COPPE/UFRJ

MÉTODOS DE FATORAÇÃO DE MATRIZES NÃO-NEGATIVAS PARA SEPARAÇÃO DE SINAIS MUSICAIS

Alan Freihof Tygel

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Luiz Wagner Pereira Biscainho

Rio de Janeiro

Dezembro de 2009

MÉTODOS DE FATORAÇÃO DE MATRIZES NÃO-NEGATIVAS PARA SEPARAÇÃO DE SINAIS MUSICAIS

Alan Freihof Tygel

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Luiz Wagner Pereira Biscainho, D.Sc.

Prof. José Manoel de Seixas, D.Sc.

Prof. Marcio Nogueira de Souza, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

DEZEMBRO DE 2009

Tygel, Alan Freihof

Métodos de fatoração de matrizes não-negativas para separação de sinais musicais/Alan Freihof Tygel. – Rio de Janeiro: UFRJ/COPPE, 2009.

XIII, 109 p.: il.; 29, 7cm.

Orientador: Luiz Wagner Pereira Biscainho

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2009.

Referências Bibliográficas: p. 103 – 109.

1. Fatoração de Matrizes Não-Negativas. 2. Processamento de Sinais de Áudio. 3. Separação de Sinais de Áudio. 4. Processamento de Sinais. I. Biscainho, Luiz Wagner Pereira *et al.*. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Agradecimentos

Agradeço ao povo brasileiro, que através dos impostos financiou os dois anos do curso de mestrado e os dois anos da bolsa de mestrado pela Capes.

Ao meus pais, pelo apoio incondicional a todas as minhas escolhas, por mais malucas que possam parecer.

Aos músicos do Brasil, por proporcionarem momentos de prazer durante a elaboração desta dissertação. Em particular, a dois compositores, João do Vale e Gordurinha, cujo trabalho tive o prazer de conhecer ao longo do mestrado. A estes dois, que na minha opinião não receberam o reconhecimento devido, um obrigado especial.

Ao meus colegas de LPS, sempre dispostos a ajudar quando preciso e a desconcentrar quando preciso. Sem citar nomes para não correr o risco de esquecer alguém, todos sabem a parcela de contribuição que tiveram.

Ao projeto CoppeTEX, que tornou a elaboração deste documento muito simples.

E finalmente, ao meu orientador Luiz Wagner, pelo parceria ao longo destes anos e, com certeza, pela amizade ao longo dos próximos.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

MÉTODOS DE FATORAÇÃO DE MATRIZES NÃO-NEGATIVAS PARA SEPARAÇÃO DE SINAIS MUSICAIS

Alan Freihof Tygel

Dezembro/2009

Orientador: Luiz Wagner Pereira Biscainho

Programa: Engenharia Elétrica

Os métodos de Fatoração de Matrizes Não-Negativas (NMF, do inglês *Non-Negative Matrix Factorization*) têm sido alvo de pesquisa em diversas aplicações de processamento de sinais desde o final dos anos 1990. Para análise de sinais musicais, a fatoração faz sentido quando desejamos decompor um espectrograma, não-negativo por definição, em matrizes que representam bases espectrais e ganhos não-negativos associados a elas.

Esta dissertação apresenta um sistema completo de separação de sinais musicais utilizando métodos de fatoração de matrizes não-negativas. O sistema parte de uma mistura, ou seja, um sinal musical composto por diversos instrumentos, e após as etapas de análise tempo-frequência, fatoração, processamento e síntese, gera os sinais musicais correspondentes aos instrumentos separados.

São apresentadas três contribuições na etapa de fatoração, todas elas no sentido de tornar o modelo mais realístico e eficiente. Além disso, uma proposta de método de avaliação de qualidade de separação também é apresentada.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

NON-NEGATIVE MATRIX FACTORIZATION METHODS FOR SEPARATION OF MUSICAL SIGNALS

Alan Freihof Tygel

December/2009

Advisor: Luiz Wagner Pereira Biscainho

Department: Electrical Engineering

Non-negative matrix factorization (NMF) methods have been target of research in many areas of signal processing since the end of 1990's. For the analysis of musical signals, the factorization is useful when one tries to decompose a spectrogram, non-negative by definition, in matrices that represent a spectral basis and their associated non-negative gains.

This work presents a complete system for separation of sound sources using non-negative matrix factorization methods. The system's input is a mixture, i.e., a musical signal composed by several instruments, and after passing through the stages of time-frequency analysis, factorization, processing and synthesis, generates the musical signals corresponding to the separated instruments.

Three contribuitions in the factorization step are presented, all of them in the sense of turning the model more realistic and efficient. Besides, a quality assessment method for evaluating the separation is presented.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

CQT	*Constant Q-Transform*, p. 14
DE	Distância Euclidiana, p. 24
DFT	*Discrete Fourier Transform*, p. 8
DKL	Divergência de Kullback-Leibler, p. 25
FFT	*Fast Fourier Transform*, p. 11
G&L	*Algoritmo de Griffin e Lim*, p. 61
ICA	*Independent Component Analysis*, p. 2
LNMF2D	*Linear Non-Negative Matrix Factor 2D Deconvolution*, p. 40
MISI	*Multiple Input Spectrogram Inversion*, p. 63
MSTFTM	*Modified Short-Time Fourier Transform Magnitude*, p. 60
MSTFT	*Modified Short-Time Fourier Transform*, p. 60
NMF2D	*Non-Negative Matrix Factor 2D Deconvolution*, p. 33
NMF	*Non-Negative Matrix Factorization*, p. 21
ODG	*Objective Degradation Grade*, p. 71
RTISI	*Real-time Iterative Spectrogram Inversion*, p. 62
SDR	*Source-to-Distortion Ratio*, p. 69
SNR	*Signal-to-Noise Ratio*, p. 67
STFT	*Short-Time Fourier Transform*, p. 10, 60

Capítulo 1

Introdução

A tarefa de separação de sinais é um tópico de pesquisa em franco desenvolvimento e que permeia diversas áreas do conhecimento, que vão desde a Engenharia até as Ciências da Saúde. A inspiração do problema, e ao mesmo tempo o indicativo de que sua solução é possível, vem do clássico problema da festa de coquetel (*Cocktail Party*, em inglês), proposto por [1] em 1954.

Em uma festa, com diversas pessoas falando ao mesmo tempo, o ser humano, utilizando as duas orelhas, consegue distinguir dentre todas as vozes aquela que lhe é de interesse; todas as outras são desconsideradas. Assim, deveria ser possível a um algoritmo fazer o mesmo.

1.1 Separação Automática de Sinais

A separação de sinais biológicos tem sido uma grande motivação para os desenvolvimentos na área de separação de sinais. Sinais captados por eletrodos no cérebro são provenientes de diversas fontes, como o piscar de olhos e o movimento da língua. O mesmo ocorre no peito, quando as batidas do coração de uma mulher grávida se misturam às do feto. Assim, eletrocardiogramas e eletroencefalogramas podem ser melhor interpretados quando se utilizam técnicas de separação de sinais [2] [3].

Na área das comunicações, podemos citar como exemplo as antenas que recebem sinais de diversas fontes e precisam separá-los. Num ambiente ruidoso, um telefone celular precisa separar a voz do usuário das demais para melhorar a qualidade da transmissão.

Frequentemente, o termo Separação Cega de Sinais (*Blind Source Separation*, BSS) é utilizado na literatura [4]. Isto significa que a solução em questão não assume nenhum modelo para o sinal, e nem para a maneira como ele foi misturado. Entretanto, na maioria das publicações, alguma informação sobre o sinal é utilizada. Um termo mais preciso para definir este tipo de problemas seria "Não-Supervisionado", quando a solução não utiliza amostras do sinal.

Esta dissertação aborda o problema da separação de sinais musicais, ou seja, partindo-se de uma mistura gravada de instrumentos musicais, deseja-se obter o sinal referente a cada instrumento separadamente. Assim, não se pode dizer que o problema é cego, já que a informação de que os sinais são musicais é utilizada. No entanto, pode-se dizer que a solução é não-supervisionada, pois não se utiliza informação do sinal a priori.

1.2 Análise de Componentes Independentes

Historicamente, o problema de separação de fontes foi tratado utilizando-se a técnica denominada Análise de Componentes Independentes (em inglês *Independent Component Analysis*, ICA) [5].

No contexto de ICA, a mistura de sinais é modelada da seguinte maneira:

$$\begin{aligned} y_1 &= a_{11}x_1 + a_{12}x_2 + \ldots + a_{1D}x_D \\ y_2 &= a_{21}x_1 + a_{22}x_2 + \ldots + a_{2D}x_D \\ &\vdots \\ y_P &= a_{P1}x_1 + a_{P2}x_2 + \ldots + a_{PD}x_D, \end{aligned}$$

onde y_p são os sinais captados pelos P sensores (microfones), x_d são as D fontes (instrumentos) e a_{pd} são os coeficientes de mistura.

A técnica é desenvolvida assumindo independência estatística entre as fontes. Assim, diversos algoritmos, como por exemplo [6, 7, 8], buscam otimizar algum tipo de medida de independência para obter os sinais mais independentes possíveis a partir de uma determinada mistura.

Uma destas medidas é a gaussianidade do sinal. O teorema do limite central diz

que a soma de duas variáveis aleatórias independentes tem sempre uma distribuição mais próxima de uma gaussiana do que a das variáveis originais. Portanto, se buscarmos os sinais menos gaussianos que combinados linearmente formam as misturas, podemos encontrar as fontes originais independentes.

O modelo básico de ICA lida apenas com misturas instantâneas (não-convolutivas) e lineares, além de considerar que $D = P$, isto é, que existem no mínimo tantos sensores quanto fontes. Entretanto, diversos algoritmos foram desenvolvidos para o tratamento de misturas não-lineares, convolutivas, ruidosas e em que $D \neq P$. Uma extensa introdução a todos estes métodos, incluindo o embasamento matemático necessário, pode ser encontrada em [5].

1.3 Fatoração de Matrizes Não-Negativas

Após uma certa estagnação nos avanços em ICA, iniciou-se grande desenvolvimento na separação baseada nos métodos de fatoração de matrizes não-negativas (*Non-Negative Matrix Factorization*, NMF)[1], principalmente após a publicação na Revista *Nature* do artigo *"Learning the parts of objects by non-negative matrix factorization"* [9], por Daniel D. Lee e H. Sebastian Seung, em 1999. Este artigo descreve a aplicação de NMF a problemas de processamento de imagens e *data-mining*.

Na aplicação a sinais musicais, uma grande vantagem dos algoritmos que utilizam NMF é que não resultam somente na separação das fontes; eles fornecem uma descrição de alto nível do sinal musical, que pode ser usada para transcrição (geração automática de partitura), edição de notas e timbre ou análise computacional da cena sonora (em inglês *Computer Auditory Scene Analysis*, CASA).

Outra diferença crucial entre os algoritmos de NMF e os de ICA é em relação ao número de sensores necessários. Enquanto os algoritmos de ICA se aproveitam da redundância presente em vários sensores que captam os sinais das fontes, os algoritmos de NMF utilizam a esparsidade tempo-frequencial presente em apenas um sensor. Algoritmos clássicos de ICA necessitam de no mínimo tantos sensores quanto

[1] A tradução exata do termo *Non-Negative Matrix Factorization* deveria ser "Fatoração de Matrizes Não-Negativas em Matrizes Não-Negativas", já que tanto a matriz fatorada quanto os fatores devem possuir todos os elementos maiores ou iguais a zero. Devido à falta de uma tradução consagrada em português, esta dissertação utilizará arbitrariamente o termo "Fatoração de Matrizes Não-Negativas".

fontes para seu funcionamento adequado; algoritmos de NMF podem funcionar com qualquer quantidade de sensores, inclusive com apenas um. Como os sinais musicais aparecem normalmente em um ou dois canais (mono/estéreo), os algoritmos de NMF também neste caso são mais adequados a gravações musicais.

Em [10], iniciou-se a aplicação do algoritmos de NMF descritos em [11] em separação de sinais musicais. Neste artigo, o objetivo é decompor um espectrograma em componentes de frequência que aparecem com uma certa intensidade ao longo dos quadros temporais. Para formar o sinal correspondente a cada instrumento, é preciso agrupar componentes. A NMF foi então estendida para a chamada *Non-Negative Matrix Factor Deconvolution* (NMFD) [12], em que os componentes têm estrutura temporal. Desta forma, cada componente toma a forma de uma nota de um instrumento. Ainda assim, é necessário agrupar todas as notas de um instrumento para gerar a fonte que corresponde ao instrumento. Um desenvolvimento posterior resultou na *Non-Negative Matrix Factor 2-D Deconvolution* (NMF2D) [13]. Este método permite que os padrões espectrais não só tenham estrutura temporal como possam ser deslocados ao longo das raias frequenciais, para formar diversas notas com um mesmo componente. A decomposição gera duas matrizes por instrumento: uma que representa o padrão espectral do instrumento, e outra que descreve em que ponto no tempo e na frequência este padrão aparece no espectrograma. Com isso, a NMF2D tem a vantagem de efetuar a separação em instrumentos sem a necessidade de agrupamento de componentes.

Esta dissertação inclui uma revisão dos métodos desenvolvidos até o presente momento, e detalha todas as fases envolvidas na construção de um sistema de separação de sinais baseado em NMF: desde a análise do sinal de áudio (construção do espectrograma), passando pela decomposição propriamente dita, até a síntese dos instrumentos separados e a avaliação de qualidade dos resultados.

1.4 Definição de Fonte Sonora

A separação de sinais musicais tem uma particularidade em relação aos outros tipos de aplicação: a definição de "fonte" não é precisa. O mais natural seria considerar como fonte um instrumento musical, o que no entanto gera várias contradições. Por

exemplo, o caso da bateria, que é um "instrumento" formado por vários instrumentos; ou ainda um naipe de violinos, onde vários instrumentos iguais tocam a mesma partitura, e soam aos ouvidos da platéia como um único "instrumento".

Outra possibilidade de definição seria considerar como fonte sonora cada elemento físico que gera vibrações. Desta forma, um violão seria dividido em 6 fontes, uma para cada corda. No entanto, o caso do piano, em que até três cordas são acionadas por um mesmo martelo ao mesmo tempo, ficaria mal definido. Afinal, as vibrações são geradas pelas cordas e pela interação entre elas, além da vibração do corpo do instrumento.

Uma definição mais plausível e menos controversa seria considerar como fonte aquilo que o ouvido entende como fonte; isso nos aproxima da definição do problema do *Cocktail Party*. No entanto, ainda assim é possível argumentar que a percepção de fonte depende do indivíduo; o maestro da orquestra consegue ouvir os violinos do naipe individualmente; o público em geral não.

Desta forma, ao longo desta dissertação será sempre esclarecido o que cada algoritmo considera como fonte.

1.5 *SoundFact*: Sistema de Separação de Fontes via Fatoração de Matrizes Não-Negativas

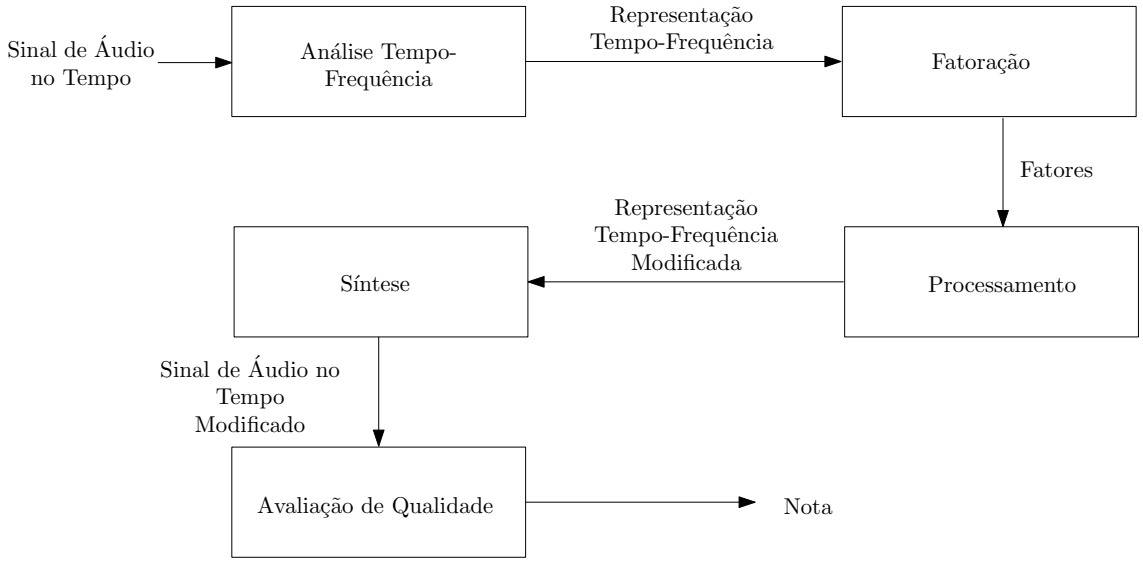


Figura 1.1: Diagrama de Blocos do Sistema. Cada bloco será tratado em um capítulo desta dissertação.

O diagrama de blocos da Figura 1.1 mostra os elementos básicos do sistema completo de separação de sinais de áudio utilizando fatoração de matrizes não-negativas. Esta dissertação tratará de todos os blocos da figura, nos capítulos que seguem.

O Capítulo 2 trata do bloco de análise tempo-frequência. Nesta etapa, ocorre a mudança de domínio: o sinal, originalmente no domínio do tempo, é transformado para o domínio tempo-frequência. Através de uma transformada de tempo curto, pequenos trechos do sinal (chamados de "quadros") são convertidos para o domínio da frequência. Desta forma, temos uma representação que alia uma quantização da dimensão temporal, através dos quadros, a uma quantização da dimensão frequencial, através das raias de frequência.

Os Capítulos 3 e 4 apresentam o cerne deste sistema. A fatoração é o tema central desta dissertação. No Capítulo 3 são apresentados os métodos de fatoração da literatura. Em seguida, no Capítulo 4, algumas contribuições desta dissertação ao assunto são propostas: o algoritmo LNMF2D, que permite que a fatoração seja aplicada a um espectrograma linearmente espaçado; a adaptação espectral, um refinamento que ajusta timbres de instrumentos entre diversas notas; e o algoritmo *online*, que possibilita o uso da fatoração em blocos, mantendo a ordenação e com baixo uso de memória.

O resultado da fatoração deve ser processado e sintetizado para que os sinais separados possam ser ouvidos. O Capítulo 5 lida com os processamentos que podem ser feitos após a fatoração, como uma espécie de ajuste do espectrograma. Os processamentos incluem a reconstrução e filtragem dos espectrogramas separados. Em seguida, o Capítulo 6 lida com a síntese, ou seja, a transformação do domínio tempo-frequencial para o domínio do tempo, quando temos finalmente o resultado audível da separação.

O Capítulo 7 aborda a questão da avaliação de qualidade. Apesar de não ser um bloco essencial para o funcionamento do sistema, é útil na medida em que desejemos comparar diferentes algoritmos e configurações de parâmetros. Neste capítulo, uma proposta de procedimento de avaliação de qualidade de separação utilizando métodos psicoacústicos é apresentada.

Finalmente, o Capítulo 8 mostra simulações do sistema apresentado, explorando

diversas configurações possíveis, que são comparadas através das métricas descritas no Capítulo 7. O Capítulo 9 tece as considerações finais sobre o trabalho, ressaltando as contribuições desta dissertação e sugerindo os caminhos a serem seguidos a partir dela.

1.5.1 *Software* Desenvolvido

Como produto desta dissertação, foi desenvolvido o *SoundFact*, um aplicativo em Matlab© que implementa todas etapas do sistema descrito acima. O pacote com os *scripts* pode ser obtido em `http://www.lps.ufrj.br/~alan/SoundFact`.

Todas as instruções de uso estão contidas no arquivo `READ-ME.txt`. A função principal chama-se `SoundFact_main`. Os argumentos de entrada especificarão todos os parâmetros e escolhas de algoritmos relativos aos blocos do sistema. Para conhecer as possibilidades de parâmetros, basta digitar:

```
>> help SoundFact_main
```

no console do Matlab©.

Cabe ressaltar que durante a realização deste trabalho somente foram encontrados elementos de *software* que implementavam algoritmos de fatoração de matrizes não-negativas. Nenhum deles, no entanto, realiza todas etapas de um sistema de separação de sinais musicais, desde o sinal de áudio misturado até as fontes separadas. O sistema *SoundFact*, portanto, vem preencher esta lacuna e facilitar a apropriação do conhecimento por novos pesquisadores.

A disponibilização do código-fonte visa a atender os conceitos da reprodutibilidade em artigos científicos. Em [14], os autores ressaltam a importância de se reportar experimentos de modo que eles possam ser reproduzidos por outros pesquisadores, seja para comprovar os resultados, seja para comparação entre métodos. Além de disponibilizar o código-fonte, procuramos ao longo do texto descrever de forma sistemática os algoritmos e seus parâmetros, para que uma possível continuação do trabalho seja facilitada.

Capítulo 2

Análise: representação tempo-frequência

Como podemos observar no diagrama de blocos da Figura 1.1, a primeira etapa de processamento do sistema consiste numa transformação do sinal de entrada do domínio temporal para o domínio tempo-frequencial. Neste capítulo serão tratados a necessidade de tal transformação e os diferentes métodos para realizá-la utilizados no trabalho. Veremos também a relação entre o tipo de transformada e a escala musical.

No próximo capítulo será visto que o ponto de partida dos algoritmos de fatoração em matrizes não-negativas é a representação do sinal de áudio como uma matriz não-negativa. Aqui veremos o processo de construção desta matriz não-negativa.

O modo mais intuitivo de se representar um sinal digital de áudio é no tempo. Como pode ser visto na Figura 2.1, o eixo das abscissas representa a contagem das amostras, que dividida pela taxa de amostragem nos fornece uma medida de tempo. No eixo das ordenadas, temos a amplitude de cada amostra.

O sinal musical mostrado na Figura 2.1 é composto por três notas (lá 220 Hz, lá 440 Hz e lá 880 Hz) tocadas em sequência. Este gráfico nos mostra com clareza o momento em que cada nota foi acionada. Entretanto, nada podemos inferir sobre quais notas estão presentes no sinal.

Outra maneira de se observar um sinal digital é no domínio da frequência. Neste caso, aplicamos uma transformada de Fourier discreta (DFT) [15], e temos o gráfico da Figura 2.2, que nos mostra nas abscissas a frequência central de cada raia, e nas

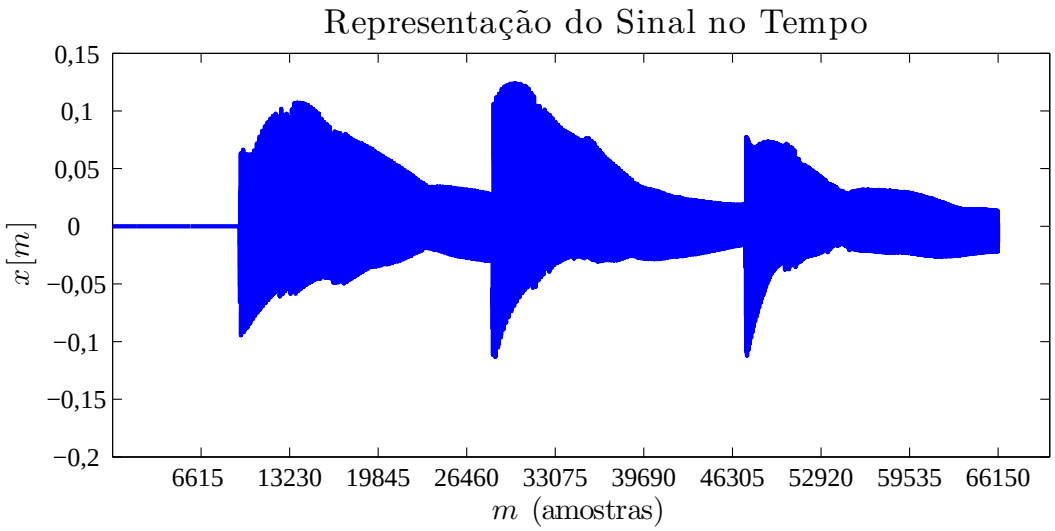


Figura 2.1: Representação no tempo de um sinal composto por 3 notas (lá 220 Hz, lá 440 Hz e lá 880 Hz) tocadas em sequência. A taxa de amostragem é de 44,1 kHz. O sinal é representado por $x[m]$, onde m são as amostras e $x[m]$ representa a amplitude de cada amostra.

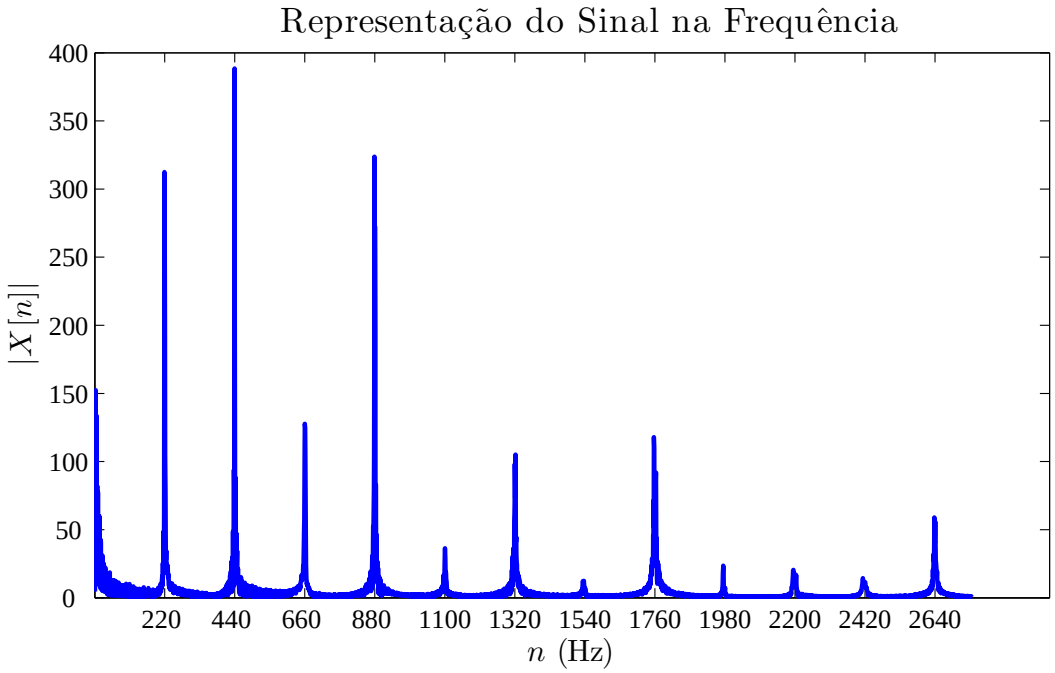


Figura 2.2: Trecho do espectro de magnitude de um sinal de áudio. O sinal é composto por 3 notas (lá 220 Hz, lá 440 Hz e lá 880 Hz) tocadas em sequência. O espectro é representado por $|X[n]|$, que é a magnitude da DFT de $x[m]$ e n é a frequência.

ordenadas a magnitude do sinal correspondente a cada raia. As raias representam intervalos lineares de frequência.

Neste gráfico enxergamos claramente as componentes de frequência presentes no

sinal: 220 Hz, 440 Hz, 880 Hz e seus harmônicos. Entretanto, não temos nenhuma informação sobre em que momento as componentes foram ativadas e desativadas.

Estas observações são apenas uma motivação para a utilização de representações que consigam aliar as vantagens que observamos separadamente nas Figuras 2.1 e 2.2. Uma discussão mais aprofundada sobre o tema pode ser encontrada em [16].

2.1 Transformada de Fourier de Tempo Curto

Como podemos então obter, na mesma representação, informações sobre o tempo (quando uma nota foi tocada) e sobre a frequência (qual nota foi tocada)?

Um exemplo clássico deste tipo de representação é a partitura musical. Na Figura 2.3 podemos ver um pentagrama: cada nota possui informação de tempo (posição e duração) e frequência. O tempo é representado pela posição horizontal e pelo valor da nota, e a frequência é representada pela posição vertical no pentagrama.

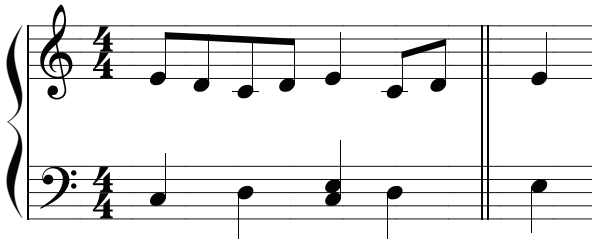


Figura 2.3: Partitura: uma forma de representação tempo-frequencial.

A maneira mais usual de se obter este tipo de representação para sinais digitais de áudio é aplicando a transformada de Fourier de tempo curto (*Short-Time Fourier Transform*, STFT) [16]. Ela funciona como uma espécie de meio termo entre as representações no tempo e na frequência: a DFT é aplicada a pequenos trechos do sinal, de modo que a descrição frequencial evolua ao longo do tempo. Desta maneira, conseguimos uma representação similar à da partitura, aliando os dois tipos de representação numa só. Entretanto, perdemos resolução no tempo, já que das amostras da Figura 2.1 passamos a quadros com duração maior do que uma amostra; e perdemos resolução na frequência, já que a DFT é aplicada a um trecho menor do que aquele da Figura 2.2, e as raias de frequência são espalhadas.

A matriz cujas linhas representam as raias de frequência e cujas colunas representam os quadros temporais é chamada de espectrograma [16]. Cada elemento da

matriz representa a magnitude e a fase do espectrograma num dado ponto (raia, quadro). O procedimento básico para geração de um espectrograma utilizando a STFT consiste nas etapas a seguir:

- Dispondo-se de um sinal no tempo $x(k), k = 0, \ldots, K-1$, ele é segmentado em trechos de W amostras, possivelmente com sobreposição de $W - S$ amostras entre os trechos. Cada um dos M trechos do sinal, indexados por m, pode ser escrito como

$$x^m(k') = x(k' + mS), \quad \text{para} \quad k' = 0, \ldots, W-1. \tag{2.1}$$

- A cada trecho m deve ser aplicada uma janela $w(k'), k' = 0, \ldots, W-1$, para suavização de bordas. As janelas mais comumente usadas são as de *Hamming* e *Hanning* [15].

$$x_w^m(k') = x^m(k')w(k'), \quad \text{para} \quad k' = 0, \ldots, W-1. \tag{2.2}$$

- A cada trecho janelado x_w^m, devemos aplicar uma DFT de N pontos, onde $N \geq W$ deve preferencialmente ser potência de 2 (para utilização da *Fast Fourier Transform*, FFT) [15]. Caso $N > W$, as últimas $N - W$ amostras de x_w^m devem receber 0.

$$X^m(n) = \sum_{k'=0}^{N-1} x_w^m(k') e^{-j\frac{2\pi n}{N}k'}, \quad \text{para} \quad n = 0, \ldots, N-1. \tag{2.3}$$

- A matriz de dimensão $N \times M$ em que cada coluna é formada pelos vetores $|X^m|$ é chamada de espectrograma de magnitude. Seus elementos são todos maiores que ou iguais a zero, o que torna esta representação adequada à aplicação dos algoritmos de fatoração de matrizes não-negativas.

□

A Figura 2.4 mostra a STFT do sinal utilizado nas Figuras 2.1 e 2.2. Neste tipo de representação, que será usada extensivamente ao longo do texto, a intensidade do

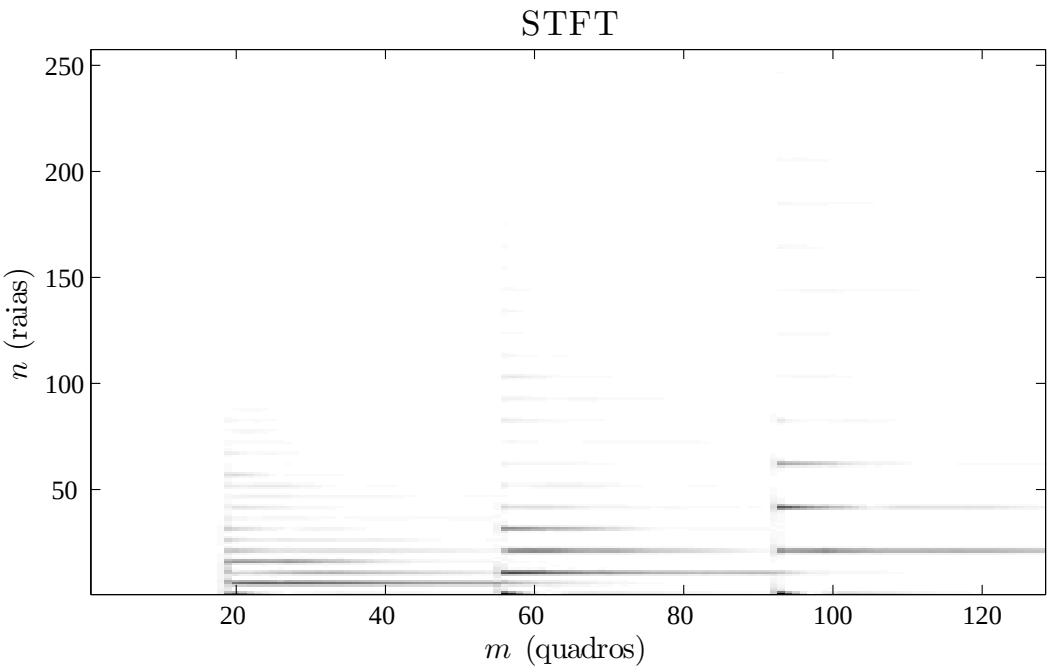


Figura 2.4: Representação tempo-frequencial utilizando a STFT do sinal apresentado nas Figuras 2.1 e 2.2, utilizando a STFT.

sinal em cada par (quadro, raia) é refletida em sua tonalidade. Quanto mais escuro for o ponto, maior é a intensidade representada.

Como visto no procedimento acima, a STFT requer a escolha dos seguintes parâmetros:

- W - Número de amostras de cada janela;
- S - Número de amostras do salto entre janelas adjacentes;
- N - Número de amostras da DFT de cada janela.

A escolha destes parâmetros irá determinar se a representação favorece o tempo ou a frequência. Como exemplo, basta observar os extremos. Supondo a utilização de uma janela muito pequena, de uma amostra, teremos a representação da Figura 2.1, muito boa no tempo e ruim na frequência. Por outro lado, se $W = K$, temos a representação da Figura 2.2, muito precisa na frequência, mas sem informação temporal. Uma discussão sobre o dilema entre precisão no tempo e na frequência pode ser encontrada em [16].

2.1.1 Relação da STFT com a escala musical

O espectro idealmente harmônico de notas musicais tocadas por instrumentos como piano ou violão é formado pela frequência fundamental (f_0) e seus harmônicos (múltiplos inteiros de f_0). De maneira simplista, podemos dizer que a frequência fundamental determina a nota percebida (*pitch*), e a relação de energia entre os harmônicos determina o timbre, que é o que nos faz distinguir entre uma mesma nota lá tocada por um piano e um violão. Uma discussão mais aprofundada sobre a relação entre f_0, *pitch*, harmônicos e timbre pode ser encontrada em [17].

Para todos os exemplos desta dissertação em que surgirem intervalos entre notas musicais, estaremos falando dos intervalos na escala de igual temperamento. Isto significa que a razão entre as frequências de duas notas espaçadas de um semitom será sempre considerada igual a $2^{\frac{1}{12}}$. Esta é a escala usada modernamente no ocidente, e tem a vantagem de permitir que qualquer peça musical possa ser livremente transposta entre as 12 tonalidades (dó, dó#, ré, ré#, mi, fá, fá#, sol, sol#, lá, lá# e si), sem alteração nos intervalos [18].

