

CODIFICADORES DE VÍDEO USANDO RIDGELETS E PLANOS DE BITS
GENERALIZADOS

Leonardo Hidd Fonteles

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS
PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE
FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM
CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

Prof. Eduardo Antônio Barros da Silva, Ph.D.

Prof. Abraham Alcaim, Ph.D.

Prof. Marcos Craizer, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

JUNHO DE 2004

FONTELES, LEONARDO HIDD

Codificadores de Vídeo Usando
Ridgelets e Planos de Bits Generalizados
[Rio de Janeiro] 2004

XI, 99 pp., 29,7 cm (COPPE/UFRJ,
M.Sc., Engenharia Elétrica, 2004)

Tese - Universidade Federal do Rio de
Janeiro, COPPE

1.Codificadores de Vídeo 2.Planos de
Bits Generalizados 3.Ridgelets

I.COPPE/UFRJ II.Título (série)

Agradecimentos

ao meu Deus e Pai que me inspirou, me sustentou e me incentivou imensamente. Estou bem certo de que tudo o que tenho, tudo o que sou e o que vier a ser vem dEle.

Gostaria de agradecer, em segundo lugar, a meus pais, Calos Alberto e Elizabeth, meus irmãos, Caius, Emmanuel, Carlos Júnior e Lucas, minha noiva Liliana e meus amigos que, de uma forma ou de outra, formaram uma base bastante forte que possibilitou o desenvolvimento deste trabalho. Família, amo muito vocês. Muito obrigado!

Novamente à Lili pela paciência, compreensão e companheirismo durante todo o decorrer desse trabalho. Meu amor, te amo!

Agradeço, também, a meu orientador, Eduardo, pelo suporte e idéias que fizeram com que esse trabalho fosse viável e trouxesse alguma contribuição para o grupo de Processamento de Sinais, bem como para as pessoas que trabalham na área de vídeo. E, ainda, pela tranqüilidade essencial que me passou através do seu comportamento sempre sereno e otimista. Edu, muito obrigado mesmo!

Ao meu amigo e “co-chefe” Caetano que durante a minha pesquisa de tese me incentivou com idéias e muito trabalho. Caetano, muito obrigado e sucesso!

Ao amigos e colegas do Laboratório de Processamento de Sinais, alunos, professores, secretárias e funcionários, que me proporcionaram um ambiente extremamente tranqüilo, amigável e motivador para o meus estudos de mestrado. Moçada, vocês foram demais, muito obrigado!

A todas estas pessoas, muito obrigado, esperando que possa retribuir à altura tudo o que fizeram por mim.

Salmos 111:10 - *O temor do SENHOR é o princípio da sabedoria; bom entendimento têm todos os que cumprem os Seus mandamentos; o Seu louvor permanece para sempre.*

Provérbios 1:7 - *O temor do SENHOR é o princípio do conhecimento; os loucos desprezam a sabedoria e a instrução.*

Provérbios 15:33 - *O temor do SENHOR é a instrução da sabedoria, e diante da honra vai a humildade.*

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

CODIFICADORES DE VÍDEO USANDO RIDGELETS E PLANOS DE BITS GENERALIZADOS

Leonardo Hidd Fonteles

Junho/2004

Orientador: Eduardo Antônio Barros da Silva

Programa: Engenharia Elétrica

Recentemente foi proposto um novo modelo de decomposição de funções usando planos de bits generalizados onde uma função é mapeada diretamente em um conjunto de índices. Esta decomposição foi também utilizada com sucesso na substituição das operações de decomposição/quantização do codificador de vídeo proposto por Neff e Zakhor, baseado em matching pursuits. Nesta Tese nós investigamos bons dicionários para tais decomposições. Estes dicionários são baseados nas funções ridgelets, propostas recentemente por Candès e Donoho. Mostramos que melhoras na relação taxa–distorção podem ser alcançadas através de combinações de produtos entre ridgelets escaladas e rotacionadas apropriadamente.

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

GENERALIZED BITPLANES-BASED VIDEO ENCODERS USING
RIDGELETS

Leonardo Hidd Fonteles

June/2004

Advisors: Eduardo Antônio Barros da Silva

Department: Electrical Engineering

A novel decomposition of functions using generalized bit-planes has been recently proposed, where a function is mapped into a set of indexes. It also has been successfully used for replacing the decomposition/quantization operations in Neff and Zakhor's matching pursuits video encoder. In this Thesis we investigate good codebooks for such decompositions whose construction is based on ridgelets, proposed by Candès and Donoho. We show that by combining products of properly scaled and rotated ridgelets one can obtain codebooks which provide improved rate-distortion performances.

Sumário

1	Introdução	1
2	O algoritmo Matching Pursuits	6
2.1	O algoritmo Matching Pursuits	10
2.2	O Algoritmo Matching Pursuits Aplicado a Compressão de Vídeo . .	12
2.2.1	Dicionário Utilizado	13
2.2.2	Procura e Codificação dos Melhores Átomos	17
2.2.3	Controle de Taxa	19
3	O algoritmo Matching Pursuits com Planos de Bits Generalizados	20
3.1	Matching Pursuits com Planos de Bits Generalizados	21
3.2	Fatores que afetam o desempenho do algoritmo MPGBP	24
3.3	Codificador de Vídeo baseado no MPGBP	28
3.4	Dicionário proposto	30
4	X-Lets – Dicionários direcionais	32
4.1	Ridgelets	34
4.2	Curvelets	40
4.3	Contourlets	44
4.4	Nossa proposição	46
5	Dicionários para o MPGBP	48
5.1	Codificador Utilizado	52
5.2	Resultados Experimentais	53
6	Conclusão	69

A Seqüências Originais	71
Referências Bibliográficas	93

Lista de Figuras

2.1	Diagrama de blocos do codificador de vídeo baseado em matching pursuits.	14
2.2	Diagrama de blocos do decodificador de vídeo baseado em matching pursuits.	14
3.1	Variação da taxa (R) com α para uma distorção $D = 10^{-2}$ e para dicionários de diferentes valores de $\Theta(\mathcal{C}_i)$, mas com dimensões e cardinalidades iguais a $N(\mathcal{C}_i) = 10$ e $q(\mathcal{C}_i) = 10^4$, $\forall i$, respectivamente. . .	25
3.2	Curva R - D para os diferentes dicionários \mathcal{C}_i especificados na Tabela 3.1.	26
3.3	$\Theta(\mathcal{C})$ em a) pode ser reduzido: b) adicionando um vetor extra no maior buraco do dicionário \mathcal{C} no \mathbb{R}^2 ; c) distribuindo melhor os vetores em \mathbb{R}^2	28
4.1	Representação de um contorno suave através do uso a) das wavelets 2-D separáveis e b) de funções retangulares rotacionadas.	33
4.2	Funções ridgelet: a) original; b) escalonada; c) transladada; d) rotacionada.	36
4.3	Visão geral da decomposição por curvelet.	42
4.4	Decomposição espacial de uma dada subbanda.	43
4.5	Esquema da Pirâmide de Laplace: a) estrutura de análise: as saídas são uma imagem de baixa definição c e uma imagem diferença d composta pela diferença entre o sinal original e uma predição; b) estrutura de síntese	45

4.6	a) partição freqüencial da promovida pelo DFB onde $l = 3$. Assim, existem $2^3 = 8$ bandas as quais são divididas em bandas mais próximas da horizontal (0–3) e mais próximas da vertical (4–7); b) visão multicanal da estrutura em árvore do DBF.	45
4.7	Estrutura do banco de filtro piramidal direcional.	46
5.1	$\Theta(\mathcal{C})$ em a) pode ser reduzido: b) adicionando um vetor extra no maior “buraco” do dicionário \mathcal{C} no \mathbb{R}^2 ; c) distribuindo melhor os vetores em \mathbb{R}^2	49
5.2	Alteração na região de suporte dos átomos do dicionário.	53
5.3	Variação do PSNR médio com o parâmetro α para as seqüências de imagem mother, container, silent e coast para a taxa de 24kbps para o dicionário GR.	56
5.4	Variação do valor de PSNR em função da taxa utilizando o codificador de vídeo MPGBP com os dicionários GR e NZ para a seqüência Mother.	66
5.5	Variação do valor de PSNR em função da taxa utilizando o codificador de vídeo MPGBP com os dicionários GR e NZ para a seqüência Weather.	66
5.6	Variação do valor de PSNR em função da taxa utilizando o codificador de vídeo MPGBP com os dicionários GR e NZ e utilizando a versão mais recente do codificador de vídeo MP com o dicionário NZ para a seqüência Foreman.	67
A.1	Quadros 000, 020, 040, 060, 080 e 100 da seqüência <i>coast-guard</i> original	72
A.2	Quadros 120, 140, 160, 180, 200 e 220 da seqüência <i>coast-guard</i> original	73
A.3	Quadros 240, 260, 280 e 299 da seqüência <i>coast-guard</i> original	74
A.4	Quadros 000, 020, 040, 060, 080 e 100 da seqüência <i>container</i> original	75
A.5	Quadros 120, 140, 160, 180, 200 e 220 da seqüência <i>container</i> original	76
A.6	Quadros 240, 260, 280 e 299 da seqüência <i>container</i> original	77
A.7	Quadros 000, 020, 040, 060, 080 e 100 da seqüência <i>foreman</i> original	78
A.8	Quadros 120, 140, 160, 180, 200 e 220 da seqüência <i>foreman</i> original	79
A.9	Quadros 240, 260, 280 e 299 da seqüência <i>foreman</i> original	80
A.10	Quadros 000, 020, 040, 060, 080 e 100 da seqüência <i>hall-monitor</i> original	81
A.11	Quadros 120, 140, 160, 180, 200 e 220 da seqüência <i>hall-monitor</i> original	82

