúdos compostos por melodias, ou sequência de notas.

2 Objetivo

Este projeto procura apresentar uma alternativa de solução para uma questão extremamente ampla, que é a busca de conteúdos musicais. Por ser este um tema muito vasto, sua solução geral é muito complexa e ampla, e por isso, para viabilizar esse empreendimento, optou-se por uma redução do escopo original. Assim decidiu-se por privilegiar nesse projeto apenas os pontos que maior relevância apresentam para a resolução desse problema.

Um dos pontos que estabelece a maior barreira para a construção de um sistema como o vislumbrado é a dificuldade de comparação entre dois trechos de melodia. Esta comparação não é bem definida, assim não há uma forma trivial de se estabelecer o conceito de distância entre dois trechos melódicos, conceito que por sua vez é fundamental para a utilização dos modelos de busca e indexação existentes.

Assim, pretende-se com este trabalho promover um avanço sobre tal questão de comparação de conteúdos musicais, e para tanto, serão adotadas técnicas pouco exploradas em tal domínio. O projeto se apoiará sobre dois pilares importantes: a utilização de conceitos do domínio musical e o uso de técnicas adaptativas, ambos aplicados à comparação de conteúdos musicais.

Diversas técnicas têm sido utilizadas para análise e comparação de sinais de áudio, porém, em sua maioria, tais métodos recaem sobre princípios de processamento de sinais, por serem genéricos e possuírem grande aplicabilidade, tendendo a ignorar conceitos específicos do domínio musical, ou seja, conceitos de notas, tempos, etc. Entretanto, tais princípios podem ser de grande relevância quando se deseja efetuar uma busca musical usando como chave de busca um trecho melódico. Assim, com este trabalho, aspira-se a utilização de tais conceitos dentro do contexto de comparação de conteúdos de áudio, com o fim de obter melhores resultados.

Sendo assim, o objetivo primário deste trabalho é apresentar o ensaio de uma proposta de um novo método de reconhecimento de padrões musicais. Este método é baseado em

técnicas adaptativas e foi utilizado como base para a construção de um protótipo de um sistema de busca musical. Como objetivos secundários, serão discutidos em detalhes o projeto completo deste protótipo, incluindo as técnicas e procedimentos utilizados em componentes de software como a extração de notas, a construção da base de dados e a quantização de notas, e os resultados obtidos a partir de uma bateria de testes.

3 Histórico e Resenha Bibliográfica

Nesta seção serão apresentados os históricos dos principais temas relacionados ao projeto desenvolvido, mostrando sucintamente a história que se desenrolou paralelamente dos sistemas de busca e dos avanços dos métodos de manipulação de conteúdos de áudio com o uso de computadores.

3.1 Histórico dos sistemas de busca

Métodos de busca baseados em texto já são antigos e utilizados há um longo tempo. Entre os métodos mais simples, pode-se citar os baseados no uso de índices remissivos. Apesar de simples, este método é extremamente útil e eficiente quando se deseja efetuar a busca de uma palavra em um conjunto de documentos. Além deste método simples, existem ainda outras formas de se indexar um documento e efetuar uma busca no mesmo. Porém a aplicação manual destes métodos sempre apresentou dificuldades, pelas dificuldades de indexação de palavras, lentidão de busca, etc.

Como advento dos computadores, tais métodos passaram a ser automatizados, o que obviamente aumentou sua capacidade e facilidade de uso. Assim nasceram os primeiros sistemas de busca, juntamente com os computadores. Porém estes métodos eram em sua essência muito simplistas, considerando em geral apenas buscas de trechos exatos.

A idéia dos sistemas de busca da forma como conhecemos hoje surgiu algum tempo depois, já por volta da década de 60, e foi se aprimorando ao longo dos anos. Conceitos como modelo de espaço vetorial, frequência inversa no documento (IDF), frequência do termo (TF), discriminação de termos, relevância e feedback começaram a ser galgados nesta época. Tais técnicas evoluíram muito ao longo dos anos, provendo ferramentas extremamente importantes para efetuar indexações e buscas em conjuntos extensos de documentos.

Conforme cresce o tamanho do espaço de busca, maior importância tais técnicas assumem, assim, com o advento da internet, estes métodos adquiriram um papel especial no mundo da computação. Isso porque a internet abriu a possibilidade de se criar espaços de buscas muito maiores do que até então construídos, já que o espaço de busca poderia ser virtualmente todo documento disponível na rede. Por volta da década de 90 começam então a surgir os sistemas web de indexação e busca. O primeiro sistema deste tipo foi o *Archie*, porém efetuava buscas apenas dos nomes de arquivos, e não de seus conteúdos. Pouco tempo depois surgiram os primeiros *crawlers*, componente dos sistemas de busca que se tornou indispensável aos sistemas atuais.

Desde seu início até o presente a tecnologia de Recuperação de Informação (*Information Retrieval*) evolui muito, e diversos métodos de busca, além de variações, foram criados ao longo destes anos, e hoje se podem citar dois modelos que assumiram importância fundamental nesta área: o modelo espaço vetorial e o modelo probabilístico.

Tais modelos e técnicas dão o tom do estudo desta tecnologia no mundo acadêmico, porém apenas tais conceitos não são o suficiente para se construir um sistema de busca similar aos que encontramos atualmente. Além destes princípios, considerações diversas relacionadas a desempenho, propriedade intelectual, conteúdo impróprio, conteúdo falso, tentativas de manipulação de resultados, etc. devem ser levadas em conta. Hoje, uma das tendências mais fortes de desenvolvimento desta área é a especialização dos sistemas de buscas, levando em conta, por exemplo, aspectos semânticos do tema que constitui o espaço de busca. Relacionado a isso, há também um grande interesse em expandir as fronteiras da tecnologia de recuperação de informação para outros tipos de conteúdo, como conteúdos de áudio e vídeo. Recentemente, a TREC - Text REtrieval Conference, uma das maiores conferências sobre recuperação de texto, incorporou o tema de busca sobre áudio como uma sub-tarefa.

3.2 Histórico de técnicas de estimação de frequência

O problema de estimar a frequência de trecho de áudio é um problema estudado há um longo tempo. Diversas técnicas e métodos já foram desenvolvidos sobre o tema, porém até o presente momento estas técnicas têm apresentado dificuldades para atingir o nível desejado de qualidade. Frente a um sinal único claro, diversas técnicas apresentam um bom desempenho, porém quando testadas com sinais ruidosos, ou contendo mais de uma linha melódica estas técnicas tendem a falhar. Diferentes conceitos podem ser aplicados

na tentativa de estimar a frequência de um trecho de áudio. Entre as principais técnicas destacam-se: métodos que se baseiam no domínio do tempo, métodos que utilizam o domínio da frequência e métodos estatísticos.

Métodos baseados na análise do domínio do tempo se valem do fato que os sinais são periódicos, o que faz alguns eventos também serem periódicos, e, portanto, podem ser contados.

Taxa de cruzamento pelo zero (ZCR - *Zero-crossing rate*). A idéia deste método consiste em contar o número de vezes que o sinal de áudio cruza o eixo dos tempos, imaginando-se que a principal componente de frequência responsável por este cruzamento será a frequência fundamental. A Figura 1 exemplifica o fato, onde uma componente de frequência mais alta não exerce grande influência sobre o número de cruzamentos do sinal com o eixo dos tempos.

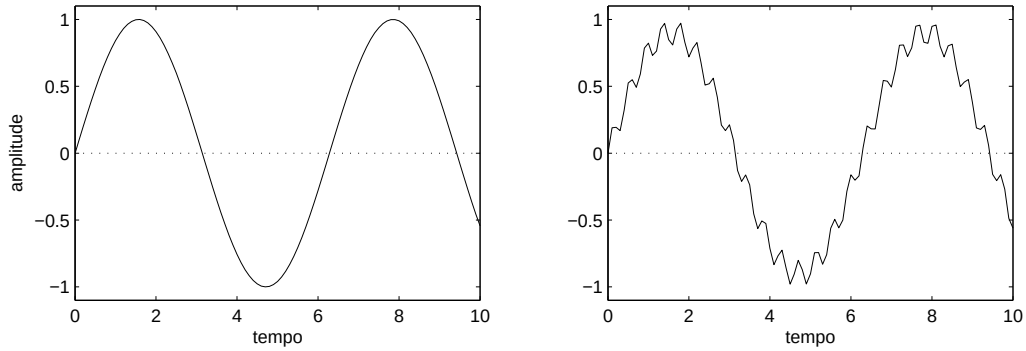

Figura 1: Influência da frequência fundamental na taxa de cruzamento pelo zero

Taxa de picos. Este método consiste em contar o número de picos por segundo em um sinal, sabendo que através do número de picos é possível inferir a frequência do sinal, tem-se então a estimativa da frequência. Analogamente ao ZCR, a frequência fundamental será a componente de frequência que mais contribuirá para a ocorrência de picos no sinal, assim é possível dizer que a estimativa obtida corresponde à estimativa da frequência fundamental.

Taxa de eventos de inclinação. Devido ao fato de o sinal ser periódico a inclinação do sinal também irá variar periodicamente, assim observar picos e zeros da inclinação do sinal pode ser mais informativo do que observar picos e zeros do sinal original.

Correlação. Existem ainda métodos que se baseiam na correlação entre duas amostras de áudio, definindo assim a similaridade entre os dois sinais. Formas de onda similares apresentariam uma correlação alta, enquanto formas de onda muito diferentes teriam uma baixa correlação.

Além de métodos baseados no domínio do tempo, existem também diversas técnicas baseadas no domínio da frequência. Estas, por sua vez, recaem sobre o fato de que o sinal pode ser modelado como uma soma de séries harmônicas, guardando um alto grau de informação sobre a frequência fundamental.

Proporção de componentes de frequência. Em 1979, Martin Piszczalski (PISZCZALSKI, 1986) (Piszczalski; Galler, 1979) trabalhava em um sistema capaz de transcrever músicas automaticamente, assim, necessariamente um dos componentes deste sistema era o componente de extração de notas. O procedimento adotado se valia do cálculo do espectro do sinal, da detecção de picos deste espectro, e de uma análise probabilística destes picos.

Métodos baseados em filtros. Estes métodos utilizam a idéia de aplicar diferentes filtros ao sinal, e analisar sua saída. Por exemplo, caso um sinal possua uma saída alta após a aplicação de um filtro passa-faixa, pode-se afirmar que este sinal possui entre suas componentes a frequência do filtro. Em 1977, James A. Moorer (MOORER, 1977) propôs um algoritmo denominado *Filtro Comb Ótimo*, baseado nestes conceitos. Uma tentativa mais recente foi proposta por John E. Lane (LANE, 1990), denominada *Filtro IIR Ajustável*. Existem ainda diversas técnicas que se apóiam sobre a *análise cepstrum*, que corresponde ao resultado da transformada de Fourier do logarítmo do espectro de magnitude do sinal de entrada.

Diversos outros métodos sobre o domínio da frequência também foram desenvolvidos, dentre estes deve-se destacar duas abordagens importantes: Redes Neurais, utilizando conceitos de I.A. e estimadores de Máxima Verossimilhança, utilizando modelos probabilísticos.

3.3 Histórico de técnicas de reconhecimento de sinais de áudio

Há uma diversidade de métodos que podem ser úteis na tentativa de se extrair algum tipo de informação de um conteúdo de áudio. De uma maneira simplificada, tais métodos podem ser divididos entre os que assume algum tipo de conteúdo relacionado à fala, e os que não assumem. Uma diversidade de métodos é direcionado a reconhecer automaticamente a fala, traduzindo um discurso para texto, por exemplo. Este trabalho porém está mais próximo de métodos que abordam o áudio de uma forma mais genérica, sendo mais adequado para conteúdos relacionados a melodias, por exemplo.

Conteúdos de áudio podem armazenar diversas classes de áudio, como melodias, efeito sonoros, sons de animais, e etc., isso deixa claro que métodos baseados na fala não são suficientes para uma tarefa geral de busca de áudio. Além desta variedade de sons, um grande complicador é que estes diversos tipos de sons muitas vezes estão misturados, e até simultâneos em um conteúdo de áudio. Por exemplo, em uma música, diversos instrumentos contribuem para gerar uma melodia, enquanto pode ainda haver a parte cantada da música.