A Figura 2.5 mostra a STFT de um sinal contendo 13 notas separadas por um tom, abarcando duas oitavas, na escala de igual temperamento. Cada nota possui 8 harmônicos de igual intensidade. As frequências fundamentais das notas seguem o padrão $440 \times 2^{\frac{2k}{12}}$, para $k = 0, \ldots, 12$. A separação entre elas é de dois semitons, ou um tom, de modo que a primeira nota tem $f_0 = 440$ Hz e a décima-terceira nota tem $f_0 = 1760$ Hz. Os harmônicos possuem frequência kf_0, para $k = 2, \ldots, 8$.

Podemos notar na Figura 2.5 que os espaçamentos entre os harmônicos de uma nota são lineares, ao passo que a f_0 cresce logaritmicamente ao longo das notas. Nosso ouvido, contudo, trata este espaçamento logarítmico como linear; a diferença de altura percebida entre um lá a 220 Hz e outro a 440 Hz é a mesma que entre um lá a 440 Hz e outro a 880 Hz [17].

Dada esta característica do sistema auditivo humano, vemos que a DFT não é a transformada mais adequada para o tratamento de sinais de áudio no domínio da frequência. Como as raias da DFT podem ser vistas como frequências centrais de filtros (canais) que dividem o espectro de maneira linear, teremos uma resolução menor em baixas frequências que em altas. Por exemplo, para raias espaçadas de 18,3 Hz, correspondendo à janela de 2410 amostras a uma taxa de amostragem de

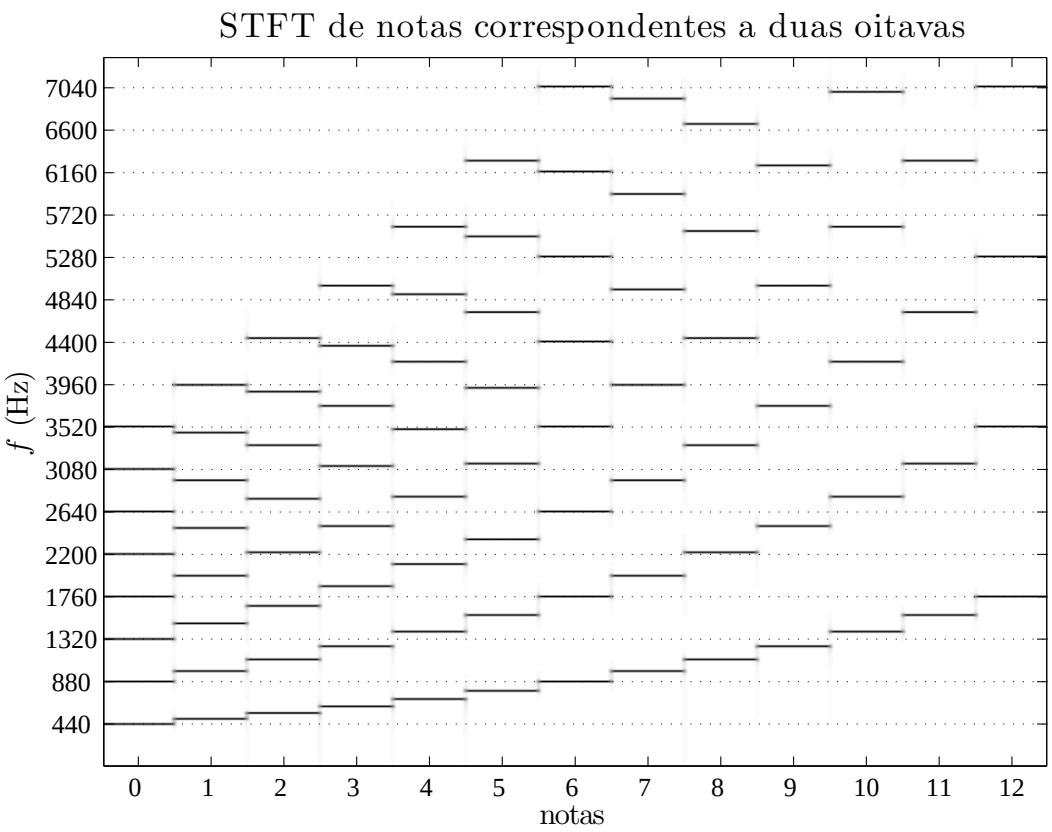


Figura 2.5: STFT de um sinal contendo 13 notas ascendentes separadas por um tom, correspondentes a duas oitavas, cada uma com 8 harmônicos.

44,1 kHz, a oitava de 220 Hz a 440 Hz é representada por 12 raias, enquanto que a oitava seguinte (440 Hz a 880 Hz) recebe 24 raias, ou seja: embora ambas as oitavas sejam perceptivamente equivalentes em extensão, a representação da primeira tem metade da precisão da segunda, devido à diferença na quantidade de raias.

2.2 Transformada de Q-Constante

A transformada de Q constante (*Constant Q-Transform*, CQT) foi apresentada em [19] e [20] com o intuito de resolver este problema. Diferentemente da DFT, que possui largura de banda constante, a CQT possui o fator Q constante. O parâmetro Q de um filtro passa-faixa é definido por [21]

$$Q = \frac{f}{\Delta f}, \tag{2.4}$$

onde f é a frequência central da faixa de passagem e Δf é a largura de banda do filtro. Uma transformada com Q constante significa que a largura dos canais

determinados por cada uma de suas raias é proporcional à frequência das raias. Desta forma, todas as oitavas recebem o mesmo número de raias. A Figura 2.6 mostra o mesmo sinal das Figuras 2.1, 2.2 e 2.4, representado pela CQT.

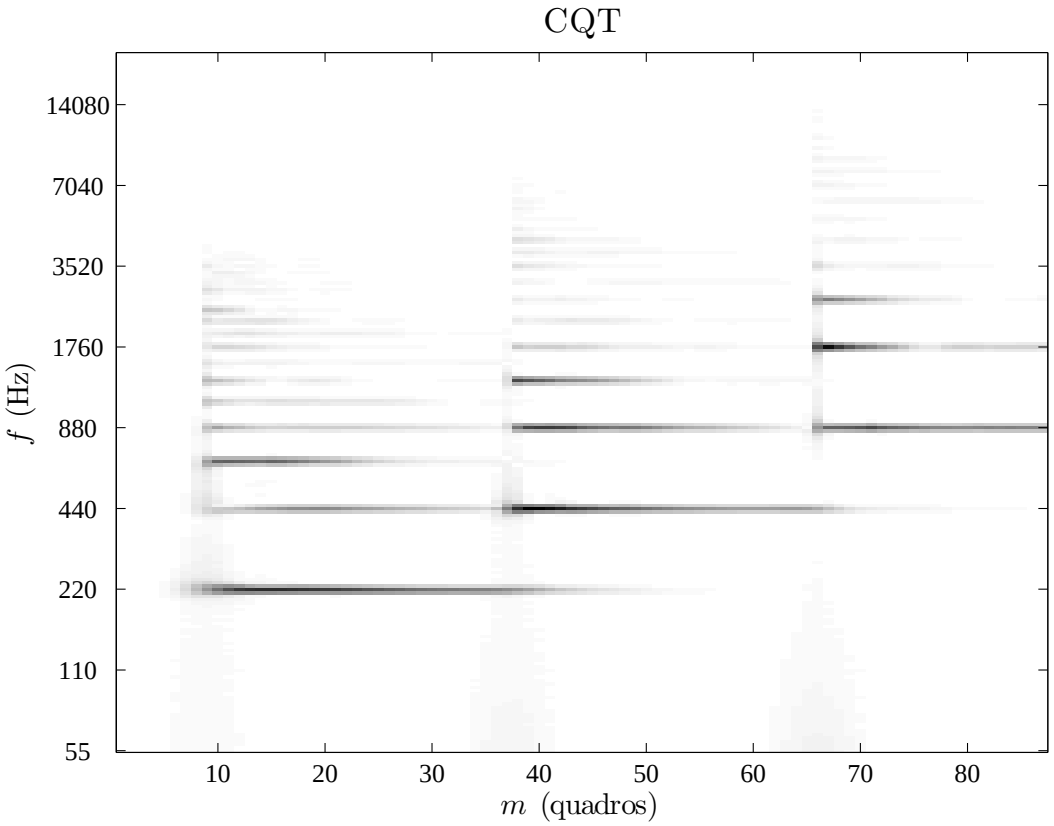


Figura 2.6: CQT de um sinal composto por 3 notas de piano. As frequências estão na escala logarítmica.

Como vimos na seção anterior, a análise através da STFT é feita dividindo-se o sinal em janelas de tamanho W constante, o que resulta em canais da mesma largura espectral independentemente da frequência; consequentemente, cada canal tem o fator Q diferente. A ideia da CQT é fazer o inverso: analisar o sinal utilizando um tamanho de janela W' diferente para cada raia na frequência, de modo a garantir um fator Q constante.

Seguindo o raciocínio apresentado em [22], definimos a frequência de cada raia da CQT como

$$f_n = f_{\min} 2^{\frac{n}{b}}, \tag{2.5}$$

onde $f_{\min}$ é a menor frequência que desejamos representar (em Hz), n é o índice da raia da CQT e b é o número de raias por oitava. Definindo ainda Q como a razão entre frequência central e largura de banda de cada canal associado, podemos dizer

que

$$Q = \left(2^{\frac{1}{b}} - 1\right)^{-1}. \tag{2.6}$$

A largura de cada canal será definida pelo tamanho da janela de análise relativa àquele canal, dada por

$$N_n = Q\frac{f_s}{f_n}, \tag{2.7}$$

onde f_s é a frequência de amostragem do sinal. Definimos, então, a CQT como

$$X^m(n) = \frac{1}{N_n}\sum_{k=0}^{N_n-1} w_{N_n}(k)x(k)\exp^{-j\frac{2\pi k}{N_n}Q}, \tag{2.8}$$

onde $X^m(n)$ é o valor da CQT para a raia n num quadro m, w_{N_n} é a janela para a raia n e x é o sinal no domínio do tempo.

Como podemos ver, a CQT tem complexidade consideravelmente maior do que a STFT. Enquanto a STFT executa uma DFT (FFT) por quadro, a CQT executa uma para cada raia, a cada quadro. Além disso, sua volta ao domínio do tempo não é exata, como será visto no Capítulo 5. Uma implementação eficiente da CQT é apresentada em [20].

2.2.1 Relação da CQT com a escala musical

A Figura 2.7 mostra o resultado da CQT aplicada ao mesmo sinal da Figura 2.5. Ao comparar as duas figuras, chama a atenção o fato de que, na CQT, a mudança de nota corresponde a um deslocamento vertical fixo de todos os harmônicos da nota anterior. Enquanto no caso da STFT cada harmônico é deslocado de um valor diferente, no caso da CQT o deslocamento é fixo.

Com isso, podemos imaginar o espectrograma da Figura 2.7 representado como apenas uma nota e seus deslocamentos. Esta ideia será útil no algoritmo NMF2D, mostrado na Seção 3.3.

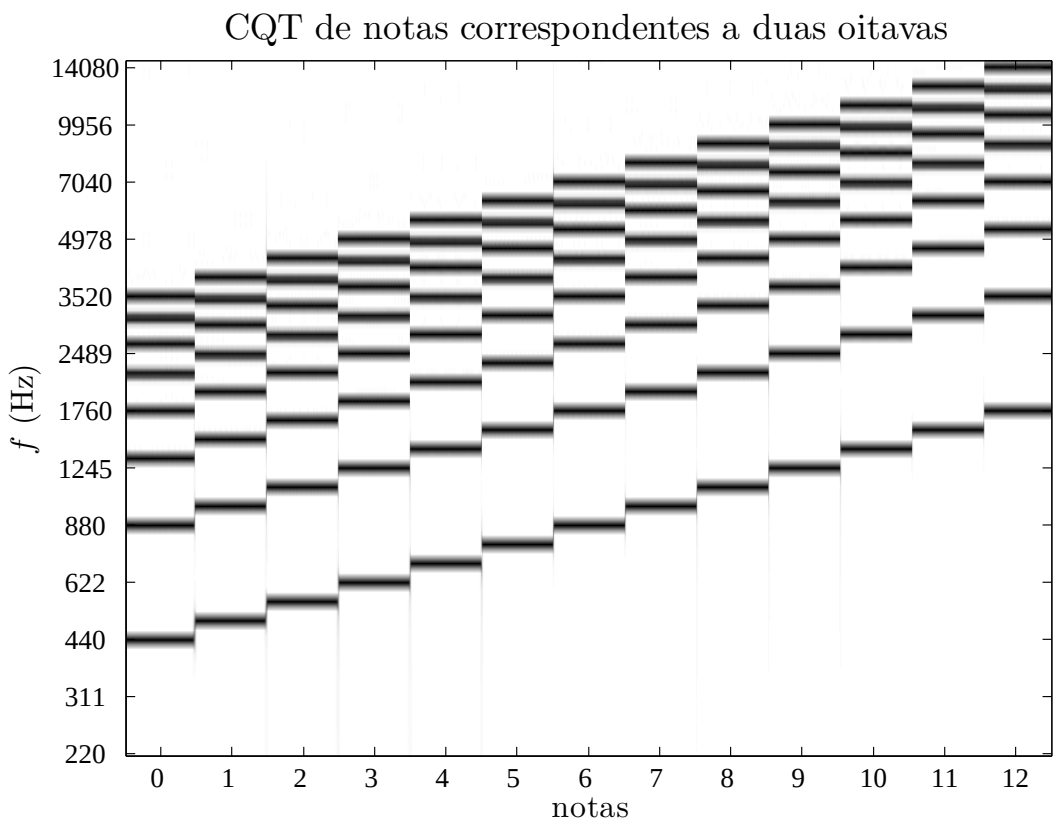


Figura 2.7: CQT de um sinal contendo 13 notas ascendentes separadas de um tom, correspondentes a duas oitavas, cada nota com 8 harmônicos. As frequências estão na escala logarítmica.

2.3 DFT com mapeamento para escala logarítmica

Uma alternativa que une a representação logarítmica da CQT com a eficiência computacional da STFT é o mapeamento logarítmico das raias da STFT [22, 23]. O procedimento consiste em simplesmente agrupar as raias da STFT de modo a conseguir uma representação com igual número de raias por oitava:

$$\mathbf{V}_{\log} = \mathbf{C}\mathbf{V}_{\text{lin}}, \tag{2.9}$$

onde $\mathbf{C} \in \mathbb{R}^{N_{\log} \times N_{\text{lin}}}$ é a matriz de mapeamento, $\mathbf{V}_{\text{lin}} \in \mathbb{R}^{N_{\text{lin}} \times M}$ é o espectrograma linear obtido pela STFT e $\mathbf{V}_{\log} \in \mathbb{R}^{N_{\log} \times M}$ é o espectrograma logarítmico que desejamos.

A matriz $\mathbf{C}$ deve ser montada de forma que as oitavas de $\mathbf{V}_{\text{lin}}$ sejam mapeadas nas oitavas de $\mathbf{V}_{\log}$. O número de raias por oitava em $\mathbf{V}_{\text{lin}}$ é variável e em $\mathbf{V}_{\log}$ é

fixo. Nas baixas frequências teremos poucas raias n_{lin} para cada raia $n_{\log}$, portanto a curva é mais próxima de uma vertical. À medida que aumentamos a frequência, a inclinação da curva cai, pois temos mais raias n_{lin} para cada raia $n_{\log}$. Os valores de cada ponto da matriz $\mathbf{C}$ são calculados de modo que todas as linhas somem 1; desta maneira, a energia de $\mathbf{V}_{\text{lin}}$ é mantida em $\mathbf{V}_{\log}$. No Anexo A é mostrado o algoritmo utilizado para construir a matriz $\mathbf{C}$.

A Figura 2.8 mostra a representação das matrizes da equação (2.9). Comparando $\mathbf{V}_{\log}$ (ao centro da Figura 2.8) com a Figura 2.7, percebemos pior resolução nas frequências baixas. Isso ocorre devido à pior representação destas frequências pela DFT, como explicitado na Seção 2.1. Caso desejemos garantir uma boa resolução na oitava de 110 Hz a 220 Hz, digamos, com 24 raias, a distância entre raias terá de ser igual a 4,58 Hz. Para atingir esta precisão num sinal amostrado a 44,1 kHz, teríamos que realizar uma DFT de 9622 canais, cuja potência de 2 mais próxima é $2^{14} = 16384$.

□

Neste capítulo, vimos que existem diversas maneiras de representar o sinal no domínio tempo-frequência. A escolha do tipo de transformada irá impactar na escolha dos algoritmos de fatoração, que serão vistos no próximo capítulo. No Capítulo 8, o desempenho do sistema de separação será avaliado com a utilização das transformadas aqui descritas.

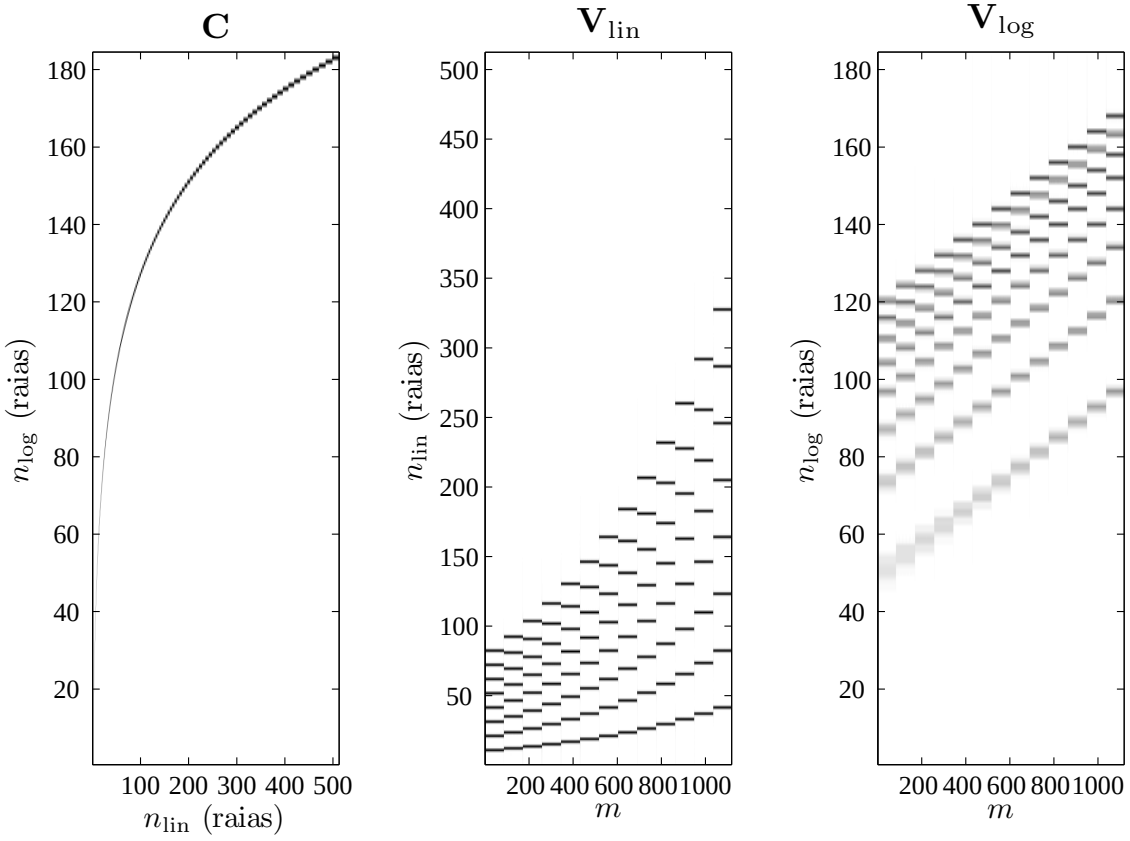


Figura 2.8: Mapeamento da escala linear para a logarítmica. O gráfico à esquerda mostra uma representação gráfica da matriz de mapeamento $\mathbf{C}$. Ao centro temos o espectro linear $\mathbf{V}_{\text{lin}}$ gerado pela STFT, e à direita temos $\mathbf{V}_{\text{log}}$, o resultado de $\mathbf{C}\mathbf{V}_{\text{lin}}$.

Capítulo 3

Fatoração de Matrizes Não-Negativas: Métodos e Aplicações

O objetivo deste capítulo é apresentar trabalhos significativos da literatura que utilizam a fatoração de matrizes não-negativas. O método pode ser visto por um prisma puramente matemático, ou seja: como representar uma matriz de elementos maiores ou iguais a zero como o produto de duas matrizes também não-negativas? A partir desta motivação, aplicações em diversas áreas podem ser encontradas, utilizando o algoritmo-base ou especializando-o.

Neste capítulo, daremos ênfase às aplicações a separação de sinais de áudio, sem no entanto desprezar outros trabalhos cujos resultados podem ser facilmente transportados para essa aplicação. Enquanto o primeiro artigo de grande impacto sobre NMF [9] foi publicado na revista *Nature* em 1999, os trabalhos descrevendo aplicações em separação de áudio surgiram apenas em 2003 [10].

Segundo [9], existem evidências psicológicas e fisiológicas de que o cérebro utiliza representações baseadas em partes na tarefa de interpretação de imagens. Entretanto, pouco se sabe sobre como ocorre o aprendizado destas partes. O algoritmo apresentado em [9], denominado *Non-Negative Matrix Factorization* (NMF, Fatoração de Matrizes Não-Negativas), busca justamente esta decomposição de matrizes não-negativas em partes. Nesse artigo, são apresentados resultados da aplicação da NMF à decomposição de um banco de imagens faciais, e as partes resultantes do

algoritmo são elementos do rosto humano, como boca, sobrancelha e olhos.

A primeira seção deste capítulo apresenta em detalhes o algoritmo de NMF, incluindo as regras de atualização e funções-custo, após o que as aplicações em áudio são revisadas. As seções seguintes apresentam evoluções da NMF que a especializam para o tratamento de sinais de áudio.

3.1 *Non-Negative Matrix Factorization* (NMF)

A grande quantidade de dados gerados por sensores de todos os tipos demanda soluções robustas e eficientes para a extração e o tratamento das informações-alvo. Uma abordagem possível para este tipo de problema são as decomposições de baixo posto, cujo objetivo geral é reduzir o número de dimensões de um problema multivariável.

Uma técnica clássica utilizada na abordagem deste tipo de problema é a Análise de Componentes Principais (*Principal Component Analysis*, PCA [5], Cap. 6). A partir de um conjunto de dados multivariáveis, a PCA busca um conjunto menor de variáveis que possuam redundância reduzida. Esta redundância pode ser medida através da correlação estatística entre variáveis. A técnica de PCA tem forte relação com a ICA, citada no Capítulo 1, sendo a principal diferença na medida de redundância, que no caso da ICA é feita pela dependência estatística.

No casos em que os dados tratados sejam por natureza não-negativos (contagem, intensidade de um píxel), gostaríamos que a decomposição resultasse em fatores não-negativos. Tanto a técnica de PCA quanto a de ICA não garantem isso. Desta maneira, uma decomposição de dados que representem, por exemplo, os píxeis de uma imagem pode resultar em variáveis com valores negativos. Com isso, a decomposição, apesar de matematicamente correta, afasta-se do sentido físico do problema.

Neste contexto, a NMF surge como uma opção de técnica de redução de dimensionalidade que preserva a característica não-negativa da representação. Ao aplicar esta restrição, outras características emergem, sendo a principal delas a decomposição em partes. A não-negatividade resulta numa decomposição puramente aditiva: um "todo" é descrito como soma de várias "partes". Uma discussão mais aprofundada sobre a decomposição em partes e as condições que levam a isto pode ser

encontrada em [24].

O problema da fatoração de matrizes não-negativas pode, então, ser definido da seguinte maneira [25]:

Problema 1. *Dada uma matriz não-negativa* $\mathbf{V} \in \mathbb{R}_+^{N\times M}$ *e um inteiro positivo* $D < \min(N, M)$*, ache as matrizes não-negativas* $\mathbf{W} \in \mathbb{R}_+^{N\times D}$ *e* $\mathbf{H} \in \mathbb{R}_+^{D\times M}$ *que minimizem a função*

$$f(\mathbf{W}, \mathbf{H}) = \frac{1}{2}||\mathbf{V} - \mathbf{WH}||_F^2, \tag{3.1}$$

onde $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$ *denota a norma de Frobenius.*

O produto **WH** é chamado de fatoração não-negativa de **V**. Entretanto, **V** não é necessariamente igual a **WH**; em geral teremos uma aproximação com posto no máximo igual a D. Ao longo do texto, usaremos a seguinte definição:

$$\mathbf{V} \approx \mathbf{\Lambda} = \mathbf{WH}, \tag{3.2}$$

onde $\mathbf{\Lambda} \in \mathbb{R}_+^{N\times M}$ será a aproximação de **V**.

Os fatores **W** e **H** devem ser calculados através de algoritmos de otimização, de forma a solucionar o Problema 1. A próxima subseção trata do algoritmo básico de otimização a ser usado.

3.1.1 Algoritmo de Otimização

Segundo [11], o Problema 1 é convexo em **W** ou **H**, mas não em ambos. Desta forma, utilizando o método de gradiente descendente [26] de forma alternada e com um passo suficientemente pequeno, garantimos que o erro $E = f(\mathbf{W}, \mathbf{H})$ sempre diminui. Definindo $W_{n,m}$ como um elemento de **W**, a equação de atualização para **W** é dada por

$$W_{n,d} \leftarrow W_{n,d} - \mu_W \frac{\partial E}{\partial W_{n,d}}, \tag{3.3}$$

com

$$\frac{\partial E}{\partial W_{n,d}} = \sum_{n=1}^{N}\sum_{m=1}^{M}(V_{n,m} - \Lambda_{n,m})\frac{\partial \mathbf{\Lambda}}{\partial W_{n,d}}. \tag{3.4}$$

A derivada de um elemento de $\mathbf{\Lambda}$ em relação a $W_{n,d}$ é

$$\frac{\partial \Lambda_{n',m'}}{\partial W_{n,d}} = \begin{cases} \frac{\partial}{\partial W_{n,d}} \sum_{k=1}^{D} W_{n,k}H_{k,m'} = H_{d,m'}, & \text{para} \quad n = n', \quad \text{e} \\ 0, & \text{para} \quad n \neq n', \end{cases} \tag{3.5}$$

e portanto

$$\frac{\partial E}{\partial W_{n,d}} = \sum_{m'=1}^{M}(V_{n,m'} - \Lambda_{n,m'})H_{d,m'}. \tag{3.6}$$

Para todos os elementos de $\mathbf{W}$, temos:

$$\frac{\partial E}{\partial \mathbf{W}} = (\mathbf{V} - \mathbf{\Lambda})\mathbf{H}^T, \tag{3.7}$$

e com isso

$$\mathbf{W} \leftarrow \mathbf{W} + \mu_W(\mathbf{V} - \mathbf{\Lambda})\mathbf{H}^T,$$

ou

$$\mathbf{W} \leftarrow \mathbf{W} + \mu_W(\mathbf{V}\mathbf{H}^T - \mathbf{\Lambda}\mathbf{H}^T). \tag{3.8}$$

Esta equação de atualização não garante o atendimento à restrição de não-negatividade do problema. No entanto, caso as matrizes sejam inicializadas com valores não-negativos e a atualização seja multiplicativa por um fator não-negativo, garantimos automaticamente que os elementos nunca assumirão valores negativos. Então, devemos escolher μ_W de forma que o valor seguinte de $\mathbf{W}$ seja ele próprio multiplicado por um número não-negativo. Este procedimento, que mostraremos a seguir, será de suma importância para o restante deste trabalho.

O passo μ_W, até aqui representado por um escalar, é substituído por uma matriz de dimensões $N \times D$. As operações de 'divisão entre matrizes' são realizadas ponto-

a-ponto, e $\otimes$ denota o produto de Hadamard, onde os elementos são multiplicados também ponto-a-ponto. Fazendo

$$\mu_W \quad = \quad \frac{\mathbf{W}}{\mathbf{\Lambda H}^T}, \tag{3.9}$$

obtemos

$$\mathbf{W} \quad \leftarrow \quad \mathbf{W} \otimes \left(\mathbf{1} + \frac{\mathbf{V H}^T}{\mathbf{\Lambda H}^T} - \frac{\mathbf{\Lambda H}^T}{\mathbf{\Lambda H}^T} \right), \tag{3.10}$$

ou

$$\mathbf{W} \quad \leftarrow \quad \mathbf{W} \otimes \frac{\mathbf{V H}^T}{\mathbf{W H H}^T}, \tag{3.11}$$

que é à regra de atualização desejada para a matriz $\mathbf{W}$. O símbolo $\mathbf{1}$ representa uma matriz $N \times D$ com todos os elementos iguais a 1.

Podemos notar que para a decomposição de $\mathbf{V}^T$, as matrizes $\mathbf{W}$ e $\mathbf{H}$ seriam substituídas por $\mathbf{H}^T$ e $\mathbf{W}^T$, respectivamente. Portanto, todo o desenvolvimento feito para a matriz $\mathbf{W}$ pode ser estendido a $\mathbf{H}$, operando-se esta troca:

$$\mathbf{H} \quad \leftarrow \quad \left(\mathbf{H}^T \otimes \frac{\mathbf{V}^T \mathbf{W}}{\mathbf{H}^T \mathbf{W}^T \mathbf{W}} \right)^T, \tag{3.12}$$

ou

$$\mathbf{H} \quad \leftarrow \quad \mathbf{H} \otimes \frac{\mathbf{W}^T \mathbf{V}}{\mathbf{W}^T \mathbf{W H}}. \tag{3.13}$$

A prova de convergência do algoritmo pode ser encontrada em [11]. O Algoritmo 3.1 mostra o procedimento básico da NMF que minimiza a distância euclidiana.

3.1.2 Função-Custo

Na definição do Problema 1, utilizamos como função-custo a distância euclidiana (DE), expressa na equação (3.1). Esta escolha traz consigo duas decisões de projeto: (1) o único objetivo da fatoração é a reconstrução, ou seja, a busca das matrizes $\mathbf{W}$

Algoritmo 3.1 NMF

Entrada: Matriz não-negativa $\mathbf{V} \in \mathbb{R}_+^{N \times M}$ e número de instrumentos D.

1. Inicialize as matrizes $\mathbf{W} \in \mathbb{R}_+^{N \times D}$ e $\mathbf{H} \in \mathbb{R}_+^{D \times M}$ com valores aleatórios não-negativos distribuídos uniformemente entre 0 e 1;
2. Atualize $\mathbf{W}$ utilizando a equação (3.11);
3. Atualize $\mathbf{H}$ utilizando a equação (3.13);
4. Volte ao passo 2 até atingir a convergência ou um número máximo de iterações.

Saída: Matrizes $\mathbf{W} \in \mathbb{R}_+^{N \times D}$ e $\mathbf{H} \in \mathbb{R}_+^{D \times M}$.

e $\mathbf{H}$ cujo produto seja o mais próximo possível de $\mathbf{V}$; e (2) a proximidade de $\mathbf{WH}$ em relação a $\mathbf{V}$ deve ser medida pela DE.

O desenvolvimento das equações de atualização foi mostrado utilizando a DE por simplicidade. Dependendo do problema, no entanto, pode ser favorável utilizar outras medidas para o cálculo da distância, e outros critérios, além da reconstrução.

Uma medida de distância comumente usada é inspirada na Divergência de Kullback-Leibler (DKL) [27]:

$$f_{\mathrm{KL}}(\mathbf{W}, \mathbf{H}) = \left\| \mathbf{V} \otimes \ln\left(\frac{\mathbf{V}}{\mathbf{WH}}\right) - \mathbf{V} + \mathbf{WH}) \right\|_F . \tag{3.14}$$

Rigorosamente, a medida só poderia ser chamada de divergência de Kullback-Leibler quando $\mathbf{V}$ e $\mathbf{\Lambda}$ representassem distribuições de probabilidades. No entanto, feita esta ressalva, adotaremos daqui em diante este nome por simplicidade. A medida não representa uma distância, pois não é simétrica, mas tem seu mínimo em zero, que só é atingido quando $\mathbf{V} = \mathbf{\Lambda}$.

Além do critério de reconstrução, outros critérios podem ser inseridos na função-custo. Essa escolha também é dependente do problema, e pode, por exemplo, garantir algum tipo de estrutura para a matriz $\mathbf{\Lambda}$, para que ela possua sentido físico. Caso se tratasse de uma distribuição de probabilidades, por exemplo, um dos critérios de otimização seria a norma unitária.

Tanto uma mudança na medida de distância quanto no critério de otimização afetam diretamente as equações de atualização, que são o cerne do algoritmo. O de-

senvolvimento apresentado anteriormente, que culmina na equações (3.11) e (3.13), foi feito de acordo com o projeto que utiliza a reconstrução como único critério, e a distância euclidiana como medida de distância. Em [9], podem ser encontradas as equações de atualização referentes ao critério de reconstrução utilizando a divergência de Kullback-Leibler.

Na Seção 3.1.5 são descritos trabalhos recentes que, entre outras modificações na NMF, sugerem novas medidas de distância e novos critérios de otimização.

3.1.3 Interpretação

Como já discutido anteriormente, a NMF é utilizada quando os dados do problema (matriz $\mathbf{V}$) a ser tratado são não-negativos por natureza. Nestes casos, a vantagem da decomposição em fatores também não-negativos (matrizes $\mathbf{W}$ e $\mathbf{H}$) reside na possibilidade de que os fatores encontrados sejam fisicamente interpretáveis.

No experimento apresentado em [9], os autores discutem a aplicação da NMF a um banco de imagens faciais contendo $M = 2429$ imagens, cada uma formada por $N = (19 \times 19)$ píxeis. As imagens são representadas numa matriz $\mathbf{V}$ com N linhas e M colunas, sendo cada coluna formada por uma imagem transformada em um vetor de tamanho N.

A decomposição foi aproximada utilizando $D = 49$. Desta maneira, a matriz $\mathbf{W} \in \mathbb{R}^{N \times D}$ é formada por 49 imagens de (19×19) píxeis. Chamaremos estas imagens de *imagens da base* $\mathbf{W}$. A matriz $\mathbf{H} \in \mathbb{R}^{D \times M}$, por sua vez, possui 49 linhas e 2429 colunas, e podemos interpretar cada ponto (d_0, m_0) como o ganho da imagem da base d_0 aplicado à reconstrução da imagem m_0. Isso dá a $\mathbf{H}$ o status de uma *matriz de ganhos*.

Para reconstruir uma imagem $\mathbf{X}_{m_0} \in \mathbb{R}^N$ do banco, fazemos

$$\mathbf{X}_{m_0} = \mathbf{W}\mathbf{H}(:, m_0) \tag{3.15}$$

$$= \sum_{d=1}^{D} \mathbf{W}(:, d)\mathbf{H}(d, m_0), \tag{3.16}$$

onde $\mathbf{A}(x, y)$ denota um elemento de $\mathbf{A}$, e $\mathbf{A}(:, y)$ ou $\mathbf{A}(x, :)$ denotam vetores contendo todas as linhas da coluna y ou todas as colunas da linha x, respectivamente. A equação (3.16) pode ser vista como a soma das contribuições de cada imagem da

base para a imagem de índice m_0.

Esta interpretação da NMF é apropriada a problemas em que o foco é decompor a informação contida nas colunas separadamente. Neste caso, a ordem das colunas da matriz $\mathbf{V}$ não importa. No caso da separação de sinais de áudio, a interpretação mais adequada será vista na próxima subseção.