A.12 Quadros 240, 260, 280 e 299 da seqüência <i>hall-monitor</i> original	83
A.13 Quadros 000, 020, 040, 060, 080 e 100 da seqüência <i>mother-and-daughter</i> original	84
A.14 Quadros 120, 140, 160, 180, 200 e 220 da seqüência <i>mother-and-daughter</i> original	85
A.15 Quadros 240, 260, 280 e 299 da seqüência <i>mother-and-daughter</i> original	86
A.16 Quadros 000, 020, 040, 060, 080 e 100 da seqüência <i>silent-voice</i> original	87
A.17 Quadros 120, 140, 160, 180, 200 e 220 da seqüência <i>silent-voice</i> original	88
A.18 Quadros 240, 260, 280 e 299 da seqüência <i>silent-voice</i> original	89
A.19 Quadros 000, 020, 040, 060, 080 e 100 da seqüência <i>weather</i> original .	90
A.20 Quadros 120, 140, 160, 180, 200 e 220 da seqüência <i>weather</i> original .	91
A.21 Quadros 240, 260, 280 e 299 da seqüência <i>weather</i> original	92

Lista de Tabelas

2.1	Composição dos átomos 1-D de Gabor escolhidos por Neff e Zakhor.	16
3.1	Valores de dimensão ($N(\mathcal{C}_i)$), cardinalidade ($q(\mathcal{C}_i)$), $\Theta(\mathcal{C}_i)$ e α_i ótimo dos dicionários (\mathcal{C}_i) utilizados na construção do gráfico R - D da Figura 3.2.	27
5.1	Valores de PSNR, $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$ obtidos com o dicionário NZ aplicado ao codificador MPGBP.	58
5.2	Comprimentos (n) e escalas (j) dos átomos utilizados na composição do dicionário RO.	59
5.3	Valores de PSNR, $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$ obtidos com o dicionário RO aplicado ao codificador MPGBP.	59
5.4	Comprimentos (n) e escalas (j) dos átomos utilizados na composição do dicionário RS.	60
5.5	PSNR, $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$ obtidos com o dicionário RS aplicado ao codificador MPGBP.	60
5.6	Comprimentos (c) e escalas (j) dos átomos utilizados na composição do dicionário MEYER.	62
5.7	PSNR, $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$ obtidos com o dicionário MEYER aplicado ao codificador MPGBP.	63
5.8	PSNR, $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$ obtidos com os dicionários $Dic1$ e $Dic2$ aplicados ao codificador MPGBP.	64
5.9	PSNR, $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$ obtidos com o dicionário GR aplicado ao codificador MPGBP.	65

Capítulo 1

Introdução

O acesso à informação tornou-se um elemento essencial da atividade econômica mundial e uma das razões da expansão e do dinamismo do setor de telecomunicações. As evoluções tecnológicas sobre o plano de componentes de infraestrutura conduziram, por um lado, ao desenvolvimento extraordinário da Internet e, por outro lado, à introdução cada vez mais importante de conteúdos audiovisuais no contexto de serviços de informação, tornando a informação essencialmente multimídia e permitindo a criação de conteúdos mais e mais ricos.

Entretanto, tais informações audiovisuais despendem vultuosas quantidades de bits, necessitando de tratamentos de compressão que promovam uma economia na banda utilizada para sua transmissão ou uma redução do espaço utilizado em seu armazenamento. Tal fato serve como motivação para o estudo de técnicas que permitam retirar de um conjunto de dados o máximo de informação redundante nele contida, de forma que sua representação seja mais compacta; além disso, deve-se garantir que seja possível, a partir da representação compacta, recuperar toda a estrutura útil presente no conjunto original.

Tais técnicas de compactação, utilizadas em aplicações de compressão de vídeo, tais como TV de alta definição (High Definition Television – HDTV), TV digital, DVD (Digital Versatile Disk), vídeo-conferência, vídeo-fone, entre outras, enfrentam uma série de restrições características de cada aplicação, tais como fluxos de bits em diferentes taxas, diferentes disponibilidades de tempo de processamento (por exemplo, aplicações em tempo real requerem menores níveis de complexidade computacional por parte dos algoritmos implementados), diferentes níveis de per-

das/distorção aceitáveis. Entretanto, algo permanece unânime: um bom compromisso taxa-distorção é sempre desejável.

Visando uma melhoria constante na relação taxa-distorção, inúmeras técnicas de compressão de vídeo têm sido desenvolvidas ao longo das últimas décadas, originando vários métodos e padrões internacionais. Vale ressaltar que um método de compressão de vídeo pode ser dividido, a grosso modo, em quatro etapas: a compensação de movimento (que é responsável pela eliminação da maior parte da redundância), a transformação, a quantização e a codificação da imagem compensada.

Atualmente, os compactadores padrão de áudio/vídeo de maior inserção no mercado são desenvolvidos por um grupo de especialistas em compressão de vídeo conhecido pela sigla MPEG [1], cuja significação é Moving Picture Experts Group. Este grupo envolve centenas de pesquisadores e engenheiros de todo o mundo que desenvolvem vários métodos de compressão de dados audiovisuais para as mais variadas aplicações. Podemos citar, dentre outras aplicações, a compactação de vídeo em DVD (Digital Versatile Disk), HDTV e TV Digital, que utilizam o padrão MPEG-2 [2], e a compactação de vídeo em gráficos interativos e multimídia interativa via internet, que utilizam MPEG-4 [3].

Estes compactadores padrões utilizam, na etapa de transformação, transformadas ortonormais tais como a Transformada Cosseno Discreta [4] e a Transformada Wavelet [5]. Entretanto, paralelamente ao desenvolvimento destes padrões, tem sido desenvolvido em todo o mundo compressores de vídeo baseados em outras técnicas de transformação/representação de imagens cujos resultados são, por vezes, superiores aos destes padrões ([6], [7], [8]).

Dentre estas outras técnicas de transformação existentes, uma em especial chama atenção por sua eficiência face àquelas utilizadas nos compactadores padrões. Stéphane G. Mallat e Zhifeng Zhang criaram em 1993 uma representação para sinais quaisquer conhecida como Matching Pursuits [9]. Esta técnica procura representar um determinado sinal por um conjunto qualquer de bases não-ortonormais. Este conjunto de bases é geralmente denominado de dicionário super-completo, pois o número de elementos deste conjunto é, em geral, bem maior do que a dimensão do sinal. Como o próprio nome sugere, esta representação é realizada por meio de um

algoritmo recursivo que imprime uma busca intensiva pelos melhores casamentos, produtos internos, entre partes do sinal a ser representado e este dicionário super-completo [9]. Logo, um bom desempenho desta representação está diretamente ligado a uma boa escolha do dicionário por ela utilizado.

No contexto de codificação de vídeo, ao qual se reporta esta Tese, Ralph Neff e Avideh Zakhor [6] apresentaram, em 1997, um compressor baseado em Matching Pursuits (MP) que se mostrou bastante eficiente face aos compactadores de vídeo padrão baseados na Transformada de Cosseno Discreta (DCT), de modo que sua participação na norma MPEG-4 foi seriamente cogitada. Esta implementação angariou melhoras significativas de desempenho com relação à DCT. Um dos principais motivos é que na etapa de quantização dos compressores de vídeo baseados na DCT e na Transformada Wavelet (WT) é necessário quantizar todos seus coeficientes, enquanto que na representação MP apenas os melhores valores de produto interno necessitam ser quantizados. Isto se traduz visivelmente na redução dos clássicos efeitos de blocos e no aumento dos valores da relação pico de sinal/ruído (Peak Signal to Noise Ratio – PSNR). Os bons resultados apresentados motivaram novas pesquisas nesta linha de decomposição/representação.

Recentemente, em 2002, uma outra implementação derivada do MP, chamada Matching Pursuits com Planos de Bits Generalizados (*Matching Pursuits with Generalized Bit-Planes* – MPGBP), foi apresentada em [10]. Nela, a decomposição MP é associada à decomposição em planos de bits generalizados, uma adaptação do método de quantização vetorial por Aproximações Sucessivas [11] aplicada ao MP. Com ela, as melhoras são ainda mais significativas, pois as etapas de transformação e quantização são realizadas em um só passo (ver Capítulo 3 para maiores detalhes). Este novo método foi utilizado com sucesso ao substituir as operações de transformação/quantização do codificador de vídeo de Neff e Zakhor baseado em Matching Pursuits, apresentando ganhos de desempenho significativos [10].

Entretanto, em ambas as implementações o dicionário utilizado foi o mesmo [6]. Foi apresentado um conjunto de 400 bases separáveis baseadas nas funções de Gabor. Como comentado anteriormente, a escolha do dicionário utilizado é de grande relevância para o desempenho final da representação tanto por Matching Pursuits quanto por Matching Pursuits com Planos de Bits Generalizados. Logo, tal fato dá

margem ao seguinte questionamento: existem outros dicionários capazes de proporcionar um desempenho ainda melhor ao codificador MPGBP?

Para responder esta pergunta é necessária uma análise criteriosa da relação taxa-distorção. Tal análise, levantada nos capítulos subseqüentes, recai sobre a equação de decomposição do MPGBP e sobre o seu teorema proposto em [10], o qual é a base para a convergência do seu algoritmo de codificação.

Esta análise indica que um dos requisitos da escolha de bons dicionários para o algoritmo MPGBP é existência uma boa semelhança entre as características das funções base constituintes deste dicionário e as características do sinal a ser representado. Este sinal, como visto anteriormente, é o quadro resultante do processo de estimação/compensação de movimento, comumente chamado de *displaced frame difference* (dfd). Os dfds são essencialmente compostos por contornos de diversas larguras e orientados nas mais variadas direções. Logo, o dicionário em questão deve possuir elementos orientados em várias direções e dispostos em várias escalas.