Um dos problemas mais básicos da análise de áudio é distinguir um conteúdo constituído por fala, de um conteúdo não vocal. Esta tarefa é importante, pelo fato de existirem diferentes técnicas adequadas para cada um dos tipos, assim, aplicar uma técnica baseada na fala sobre um conteúdo constituído por uma melodia não produzirá resultados úteis. John Saunders (SAUNDERS, 1996) apresentou uma técnica baseada em estatísticas do contorno de energia e da taxa de cruzamentos pelo zero, e reportou um acerto de 98% na classificação de comerciais de rádio. Eric Scheirer e Malcolm Slaney utilizaram uma técnica baseada na combinação de diversas características, energia de modulação, "centróide espectral", e taxa de cruzamento pelo zero, além de se valer de diversos classificadores, e reportam uma taxa de erro de 1.4% sobre uma grande coleção de radiodifusão FM. Michelle Spina e Victor Zue (SPINA; ZUE, 1996) utilizaram noticiários de rádio, e foram capazes de atingir 80,9% de acerto na classifcação dos conteúdos em sete categorias: limpo, telefone, fala ruidosa, silêncio, música e fala mais música.

O passo seguinte, após a distinção de um conteúdo de áudio seria permitir a busca de conteúdos. Isto requer alguma medida de similaridade entre conteúdos de áudio, o que é um assunto bastante complexo. Pode-se usar conceitos simples para similaridades de texto, como o número de palavras em comum, a ordem em que as palavras aparecem,

etc., porém, no domínio do áudio, estas medidas não estão tão claras. Para tentar evitar definir tal conceito, técnicas de inteligência artificial como redes neurais e mapas auto-organizáveis foram utilizadas, pelo fato de serem capazes de lidar com conceitos que não estão formalmente definidos.

Em 1996, um grupo em Muscle Fish LLC (WOLD et al., 1996) produziu um interessante trabalho, que se valia de características psicoacústicas para caracterizar arquivos de áudio. Um classificador Gaussiano foi utilizado para analisar os arquivos, e uma distância de Mahalanobis foi utilizada para estabelecer a similaridade entre os conteúdos.

Simplificações do problema foram feitas para se permitir um entendimento melhor da área, assim alguns trabalhos utilizaram arquivos MIDI (Musical Instrument Digital Interface), que já possuem uma representação das notas da música. Pesquisadores em Cornell conseguiram bons resultados ao estabelecer três níveis de quantização, dependendo se uma nota seguinte fosse mais alta, mais baixa, ou similar à nota anterior (GHIAS et al., 1995). Algoritmos de busca baseados em cadeias também foram utilizados em tal tarefa, como nos trabalhos produzidos pela Universidade de Waikato na Nova Zelândia (MCNAB LLOYD A. SMITH, 1996).

Na *web*, podem ser encontrados alguns sistemas que se propõem a realizar esta tarefa de busca de áudio. Nestes, diversas formas de entrada para a busca são utilizadas, além é claro da diversidade de algoritmos que são empregados. Entre os principais, podem ser citados: midomi[1] e musipedia[2].

[1] `http://www.midomi.com/`
[2] `http://www.musipedia.org/`

4 Conceitos

Na presente seção serão apresentados os diversos conceitos nas quais este trabalho se fundamenta. Além de uma visão geral dos aspectos fundamentais de cada assunto, são dadas referências para material detalhado para o completo entendimento destes.

4.1 Processamento de sinais

4.1.1 Transformada de Fourier

A tranformada de Fourier é a generalização das séries de Fourier, onde, a idéia é aproximar uma função f em um determinado intervalo por um soma de *cossenos*.

A definição formal é apresentada na equação 4.1.

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i k x}\, dx \tag{4.1}$$

Assim, no contexto deste trabalho, em um sinal dado por $f(x)$, para cada frequência k, $F(k)$ denota a intensidade da frequência k do sinal $f(x)$.

Maiores referências sobre o assunto podem ser encontradas em (BRACEWELL, 1980).

Esta definição formal não é adequada para utilização computacional, já que seu tratamento simbólico é difícil, assim técnicas que adequam esta equação à utilização computacional foram desenvolvidas, e uma de grande importância no contexto abordado, é a FFT - Fast Fourier Transform (Transformada Rápida de Fourier). Este algoritmo permite calcular a transformada discreta de um sinal de forma eficiente. A definição formal da transformada discreta é dada pela equação 4.2.

$$F_n \equiv \sum_{k=0}^{N-1} f_k e^{2\pi i n \frac{k}{N}} \tag{4.2}$$

Maiores referências podem ser encontradas em (BRIGHAM, 1988).

No contexto da análise de sinal a FFT será usada para o cálculo do espectro de intensidade de um sinal, que consiste em calcular a FFT do sinal para janelas de tempo definidas, calculando a intensidade do sinal para cada intervalo e frequencia, gerando assim uma superfície que contém a informação de intensidade para todas as frequências do sinal, em cada instante de tempo. A teoria sobre espectros de intesidade pode ser encontrada em (OPPENHEIM; SCHAFER; BUCK, 1999).

4.2 Método dos mínimos quadrados

O método dos mínimos quadrados é uma técnica matemática de otimização que surgiu no início do século XIX, a partir de necessidades relacionadas a geolocalização. Este método permite encontrar os parâmetros para uma função modelo f de forma a melhor aproximar uma relação entre grandezas. Esta relação é usualmente dada por um conjunto de pares ordenados (x_i, y_i). Neste caso, se imaginarmos que $f(x_i) \approx y_i$, deseja-se ajustar seus parâmetros de modo a minimizar a soma dos quadrados dos erros:

$$S = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

A teoria mostra que para o caso em que a função modelo f é linear com relação aos seus parâmetros, a solução para o problema é única e ocorre quando as derivadas parciais do erro quadrático com relação a cada parâmetro é zero. Estas equações resultam em um sistema linear possível e determinado, de forma que a solução pode ser facilmente obtida utilizando os métodos numéricos de resolução de sistemas lineares.

Para um aprofundamento sobre o método dos mínimos quadrados ou sobre métodos de resolução de sistemas lineares, consulte a referência (HUMES I.S.H. DE MELO, 1984).

4.3 Adaptatividade

O conceito de adaptatividade (NETO, 2004) está vinculado à característica fundamental que diferencia os assim chamados dispositivos adaptativos dos demais. Um dispositivo adaptativo nada mais é do que um formalismo computacional (JOHNSONBAUGH, 2000) com capacidade de auto-reconfiguração dinâmica. Como todo formalismo computacional, estes dispositivos tem seu comportamento baseado em um conjunto de regras. A

auto-reconfiguração dinâmica significa a possibilidade de estas regras serem modificadas durante a operação do dispositivo.

Na prática, a adaptatividade pode se traduzir de diferentes maneiras. Usualmente, a característica adaptativa é introduzida a dispositivos já definidos, como autômatos finitos, gramáticas, autômatos de pilha, linguagens de programação etc. O que define o comportamento de um autômato finito, por exemplo, é o seu conjunto de estados e suas regras de transição. Portanto, um autômato adaptativo (NETO, 1994) é como um autômato comum com a diferença que a recepção de um símbolo de entrada, além de ocasionar uma transição de estados, opcionalmente pode disparar uma ação adaptativa que irá modificar as regras de transição e o conjunto de estados.

A título de exemplo, considere o autômato finito da figura 2.

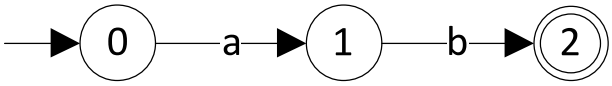

Figura 2: Primeiro exemplo de autômato

Este autômato reconhece a linguagem $L = \{ab\}$. Podemos conceber um autômato adaptativo definindo uma ação adaptativa que adiciona novos estados e transições ao receber um símbolo a no estado 1, resultando na configuração ilustrada pela figura 3.

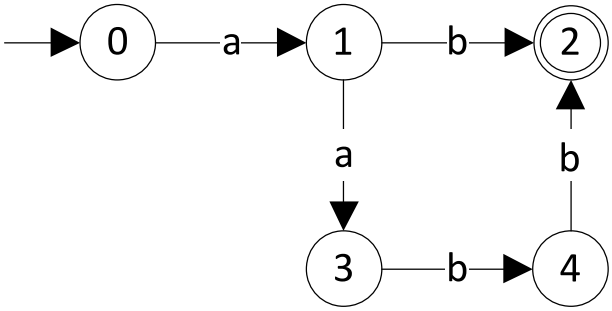

Figura 3: Configuração do autômato após primeira ação adaptativa

Este novo autômato adaptativo é capaz de reconhecer a linguagem $L = \{ab, aabb\}$. Pode-se aplicar esta mesma idéia não apenas ao estado 1 como também para todos os estados ímpares do autômato. Após receber a subcadeia aaa, tal autômato assumiria a configuração mostrada na figura 4. Temos então um dispositivo capaz de reconhecer a linguagem livre de contexto definida pela expressão $L = aa^nbb^n$.

No caso dos autômatos finitos, é simples notar que a introdução da adaptatividade torna o dispositivo muito mais poderoso, sob o aspecto de classes de linguagens que o dispositivo é capaz de reconhecer. Porém a teoria nos mostra que os formalismos

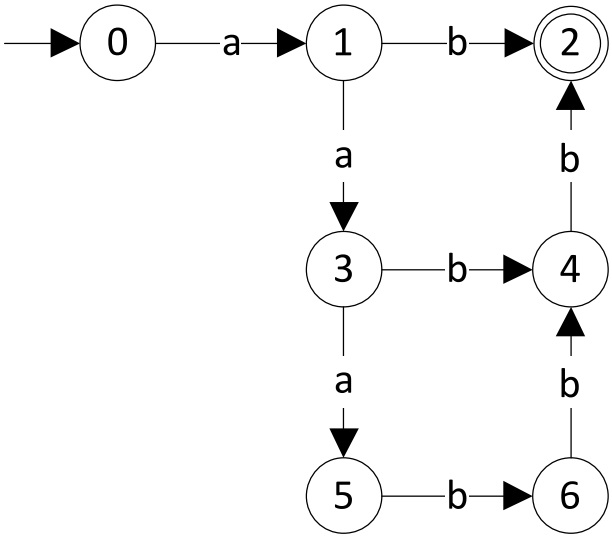

Figura 4: Configuração do autômato após segunda ação adaptativa

adaptativos conhecidos até então são equivalentes à máquina de Turing.

Apesar desta equivalência, os dispositivos adaptativos não perdem sua relevância. Isto ocorre pelo fato de existirem muitos mecanismos computacionais cuja descrição completa não é explícita. Nestes casos, torna-se inviável a implementação destes a partir de formalismos estáticos. Por outro lado, uma descrição "indireta" e incremental eventualmente pode ser muito mais facilmente concebida, e servir de base para a implementação do mecanismo através de um formalismo adaptativo.

5 *Técnicas e procedimentos usados*

Este capítulo descreve detalhes do sistema desenvolvido, apresentando sua especificação, detalhes de técnicas utilizadas, além de procedimentos envolvidos.

5.1 Especificação do sistema

Propõe-se o desenvolvimento de um módulo de busca por músicas baseado em conteúdo apoiado em técnicas adaptativas, e para teste e avaliação da tecnologia empregada um protótipo de um sistema mais completo será construído.

5.1.1 Descrição

A idéia geral de um sistema de busca está implícita na maioria das pessoas que se utilizam de serviços como os de busca por documentos, nestes serviços, em geral o usuário entra com um trecho do documento que ele procura, e o sistema de busca encontra documentos que mais se aproximem do trecho que o usuário proveu. Analogamente, em um sistema de busca por áudio baseado em conteúdo, o usuário provê um trecho do áudio que deseja encontrar, e o sistema encontra os áudios mais similares.

5.1.2 Uso do sistema

Propõe-se o desenvolvimento de um protótipo de um sistema de buscas por músicas baseado em conteúdo, que receba uma entrada do usuário que corresponde à sua *query*, ou seja, algum trecho da música buscada que o usuário reproduza através de um assobio, e em seguida compara com as músicas presentes em seu repositório, calculando a similaridade entre cada música e a *query*, a partir das comparações efetuadas, o sistema é capaz de apresentar quais as entradas mais prováveis de corresponderem à música procurada.

Em linhas gerais, a idéia de uso do sistema pode ser vista na figura 5.