Na Figura 3.1 podemos observar uma decomposição ideal, onde $\mathbf{\Lambda} = \mathbf{V}$. A matriz $\mathbf{W}$, à direita, mostra as $D = 3$ imagens da base decompostas, enquanto a matriz $\mathbf{H}$, acima, mostra em cada linha os ganhos relativos a cada imagem da base.

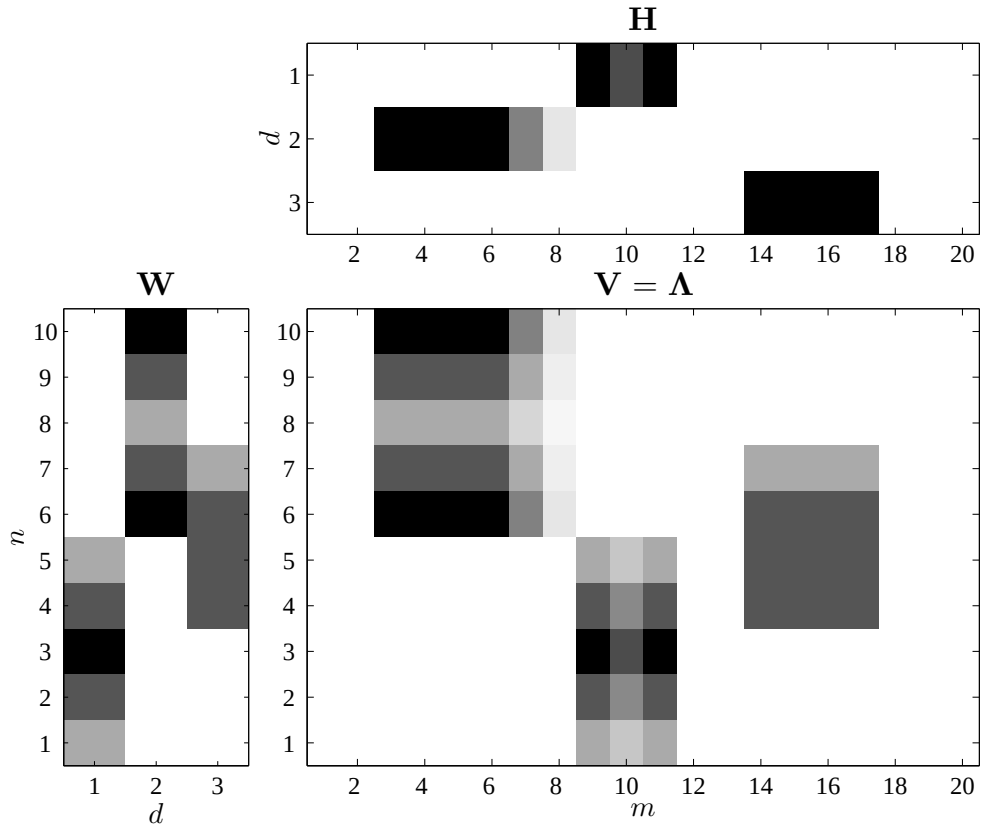


Figura 3.1: Exemplo ilustrativo do modelo utilizado pelo algoritmo NMF, utilizando $N = 10$ e $M = 20$. A matriz $\mathbf{V}$ foi decomposta utilizando uma base de tamanho $D = 3$, cujos vetores podem ser vistos nas colunas de $\mathbf{W}$. As linhas de $\mathbf{H}$ mostram os ganhos aplicados a cada vetor da base, e o resultado do produto $\mathbf{WH}$ resulta na matriz $\mathbf{\Lambda}$, neste caso igual a $\mathbf{V}$.

3.1.4 Aplicações em Áudio

A aplicação do algoritmo de NMF à separação de sinais de áudio se dá utilizando uma interpretação ligeiramente diferente daquela apresentada na subseção anterior. A partir deste ponto, vamos considerar que a matriz $\mathbf{V}$ representa um espectrograma,

ou seja, o sinal no tempo é dividido em quadros, e para cada quadro aplicamos uma transformada para o domínio da frequência, como visto no Capítulo 2. Assim, cada coluna da matriz $\mathbf{V}$ representa um quadro, e cada linha, uma raia de frequência. Lembrando que o espectro de magnitude de um sinal real é simétrico, utilizaremos apenas metade do espectrograma na decomposição.

Os vetores da base representados pelo modelo nas colunas da matriz $\mathbf{W}$ podem então ser compreendidos como padrões espectrais que se repetem ao longo dos quadros do espectrograma representado pela matriz $\mathbf{V}$. A matriz $\mathbf{H}$ descreve o ganho com que cada vetor da base ocorre em cada quadro. Esta interpretação nos leva ao conceito de componentes, que são o resultado da multiplicação de uma coluna d_0 de $\mathbf{W}$ por uma linha d_0 de $\mathbf{H}$. Mais adiante, os componentes serão relacionados a fontes sonoras.

O d-ésimo componente tem seu padrão espectral descrito na d-ésima coluna de $\mathbf{W}$. Sua descrição temporal, isto é, de quando ao longo dos quadros ele deve ocorrer, pode ser encontrada na d-ésima linha da matriz $\mathbf{H}$. Quando um padrão é multiplicado por sua descrição temporal, temos como resultado o espectrograma individual relativo ao d-ésimo componente. Ao somarmos estes espectrogramas individuais chegamos à matriz $\mathbf{\Lambda}$, que é uma estimativa de $\mathbf{V}$, como mostra a equação (3.2).

Neste ponto, devemos observar que uma importante aproximação é realizada. O espectrograma de magnitude de uma mistura não é formado pelas somas dos espectrogramas de magnitude das fontes separadas [16]. Partindo da mistura de dois sinais no tempo,

$$y = x_1 + x_2, \tag{3.17}$$

onde y é a mistura e x_1 e x_2 são as fontes separadas, temos que

$$Y = X_1 + X_2, \tag{3.18}$$

onde Y, X_1 e X_2 representam as STFTs de y, x_1 e x_2, respectivamente. Os espec-

trogramas de magnitude são escritos como

$$
\begin{aligned}
|Y| &= |X_1 + X_2| && (3.19)\\
&= |X_1| + |X_2| + 2\mathrm{Re}\{X_1^* X_2\} && (3.20)\\
&\neq |X_1| + |X_2|, && (3.21)
\end{aligned}
$$

onde $\mathrm{Re}\{\cdot\}$ é a parte real de um sinal complexo, e $(\cdot)^*$ é o complexo conjugado do sinal $(\cdot)$.

A não-aditividade de espectrogramas de magnitude decorre do fato de que as diversas componentes emitidas pelas fontes sonoras que se deseja separar interferem mutuamente no tempo, o que acaba por modificá-las (ponderá-las diferentemente) no espectro. Estas interferências serão tanto mais significativas quanto maior for a sobreposição tempo-frequencial entre as fontes.

Entretanto, podemos justificar a aproximação observando que o sinal de uma mistura que chega aos nossos ouvidos tampouco é interpretado como se estivéssemos ouvindo as fontes separadamente. Estas fontes que formam a mistura também sofrem interferências que afetam a maneira como as ouvimos. Esta discussão nos remete mais uma vez à definição de fonte sonora. O maior grau de separação a que podemos almejar é aquele realizado por nosso cérebro, ou seja, a separação do conteúdo da mistura relativo a cada fonte. E a aproximação que fazemos segue esta linha de raciocínio ao separar no espectrograma da mistura o que há de cada fonte.

A Figura 3.2 mostra um exemplo real da aplicação da NMF a sinais musicais. Cinco notas de piano são tocadas em sequência, e posteriormente a primeira, a terceira e a quinta são tocadas simultaneamente. A decomposição foi feita utilizando a divergência de Kullback-Leibler como medida de reconstrução. Segundo [27], ela é a mais indicada para o caso de sinais de áudio por ser mais sensível a diferenças em baixas energias, comportamento similar ao do sistema auditivo humano.

3.1.5 Modificações sobre o algoritmo básico de NMF

Além da distância euclidiana e da divergência de Kullback-Leibler, trabalhos recentes propuseram a utilização de outras medidas de distância para derivação das equações de atualização da NMF. Além disso, alguns trabalhos incorporaram outros

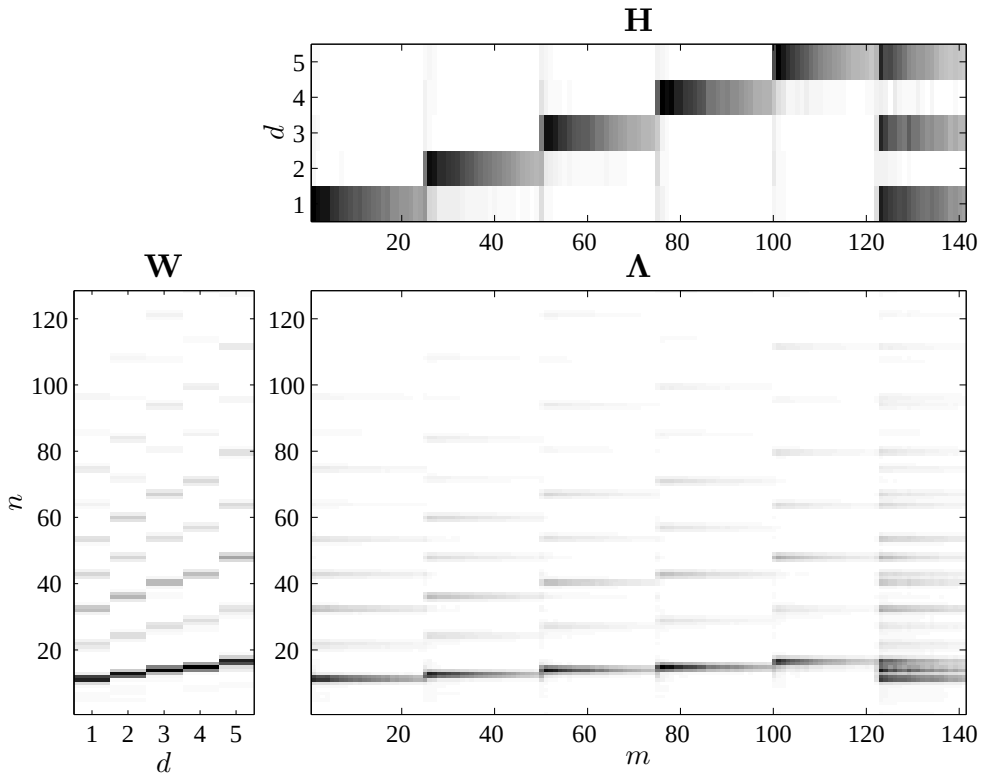


Figura 3.2: Exemplo de aplicação da NMF a um sinal de áudio, para $N = 128$, $M = 142$ e $D = 5$. Neste exemplo, 5 notas são tocadas de forma consecutiva, e ao final 3 delas são tocadas simultaneamente. A matriz **H** mostra os acionamentos das notas, e a matriz **W** mostra o padrão espectral relativo às notas.

critérios à função-custo, além do critério de reconstrução.

Em [28], o autor introduz o critério de esparsidade para fatoração de matrizes não-negativas, em problemas gerais. O artigo mostra como a incorporação deste critério contribui para uma solução baseada em partes. A ideia é assumir que as matrizes **W** e/ou **H** são esparsas, isto é, possuem a maioria dos elementos nulos, o que é razoável para sinais de áudio. Pensando em **H** como a matriz de ocorrência de notas, e assumindo o sinal musical nos padrões ocidentais, apenas poucas notas estarão ativas no mesmo quadro. E pensando em instrumentos harmônicos, os padrões descritos em **W** também serão esparsos, já que apresentarão energia apenas nos múltiplos da frequência fundamental.

Em [27], além de minimizar o erro de reconstrução, o autor inclui mais dois critérios para minimização, específicos para aplicação a sinais de áudio. O primeiro deles é o de continuidade temporal. Partindo do pressuposto de que os sinais têm variação lenta de ganho, o algoritmo impõe uma penalidade a mudanças bruscas ao longo das linhas de **H**. Além disso, o autor também impõe um critério de esparsidade

à matriz **H**, penalizando elementos diferentes de zero.

Outra questão bastante discutida em relação aos algoritmos de NMF é a unicidade da solução. No algoritmo básico não há nada que a garanta. Em [24], este tema é discutido em profundidade, e são dadas as condições sobre os dados para que a solução seja única. Em [29], os autores propõem o critério de mínimo determinante, que daria ao algoritmo uma solução única. Os resultados são comparados com os do algoritmo que impõe esparsidade, descrito em [28].

Em [30] e [31], os autores discutem uma classe de funções-custo para NMF e derivam os respectivos algoritmos. Neste caso, o foco são sinais ruidosos. A classe proposta generaliza diversas funções conhecidas, como por exemplo a distância euclidiana, a divergência de Kullback-Leibler, e as distâncias de Itakura-Saito, Hellinger, Pearson, e Neyman. O efeito da regularização também é analisado.

Em [32], o problema de NMF é abordado com a utilização de algoritmos de gradiente projetado [26]. Em particular, são analisados os algoritmos Landweber projetado, gradiente projetado de Barzilai-Borwein, otimização de subespaço sequencial otimizado (PSESOP) e pontos interiores de Newton.

Em [33], os autores propõem a divergência de Itakura-Saito como função-custo para a NMF, e constroem o algoritmo denominado IS-NMF. Neste algoritmo, restrições de regularização baseadas em prioris bayesianas são utilizadas. Os resultados são comparados com os algoritmos comuns de NMF baseados na distância euclidiana e na divergência de Kullback-Leibler, com bons resultados.

3.2 *Non-Negative Matrix Factor Deconvolution* (NMFD)

A NMFD [12] é uma expansão do modelo NMF. Ela permite que os padrões evoluam no tempo, ocupando mais de um quadro. A ideia deste algoritmo é representar notas de instrumentos.

Uma nota de piano, por exemplo, possui um ataque percussivo e uma sustentação harmônica. No caso da NMF, não seria possível representar esta nota como um componente, já que seu espectro é variante no tempo. No caso da NMFD isto é possível, já que os vetores da base espectral ocupam mais de um quadro, tornando-

se portanto matrizes da base espectral, e assim comportando variações no espectro ao longo do tempo.

Neste ponto devemos introduzir o operador deslocamento horizontal, que aplicado por exemplo à matriz

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \tag{3.22}$$

resulta em

$$\overset{\rightarrow 0}{\mathbf{A}} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \overset{\rightarrow 1}{\mathbf{A}} = \begin{bmatrix} 0 & 1 & 2 \\ 0 & 4 & 5 \\ 0 & 7 & 8 \end{bmatrix}, \overset{\rightarrow 2}{\mathbf{A}} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 4 \\ 0 & 0 & 7 \end{bmatrix}, \ldots . \tag{3.23}$$

Com isso, o modelo NMFD pode então ser representado pela expressão

$$\mathbf{V} \approx \mathbf{\Lambda} = \sum_{t=0}^{\tau-1} \mathbf{W}^t \overset{\rightarrow t}{\mathbf{H}}, \tag{3.24}$$

onde τ é o número de quadros permitidos para cada componente. Cada matriz $\mathbf{W}^t \in \mathbb{R}_+^{N \times D}$, para $t = 0, \ldots, \tau - 1$, representa os padrões espectrais de cada componente relativos a um determinado quadro t.

Seguindo o mesmo procedimento da seção anterior para encontrar as atualizações multiplicativas, chegamos a [12]

$$\mathbf{W}^t \leftarrow \mathbf{W}^t \otimes \frac{\frac{\mathbf{V}}{\mathbf{\Lambda}} \left(\overset{\rightarrow t}{\mathbf{H}^T}\right)}{\mathbf{1} \left(\overset{\rightarrow t}{\mathbf{H}^T}\right)} \quad \text{para} \quad t = 0, \ldots, \tau - 1 \quad \text{e} \tag{3.25}$$

$$\mathbf{H} \leftarrow \mathbf{H} \otimes \frac{\mathbf{W}^{tT} \left(\overset{\leftarrow t}{\frac{\mathbf{V}}{\mathbf{\Lambda}}}\right)}{\mathbf{W}^{tT} \mathbf{1}} \quad \text{para} \quad t = 0, \ldots, \tau - 1, \tag{3.26}$$

onde $\mathbf{1}$ é uma matriz $N \times M$ com todos os elementos iguais a um. A Figura 3.3 ilustra o modelo para um caso em que $\mathbf{V} = \mathbf{\Lambda}$, e o Algoritmo 3.2 descreve os passos da NMFD.

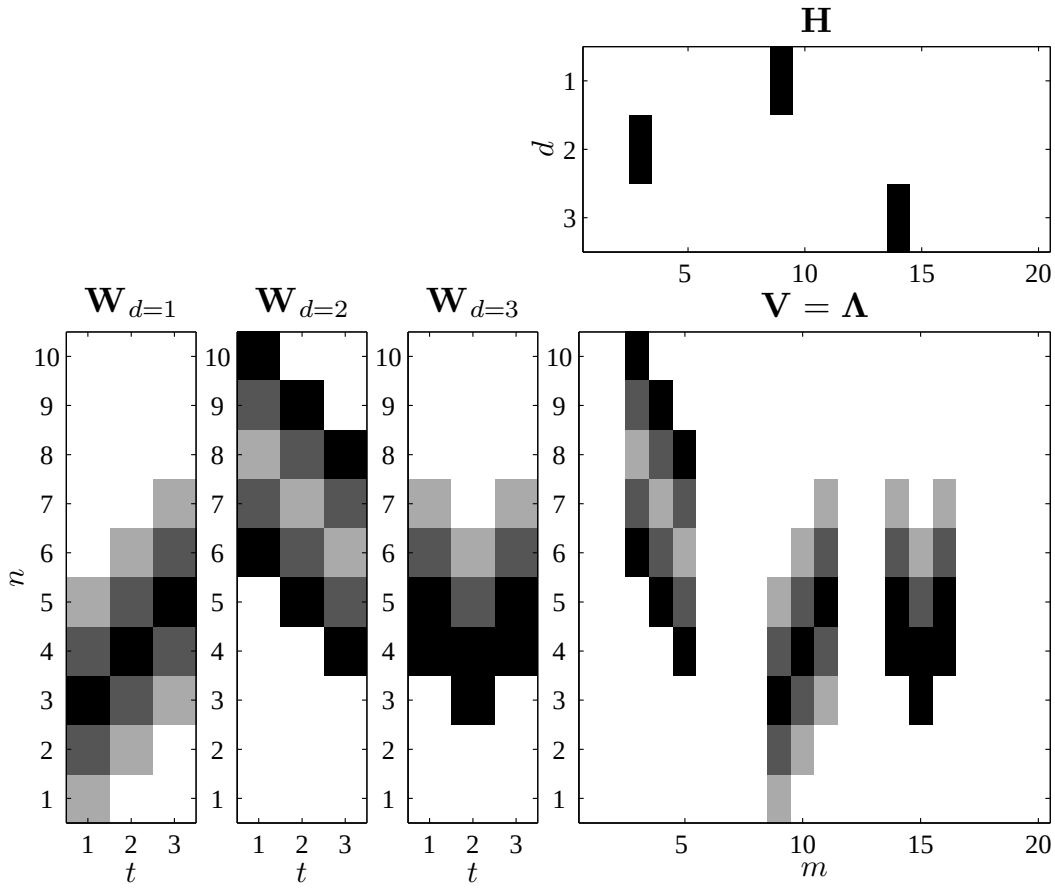


Figura 3.3: Exemplo ilustrativo da NMFD para $N = 10$, $M = 20$, $D = 3$ e $\tau = 3$. Os três gráficos à esquerda mostram as matrizes da base espectral, com dimensão $N \times \tau$. Desta forma, podem representar padrões com variação no tempo. O gráfico superior mostra a matriz $\mathbf{H}$, que não muda em relação ao modelo da NMF. Neste exemplo, para melhor visualização, cada componente tem apenas uma ativação unitária. A matriz $\mathbf{\Lambda}$, neste caso idêntica à $\mathbf{V}$, pode ser vista como a soma das convoluções de cada padrão com a respectiva linha da matriz $\mathbf{H}$, isto é, do padrão $d = 1$ com a primeira linha de $\mathbf{H}$, e assim por diante.

3.3 *Non-Negative Matrix Factor 2D Deconvolution* (NMF2D)

Assim como o modelo anterior permitiu a expansão na dimensão temporal, a NMF2D [13] adiciona a dimensão frequencial ao modelo. Ele permite que os padrões sejam deslocados também no eixo das frequências, modelando assim um instrumento que sempre emite o mesmo padrão espectral, apenas deslocado no espectrograma para cima ou para baixo, de acordo com a nota musical emitida.

Neste ponto, devemos introduzir o operador de deslocamento para baixo, que

Algoritmo 3.2 NMFD

Entrada: Matriz não-negativa $\mathbf{V} \in \mathbb{R}_+^{N \times M}$, número de instrumentos D e número de quadros τ para cada componente.

1. Inicialize o tensor $\mathbf{W} \in \mathbb{R}_+^{N \times D \times \tau}$ e a matriz $\mathbf{H} \in \mathbb{R}_+^{D \times M}$ com valores aleatórios não-negativos distribuídos uniformemente entre 0 e 1;
2. Atualize $\mathbf{W}$ utilizando a equação (3.25);
3. Atualize $\mathbf{H}$ utilizando a equação (3.26);
4. Volte ao passo 2 até atingir a convergência ou um número máximo de iterações.

Saída: Tensor $\mathbf{W} \in \mathbb{R}_+^{N \times D \times \tau}$ e matriz $\mathbf{H} \in \mathbb{R}_+^{D \times M}$.

---

aplicado por exemplo à matriz

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad \text{resulta em} \tag{3.27}$$

$$\overset{\downarrow 0}{\mathbf{A}} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \ \overset{\downarrow 1}{\mathbf{A}} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \ \overset{\downarrow 2}{\mathbf{A}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 2 & 3 \end{bmatrix}, \ \ldots \ . \tag{3.28}$$

Podemos, então, escrever o modelo matemático da NMF2D:

$$\mathbf{V} \approx \mathbf{\Lambda} = \sum_{t=0}^{\tau-1} \sum_{p=0}^{\phi-1} \overset{\downarrow p}{\mathbf{W}^t} \overset{\rightarrow t}{\mathbf{H}^p}, \tag{3.29}$$

onde a matriz $\mathbf{W}^t \in \mathbb{R}_+^{N \times D}$ representa o t-ésimo quadro temporal do padrão espectral de cada instrumento e $\mathbf{H}^p \in \mathbb{R}_+^{D \times M}$ pode ser vista como a descrição de ocorrências da p-ésima nota de cada instrumento. Novamente, N é o número de raias de frequência, M é o número de quadros e D é o número de componentes (nesse caso, instrumentos) a serem separados. O parâmetro τ, como antes, denota o número de quadros que se permite que cada componente dure, enquanto o novo parâmetro ϕ representa o número de possíveis deslocamentos na frequência.

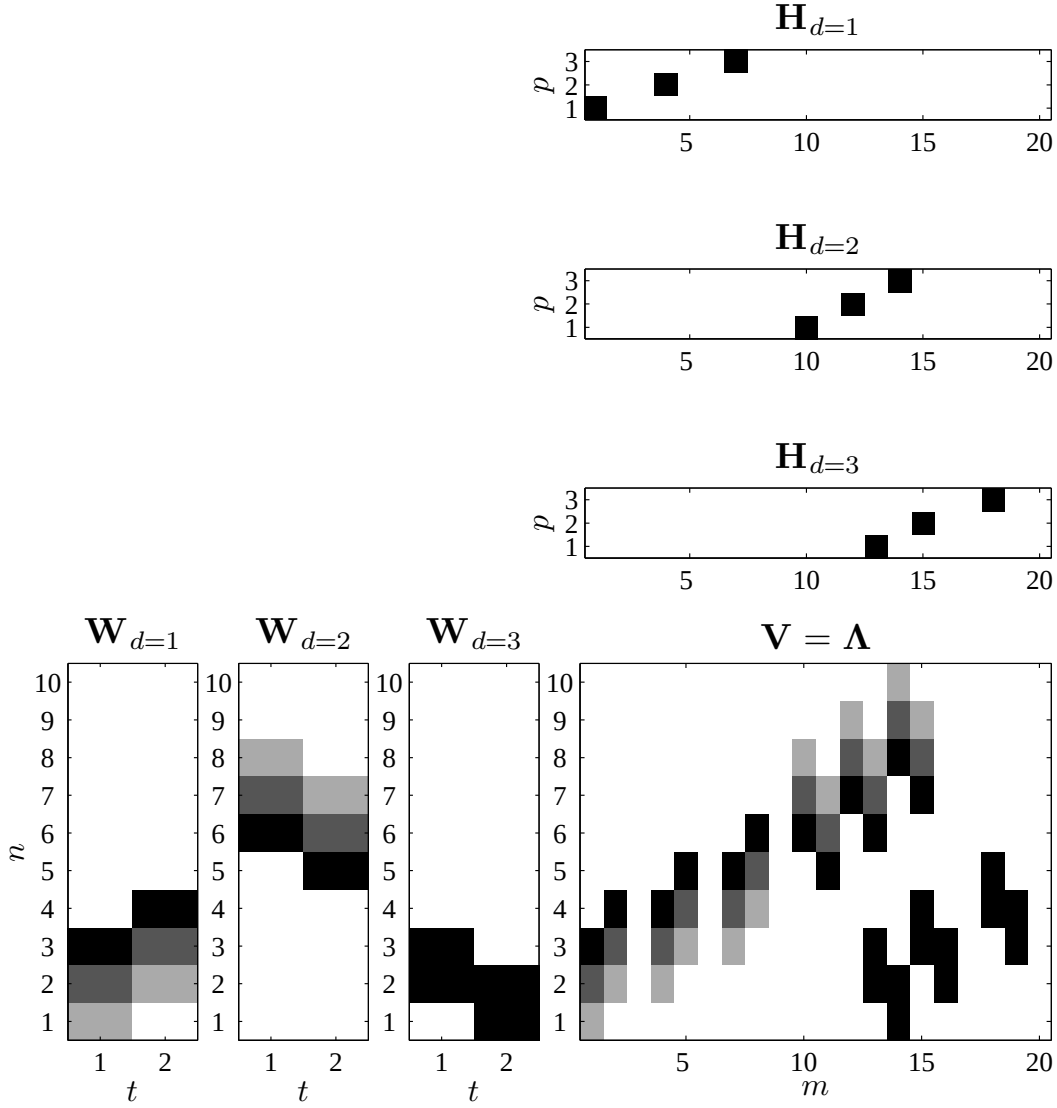


Figura 3.4: Exemplo ilustrativo da NMF2D para $N = 10$, $M = 20$, $D = 3$, $\tau = 2$ e $\phi = 3$. Os três gráficos à esquerda mostram os padrões espectrais relativos aos componentes, com dimensão $N \times \tau$. Os gráficos superiores mostram as ocorrências no tempo e na frequência de cada componente, em matrizes com dimensão $\phi \times M$. Neste exemplo, para melhor visualização, cada componente tem apenas uma ativação unitária para cada deslocamento.

A Figura 3.4 mostra um exemplo ilustrativo para o modelo NMF2D. As matrizes $\mathbf{H}^p$ e $\mathbf{W}^t$ são otimizadas de modo a atingir o mínimo erro de reconstrução. Novamente, a distância euclidiana pode ser usada, mas a divergência de Kullback-Leibler (DKL) da equação (3.14) é a mais indicada para o caso de sinais musicais, como visto no caso da NMF. Os passos do procedimento podem ser vistos no Algoritmo 3.3.

Calculando a derivada da DKL em relação a $\mathbf{H}^p_{n,m}$ e $\mathbf{W}^t_{n,m}$, e escolhendo um passo de minimização tal que obtenhamos uma regra multiplicativa, a atualização

resulta em [13]

$$\mathbf{W}^t \rightarrow \mathbf{W}^t \otimes \frac{\sum_{p=0}^{\phi-1} \left(\frac{\overset{\uparrow p}{\mathbf{V}}}{\mathbf{\Lambda}}\right) \left(\overset{\rightarrow t}{\mathbf{H}^p}\right)^T}{\sum_{p=0}^{\phi-1} \mathbf{1} \otimes \left(\overset{\rightarrow t}{\mathbf{H}^p}\right)^T}, \quad \text{e} \tag{3.30}$$

$$\mathbf{H}^p \rightarrow \mathbf{H}^p \otimes \frac{\sum_{t=0}^{\tau-1} \left(\overset{\downarrow p}{\mathbf{W}^t}\right)^T \left(\frac{\overset{\leftarrow t}{\mathbf{V}}}{\mathbf{\Lambda}}\right)}{\sum_{t=0}^{\tau-1} \left(\overset{\downarrow p}{\mathbf{W}^t}\right)^T \otimes \mathbf{1}}, \tag{3.31}$$

onde $\mathbf{A} \otimes \mathbf{B}$ denota o produto de Hadamard (multiplicação ponto-a-ponto); $\frac{\mathbf{A}}{\mathbf{B}}$ denota a divisão ponto-a-ponto entre as matrizes $\mathbf{A}$ e $\mathbf{B}$; $\mathbf{1}$ é uma matriz $N \times M$ com todos os elementos iguais a 1; e o significado dos operadores de deslocamento para cima e à esquerda pode ser facilmente inferido a partir dos seus correspondentes para baixo e à direita.

O modelo NMF2D considera que cada nota de um instrumento possui o mesmo padrão espectral. Apesar de ser um modelo bem simplificado, esta aproximação pode ser considerada válida num intervalo reduzido de notas. Um aprimoramento deste modelo será proposto na Seção 4.2.

Uma característica da NMF2D é a necessidade da utilização de um espectrograma com espaçamento logarítmico. Isto ocorre em consequência dos operadores de deslocamento vertical. Considerando que um deslocamento vertical unitário para baixo resulta num intervalo constante na escala de temperamento igual, devemos amostrar o espectro de modo que as distâncias entre raias correspondam a um intervalo deste tipo.

Como visto no Capítulo 2, o espectrograma com espaçamento logarítmico pode ser obtido através da CQT ou de algum tipo de mapeamento de escala linear para logarítmica.

O grande problema das soluções que utilizam espectro logaritmicamente espaçado é que, por usarem matrizes de transformação retangulares, não possuem inversa exata para o domínio do tempo, como será visto no Capítulo 5. Desse modo, a aproximação da transformada inversa sempre causará uma distorção na etapa de síntese. O problema da aproximação na reversão ao domínio do tempo é tratado em [23] e [22]. Na Seção 4.1, será apresentada uma proposta de solução para este problema.

A Figura 3.5 mostra um exemplo real de aplicação do algoritmo NMF2D, utili-

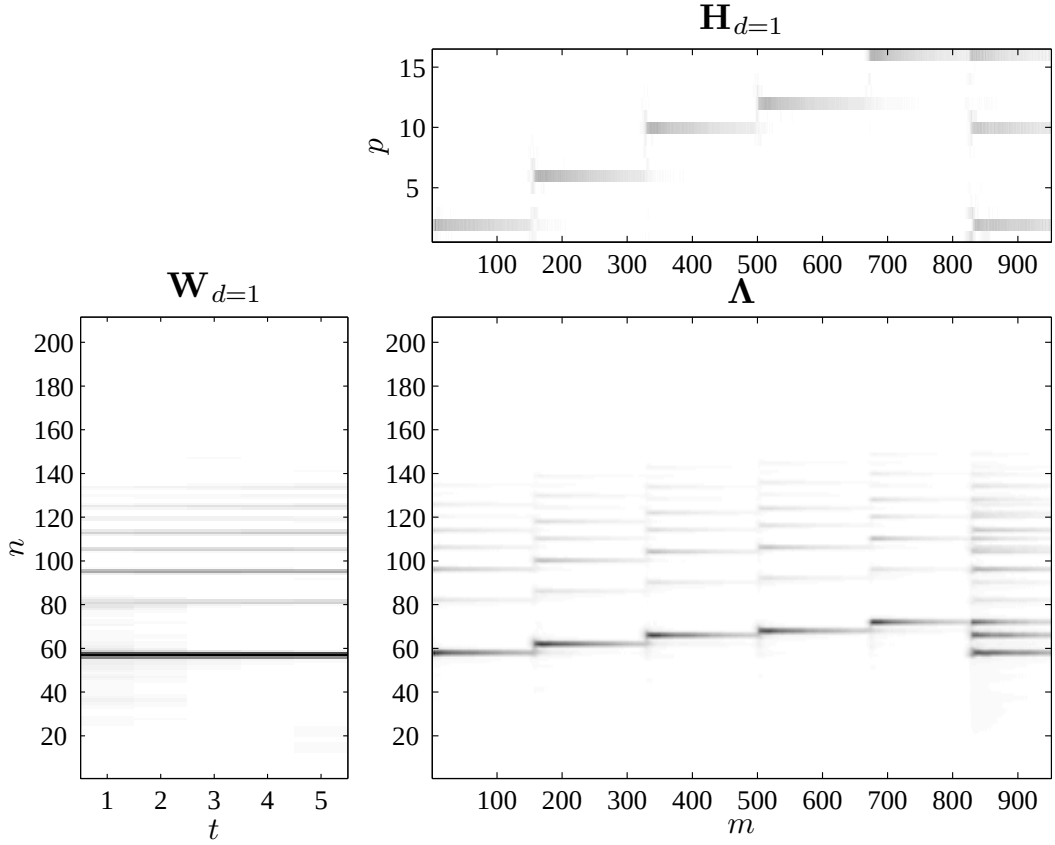


Figura 3.5: Exemplo real da NMF2D para $N = 211$, $M = 961$, $D = 1$, $\tau = 5$ e $\phi = 16$. O gráfico à esquerda mostra o padrão espectral relativo ao componente, com dimensão $N \times \tau$. O gráfico superior mostra as ocorrências no tempo e na frequência do componente, em uma matriz com dimensão $\phi \times M$. Neste exemplo foi utilizado o mesmo sinal da Figura 3.2, que é composto pelas notas de dó a sol tocadas em sequência, seguidas de um acorde de dó maior (dó, mi e sol simultâneos). Como a resolução utilizada é de um quarto de tom, temos quatro saltos a cada tom, e dois a cada semitom. Pode-se notar o intervalo de semitom entre a terceira e quarta notas (mi e fá), e de um tom entre as outras.

zando o mesmo arquivo sonoro da Figura 3.2. O sinal que precisou ser decomposto pela NMF utilizando 5 componentes agora pode ser modelado com apenas um.

Observando o Algoritmo 3.3, podemos calcular o número de multiplicações e somas necessárias. Para o cálculo de $\mathbf{\Lambda}$, temos uma multiplicação de matrizes ($N \times D \times M$) para cada ($\tau \times \phi$). Portanto, temos ($\tau \times \phi$) $\times$ ($N \times M \times D$) multiplicações e $N \times M \times \tau \times \phi \times (2 \times D + 1)$ somas.

A divisão de $\mathbf{\Lambda}$ por $\mathbf{V}$ requer $(N \times M)$ multiplicações e a atualização de $\mathbf{H}$ utiliza $\tau \times (\phi \times N \times M \times D \times 2 + M \times D \times 2)$ multiplicações e $\tau \times (\phi \times N \times M \times D \times 2 + M \times D)$ somas. A atualização de $\mathbf{W}$ precisa de $\phi \times (\tau \times N \times M \times D \times 2 + N \times D \times 2)$ multiplicações e $\phi \times (\tau \times N \times M \times D \times 2 + N \times D)$ somas.

Algoritmo 3.3 NMF2D

Entrada: Matriz não-negativa $\mathbf{V} \in \mathbb{R}_+^{N \times M}$, número de instrumentos D, número de quadros τ e translações possíveis ϕ para cada componente.