Um levantamento bibliográfico realizado em busca de candidatos para desempenhar o papel de elementos do dicionário na representação MPGBP apontou para um novo conjunto de funções que tem despertado o interesse da comunidade de processamento de imagem/vídeo, as Ridgelets [12]. A Transformada Ridgelet foi primeiramente desenvolvida por Candès [13] e Murata [14] por volta do ano de 1996 como sendo derivações direta da aplicação das funções wavelet às funções *ridge*, desenvolvidas por Logan e Shepp [15] em 1975. As ridgelets foram mais tarde aperfeiçoadas por Donoho [12], no ano de 1998, a partir da aplicação da Transformada Wavelet à Transformada de Radon [16] (para maiores detalhes, ver Capítulo 4).

As ridgelets são conhecidas por seu bom desempenho alcançado na representação de descontinuidades uni-dimensionais, ou seja, descontinuidades dispostas ao longo de linhas, que são componentes básicos das imagens. Elas possuem propriedades de direcionalidade e escalabilidade ideais para a análise de taxa-distorção levantada, pois tais propriedades lhe garantem uma alguma semelhança com as características da dfd. É, também, a base para outras transformadas, tais como as Curvelets [8] e as Contourlets [17], que também serão analisadas ao longo desta dissertação.

A contribuição desta Tese reside, então, na descoberta de novos dicionários

apoiados nesta análise da relação taxa-distorção para o codificador de vídeo baseado no algoritmo MPGBP, os quais são baseados em funções derivadas das ridgelets discreta [18]. No entanto, de modo a melhor adaptar tais funções ao contexto do codificador de vídeo MPGBP e, por conseguinte, atingir melhores resultados de compressão, propomos três importantes modificações para estas funções, quais sejam: introdução de redundância (visando a construção de dicionários supercompletos), eliminação da isotropia e utilização de outras funções, além das wavelets, para a sua confecção (veja o Capítulo 4).

Esta Tese está organizada da seguinte forma: no Capítulo 2 descrevemos o método de representação de sinais MP e sua aplicação mais relevante no contexto de vídeo que é o compactador de Neff e Zakhor [6]; no Capítulo 3 descrevemos o sistema de decomposição de sinais MPGBP de Caetano, da Silva *et al.* [10], assim como o codificador de vídeo baseado neste algoritmo, o qual foi utilizado para a análise de desempenho dos dicionários construídos. Neste contexto, inferimos uma análise de taxa-distorção a partir da análise do teorema proposto e da equação de decomposição; no Capítulo 4 descrevemos toda a teoria sobre as X-Lets (ridgelets, curvelets e contourlets) utilizadas neste trabalho; no Capítulo 5 equacionamos nossa proposta de dicionário e descrevemos especificamente os dicionários construídos juntamente com seus resultados e as discussões cabíveis sobre estes; por fim, as conclusões sobre este trabalho estão expostas no Capítulo 6.

Capítulo 2

O algoritmo Matching Pursuits

Como mencionado anteriormente, o enfoque desta tese está na compressão de informações de vídeo. Entretanto, para que se possa ter um melhor entendimento do processo de compressão de vídeo, é importante se ter uma visão macro do problema.

Uma informação de vídeo é composta por uma seqüência de imagens coloridas que podem ser desmembradas em três componentes diferentes: uma de luminância (Y) e duas de cromaticidade (Cb e Cr). Sua relação com o sistema RGB, o qual divide uma imagem colorida nas componentes R (vermelho), G (verde) e B (azul), é dada como segue [19]:

$$Y = 0,2990 \cdot R + 0,5870 \cdot G + 0,1140 \cdot B$$

$$Cb = -0,1684 \cdot R - 0,3316 \cdot G + 0,5000 \cdot B$$

$$Cr = 0,5000 \cdot R - 0,4187 \cdot G - 0,0813 \cdot B$$

Como resultado, a componente Y carrega apenas a informação de vídeo monocromática. Neste trabalho de tese nos concentramos na compressão desta componente, uma vez que as componentes de cromaticidade, que carregam a informação de cor, podem ser processadas similarmente.

Para facilitar o entendimento do leitor, trataremos aqui a informação de vídeo monocromática como sendo uma seqüência de quadros em escala de cinza de resolução $M \times N$. Tal quadro é matematicamente representado por uma matriz $M \times N$, onde cada elemento da matriz representa um pixel deste quadro e apresenta um valor numérico $n \in \mathbb{N}$ que varia de 0 (preto) a 255 (branco). Isto significa que gasta-se 8 bits de informação por pixel para se representar um quadro.

A visão macro do problema de compressão de vídeo nos mostra que se transmitirmos/armazenarmos todos os $M \cdot N$ pixels de cada quadro da seqüência completa de vídeo gastaríamos uma quantidade enorme de bits. No entanto, podemos reduzir drasticamente esta quantidade devido ao alto volume de redundância contida nas informações de vídeo.

Os estudos em compressão de vídeo desenvolvidos ao longo das últimas décadas nos mostram que existem dois tipos de redundância comumente eliminados. A primeira, e a mais intuitiva, é a repetição de informação que existe entre os diferentes quadros de uma mesma seqüência, sobretudo entre os quadros vizinhos. Este tipo de redundância temporal é reduzida com o uso das técnicas de estimação/compensação de movimento ([20, 21, 22]), que atuam nos quadros vizinhos gerando uma imagem residual. Esta imagem residual, comumente chamada de imagem diferença, contém informações referentes somente às mudanças sofridas na passagem de um quadro para o outro, o que, na maioria dos casos, diminui consideravelmente o número de bits necessários para reproduzir o vídeo original. A outra forma de redundância está embutida neste quadro diferença, o qual, se representado e codificado adequadamente, pode diminuir ainda mais o número de bits gastos para a transmissão/armazenamento do sinal de vídeo. Vale ressaltar que esta tese se propõe a pesquisar codificadores para as imagens diferença; assim sendo, o algoritmo de compensação de movimento usado aqui é o mesmo daquele usado no modelo TMN [23], o qual é uma implementação particular do padrão H.263 [24], como veremos posteriormente. A seguir, falaremos mais especificamente sobre o processo de compressão da imagem diferença.

Em geral, os algoritmos utilizados para a codificação das imagens diferença (comumente chamadas de *dfd*, displaced frame difference) realizam, como primeiro passo, uma representação/decomposição da imagem residual \mathbf{x} em um conjunto de bases vetoriais \mathbf{b} do espaço $\mathbb{R}^{M \times N}$ da seguinte forma:

$$\mathbf{x} = \sum_i c_i \mathbf{b}_i \quad (2.1)$$

onde c_i representa cada um dos coeficientes de projeção da *dfd* no espaço vetorial $\mathbb{R}^{M \times N}$ na direção da base \mathbf{b} .

Dependendo da escolha da base para uma determinada classe de imagens, pode ser mais vantajoso codificar somente estes coeficientes ao invés de codificar

toda a imagem residual.

Note que são os diferentes tratamentos de projeção, ou os diferentes tipos de base vetorial escolhidos que diferenciam as técnicas de decomposição/transformação. Obter uma boa representação/projeção da dfd significa agrupar em poucos coeficientes a maior quantidade de energia (informação do elemento projetado) possível. Assim, para uma mesma qualidade da imagem dfd reconstruída, a quantidade de coeficientes a serem quantizados e codificados será menor. Como consequência, se gasta menos bits para a reconstrução da imagem dfd original.

Dentre as técnicas de transformação existentes, a Transformada Karhunen-Loève (KLT) [19] é a que produz maior concentração de energia com a menor quantidade de coeficientes possível, sendo considerada a *transformada ótima*. Entretanto sua implementação não é viável devido à sua própria concepção, pois a base utilizada para a representação do sinal não é única, fixa, como na maior parte das transformadas. Ou seja, para cada classe de sinal diferente a KLT possui uma base do espaço vetorial própria, a saber, os autovetores da matriz de autocorrelação do sinal (no domínio discreto). Assim, para cada classe de sinal é necessário um novo cálculo da matriz de autocorrelação, sendo que este cálculo numérico possui um custo computacional excessivamente elevado. Além disso, o cálculo desta matriz envolve informações relativas a todo o processo estocástico do qual o sinal a ser transformado é apenas uma das muitas ou infinitas realizações. Como custo adicional, é necessária a transmissão de cada uma das bases para o decodificador, exatamente pelo fato da base não ser determinística.

Todavia, uma boa aproximação da KLT é atingida pela Transformada Cosseno Discreto (DCT) [4] quando aplicada a blocos de imagens naturais de tamanho até 8×8 . Além desta proximidade com a KLT, a DCT possui algoritmos rápidos para a sua implementação, de maneira que ela é a transformada mais difundida e mais utilizada em compressão de vídeo na atualidade.

A base \mathbf{b} do espaço $\mathbb{R}^{M \times N}$ é tratada pela DCT como sendo um conjunto fixo de vetores (ao contrário da KLT), os quais são cossenóides. Um sinal unidimensional $x(n)$, por exemplo, pode ser representado pela DCT através do seguinte

somatório [25]:

$$x(n) = \sum_{k=0}^{N-1} \alpha(k) C(k) \cos \left[\frac{\pi(n + \frac{1}{2})k}{N} \right], \quad \text{para } 0 \leq n \leq N-1 \quad (2.2)$$

onde

$$\alpha(k) = \begin{cases} \sqrt{\frac{1}{N}}, & k = 0 \\ \sqrt{\frac{2}{N}}, & 1 \leq k \leq N-1 \end{cases} \quad (2.3)$$

e $C(k)$ são os coeficientes dados por

$$C(k) = \alpha(k) \sum_{n=0}^{N-1} x(n) \cos \left[\frac{\pi(n + \frac{1}{2})k}{N} \right], \quad \text{para } 0 \leq k \leq N-1 \quad (2.4)$$

Não obstante ao fato de a DCT ser uma boa aproximação da KLT, ela, assim como a Transformada de Fourier, carece de uma maior adaptabilidade aos sinais a serem decompostos. Suas estruturas possuem janelas de tamanhos fixos (uni-escalares), de modo que sinais muito maiores ou muito menores do que suas estruturas não são bem representados em suas bases. Assim, informações sobre a localização espacial e os aspectos freqüenciais de tais sinais não são retratados com fidelidade. Logo, para se analisar componentes de tamanhos variados é necessário o uso de uma representação onde as suas bases possuam diferentes escalas no domínio espacial e no freqüencial.