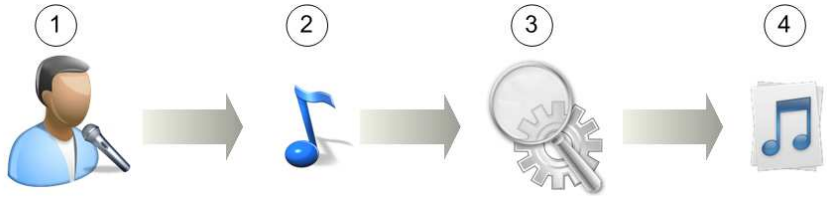

Figura 5: Caso de uso típico do sistema

1. Usuário do sistema
2. Trecho de música (ou conteúdo) gerado pelo usuário
3. Sistema de busca
4. Conjunto de músicas mais próximas do trecho gerado pelo usuário

5.1.3 Requisitos funcionais e Premissas

Apesar da simplicidade da idéia geral, existe uma complexidade considerável nos detalhes. Assim deve-se explicitar alguns requisitos que definirão o sistema, além de premissas que serão assumidas durante o desenvolvimento.

5.1.3.1 Entrada de dados

O usuário deve fornecer como entrada ao sistema um assobio de um trecho de uma música que deseja buscar, originando a *query* que será processada. As informações fornecidas pelo usuário, em geral, são muito limitadas e com grandes variações com relação ao conteúdo original, além disso tipicamente o trecho é curto e erros são frequentes. Para fins de processamento da *query* a sua origem é irrelevante. Assim, outras formas de entrada que produzissem notas musicais diretamente, como um teclado MIDI por exemplo, seriam passíveis de utilização. No caso do assobio fornecido pelo usuário, o mesmo deve estar codificado no formato WAV.

5.1.3.2 Tipo de busca

O sistema restringe-se a procurar por melodias, ou no caso mais geral, sequências de notas musicais, não sendo adequado portanto para encontrar trechos cantados, por exemplo. Outras formas de busca devem ser abordadas com diferentes técnicas.

5.1.3.3 Espaço de busca

O espaço de busca é constituído por músicas em formato MIDI, devidamente preparadas para o ambiente de execução de buscas. Na maioria dos casos, os arquivos MIDI possuírem diversas trilhas. Por este motivo, precisam passar por um processo de preparação em que apenas as trilhas melódicas mais relevantes da música são extraídas e adicionadas ao repositório.

5.1.3.4 Resposta do sistema

A resposta do sistema deve conter a lista de músicas do repositório mais similares à entrada do usuário. Para ordenação das músicas um critério de similaridade será definido, ao qual o número de notas em comum (entre uma música e a *query* fornecida) deverá exercer grande influência.

5.1.4 Arquitetura do Sistema

Nesta seção serão apresentados diagramas de descrição específica do sistema, mostrando o seu fluxo de informação, seus componentes e suas interações.

A arquitetura proposta para o protótipo é dividida em componentes. Cada um destes tem seu papel funcional bem determinado no conjunto, de modo que as implementações podem ser facilmente substituídas. Esta característica possibilitou uma evolução contínua do protótipo.

5.1.4.1 Fluxo de informação do sistema

O usuário deverá fornecer a entrada ao sistema, que por sua vez, digitalizará e converterá o sinal para um modelo musical que representará o trecho fornecido, ou seja, a *query*. Em seguida, esta será apresentada ao módulo de busca do sistema, que analisará a mesma comparando-a com as músicas existentes no repositório. Um dos pilares da comparação é a adaptatividade, que ajusta o comparador principal. Ao fim da comparação com as entradas do repositório, é possível estabelecer o conjunto de músicas mais prováveis de atender à *query* do usuário. Este fluxo pode ser acompanhado na figura 6.

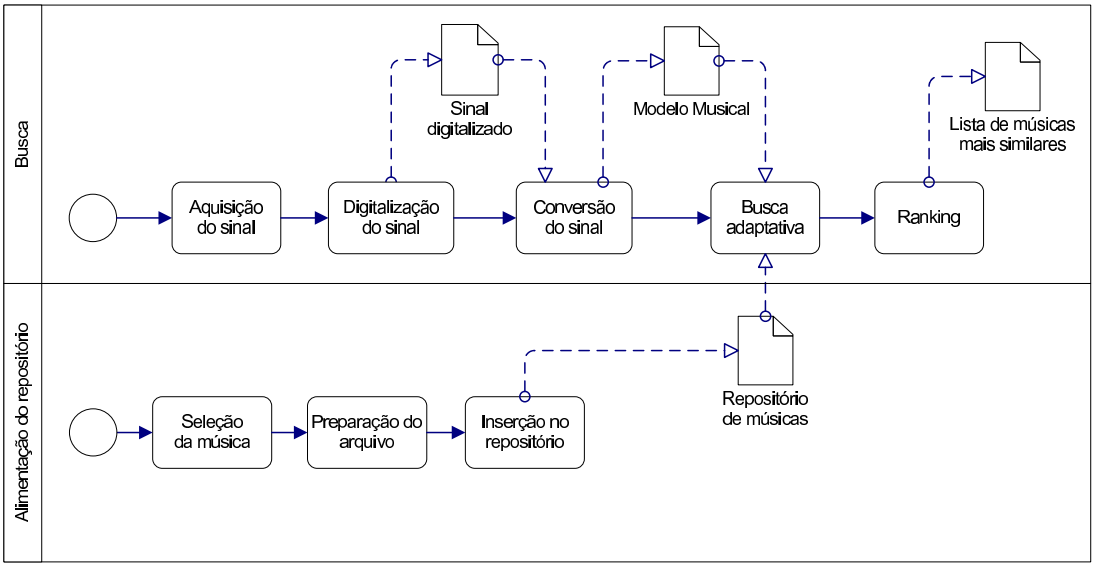

Figura 6: Fluxo de informação do sistema

5.1.4.2 Diagrama de componentes

O diagrama de componentes da figura 7 mostra como as diversas partes do sistema interagem. Este diagrama é especialmente importante por delimitar as fronteiras entre as partes do sistema, permitindo assim que os componentes evoluam separadamente, podendo-se trocar implementações ou métodos utilizados sem afetar o funcionamento geral do sistema.

1. *Aquisição do sinal*: É responsável por gravar o áudio produzido pelo usuário, e armazená-lo para que o conversor em seguida possa analisá-lo. Um arquivo WAV é usado para guardar o áudio, já que este tipo de formato apresenta perdas desprezíveis para o processo.

2. *Conversor*: A partir do sinal de áudio digitalizado, indentifica e extrái as informações musicais. Ou seja, produz um modelo musical contendo informações de alturas e tempos das notas identificadas na amostra.

3. *Comparador*: Recebe dois modelos musicais e os compara utilizando um autômato adaptativo, gerando uma medida de similaridade entre os modelos recebidos.

4. *Repositório*: Armazena as músicas que servem de base para a busca (em formato MIDI), e permite serviços de gerenciamento do repositório.

5. *Ranking*: Ordena uma lista de músicas de acordo com critérios previamente definidos, baseando-se na similaridade e nas informações geradas a partir da comparação.

6. *Buscador*: A partir do modelo musical do assobio do usuário e das músicas contidas no repositório, utiliza o comparador para levantar as medidas de similaridade entre o trecho recebido as músicas existentes no repositório. Após isso, utiliza-se do Ranking para ordenar a lista de músicas, para finalmente apresentar a resposta do sistema.

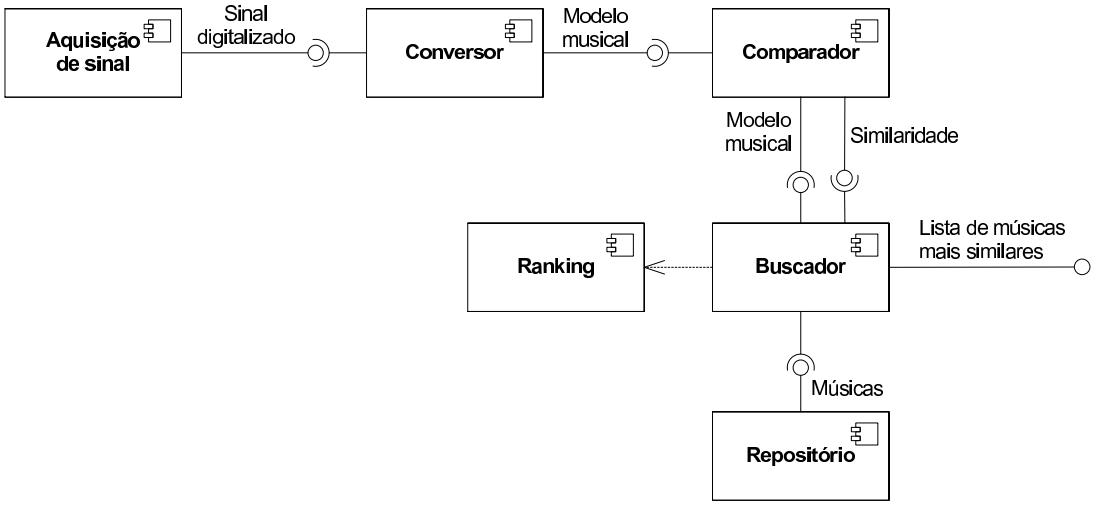

Figura 7: Diagrama de componentes do sistema

5.1.5 Escopo

O sistema proposto é bastante amplo e permeia diversas áreas do conhecimento. Devido a esta abrangência, uma limitação do escopo é fundamental. O foco do trabalho é a utilização de técnicas adaptativas no reconhecimento de padrões musicais, assim o componente de maior importância é naturalmente o *comparador* que é o componente que se utilizará de tais técnicas. Esta é uma aplicação inovadora, e tem o potencial de gerar bons resultados, justificando a concentração dos esforços.

Porém, a validação do *comparador* e a avaliação de seu desempenho depende fundamentalmente dos dados gerados pelos outros componentes. Assim, a idéia foi criar implementações simples, que fossem capazes de prover uma infra-estrutura de teste para o *comparador*, porém tal método não foi aplicável a todos os componentes, como foi o caso do *conversor*, exigindo um certo refino de sua implementação. Assim foi dada uma atenção secundária ao *conversor*, com o fim de poder prover dados reais de teste para o *comparador*. Os outros componentes tiveram implementações simples, porém suficientemente boas para prover uma prova de conceito adequada.

5.2 Extração de notas

A idéia utilizada para extração de notas constitui da quebra o sinal de áudio em diversas janelas de tempo pequenas, extraindo-se em seguida a transformada de Fourier destas janelas, encontrando-se assim a distribuição de frequências para cada intervalo. A análise destas distribuições permite encontrar os picos de frequência, e a partir destes picos pode-se encontrar as notas entoadas pelo usuário.

Assim, neste sistema o processo de extração de notas envolve dois componentes distintos: o componente de *aquisição de sinal* e o *conversor*, sendo constituído das seguintes fases:

- Aquisição do sinal
- Extração do espectrograma
- Filtro de intensidade
- Extração de picos de frequência
- Identificação das notas
- Geração do modelo musical
- Sintetização de áudio

Estas fases são executadas em sequência, até gerar a principal saída: o modelo musical, que proverá a entrada para o sistema de busca, enquanto o áudio sintetizado posteriormente tem o objetivo de prover um feedback do processo de extração de notas, permitindo avaliar a qualidade do sistema. A seguir cada uma destas fases serão detalhadas, mostrando as técnicas utilizadas em cada uma destas.

A técnica descrita faz parte de uma classe de métodos baseados em análise de frequência, diversas abordagens similares foram desenvolvidas ao longo dos anos, em trabalhos como (PISZCZALSKI, 1986) e (Piszczalski; Galler, 1979).

5.2.1 Aquisição do sinal

O primeiro passo para permitir o reconhecimento é a aquisição do sinal de áudio (assobio) gerado pelo usuário. Isso pode ser facilmente obtido através de um dispositivo

de gravação, como um microfone, e um software de captura de áudio. O resultado dessa etapa é armazenado em um arquivo WAV, que reproduz o assobio de forma integral.

5.2.2 Extração do espectrograma

A partir deste arquivo de áudio se extrai seu espectrograma, que é a transformada de Fourier para cada janela de tempo do sinal. O sinal é particionado em intervalos de tempo regulares, e para cada intervalo a distribuição de frequências é calculada. O espectrograma é uma reprodução muito próxima do sinal original, porém, transportado para o domínio da frequência. A fase de cada componente de frequência não é relevante para a análise, porém sua intensidade é extremamente importante, assim, as intensidades de cada componente são calculadas, o resultado pode ser encarado como uma superfície 3D, tendo como eixo X o tempo, eixo Y a frequência e eixo Z a intensidade do sinal (em dB).

A figura 8 mostra o sinal de áudio original de um assobio contendo um trecho da 9.ª Sinfonia de Beethoven, também conhecido como Ode à Alegria, e a figura 9 mostra seu espectrograma. Já pelo espectrograma é possível perceber os limiares das notas reproduzidas.