1. Inicialize os tensores $\mathbf{W} \in \mathbb{R}_+^{N \times D \times \tau}$ e $\mathbf{H} \in \mathbb{R}_+^{D \times M \times \phi}$ com valores aleatórios não-negativos distribuídos uniformemente entre 0 e 1;
2. Calcule $\mathbf{\Lambda}$ utilizando a equação (3.29);
3. Atualize $\mathbf{W}$ utilizando a equação (3.30);
4. Calcule $\mathbf{\Lambda}$ utilizando a equação (3.29);
5. Atualize $\mathbf{H}$ utilizando a equação (3.31);
6. Volte ao passo 2 até atingir a convergência ou um número máximo de iterações.

Saída: Tensores $\mathbf{W} \in \mathbb{R}_+^{N \times D \times \tau}$ e $\mathbf{H} \in \mathbb{R}_+^{D \times M \times \phi}$.

Portanto, para cada iteração, são necessárias

$$6NMD\tau\phi + NM + 2D(M\tau + N\phi) \quad \text{multiplicações e} \tag{3.32}$$

$$2NM\tau\phi(4D + 1) + D(M\tau + N\phi) \quad \text{somas.} \tag{3.33}$$

3.4 *Non-negative Tensor Factorization* (NTF)

Os métodos de fatoração apresentados até aqui utilizam apenas um canal de áudio, já que trabalham apenas sobre um espectrograma. Em [34], [35], [36] e [37] são apresentadas soluções que possibilitam a utilização de mais canais, caso estejam disponíveis.

O modelo NTF básico relativo à NMF da equação (3.2) pode ser escrito como

$$\mathbf{V}_c \approx \mathbf{\Lambda}_c = \sum_{c=1}^{C} \mathbf{W}\mathbf{g}_c\mathbf{H}, \tag{3.34}$$

onde C é o número de canais, $\mathbf{V} \in \mathbb{R}_+^{C \times N \times M}$ é um tensor contendo o espectrograma de cada canal, $\mathbf{V}_c \in \mathbb{R}_+^{N \times M}$ é o espectrograma de um dos canais, $\mathbf{\Lambda}_c \in \mathbb{R}_+^{N \times M}$ é sua aproximação e $\mathbf{g} \in \mathbb{R}_+^{C \times D}$ é uma matriz contendo o ganho de cada instrumento em cada canal. Os fatores $\mathbf{W} \in \mathbb{R}_+^{N \times D}$ e $\mathbf{H} \in \mathbb{R}_+^{D \times M}$ têm a mesma definição utlizada ao

longo deste capítulo.

É interessante notar que foi adicionada mais uma variável ao problema. A matriz **g** tem a função de descrever o ganho de cada componente nos canais. Desta maneira, podemos reconstruir a imagem estéreo num sinal de dois canais, por exemplo. Uma extensa revisão de métodos de fatoração tensorial pode ser encontrada em [37].

□

Com esta revisão dos diversos métodos inspirados em fatoração de matrizes não-negativas, vimos que a técnica pode ser útil não só para a separação de sinais, mas também para outros tipos de processamento. Com pequeno esforço de processamento, o tensor **H** fornece diretamente as notas musicais, que por sua vez podem ser convertidas para o padrão MIDI (*Musical Instrument Digital Interface*, [38]) e editadas com facilidade.

Também é possível imaginar aplicações que processem o tensor **W**. Podemos, por exemplo, modificar o timbre de um instrumento sem alterar a interpretação do músico, contida na matriz **H**.

Desta forma, vemos que os métodos inspirados em NMF são bastante promissores. Caso se consiga realizar a decomposição de maneira adequada, um sem-número de aplicações se torna possível.

O próximo capítulo apresenta algumas propostas desta dissertação relacionadas aos algoritmos citados neste capítulo. No Capítulo 8, serão mostrados resultados de experimentos relativos aos algoritmos apresentados neste capítulo e no próximo.

Capítulo 4

Contribuições aos Algoritmos de Fatoração

Este capítulo concentra as contribuições desta dissertação. Serão mostrados os fundamentos teóricos referentes a um algoritmo alternativo de fatoração, um método de refinamento e um algoritmo *online*.

Como os resultados da aplicação destas contribuições dependem do sistema inteiro (Figura 1.1), serão mostrados apenas no Capítulo 8, juntamente com os resultados das técnicas da literatura e as devidas comparações, quando aplicáveis.

Novamente, neste capítulo $\mathbf{A} \otimes \mathbf{B}$ denota o produto de Hadamard (multiplicação ponto-a-ponto) e $\frac{\mathbf{A}}{\mathbf{B}}$ denota a divisão ponto-a-ponto, entre as matrizes $\mathbf{A}$ e $\mathbf{B}$.

4.1 *Linear Non-Negative Matrix Factor 2D Deconvolution* (LNMF2D)

O modelo utilizado pelo algoritmo NMF2D supõe que o espectrograma que se deseja decompor tem espaçamento logarítmico. Com isso, deslocamentos verticais na matriz da base espectral resultam em mudanças compatíveis com a escala de temperamento igual.

Por exemplo, suponhamos uma nota de piano representada por um espectrograma de espaçamento logarítmico com resolução de um quarto de tom. Se deslocarmos a matriz que representa o espectrograma duas linhas para cima, teremos a nota um semitom acima.

Apesar da simplicidade do modelo, o espectrograma com espaçamento logarítmico apresenta algumas desvantagens, sendo a maior delas a não reversibilidade exata ao domínio do tempo. Devido à resolução variável, a volta ao domínio do tempo somente pode ser aproximada, como veremos na Seção 5.1. Portanto, para uma aplicação em que seja necessário ressintetizar o sinal após tê-lo modificado, este tipo de espectrograma não se mostra adequado.

Por esse motivo, propomos nesta seção a NMF2D com deslocamento linear (LNMF2D) [39]. Para que possamos usar o modelo NMF2D sem a necessidade de um espectro espaçado de maneira logarítmica, aplicamos um operador de deslocamento compatível com um intervalo na escala de temperamento igual em um espectrograma espaçado linearmente. Para isso, cada raia será deslocada de uma distância diferente, de modo que os novos índices correspondam a intervalos da escala de temperamento igual. A equação (4.1) descreve o operador aplicado aos elementos $W_{n,m}$ de uma matriz $\mathbf{W}$:

$$\overset{\downarrow\downarrow p}{W}_{n2^{p/b},m} = W_{n,m}, \qquad \text{para} \quad n2^{p/b} < N, \tag{4.1}$$

onde b é a resolução desejada. Por exemplo, $b = 12$ resulta em um semitom por deslocamento; $b = 24$ significa um quarto de tom por deslocamento. Operando desta maneira, o parâmetro p segue com o mesmo significado que possuía no modelo da NMF2D: um deslocamento na escala de temperamento igual.

Tal formulação, no entanto, gera um problema: em geral, o valor $n2^{p/b}$ não é inteiro, e portanto não pode ser usado como índice. Caso $n2^{p/b}$ seja arredondado, a operação pode resultar num mapeamento de índices contínuos em novos índices não contínuos. A Tabela 4.1 ilustra a situação.

Tabela 4.1: Mapeamento logarítmico. O operador $\lceil\cdot\rceil$ significa arredondamento.

n	$n \cdot 2^{5/12}$	$\lceil n \cdot 2^{5/12} \rceil$
30	$40,0452$	40
31	$41,3800$	41
32	$42,7149$	43
33	$44,0497$	44
34	$45,3846$	45
35	$46,7194$	47

Este fenômeno resulta da diferença de resolução provocada pela DFT. O uso do espectro linearmente espaçado torna a representação de componentes de alta frequência mais espalhada do que nas de baixas frequências: a maior parte da energia em baixas frequências fica concentrada em poucas raias, enquanto em altas frequências a mesma quantidade de energia fica dividida entre mais raias.

A operação $\downarrow\downarrow p$ entrega toda a energia $12(p/b)$ semitons acima, mas seu espalhamento é irregular, como mostra a Tabela 4.1. Algumas raias recebem mais energia, enquanto outras ficam vazias.

Muitas soluções poderiam ser propostas para resolver o problema de redistribuição entre as raias vazias. Esta dissertação irá adotar uma solução simples, mas que funciona de forma adequada. As raias vazias são preenchidas com a média das raias adjacentes:

$$\overset{\downarrow\downarrow p}{W}_{n-1,m} = \frac{1}{2}(\overset{\downarrow\downarrow p}{W}_{n,m} + \overset{\downarrow\downarrow p}{W}_{n-2,m}), \quad \text{se} \quad \lceil n \cdot 2^{p/b} \rceil - \lceil (n-1) \cdot 2^{p/b} \rceil > 1 \tag{4.2}$$

Devemos notar que essa operação deve ser efetuada após a equação (4.1) ser aplicada a n. Para todo n na equação (4.1), devemos procurar em $n-1$ para verificar se há raias vazias, que então são preenchidas utilizando-se a equação (4.2).

O modelo de reconstrução pode ser reescrito como

$$\mathbf{V} \approx \mathbf{\Lambda} = \sum_{t=0}^{\tau-1} \sum_{p=0}^{\phi-1} \overset{\downarrow\downarrow p}{\mathbf{W}}{}^t \overset{\rightarrow t}{\mathbf{H}}{}^p . \tag{4.3}$$

As regras de atualização podem ser escritas de forma direta, observando-se as equações (3.30) e (3.31):

$$\mathbf{W}^t \;\rightarrow\; \mathbf{W}^t \otimes \frac{\sum_{p=0}^{\phi-1} (\overset{\uparrow\uparrow p}{\frac{\mathbf{V}}{\mathbf{\Lambda}}}) \left(\overset{\rightarrow t}{\mathbf{H}}{}^p \right)^T}{\sum_{p=0}^{\phi-1} \mathbf{1} \cdot \left(\overset{\rightarrow t}{\mathbf{H}}{}^p \right)^T} \quad \text{e} \tag{4.4}$$

$$\mathbf{H}^p \;\rightarrow\; \mathbf{H}^p \otimes \frac{\sum_{t=0}^{\tau-1} \left(\overset{\downarrow\downarrow p}{\mathbf{W}}{}^t \right)^T (\overset{\leftarrow t}{\frac{\mathbf{V}}{\mathbf{\Lambda}}})}{\sum_{t=0}^{\tau-1} \left(\overset{\downarrow\downarrow p}{\mathbf{W}}{}^t \right)^T \cdot \mathbf{1}}. \tag{4.5}$$

Devemos lembrar que o problema da fatoração não-negativa é convexo em $\mathbf{W}$ ou

em **H**, mas não em ambos [11]. Isto significa que a otimização, tanto na LNMF2D quanto na NMF2D, deve ser feita de maneira alternada, isto é, uma vez que **W** ou **H** é atualizado, a função-custo deve ser novamente calculada antes da próxima atualização de **H** ou **W**, respectivamente. Desta maneira, a função-objetivo irá sempre decrescer em relação à penúltima iteração.

A Equação (4.2) mostra que o operador $\downdownarrows$ pode alterar a quantidade de energia no espectrograma. Embora o formato geral das ressonâncias não seja afetado pela operação de média, as razões de energia podem se alterar. Este problema é inerente à solução simples adotada, e deve ser reexaminado em trabalhos futuros.

A Figura 4.1 mostra um exemplo de decomposição realizada utilizando o algoritmo LNMF2D, e o Algoritmo 4.1 descreve os passos do procedimento. Na Seção 8.2, serão apresentados resultados comparativos entre a LNMF2D e NMF2D.

Algoritmo 4.1 LNMF2D

Entrada: Matriz não-negativa $\mathbf{V} \in \mathbb{R}_+^{N \times M}$, número de instrumentos D, número de quadros τ e translações possíveis ϕ para cada componente.

1. Inicialize os tensores $\mathbf{W} \in \mathbb{R}_+^{N \times D \times \tau}$ e $\mathbf{H} \in \mathbb{R}_+^{D \times M \times \phi}$ com valores aleatórios não-negativos distribuídos uniformemente entre 0 e 1;
2. Calcule $\mathbf{\Lambda}$ utilizando a equação (4.3);
3. Atualize $\mathbf{W}$ utilizando a equação (4.4);
4. Calcule $\mathbf{\Lambda}$ utilizando a equação (4.3);
5. Atualize $\mathbf{H}$ utilizando a equação (4.5);
6. Volte ao passo 2 até atingir a convergência ou um número máximo de iterações.

Saída: Tensores $\mathbf{W} \in \mathbb{R}_+^{N \times D \times \tau}$ e $\mathbf{H} \in \mathbb{R}_+^{D \times M \times \phi}$.

O número de operações em função dos parâmetros é o mesmo apresentado para a NMF2D, nas equações (3.32) e (3.33). Entretanto, devemos notar que devido às diferenças nos métodos de análise tempo-frequência, os valores de N e M tendem a ser diferentes entre espectrogramas logarítmicos e lineares. Entretanto, é possível que o produto NM seja próximo para os dois tipos de espectrogramas. A STFT sempre terá mais raias (N) do que a CQT, mas como esta possui menor salto entre quadros, seu número de quadros (M) tende a ser maior.

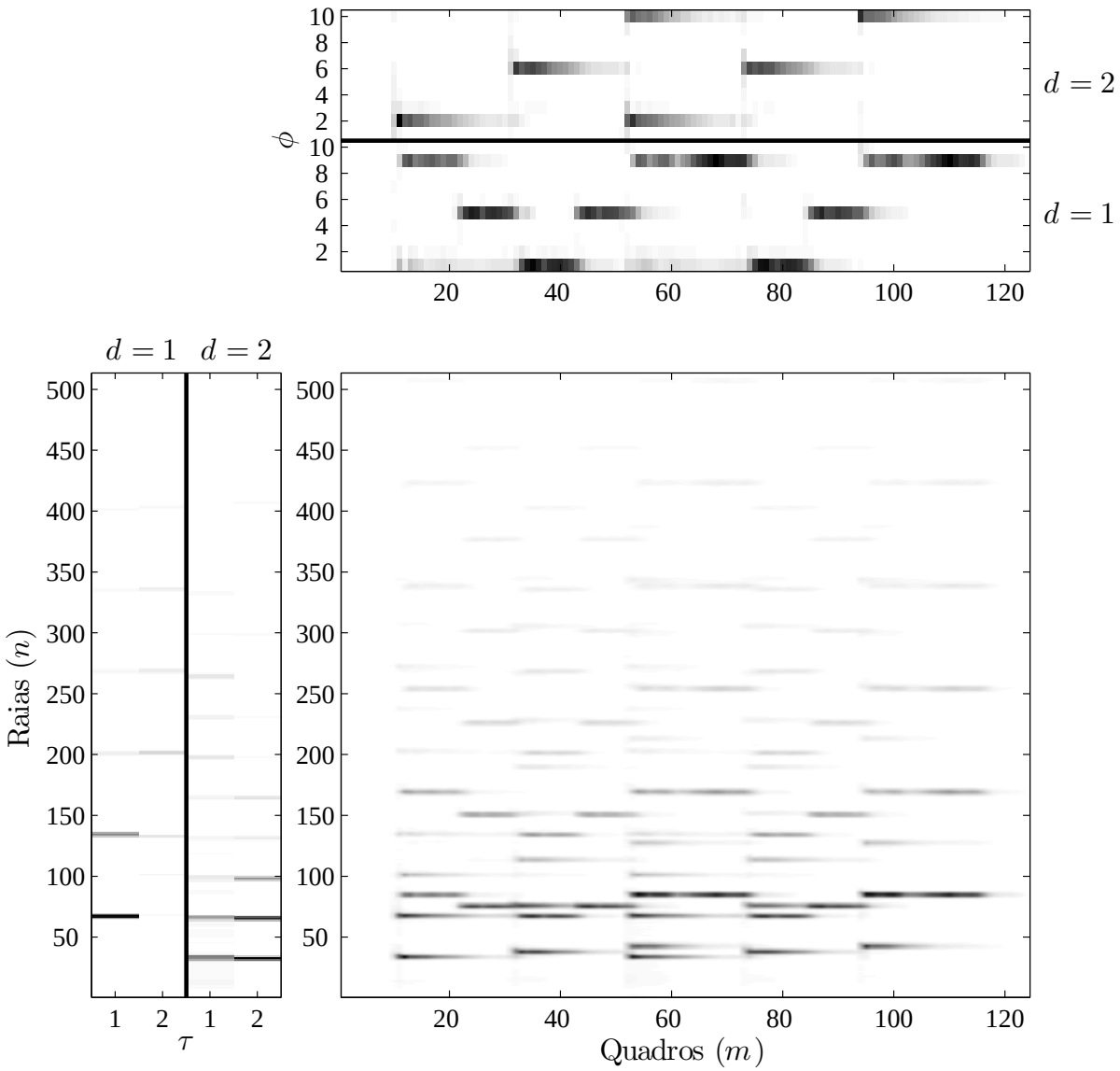


Figura 4.1: Resultados da decomposição utilizando a LNMF2D. O gráfico mais à esquerda mostra os vetores da base espectral encontrados para cada instrumento. O gráfico acima mostra onde no tempo e no *pitch* os vetores da base aparecem. O gráfico à esquerda embaixo mostra o espectrograma resultante, ou seja, a matriz $\mathbf{\Lambda}$. O sinal analisado é descrito na partitura da Figura 2.3, que pode ser comparada com a matriz $\mathbf{H}$.

Portanto, o benefício da inversão direta ao domínio do tempo deve ser observado considerando-se uma possível mudança no número de operações.

4.2 Adaptação da Base Espectral

Como já foi citado, um dos problemas da NMF2D é assumir que o espectro de um instrumento é invariante à nota, ou seja, que cada mudança de nota corresponde apenas a um deslocamento dos vetores da base espectral ao longo frequência. Este modelo pode até ser considerado uma aproximação válida para pequenos intervalos, mas para intervalos maiores o erro é muito grande. Em certos instrumentos, o próprio método de geração do som muda ao longo das notas; um exemplo é o piano, em que notas graves são geradas por uma corda, as intermediárias por duas e as mais agudas por 3.

Desta maneira, faz-se necessário algum ajuste para que os vetores da base espectral de um instrumento possam se adaptar às especificidades de cada nota, mantendo uma estrutura geral que faz dela a representação de um instrumento musical. Ao incorporar esta adaptação, o modelo (partindo da NMF2D) pode ser escrito como

$$\mathbf{V} \approx \mathbf{\Lambda} = \sum_{t=0}^{\tau-1} \sum_{p=0}^{\phi-1} (\overset{\downarrow p}{\mathbf{W}^t} \otimes \mathbf{M}^{t,p}) \overset{\rightarrow t}{\mathbf{H}^p}, \tag{4.6}$$

onde $\mathbf{M}$ será chamado tensor de adaptação. Este tensor possui quatro dimensões: $N \times \tau \times \phi \times D$. Isto significa que haverá uma matriz de adaptação ($N \times \tau$, mesmas dimensões de $\mathbf{W}^t$) para cada nota ($p = 0, \ldots, \phi - 1$) de cada instrumento ($d = 0, \ldots, D - 1$), ou seja: a cada vez que uma determinada nota aparecer no modelo, ela sofrerá uma adaptação multiplicativa ponto-a-ponto de modo a diminuir o erro cometido. Devemos observar que, caso todos os elementos de $\mathbf{M}$ sejam unitários, o modelo volta a ser a NMF2D tradicional.

Como a multiplicação é feita ponto-a-ponto para cada instância da nota, o modelo se ajusta bem ao espectrograma com espaçamento tanto linear, quanto logarítmico. A atualização de $\mathbf{M}$ pode ser feita usando qualquer critério de reconstrução, como por exemplo a distância euclidiana. Utilizando-se o modelo NMF2D, podemos

escrever um elemento de $\mathbf{\Lambda}$ como

$$\mathbf{\Lambda}(n,m) = \sum_{d=1}^{D} \sum_{t=0}^{\tau-1} \sum_{p=0}^{\phi-1} \mathbf{W}(n-p,d,t)\mathbf{M}(n,t,p,d)\mathbf{H}(d,m-t,p). \tag{4.7}$$

O erro quadrático é representado por

$$E = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} (\mathbf{V}(n,m) - \mathbf{\Lambda}(n,m))^2. \tag{4.8}$$

Queremos encontrar $\mathbf{M}$ tal que o erro de reconstrução (4.8) seja o menor possível. Para isso, calculamos a derivada do erro em relação a cada elemento de $\mathbf{M}$:

$$\frac{\partial E}{\partial \mathbf{M}(n',t',p',d')} = -2 \sum_{m=1}^{M} (\mathbf{V}(n',m) - \mathbf{\Lambda}(n',m)) \frac{\partial \mathbf{\Lambda}(n',m)}{\partial \mathbf{M}(n',t',p',d')} \tag{4.9}$$

$$\frac{\partial \mathbf{\Lambda}(n',m)}{\partial \mathbf{M}(n',t',p',d')} = \mathbf{W}(n'-p',d',t')\mathbf{H}(d',m-t',p') \tag{4.10}$$

$$\frac{\partial E}{\partial \mathbf{M}(n',t',p',d')} = -2 \sum_{m=1}^{M} [(\mathbf{V}(n',m) - \mathbf{\Lambda}(n',m)) \mathbf{W}(n'-p',d',t')\mathbf{H}(d',m-t',p')]. \tag{4.11}$$

Com isso, a equação de atualização de $\mathbf{M}$ pode ser escrita como

$$\mathbf{M}(n',t',p',d') \leftarrow \mathbf{M}(n',t',p',d') - \mu \frac{\partial E}{\partial \mathbf{M}(n',t',p',d')}, \tag{4.12}$$

onde μ é o passo de atualização, que deve ser suficientemente pequeno para que a convergência seja atingida. Agora devemos calcular μ tal que a atualização seja multiplicativa, do mesmo modo que na Subseção 3.1.1. Fazendo

$$\mu = \frac{\mathbf{M}(n',t',p',d')}{2 \sum_{m=1}^{M} \mathbf{\Lambda}(n',m)\mathbf{W}(n'-p',d',t')\mathbf{H}(d',m-t',p')} \tag{4.13}$$

e substituindo (4.13) em (4.12), chegamos em

$$\mathbf{M}(n',t',p',d') \leftarrow \mathbf{M}(n',t',p',d') + \\ 2\mathbf{M}(n',t',p',d')\frac{\left(\begin{array}{c}\sum_{m=1}^{M}\mathbf{V}(n',m)\mathbf{W}(n'-p',d',t')\mathbf{H}(d',m-t',p') - \\ \sum_{m=1}^{M}\mathbf{\Lambda}(n',m)\mathbf{W}(n'-p',d',t')\mathbf{H}(d',m-t',p')\end{array}\right)}{2\sum_{m=1}^{M}\mathbf{\Lambda}(n',m)\mathbf{W}(n'-p',d',t')\mathbf{H}(d',m-t',p')}, \quad (4.14)$$

ou

$$\mathbf{M}(n',t',p',d') \leftarrow \mathbf{M}(n',t',p',d') \\ \left(1+\frac{\sum_{m=1}^{M}\mathbf{V}(n',m)\mathbf{W}(n'-p',d',t')\mathbf{H}(d',m-t',p')}{\sum_{m=1}^{M}\mathbf{\Lambda}(n',m)\mathbf{W}(n'-p',d',t')\mathbf{H}(d',m-t',p')}-1\right), \quad (4.15)$$

isto é,

$$\mathbf{M}(n',t',p',d') \leftarrow \mathbf{M}(n',t',p',d')\frac{\sum_{m=1}^{M}\mathbf{V}(n',m)\mathbf{W}(n'-p',d',t')\mathbf{H}(d',m-t',p')}{\sum_{m=1}^{M}\mathbf{\Lambda}(n',m)\mathbf{W}(n'-p',d',t')\mathbf{H}(d',m-t',p')}.$$

Como $\mathbf{W}$ não depende de m, a atualização se reduz a

$$\mathbf{M}(n,t,p,d) \leftarrow \mathbf{M}(n,t,p,d)\frac{\sum_{m=1}^{M}\mathbf{V}(n',m)\mathbf{H}(d',m-t',p')}{\sum_{m=1}^{M}\mathbf{\Lambda}(n',m)\mathbf{H}(d',m-t',p')}. \quad (4.16)$$

Caso se deseje utilizar a divergência de Kullback-Leibler como medida de reconstrução, temos outra derivação para a atualização de $\mathbf{M}$. Relembrando a expressão da divergência de Kullback-Leibler

$$D_{\mathrm{KL}} = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1}\left(\mathbf{V}(n,m)\ln\frac{\mathbf{V}(n,m)}{\mathbf{\Lambda}(n,m)} - \mathbf{V}(n,m) + \mathbf{\Lambda}(n,m)\right), \quad (4.17)$$

a derivada de D_{KL} em relação a um elemento do tensor $\mathbf{M}$ é

$$\frac{\partial D_{\mathrm{KL}}}{\partial \mathbf{M}(n',t',p',d')} = \frac{\partial\sum_{m=0}^{M-1}\mathbf{V}(n',m)\ln\frac{\mathbf{V}(n',m)}{\mathbf{\Lambda}(n',m)}}{\partial \mathbf{M}(n',t',p',d')} + \frac{\partial\sum_{m=0}^{M-1}(\mathbf{\Lambda}(n',m)}{\partial \mathbf{M}(n',t',p',d')}. \quad (4.18)$$

Avaliando a primeira parcela da soma, temos

$$\frac{\partial \sum_{m=0}^{M-1} \mathbf{V}(n',m) \ln \frac{\mathbf{V}(n',m)}{\mathbf{\Lambda}(n',m)}}{\partial \mathbf{M}(n',t',p',d')} = \\ - \sum_{m=1}^{M} \frac{\mathbf{V}(n',m)}{\mathbf{\Lambda}(n',m)} \mathbf{W}(n'-p',d',t') \mathbf{H}(d',m-t',p'); \quad (4.19)$$

a segunda parcela é

$$\frac{\partial \sum_{m=0}^{M-1} \mathbf{\Lambda}(n',m)}{\partial \mathbf{M}(n',t',p',d')} = \sum_{m=1}^{M} \mathbf{W}(n'-p',d',t') \mathbf{H}(d',m-t',p'). \quad (4.20)$$

Com isso, a atualização de $\mathbf{M}$ pode ser escrita como

$$\mathbf{M}(n',t',p',d') \leftarrow \mathbf{M}(n',t',p',d') - \\ \mu \left(- \sum_{m=1}^{M} \frac{\mathbf{V}(n',m)}{\mathbf{\Lambda}(n',m)} \mathbf{W}(n'-p',d',t') \mathbf{H}(d',m-t',p') + \right. \\ \left. \sum_{m=1}^{M} \mathbf{W}(n'-p',d',t') \mathbf{H}(d',m-t',p') \right), \quad (4.21)$$

onde μ deve ser calculado de forma a se obter a atualização multiplicativa:

$$\mu = \frac{\mathbf{M}(n',t',p',d')}{\sum_{m=1}^{M} \mathbf{W}(n'-p',d',t') \mathbf{H}(d',m-t',p')}. \quad (4.22)$$

Com isso, a atualização multiplicativa resulta em

$$\mathbf{M}(n',t',p',d') = \mathbf{M}(n',t',p',d') \frac{\sum_{m=1}^{M} \frac{\mathbf{V}(n',m)}{\mathbf{\Lambda}(n',m)} \mathbf{H}(d',m-t',p')}{\sum_{m=1}^{M} \mathbf{H}(d',m-t',p')}. \quad (4.23)$$

□

O uso da adaptação espectral estrutura o algoritmo em dois passos: primeiro, a decomposição utilizando NMF2D ou LNMF2D, e em seguida a adaptação espectral. Este segundo passo funciona como um refinamento do algoritmo anterior, de modo que caso a primeira etapa não tenha bom resultado, sua utilização não trará melhorias significativas. A adaptação espectral supõe uma fatoração razoável, onde houve ao menos o reconhecimento das notas no tensor $\mathbf{H}$.

Este algoritmo deve ser aplicado entre os blocos de *Fatoração* e *Processamento*

do diagrama da Figura 1.1. Ele recebe os tensores $\mathbf{W}$ e $\mathbf{H}$ como entrada, e deve passar ao bloco seguinte o tensor de adaptação $\mathbf{M}$. O procedimento completo está sistematizado no Algoritmo 4.2.

Algoritmo 4.2 Adaptação espectral

Entrada: Tensores $\mathbf{W} \in \mathbb{R}_+^{N\times D\times \tau}$ e $\mathbf{H} \in \mathbb{R}_+^{D\times M\times \phi}$, que podem ser a saída do Algoritmos 3.3 ou 4.1.

1. Inicialize o tensor $\mathbf{M} \in \mathbb{R}_+^{N\times \tau\times \phi\times D}$ com todos os elementos unitários;
2. Atualize $\mathbf{M}$ pela equação (4.16) para utilizar a distância euclidiana, ou pela equação (4.23) para usar a divergência de Kullback-Leibler;
3. Calcule $\mathbf{\Lambda}$ pela equação (4.6) e verifique o erro pela equação (4.8);
4. Volte ao passo 2 até atingir a convergência ou um número máximo de iterações.

Saída: Tensor $\mathbf{M} \in \mathbb{R}_+^{N\times \tau\times \phi\times D}$.

A Seção 8.3 mostrará os resultados da aplicação da adaptação espectral aos algoritmos NMF2D e LNMF2D. Será analisada a curva do erro de reconstrução e a melhora na qualidade do sinal sintetizado.

Por fim, devemos notar que o uso do algoritmo sempre aumenta a complexidade do sistema, já que a adaptação espectral insere uma etapa a mais de processamento e não altera as anteriores. Analisando a equação (4.16), vemos que cada iteração do Algoritmo 4.2 consome:

$$(2M+2)DN\tau\phi \quad \text{multiplicações e} \tag{4.24}$$

$$2MDN\tau\phi \qquad \text{somas,} \tag{4.25}$$

além do custo do algoritmo de fatoração. Portanto, seus benefícios devem ser ponderados à luz da complexidade adicional trazida pela solução.

Para deixar mais clara a diferença de complexidade entre os algoritmos, vamos atribuir valores aos parâmetros e calcular o número de multiplicações e somas para os algoritmos NMF2D e LNMF2D, ambos utilizando a adaptação espectral. Por exemplo, consideremos a aplicação:

$$D = 3, \phi = 10, \tau = 4, N_{\text{STFT}} = 1024, M_{\text{STFT}} = 124,$$
$$N_{\text{CQT}} = 101 \quad \text{e} \quad M_{\text{CQT}} = 976,$$

onde N_{STFT} e M_{STFT} são o número de raias e quadros utilizados na STFT, e N_{CQT} e M_{CQT} são o número de raias e quadros utilizados pela CQT.

Com isso, teremos a cada iteração da NMF2D cerca de

$$71 \times 10^6 \quad \text{multiplicações e}$$
$$102 \times 10^6 \quad \text{somas,}$$

além de

$$24 \times 10^6 \quad \text{multiplicações e}$$
$$24 \times 10^6 \quad \text{somas}$$

a cada iteração do algoritmo de adaptação espectral. No caso da LNMF2D, teremos cerca de

$$91 \times 10^6 \quad \text{multiplicações e}$$
$$132 \times 10^6 \quad \text{somas}$$

por iteração para o algoritmo de fatoração, além de

$$31 \times 10^6 \quad \text{multiplicações e}$$
$$30 \times 10^6 \quad \text{somas}$$

a cada iteração do algoritmo de adaptação espectral.

Vemos que a LNMF2D gasta mais operações devido ao maior número de elementos na matriz $\mathbf{V} \in \mathbb{R}^{N \times M}$. Entretanto, nas simulações realizadas no Capítulo 8, não percebemos diferença notável entre os tempos de execução dos algoritmos de fatoração. A opção pelo uso da adaptação espectral, no entanto, aumenta consideravelmente o tempo de execução nos dois casos (no exemplo acima, cerca 30% nas multiplicações).

4.3 Processamento *Online*

Um problema pouco abordado na literatura sobre os métodos de decomposição espectral é a aplicação a sinais musicais de longa duração. Como vimos ao longo deste capítulo e do anterior, os espectrogramas são tratados na forma de matrizes. Supondo um sinal de áudio de 3 minutos, amostrado a 44,1kHz, com janelas de 1024 amostras e sobreposição de 50%, teremos uma matriz com mais de 15000 colunas. Supondo a análise com 2048 pontos, teremos ainda 1024 linhas (metade do espectro), o que nos dá mais de 5 milhões de elementos, que representados com 32 bits nos levam a uma matriz com cerca de 60 MB.

Como se vê, não é difícil que o problema se torne computacionalmente pesado, por requerer a manipulação de estruturas de dados muito extensas. Além disso, o processamento do sinal inteiro impõe atraso igual a, no mínimo, o tamanho do sinal. Um possível contorno para este problema é a realização do processamento em blocos.

A divisão em blocos não diminui o número de operações, pelo contrário: pode aumentar o número total de operações caso haja sobreposição entre os blocos. Sua vantagem, no entanto, consiste em (1) usar menos memória, já que as matrizes a serem armazenadas serão menores, e (2) possibilitar o processamento sem que se disponha do sinal completo (processamento *online*).

Considerando que serão feitas várias análises sobre blocos do mesmo sinal, é possível acelerar o processo em cada bloco, diminuindo o número de iterações necessárias. Caso o número de instrumentos esteja fixado e os padrões espectrais estejam otimizados, podemos utilizar o tensor $\mathbf{W}$ na inicialização ou até mesmo fixá-lo nos blocos seguintes.

Neste algoritmo, iremos assumir que um estimador de número de fontes é utilizado, determinando o número máximo de instrumentos que será considerado ao longo de todo o sinal. A mistura original y é dividida em blocos com sobreposição. É desejável que o bloco tenha tamanho múltiplo do tamanho da janela de análise. O sistema de separação inteiro (Figura 1.1) é então aplicado a cada bloco, desde a análise até a síntese, passando naturalmente pela fatoração. Os sinais separados resultantes são então sobrepostos e somados, para gerar as fontes contidas na mistura y. Não é necessário aplicar um janelamento aos blocos, pois isso já é feito na etapa

de análise tempo-frequência.

É importante notar que o tensor $\mathbf{W}$ otimizado no bloco anterior será usado como estimativa inicial para o próximo bloco. Este procedimento é imprescindível para que se preserve a ordem das fontes; ele tenta garantir que uma mesma fonte mantenha sua ordenação ao longo dos blocos. Caso contrário, a junção dos blocos ao final do algoritmo seria dificultada. Além disso, espera-se aumentar a velocidade de convergência com o tensor $\mathbf{W}$ inicializado convenientemente.

Caso haja algum problema de convergência em um dos blocos, é possível que ocorra troca da ordenação das fontes. Entretanto, a inicialização utilizando a base espectral já otimizada também coopera para que a convergência ocorra de maneira correta. O Algoritmo 4.3 descreve o procedimento de forma sistemática, e os resultados serão apresentados na Seção 8.5.

Algoritmo 4.3 Algoritmo *Online*

Entrada: Sinal y da mistura, no domínio do tempo, número de instrumentos D, número de quadros τ e translações possíveis ϕ para cada componente.

1. Divida o sinal y em blocos de tamanho B, gerando os blocos de sinal y_b;
2. Para todos os blocos y_b:
   (a) Aplique o sistema de separação da Figura 1.1 ao bloco y_b, utilizando o tensor $\mathbf{W}$ da iteração anterior como estimativa inicial;
   (b) Salve o tensor $\mathbf{W}$ e as fontes separadas sintetizadas;
3. Sobreponha e some a fontes separadas de cada bloco para gerar o sinal completo das fontes.

Saída: Sinais separados.

□

Com estas contribuições, pretende-se superar alguns obstáculos ao bom funcionamento dos algoritmos de NMF. A motivação para os desenvolvimentos veio das dificuldades encontradas ao implementar trabalhos da literatura. Cabe ressaltar que a aplicação destas técnicas à separação de sinais de áudio encontra-se em fase inicial de desenvolvimento, e o espaço para novas propostas ainda é vasto. Os resultados apresentados no Capítulo 8, apesar de evidenciarem o estágio inicial da pesquisa, mostram-se promissores.