Nesse contexto, a Transformada Wavelet [5] surgiu na tentativa de solucionar este problema, introduzindo como vetores base elementos formados de dilatações e translações de uma única função $\bar{\psi}$, denominada wavelet mãe, como segue [25]:

$$x(t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} c_{m,n} 2^{-\frac{m}{2}} \bar{\psi}(2^{-m}t - n) \quad (2.5)$$

Tal adaptabilidade levou a WT a ser largamente utilizada em aplicações de processamento de sinais. Entretanto, a construção da família de funções wavelet é feita associando-se o parâmetro freqüência ao parâmetro escala através de uma relação inversa $f \propto \frac{1}{s}$, de modo que, ao mesmo tempo em que uma base de alta freqüência é bem localizada no espaço, é mal localizada na freqüência, e vice-versa. Assim, certas faixas de freqüência, especialmente as mais altas, não são devidamente representadas.

Como saída para estas limitações, vários pesquisadores desenvolveram métodos de decomposição adaptativos os quais não fazem mais uso de bases (também conhecidas como o menor *dicionário completo* – conjunto de m vetores n -dimensionais

que representam qualquer outro vetor do espaço n -dimensional, onde $m > n$). Dentre os vários métodos desenvolvidos [26, 27, 28, 29], aquele que obteve maior sucesso foi o chamado Matching Pursuits [9].

O método Matching Pursuits (MP) obteve grande êxito ao ser aplicado em várias áreas de processamento de sinais. Diversos trabalhos têm sido publicados desde a sua divulgação no ano de 1993, demonstrando a sua eficiência e o grande interesse da comunidade acadêmica de processamento de sinais em seu uso [30, 31, 32, 33].

Como o compressor de vídeo usado nesta tese de mestrado se baseia em parte no algoritmo MP, na Seção 2.1 esta decomposição é descrita mais detalhadamente. Já na Seção 2.2, descrevemos a implementação mais relevante desta decomposição/representação de sinais no contexto de compressão de vídeo, implementação esta desenvolvida por Neff e Zakhor [6].

2.1 O algoritmo Matching Pursuits

Mallat e Zhang propuseram o Matching Pursuits como uma nova forma de decomposição de sinais onde o conceito de base do espaço vetorial não é mais utilizado. Ao invés disso, esta decomposição representa uma função f do espaço de Hilbert (onde $\|f\|^2 = \int_{-\infty}^{\infty} |f(t)|^2 dt < +\infty$) projetando-a em um conjunto de funções redundantes cujo número de elementos é geralmente muito superior à dimensão da função. Este conjunto, representado por $\mathcal{D} = \{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_M\}$, é denominado de *dicionário supercompleto* e cada um de seus elementos \mathbf{g}_i é chamado de átomo, onde $\|\mathbf{g}_i\| = 1, \forall i$.

Com o número de átomos muito superior ao da dimensão de f , é possível povoar o domínio espacial e o freqüencial de uma maneira mais completa, de modo a atenuar as deficiências apresentadas pelas transformadas acima destacadas. Com uma distribuição eficiente destes átomos tanto no domínio espacial quanto no freqüencial, garante-se uma melhor representação para uma maior variedade de sinais.

Assim, para a versão discreta desta representação, a qual assume que o sinal é real e possui N amostras, foi proposta a utilização do dicionário de Gabor. Este dicionário possui átomos oriundos de expansões/contrações, translações e modulações

de uma única função Gaussiana, a qual é dada, no domínio contínuo, por:

$$g(t) = \sqrt[4]{2}e^{-\pi t^2} \quad (2.6)$$

de forma que sua amostragem e conseqüente periodização em N pontos é dada por:

$$g_s(n) = \frac{K_s}{\sqrt{s}} \sum_{p=-\infty}^{\infty} g\left(\frac{n-pN}{s}\right) \quad (2.7)$$

onde K_s normaliza g_s e os inteiros s e p são limitados por $s \in]1, N[$ e $0 \leq p < N$.

Denotando $\gamma = (s, p, \frac{2k\pi}{N})$, o dicionário de Gabor discreto e real é composto por todos os átomos dados por:

$$g_{(\gamma, \phi)}(n) = K_{(\gamma, \phi)} g_s(n-p) \cos\left(\frac{2k\pi}{N}n + \phi\right) \quad (2.8)$$

onde $K_{(\gamma, \phi)}$ normaliza $g_{(\gamma, \phi)}$, $\phi \in [0, 2\pi[$ e $0 \leq k < N$.

Entretanto, para este novo tipo de “base” já não faz mais sentido representarmos sinais utilizando o conceito de base do espaço vetorial, onde se aplica a decomposição por meio do somatório da Equação (2.1). Assim, além do uso de um outro conceito de base, foi proposta uma outra maneira de se decompor f .

A expansão linear de f sobre os átomos de \mathcal{D} que o MP propõe-se a realizar é feita através de sucessivas aproximações de f com projeções ortogonais em tais átomos. Dado um vetor g_{γ_0} de \mathcal{D} , o vetor f pode ser decomposto como segue:

$$f = \langle g_{\gamma_0}, f \rangle g_{\gamma_0} + Rf \quad (2.9)$$

onde Rf é o vetor residual obtido após a aproximação de f na direção de g_{γ_0} . Como g_{γ_0} é ortogonal a Rf , temos:

$$\|f\|^2 = |\langle g_{\gamma_0}, f \rangle|^2 + \|Rf\|^2 \quad (2.10)$$

Vemos então que para minimizar $\|Rf\|^2$ é necessário que o produto interno $p_0 = \langle g_{\gamma_0}, f \rangle$ seja o maior possível. Isto é possibilitado por uma busca intensiva (pursuit) do vetor $g_i \in \mathcal{D}$ que garanta o melhor casamento (match) com f .

Na próxima realização desta decomposição, o resíduo Rf é projetado na estrutura de \mathcal{D} que se adapte melhor (possua maior valor de produto interno) a ele, acarretando na geração de outro resíduo $R^{(2)}f$:

$$Rf = \langle g_{\gamma_1}, Rf \rangle g_{\gamma_1} + R^{(2)}f \quad (2.11)$$

Assim, após P realizações temos que f é dada por:

$$f = \sum_{n=0}^{P-1} \langle g_{\gamma_n}, R^{(n)} f \rangle g_{\gamma_n} + R^{(P)} f \quad (2.12)$$

e sua aproximação dada por:

$$f^{(P)} = \sum_{n=0}^{P-1} p_n g_{\gamma_n} \quad (2.13)$$

onde $p_n = \langle g_{\gamma_n}, R^{(n)} f \rangle$.

Este processo iterativo ocorrerá até que um determinado número de passos P , ou um limite de energia do resíduo $R^{(n)} f$ seja atingido.

Note que ao final deste procedimento, apenas os átomos melhor adaptados à f são utilizados na sua decomposição, ao contrário das Transformadas de Fourier, Cosseno Discreto e Wavelets, para as quais importa que todas as funções base sejam utilizadas, implicando na quantização e codificação de todos os coeficientes c_i .

Para maiores detalhes, consultar [9].

2.2 O Algoritmo Matching Pursuits Aplicado a Compressão de Vídeo

Nesta seção é descrita com algum detalhe uma das implementações do algoritmo MP no contexto de compressão de vídeo, mais precisamente na codificação da imagem residual resultante do processo de estimação/compensação de movimento. Tal detalhamento é pertinente uma vez que o codificador proposto neste trabalho utiliza alguns dos procedimentos e considerações adotados por seus idealizadores, Neff e Zakhor, em [6]. Como ponto de partida, é importante contextualizar seu surgimento e seus propósitos iniciais.

Nos anos 90 alguns grupos de pesquisa tais como o MPEG [1] e o ITU-T [34] empregaram um grande esforço técnico com o objetivo de padronizar os procedimentos utilizados na compressão de informações audiovisuais limitadas a baixas taxas de vídeo. Como consequência, surgiram os padrões MPEG-4 [3] e H.263 [24] que, embora projetados com uma certa flexibilidade, foram inicialmente designados principalmente à comunicação de vídeo em tempo-real e aos serviços de multimídia e acesso remoto de banco de dados, com taxas de operação situadas entre 10kbps e 24kbps.

A composição de ambos os padrões é bastante semelhante, sendo baseada em uma estrutura híbrida de estimação/compensação de movimento aliada à DCT. Entretanto, o uso da DCT na codificação da dfd produz efeitos subjetivos indesejados, os quais são consequência direta da blocagem da imagem diferença em pequenos quadrados de tamanho 8×8 (lembrar da discussão levantada anteriormente sobre a KLT e a sua aproximação através da DCT). Esta distorção é ainda mais notória quando a disponibilidade de bits por segundo é muito pequena, pois o número de coeficientes da DCT a ser quantizado deve ser muito restrito, além do fato da quantização em si ser limitada a uma baixa resolução.

Logo, com o intuito de mostrar que melhores resultados na compressão de vídeo poderiam ser alcançados com a substituição da popular DCT, Neff e Zakhor utilizaram a mesma estrutura de compensação de movimento do padrão H.263, retirando apenas a DCT e inserindo no seu lugar o algoritmo matching pursuits. Assim, o modelo de predição usado pelo codificador de Neff e Zakhor para a eliminação da redundância temporal é exatamente o mesmo usado pelo TMN [23], o qual é uma implementação específica do H.263 [24]. Tal sistema atua em imagens QCIF, cuja componente de luminância é um quadro de resolução 176×144 e as componentes de crominância são subamostradas de 2, possuindo um tamanho de 88×72 .