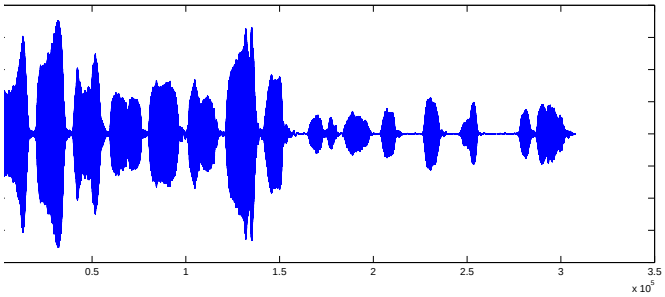

Figura 8: Sinal de áudio de um assobio

5.2.3 Filtro de intensidade

O espectrograma original possui uma distribuição de frequências contendo diversas componentes provenientes de ruído, ou de intensidade muito baixa, que são pouco relevantes para a identificação das notas reproduzidas pelo usuário, assim aplica-se um filtro de intensidade, eliminando as frequências de baixa intensidade, produzindo assim um espectro mais simples de ser trabalhado. O resultado da aplicação deste filtro pode ser visto na figura 10.

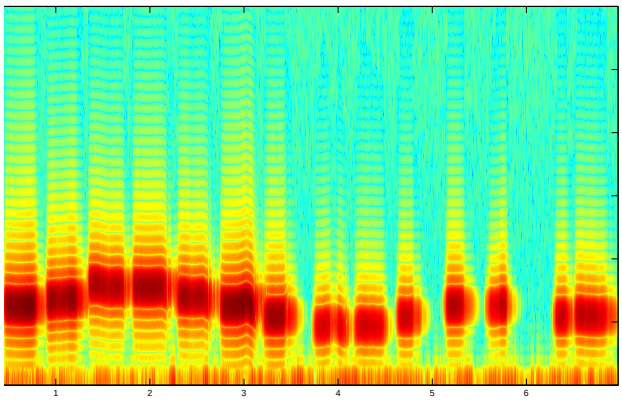

Figura 9: Espectrograma do sinal correspondente a um assobio

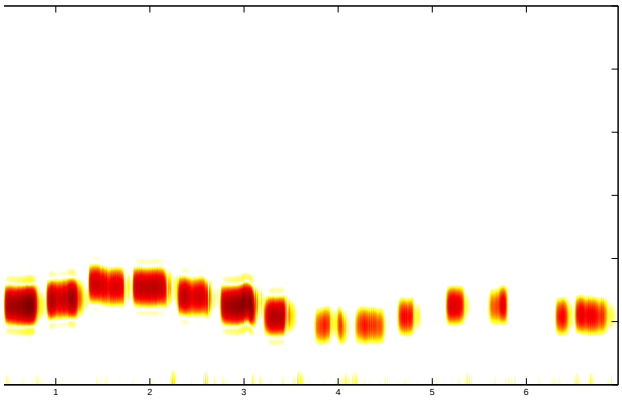

Figura 10: Espectrograma após aplicação do filtro de intensidade

5.2.4 Extração de picos de frequência

Após a eliminação do ruído, o sinal é analisado, extraindo-se os picos de frequências, que em geral representam o contorno da melodia que o usuário tentou reproduzir. Este contorno de frequências servirá de base para a identificação das notas, o resultado pode ser visto na figura 11.

5.2.5 Identificação das notas e Geração do modelo musical

A partir dos picos de frequências encontrados na etapa anterior, já é possível, visualmente, identificar aproximadamente as notas (início, fim e frequência). Assim, como o objetivo do trabalho consistia da criação de um prtótipo que teria o fim de testar e avaliar o desempenho do mecanismo de comparação adaptativo, percebe-se que não seria necessário avançar além deste ponto na extração de notas, já que com tais resultados seria

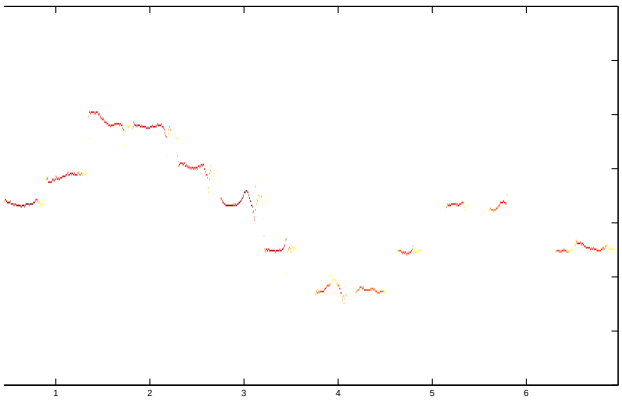

Figura 11: Espectrograma após extração dos picos de frequência

possivel extrair dados de testes reais para o *comparador*.

Os dados extraídos dão origem ao modelo musical que servirá como uma representação do áudio original, este modelo pode ser visto como uma tabela, que possui a lista das notas geradas pelo usuário, um exemplo pode ser visto na tabela 1.

Tabela 1: Notas extraídas a partir do espectrograma

Nota	Início	Fim	Frequência
1	0.0174	0.3483	1277.2
2	0.4586	0.8533	1273.2
3	0.9172	1.2945	1370.1
4	1.3468	1.7125	1573.3
5	1.8460	2.1595	1573.3
6	2.3046	2.6122	1407.7
7	2.7516	3.1057	1283.9
8	3.2044	3.4540	1098.2
9	3.7326	3.9706	944.77
10	4.1796	4.4873	948.81
11	4.6150	4.8414	1092.8
12	5.1316	5.3348	1267.8
13	5.6018	5.7992	1265.1
14	6.3042	6.8557	1102.2

5.2.6 Sintetização de áudio

A partir do modelo extraído do áudio é possível produzir um arquivo MIDI contendo a reprodução das notas encontradas num piano. Com a reprodução deste arquivo é possível avaliar o desempenho do processo de extração de notas, além de avaliar a qualidade dos

dados que serão providos para a busca. Uma avaliação simples e imediata, é a comparação auditiva da reprodução do arquivo MIDI contendo as notas extraídas com o arquivo contendo o aúdio original do assobio.

5.3 Representação de notas e melodias

A seção anterior descreveu o processo de extração de notas. Conforme foi visto, este processo analisa uma amostra de áudio de duração típica de três a quinze segundos, e produz uma tabela contendo os seguintes dados sobre as notas extraídas: altura (frequência fundamental de vibração ou *pitch*) e tempos de início e fim. Vale observar que nesta tabela as notas aparecem ordenadas cronologicamente e que não há sobreposição temporal destas.

A representação tabular guarda todas as informações musicais da melodia extraída e, portanto, define o formato de entrada para consultas ao mecanismo de busca musical. Porém, internamente a este mecanismo, as melodias são representadas na forma de listas de eventos musicais. Neste modelo interno, um evento musical pode ser uma nota ou um silêncio[1], e guarda suas duas características fundamentais: altura e duração. A altura é mantida em *Hertz* e a duração em segundos. Para simplificar a representação, adotou-se um valor zero para a altura dos silêncios. A adoção desta representação justifica-se por uma questão de conveniência, uma vez que os algoritmos do mecanismo de busca são fundamentalmente baseados em listas.

Musicalmente os silêncios, ou pausas, são considerados elementos tão importantes quanto as próprias notas. Porém analisando reproduções de uma mesma melodia por diferentes interpretes, nota-se uma maior simetria nos ataques[2] das notas do que em seus fins. Assim, os tempos de ressonância das notas, e as durações dos silêncios são muito variáveis. Ao contrário dos intervalos entre os ataques, que tendem a se manter mais estáveis.

Conclui-se, portanto, que os intervalos entre os ataques das notas é mais relevante do que as durações exatas das notas e silêncios para definição da característica psicoacústica de uma melodia. Por este motivo, para os fins deste trabalho, os silêncios foram eliminados das representações. Para manter a característica da distância temporal entre ataques de notas, a duração de cada silêncio foi incorporada à duração da nota imediatamente

[1] ou *pausa*
[2] inícios

anterior.

5.4 Proximidade de melodias

Quando uma pessoa canta uma melodia ou a toca em um instrumento, somos eventualmente capazes de identificar a que música aquela melodia pertence. Nosso cérebro é capaz de reconhecer estas semelhanças mesmo na presença de variações ou imprecisões na melodia que ouvimos.

Um exemplo típico de tais variações é a transposição tonal, em que a melodia é reproduzida com uma variação fixa[3] na altura de todas as notas, para mais ou para menos. Outro exemplo é a dilatação ou contração das durações das notas que compõe aquela melodia.

Em muitos casos, somos capazes de identificar músicas mesmo na ocorrência de *erros* na reprodução, tais como uma nota errada (com altura diferente), ou mesmo a omissão ou adição de notas à melodia original. Estes erros são, em geral, provenientes da incapacidade ou imprecisão do próprio executor.

5.5 Comparação numérica

Em uma situação hipotética onde não há presença de erros, pode-se analisar a proximidade entre duas melodias com a mesma quantidade de notas, definindo um modelo matemático que mapeia as notas de uma melodia nas notas da outra. Sendo p_1 e p_2, respectivamente as alturas de uma nota do trecho 1 e sua correspondente no trecho 2; e d_1 e d_2 as durações destas; a relação que mapeia as durações é do tipo:

$$d_1 = A.d_2$$

A constante A representa uma proporcionalidade entre as durações, portanto o modelo adotado permite dilatações e contrações proporcionais.

Para mapear as alturas utiliza-se a seguinte relação:

$$\log p_1 = \log p_2 + B$$

A constante B representa a transposição tonal. A relação logarítmica é necessária

[3]Variação fixa na escala logarítmica significa o produto por uma constante

pelo fato de que a percepção do ouvido humano para alturas de notas é exponencial.

A partir destas relações de aproximação, calculam-se os parâmetros A e B que melhor aproximam a distribuição segundo o critério de proximidade do método dos mínimos quadrados, isto é, aqueles que minimizem a soma dos erros quadráticos:

$$S_d = \sum_{i=1}^{N} (A.d_{2i} - d_{1i})^2 \tag{5.1}$$

$$S_p = \sum_{i=1}^{N} (\log p_{2i} + B - \log p_{1i})^2 \tag{5.2}$$

A teoria (HUMES I.S.H. DE MELO, 1984) nos mostra que o mínimo de cada uma destas funções ocorre quando suas derivadas com relação ao parâmetro atingem o valor zero:

$$\frac{\partial S_d}{\partial A} = 0$$

$$2\sum_{i=1}^{N} (A.d_{2i} - d_{1i}) d_{2i} = 0$$

$$\sum_{i=1}^{N} (A.d_{2i}^2 - d_{1i} d_{2i}) = 0$$

$$A = \frac{\sum_{i=1}^{N} d_{1i} d_{2i}}{\sum_{i=1}^{N} d_{2i}^2} \tag{5.3}$$

$$\frac{\partial S_p}{\partial B} = 0$$

$$2\sum_{i=1}^{N} (\log p_{2i} + B - \log p_{1i}) = 0$$

$$\sum_{i=1}^{N} (\log \frac{p_{2i}}{p_{1i}} + B) = 0$$

$$B = \frac{\sum_{i=1}^{N} \log \frac{p_{2i}}{p_{1i}}}{N} \tag{5.4}$$

Os valores A e B obtidos, eventualmente podem ser utilizados para avaliar a proximidade entre as melodias. Porém, nesta modelagem, o relevante não são os parâmetros obtidos da redução, e sim, a soma quadrática dos erros ao utilizá-los, dados pelas equações 5.1 e 5.2.

Quanto menor forem os valores destas somas, mais próximas são as melodias comparadas. A distância entre estas é então definida por uma soma ponderada destes valores, com pesos ajustáveis:

$$d = \alpha S_d + \beta S_p \tag{5.5}$$

5.6 Busca inexata com autômato adaptativo

Com a definição de proximidade entre melodias apresentada acima, seria possível construir um mecanismo de busca de uma melodia sobre um repositório de músicas, utilizando uma janela deslizante do tamanho da melodia de entrada e varrendo sobre todas as melodias do repositório. Porém tal mecanismo só seria efetivo no caso restrito em que a melodia de entrada não possui imperfeições como a ausência ou adição de notas.

Conforme já discutido anteriormente, estas imperfeições ocorrem com certa frequência e, em condições habituais, não são suficientes para impedir que uma pessoa seja capaz de identificar a música executada. Esta consideração motiva a idealização de um mecanismo de comparação que seja capaz de lidar com tais imperfeições.