Capítulo 5

Processamento

Seguindo o diagrama de blocos da Figura 1.1, após o bloco de fatoração, discutido nos Capítulos 3 e 4, encontramos o bloco de *Processamento*. Ele recebe como entrada os fatores, isto é, os tensores **W** e **H** obtidos no bloco de *Fatoração*, e os processa de modo a obter os espectrogramas lineares das fontes separadas. Estes espectrogramas serão convertidos ao domínio do tempo em seguida, no bloco de *Síntese*, descrito no Capítulo 6.

Neste ponto, seria possível imaginar a implementação de diversas aplicações. Por exemplo, podemos processar o tensor **H** de modo a mudar a melodia da peça musical, ou processar o tensor **W** para alterar o timbre dos instrumentos. Além disso, o tensor **H** pode ser utilizado para transcrição da fonte em partitura musical. Contudo, essas abordagens fogem ao escopo desta dissertação.

Neste capítulo, mostraremos apenas dois tipos de processamento que podem ser feitos sobre o espectrograma dos sinais separados. O primeiro deles se refere à reconstrução dos espectrogramas a partir dos fatores, o que inclui a transformação dos espectrogramas com espaçamento logarítmico em lineares. Esta etapa é necessária quando se usa a NMF2D, e deve ser feita de forma a minimizar as distorções causadas pela aproximação. O segundo trata de melhoramentos que podem ser aplicados ao espectrograma, utilizando-se o espectrograma da mistura. Estas melhorias devem se refletir em maior qualidade no sinal após a síntese.

5.1 Reconstrução

Após a convergência do algoritmo de fatoração, o próximo passo é obter os espectrogramas das fontes separadas. Para isso, podemos simplesmente utilizar as equações que descrevem os modelos (3.2), (3.24), (3.29), (4.3) ou (4.6), e fixar um valor para d. Usaremos o modelo da LNMF2D como exemplo, ressaltando que o desenvolvimento é válido para todos os algoritmos apresentados nos Capítulos 3 e 4, salvo diferenças que serão ressaltadas no texto.

Na LNMF2D podemos obter o espectrograma relativo a um componente d fazendo

$$\mathbf{\Lambda}_d = \sum_{t=1}^{\tau} \sum_{p=1}^{\phi} \mathbf{W}_d^t \overset{\downarrow p}{} \mathbf{H}_d^p \overset{\rightarrow t}{}, \tag{5.1}$$

onde $\mathbf{\Lambda}_d$ é o espectrograma referente à d-ésima fonte, $1 \leq d \leq D$, $\mathbf{W}_d^t$ é um vetor-coluna de tamanho N e $\mathbf{H}_d^p$ é um vetor-linha de tamanho M.

Neste ponto, devemos salientar uma diferença importante entre os algoritmos. Caso o espectrograma fatorado seja linearmente espaçado, como mostrado na Seção 2.1, a matriz $\mathbf{\Lambda}_d$ já pode ser diretamente convertida ao domínio do tempo através dos métodos que serão discutidos no Capítulo 6. No entanto, caso o espectrograma seja logaritmicamente espaçado, como mostrado na Seção 2.2, é preciso processá-lo de forma que suas raias se tornem linearmente espaçadas na frequência. O espectrograma com espaçamento logarítmico é obrigatoriamente utilizado na NMF2D.

A transformação do espectrograma logarítmico em linear será sempre uma aproximação. Podemos ver isto analisando os dois métodos apresentados para obtenção do espectrograma logarítmico.

No método da DFT com mapeamento para escala logarítmica apresentado na Seção 2.3, podemos ver claramente que a transformação acarreta em aproximação. Por conveniência, reproduzimos aqui a equação (2.9):

$$\mathbf{V}_{\text{log}} = \mathbf{C}\mathbf{V}_{\text{lin}}, \tag{5.2}$$

onde $\mathbf{C} \in \mathbb{R}^{N_{\text{log}} \times N_{\text{lin}}}$ é a matriz de mapeamento, $\mathbf{V}_{\text{lin}} \in \mathbb{R}^{N_{\text{lin}} \times M}$ é espectrograma linear obtido pela STFT e $\mathbf{V}_{\text{log}} \in \mathbb{R}^{N_{\text{log}} \times M}$ espectrograma logarítmico. A matriz $\mathbf{C}$

é em geral retangular, e portanto não possui inversa exata.

No caso do espectrograma gerado pela CQT, podemos usar o mesmo argumento se representarmos a transformada matricialmente. Uma implementação rápida da CQT [20], [22] pode ser escrita como

$$\mathbf{X} = \mathbf{T}x, \tag{5.3}$$

onde $\mathbf{X}$ é o espectrograma gerado pela CQT, x é o vetor com as amostras do sinal a ser analisado e os elementos de $\mathbf{T}$ são

$$T_{nm} = \begin{cases} \frac{1}{N_m} w_{N_m}[n] \exp^{-j\frac{2\pi n}{N_m}Q}, & \text{para } n < N_m \\ 0, & \text{caso contrário} \end{cases}, \tag{5.4}$$

onde w_{N_m} é a janela para a raia m, que tem comprimento N_m, e x é o sinal no domínio do tempo. Podemos então escrever a CQT como

$$\mathbf{X} = \mathbf{C}\mathbf{Y}, \tag{5.5}$$

onde $\mathbf{C}$ é a matriz obtida aplicando-se a DTF a cada coluna de $\mathbf{T}$, e $\mathbf{Y}$ é a DFT de x. Novamente, percebemos que a matriz $\mathbf{C}$ não é quadrada, e portanto não possui inversa exata. Em [23], três possíveis soluções são apresentadas para este problema:

- **Pela pseudoinversa**

  $$\hat{\mathbf{\Lambda}}_d^{\text{lin}} = \mathbf{C}^{+}\mathbf{\Lambda}_d^{\text{log}}, \tag{5.6}$$

  onde $\hat{\mathbf{\Lambda}}_d^{\text{lin}}$ é a aproximação do espectrograma linear, $\mathbf{C}^{+}$ é a pseudoinversa de Moore-Penrose de $\mathbf{C}$ e $\mathbf{\Lambda}_d^{\text{log}}$ é o espectrograma logarítmico. Esta é a melhor aproximação para uma inversa de $\mathbf{C}$, no sentido de mínimos quadrados, entretanto pode resultar em número negativos para $\hat{\mathbf{\Lambda}}_d^{\text{log}}$, o que não faz sentido quando representamos um espectrograma de magnitude.

- **Pela transposta**

  $$\hat{\mathbf{\Lambda}}_d^{\text{lin}} = \mathbf{C}^{T}\mathbf{\Lambda}_d^{\text{log}}, \tag{5.7}$$

  onde $\mathbf{C}^{T}$ é a transposta $\mathbf{C}$. Esta aproximação tem a vantagem de garantir a não-negatividade de $\hat{\mathbf{\Lambda}}_d^{\text{lin}}$. Entretanto, o espectrograma resultante terá escalamento diferente do original.

- **Iterativa**

$$\hat{\mathbf{\Lambda}}_d^{\text{lin}} \leftarrow \hat{\mathbf{\Lambda}}_d^{\text{lin}} \otimes \frac{\mathbf{C}\mathbf{\Lambda}_d^{\log}}{\mathbf{C}^T\mathbf{C}\mathbf{\Lambda}_d^{\log}}, \tag{5.8}$$

  onde $\hat{\mathbf{\Lambda}}_d^{\text{lin}}$ é inicializado aleatoriamente e a Equação (5.8) é aplicada até a convergência de $\hat{\mathbf{\Lambda}}_d^{\text{lin}}$. Este método atinge a melhor solução não-negativa no sentido de mínimos quadrados, mas é computacionalmente mais intenso do que os anteriores.

5.2 Filtragem do Espectrograma Separado

Com um espectrograma linearmente amostrado das fontes separadas, já poderíamos partir para a etapa de síntese, que será descrita no próximo capítulo. Entretanto, o espectrograma da mistura pode nos ajudar a refinar a estimativa do espectrograma de cada de fonte. Abaixo, mostramos quatro técnicas de refinamento apresentadas em [23]:

5.2.1 Mascaramento Espectral

O método de mascaramento espectral simplesmente multiplica ponto-a-ponto a estimativa do espectrograma separado pelo da mistura, procurando ressaltar as características de $\mathbf{\Lambda}_d$ presentes em $\mathbf{V}$:

$$\hat{\mathbf{\Lambda}}_d = \mathbf{\Lambda}_d \otimes \mathbf{V}. \tag{5.9}$$

5.2.2 Cancelamento Cruzado

Um procedimento inverso ao anterior poderia ser usado, dividindo-se o espectrograma $\mathbf{V}$ pelo somatório de todos $\mathbf{\Lambda}_q$, para $q \neq d$. O inconveniente desta técnica é que trechos inativos dos espectrogramas $\mathbf{\Lambda}_q$ resultariam em divisões por valores pequenos, gerando problemas numéricos. Uma abordagem semelhante, porém numericamente mais estável, pode ser escrita como

$$\hat{\mathbf{\Lambda}}_d = \frac{\mathbf{\Lambda}_d}{\sum_{i=1}^{D} \mathbf{\Lambda}_i} \otimes \mathbf{V}. \tag{5.10}$$

Neste caso, a divisão por $\sum_{i=1}^{D} \mathbf{\Lambda}_i$ reduz o efeito das fontes não desejadas, enquanto a multiplicação por $\mathbf{V}$ ressalta a fonte desejada.

5.2.3 Filtragem de Wiener

O problema de estimação do espectrograma de uma fonte, dado o espectrograma de uma mistura, pode ser formulado no contexto da filtragem de Wiener [40]:

$$\hat{\mathbf{\Lambda}}_d = \frac{\mathbf{\Lambda}_d^2}{\sum_{i=1}^{D} \mathbf{\Lambda}_i^2} \otimes \mathbf{V}, \tag{5.11}$$

onde $\mathbf{\Lambda}_d^2$ representa o espectro de potência do sinal desejado, $\sum_{i=1}^{D} \mathbf{\Lambda}_i^2$ representa o espectro de potência da estimativa do sinal com ruído, e a razão $\frac{\mathbf{\Lambda}_d^2}{\sum_{i=1}^{D} \mathbf{\Lambda}_i^2}$ pode ser vista como o filtro de Wiener. O espectrograma $\mathbf{V}$ da mistura é o espectro do sinal com ruído (a fonte d é o sinal e as fontes $\neq d$ são o ruído), e com isso, $\hat{\mathbf{\Lambda}}_d$ pode ser olhado como a estimativa MMSE (*Minimum Mean Squared Error*) do espectrograma da fonte d.

5.2.4 Máscara Binária

Este procedimento assume que as fontes não têm superposição na representação tempo-frequencial, o que para sinais de voz pode ser considerado verdadeiro. A estimativa de $\mathbf{\Lambda}_d$ é calculada como

$$\hat{\mathbf{\Lambda}}_d = \mathbf{M}_d \otimes \mathbf{V}, \tag{5.12}$$

onde $\mathbf{M}_d$ é uma matriz composta por zeros e uns. Um elemento de $\mathbf{M}_d$ é unitário quando a d-ésima fonte tem a maior potência naquela posição; caso contrário, o elemento recebe zero.

□

Em [39], fazemos uma comparação entre os métodos de processamento aqui revisados. Os resultados utilizando a filtragem de Wiener mostraram-se melhores. Desta forma, nos experimentos do Capítulo 8 será sempre utilizada a filtragem de Wiener na etapa de processamento.

Capítulo 6

Síntese

Seguindo o diagrama de blocos da Figura 1.1, encontramos o bloco de *Síntese* após o de *Processamento*. Recebendo como entrada os espectrogramas lineares $\mathbf{\Lambda}_d$ de cada fonte, este bloco tem o objetivo de gerar os sinais das fontes no domínio do tempo.

No contexto de processamento de sinais de áudio, o problema da síntese surge naturalmente ao trabalharmos no domínio da frequência. Quando o processamento envolve modificação de um espectrograma de magnitude, a informação de fase é perdida. Em geral, o espectrograma modificado não é válido, no sentido de que é possível que nenhum sinal real possua tal espectrograma de magnitude [41].

Uma aplicação clássica onde é necessário aproximar um sinal a partir de seu espectrograma de magnitude modificado é o *vocoder*, utilizado no processamento de voz. Quando queremos alongar um sinal no tempo, podemos inserir quadros entre os quadros existentes, utilizando alguma forma de interpolação. O resultado é um novo espectrograma de magnitude, e neste caso também não é possível utilizar a fase do sinal original, pois a base de tempo foi alterada. Um procedimento semelhante é usado na modificação de *pitch*.

A motivação para a estimação da fase de um espectrograma de magnitude nesta dissertação fica clara ao observarmos o diagrama de blocos da Figura 1.1. O bloco de *Processamento* tem como saída espectrogramas de magnitude referentes às fontes separadas pelo algoritmos de fatoração em matrizes não-negativas. Estes espectrogramas não possuem nenhuma informação de fase, de modo que precisamos estimá-la para obter o sinal separado no domínio do tempo.

Uma solução comumente adotada nestes casos é a utilização da fase da mistura

original em conjunto com o espectrograma de magnitude modificado para gerar um sinal no tempo. O resultado é tão pior quanto mais profundas forem as modificações realizadas, e no entanto o procedimento tem sido utilizado em diversos trabalhos como [27], [13], [37], por sua simplicidade de aplicação e pelos resultados razoáveis que apresenta.

Entretanto, dependendo do tipo de sinal, a utilização da fase da mistura pode ser muito prejudicial à síntese. Para sinais com ataques rápidos, por exemplo, a fase correta é decisiva na determinação do instante do ataque.

Neste capítulo, após uma definição de notação, descreveremos algumas soluções utilizadas na literatura.

6.1 Definições

Neste capítulo, trabalharemos com a seguinte definição para o Transformada de Fourier de Tempo Curto (*Short-Time Fourier Transform*, STFT) para um sinal $x(k), k = 1, \ldots, K-1$:

$$X^m(n) = \sum_{k=0}^{N-1} x(k)w(k - mS)e^{-j\frac{2\pi n}{N}k}, \quad \text{para} \quad n = 0, \ldots, N-1, \tag{6.1}$$

onde n é o contador de raias, m é o contador de quadros, S é o avanço em amostras a cada quadro, N é o tamanho da janela (igual ao número de raias), e w é a janela de Hamming, definida como:

$$w(k) = \begin{cases} 2\frac{\sqrt{S/L}}{\sqrt{4a^2+2b^2}}\left(a + b\cos(\frac{2\pi k}{L} + \frac{\pi}{L})\right), \text{ para } 0 \leq n < L \\ 0, \text{ em outros casos,} \end{cases} \tag{6.2}$$

onde L é o tamanho da janela, $a = 0,54$ e $b = -0,46$.

Além disso, definimos a magnitude da transformada de Fourier de tempo curto (*Short-Time Fourier Transform Magnitude*, STFTM) como $|X^m(n)|$. Ao separarmos um espectrograma de magnitude como a soma de espectrogramas gerados pelas fontes, criamos as transformadas de Fourier de tempo curto modificadas (*Modified Short-Time Fourier Transform*, MSTFTMs).

6.2 Algoritmo de Griffin e Lim

Uma das primeiras soluções dadas na literatura para o problema da determinação de fase foi o algoritmo de Griffin e Lim (G&L) [41], cuja idéia é procurar um sinal no tempo cuja STFTM seja a mais próxima possível da MSTFTM desejada, no sentido de mínimos quadrados. Para isso, é utilizado um procedimento iterativo, no qual a estimativa da fase vai sendo aproximada enquanto o espectrograma de magnitude é fixado.

Os passos do algoritmo G&L estão descritos no Algoritmo 6.1. Como entrada, o algoritmo recebe $|Y^m(n)|$, o MSTFTM alvo. Além disso, é necessário fornecer o tamanho da janela de análise L e o salto entre janelas S. Na saída, temos a estimativa $\hat{x}(k)$ do sinal no tempo.

Algoritmo 6.1 Algoritmo G&L

Entrada: A MSTFTM $|Y^m(n)|$, o tamanho da janela de análise L e o salto entre janelas S.

1. Estimação inicial do sinal no tempo $\hat{x}(k)$, que pode ser feita com amostras de uma distribuição uniforme.

2. Geração de uma estimativa de STFT: $\hat{X}^m(n) = |Y^m(n)|e^{-j\angle\hat{x}(k)}$. Este passo é chamado de *Magnitude-Constrained*, pois gera-se uma STFT que possui a magnitude alvo $|Y^m(n)|$ e a fase da estimativa $\hat{x}(k)$.

3. Atualização de $\hat{x}(k)$ segundo a equação:

$$\hat{x}(k) = \frac{\sum_{m=-\infty}^{\infty} w(k - mS)\frac{1}{2\pi}\sum_{n=0}^{N-1}\hat{X}^m(n)e^{j\frac{2\pi n}{N}k}}{\sum_{m=-\infty}^{\infty} w(k - mS)^2} \tag{6.3}$$

   Esta equação busca a minimização do erro quadrático entre o alvo $|Y^m(n)|$ e a STFTM $|\hat{X}^m(n)|$ do sinal $\hat{x}(k)$ que está sendo estimado.

4. Volta ao passo 2 até a convergência: O passo 2 é executado novamente, desta vez com uma estimativa melhor do sinal no tempo e sua fase. A convergência pode ser medida pela diferença entre as estimativas $\hat{x}(k)$ a cada iteração.

Saída: Estimativa do sinal no tempo $\hat{x}(k)$.

6.3 Algoritmos *Real-time Iterative Spectrogram Inversion* (RTISI)

No algoritmo G&L, cada quadro utiliza informações de quadros passados e futuros, o que torna a sua utilização em tempo real inviável por definição. Além disso, o alto número de transformadas de Fourier torna o algoritmo custoso computacionalmente.

O algoritmo RTISI [42] propõe uma solução para ambos os problemas. Cada quadro só depende dos quadros anteriores, e a convergência é acelerada utilizando uma inicialização melhor.

A ideia principal do algoritmo é estimar um quadro por vez, ao contrário do algoritmo G&L, que estima o sinal inteiro. Considerando $L = 4S$, ou seja, uma sobreposição entre janelas de 75%, antes de iniciarmos a estimação do quadro m, ele já possui 75% das amostras preenchidas pelos 3 quadro anteriores, e os 25% finais são nulos. Assim, em vez de começarmos a estimativa da fase do quadro m com zeros, já temos parte das amostras preenchidas, o que permite fazer uma inicialização mais próxima e coerente com a do quadro anterior. Em seguida, aplicamos esta fase à magnitude alvo do quadro e iteramos até a convergência. O método encontra-se sistematizado no Algoritmo 6.2.

Algoritmo 6.2 Algoritmo RTISI

Entrada: A MSTFTM $|Y^m(n)|$, o tamanho da janela de análise L e o salto entre janelas S.

1. Estimativa inicial do sinal no tempo, $\hat{x}(k)$;
2. Para cada quadro m de $\hat{x}(k)$, definição do sinal $\hat{x}^m(k)$;
3. Até que a estimativa de $\hat{x}^m(k)$ convirja:
   (a) DFT de $\hat{x}^m(k)$, $X^m(n)$;
   (b) Geração de um quadro *Magnitude-Constrained*, $\hat{X}_m(n) = |Y^m(n)|\angle X^m(n)$;
   (c) iDFT de $\hat{X}^m(n)$, que resulta em $\hat{x}^m(k)$.
4. Após a convergência do quadro, *overlap-and-add* em $\hat{x}(k)$ e volta ao passo 2, com o quadro $m + 1$.

Saída: Estimativa do sinal no tempo $\hat{x}(k)$.

Este algoritmo possui uma versão avançada, também descrita em [42], denominada RTISI *Look Ahead* (RTISI-LA). Neste método, também são utilizados p quadros à frente na estimação do quadro m. Isso torna o algoritmo mais custoso computacionalmente, além de impor um atraso estrutural de p quadros. Entretanto, a estimativa da fase tem melhora substancial.

No RTISI-LA, cada quadro m tem influência de amostras de quadros anteriores e posteriores, ao contrário do RTISI, em que apenas os quadros anteriores eram utilizados. No caso do RTISI-LA, após a estimação do quadro $m+p$, o quadro m é reestimado, desta vez levando em conta os p quadros posteriores que o influenciam.

6.4 Algoritmo *Multiple Input Spectrogram Inversion* (MISI)

O algoritmo MISI [43] foi criado especificamente para aplicação em separação de sinais. A etapa de estimação é baseada no algoritmo G&L. Entretanto, as fontes provenientes da separação são estimadas conjuntamente, o que permite a utilização do erro de reconstrução

$$e(k) = y(k) - \sum_{d=1}^{D} \hat{x}_d(k) \tag{6.4}$$

no algoritmo. Ao final de cada iteração de todas as D fontes, é adicionado a cada fonte o valor e/D, de modo que a energia total é preservada. O Algoritmo 6.3 detalha o procedimento.

6.5 Outros métodos

Uma outra abordagem para o problema de estimação da fase em separação de sinais de áudio é apresentado em [44]. Neste trabalho, os autores incorporam a fase no algoritmo de NMF ao invés de descartá-la no início. Desta maneira, o resultado da decomposição já inclui a fase e não há necessidade de estimá-la.

Uma abordagem probabilística é apresentada em [45]. Ao invés de alternar a estimação da fase com a estimação de um sinal consistente com o espectrograma-alvo, os autores propõem um método que otimiza ambos conjuntamente, utilizando

a regra MAP (*Maximum a Posteriori* [40]).

Algoritmo 6.3 Algoritmo MISI

Entrada: O sinal da mistura no tempo, $y(k)$, o tamanho da janela de análise L e o salto entre janelas S.

1. Estimação inicial dos sinais no tempo $\hat{x}_d(k)$, para $d = 1, \ldots, D$. Os sinais de todas as fontes são inicializados com a mistura original $y(k)$, e o erro é calculado através da equação (6.4).
2. Para cada fonte d:
   (a) Geração de uma estimativa de STFT: $\hat{X}_d^m(n) = |Y_d^m(n)|e^{-j\angle\hat{x}_d(k)}$;
   (b) Atualização de $\hat{x}_d(k)$, extraindo a iDFT de $\hat{X}_d^m(n)$.
3. Com todas as estimativas $\hat{x}_d(k)$, cálculo do erro utilizando a equação (6.4), e correção:
$$\hat{x}_d(k) = \hat{x}_d(k) + e(k)/D. \tag{6.5}$$
4. Volta ao passo 2 até a convergência.

Saída: Estimativa dos sinais no tempo $\hat{x}_d(k), d = 1, \ldots, D$.

6.6 Testes

Para verificar o funcionamento dos algoritmos, dois testes serão realizados. No primeiro deles, mais simples, duas fontes serão misturadas, e aos algoritmos serão entregues as próprias STFTMs das fontes. Ao algoritmo MISI, será entregue ainda o sinal misturado. O segundo teste tentará submeter os métodos a uma situação mais realista: as STFTMs das fontes serão contaminadas com ruído branco uniforme antes de serem entregues aos algoritmos, simulando assim uma MSTFTM. Neste caso, novamente será entregue ao algoritmo MISI o sinal limpo das duas fontes misturadas, como se as MSTFTMs ruidosas viessem de uma separação não-ideal.

Nos dois casos, serão executadas de 1 a 50 iterações para cada algoritmo para verficar a redução do erro, que é calculado como:

$$E = \sum_{m=1}^{M}\sum_{n=1}^{N}(|Y^m(n)| - |\hat{X}^m(n)|)^2. \tag{6.6}$$

As Figuras 6.1 e 6.2 mostram os resultados de uma das fontes para os casos com e sem ruído, respectivamente. Podemos ver na Figura 6.1 que os algoritmos RTISI e RTISI-LA têm comportamentos muito semelhantes, o que pode estar relacionado ao tipo do sinal usado. Ambos de fato aceleram a convergência em relação ao G&L. O pior desempenho do algoritmo G&L é compreensível, na medida em que os outros algoritmos se basearam nele para seu desenvolvimento.

O algoritmo MISI tem um desempenho excepcional, e o seu erro de aproximação do espectrograma chega quase a zero. Entretando devemos notar que o cenário ideal que está sendo testado favorece bastante este método. A equação (6.4) baseia-se fortemente na condição de que não houve alteração de energia entre os espectrogramas-alvo e o sinal da mistura no tempo, condição que é atendida neste cenário. Caso contrário, o módulo do erro será elevado, e na equação (6.5) grandes porções da mistura serão adicionadas aos sinais separados, prejudicando assim a separação.

No segundo cenário, para o qual os resultados são mostrados na Figura 6.2, o equilíbrio de energias se perde, e o MISI se equipara aos outros algoritmos. Neste teste, vemos novamente que não foi possível determinar diferenças entre o RTISI e RTISI-LA. Ambos têm descréscimo bem rápido, mas o patamar final é mais alto do que o dos outros dois métodos. Entretanto, o resultado final para os quatro métodos é bem parecido.

□

Cabe ressaltar que uma grande dificuldade nesta etapa do trabalho foi a implementação dos métodos acima citados. Nenhum dos autores ofereceu código-fonte, e nos artigos não existe descrição sistemática dos métodos. Portanto, não é possível garantir que as implementações realizadas neste trabalho estejam idênticas às dos autores. Um exemplo e indicativo disto é que a razão L/S (tamanho da janela/salto entre janelas) altera radicalmente os resultados. Apesar de ser sugerido o valor 4, não há nenhum motivo para que os métodos não funcionem com valores diferentes, já que as janelas são adaptadas a esta razão.

No Capítulo 8, os métodos de síntese serão testados dentro do sistema de separação de sinais. Nos testes, veremos o desempenho dos métodos em comparação com o resultado da síntese utilizando a fase da mistura original.

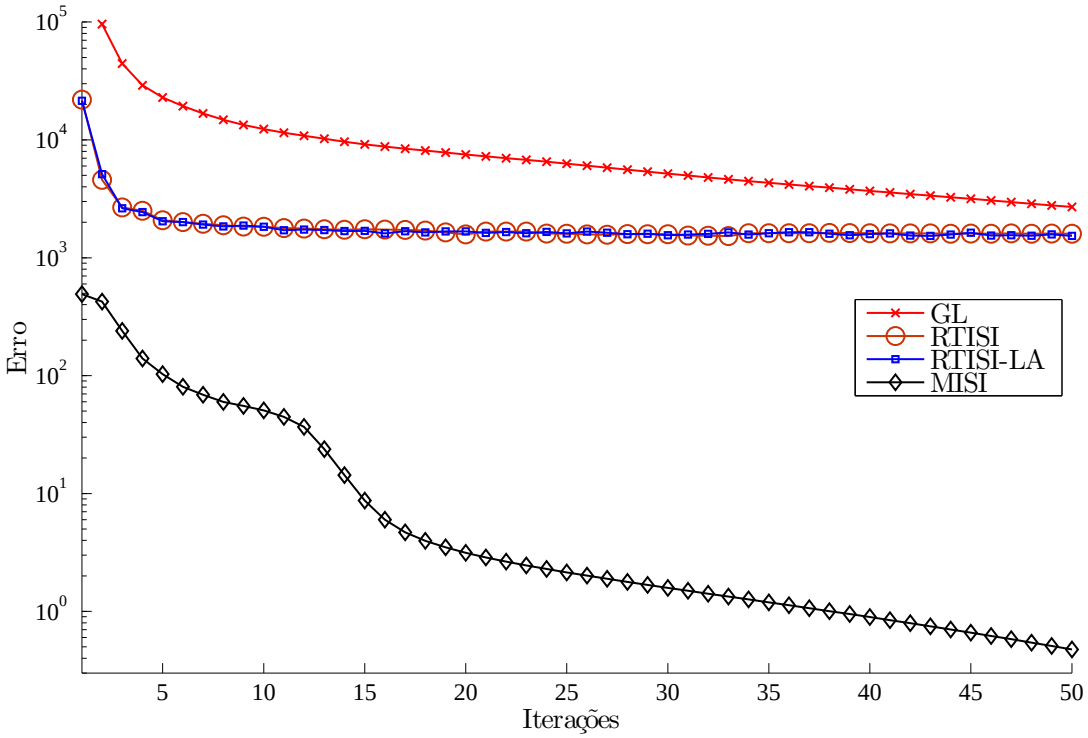


Figura 6.1: Erro quadrático entre o espectrograma-alvo e o espectrograma do sinal estimado pelos métodos de síntese. Neste exemplo, o espectrograma-alvo é uma STFTM, ou seja, foi gerado por um sinal real.

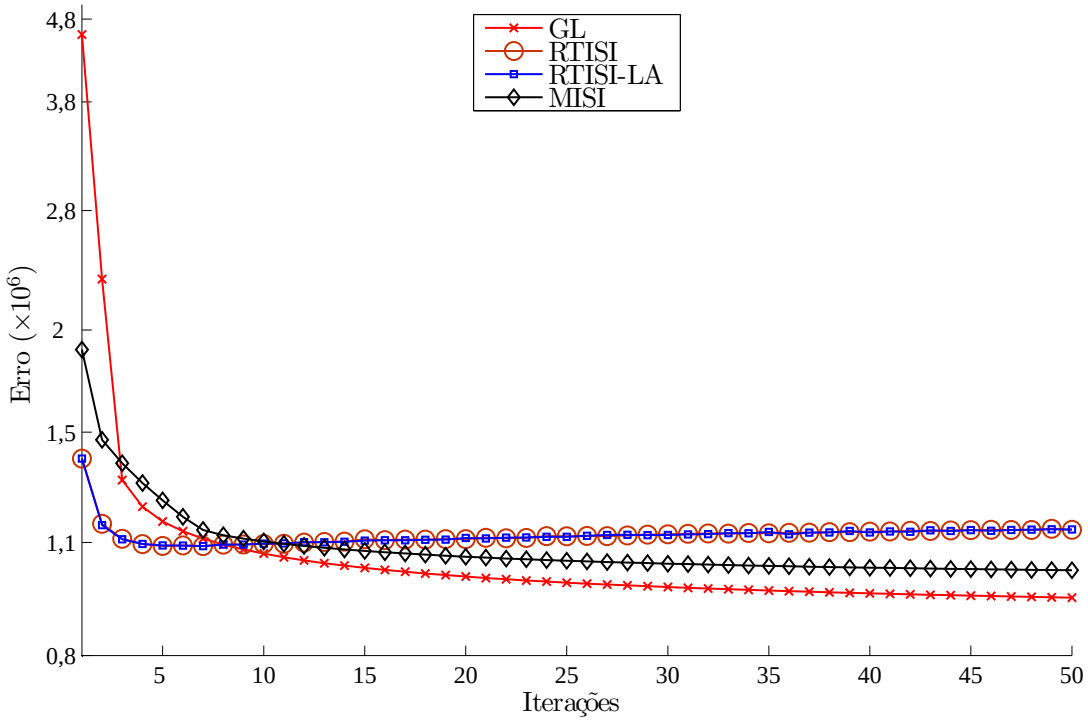


Figura 6.2: Erro quadrático entre o espectrograma alvo e o espectrograma do sinal estimado pelos métodos de síntese. Neste exemplo, o espectrograma-alvo foi gerado por um sinal real e depois modificado, tornando-se uma MSTFTM. Esta situação será encontrada no sistema de separação de sinais.

Capítulo 7

Avaliação de Qualidade

O último bloco do diagrama da Figura 1.1 mostra a etapa de *Avaliação de Qualidade*. Terminada a tarefa principal de separar as fontes sonoras, convém dispor de algum mecanismo que permita aferir a qualidade dos resultados.

7.1 Introdução

A avaliação de qualidade é necessária quando se deseja comparar resultados de algoritmos. No caso de sinais musicais, seria possível solicitar que pessoas avaliassem os resultados e que, por exemplo, dessem uma nota para auxílio na decisão sobre o melhor algoritmo. No entanto, um teste subjetivo, como este procedimento é chamado, necessita de muitas pessoas e condições rigidamente controladas, de modo que só deve ser realizado em situações que o demandem estritamente. Em [46] é possível conhecer toda a complexidade que envolve este tipo de teste.

Para os testes de algoritmos no dia-a-dia, usamos a chamada avaliação objetiva de qualidade. São algoritmos que tentam de alguma maneira simular a percepção humana de modo a possibilitar a comparação entre resultados.

Existem basicamente dois tipos de abordagem quando se trata de avaliação objetiva de qualidade: os métodos inspirados em razão sinal-ruído (*signal-to-noise ratio*, SNR) e aqueles que se baseiam em modelos psicoacústicos. Os primeiros são mais ingênuos, pois consideram a potência do sinal e do ruído sem atentar para as especificidades do aparelho auditivo humano. São úteis apenas como uma estimativa grosseira. Já os métodos psicoacústicos atingem resultados bem melhores, pois só

consideram na avaliação o que realmente se escuta.

Os algoritmos de avaliação de qualidade aplicados especificamente à separação de sinais ainda são pouco desenvolvidos. Apesar de desde 2007 existir uma competição de separação de sinais[1], em que os resultados são avaliados objetivamente, as alternativas para avaliação ainda são restritas.

Em outras áreas, como por exemplo a avaliação de codificadores, os algoritmos de avaliação automática conseguem taxas de correlação com testes subjetivos acima dos 90%. Exemplos dessa categoria de algoritmos são o PESQ (*Perceptual Evaluation of Speech Quality*, [47]), voltado para fala, e o PEAQ (*Perceptual Evaluation of Audio Quality*, [48]), voltado para áudio. Ambos são avaliadores objetivos com referência, isto é, avaliam um sinal codificado em relação à sua versão original e dão uma nota que representa a degradação percebida. Para tal, utilizam modelos psicoacústicos que incluem curvas de mascaramento e de audibilidade, e as mínimas diferenças percebidas, conhecidas como JND (*Just Noticeable Difference*).

Os trabalhos [49], [50], [51] e [52] tratam especificamente de avaliação objetiva de qualidade aplicada a separação de sinais, sobre o que será feita uma breve revisão a seguir.

Além da qualidade do sinal separado, existem outros critérios de avaliação de qualidade da separação, que estão fora do escopo deste trabalho. Pode-se avaliar, por exemplo, a detecção correta do número de fontes, da matriz de misturas ou da localização espacial das fontes. Nesta dissertação, consideramos apenas a qualidade do sinal separado em relação ao original.

A próxima seção descreve o método apresentado em [49], que, apesar de baseado em critérios de SNR, ainda é bastante usado. Em seguida, apresentamos um método da literatura [50] [51] que utiliza o PESQ para avaliação da separação de voz, e por fim propomos a utilização do PEAQ como medida de avaliação de separação de sinais musicais. Na Seção 7.4, serão apresentados testes envolvendo algoritmos de fatoração e seus resultados serão analisados a partir das medidas de avaliação de qualidade.

[1] `http://www.irisa.fr/metiss/SASSEC07/`

7.2 Medidas baseadas em SNR

Quando falamos em avaliação de qualidade da separação, diversos critérios podem ser considerados. Por exemplo, para saber se o resultado final tem boa qualidade sonora sob vários aspectos, é possível modelar [49] o sinal de áudio separado como

$$s_d = s_{\text{alvo}} + e_{\text{inter}} + e_{\text{artef}} + e_{\text{ruído}}, \tag{7.1}$$

onde s_d é a fonte separada, s_{alvo} é a fonte original, e_{inter} é a interferência causada por outras fontes e e_{artef} são os defeitos possivelmente inseridos pelo processo de separação. O termo $e_{\text{ruído}}$ é utilizado caso haja presença de ruído na mistura.