Desta forma, os diagramas de blocos do codificador e do decodificador do sistema híbrido estimação/compensação de movimento com matching pursuits são ilustrados nas Figuras 2.1 e 2.2, respectivamente.

Veremos a seguir algumas suposições e adaptações propostas por Neff e Zakhor na sua implementação do MP para vídeo.

2.2.1 Dicionário Utilizado

Assim como a implementação original do matching pursuits para sinais discretos utilizou o dicionário de Gabor (ver [9]), o codificador de vídeo MP também utilizou tal dicionário supercompleto. Entretanto, por se tratar de uma aplicação 2-D (codificação da imagem diferença), houve a necessidade de se adaptar o dicionário 1-D de Gabor ao domínio bi-dimensional.

Fazendo uma analogia à Equação (2.8), Neff e Zakhor definiram o átomo 1-D

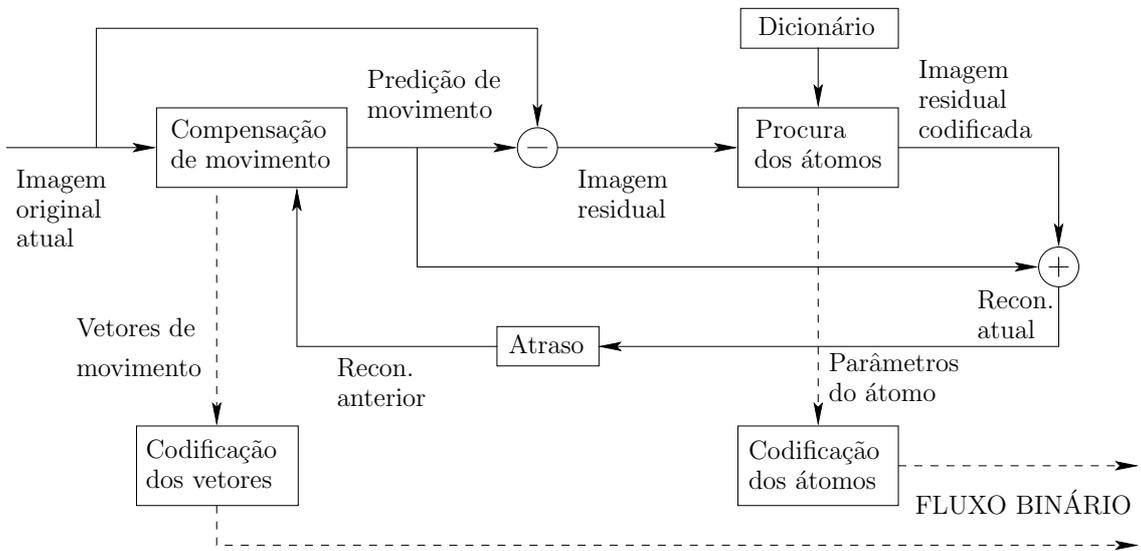


Figura 2.1: Diagrama de blocos do codificador de vídeo baseado em matching pursuits.

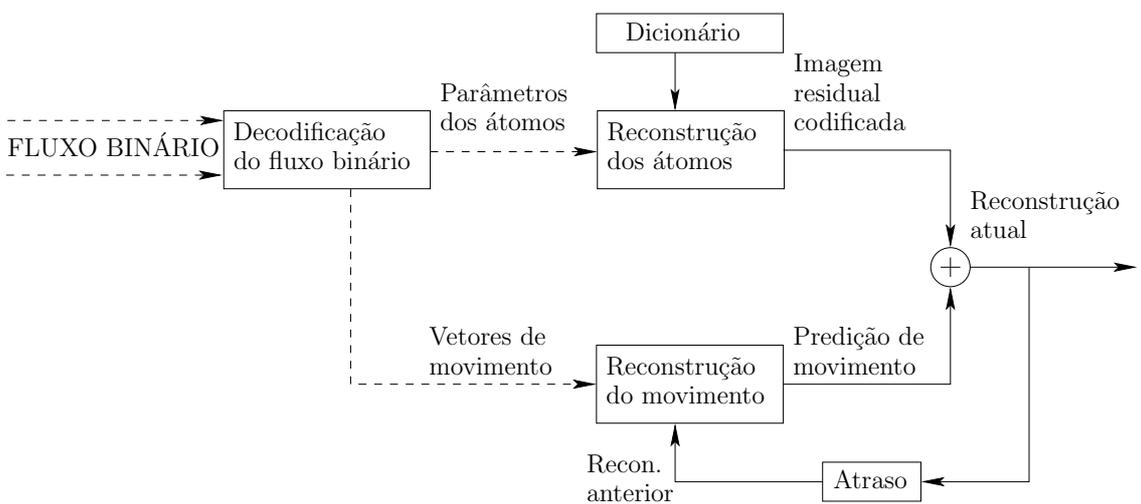


Figura 2.2: Diagrama de blocos do decodificador de vídeo baseado em matching pursuits.

discreto de Gabor como:

$$g_{\vec{\alpha}}(i) = K_{\vec{\alpha}} \cdot g\left(\frac{i - \frac{N-1}{2}}{s}\right) \cdot \cos\left(\frac{2\pi\xi(i - \frac{N-1}{2})}{N} + \phi\right) \quad (2.14)$$

onde $i \in 0, 1, \dots, N - 1$ e a constante $K_{\vec{\alpha}}$ é escolhida de modo a normalizar $g_{\vec{\alpha}}$.

Desta feita, $\vec{\alpha}$ é o vetor de parâmetros $\vec{\alpha} = (s, \xi, \phi)$ que contem as informações de escala, frequência de modulação e deslocamento de fase, respectivamente, sendo assim análogo ao parâmetro γ definido na Seção 2.1. Então, definindo \mathcal{B} como sendo o conjunto de todos os vetores $\vec{\alpha}$, o dicionário 2-D discreto de Gabor é o conjunto de funções separáveis definidas como segue:

$$\begin{aligned} G_{\vec{\alpha}, \vec{\beta}}(i, j) &= g_{\vec{\alpha}}(i)g_{\vec{\beta}}(j) & i, j \in 0, 1, \dots, N - 1 \\ & & \vec{\alpha}, \vec{\beta} \in \mathcal{B} \end{aligned} \quad (2.15)$$

No entanto, para que este dicionário seja funcional, é necessário que apenas um número limitado de vetores $\vec{\alpha}$ seja escolhido. É importante também que o comprimento N dos átomos seja pequeno, evitando um alto custo computacional no cálculo dos produtos internos.

O procedimento adotado por Neff e Zakhor para a escolha de um pequeno grupo de átomos consistiu de um treinamento com algumas seqüências de vídeo, no qual foi utilizado um grande conjunto de vetores de parâmetros. Os átomos mais freqüentemente selecionados no processo de casamento com as imagens residuais de treinamento foram retidos para a composição de um dicionário reduzido. Dessa forma, o tamanho dos átomos, bem como vetores $\vec{\alpha}$ escolhidos, estão descritos na tabela 2.1. Note que com esses 20 átomos 1-D são geradas 400 bases 2-D separáveis, uma vez que na Equação (2.15) tanto $\vec{\alpha}$ como $\vec{\beta}$ assumem estas 20 combinações de parâmetros.

Observe que os tamanhos destes vetores foram escolhidos propositalmente com valores ímpares. Aliado a isto, note que na formação dos vetores $g_{\vec{\alpha}}$ na Equação (2.14), o parâmetro de translação p da Equação (2.8) é ajustado para que todas as funções passem pelo ponto central. Assim, todos os átomos do dicionário 2-D estão centralizados no pixel $(\frac{N_{\vec{\alpha}}-1}{2}, \frac{N_{\vec{\beta}}-1}{2})$, o qual é o ponto de referência de encaixe com os pixels da dfd. E, como veremos a seguir, uma vez que no processo de procura dos átomos para a decomposição da dfd os átomos do dicionário varrem

Tabela 2.1: Composição dos átomos 1-D de Gabor escolhidos por Neff e Zakhor.

Escala (s)	Frequência (ξ)	Fase (ϕ)	Tamanho ($N_{\vec{\alpha}}$)
1	0	0	1
3	0	0	5
5	0	0	9
7	0	0	11
9	0	0	15
12	0	0	21
14	0	0	23
17	0	0	29
20	0	0	35
1.4	1	$\pi/2$	3
5	1	$\pi/2$	9
12	1	$\pi/2$	21
16	1	$\pi/2$	27
20	1	$\pi/2$	35
4	2	0	7
4	3	0	7
8	3	0	13
4	4	0	5
4	2	$\pi/4$	7
4	4	$\pi/4$	7

toda a imagem diferença se deslocando pixel a pixel nesta imagem, são evitados gastos desnecessários com a geração de versões transladadas de um mesmo átomo, já que este processo de varredura simula tal translação. Assim, como todas os 400 vetores assumem todas as 25344 posições (pixels) de uma dfd de tamanho 176×144 , temos, na realidade, um total de mais de 10 milhões de vetores.

O uso de um dicionário com este tamanho permite que o matching pursuits codifique a dfd utilizando, em geral, menos coeficientes que a DCT, a qual usa 64 bases para representar cada um dos 396 blocos de tamanho 8×8 . Entretanto, este aumento notório do número de vetores representa um custo adicional de representação. Mas, devido a alta eficiência deste dicionário, os resultados do codificador MP ainda são melhores que os do modelo H.263 [6].