No capítulo 3, diversas abordagens dadas a este problema foram apresentadas. Porém, não se localizou na literatura nenhum estudo que cite o uso de técnicas adaptativas com este fim. Propõe-se, então, um novo método de reconhecimento de padrões musicais baseado em autômatos adaptativos.

Neste método, constrói-se um autômato adaptativo (NETO, 1994) automaticamente, a partir da melodia de entrada (consultada), que funciona como um reconhecedor de melodias semelhantes a esta. Este autômato é então utilizado para processar todo o repositório de melodias e elencar as melhores semelhanças.

Como já foi dito anteriormente, a utilização deste formalismo para a comparação de melodias vem da necessidade de reconhecer melodias contendo imprecisões, naturais da reprodução humana. Sendo assim, o autômato foi projetado para lidar com três tipos de situações de erro na melodia de entrada. São elas:

1. Omissão de uma nota

   Situação em que uma nota da melodia procurada foi omitida da melodia de entrada.

2. Adição de uma nota

Situação em que uma nota que não faz parte da melodia procurada foi inserida na melodia de entrada.

3. Troca de uma nota

   Situação em que uma nota da melodia procurada foi substituída por outra qualquer na melodia de entrada.

5.7 Notas como símbolos

Para ser capaz de processar melodias (tanto a da consulta como as da base de dados), o autômato precisa enxergá-las na forma de cadeias de símbolos de um alfabeto finito. Como nesta etapa deseja-se especificamente reconhecer melodias contornando os três tipos de erros enumerados anteriormente, pode-se considerar apenas a altura das notas, desprezando inicialmente as durações.

Porém, os valores possíveis de altura das notas constituem um domínio contínuo e portanto precisam ser ajustados para um domínio discreto, que constituirá o alfabeto do autômato. Para este domínio discreto, escolheu-se utilizar o conjunto de alturas das notas de um piano. Este conjunto, conhecido musicalmente como *temperamento igual de 12 tons*, é o sistema de afinação predominantemente utilizado na música ocidental moderna (BURNS, 1999).

O MIDI[4] (MIDI...,) é um padrão *de facto* que define um protocolo para comunicação entre instrumentos musicais eletrônicos e outros equipamentos de áudio. Entre os diversos outros detalhes do protocolo, o MIDI define um código para representação das notas do sistema de afinação ocidental (notas do piano). Este código é um número inteiro entre 0 e 127, que é capaz de representar muito além da capacidade audível da maioria dos seres humanos. A nota 0, por exemplo, é uma nota Dó cinco oitavas abaixo do Dó central e corresponde a uma frequência de 8,176 Hz. Por conveniência, adotaremos este código de notas MIDI como alfabeto do autômato.

A primeira etapa do ajuste de domínio é basicamente uma conversão de unidades. A conversão da altura em Hertz para o código MIDI é dada pela seguinte relação:

$$p = 69 + 12 \times \log_2 \left(\frac{f}{440\,\mathrm{Hz}} \right).$$

Após esta conversão é necessário realizar uma quantização a fim de obter valores in-

[4]Musical Instrument Digital Interface

teiros, discretizando o domínio. Note, porém, que este processo não é tão simples quanto um arredondamento. Pois o que define a característica perceptiva de uma melodia é a relação entre as alturas das notas e não seus valores absolutos, haja visto que a transposição tonal não altera esta característica. Sendo assim, um simples arredondamento poderia ocasionar erros de quantização consideráveis.

Outro aspecto a se considerar é que o executor da melodia pode perder a referência absoluta de afinação durante sua reprodução. Ou seja, para uma melodia suficientemente grande, a referência de afinação para uma determinada nota vem das k notas anteriores e não da melodia inteira.

Considerando este aspecto relativo da reprodução humana, uma pesquisa da Universidade de Waikato (Nova Zelândia) apresentou um simples e interessante método de quantizar estes valores a partir da referência de afinação da nota anterior.

Este método foi reproduzido e testado utilizando valores extraídos de gravações de melodias assobiadas. Porém, os resultados mostraram que, em alguns casos, ocorre uma divergência elevada, maior que eventuais variações da referência absoluta do executor. O gráfico da figura 12 ilustra um destes casos.

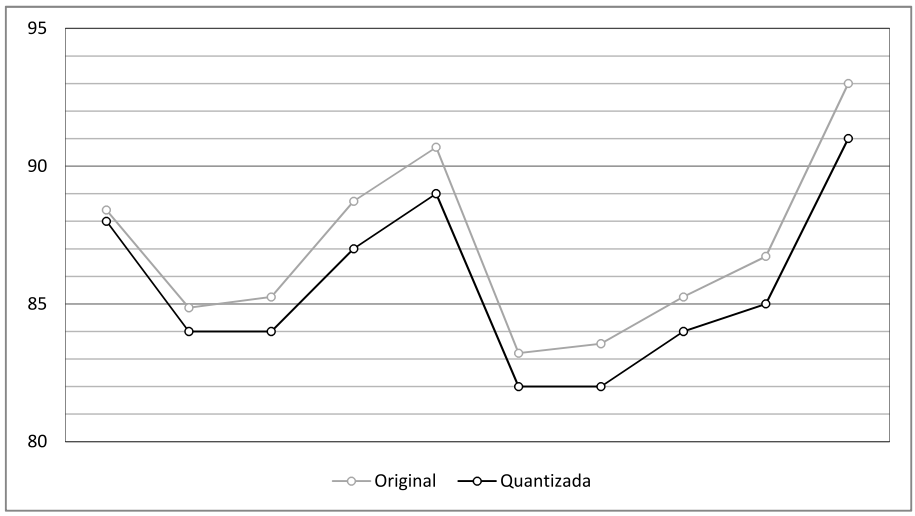

Figura 12: Alturas quantizadas pelo método relativo

Tendo em vista as limitações deste método de quantização relativo, foi desenvolvido um método de quantização absoluto que será descrito em detalhes na seção seguinte.

5.7.1 Quantização absoluta das alturas das notas

O método proposto parte do princípio que a melodia original pode ser transposta de tonalidade livremente, ou seja, pode-se somar uma constante em todas as notas sem que o resultado da quantização perca significado. A partir disto encontra-se analiticamente a constante que minimiza uma métrica de erro de quantização e, por fim, faz-se o arredondamento das notas somadas a esta constante.

A soma do erro quadrático de quantização por arredondamento é dada por:

$$E(0) = \sum_{i=1}^{N} (p_i - \lfloor p_i + \frac{1}{2} \rfloor)^2$$

Com a adição de uma constante c em todos os valores, torna-se:

$$E(c) = \sum_{i=1}^{N} (p_i + c - \lfloor p_i + c + \frac{1}{2} \rfloor)^2$$

Este valor varia com a constante c. Em suma, queremos encontrar o valor de $0 < c \leq 1$ que minimiza E, para então obter os valores quantizados v_i da seguinte maneira:

$$v_i = \lfloor p_i + c + \frac{1}{2} \rfloor.$$

Note que a função E não é contínua. Por este motivo, seu mínimo pode estar ou nos pontos de descontinuidade ou nos pontos em que:

$$\frac{\partial E}{\partial c} = 0. \tag{5.6}$$

Os pontos de descontinuidade ocorrem quando $c = \frac{1}{2} + \lfloor p_i + \frac{1}{2} \rfloor - p_i + k$, $k \in \mathbb{Z}$, para qualquer p_i. Estes pontos são candidatos a mínimo de E. Entre dois destes pontos consecutivos c_1 e c_2, E é contínua e então podemos desenvolver a equação 5.6:

$$\sum_{i=1}^{N} (p_i + c - \lfloor p_i + c + \frac{1}{2} \rfloor) = 0 \tag{5.7}$$

Por termos restringido o intervalo para uma região contínua, o termo $\lfloor p_i + c + \frac{1}{2} \rfloor$ agora passa a ser constante. Para calculá-lo basta utilizar para c um valor qualquer do intervalo, como por exemplo a média dos extremos:

$$\bar{c} = \frac{c_1 + c_2}{2}$$

Com isso, a equação 5.7 fica:

$$c = \frac{\sum_{i=1}^{N}(\lfloor p_i + \bar{c} + \frac{1}{2} \rfloor - p_i)}{N} \tag{5.8}$$

Se o valor obtido para c estiver no intervalo $]c_1, c_2[$ este será solução da equação 5.6 e, portanto, um novo candidato a mínimo de E. Aplica-se este procedimento para todos os trechos entre pontos de descontinuidade do intervalo $]0, 1]$ e se obtem desta forma todos os candidatos a mínimo de E. Basta verificar os valores de E para cada candidato e escolher aquele que a minimiza.

O gráfico da figura 13 mostra uma comparação dos resultados dos dois métodos de quantização utilizando a mesma melodia da figura 12.

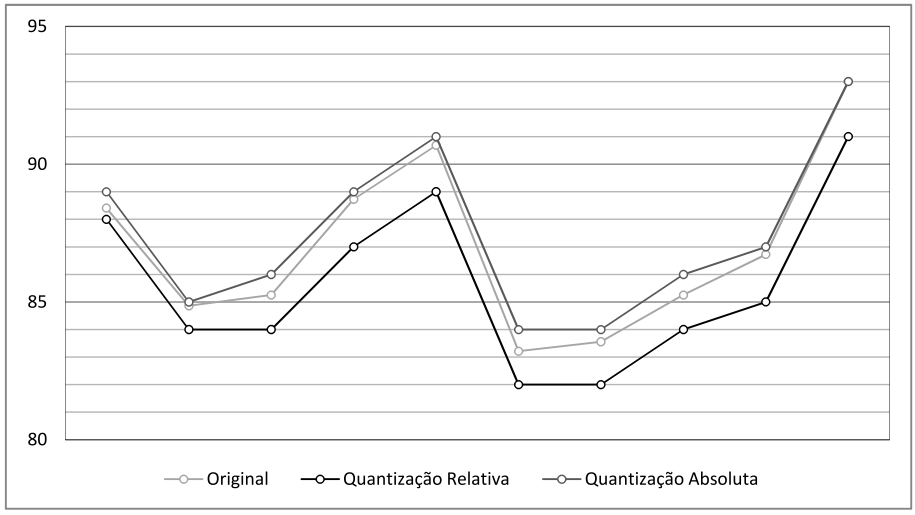

Figura 13: Comparação dos dois métodos de quantização de alturas

5.7.2 Construção do autômato adaptativo

A configuração inicial do autômato adaptativo é obtida através da cadeia de entrada do processo de busca, isto é, a cadeia a ser localizada no repositório. Seja a cadeia de entrada v_i, $i = 0, \ldots, N-1$. O autômato é construído inicialmente com $N+1$ estados, numerados de 0 a N, onde apenas o estado N é final. Adiciona-se transições do estado i para o estado $i+1$ com o símbolo v_i, para $i = 0, \ldots, N-1$.

Para ilustrar o funcionamento do autômato, considere a seguinte sequência de notas: 69, 71, 73, 74, 76, 77, 76. A figura 14 mostra a configuração inicial do autômato adaptativo gerado para esta sequência.

A execução do autômato sobre uma cadeia qualquer se inicia com o cálculo de uma

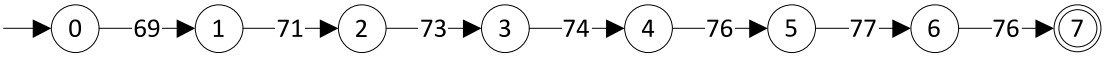

Figura 14: Configuração inicial do autômato adaptativo

constante de transposição. Esta constante nada mais é do que a diferença entre a primeira nota da cadeia de entrada e a da cadeia que originou o autômato. Este valor é descontado dos símbolos de entrada a cada leitura, e serve fundamentalmente para desfazer uma possível transposição tonal.

O caminho definido pelos estados de 0 a N, deste ponto adiante chamado de caminho de referência, ocorre quando a cadeia de entrada equivale exatamente à cadeia procurada, a menos da constante de transposição.

Estando no caminho de referência, ao receber um símbolo para o qual não existe transição, ocorre uma ação adaptativa em que uma estrutura de novos estados e transições é incorporada ao autômato a partir do estado corrente.

Suponha que o autômato da figura 14 recebeu na entrada as notas 69 e 71, atingindo o estado 2. Em seguida, o autômato recebeu o símbolo 74 e disparou a ação adaptativa. A figura 15 mostra a configuração deste autômato após esta ação.