Para que esta separação seja possível, o modelo necessita da estimativa da fonte separada (o que se quer avaliar), de todas as fontes originalmente separadas e do ruído separado.

A partir deste modelo, podemos definir quatro medidas de qualidade:

***Source-to-Distortion Ratio* (SDR):** A razão fonte-distorção nos dá uma ideia de qualidade geral da separação, e será usada no Capítulo 8 como uma das principais medidas de avaliação de qualidade. É calculada como

$$\text{SDR} = 10\log_{10}\frac{||s_{\text{alvo}}||^2}{||e_{\text{inter}} + e_{\text{artef}} + e_{\text{ruído}}||^2}. \tag{7.2}$$

***Source-to-Interferences Ratio* (SIR):** A razão fonte-interferência nos informa a potência de sinal das outras fontes que foi inserida na fonte de interesse. É uma medida de qualidade da separação em si, e é calculada como

$$\text{SIR} = 10\log_{10}\frac{||s_{\text{alvo}}||^2}{||e_{\text{inter}}||^2}. \tag{7.3}$$

***Sources-to-Artifacts Ratio* (SAR):** A razão fontes-defeitos[2] nos dá uma ideia da quantidade de defeitos que foram inseridos no processo de separação, ou seja, a inserção de elementos que não estavam presentes na mistura original. A etapa de

[2]No nome desta métrica e da próxima, *fontes* está no plural porque comparamos s_{alvo} e e_{inter}, o que inclui outras fontes além da desejada, com os defeitos inseridos.

síntese é uma das maiores causadoras de defeitos. A SAR é calculada como

$$\mathrm{SAR} = 10\log_{10}\frac{||s_{\mathrm{alvo}} + e_{\mathrm{inter}} + e_{\mathrm{ruído}}||^2}{||e_{\mathrm{artef}}||^2}. \tag{7.4}$$

***Sources-to-Noise Ratio* (SNR):** Nos casos em que a mistura contém ruído, a razão fontes-ruído fornece uma medida da quantidade de ruído que restou junto às fontes.

$$\mathrm{SNR} = 10\log_{10}\frac{||s_{\mathrm{alvo}} + e_{\mathrm{inter}}||^2}{||e_{\mathrm{ruído}}||^2}. \tag{7.5}$$

As medidas são invariantes ao ganho e à ordenação dos sinais. Isto significa que variações de ganho não são penalizadas, e que cada estimativa de fonte separada é comparada com todas as fontes originais, e aquela que possuir maior SDR é considerada a fonte correta. Todas as medidas podem ser calculadas utilizando-se o pacote disponível em `http://bass-db.gforge.inria.fr/bss_eval/`.

7.3 Medidas baseadas em psicoacústica

A utilização de medidas de qualidade objetivas fundamentadas na psicoacústica envolve uma transformação do sinal de áudio, que passa do domínio do tempo para o domínio psicoacústico. Nesta transformação, procura-se emular a percepção sonora humana através de filtros que simulam o funcionamento do sistema auditivo e cognitivo. Após ser modelado psicoacusticamente, o sinal é então processado e diversos tipos de medidas podem ser realizadas.

7.3.1 Avaliação utilizando PESQ

O avaliador PESQ (*Perceptual Evaluation of Speech Quality*) foi projetado para sinais de voz restritos à banda telefônica (300 a 3400 Hz). É um avaliador com referência, ou seja, compara o sinal de teste com um sinal de referência, medindo as diferenças no domínio psicoacústico. Seu funcionamento pode ser dividido em 3 fases: pré-processamento, modelo perceptual e modelo cognitivo.

O pré-processamento inclui alinhamento temporal e de potência entre o sinal de referência e o de teste. O modelo perceptual simula o sistema auditivo humano, e o

modelo cognitivo processa as diferenças e calcula a nota final.

Recentemente, [50] [51] descreveram um procedimento de avaliação de qualidade de separação de vozes utilizando o PESQ. Após passar por um sistema de separação, sinais contendo misturas de vozes foram avaliados pelo PESQ utilizando as vozes originalmente separadas como referência.

Os resultados do algoritmo PESQ foram então comparados com a taxa de reconhecimento de palavras de um sistema de reconhecimento automático de voz, e correlações de mais de 90% foram encontradas, inclusive na presença de ruído.

Vale lembrar que a não realização de testes subjetivos com seres humanos deixa aberta uma área de pesquisa. Os resultados bem correlacionados com o reconhecedor de voz indicam que a solução é viável, mas não significam necessariamente alta correlação com a opinião de indivíduos.

7.3.2 Avaliação utilizando PEAQ

O PEAQ (*Perceptual Evaluation of Audio Quality*) é um avaliador de qualidade objetivo, com referência e projetado para sinais de áudio e voz com largura de banda de 24 kHz (taxa de amostragem de 48 kHz). Embora tenha sido especificado para avaliar degradações provocadas por codificadores, diversos trabalhos estenderam seu funcionamento para outros tipos de degradações [53].

Este avaliador tem seu funcionamento dividido em duas etapas. Na primeira, são calculadas 11 medidas, chamadas de MOVs (*Model Output Variables*). As MOVs são calculadas a partir de diferenças entre o modelo psicoacústico do sinal de referência e do sinal de teste. Na segunda etapa, os valores de saída das MOVs são combinados em uma rede neural, treinada utilizando medidas subjetivas atribuídas a sinais de áudio e voz degradados por codificadores. A saída desta rede fornece uma nota final para o sinal, que é chamada de ODG (*Objective Difference Grade*).

Assim como o PESQ foi utilizado para avaliação de separação de voz, investigaremos o uso do PEAQ para avaliação de separação de áudio. Para isso, será necessária uma adequação da taxa de amostragem, já que o avaliador trabalha apenas com sinais amostrados a 48 kHz, e os sinais utilizados nos testes do Capítulo 8 têm taxas entre 8 e 44,1 kHz. Esta adequação será feita utilizando a função `resample` do Matlab©.

Será utilizada a implementação do PEAQ disponível em `http://www-mmsp.ece.mcgill.ca/Documents/Downloads/PQevalAudio/`. O *software* recebe como entrada o nome dos sinais de referência e de teste, e entrega na saída a ODG. A nota é dada em relação ao grau de incômodo provocado pela degradação que sinal de teste sofreu, e varia de -4 (degradação muito incômoda) a 0 (degradação imperceptível).

7.4 Testes

Nesta seção iremos utilizar sinais fatorados pelo sistema apresentado nesta dissertação para verificar o funcionamento das medidas de avaliação de qualidade mencionadas acima.

O primeiro teste irá verificar a melhora na qualidade de síntese do sinal à medida em que aumentamos o número de iterações do algoritmo de fatoração. O sinal é uma mistura composta por 2 instrumentos, e será decomposto utilizando o algoritmo LNMF2D. A expectativa é de que, ao aumentar o número de iterações, a qualidade do sinal sintetizado também melhore. Para a síntese, utilizaremos o método mais simples, aproveitando a fase do sinal original sem processamento. A Figura 7.1 mostra o valor do erro de reconstrução total, e a SDR e o ODG em função do número de iterações, para cada um dos instrumentos.

A Figura 7.1 nos mostra que os resultados para ambas as medidas são coerentes, ou seja, quanto menor o erro de reconstrução, maior a qualidade do sinal sintetizado. Nota-se, no entanto, que os resultados numéricos do PEAQ são bem rigorosos, já que, para resultados auditivamente bons, as notas ficaram todas próximas do limite inferior (-4). Isto indica que a ODG pode ser usada como medida de comparação de qualidade entre configurações do sistema, mediante algum tipo de remapeamento.

A fonte 1 teve melhor resultado para SDR e a fonte 2 apresentou melhor desempenho em relação à ODG. Isto pode ser um indicativo de que o método baseado em SNR avalia defeitos que não são necessariamente percebidos. No entanto, a ausência de testes subjetivos rigorosos não nos permite ir mais além nesta análise.

No segundo teste, iremos verificar a diferença na qualidade final do sinal utilizando os métodos de filtragem espectral apresentados no Capítulo 5. Após passar

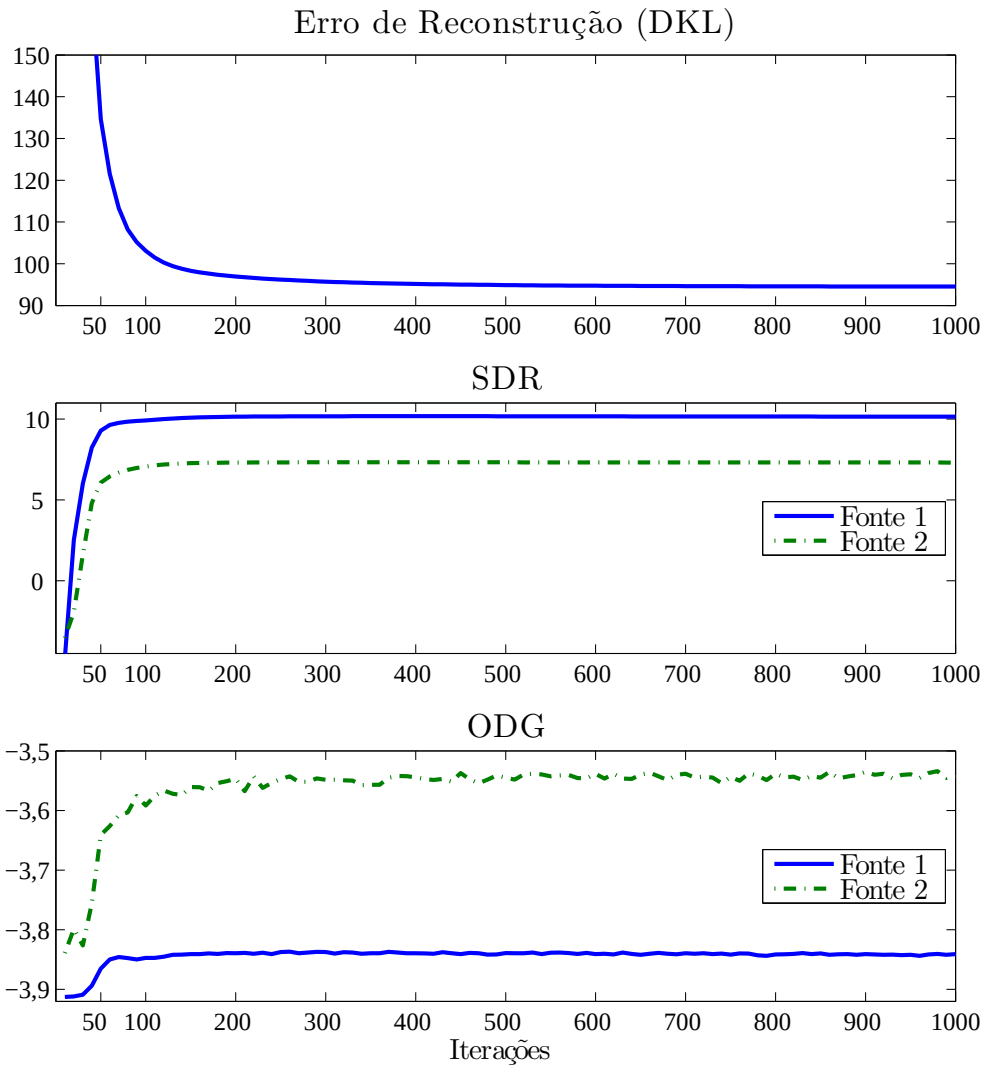


Figura 7.1: Erro, SDR e ODG em função do número de iterações.

pelo bloco de *Fatoração*, os tensores otimizados **W** e **H** são salvos. Cada par é entregue ao bloco seguinte diversas vezes, uma para cada método diferente de filtragem espectral do bloco de *Processamento*.

Na Tabela 7.1 vemos a SDR e a ODG para cada uma das fontes em cada método. O erro de reconstrução foi o mesmo, pois o experimento partiu da mesma fatoração, a fim de se isolar o efeito do processamento. Notamos que, como visto em [39], o processamento utilizando o filtro de Wiener é a melhor solução.

Mais uma vez, o PEAQ mostrou-se coerente na escolha do melhor método. Entretanto, podemos notar divergências, como por exemplo no resultado do Mascaramento Espectral (ME). Para a fonte 1, segundo a métrica SDR, o ME é o pior método, enquanto que para o PEAQ é o segundo melhor, atrás apenas da filtragem

Tabela 7.1: SDR e ODG em função do método de processamento. Para cada método, a primeira linha se refere ao instrumento 1, e a segunda, ao instrumento 2.

Método de processamento	SDR (dB)	ODG
Fase da Mistura	7,30	-3,84
	10,15	-3,56
Filtragem de Wiener	**11,59**	**-3,52**
	12,94	**-3,49**
Cancelamento Cruzado	10,41	-3,79
	11,85	-3,59
Máscara Binária	10,53	-3,63
	11,89	-3,59
Mascaramento Espectral	5,71	-3,57
	6,34	-3,65

de Wiener. Já para a fonte 2, ambas as medidas concordam em que o ME é pior método. Vale observar novamente a faixa de variação reduzida dos valores indicados pelo PEAQ, que indica a necessidade de um novo mapeamento.

□

Os testes apresentados nesta seção apontam que é possível vir a usar o PEAQ como avaliador de qualidade para separação de sinais de áudio. Entretanto, para calcular sua eficiência de maneira conclusiva, seria necessário verificar sua correlação com testes subjetivos. Como este procedimento é muito custoso, e não foi encontrada nenhuma base de sinais separados para a qual tenham sido feitos testes subjetivos, deixamos esta tarefa para trabalhos futuros.

Capítulo 8

Experimentos

O objetivo deste capítulo é verificar o funcionamento do sistema apresentado ao longo desta dissertação. Como existem diversas opções de algoritmos para cada bloco da Figura 1.1, não é viável testar todas as possibilidades de combinações. Portanto, escolheremos algumas com base em testes preliminares e nos resultados publicados em [39]. Na Tabela 8.1 mostramos os algoritmos que foram apresentados para cada bloco do sistema:

Tabela 8.1: Resumo dos algoritmos apresentados nesta dissertação.

Bloco	**Algoritmos**
Análise Tempo-Frequência	STFT, CQT e STFT com mapeamento logarítmico;
Fatoração	NMF, NMFD, NMF2D, LNMF2D, todos utilizando DE ou DKL como medida de reconstrução, e com a possibilidade de utilizar Adaptação Espectral;
Processamento	Reconstrução: pela pseudo-inversa, pela transposta e iterativa; Filtragem do Espectrograma: Mascaramento, Cancelamento Cruzado, Filtragem de Wiener e Mascaramento Binário;
Síntese	iSTFT, G&L, RTISI, RTISI-LA, MISI;
Avalição de Qualidade	SDR, SIR, SAR, SNR e ODG;
Obs: Todo o sistema pode ser utilizado no modo *offline*, ou seja, processando o sinal de uma só vez, ou no modo *online*, em blocos.	

Apresentaremos neste capítulo quatro testes, sendo 3 escolhidos de modo a verificar o funcionamento dos algoritmos propostos no Capítulo 4 e um destinado a observar o comportamento dos algoritmos de síntese apresentados no Capítulo 6. Na Seção 8.2, o algoritmo LNMF2D (Seção 4.1) será comparado com o NMF2D [13]. A comparação se dará utilizando as medidas de avaliação de qualidade SDR [49]

e ODG (Seção 7.3.2), apresentadas no capítulo anterior. Os sinais utilizados nos testes estão detalhados na Seção 8.1.

Na Seção 8.3, verficaremos a melhora no desempenho do sistema ao utilizarmos a adaptação espectral (Seção 4.2). Neste caso, observaremos se o decréscimo no erro de reconstrução, que é a que o algoritmo se propõe, se reflete também numa melhora nas medidas de avaliação de qualidade. Novamente, utilizaremos a SDR e a ODG para comparação.

A Seção 8.4 se destina a verificar a eficácia dos algoritmos de estimação de fase de um espectrograma de magnitude apresentados no Capítulo 6. Iremos avaliar se é mais vantajoso no momento da síntese usar a fase da mistura ou algum dos métodos.

Finalmente, na Seção 8.5, será demonstrado o uso do algoritmo *online* (Seção 4.3). Na comparação com o algoritmo *offline*, iremos verificar a qualidade dos sinais separados e o uso de memória RAM durante o processamento, que esperamos ser menor no caso *online*.

8.1 Banco de Sinais

Para realizar os testes dos algoritmos, será necessário utilizar um banco de sinais. O banco foi montado de forma a conter sinais com diferentes tipos de instrumentos (sopro, percussão, cordas), e variando o número deles entre 1 a 3. No caso de sinais com apenas um instrumento, deseja-se avaliar não a separação, mas a adequação aos modelos propostos. Além disso, há instrumentos em que se pode perceber notas (*pitched*), e outros em que isto não é possível (*unpitched*).

A Tabela 8.2 contém uma descrição de cada sinal, composta pelo nome, número de instrumentos (D) e sua natureza (*pitched* - P ou *unpitched* - U), os parâmetros ϕ e τ utilizados, a taxa de amostragem e a duração em segundos. Cabe ressaltar que ϕ é escolhido de acordo com a extensão em notas do sinal. Por exemplo, se a nota mais grave de um sinal é o dó, e a mais aguda um mi na mesma oitava, temos 5 semitons de extensão, e como $b = 24$, temos $\phi \geq 10$. O uso do menor ϕ possível reduz o tamanho das matrizes. Já para o parâmetro τ, que indica o número de quadros da matriz da base espectral, a escolha é feita de forma empírica, baseada em testes preliminares.

Tabela 8.2: Descrição do banco de sinais usado nos experimentos.

	Nome	D	**P/U**	ϕ	τ	**Fs (Hz)**	**Duração (s)**
1	`flute_perc_org.wav`	3	P/U	10	1	8000	4,00
2	`flute_perc.wav`	2	P/U	10	3	8000	3,26
3	`piano_trumpet_t.wav`	2	P	10	2	8000	4,00
4	`piano_trumpet2.wav`	2	P	10	3	8000	3,26
5	`piano_flute.wav`	2	P	16	3	8000	4,00
6	`piano_sinal_1.wav`	1	P	10	5	44100	15,07
7	`trumpet_sinal_1.wav`	1	P	10	5	44100	18,65
8	`piano_sinal_2.wav`	1	P	20	5	44100	29,35
9	`trumpet_sinal_2.wav`	1	P	20	5	44100	37,57
10	`piano_sinal_3.wav`	1	P	10	5	44100	9,12
11	`trumpet_sinal_3.wav`	1	P	10	5	44100	11,72
12	`piano_sinal_4.wav`	1	P	20	5	44100	15,20
13	`trumpet_sinal_4.wav`	1	P	20	5	44100	19,54
14	`long_tpt.wav`	2	P	10	2	8000	320,00

Os sinais de 1 a 5 e 14 são compostos por misturas de instrumentos sintéticos, gerados por computador. Os instrumentos utilizados foram flauta, percussão e orgão (sinal 1), flauta e percussão (sinal 2), piano e trompete (sinais 3 e 4) e piano e flauta (sinal 5). O primeiro sinal possui 3 instrumentos, o que representa um desafio maior para o sistema. Os cinco sinais têm curta duração (≤ 4 s), e ocupam uma faixa de 5 semitons ($\phi \geq 10$). A escolha de $\phi = 10$ mostrou bons resultados para todos os sinais exceto o sinal 5, que apresentou melhor fatoração com $\phi = 16$.

De 6 a 13, os sinais são compostos por notas de instrumentos reais, retiradas da base da dados RWC [54]. Os sinais 6 e 7 possuem 5 notas consecutivas separadas por um semitom cada, emitidas por um piano e um trompete, respectivamente. Os sinais 8 e 9 seguem a mesma estrutura, mas agora com 10 notas consecutivas. Os sinais 10, 11, 12 e 13 têm a mesma composição dos sinais de 6 a 9, mas as notas são dispostas de forma aleatória, com sobreposição. Desta forma, pretende-se testar a habilidade dos métodos em modelar notas concorrentes de um mesmo instrumento.

O sinal 14 foi construído para testar o algoritmo *online*. É formado por várias repetições do Sinal 3 concatenadas, e possui a duração de 5min20s.

8.2 LNMF2D

Neste experimento, iremos analisar o desempenho do algoritmo NMF2D [13] em relação à sua versão que utiliza espectrograma espaçado linearmente, o LNMF2D, proposto na Seção 4.1 e em [39]. Para o NMF2D, será utilizada a análise tempo-frequência através da STFT mapeada, e para a LNMF2D usaremos a STFT. Ambos os algoritmos utilizarão a DKL como medida de reconstrução, por ter mostrado melhores resultados em testes preliminares. No bloco de processamento, iremos utilizar a Filtragem de Wiener, e para a síntese usaremos a fase da mistura original. Neste experimento não será usada a adaptação espectral.

O problema da ordenação das fontes será resolvido da seguinte forma: após o término do algoritmo de separação, cada fonte original é comparada com cada uma das fontes separadas, e destas a que apresentar melhor medida de qualidade será associada à fonte original. Desta maneira forma-se o vetor de correspondências entre os sinais fontes e separados. Este vetor de tamanho D indica, na posição d_0, o índice do sinal separado correspondente à fonte d_0. Por exemplo, o vetor $\{2, 1, 3\}$ mostra que a fonte 1 corresponde ao sinal separado 2, a fonte 2 ao sinal separado 1, e a fonte 3 ao sinal separado 3.

A convergência dos algoritmos ainda é um problema em aberto. A inicialização dos tensores $\mathbf{W}$ e $\mathbf{H}$ é feita de forma aleatória (distribuição uniforme entre 0 e 1), e foi constatado nos testes preliminares que nem sempre o algoritmo converge para a solução desejada. Quando a convergência não ocorre, o erro passa a crescer a partir de uma determinada iteração até ultrapassar o limite de representação do computador. Assim, decidiu-se conduzir os experimentos da seguinte maneira:

- Cada configuração (sinal, algoritmo) é executada 50 vezes, com um máximo de 1000 iterações;
- Para considerar que um algoritmo convergiu para o resultado esperado, ele deve atender a todos os seguintes critérios:
  - Os vetores de correspondência devem ser iguais para a SDR e ODG, ou seja, as fontes devem ser associadas aos mesmos sinais separados em ambas as medidas de qualidade;

  - Os vetores de correspondência devem ser consistentes, ou seja, cada sinal separado deve corresponder a uma e apenas uma fonte;
  - Todos os valores de SDR devem ser positivos.

- Caso uma rodada não convirja, ela é executada novamente, de modo que as 50 execuções sejam convergentes.

As Figuras 8.2 e 8.3 mostram os resultados para ambas as medidas de qualidade, a primeira para misturas de sinais sintéticos, e a segunda para várias notas de instrumentos reais. Nos gráficos, o eixo das abscissas indica as fontes, e o das ordenadas mostra a medida de qualidade. As linhas cheias separam os sinais, e as pontilhadas, os métodos. Dentro de cada método (**L**NMF2D e **N**MF2D), cada coluna representa uma fonte, e a sua ordenação é mantida para um mesmo sinal. Os pontos marcados com o símbolo $\circ$ indicam a média das iterações que convergiram, com uma barra indicando o desvio-padrão, e os pontos marcados com o símbolo $\bullet$ indicam o valor na melhor iteração. Dentro dos parênteses mostramos o número de rodadas que não convergiram de acordo com os critérios acima, até que se atingissem 50 rodadas corretas.

Assim, para comparação de desempenho entre métodos, deve-se comparar dentro de uma coluna limitada por linhas cheias, a primeira fonte de um método com a primeira do segundo método (após a linha pontilhada), e assim por diante.

Na Figura 8.2, vemos que o Sinal 1 teve mais dificuldades para convergir na NMF2D, e atingiu resultados um pouco piores. A alta quantidade de rodadas que não convergiram nos dois métodos se deve à quantidade de instrumentos. Olhando para os melhores resultados para a medida SDR, a LNMF2D sempre atinge resultados melhores ou iguais. Já considerando-se a média da SDR, a NMF2D tem desempenho melhor apenas no Sinal 4. Devemos notar, entretanto, que analisando o desvio-padrão, todos os resultados são compatíveis.

Para a medida ODG, os resultados têm ligeiras diferenças, mas novamente em relação à média e ao melhor resultado, a LNMF2D é igual ou superior à NMF2D, a não ser na primeira fonte do Sinal 4, onde a NMF2D é superior. Neste caso, o primeiro instrumento dos sinais 1 e 2 têm melhor resultado mesmo considerando o desvio-padrão.

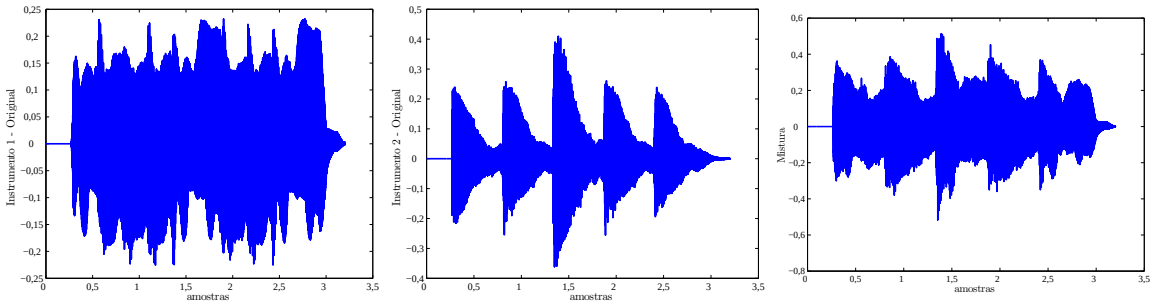
Podemos perceber que o número de rodadas que não converge é altamente dependente do sinal. O Sinal 1 teve mais dificuldades para convergir por causa do número de instrumentos. Entretanto, os Sinais 2 e 3 possuem o mesmo número de instrumentos e o Sinal 2 falhou em 26 vezes para a NMF2D, enquanto o Sinal 3 não falhou para o mesmo método.

As características espectro-temporais que levam um sinal a uma melhor ou pior separabilidade ainda devem ser estudadas mais profundamente. Nos testes acima, vemos que o Sinal 4 apresenta alguma peculiaridade, já que é o único que apresenta melhores resultados para a NMF2D. É provável que a esparsidade tempo-frequencial tenha alguma influência nos resultados, e no número de vezes em que o algoritmo não converge.

Na Figura 8.3, os resultados são mostrados para os sinais com apenas um instrumento. Neste caso, na etapa de processamento não houve filtragem do espectrograma, já que com apenas um instrumento os resultados seriam falsamente inflados, pois o sinal alvo seria o próprio sinal separado. Isto explica as notas mais baixas do que na figura anterior. Novamente, a LNMF2D foi superior à NMF2D, desta vez em todos os sinais, nas médias e nos valores máximos. Nestes exemplos, os algoritmos sempre convergem.

Como forma de ilustrar o funcionamento de cada bloco do sistema, iremos apresentar todas as etapas do processamento para o sinal `piano_trumpet_t`, seguindo os blocos da Figura 1.1. A seguir, todos os passos do processamento:

Sinais no domínio do tempo: A Figura 8.1 mostra as fontes originais no domínio do tempo, e a mistura também no domínio do tempo.



(a) (b) (c)

Figura 8.1: Fontes originais e mistura no domínio do tempo para o sinal`piano_trumpet_t.wav`: (a) Instrumento 1 original; (b) Instrumento 2 original; e (c) Mistura

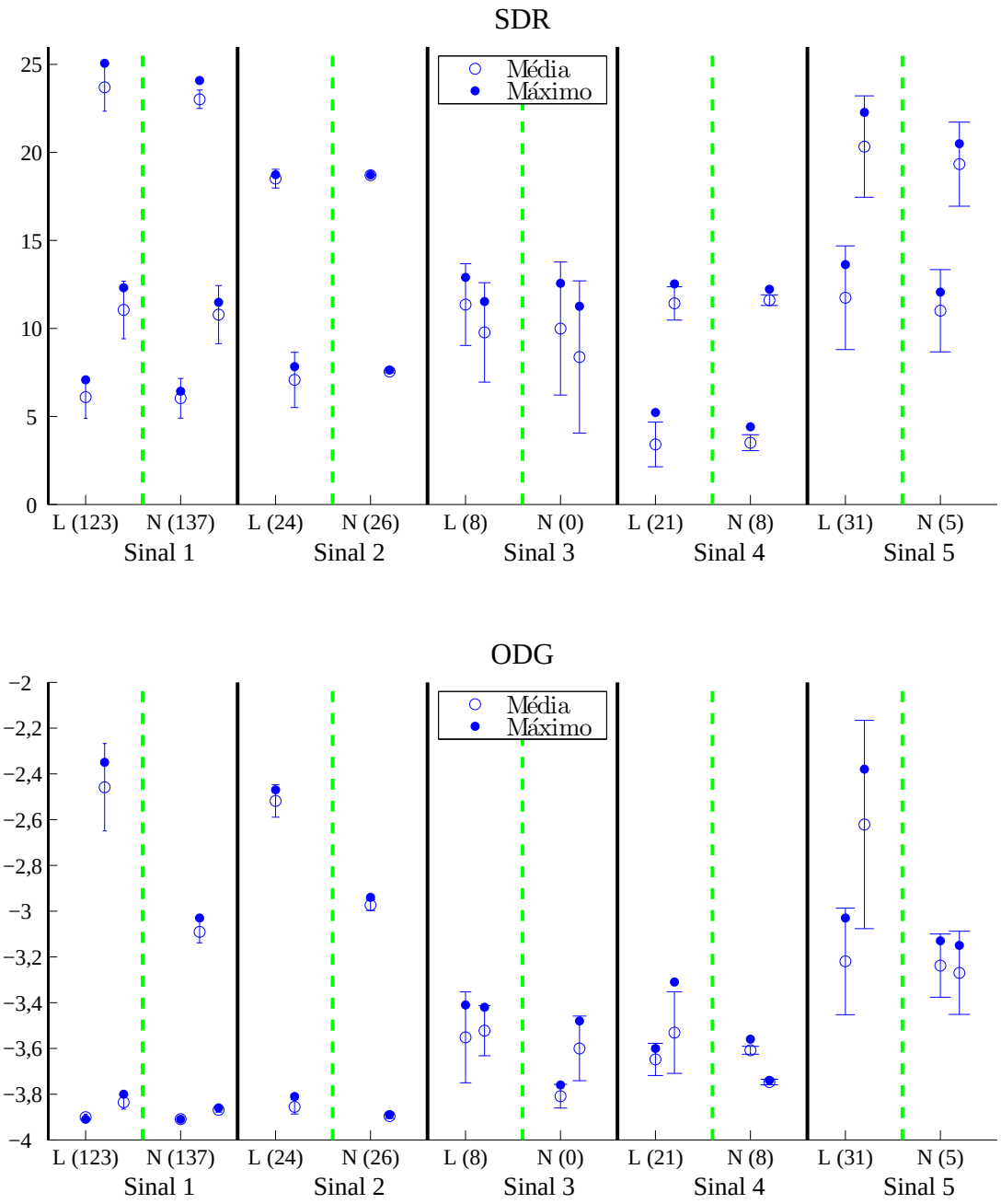


Figura 8.2: Comparação entre os algoritmos LNMF2D e NMF2D. Os gráficos mostram as medidas de qualidade (SDR, acima, e ODG, abaixo) para cada instrumento, indicando a média (○), o máximo (●) e o desvio-padrão entre 50 rodadas. As linhas cheias dividem os sinais, que são detalhados na Tabela 8.2, e as linhas pontilhadas separam os métodos. Dentro de cada método, as colunas representam os instrumentos, sempre na mesma ordem para um mesmo sinal. No eixo das abscissas, L denota LNMF2D e N, NMF2D, e os números entre parênteses indicam quantas rodadas não convergiram até se chegasse a 50 rodadas convergentes.

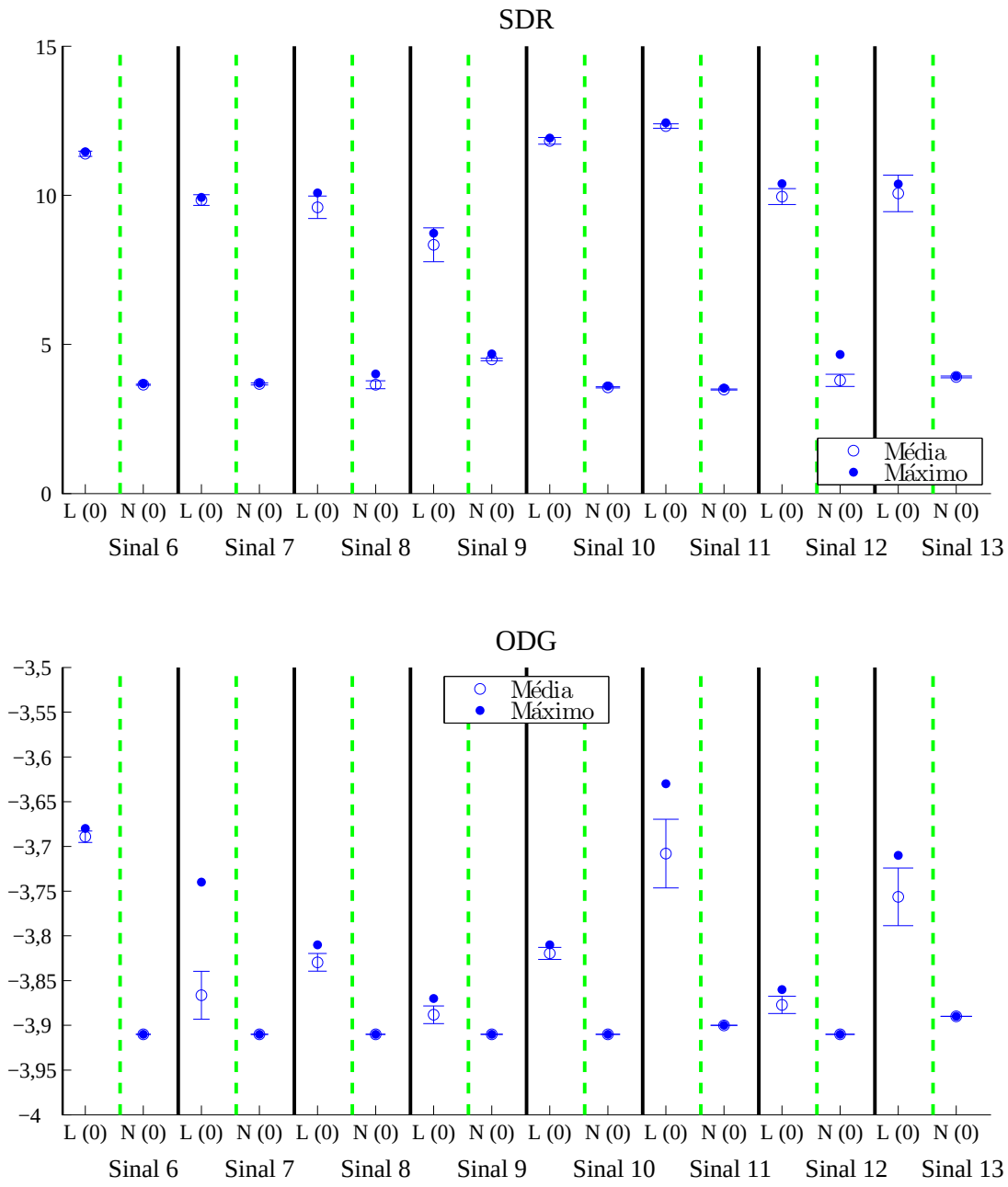


Figura 8.3: Comparação entre os algoritmos LNMF2D e NMF2D. Os gráficos mostram as medidas de qualidade (SDR, acima, e ODG, abaixo) para cada instrumento, indicando a média (○), o máximo (●) e o desvio-padrão entre 50 rodadas. As linhas cheias dividem os sinais, que são detalhados na Tabela 8.2, e as linhas pontilhadas separam os métodos. Dentro de cada método, as colunas representam os instrumentos, sempre na mesma ordem para um mesmo sinal. No eixo das abscissas, L denota LNMF2D e N, NMF2D, e os números entre parênteses indicam quantas rodadas não convergiram até se chegasse a 50 rodadas convergentes. Neste experimento, não foi utilizada a filtragem de Wiener no processamento, já que no caso de apenas um instrumento este procedimento não se justifica.