2.2.2 Procura e Codificação dos Melhores Átomos

Uma das vantagens do matching pursuits sobre a DCT na codificação da imagem diferença é que o efeito de blocagem introduzido pela DCT é reduzido a níveis imperceptíveis. Isto se deve ao fato do matching pursuits calcular os produtos internos posicionando os átomos em todas os pixels da imagem. Como a subtração de $p_n g_{\gamma_n}$ é feita com átomos (blocos) de dimensões e posições “aleatórias” em todos os passos desta decomposição, a blocagem em quadros iguais e adjacentes promovida pela DCT é eliminada, reduzindo sensivelmente efeitos subjetivos indesejados (*efeitos de bloco*), mesmo a baixas taxas de bits [6]. No entanto, a implementação direta do matching pursuits no contexto de vídeo resultaria na realização de um procedimento de elevado custo computacional, o qual envolveria a busca dos melhores átomos varrendo a dfd pixel a pixel e calculando todos os produtos internos com cada um dos vetores do dicionário.

Assim, Neff e Zakhor propuseram um esquema de busca mais eficiente, o qual é baseado na seguinte suposição: as imagens dfds possuem um conteúdo de informação esparso (pouco denso) e apresentado em forma de bolsões de energia localizados em regiões onde o processo de estimação/compensação de movimento não foi bem sucedido. Partindo desta suposição, ao invés de realizar os casamentos em toda a imagem residual de uma só vez, pequenas áreas ao redor dos bolsões de energia seriam priorizadas. Um simples processo de “busca de energia” seria pré-

realizado, dividindo a dfd em pequenos blocos e calculando a energia de cada um deles através da soma dos quadrados de cada um dos seus pixels. Ao redor do bloco de maior energia, se estabeleceria uma área de dimensão $S \times S$ onde seria realizada em busca intensiva, pixel a pixel, pelos melhores átomos.

Numericamente falando, o processo de busca de energia divide a imagem em blocos 12×12 . Após a escolha do bloco de maior energia, a janela $S \times S$ estabelecida ao redor deste bloco tem dimensão 16×16 . Nesta janela são então realizados os exaustivos produtos internos com os vetores do dicionário cujas dimensões máximas e mínimas são, respectivamente 35×35 e 1×1 . Vale lembrar que o ponto de referência no encaixe entre os pixels da janela $S \times S$ e os vetores do dicionário é seu ponto central $\left(\frac{N_{\vec{\alpha}}-1}{2}, \frac{N_{\vec{\beta}}-1}{2}\right)$.

Observe que além de tal estratégia reduzir o tempo de procura de átomos, ela também prioriza as áreas que possuem uma maior quantidade de energia/informação, gastando uma maior quantidade de bits na sua codificação. Assim, para baixas taxas de bits, informações de baixa energia, tais como detalhes e ruídos, não são codificados.

Dada a escolha do melhor elemento do dicionário, um conjunto de 5 parâmetros é gerado: os índices dos vetores $\vec{\alpha}$ e $\vec{\beta}$; a localização (x, y) do encaixe (posição do ponto central do elemento do dicionário na janela $S \times S$); valor do produto interno p_n . Estes 5 parâmetros é que definem o átomo, estrutura que de fato é codificada na imagem.

Após a decomposição de cada imagem residual, os parâmetros dos átomos selecionados são passados para o codificador [35]. Os parâmetros de localização são codificados utilizando códigos de Huffman adaptativos derivados dos valores de localização dos 10 últimos quadros. Já os outros 3 parâmetros são codificados com códigos de Huffman de comprimento fixo. No entanto, a codificação do valor do produto interno é precedida por um processo de quantização. Em [6], p_n é quantizado por um simples quantizador linear com passo de quantização fixo. Entretanto, alguns trabalhos posteriores foram realizados com o intuito de melhorar o desempenho do codificador através do uso estratégias de quantização mais sofisticadas [36, 37, 38].

2.2.3 Controle de Taxa

A despeito das estratégias de quantização e codificação empregadas na compressão da imagem residual, o número total de bits disponível para todo o processo de compressão é limitado. Assim, se faz necessário um controle sobre o gasto dos bits disponíveis, de modo a permitir uma distribuição o mais eficiente possível destes bits entre todos os quadros da seqüência. No contexto do codificador MP, este controle se reflete diretamente no número de átomos a serem codificados.

Em [6], Neff e Zakhor realizaram um controle simples de taxa, cuja finalidade era permitir um bom nível de comparação com a taxa real utilizada pelo modelo H.263. Este controle estabelece um número limite de bits disponível para a codificação de cada quadro da seqüência da seguinte maneira: primeiramente, é codificado o cabeçalho (que contem algumas informações sobre o quadro) e os vetores de movimento (obtidos com o processo de estimação/compensação de movimento) e o número de bits gastos na sua codificação é retirado do limite pré-estabelecido, oferecendo ao processo de codificação da dfd um número máximo de $B_{\text{resíduo}}$ bits; em seguida, o codificador calcula do número médio de bits gastos na codificação dos átomos do quadro anterior, $B_{\text{átomo}}$; por último, obtem-se do número de átomos a serem codificados no quadro atual através a seguinte equação:

$$N_{\text{atual}} = \frac{B_{\text{resíduo}}}{B_{\text{átomo}}} \quad (2.16)$$

Logo, o matching pursuits realiza a decomposição de uma dada imagem residual até que o número N_{atual} de átomos seja atingido. Este procedimento despense cerca de $B_{\text{resíduo}}$ bits, atingindo aproximadamente o limite de bits a serem gastos por quadro. Na prática, este número dista até 1% da taxa de bits utilizada pelo modelo H.263 [6].

Para maiores detalhes sobre este codificador de vídeo, consultar [6].

Capítulo 3

O algoritmo Matching Pursuits com Planos de Bits Generalizados

Neste capítulo, descrevemos uma nova técnica de decomposição de sinais, na qual se baseia o codificador de imagens residuais utilizado nesta tese. Esta técnica, recém desenvolvida por Caetano e Eduardo [10, 39, 40], realiza uma decomposição em planos de bit de um sinal \mathbf{x} qualquer, a qual o representa através de sucessivos passos de aproximação por um conjunto de planos de bits generalizados.

Seu surgimento foi motivado em parte pelo sucesso do Matching Pursuits, de forma que algumas das considerações feitas por Mallat e Zhang foram adotadas por Caetano e Eduardo. Tal decomposição foi, então, denominada de Matching Pursuits with Generalized Bit-Planes (MPGBP).

Uma de suas implementações se deu no contexto de codificação de vídeo [10], atuando como codificador da imagem residual. Os resultados apresentados superaram aqueles obtidos por Neff e Zakhor em [6], mostrando o MPGBP como um método de decomposição de sinais bastante promissor. Na Seção 3.3 analisaremos mais detalhadamente tal implementação.

A seguir, na Seção 3.1 veremos a teoria por trás desta decomposição e nas Seções 3.2 e 3.4 faremos a sua relação com esta tese.

3.1 Matching Pursuits com Planos de Bits Generalizados

Como visto anteriormente, a expansão de um sinal \mathbf{x} N -dimensional em um dicionário $\mathcal{D} = \{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_M\}$ completo pode ser expressa como segue:

$$\mathbf{x} = \sum_{n=1}^M p_n \mathbf{g}_n \quad (3.1)$$

Contudo, na prática utiliza-se somente $Q < M$ átomos, aproximando \mathbf{x} como segue:

$$\mathbf{x} \approx \mathbf{x}^{(Q)} = \sum_{n=1}^Q p_n \mathbf{g}_{\gamma_n} \quad (3.2)$$

Além disto, se assumirmos, sem perda de generalidade, que $\|\mathbf{g}_i\| = 1$ e $\|\mathbf{x}\| \leq 1$, $\forall i$, esta expansão pode ser dada de tal forma que os coeficientes p_n possuam $|p_n| \leq 1$. Com isto, podemos representar estes coeficientes de forma binária, como segue: $p_n = s_n \sum_{j=1}^{\infty} 2^{-j} b_{j,n}$, onde $s_n \in \{-1, 1\}$ é o sinal de p_n e $b_{j,n}$ assume os valores binários $\{0, 1\}$.

Aplicando esta representação binária à Equação (3.2), temos que:

$$\begin{aligned} \mathbf{x} \approx \mathbf{x}^{(Q)} &= \sum_{n=1}^Q s_n \sum_{j=1}^{\infty} 2^{-j} b_{j,n} \mathbf{g}_{\gamma_n} = \sum_{j=1}^{\infty} 2^{-j} \sum_{n=1}^Q b_{j,n} s_n \mathbf{g}_{\gamma_n} \\ &= \sum_{j=1}^{\infty} 2^{-j} \sum_{n=1}^Q b_{j,n} \bar{\mathbf{g}}_{\gamma_n} \end{aligned} \quad (3.3)$$

onde $\bar{\mathbf{g}}_{\gamma_n} \in \bar{\mathcal{D}} = \{\pm \mathbf{g}_1, \pm \mathbf{g}_2, \dots, \pm \mathbf{g}_M\}$, uma vez que $\bar{\mathbf{g}}_{\gamma_n} = s_n \mathbf{g}_{\gamma_n}$ e $s_n \in \{-1, 1\}$.

Então, definindo o índice $i_{j,l}$ de modo que:

$$b_{j,i_{j,l}} = \begin{cases} 1, & l \in \{1, 2, \dots, L_j\} \quad \text{com } L_j \leq Q \\ 0, & \text{demais valores de } l \end{cases} \quad (3.4)$$

podemos reescrever a Equação (3.3) como:

$$\mathbf{x} \approx \mathbf{x}^{(L_j)} = \sum_{j=1}^{\infty} 2^{-j} \sum_{l=1}^{L_j} \bar{\mathbf{g}}_{\gamma_{i_{j,l}}} \quad (3.5)$$

Observe que no MP o sinal \mathbf{x} é representado por um conjunto de vetores unitários $\mathbf{g}_{\gamma_1}, \mathbf{g}_{\gamma_2}, \dots$ juntamente com uma seqüência de escalares p_1, p_2, \dots , enquanto que a decomposição da Equação (3.5) é representado apenas por um conjunto de vetores unitários $\bar{\mathbf{g}}_{\gamma_1}, \bar{\mathbf{g}}_{\gamma_2}, \dots$

De fato, no começo dos anos 90, paralelamente às pesquisas de Mallat e Zhang em métodos de decomposição por aproximação sucessiva, Craizer, Eduardo *et al.* concebiam um outro algoritmo de aproximação sucessiva, o qual apresenta uma decomposição semelhante a da Equação (3.5), sendo, no entanto, mais generalizada. Desenvolvido como sendo uma evolução/generalização de dois bem-sucedidos algoritmos de decomposição de imagens, o EZW [41] e o SPIHT [42], que realizam decomposições escalares, este algoritmo realiza uma eficiente decomposição vetorial através de sucessivas aproximações da imagem alvo, sendo assim chamado de Successive Approximation Vector Quantisation (SAVQ) [11].