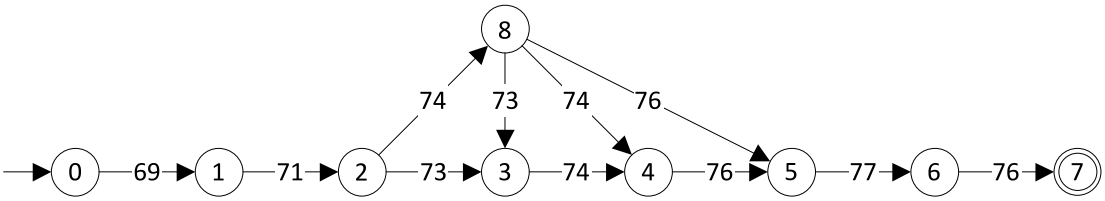

Figura 15: Configuração do autômato após ação adaptativa 1

O autômato atinge o novo estado 8 e, a partir das novas transições incorporadas, torna-se capaz de lidar com as três situações de erro.

1. Omissão de uma nota

   Se a nota 74 recebida fora omitida da cadeia que gerou o autômato, o reconhecimento volta para o caminho de referência a partir do estado 3 ao receber a nota 73, que esperava anteriormente.

2. Adição de uma nota

   Se a nota 73, que era esperada, fora inserida erroneamente na cadeia que gerou o autômato, o reconhecimento volta para o caminho de referência a partir do estado 5 ao receber a próxima nota da sequência: 76.

3. Troca de uma nota

   Se a nota 74 fora trocada por engano pela 73 na cadeia que gerou o autômato, o reconhecimento volta para o caminho de referência a partir do estado 4 ao receber a nota seguinte: 74.

Note que o caso de adição de uma nota só pôde ser contornado por que a nota recebida coincidiu com a segunda nota esperada do autômato. Quando isto não ocorrer, a transição que contorna este caso não é gerada. Um exemplo desta situação pode ser observado na figura 16.

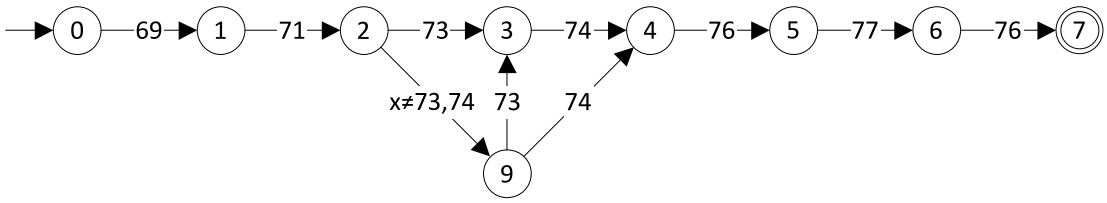

Figura 16: Configuração do autômato após ação adaptativa 2

É importante observar que este autômato é capaz de reconhecer melodias contendo múltiplos erros, desde que devidamente espaçados. A ocorrência de dois erros consecutivos implica na não aceitação da cadeia, pois não há ações adaptativas nos estados fora do caminho de referência. Tal possibilidade implicaria em um tratamento muito mais complexo das possibilidades de combinação de erros e transições de recuperação.

Uma limitação importante deste tipo autômato é percebida em situações em que as transições de recuperação de erro são conflitantes. No exemplo apresentado, as transições de recuperação de erro são independentes, por serem disparadas com símbolos diferentes. Porém, em alguns casos, quando as melodias possuem notas repetidas, transições de recuperação diferentes podem ser disparadas com um mesmo símbolo. Nestes casos, para evitar o não-determinismo do autômato, elimina-se a transição de menor prioridade. Os casos de erro em ordem decrescente de prioridade são: adição, troca e omissão.

Durante o reconhecimento de uma cadeia, o autômato registra um código de resultado para cada nota lida, em uma lista. Quando a nota recebida era esperada, o algoritmo registra `OK`. Quando a nota não era esperada o algoritmo registra o código da situação de erro ocorrida: `EXCHANGE`, `ADDITION` ou `OMISSION`. Note que neste caso o resultado depende da transição de recuperação que for utilizada e só pode ser determinado ao tratar a nota seguinte.

Quando o autômato atinge o estado final, a cadeia é considerada aceita e a lista de código de resultado representa os detalhes do reconhecimento. Caso contrário, a cadeia é

rejeitada, porém, para se saber até que ponto a cadeia foi reconhecida retorna-se a lista de código de resultado parcial. Esta lista contém os resultados até o momento em que a cadeia foi rejeitada.

Abaixo segue uma simulação de execução do autômato da figura 14, para cadeias de entrada contendo cada uma das três situações de erro descritas há pouco. Primeiramente mostra-se a cadeia utilizada para construção do autômato. Esta cadeia representa a consulta ao sistema de busca. Em seguida, entra-se com cadeias para serem processadas pelo autômato, representando trechos de melodias da base de dados. Para cada entrada o autômato retorna a lista de códigos de resultado.

Listagem 1 Resumo da simulação de execução do autômato

```
Cadeia para produzir o autômato:
69 71 73 74 76 77 76 0

Exemplo de troca:
69 71 74 74 76 77 76 0
[OK, OK, EXCHANGE, OK, OK, OK, OK]

Exemplo de adição:
69 71 74 76 77 76 0
[OK, OK, ADDITION, OK, OK, OK]

Exemplo de omissão:
69 71 72 73 74 76 77 76 0
[OK, OK, OMISSION, OK, OK, OK, OK, OK]
```

Uma cadeia ser aceita pelo autômato significa na prática que contornando eventuais situações de erro devidamente isoladas as cadeias são semelhantes. A partir deste momento a lista de códigos de resultados é analisada com o objetivo de mensurar a distância entre as cadeias comparadas.

Para esta análise, dois aspectos são considerados. O primeiro deles é direto e corresponde à quantidade de erros observados. Quanto menor for o número de erros, mais próximas serão as cadeias. O segundo aspecto envolve um procedimento mais elaborado. Note que a partir da cadeia que originou o autômato e da lista de códigos de resultados, é possível construir uma cadeia artificial, corrigindo os erros registrados. Assim se, por exemplo, uma nota for omitida, pode-se adicioná-la de volta à cadeia. O objetivo desta reconstrução é possibilitar a aplicação dos métodos numéricos de comparação nota a nota apresentados anteriormente. Principalmente com relação às durações das notas, aspecto que fora desconsiderado nesta nova abordagem até então.

Listagem 2 Saída completa da simulação de execução do autômato

```
Entre com uma cadeia terminada por 0 para gerar o autômato:
69 71 73 74 76 77 76 0
Digite:
1, para digitar uma cadeia de entrada.
2, para gerar uma cadeia de entrada.
0, para sair.
1
Entre com a cadeia de entrada terminando com 0:
69 71 74 74 76 77 76 0
true
[OK, OK, EXCHANGE, OK, OK, OK, OK]
Digite:
1, para digitar uma cadeia de entrada.
2, para gerar uma cadeia de entrada.
0, para sair.
1
Entre com a cadeia de entrada terminando com 0:
69 71 74 76 77 76 0
true
[OK, OK, ADDITION, OK, OK, OK]
Digite:
1, para digitar uma cadeia de entrada.
2, para gerar uma cadeia de entrada.
0, para sair.
1
Entre com a cadeia de entrada terminando com 0:
69 71 72 73 74 76 77 76 0
true
[OK, OK, OMISSION, OK, OK, OK, OK, OK]
Digite:
1, para digitar uma cadeia de entrada.
2, para gerar uma cadeia de entrada.
0, para sair.
```

Algumas considerações são importantes no que tange à reconstrução da cadeia com base nas informações de erros. Para notas trocadas deve-se manter a duração original. Para notas omitidas, estas deverão ser adicionadas de volta com duração zero, para que não haja interferência no contorno temporal geral da cadeia. A nota é adicionada somente para alinhar o emparelhamento necessário para a comparação nota a nota. E, por fim, para o caso de notas adicionadas, estas são retiradas e sua duração é incorporada à nota imediatamente anterior.

Com base nestas considerações constrói-se a cadeia corrigida e, executando o procedimento de comparação de durações, consegue-se obter uma nova métrica de distância entre as melodias, desta vez utilizando métodos numéricos e considerando as durações das notas.

5.7.3 Critério para avaliar a semelhança entre melodias

Com o modelo de comparação que foi definido, tem-se algumas métricas para avaliação da semelhança entre melodias. A primeira delas é a aceitação ou não da cadeia, isto é, se o autômato chegou ou não ao estado final. Outra métrica é representada pela lista de códigos de resultado e, por fim, a distância numérica das durações.

Existe uma relação de importância que cada um destes fatores tem sobre a semelhança global percebida. Porém, a falta de testes massivos impede uma percepção apurada destas importâncias e por consequência impossibilita a atribuição de pesos para cada fator.

Porém, faz-se necessário definir um critério simples e direto com o objetivo de classificar as correspondências encontradas. Adotou-se para este fim a contagem dos erros a partir da lista de códigos de resultado. Para as listas parciais (quando a cadeia não é aceita), a contagem de erros é somada com o número de notas que sobraram no momento em que o reconhecimento parou (número de notas corretas que faltaram para que o estado final fosse alcançado) e com a constante 100, para priorizar as situações em que a cadeia é aceita.

5.7.4 Busca

Sumarizando o procedimento de busca, temos inicialmente a construção do autômato adaptativo a partir da melodia de entrada quantizada. Em seguida aplica-se o autômato sobre todas as melodias do repositório. Este é aplicado inicialmente à cadeia que corresponde a uma melodia completa. Em seguida, a primeira nota desta cadeia é eliminada

e o autômato é aplicado novamente. E assim sucessivamente até que a cadeia torne-se vazia, então parte-se para a próxima melodia do repositório.

Ao longo deste processo, o mecanismo de busca mantém um conjunto com as N melhores correspondências que encontrou, com base no critério definido há pouco. Desta forma, ao final da varredura de todo o repositório, tem-se os resultados da busca.

6 Resultados

Este capítulo irá apresentar os experimentos e testes realizados sobre os componentes construídos, seus procedimentos, objetivos e resultados obtidos.

6.1 Rotinas para manipulação de melodias e arquivos MIDI

Para poder testar e realizar experimentos com os diversos componentes que fazem tratamento, comparação ou busca de melodias de notas, preparou-se um conjunto de rotinas capazes de manipular melodias de notas. Sempre utilizando a representação de notas discutidas na seção 5.3. São elas:

1. Importação de melodias a partir de trilhas de arquivos MIDI

   Rotina capaz de ler arquivos no formado Standard MIDI File, e extrair a cadeia de notas de uma das trilhas deste.

2. Exportação de melodias para arquivos MIDI

   Rotina que possibilita a exportação de uma cadeia de notas do formato interno para um arquivo MIDI.

3. Cálculo da taxa de polifonia de uma trilha MIDI

   Esta taxa significa a razão entre a soma dos intervalos em que mais de uma nota está ressonando pela soma dos intervalo em que há pelo menos um nota ressonando. Uma baixa taxa de polifonia é um indício de que a trilha é melódica.

4. Extração de sub-cadeia de uma cadeia

   Rotina simples que isola um trecho de uma cadeia, para viabilizar a realização de experimentos.

5. Geração de perturbações

   Esta rotina introduz perturbações parametrizáveis nas alturas e durações da cadeia. Os parâmetros são:

   Variação fixa de altura Um valor em semi-tons que será adicionado às alturas da cadeia (transposição tonal fixa).

   Variação aleatória de altura O valor do desvio-padrão em semi-tons de uma variação aleatória de altura a ser introduzida em cada nota.

   Dilatação/contração fixa de durações Um fator que multiplicará todas as durações da cadeia.

   Variação aleatória de durações O valor do desvio-padrão de uma variação aleatória de duração a ser introduzida em cada nota.

6. Geração de falhas

   Rotina que introduz falhas do tipo retirada ou adição de nota, ao longo da cadeia, de forma aleatória.

6.2 Repertório musical e construção do repositório

Com o fim de testar e avaliar os componentes do protótipo construído, preparou-se um repertório musical com vinte músicas populares. Conforme descrito anteriormente a construção do repositório de dados é fortemente baseada em arquivos MIDI, pois estes contém as melodias das músicas em um formato de fácil tratamento. Por isso, a escolha deste repertório embasou-se na disponibilidade das músicas neste formato.