Análise tempo-frequência: O resultado da análise tempo-frequência é mostrado na Figura 8.4. No caso da STFT a análise foi feita utilizando $W = 512, M = 512$ e $S = 256$. Para a STFT mapeada, foram utilizados $F_{\min} = 220$ Hz, $F_{\max} = 4000$ Hz e $b = 24$.

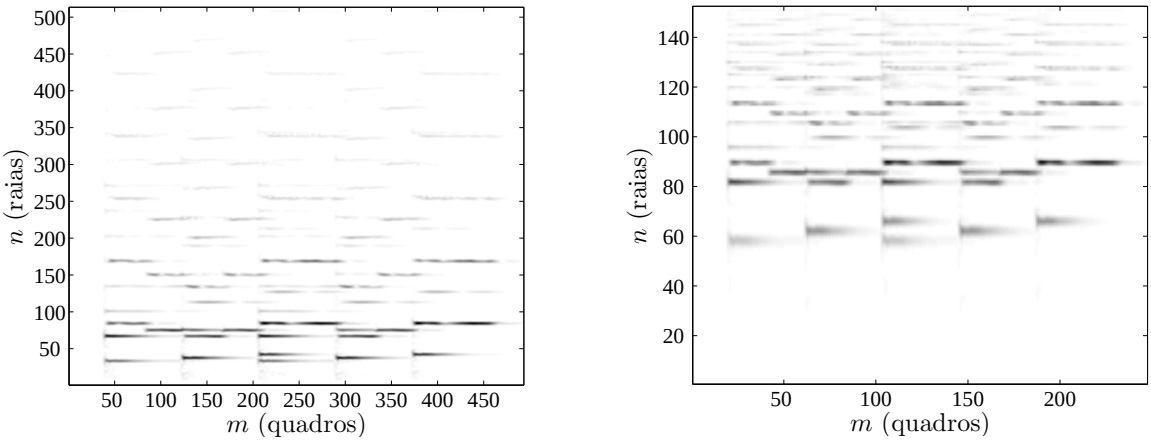


Figura 8.4: Resultados dos métodos de análise para o sinal `piano_trumpet_t.wav`: (a) STFT; (b) STFT mapeada.

Fatoração: O resultado da fatoração utilizando a LNMF2D e a NMF2D é mostrado na Figura 8.5. Na figura, podemos observar as matrizes $\mathbf{W}$, $\mathbf{H}$ e $\mathbf{\Lambda}$ para cada método.

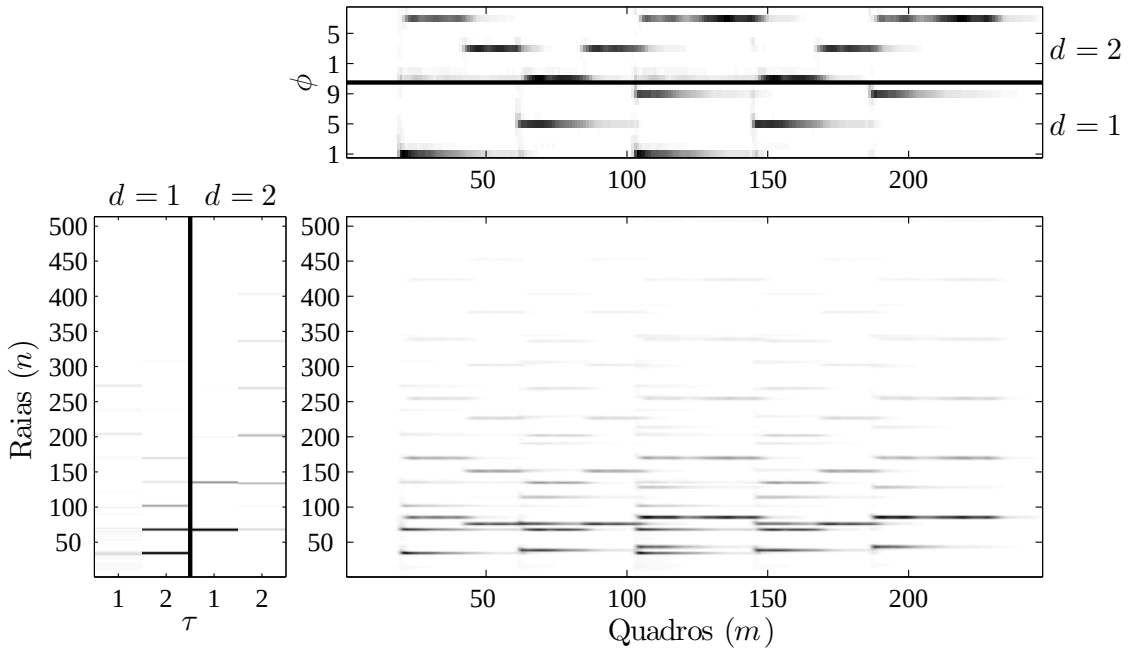
Espectrograma dos instrumentos separados: Os espectrogramas $\mathbf{\Lambda}_d$ são mostrados na Figura 8.6.

Convergência: As curvas de convergência para a distância euclidiana podem ser vistas na Figura 8.7.

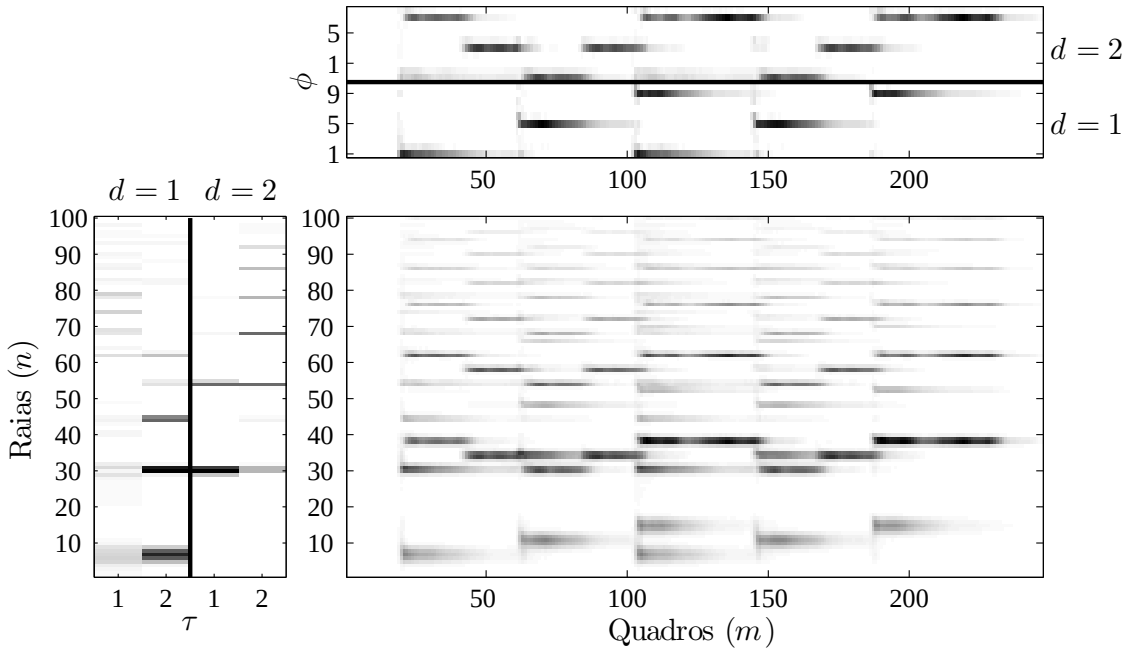
Instrumentos separados no domínio do tempo: Por fim, a Figura 8.8 mostra os sinais separados e ressintetizados, no domínio do tempo.

8.3 Adaptação Espectral

Nesta seção iremos analisar o algoritmo proposto na Seção 4.2. Neste caso, o objetivo é adaptar os padrões espectrais de modo a reduzir ainda mais o erro de reconstrução,



(a)



(b)

Figura 8.5: Resultados para a decomposição usando (a) LNMF2D e (b) NMF2D. O gráfico à esquerda mostra as bases espectrais para cada instrumento. O gráfico acima mostra a localização de cada base no tempo e *pitch*. O gráfico à direita abaixo mostra o espectrograma resultante, ou seja, a soma dos produtos de $\mathbf{W}$ e $\mathbf{H}$ para cada instrumento.

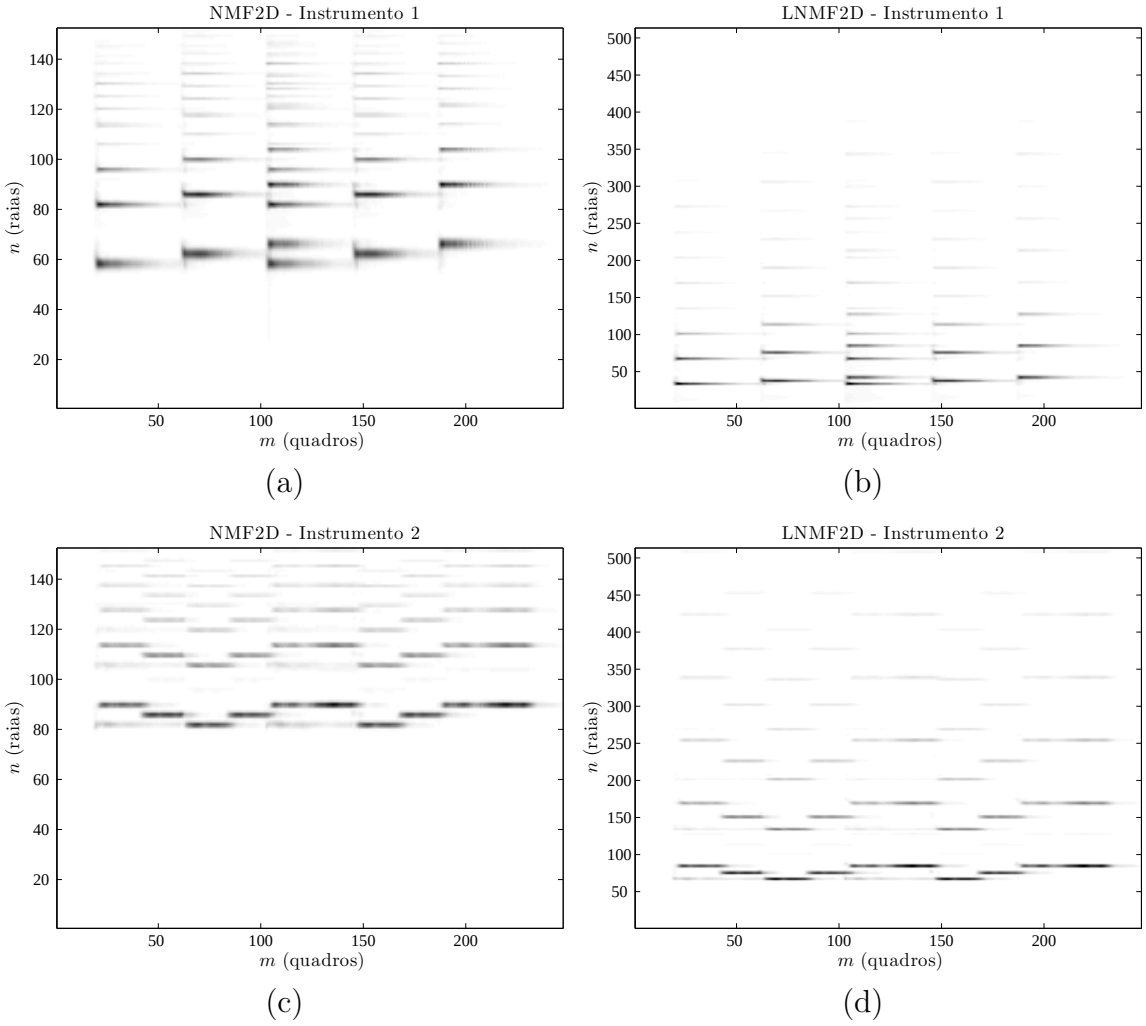


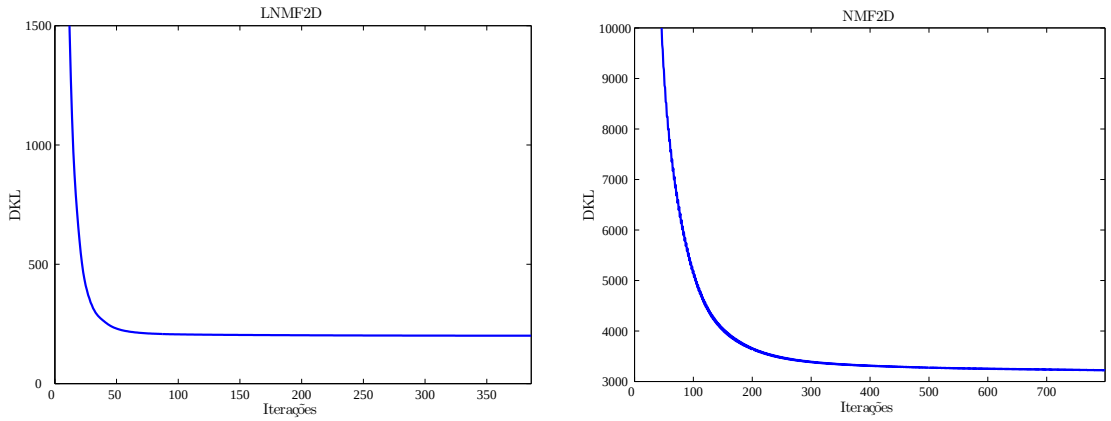
Figura 8.6: Espectrogramas dos instrumentos separados, resultantes dos métodos de fatoração para o sinal `piano_trumpet_t.wav` : (a) Instrumento 1 via NMF2D; (b) Instrumento 1 via LNMF2D; (c) Instrumento 2 via NMF2D; e (d) Instrumento 2 via LNMF2D.

possibilitando que as notas de um instrumento sejam diferentes entre si.

O ponto de partida destes experimentos será o melhor resultado da fatoração verificado no experimento anterior. O objetivo é observar uma queda ainda maior no erro de reconstrução, que supostamente ocasiona uma melhora nos indicadores objetivos de qualidade.

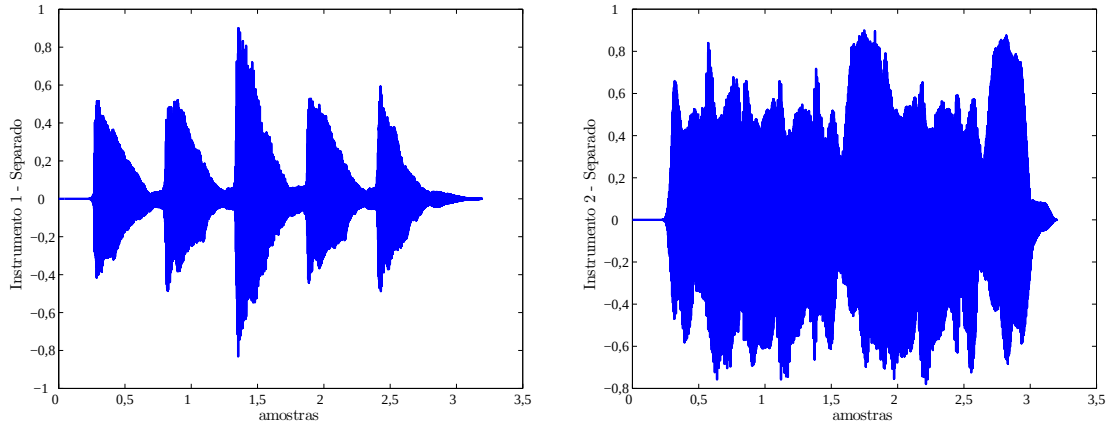
A Figura 8.9 mostra a curva de erro desde o início da fatoração até o fim da adaptação espectral, ressaltando a queda após o começo do algoritmo de refinamento. Vemos uma queda brusca no erro após as primeiras iterações do método, e em seguida há uma estabilização.

Na Figura 8.10, mostramos a evolução do erro, SDR e ODG a cada iteração da adaptação espectral, utilizando a divergência de Kullback-Leibler como medida de



(a) (b)

Figura 8.7: Curva de convergência (DKL) para o sinal `piano_trumpet_t.wav`: (a) LNMF2D e (b) NMF2D.



(a) (b)

Figura 8.8: Sinais separados, no domínio do tempo : (a) Instrumento 1; e (b) Instrumento 2

reconstrução. O sinal utilizado é o Sinal 6, formado por notas de um piano real. Como esperado, as 3 curvas são bastante correlacionadas: na primeira iteração, temos uma queda brusca no erro, e uma subida também brusca na SDR e na ODG. Em seguida, observamos uma tendência de estabilização nas 3 curvas, apesar de a SDR apresentar uma derivada mais acentuada do que as outras duas.

A Tabela 8.3 mostra a SDR e a ODG das melhores rodadas da Seção 8.2 e os valores após aplicação de 20 iterações da adaptação espectral. Cabe lembrar que neste algoritmo, o tensor $\mathbf{M}$ é inicializado com todos os elementos iguais a um, portanto ele é determinístico e não há a necessidade de várias rodadas para verificiar seu desempenho.

Analisando a tabela, vemos que apesar de a nota melhorar na maioria dos casos, há situações em que a adaptação espectral piora ou mantém a nota. No caso da SDR, a adaptação espectral melhora o resultado em 17 sinais, mantém em 3 e piora em 2 situações. Já para a ODG, a melhora ocorre em 11 casos, enquanto 6 sinais se mantêm iguais e em 5 o desempenho piora.

O algoritmo proposto visa à redução do erro de reconstrução; assim, faz sentido que o desempenho seja melhor em termos de SDR do que de ODG, já que a SDR está mais associada a erro, e a ODG utiliza critérios psicoacústicos. Este problema pode ser resolvido pensando-se numa função-objetivo que otimize algum critério psicoacústico, em lugar do erro.

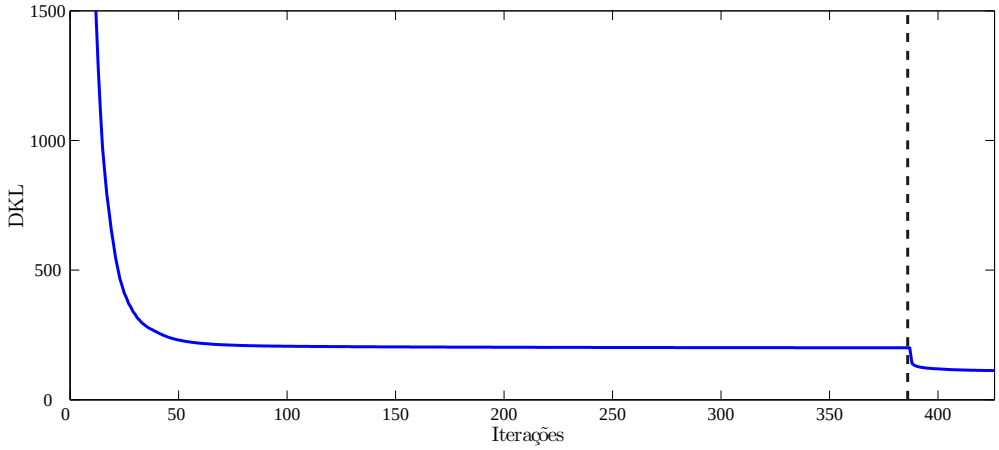


Figura 8.9: Curva de convergência utilizando adaptação espectral. A linha tracejada indica o ponto onde o algoritmo LNMF2D termina e a adaptação espectral se inicia.

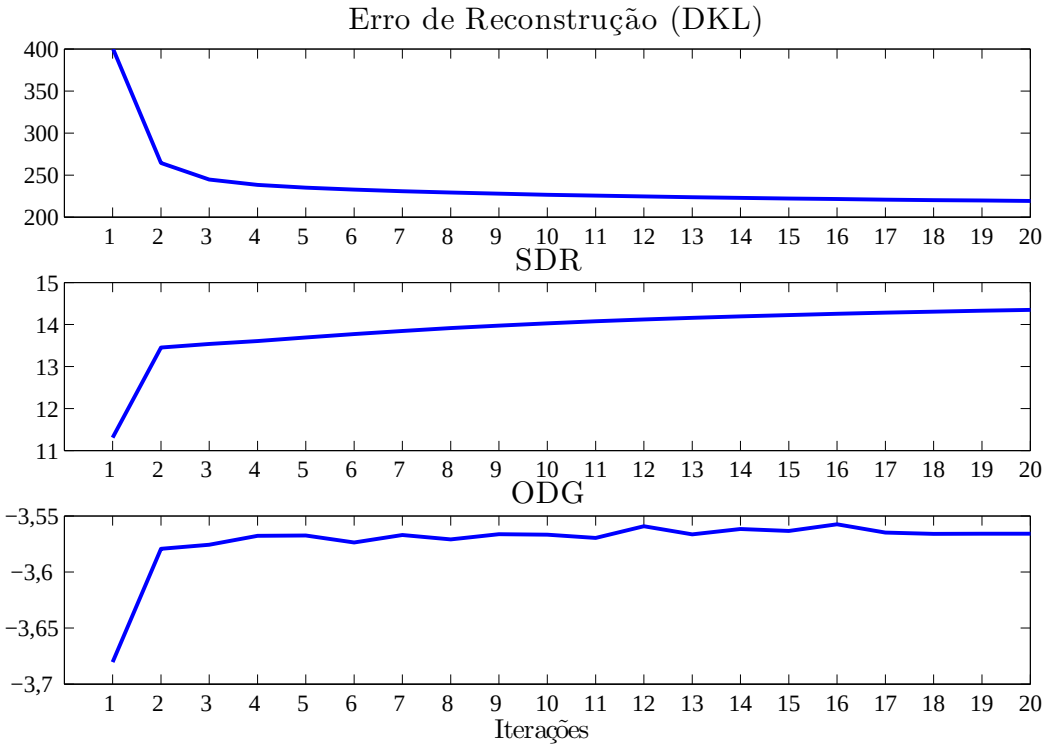


Figura 8.10: Erro (DKL), SDR e ODG em função do número de iterações de adaptação espectral.

Tabela 8.3: Medidas de qualidade antes e depois da aplicação do algoritmo de adaptação espectral. Na tabela, L denota LNMF2D, N quer dizer NMF2D e AE significa Adaptação Espectral.

		SDR				ODG			
Sinal	Instrumento	L	L+AE	N	N +AE	L	L+AE	N	N+AE
1	Percussão	7,54	7,54	7,07	7,34	-3,90	-3,90	-3,91	-3,91
1	Flauta	24,59	24,38	24,08	24,42	-2,41	-2,40	-3,19	-2,61
1	Órgão	12,44	12,37	11,77	11,85	-3,85	-3,83	-3,88	-3,88
2	Flauta	17,55	17,59	17,56	17,57	-2,60	-2,61	-3,43	-3,04
2	Percussão	8,55	8,75	8,30	8,55	-3,89	-3,89	-3,91	-3,90
3	Trompete	12,97	13,06	12,47	12,47	-3,37	-3,52	-3,79	-3,72
3	Piano	11,61	11,69	11,16	11,16	-3,50	-3,63	-3,52	-3,52
4	Piano	5,73	6,24	4,23	4,84	-3,80	-3,76	-3,69	-3,69
4	Trompete	12,35	12,48	11,77	12,18	-3,77	-3,48	-3,80	-3,69
5	Piano	13,83	14,32	12,15	12,25	-3,18	-3,32	-3,25	-3,46
5	Flauta	22,29	22,81	20,45	20,95	-2,44	-2,41	-3,45	-2,91

8.4 Síntese

Nesta seção, iremos verificar os algoritmos de síntese frente à opção de utilizar a fase da mistura original. Assim como na seção anterior, utilizaremos as melhores iterações mostradas na Seção 8.2. No entanto, aqui utilizaremos apenas o algoritmo LNMF2D. Desta forma, para um mesmo sinal, teremos a mesma fatoração e a única variação será no algoritmo de síntese.

A Tabela 8.4 mostra os resultados do ponto de vista da SDR. Apesar de se tratar de resultados auditivamente bons, vemos que a maioria dos valores de SDR são negativos para os métodos de estimação de fase, o que indicaria mais distorção do que sinal separado. A exceção é o método MISI, que possui alguns valores positivos. Isto pode ser justificado observando-se a equação (6.5) da página 64. Nela, os sinais separados recebem diretamente uma fração da mistura, o que melhora a qualidade do sinal, mas piora a separação.

Tabela 8.4: Valor da SDR para diversos métodos de síntese.

Sinal	Instrumento	G&L	RTISI	RTISI-LA	MISI	Fase da Mistura
1	Percussão	-20.12	-19.57	-21.70	-13.50	6.69
1	Flauta	-14.24	-17.32	-36.60	9.23	25.09
1	Órgão	-17.30	-14.07	-10.21	-11.73	12.28
2	Flauta	-6.96	-10.63	-13.38	16.31	18.50
2	Percussão	-19.27	-22.80	-31.14	-15.99	7.40
3	Trompete	-13.69	-11.82	-25.15	1.53	12.90
3	Piano	-20.34	-18.09	-35.50	-0.88	11.51
4	Piano	-12.01	-15.84	-14.15	-8.19	5.16
4	Trompete	-34.72	-31.58	-17.93	6.72	12.46
5	Piano	-21.64	-24.52	-39.50	-8.01	13.28
5	Flauta	-18.61	-22.29	-19.66	7.72	22.15

A Figura 8.11 nos ajuda a entender o porquê dos maus resultados na avaliação de qualidade. Na figura utilizamos como exemplo o sinal `piano_flute`; a coluna da esquerda se refere ao sinal de um dos instrumentos separado e sintetizado com a fase da mistura, e a coluna da direita se refere ao sinal do mesmo instrumento separado e sintetizado através do método RTISI. Na primeira linha, vemos que as envoltórias dos dois sinais no tempo não apresentam grandes diferenças. Da mesma maneira, a segunda linha mostra que os espectrogramas de ambos os sinais sintetizados são semelhantes.

Entretanto, a terceira linha da Figura 8.11 revela grandes diferenças quando olhamos o sinal no tempo com maior grau de detalhe. As formas de onda são diferentes, e isto pode explicar o mau desempenho na avaliação de qualidade via SDR. Dado que os espectrogramas de magnitude são parecidos, e ambos os sinais soam bem, a explicação para a SDR negativa se coloca na fase do sinal sintetizado pelo método RTISI. Mais especificamente, as parciais do sinal estão todas presentes (vide espectrograma), mas as fases não são coerentes entre si. É como se cada parcial possuísse um atraso diferente, mas devido à insensibilidade do nosso ouvido a diferenças de fase em regime permanente [17], os sinais soam de maneira parecida.

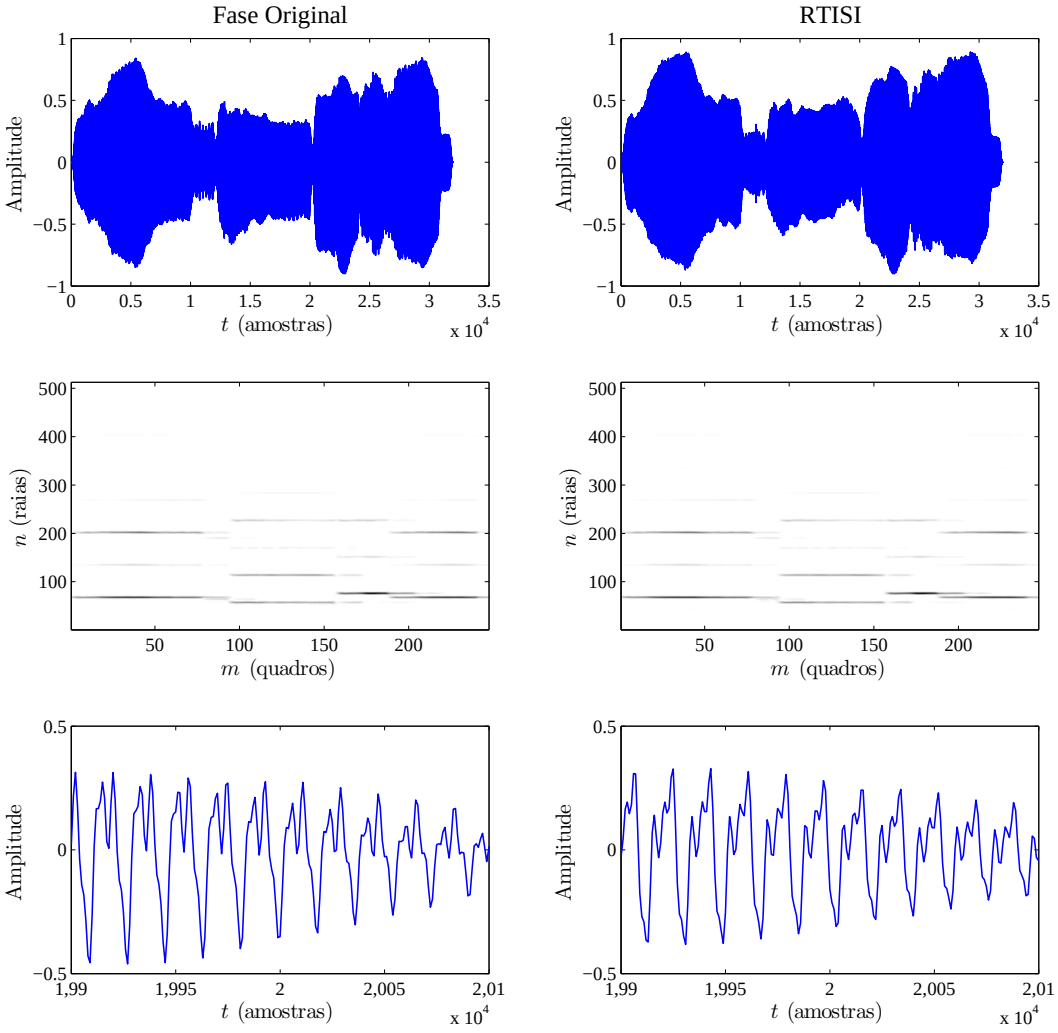


Figura 8.11: Comparação entre sinais sintetizados com a fase da mistura (coluna da esquerda) e com a fase estimada pelo RTISI (coluna da direita). Na primeira linha, vemos a envoltória dos sinais no tempo. Em seguida, vemos na segunda linha o espectrograma gerado por cada método de síntese. Na terceira linha, o detalhe na forma de onda revela diferenças que explicam o mau desempenho na avaliação de qualidade, apesar de resultados auditivamente satisfatórios.

Neste cenário, a avaliação de qualidade através de medidas baseadas em psicoacústica pode trazer grandes vantagens. Afinal, qualquer que seja o formato de onda, se os sinais soam de maneira semelhante, a avaliação deve ser boa. A Tabela 8.5 mostra o resultado da avaliação pelo PEAQ dos sinais sintetizados pelos métodos de estimação de fase, em comparação com o uso da fase da mistura. Na tabela vemos que os métodos G&L, RTISI e RTISI-LA atingem marcas comparáveis àquelas atingidas com o uso da fase da mistura. O método MISI, por sua vez, teve resultados ruins, que são esperados quando ouvimos a síntese feita através dele. É interessante notar que este método foi único a obter SDR positivo em alguns casos na Tabela 8.4, o que mostra a superioridade da avaliação através do PEAQ nos casos de estimação de fase.

Em 5 dos 11 instrumentos mostrados na Tabela 8.5, os métodos RTISI e RTISI-LA superam o uso da fase da mistura. O desempenho do algoritmo G&L é um pouco pior, o que já era esperado pelos resultados descritos na literatura. Com isso, podemos concluir que as vantagens trazidas pelos métodos de estimação de fase são dependentes do sinal. Mais uma vez a esparsidade tempo-frequencial tem grande influência no resultado, já que caso a fonte a ser sintetizada não tenha sobreposição com nenhuma outra fonte, a síntese com a fase da mistura será bem sucedida. Deste modo, quanto maior a sobreposição entre fontes, maiores os danos causados por este procedimento e maiores serão as vantagens que a estimação de fase pode oferecer.

É interessante notar que, ao contrário do algoritmo que calcula a SDR, aquele que calcula a ODG é sensível à diferença de potência, ou seja, caso se deseje comparar dois sinais idênticos a menos de um escalamento, a nota resultante será baixa. Devido a este problema, o sistema desenvolvido nesta dissertação realiza uma equalização de potência antes da avaliação de qualidade, utilizando um algoritmo distribuído juntamente com o PEAQ em [55]. Apesar de apresentar bons resultados na síntese que utiliza a fase da mistura, o algoritmo de equalização de potência falha quando aplicado a sinais sintetizados pelo métodos de estimação de fase. E esta falha está diretamente relacionada às diferenças de fase entre os sinais, já que o algoritmo trabalha com uma comparação ponto-a-ponto.

Portanto, os sinais da Tabela 8.5 não foram equalizados em potência, ao contrário do que ocorreu em todos os outros resultados de ODG apresentados neste trabalho.

Contudo, a primeira linha da Figura 8.11 nos dá uma idéia de que a diferença não é acentuada.

Tabela 8.5: Valor da ODG para diversos métodos de síntese.

Sinal	Instrumento	G&L	RTISI	RTISI-LA	MISI	Fase da Mistura
1	Percussão	-3.89	-3.91	-3.91	-3.91	-3.90
1	Flauta	-2.63	-2.45	-2.47	-3.91	-2.50
1	Órgão	-3.34	-3.57	-3.56	-3.91	-3.08
2	Flauta	-3.25	-2.75	-2.72	-3.90	-2.85
2	Percussão	-3.87	-3.91	-3.91	-3.91	-3.86
3	Trompete	-3.88	-3.86	-3.87	-3.91	-3.88
3	Piano	-3.79	-3.78	-3.79	-3.90	-3.65
4	Piano	-3.77	-3.59	-3.60	-3.82	-3.54
4	Trompete	-3.79	-3.50	-3.49	-3.91	-3.55
5	Piano	-3.52	-3.23	-3.17	-3.90	-2.93
5	Flauta	-2.59	-2.36	-2.36	-3.91	-2.42

A diferença dos resultados apresentados nas Tabelas 8.4 e 8.5 evidencia a necessidade de uma avaliação que leve em conta critérios psicoacústicos. Nesta seção, tivemos um exemplo claro em que o método baseado em SNR falha completamente, e o método baseado em psicoacústica mostra resultados coerentes com a audição.

□

Como já se mencionou no Capítulo 6, houve muita dificuldade na implementação dos métodos devido à falta de clareza no artigos citados. Esta situação traz à tona a discussão sobre reprodutibilidade da pesquisa científica. Em ótimo artigo [14] publicado na mais importante revista de processamento de sinais do IEEE, os autores Patrick Vandewalle, Jelena Kovačević e Martin Vetterli salientam a importância de se reportar trabalhos de forma que eles possam ser reproduzidos por outros pesquisadores. Isto inclui a descrição de todos os parâmetros e configurações, e a disponibilização dos dados e códigos utilizados para gerar tabelas e figuras. Infelizmente, estes preceitos não foram seguidos em boa parte dos artigos citados no Capítulo 6.