A SAVQ decompõe o sinal \mathbf{x} através do seguinte somatório:

$$\mathbf{x} \approx \mathbf{x}^{(L_j)} = \sum_{j=1}^{\infty} \alpha^j \sum_{l=1}^{L_j} \bar{\mathbf{g}}_{i_{j,l}} \quad (3.6)$$

Note, então, que SAVQ é um algoritmo de aproximações sucessivas capaz de realizar de maneira ainda mais generalizada a decomposição em planos de bits apresentada na Equação (3.5). Note que esta equação é um caso particular da Equação (3.6) para $\alpha = 0,5$, onde α é chamado de *fator de aproximação de escala* e seu valor está compreendido no intervalo $0 < \alpha < 1$.

No entanto, em [10, 40] algumas modificações foram propostas com relação ao SAVQ [11]. Estas mudanças abrangem dois pontos principais: regras de convergência e indexação.

Em [11] a condição de convergência – condição na qual qualquer sinal \mathbf{x} é aproximado com uma precisão arbitrária com a adição de um número suficiente de termos dos somatórios – é tal que $\Theta(\bar{\mathcal{D}}) \leq \frac{\pi}{3}$ para $0 < \alpha < 1$, onde $\Theta(\bar{\mathcal{D}})$ é o maior ângulo entre um sinal qualquer $\mathbf{x} \in \mathbb{R}^N$ e o átomo de $\bar{\mathcal{D}}$ mais próximo. Entretanto, mesmo para sinais de dimensão medianas, tais como $N > 64$, os dicionários que possuem $\Theta(\bar{\mathcal{D}}) \leq \frac{\pi}{3}$ apresentam grande cardinalidade $q(\bar{\mathcal{D}})$, levando a uma decomposição pouco eficiente do ponto de vista da relação taxa-distorção, pois um bom número dos bits disponíveis seria deslocado somente para a codificação os índices $i_{j,l}$.

Já em [10, 40], foram propostas como condições de convergência $\Theta(\bar{\mathcal{D}}) \leq \frac{\pi}{2}$ e $0 < \alpha < 1$, condições estas que podem ser satisfeitas por qualquer dicionário completo.

No que concerne à indexação, considere L_j como sendo o número de valores m tal que $k_m = j$. Assim, pode-se substituir o índice $i_{j,l}$ por r_m para $l = 1, 2, 3, \dots, L_j$. Logo, dado um dicionário $\mathcal{C} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_q\}$, $\|\mathbf{v}_i\| = 1, \forall i$, e um sinal \mathbf{x} de módulo menor ou igual a 1, a decomposição proposta em [10, 40], após realizados Q passos, é dada por:

$$\mathbf{x} \approx \mathbf{x}^{(Q)} = \sum_{m=1}^Q \alpha^{k_m} \mathbf{v}_{r_m} \quad (3.7)$$

O algoritmo proposto para a implementação de tal decomposição realiza uma busca voraz pelos melhores átomos \mathbf{v}_{r_m} , adicionando um por vez, até que o critério taxa-distorção seja atingido. Este algoritmo é descrito como segue:

Algoritmo MPGBP

- ① Inicialmente $\mathbf{w} = \mathbf{x}$, $m = 1$;
- ② Repetir até um critério de parada seja encontrado:
 - (a) Escolher $r_m \in \{1, \dots, M\}$ tal que
$$\mathbf{w} \cdot \mathbf{v}_{r_m} = \max_{1 \leq j \leq M} \{\mathbf{w} \cdot \mathbf{v}_j\};$$
 - (b) Escolher $k_m = \left\lceil \frac{\ln(\mathbf{w} \cdot \mathbf{v}_{r_m})}{\ln(\alpha)} \right\rceil$;
onde $\lceil \cdot \rceil$ é o menor inteiro maior ou igual ao argumento;
 - (c) Substituir \mathbf{w} por $\mathbf{w} - \alpha^{k_m} \mathbf{v}_{r_m}$;
 - (d) Incrementar m ;
- ③ Parar.

Note que esta representação é bastante similar à MP. A principal diferença está no tratamento que é dado ao valor do produto interno p_n resultante dos casamentos entre os átomos e o sinal \mathbf{x} . Comparando as Equações (3.2) e (3.7), observa-se que nesta decomposição em planos de bits generalizados o valor do casamento entre os vetores do dicionário \mathcal{C} e \mathbf{x} é diretamente mapeado em um conjunto de índices k_m diretamente codificáveis, enquanto que no MP a codificação de p_n envolve uma

pré-quantização do mesmo, pois p_n é um número real que, em geral, possui várias casas decimais de precisão. Assim, devido a sua similaridade com o MP e sua decomposição em planos de bits generalizados, esta decomposição/algoritmo foi chamada de Matching Pursuits com Planos de Bits Generalizados (*Matching Pursuits with Generalized Bit-Planes* – MPGBP).

As condições de convergência do algoritmo acima foram formalizadas no seguinte teorema [40]:

Teorema 1: Seja $\mathbf{x} \in \mathbb{R}^N$, $\|\mathbf{x}\| \leq 1$, um sinal aproximado pelo algoritmo MPGBP usando um dicionário \mathcal{C} com Q passos, gerando $\mathbf{x}^{(Q)}$ como na Equação (3.7), e seja $\Theta(\mathcal{C})$ o maior ângulo entre qualquer sinal $\mathbf{y} \in \mathbb{R}^N$ e o átomo mais próximo do dicionário \mathcal{C} . Temos que $\|\mathbf{r}^{(Q)}\| = \|\mathbf{x} - \mathbf{x}^{(Q)}\| \leq \beta_c^{(Q)}$, onde $\beta_c = \sqrt{1 - (2\alpha - \alpha^2) \cos^2(\Theta(\mathcal{C}))} < 1$, $\forall 0 < \alpha < 1$ e $0 \leq \Theta(\mathcal{C}) < \frac{\pi}{2}$.

A seguir, veremos que fatores contribuem para o bom desempenho do algoritmo MPGBP.

3.2 Fatores que afetam o desempenho do algoritmo MPGBP

Um breve olhar sobre a Equação (3.7) e sobre o Teorema 3.1 nos faz observar que existem dois fatores determinantes no processo de aproximação do sinal original \mathbf{x} , quais sejam: o fator de aproximação de escala α e o dicionário utilizado. Ainda é possível notar que a influência de α é advinda apenas do seu valor numérico, enquanto que o dicionário \mathcal{C} manifesta sua influência de duas maneiras distintas: forma do seus átomos \mathbf{v}_{r_m} (Equação (3.7)) e valor de $\Theta(\mathcal{C})$ (Teorema 3.1).

Não obstante a este fato, a análise realizada em [40] a partir do Teorema 3.1 mostra que o limite teórico da relação entre a distorção média quadrática D e a taxa por coeficiente R é dada, após Q passos, por:

$$R \leq \frac{1}{2N(\mathcal{C})} \frac{\log_2(N(\mathcal{C})D)}{\log_2 \beta_c} \log_2 \left[\frac{q(\mathcal{C})}{2 \log_2 \alpha} \log_2 \left(\frac{N(\mathcal{C})D\alpha^2 \cos^2(\Theta(\mathcal{C}))}{\beta_c^2} \right) \right] \quad (3.8)$$

onde D e R são limitados individualmente por

$$D \leq \frac{\beta_c^{2P}}{N(\mathcal{C})} \quad (3.9)$$

$$R \leq \frac{1}{N(\mathcal{C})} [P \log_2 q(\mathcal{C}) + P b_e] \quad (3.10)$$

com $N(\mathcal{C})$ e $q(\mathcal{C})$ representando, respectivamente, a dimensão e a cardinalidade (número de átomos) do dicionário \mathcal{C} e b_e representando o número de bits gastos na codificação de cada índice k_m .

Na Figura 3.1 vemos o exemplo de um gráfico $R \times \alpha$ construído a partir da Equação (3.8) para $D = 10^{-2}$, $N(\mathcal{C}_i) = 10$ e $q(\mathcal{C}_i) = 10^4$, $\forall i$, com valores diversos de $\Theta(\mathcal{C})$. Esta figura mostra que para cada $\Theta(\mathcal{C})$ diferente existe um α ótimo diferente. Logo, a escolha do valor ótimo de α , para um determinado compromisso taxa-distorção, depende diretamente da dimensão $N(\mathcal{C})$, da cardinalidade $q(\mathcal{C})$ e do valor de $\Theta(\mathcal{C})$ do dicionário \mathcal{C} . Assim, vemos que o bom desempenho do algoritmo MPGBP está intimamente ligado à escolha do dicionário utilizado, de modo que nesta seção nos propomos a investigar que características deve apresentar um bom dicionário para aumentar o desempenho do algoritmo MPGBP. Para tanto, se faz necessária uma análise criteriosa do Teorema 3.1.