A partir dos arquivos MIDI destas músicas foram extraídas as trilhas com mais probabilidade de serem melódicas, utilizando a rotina de cálculo da taxa de polifonia descrita acima. Manualmente, selecionou-se as trilhas que de fato continham as melodias características das músicas. Em geral, para músicas populares estas trilhas representam a linha do cantor ou de algum instrumento monofônico, como um saxofone.

Uma rotina de inicialização do repositório foi então preparada. Esta extraí as melodias das trilhas selecionadas de cada arquivo MIDI, e carrega uma estrutura de dados em memória relacionando estas melodias aos metadados das músicas (i.e. título e nome do compositor).

6.3 Processo de extração de notas

O mecanismo implementado para extração de notas foi submetido a diversos testes. O procedimento adotado para estes é o seguinte:

1. Gravação de amostras de áudio

   Foram escolhidos dezenove trechos característicos das músicas do repertório, de três a oito segundos de duração. Estes trechos foram assobiados por um executor. Para captura do áudio utilizou-se um microfone comum de PC e um software simples de gravação. Estas amostras foram armazenadas em arquivos no formato WAVE[1].

2. Extração de notas

   O arquivo WAV é utilizado como entrada para o procedimento de extração de notas descrito no item 5.2.

3. Avaliação

   Com base no arquivo MIDI audível produzido, avalia-se a semelhança com o material de áudio original.

O objetivo de tal procedimento de testes é verificar a qualidade do processo de extração. Uma vez que o objetivo conceitual da construção de tal processo é avaliar a possibilidade prática de se utilizar amostras audíveis como entrada para sistemas de busca musical, a qualidade deste processo está intimamente relacionada com a possibilidade de se identificar padrões musicais em seus resultados. Ou seja, está relacionado com a possibilidade de se identificar a música em que foi baseada a reprodução original, a partir da audição do material audível produzido pela extração.

Para avaliar esta capacidade, realizamos um teste auditivo com um voluntário com habilidades musicais amadoras. Este voluntário conhecia todas as músicas do repositório e, ouvindo os arquivos MIDI produzidos foi solicitado a identificar a qual daquelas musicas o trecho lhe parecia mais semelhante.

Em seguida, para aquelas musicas as quais o voluntário não soube identificar ou identificou incorretamente, foram apresentadas as amostras de áudio originais com a gravação do assobio. Solicitou-se novamente que o voluntário tentasse identificar as músicas.

A tabela 2 sumariza os resultados deste teste.

[1]Mais detalhes sobre este formato em: http://ccrma.stanford.edu/CCRMA/Courses/422/projects/WaveFormat/

Tabela 2: Resultados do teste auditivo

Situação	Número de ocorrências
Música corretamente identificada	15
Música identificada ao ouvir o assobio	1
Música não identificada	3

A partir destes resultados conclui-se que a qualidade do procedimento de extração está aceitável.

6.4 Comparação numérica

A comparação numérica foi testada utilizando principalmente a rotina que gera perturbações nas melodias. A partir de uma dada cadeia de notas de referência, gerou-se um conjunto de variações com a introdução de perturbações de intensidade variável. Executou-se o algoritmo de comparação numérica tanto para alturas como para durações, e calculou-se a média dos valores de distância obtidos.

O gráfico da figura 17 nos mostra uma comparação de durações nota a nota graficamente. Neste gráfico é ilustrada uma comparação entre uma cadeia do acervo e uma cadeia de entrada extraída de uma melodia assobiada. O emparelhamento necessário para este método, torna possível a plotagem da distribuição de notas em um plano como o ilustrado. No eixo das abscissas, tem-se a alturas da cadeia do acervo e no eixo das ordenadas as alturas da cadeia de entrada. O algoritmo é análogo ao processo de encontrar a reta neste plano que minimiza um certo critério de erro. Uma vez encontrada esta reta, a idéia de distância está associada à distância total dos pontos a esta reta de aproximação.

6.5 Quantização das alturas

Como foi detalhado no capítulo 5, elaborou-se um algoritmo para quantizar os valores das alturas das notas, minimizando de forma absoluta o erro de quantização. Para ilustrar este processo, gerou-se uma cadeia com perturbações aleatórias a partir de uma cadeia de referência. Os parâmetros para esta geração foram de $1,5$ semitom para variação fixa e $0,3$ para o desvio-padrão da variação aleatória. Em seguida aplicou-se o processo de quantização absoluta sobre esta.

O gráfico da figura 18 mostra as alturas da cadeia em cada uma das etapas deste procedimento. A escala do eixo das ordenadas está na notação MIDI e estão destacados

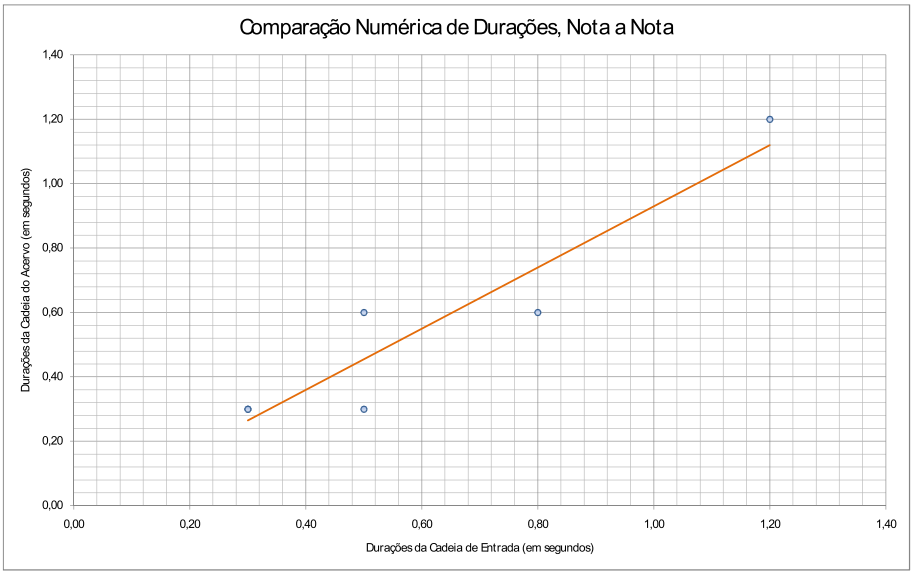

Figura 17: Emparelhamento de durações na comparação nota a nota

os níveis quantizados. É possível observar que somente três notas "erradas" foram geradas.

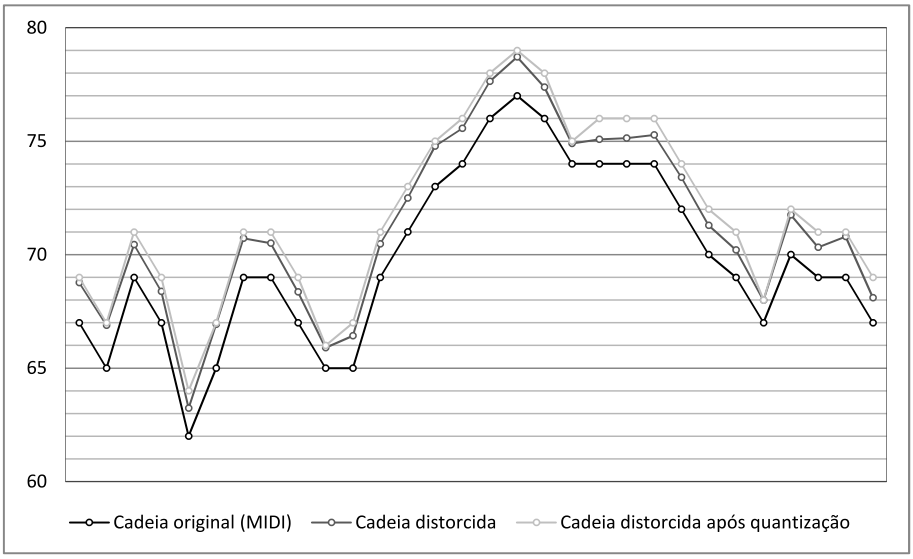

Figura 18: Quantização absoluta aplicada a uma melodia com perturbações aleatórias

Para comparar a eficiência do método absoluto proposto com relação ao método relativo, realizou-se um experimento envolvendo 21 melodias extraídas de gravações de assobios. Os dois métodos foram aplicados a estas melodias e observou-se a qualidade dos resultados produzidos com relação às melodias de referência. A tabela 3 mostra os resultados deste experimento.

Tabela 3: Comparação dos métodos de quantização

Método com bons resultados	Número de casos
Ambos	11
Absoluto	5
Relativo	1
Nenhum	4

6.6 Comparação com autômato adaptativo

Para testar a capacidade de reconhecimento inexato do autômato adaptativo, escolheu-se uma cadeia de referência e a partir dela foram geradas dez mil melodias com perturbações de altura. Aplicou-se, então, o autômato sobre este conjunto e obteve-se um percentual médio de aceitação, ou seja, uma representação da efetividade do autômato. Repetiu-se este procedimento para diversos valores do parâmetro de variação aleatória de altura, e com os resultados obtidos, compilou-se a tabela 4.

Tabela 4: Efetividade do autômato com perturbações

Perturbação	Efetividade (%)
0,00	100,0
0,05	100,0
0,10	100,0
0,15	99,8
0,20	97,4
0,25	87,2
0,30	71,8
0,35	55,0
0,40	42,4
0,45	32,6
0,50	24,2
0,55	18,3
0,60	14,6
0,65	10,5
0,70	8,5
0,75	6,5
0,80	5,7
0,85	4,1
0,90	3,2
0,95	2,9

6.7 Comparação de melodias e busca

Os componentes relacionados a comparação de melodias e busca foram testados utilizando as mesmas amostras de áudio preparadas para o item 6.3. As representações tabulares de notas produzidas para cada amostra durante os testes do processo de extração foram armazenadas em arquivos de texto.

Preparou-se então uma rotina de testes que inicializa o repositório, lê cada um dos arquivos de texto, constrói a melodia a partir da tabela e chama a rotina de busca, que por sua vez realiza a busca no repositório.

A saída da rotina de busca mostra uma lista contendo a melhor correspondência encontrada para cada música do repositório, ordenada pela semelhança (calculada segundo os critérios do item 5.7.3). São mostrados também a posição na melodia da música onde ocorreu a correspondência (*start*), a contagem de erros (*err*), a lista de códigos de resultado e a distância numérica das durações das notas das melodias (*distance*).

A listagem a seguir mostra o início da saída para a melodia extraída de um assobio da música *Yesterday* (*The Beatles*).

```
Yesterday
start = 22
err = 0
[OK, OK, OK, OK, OK, OK, OK]
distance = 0.012336956889409096

Chega De Saudade
start = 162
err = 1
[OK, OK, EXCHANGE, OK, OK, OK, OK]
distance = 0.6443477972823642

Palco
start = 86
err = 2
[OK, OK, OK, EXCHANGE, OK, OK, EXCHANGE]
distance = 4.645081298991189
```

```
Ode To Joy
start = 118
err = 2
[OK, OMISSION, OK, OK, EXCHANGE, OK, OK, OK]
distance = 0.17238503370165567

Aquarela
start = 311
err = 2
[OK, EXCHANGE, OK, ADDITION, OK, OK]
distance = 0.058781102023442666

Money For Nothing
start = 221
err = 2
[OK, OK, OK, ADDITION, OK, EXCHANGE]
distance = 77.8322420290857

Hino Nacional Brasileiro
start = 12
err = 3
[OK, EXCHANGE, OK, ADDITION, OK, EXCHANGE]
distance = 0.015761535349990998

We Are The Champions
start = 9
err = 3
[OK, EXCHANGE, OK, OK, EXCHANGE, OK, EXCHANGE]
distance = 0.6035584560535664
```

Analisando todas as listagens deste teste, pode-se levantar uma tabela ilustrando a taxa de acerto da busca. Para cada caso, observou-se em que posição da listagem aparece a música na qual foi baseado o assobio. A figura 19 mostra a quantidade de casos em que a busca classificou a música correta em primeiro lugar, em segundo lugar e assim por

diante.