8.5 Processamento *Online*

Nesta seção iremos analisar o desempenho do algoritmo *online* proposto. Como não foi encontrada nenhuma proposta semelhante na literatura, iremos comparar os resultados do algoritmo *online* com aqueles do algoritmo *offline* convencional. A expectativa é de que o algoritmo convencional use grande quantidade de memória, já que terá que lidar com matrizes grandes.

Em termos de qualidade, é esperado que o algoritmo *online* ao menos mantenha os indicadores no mesmo nível daqueles obtidos para o *offline*. Para o teste, será utilizado o sinal 14, que possui duração de 5 minutos e 20 segundos. Cada blocos do algoritmo *online* tem tamanho de 200 janelas de análise de 512 amostras, o que equivale a 6,46 segundos com sobreposição de 50%.

A Figura 8.12 mostra o uso de memória RAM durante o processamento. Os valores foram medidos observando-se a cada meio segundo o campo `VmRSS` do arquivo `/proc/PID/status`, do sistema UNIX. Este campo reporta a memória do tipo *resident*, que é aquela que está fisicamente alocada e sendo usada pelo processo. A memória que está alocada apenas virtualmente não é levada em conta, assim como a memória ocupada por bibliotecas compartilhadas. As primeiras amostras do gráfico mostram o uso de memória pelo processo antes do início dos algoritmos. Após o início, o algoritmo *offline* passa pelo trecho de análise tempo-frequência, onde a ocupação de memória cresce linearmente à medida em que constrói a matriz do espectrograma. Apesar de a memória ser alocada previamente, o crescimento é linear porque a memória só é de fato usada quando se completa a transformada em cada quadro. Após este período, começa a fatoração, onde, além do espectrograma, temos os tensores $\mathbf{W}$ e $\mathbf{H}$ e as variáveis auxiliares ocupando cerca de 450 MB acima do patamar inicial. Após a fatoração, o algoritmo começa o processo de avaliação de qualidade, onde há um crescimente brusco na ocupação de memória, e em seguida sua execução é encerrada.

Já no algoritmo *online* proposto, após o início da execução a ocupação é de cerca de 50 MB acima do patamar inicial. Esta ocupação sobe lentamente, devido à acumulação dos resultados da separação em cada bloco. Após subir cerca de 90 MB, inicia-se a etapa de avaliação de qualidade, onde, a exemplo do algoritmo *offline*, há um aumento acentuado na ocupação de memória, já que a avaliação é feita

utilizando-se o sinal inteiro.

Como esperado, a menor utilização de memória no algoritmo *online* vem acompanhada de um maior tempo de execução. Isto pode ser explicado pela sobreposição entre os blocos, o que resulta em aproximadamente o dobro de operações realizadas na etapa de fatoração. De fato, o tempo gasto na fatoração pelo algoritmo *offline* é de aproximadamente metade do tempo gasto pelo algoritmo *online*.

A Figura 8.13 mostra o espectrograma original de uma das fontes, e o resultante da separação para os métodos *online* e *offline*. Vemos que os três espectrogramas são parecidos, o que indica que o algoritmo proposto foi capaz de manter a ordenação das fontes, que era um dos objetivos. A Tabela 8.6 mostra que a diferença nos indicadores de qualidade manteve-se dentro da variância do Sinal 3 na Figura 8.2, indicando que os métodos podem ser considerados equivalentes em termos de qualidade.

Tabela 8.6: Comparação entre os algoritmos *online* e *offline*.

Instrumento	SDR *online*	SDR *offline*	ODG *online*	ODG *offline*
1	11,08±2,23	10,57±2,99	-3,85 ±0,03	-3,85±0,04
2	9,61±2,36	8,91± 3,59	-3,76 ±0,05	-3,77±0,06

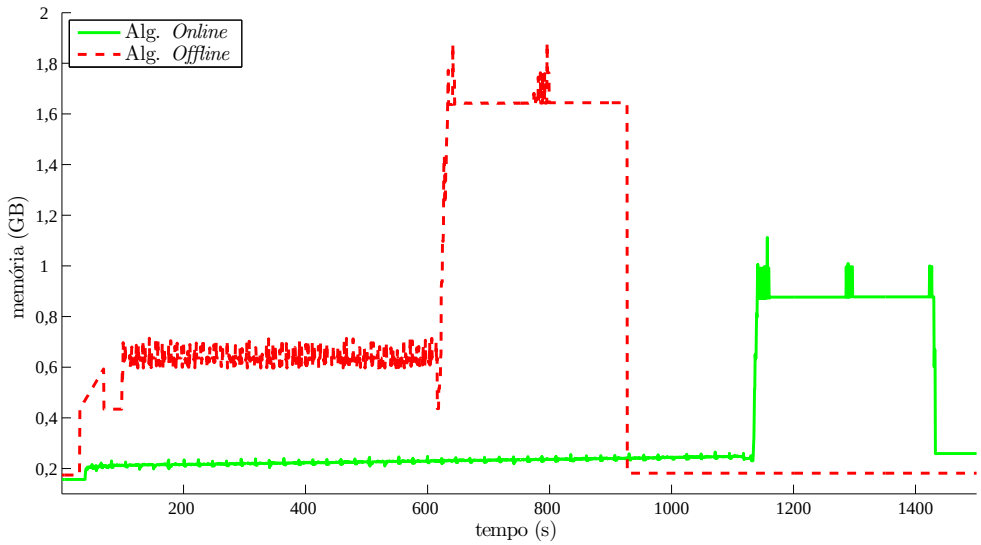


Figura 8.12: Uso de memória pelos algoritmos *offline* e *online*. Durante a fatoração, o algoritmo *online* utiliza até cerca de 90 MB acima do patamar inicial, enquanto o algoritmo *offline* utiliza até 450 MB acima do mesmo patamar. A subida repentina no final do processamento, em ambos os sinais, ocorre por ocasião da avaliação de qualidade. Pode-se perceber que o algoritmo *online* termina a execução mais tarde, devido à sobreposição entre blocos. Os resultados de avaliação de qualidades são equivalentes, como mostra a Tabela 8.6.

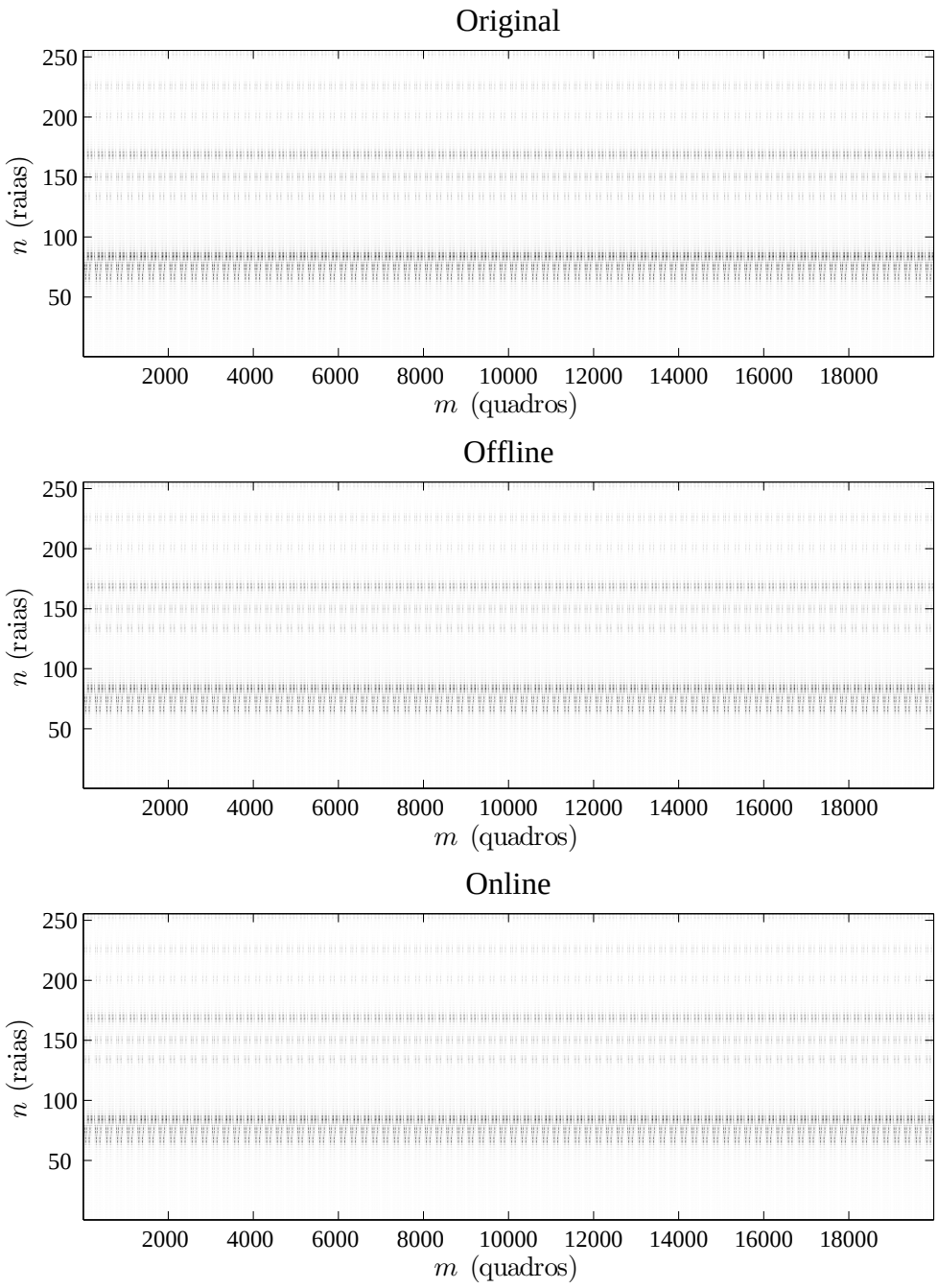


Figura 8.13: Comparação entre espectrogramas original, separado pelo algoritmo *offline* e pelo *online*. Os espectrogramas semelhantes confirmam que o algoritmo *online* conseguiu manter a ordenação das fontes entre os blocos, de modo que o instrumento separado contém informação sempre da mesma fonte.

Capítulo 9

Conclusões

9.1 Contribuições desta Dissertação

Esta dissertação desenvolveu um sistema completo de separação de sinais utilizando métodos de fatoração de matrizes não-negativas. Além da descrição de todos os blocos do sistema, contendo diversos algoritmos da literatura, as seguintes propostas foram apresentadas:

LNMF2D: A *Linear Non-Negative Matrix Factor 2D Deconvolution* foi apresentada como forma de superar a exigência do uso de um espectrograma com espaçamento logarítmico pela NMF2D. A LNMF2D possibilita o uso do algoritmo de fatoração com um espectrograma linearmente espaçado, eliminando a aproximação que é feita na inversão de espectrogramas logarítmicos ao domínio do tempo, e facilitando a etapa de síntese dos instrumentos separados. A desvantagem se dá na necessidade de uma aproximação no operador de deslocamento vertical.

Os experimentos confirmaram que o uso da LNMF2D traz benefícios à qualidade final dos sinais separados, na grande maioria dos casos. Este trabalho foi publicado em [39].

Adaptação dos Padrões Espectrais: Esta contribuição foi proposta como forma de tornar o modelo da (L)NMF2D mais adequado aos sinais reais. A exigência de que um instrumento emita o mesmo padrão espectral para todas as notas foi suavizada com a adição de um tensor de adaptação no modelo. A adaptação

espectral permite que os padrões se adaptem a cada nota, com a contrapartida de um aumento na complexidade computacional do algoritmo.

Os experimentos mostraram que a aplicação deste refinamento ao algoritmo de fatoração resulta num decréscimo no erro de reconstrução, o que na maioria dos casos significa uma melhora nos indicadores de qualidade.

Algoritmo *Online*: O tratamento de sinais longos ainda não havia sido abordado na literatura sobre métodos de fatoração de matrizes não-negativas para separação de sinais. Foi proposto um procedimento que permite tratar o sinal em partes, mantendo a ordenação das fontes e uma baixa utilização de memória. Além de sinais longos, o procedimento também pode lidar com o caso em que não se dispõe do sinal completo para o processamento.

Os resultados mostraram que de fato o método utiliza uma quantidade menor de memória durante o processamento e mantém a ordenação das fontes entre os blocos. A qualidade final do sinais separados é equivalente àquela que seria obtida aplicando-se o mesmo método no modo *offline*.

Avaliação de Qualidade de Separação: Foi proposta a utilização do algoritmo PEAQ para a avaliação da qualidade da separação de sinais de áudio. Por utilizar um modelo psicoacústico, este método apresenta vantagens sobre os tradicionalmente utilizados na literatura. A comparação desta métrica com a SDR, tradicionalmente usada na literatura, mostrou coerência nos resultados. Entretanto, para confirmar sua eficácia nesta aplicação, será necessária a realização de testes subjetivos.

***SoundFact*:** Como produto desta dissertação, o *software SoundFact* encontra-se disponível no endereço `http://www.lps.ufrj.br/~alan/SoundFact`. Nele estão implementados todos os algoritmos citados neste trabalho, permitindo a reprodução dos experimentos aqui mencionados. As únicas implementações disponiblizadas por autores encontradas tratam apenas de partes do sistema, e portanto a disponibilização de um sistema completo é uma contribuição para futuras pesquisas na área.

9.2 Trabalhos Futuros

As sugestões para continuação deste trabalho são motivadas por problemas encontrados durante sua elaboração, e que não puderam ser resolvidos. Elas podem ser dividas em três classes:

Fatoração: Os algoritmos mencionados neste trabalho pressupõem que o número de instrumentos na mistura é conhecido. Entretanto, para um sistema não-supervisionado o ideal seria que este valor fosse estimado automaticamente. Além disso, os valores de τ (duração de cada matriz da base espectral) e ϕ (faixa de variação das notas) também deveriam ser estimados automaticamente.

Os algoritmos de otimização, incluindo medidas de distância e outros critérios, também deveriam ser alvo de pesquisa específica. É necessário realizar análises de convergência e propor critérios que melhorem a convergência para sinais musicais, sem no entanto criar condições que o tornem específicos demais.

Além disso, o modelo utilizado pelos algoritmos tratados é apropriado para instrumentos sintéticos, mas no caso dos instrumentos reais se distancia demais da realidade. Neste caso, seria interessante propor soluções baseadas em psicoacústica, modelando o que faz nosso sistema auditivo distinguir entre dois instrumentos.

Síntese: Apesar de obter resultados bons do ponto de vista psicoacústico, os métodos de síntese apresentam péssima coerência de fase, o que prejudica muito a síntese no caso de ataques rápidos. É necessário desenvolver métodos de síntese que consigam manter a coerência de fase entre as parciais.

Avaliação de Qualidade: Apesar de termos mostrado fortes indícios de que o PEAQ é uma medida válida para avaliação da qualidade da separação, a falta de testes subjetivos deixa uma lacuna que deve ser preenchida. É fundamental a construção de uma base de dados seguida de avaliações subjetivas, de modo que se possa validar os resultados do PEAQ (e possivelmente de suas métricas internas), além de outros avaliadores psicoacústicos.

□

Desta forma, encerra-se esta dissertação, esperando que mais um passo tenha sido dado na busca da solução para o ainda não resolvido problema da separação de fontes.

Apêndice A

Algoritmo de Mapeamento de Espectrogramas

Este apêndice mostra o algoritmo, em linguagem de Matlab©, que realiza a construção da matriz $\mathbf{C} \in \mathbb{R}^{N_{\log} \times N_{\text{lin}}}$ apresentada na equação (2.9). Esta matriz faz o mapeamento entre um espectrograma linearmente espaçado e outro logaritmicamente espaçado.

Em linhas gerais, o objetivo da matriz é agrupar as raias do espectrograma linear, que ocorrem em número diferente a cada oitava, em raias logarítmicas, que ocorrem em igual quantidade a cada oitava. O algoritmo passa por todas as raias lineares, e a partir da frequência central que representam, escolhe para qual ou quais das raias logarítmicas deve ir a energia da raia linear. Esta escolha é feita considerando que cada raia logarítmica ocupa uma banda que vai da metade de sua distância à frequência central da raia inferior até a metade da distância à frequência central da raia superior. Caso deva ir para mais de uma raia logarítmica, a energia total deve mantida constante.

A Figura A.1 ilustra o procedimento. Tomando como exemplo a raia linear 1, vemos sua banda delimitada pelas linhas cheias. Podemos ver que parte da sua energia deve ir para a raia logarítmica A, parte para a B, e parte para a C. As bandas logarítmicas das raias A, B e C têm interseção com a banda de 1, portanto cada uma deve receber uma quantidade de energia da banda 1 proporcional à area de interseção. Desta forma, as colunas da matriz $\mathbf{C}$ têm soma unitária, mostrando que a energia total é mantida no mapeamento.

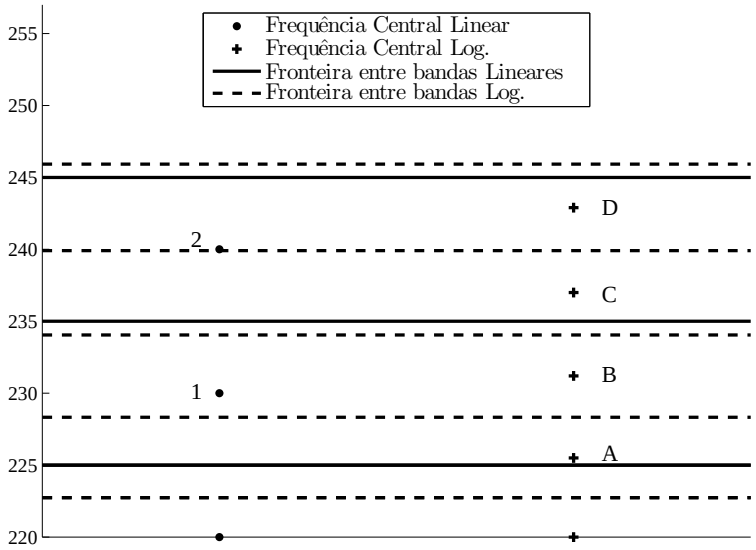


Figura A.1: Mapeamento linear-logarítmico. A energia da banda referente à raia linear cuja frequência central é 1 deve ser transferida para as bandas das raias logarítmicas A, B e C. Já a energia da banda linear 2 deve ser distribuída entre C e D. Esta distribuição é feita proporcionalmente à área ocupada dentro de cada banda.

Entrada: Frequência mínima `fMin`, frequência máxima `fMax`, frequência de amostragem `Fs`, tamanho da janela `WS`, tamanho da FFT `FFT_size`, número de raias por oitava `b`.

```
function C = SF_log_frequency_mapping(fMin, fMax, Fs, WS, FFT_size ,b)

% C = SF_log_frequency_mapping(fMin, fMax, Fs, WS, FFT_size ,b)
%
% Builds a linear to log mapping matrix
% INPUT:
% fMin: Lower frequency
% fMax: Highestfrequency
% FS: Sample Rate
% WS: Window Size
% FFT_size: must be greater than WS
% b: log bins for each octave
%
% OUTPUT
% C: linear to log mapping matrix
%
% Author: Alan F. Tygel
% Last Modified: 02/11/2009

  t = 1;
  k = 1;
  f = fMin;

  n_log_bins = round(log2(fMax/fMin)*b);

  %number of logarithmic bins
```

```
n_log_bins = round(log2(fMax/fMin)*b);

%transformation matrix
C = zeros(n_log_bins, FFT_size/2);

for l_b = 1:FFT_size/2
  %central freq of lin bin
  f = lin_bin_to_freq(l_b,FFT_size,Fs);

  if(f >= fMin)&&(f <= fMax)

    f_top = lin_bin_to_freq(l_b+1,FFT_size,Fs); %freq. of upper bin
    f_bot = lin_bin_to_freq(l_b-1,FFT_size,Fs); %freq. of bottom bin

    f_topedge = (f_top+f)/2; %top edge of linear bin (freq)
    f_botedge = (f_bot+f)/2; %bottom edge of linear bin (freq)
    band_size = f_topedge - f_botedge; %freq bandsize

    %top edge of linear bin (log bin)
    [log_b_topedge log_b_topedge_b log_b_topedge_t] = freq_to_log_bin(f_topedge,fMin,b);
    %bot edge of linear bin (log bin)
    [log_b_botedge log_b_botedge_b log_b_botedge_t] = freq_to_log_bin(f_botedge,fMin,b);

    %frequency of the geometric mean between bins top and bottom of the edge
    flog_b_fronteira_top = log_bin_to_freq(log_b_topedge_t-.5,fMin,b);
    flog_b_fronteira_bot = log_bin_to_freq(log_b_botedge_b+.5,fMin,b);

    %verifiyng if they are inside the band; else, take the first inside
    if(flog_b_fronteira_top < f_topedge)
      flog_b_fronteira_top = log_bin_to_freq(log_b_topedge_t+.5,fMin,b);
    end
    if(flog_b_fronteira_bot > f_botedge)
      flog_b_fronteira_bot = log_bin_to_freq(log_b_botedge_b-.5,fMin,b);
    end

    %log bin under top edge
    [xx log_b_topedge xx] = freq_to_log_bin(flog_b_fronteira_top,fMin,b);

    %log bin over bottom edge
    [xx xx log_b_botedge] = freq_to_log_bin(flog_b_fronteira_bot,fMin,b);
```

```
      %assign energies for each log bin
      for log_b = log_b_botedge:log_b_topedge
        if((log_b > 0) && (log_b <= n_log_bins))
          bot_edge = log_bin_to_freq(log_b-.5,fMin,b);

          if(bot_edge < f_botedge)
            bot_edge = f_botedge;
          end

          top_edge = log_bin_to_freq(log_b+.5,fMin,b);
          if(top_edge > f_topedge)
            top_edge = f_topedge;
          end

          log_band_size = top_edge - bot_edge;
          %assign proportional energy
          C(log_b,l_b) = C(log_b,l_b) + log_band_size/band_size;
        end
      end
    end
  end
end

function f = lin_bin_to_freq(bin,WS,Fs)
  bin_width = Fs/WS;
  f = bin*bin_width;
end

function [b1 b2] = freq_to_lin_bin(f,WS,Fs)
  b = f/Fs*WS;
  b1 = floor(b);
  b2 = ceil(b);
end

function f = log_bin_to_freq(bin,fMin,b)
  f = fMin*power(2,bin/b);
end

function [bin b1 b2] = freq_to_log_bin(f,fMin,b)
  bin = log2(f/fMin)*b;
  b1 = floor(bin);
  b2 = ceil(bin);
end
```

Saída: Matriz $\mathbf{C} \in \mathbb{R}^{N_{\log} \times N_{\text{lin}}}$.

Referências Bibliográficas

[1] CHERRY, E. C. "Some experiments on the recognition of speech, with one and with two ears", *Journal of the Acoustic Society of America*, v. 26, pp. 554–559, Jul. 1954.

[2] FERREIRA, D. D., SÁ, A. M., CERQUEIRA, A. S., et al. "ICA-Based Method for Quantifying EEG Event-Related Desynchronization". In: *Proceedings of the 8th International Conference on Independent Component Analysis and Signal Separation*, v. 5441, pp. 403–410, Paraty, Mar. 2009.

[3] MAKEIG, S., BELL, A. J., JUNG, T. P., et al. "Independent component analysis of electroencephalographic data", *Advances in Neural Information Processing Systems*, v. 8, pp. 145–151, Jun. 1996.

[4] HAYKIN, S. *Unsupervised Adaptive Filtering, Volume 1: Blind Source Separation*. New York, Wiley-Interscience, 2000.

[5] HYVÄRINEN, A., KARHUNEN, J., OJA, E. *Independent Component Analysis*. New York, John Wiley & Sons, 2001.

[6] HYVAÄRINEN, A. "Fast and Robust Fixed-Point Algorithms for Independent Component Analysis", *IEEE Transactions on Neural Networks*, v. 10, n. 3, pp. 626–634, Maio 1999.

[7] CARDOSO, J.-F. "High-order contrasts for independent component analysis", *Neural Computation*, v. 11, n. 1, pp. 157–192, Jul. 1999.

[8] BELOUCHRANI, A., ABED-MERAIM, K., CARDOSO, J.-F., et al. "A blind source separation technique using second-order statistics", *IEEE Transactions on Signal Processing*, v. 45, n. 2, pp. 434–444, Fev. 1997.

[9] LEE, D. D., SEUNG, S. H. "Learning the parts of objects by non-negative matrix factorization", *Nature*, v. 401, pp. 788–791, Out. 1999.

[10] SMARAGDIS, P., BROWN, J. C. "Non-negative matrix factorization for polyphonic music transcription". In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 177–180, New Paltz, Out. 2003.

[11] LEE, D. D., SEUNG, S. H. "Algorithms for Non-negative Matrix Factorization", *Neural Information Processing Systems*, v. 13, pp. 556–562, Abr. 2001.

[12] SMARAGDIS, P. "Non-negative Matrix Factor Deconvolution; Extracation of Multiple Sound Sources from Monophonic Inputs". In: *Proceedings of the 5th International Congress on Independent Component Analysis and Blind Signal Separation*, v. 3195, pp. 494–499, Grenada, Set. 2004.

[13] SCHMIDT, M. N., MØRUP, M. "Nonnegative Matrix Factor 2-D Deconvolution for Blind Single Channel Source Separation". In: *Proceedings of the 6th International Conference on Independent Component Analysis and Blind Signal Separation*, v. 3889, pp. 700–707, Charleston, Mar. 2006.

[14] VANDEWALLE, P., KOVACEVIC, J., VETTERLI, M. "Reproducible research in signal processing", *IEEE Signal Processing Magazine*, v. 26, n. 3, pp. 37–47, Maio 2009.

[15] DA SILVA, E. A. B., NETTO, S. L., DINIZ, P. S. R. *Digital Signal Processing: System Analysis and Design*. United Kingdom, Cambridge, 2002.

[16] COHEN, L. *Time Frequency Analysis: Theory and Applications*. New Jersey, Prentice Hall PTR, 1994.

[17] FASTL, H., ZWICKER, E. *Psychoacoustics: Facts and Models*. 3 ed. Berlin, Springer, 2006.

[18] MED, B. *Teoria da Música*. 4 ed. Brasília, Musimed, 2001.

[19] BROWN, J. C. "Calculation of a Constant Q Spectral Transform", *Journal of the Acoustical Society of America*, v. 89, pp. 425–434, Jan. 1991.

[20] BROWN, J. C., PUCKETTE, M. S. "An Efficient Algorithm for the Calculation of a Constant Q Transform", *Journal of the Acoustical Society of America*, v. 92, pp. 2698–2701, Nov. 1992.

[21] SEDRA, A. S., SMITH, K. C. *Microelectronic Circuits*. Oxford, Oxford University Press, 2004.

[22] FITZGERALD, D., CRANITCH, M., CYCHOWSKI, M. "Towards an Inverse Constant Q Transform". In: *120th AES Convention*, Paris, Maio 2006. Preprint 6671.

[23] FITZGERALD, D., CRANITCH, M., COYLE, E. "Resynthesis methods for Sound Source Separation using shifted Non-negative Factorisation Models". In: *Proceedings of the Irish Signals and Systems Conference*, Derry, Set. 2007.

[24] DONOHO, D., STODDEN, V. "When Does Non-Negative Matrix Factorization Give a Correct Decomposition into Parts?" *Advances in Neural Information Processing Systems*, v. 16, Jun. 2004.

[25] BERRY, M. W., BROWNE, M., LANGVILLE, A. N., et al. "Algorithms and applications for approximate nonnegative matrix factorization", *Computational Statistics & Data Analysis*, v. 52, n. 1, pp. 155–173, Set. 2007.

[26] ANTONIOU, A., LU, W.-S. *Practical Optimization: Algorithms and Engineering Applications*. New York, Springer Publishing Company, Incorporated, 2007.

[27] VIRTANEN, T. "Monaural Sound Source Separation by Nonnegative Matrix Factorization With Temporal Continuity and Sparsen ess Criteria", *IEEE Transactions on Audio, Speech and Language Processing*, v. 15, n. 3, pp. 1066–1074, Mar. 2007.

[28] HOYER, P. O. "Non-negative matrix factorization with sparseness constraints", *Journal of Machine Learning Research*, v. 5, pp. 1457–1469, Nov. 2004.

[29] SCHACHTNER, R., PÖPPEL, G., TOMÉ, A. M., et al. "Minimum Determinant Constraint for Non-negative Matrix Factorization". In: *Proceedings of the 8th International Conference on Independent Component Analysis and Signal Separation*, v. 5441, pp. 106–113, Paraty, Mar. 2009.

[30] CICHOCKI, A., ZDUNEK, R., AMARI, S.-I. "Csiszar's divergences for non-negative matrix factorization: Family of new algorithms". In: *Proceedings of the 6th International Conference on Independent Component Analysis and Blind Signal Separation*, v. 3889, pp. 32–39, Charleston, Mar. 2006.

[31] CICHOCKI, A., ZDUNEK, R., AMARI, S.-I. "New Algorithms for Non-Negative Matrix Factorization in Applications to Blind Source Separation". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, v. 5, pp. V–V, Toulouse, Maio 2006.

[32] ZDUNEK, R., CICHOCKI, A. "Fast nonnegative matrix factorization algorithms using projected gradient approaches for large-scale problems", *Computational Intelligence and Neuroscience*, v. 2008, n. 3, pp. 1–13, Jun. 2008.

[33] FÉVOTTE, C., BERTIN, N., DURRIEU, J.-L. "Nonnegative Matrix Factorization with the Itakura-Saito Divergence: With Application to Music Analysis", *Neural Computation*, v. 21, n. 3, pp. 793–830, Mar. 2009.

[34] FITZGERALD, D., CRANITCH, M., COYLE, E. "Shifted 2D Non-negative Tensor Factorisation". In: *Proceedings of the Irish Signals and Systems Conference*, pp. 509–513, Dublin, Jun. 2006.

[35] FITZGERALD, D., CRANITCH, M., COYLE, E. "Musical Source Separation using Generalised Non-negative Tensor Factorisation Models". In: *Workshop on Music and Machine Learning, International Conference on Machine Learning*, pp. 1–2, Helsinki, Maio 2008.

[36] MØRUP, M., SCHMIDT, M. N., HANSEN, L. K. *Shift Invariant Sparse Coding of Image and Music Data*. Relatório técnico, DTU Informatics, Technical University of Denmark, Lyngby, 2008.

[37] FITZGERALD, D., CRANITCH, M., COYLE, E. "Extended Nonnegative Tensor Factorisation Models for Musical Sound Source Separation", *Computational Intelligence and Neuroscience*, v. 2008, pp. 1–15, Abr. 2008.

[38] MIDI MANUFACTURERS ASSOCIATION INCORPORATED. Disponível em: <http://www.midi.org.>. Acesso em 20/12/2009.

[39] TYGEL, A., BISCAINHO, L. W. P. "Sound Source Separation via Nonnegative Matrix Factor 2-D Deconvolution Using Linearly Sampled Spectrum". In: *Anais do 7o Congresso Nacional da AES Brasil*, pp. 58–65, São Paulo, SP, Maio 2009.

[40] GODSILL, S. J., RAYNER, P. J. W. *Digital Audio Restoration, A Statistical Model Based Approach*. Surrey, Springer, 1998.

[41] GRIFFIN, D. W., LIM, J. S. "Signal estimation from modified shorttime fourier transform", *IEEE Transactions on Acoustics, Speech and Signal Processing*, v. 32, n. 2, pp. 236–243, Abr. 1984.

[42] ZHU, X., BEAUREGARD, G. T., WYSE, L. L. "Real-Time Signal Estimation From Modified Short-Time Fourier Transform Magnitude Spectra", *IEEE Transactions on Audio, Speech and Language Processing*, v. 15, n. 5, pp. 1645–1653, Jul. 2007.

[43] GUNAWAN, D., SEN, D. "Music Source Separation Synthesis using Multiple Input Spectrogram Inversion". In: *Workshop on Multimedia Signal Processing*, Rio de Janeiro, Out. 2009.

[44] PARRY, R. M., ESSA, I. "Incorporating Phase Information for Source Separation via Spectrogram Factorization". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, v. 2, pp. 661–664, Honolulu, Abr. 2007.

[45] ACHAN, K., ROWEIS, S. T., FREY, B. J. "Probabilistic Inference of Speech Signals from Phaseless Spectrograms", *Advances in Neural Information Processing Systems*, v. 16, Jun. 2004.

[46] BECH, S., ZACHAROV, N. *Perceptual Audio Evaluation - Theory, Method and Application*. West Sussex, Wiley, 2006.

[47] ITU-T RECOMMENDATION P.862. "Perceptual Evaluation of Speech Quality (PESQ): Objective Method for End-to-end Speech Quality Assessment of Narrow Band Telephone Networks and Speech Codecs". . International Telecommunication Union, Geneva, Switzerland 2005.

[48] ITU-R RECOMMENDATION BS.1387. "Method for objective measurements of perceived audio quality". . International Telecommunication Union, Geneva, Switzerland 1998.

[49] VINCENT, E., GRIBONVAL, R., FÉVOTTE, C. "Performance measurement in blind audio source separation", *IEEE Transactions on Audio, Speech, and Language Processing*, v. 14, n. 4, pp. 1462–1469, Jul. 2006.

[50] PERSIA, L. E. D., YANAGIDA, M., RUFINER, H. L., et al. "Objective quality evaluation in blind source separation for speech recognition in a real room", *Signal Processing*, v. 87, n. 8, pp. 1951–1965, Ago. 2007.

[51] PERSIA, L. D., MILONE, D., RUFINER, H. L., et al. "Perceptual evaluation of blind source separation for robust speech recognition", *Signal Processing*, v. 88, n. 10, pp. 2578 – 2583, Out. 2008.

[52] ELLIS, D. "Evaluating Speech Separation Systems". In: Divenyi, P. (Ed.), *Speech Separation by Humans and Machines*, pp. 295–304, Norwell, Springer, 2004.

[53] BISCAINHO, L. W. P., ESQUEF, P. A. A., FREELAND, F. P., et al. "An Objective Method for Quality Assessment of Ultra-Wideband Speech Corrupted by Echo". In: *Proceedings of the 2009 IEEE International Workshop on Multimedia Signal Processing*, Rio de Janeiro, Out. 2009. IEEE.

[54] GOTO, M., HASHIGUCHI, H., NISHIMURA, T., et al. "RWC Music Database: Music Genre Database and Musical Instrument Sound Database". In: *Proceedings of the 4th International Conference on Music Information Retrieval*, pp. 229–230, Baltimore, Out. 2003.

[55] KABAL, P. *An Examination and Interpretation of ITU-R BS.1387: Perceptual Evaluation of Audio Quality*. Relatório técnico, Dept. Electrical & Computer Engineering, McGill University, Montreal, 2002. Disponível em: <http://www-mmsp.ece.mcgill.ca/Documents/Downloads/PQevalAudio/>. Acesso em: 26/11/2009.

[56] YANG, B. "A Study of Inverse Short-Time Fourier Transform", *Proc. IEEE ICASSP 2008, Las Vegas, USA*, pp. 3541–3544, Abr. 2008.

[57] FITZGERALD, D., CRANITCH, M., COYLE, E. "Shifted non-negative matrix factorisation for sound source separation". In: *13th Workshop on Statistical Signal Processing*, pp. 1132–1137, Bordeaux, Jul. 2005.

[58] MØRUP, M., SCHMIDT, M. N. *Sparse Non-negative Matrix Factor 2-D Deconvolution*. Relatório técnico, DTU Informatics, Technical University of Denmark, Lyngby, Maio 2006.

[59] SCHMIDT, M. N., MØRUP, M. *Sparse Non-negative Matrix Factor 2-D Deconvolution for Automatic Transcription of Polyphonic Music*. Relatório técnico, DTU Informatics, Technical University of Denmark, Lyngby, Maio 2006.