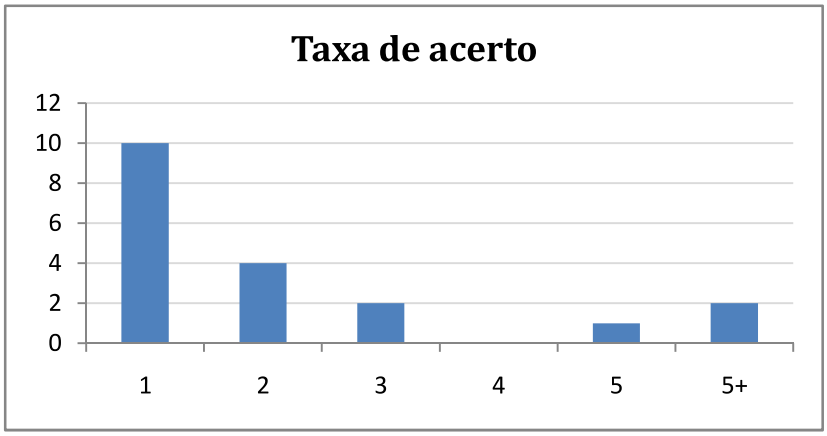

Figura 19: Análise de acerto da busca

7 Análise e crítica

Neste capítulo é feito um breve comentário de duas limitações observadas nos componentes construídos para este trabalho.

7.1 Quantização de notas

Ao analisar os resultados dos dois processos de quantização implementados, observou-se que nenhum dos dois oferece uma qualidade ótima. Ambos tentam lidar com a desafinação do reprodutor, porém as abordagens dadas por estes fazem considerações extremas sobre esta questão. O que sugere que uma abordagem híbrida tenha potencial de apresentar melhores resultados.

O método relativo, por ser baseado na referência apenas da nota anterior, muitas vezes produz melodias com muita divergência com relação a original. Este método acaba por considerar uma oscilação de afinação muito maior do que a observada na reprodução vocal humana, mesmo por amadores.

Analisando reproduções vocais humanas de melodias, observa-se a existência de uma variação aleatória de afinação em cada uma das nota individualmente, proveniente da imprecisão do reprodutor. Por outro lado existe também uma variação progressiva da referência absoluta de afinação, que ocorre de maneira bem mais lenta.

O método de quantização absoluto parte da premissa de que esta variação de referência absoluta não existe. Apesar disso este método apresentou bons resultados na maioria dos casos. Isso pode ser justificado pela duraçao reduzida dos casos testados, pois este efeito tende a aparecer mais significativamente em melodias mais longas. Porém, isto representa uma limitação importante deste método comprovada na prática por alguns casos.

Um método híbrido pode ser idealizado de diversas maneiras. Uma possibilidade é aplicar o método absoluto em trechos de tamanho fixo ao longo da melodia, e utilizar as idéias do método relativo para corrigir a variação da referência absoluta entre estes

trechos. Ao aplicar o método absoluto sobre o primeiro trecho obtem-se um offset de correção (entre $-0,5$ e $0,5$ semitom) para este trecho. Este valor é somado no restante da melodia para corrigir a variação de referência. Em seguida o processo se repete para os demais trechos.

Dada esta idéia geral de como seria o método de quantização híbrido, pode-se pensar nos dois métodos implementados como sendo casos particulares deste primeiro. O método relativo é o caso particular em o tamanho do trecho é 1, e o método absoluto é o caso em o tamanho do trecho coincide com o tamanho da melodia inteira.

7.2 Autômato adaptativo

Como foi visto, o autômato adaptativo proposto representa um dispositivo reconhecedor de melodias semelhantes àquela que o gerou. Da maneira que foi construído, este dispositivo é capaz de tolerar diversos tipos de variações, inclusive múltiplas, para aceitar melodias. Porém, este sistema de tolerância depende do acerto da primeira nota da melodia que o originou.

Isto ocorre por que o processo utiliza a primeira nota para calcular a transposição tonal e corrigí-las nas demais notas. Por isso a primeira nota é sempre considerada correta. Se eventualmente esta nota estiver errada (caso de troca) com relação ao restante das notas, o autômato gerado não seria capaz de reconhecer a melodia correta, isto é, aquela sem a troca da primeira nota. Isto porque a primeira nota seria considerada correta e todas as demais erradas.

Esta limitação não é trivialmente contornável. Pensou-se em utilizar os intervalos (diferenças) entre as notas como sendo os símbolos do autômato, ao invés das próprias notas. Mas esta representação não resolve o problema pois o autômato não conseguiria decidir entre um caso de troca ou de adição sem calcular a soma dos intervalos, o que seria equivalente ao que é feito atualmente.

Felizmente a maioria das reproduções não apresenta erro logo na primeira nota. E a utilização de um método de quantização híbrido evitaria ainda mais a possibilidade de ocorrência destes casos.

8 Melhorias e Trabalhos futuros

Desde o início a proposta do trabalho foi desenvolver uma técnica de comparação de conteúdos musicais utilizando técnicas adaptativas, associada a um protótipo funcional que fosse capaz de exercitar e avaliar a técnica de comparação desenvolvida. Assim, diversos pontos do trabalho foram abordados de forma superficial e são passíveis de grandes melhorias. A seguir serão apresentados alguns dos pontos de extensão mais importantes.

Automatização da identificação de notas. Parte deste processo foi executada de forma manual e é passível de automatização. Para tanto a análise das variações de intensidade e frequência ao longo da linha de tempo podem ser extremamente úteis. Pode-se utilizar como base trabalhos já desenvolvidos, a fim de melhorar este componente do protótipo.

Maior granularidade de erros. Apenas três tipos de erros foram considerados para este trabalho (*OMISSÃO*, *ADIÇÃO* e *TROCA* de uma nota). Porém, ao longo do desenvolvimento percebeu-se que com apenas estas classes de erro não é possível avaliar com precisão a intensidade do desvio entre as melodias comparadas. Isto ocorre pelo fato de que o caso de erro *TROCA* permite a substituição por qualquer outra nota. Assim sugere-se a criação de uma nova classe de erro, denominada *VARIAÇÃO MENOR*, similar a uma *TROCA* porém representando um desvio menor de nota, dentro de uma tolerância pré-determinada. Esta nova classe tem o objetivo de descrever um erro muito comum, em que há um pequeno desvio com relação à nota original.

Múltiplos erros. O autômato desenvolvido é capaz de lidar com erros simples apenas (dentro da definição proposta). Mais de um erro pode ser aceito desde que haja um espaçamento mínimo entre estes. Porém é relativamente comum, em especial para o caso de *TROCA*, que ocorram dois erros seguidos. O projeto atual do autômato não é capaz de lidar com esta situação, gerando *scores* de proximidade muito mais baixos do que o

esperado para trocas seguidas.

Definição de pesos de cada classe de erro. Para o cálculo da similaridade, diversos fatores devem considerados, como:

- Número de erros
- Tipos de erros cometidos
- Aceitação ou não da cadeia (atingir o fim do autômato)

Tais fatores influenciam diferentemente na similaridade entre dois trechos. A omissão de uma nota, por exemplo, é intuitivamente um erro mais grave do que uma pequena troca de nota (*VARIAÇÃO MENOR*). Porém, não é feita distinção entre estes dois casos. Testes com repositórios maiores e associados a um *feedaback* humano poderia prover dados que ajudassem a calibração dos pesos de cada um destes fatores no *score* final.

Generalização do método de comparação. O método comparação descrito independe do fato de a cadeia de entrada ser proveniente especificamente de uma sequência de notas. Assim, este método pode ser generalizado para lidar com cadeias de símbolos de qualquer natureza. Seria necessário generalizar todos os conceitos envolvidos, como alfabeto de entrada, classes de erro aceitáveis, etc. Este método de comparação seria apropriado para busca padrões em cadeias, acrescentando uma flexibilização maior do que uma busca exata ou uma expressão regular.

9 Contribuições

O protótipo desenvolvido ao longo deste trabalho tocou diversas áreas de pesquisa, desde a análise de sinal, até a adaptatividade. Em sua maioria, foram utilizadas técnicas previamente criadas e desenvolvidas em trabalhos anteriores, porém o foco do trabalho:

Uso de autômatos adaptativos para reconhecimento de padrões musicais

apresentou uma abordagem diferente para um tema de grande importância: o estabelecimento de um algoritmo de similaridade entre conteúdos de áudio.

A técnica descrita passa a fazer parte de um leque de possibilidades para utilização dentro de um sistema de busca de áudio por conteúdo mais complexo. O método descrito traz a aplicação de uma técnica adaptativa como ferramenta de reconhecimento de padrões melódicos e avaliação de similaridade. Apesar de ainda haver a necessidade da determinação dos pesos ideiais para se atingir um critério adequado de cálculo de similaridade, o trabalho provê a base para tal utilização.

Técnicas adaptativas já foram usadas em diversos trabalhos com o fim de reconhecer padrões, porém pouco utilizadas no domínio de áudio, esse trabalho apresenta uma possível aplicação de uma técnica adaptativa neste contexto, no caso, um autômato adaptativo, mostrando que pode-se obter bons resultados.

Referências

BRACEWELL, R. N. *Fourier Transform and Its Applications*. [S.l.]: McGraw-Hill Education, 1980. Hardcover. ISBN 0070661960.

BRIGHAM, E. O. *The fast Fourier transform and its applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988. ISBN 0-13-307505-2.

BRøNDSTED, T. et al. A system for recognition of hummed tunes. In: *COST G-6 Conference on Digital Audio Effects*. [S.l.: s.n.], 2001.

BURNS, E. M. *"Intervals, Scales, and Tuning", The Psychology of Music*. 1999.

FOOTE, J. An overview of audio information retrieval. *ACM Multimedia Systems*, v. 7, p. 2–10, 1999.

GERHARD, D. *Pitch Extraction and Fundamental Frequency: History and Current Techniques*. [S.l.], 2003.

GHIAS, A. et al. Query by humming: Musical information retrieval in an audio database. In: *ACM Multimedia*. [S.l.: s.n.], 1995. p. 231–236.

HUMES I.S.H. DE MELO, L. Y. W. M. A.F.P. de C. *Noções de Cálculo numérico*. São Paulo: McGraw Hill, 1984.

JOHNSONBAUGH, R. *Discrete Mathematics*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000. ISBN 0130890081.

LANE, J. E. Pitch detection using a tunable iir filter. In: *Computer Music Journal*. [S.l.: s.n.], 1990. v. 14, p. 46–57.

MCNAB LLOYD A. SMITH, I. H. W. C. L. H. S. J. C. R. J. Towards the digital music library: tune retrieval from acoustic input. In: *DL '96: Proceedings of the first ACM international conference on Digital libraries*. New York, NY, USA: ACM, 1996. p. 11–18. ISBN 0-89791-830-4.

MIDI Manufacturers Association. `http://www.midi.org/`.

MIDOMI. `http://www.midomi.com/`.

MOORER, J. A. On the transcription of musical sound by computer. In: *Computer Music Journal*. [S.l.: s.n.], 1977. v. 3, p. 32–38.

NETO, J. J. Adaptive automata for context-sensitive languages. *SIGPLAN NOTICES*, v. 29, September 1994.

NETO, P. D. J. J. *Roteiro de Estudos de Tecnologia Adaptativa*. 2004. `http://www.pcs.usp.br/~lta/roteiro_estudo/`.

OPPENHEIM, A. V.; SCHAFER, R. W.; BUCK, J. R. *Discrete-time signal processing (2nd ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1999. 713–718 p. ISBN 0-13-754920-2.

PARSONS, D. Book. *The directory of tunes and musical themes*. [S.l.]: S. Brown, Cambridge, Eng. :, 1975. 288 p. : p. ISBN 090474700.

PISZCZALSKI, M. *A computational model of music transcription*. Tese (Doutorado) — University of Stanford, Ann Arbor, MI, USA, 1986.

Piszczalski, M.; Galler, B. A. Predicting musical pitch from component frequency ratios. *Acoustical Society of America Journal*, v. 66, p. 710–720, set. 1979.

SAUNDERS, J. Real-time discrimination of broadcast speech/music. In: *ICASSP '96: Proceedings of the Acoustics, Speech, and Signal Processing, 1996. on Conference Proceedings., 1996 IEEE International Conference*. Washington, DC, USA: IEEE Computer Society, 1996. p. 993–996. ISBN 0-7803-3192-3.

SPINA, M. S.; ZUE, V. Automatic transcription of general audio data: Preliminary analyses. In: *Proc. ICSLP '96*. Philadelphia, PA: [s.n.], 1996. v. 2, p. 594–597.

THE Open Music Encyclopedia. `http://www.musipedia.org/`.

WALL, A. *History of Search Engines: From 1945 to Google 2007*. `http://www.searchenginehistory.com/`.

WIERING, F. Can humans benefit from music information retrieval? In: *Adaptive Multimedia Retrieval*. [S.l.: s.n.], 2006. p. 82–94.

WOLD, E. et al. Content-based classification, search, and retrieval of audio. *IEEE MultiMedia*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 3, n. 3, p. 27–36, 1996. ISSN 1070-986X.