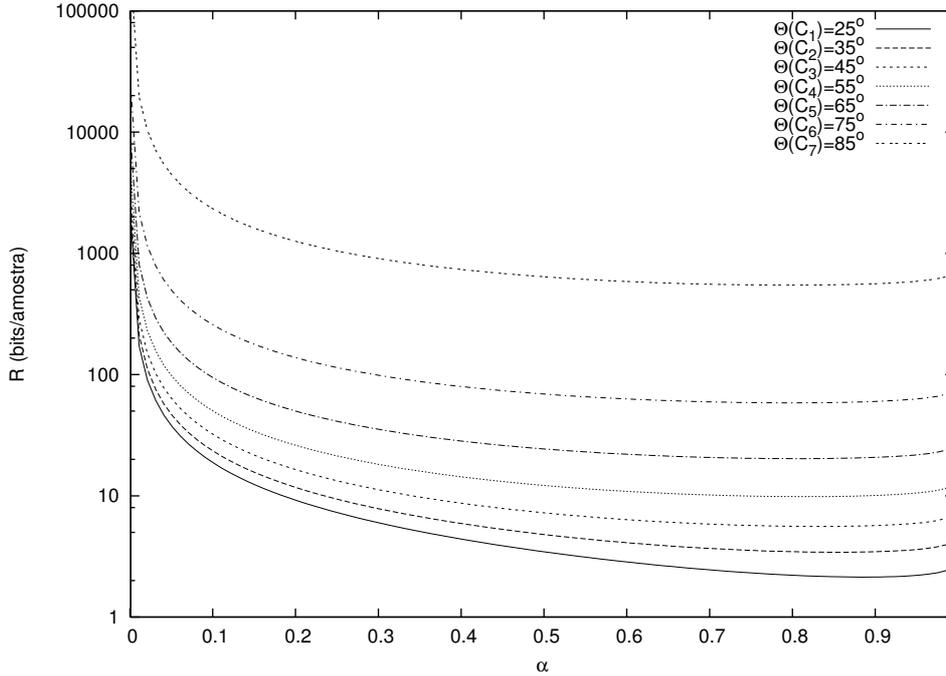


Figura 3.1: Variação da taxa (R) com α para uma distorção $D = 10^{-2}$ e para dicionários de diferentes valores de $\Theta(\mathcal{C}_i)$, mas com dimensões e cardinalidades iguais a $N(\mathcal{C}_i) = 10$ e $q(\mathcal{C}_i) = 10^4$, $\forall i$, respectivamente.

Deste teorema, vemos que no passo Q o erro de aproximação é limitado por $\beta_c^{(Q)}$. Logo, quanto menor for o valor de β_c , menor será a distorção após Q

passos. Como β_c decresce juntamente com $\Theta(\mathcal{C})$, então, para se ter uma distorção pequena, $\Theta(\mathcal{C})$ também deve ser pequeno. Como a cardinalidade precisa, em geral, aumentar para diminuir $\Theta(\mathcal{C})$, e a taxa após Q passos aumenta à medida que a cardinalidade $q(\mathcal{C})$ do dicionário cresce (mais precisamente, a taxa cresce com o aumento de $\log(q(\mathcal{C}))$, ver Equação 3.10), então um bom compromisso entre taxa e distorção é atingido para dicionários com valores de $\Theta(\mathcal{C})$ pequenos, mas que não possuam $q(\mathcal{C})$ muito grande. Isto pode ser ilustrado através do gráfico da Figura 3.2. Neste gráfico são apresentadas curvas taxa-distorção para os dicionários \mathcal{C}_i de mesma dimensão, mas com valores de $\Theta(\mathcal{C})$ e $q(\mathcal{C})$ especificados na Tabela 3.1. (note que cada dicionário possui um α ótimo próprio, corroborando com a análise levantada anteriormente.)

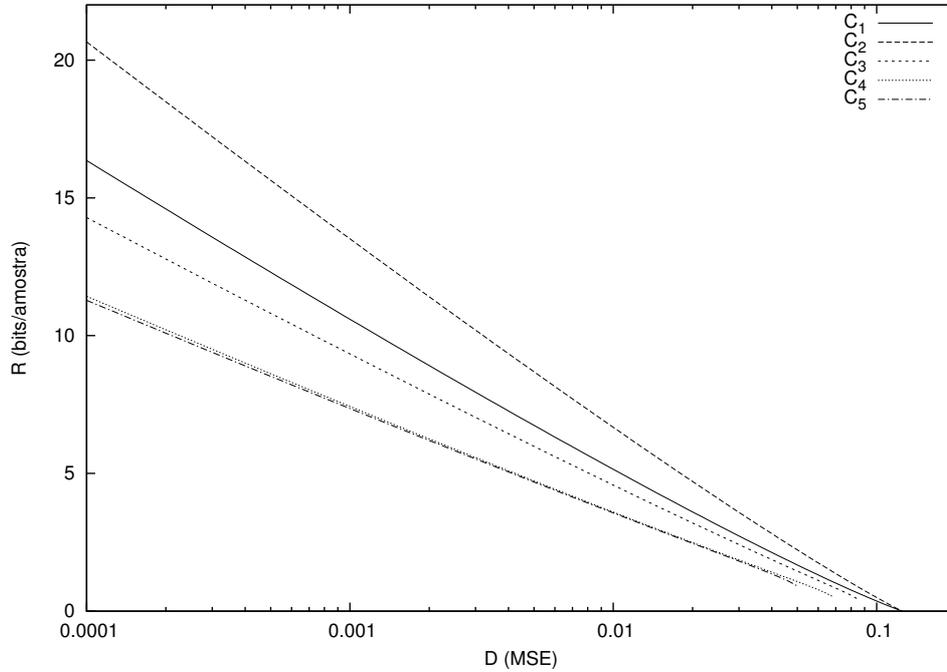


Figura 3.2: Curva R - D para os diferentes dicionários \mathcal{C}_i especificados na Tabela 3.1.

Confrontando as curvas dos dicionários \mathcal{C}_1 e \mathcal{C}_2 vemos que apesar de $\Theta(\mathcal{C}_1) = \Theta(\mathcal{C}_2)$, \mathcal{C}_1 apresenta um desempenho superior ao de \mathcal{C}_2 por possuir um menor valor de cardinalidade. Por outro lado, o dicionário \mathcal{C}_2 apresenta uma cardinalidade inferior a dos dicionários \mathcal{C}_3 e \mathcal{C}_5 . No entanto, o valor de $\Theta(\mathcal{C}_2)$ é superior aos valores de $\Theta(\mathcal{C}_3)$ e $\Theta(\mathcal{C}_5)$, levando \mathcal{C}_2 a atingir o pior desempenho dentre estes dicionários. Por último, note que \mathcal{C}_4 e \mathcal{C}_5 apresentam os melhores compromissos taxa-distorção, pois por um lado \mathcal{C}_4 apresenta uma pequena cardinalidade com um valor de $\Theta(\mathcal{C})$ pequeno e, por

Tabela 3.1: Valores de dimensão ($N(\mathcal{C}_i)$), cardinalidade ($q(\mathcal{C}_i)$), $\Theta(\mathcal{C}_i)$ e α_i ótimo dos dicionários (\mathcal{C}_i) utilizados na construção do gráfico R - D da Figura 3.2.

Dicionário (\mathcal{C}_i)	Dimensão ($N(\mathcal{C}_i)$)	Cardinalidade ($q(\mathcal{C}_i)$)	$\Theta(\mathcal{C}_i)$	α_i ótimo
\mathcal{C}_1	8	240	45°	0,80
\mathcal{C}_2	8	2160	45°	0,83
\mathcal{C}_3	8	6720	35°	0,85
\mathcal{C}_4	8	2400	32°	0,85
\mathcal{C}_5	8	9120	29°	0,87

outro lado, \mathcal{C}_5 possui um pequeno valor de $\Theta(\mathcal{C})$, mas com $q(\mathcal{C})$ não muito grande, garantindo a ambos uma boa relação $\Theta(\mathcal{C})$ - $q(\mathcal{C})$.

Como $\Theta(\mathcal{C})$ é o maior ângulo entre um sinal y qualquer do \mathbb{R}^N e o vetor de \mathcal{C} mais próximo, ele é obtido na região de \mathbb{R}^N que possui o maior “buraco” – região do espaço \mathbb{R}^N na qual o sinal y , correspondente a $\Theta(\mathcal{C})$, se localiza – (ver Figura 3.3a). Do exemplo em 2-D mostrado na Figura 3.3, podemos estipular que, dado um dicionário \mathcal{C} (Figura 3.3a), $\Theta(\mathcal{C})$ pode ser reduzido através: **1**) da inserção de vetores extras de modo a preencher apropriadamente as regiões “vazias” do espaço (Figura 3.3b) ou **2**) de uma melhor distribuição dos vetores (Figura 3.3c). Observe que com **2**) a cardinalidade $q(\mathcal{C})$ do dicionário \mathcal{C} é preservada. Por outro lado, com **1**) a cardinalidade $q(\mathcal{C})$ aumenta e há um compromisso entre o decréscimo de $\Theta(\mathcal{C})$ e o aumento de $q(\mathcal{C})$ que deve ser observado.

Note ainda que qualquer dicionário \mathcal{C}''' obtido por rotação dos elementos de \mathcal{C} , encarados como vetores em \mathbb{R}^N , possui $\Theta(\mathcal{C}''') = \Theta(\mathcal{C})$, o que, através deste teorema, equivaleria dizer que ele teria o mesmo desempenho do dicionário \mathcal{C} quando utilizado no codificador MPGBP. Contudo, durante a primeira iteração do algoritmo MPGBP, a redução do erro de aproximação depende primordialmente das formas de onda dos átomos do dicionário, como mostra a seguinte equação:

$$\mathbf{r}^{(1)} = \mathbf{x} - \langle \mathbf{v}_{\mathbf{r}_1}, \mathbf{x} \rangle \mathbf{v}_{\mathbf{r}_1} \quad (3.11)$$

Quanto mais estas formas de onda forem parecidas com trechos do sinal \mathbf{x} , maior será o valor do produto interno entre eles e, por conseguinte, menor será o

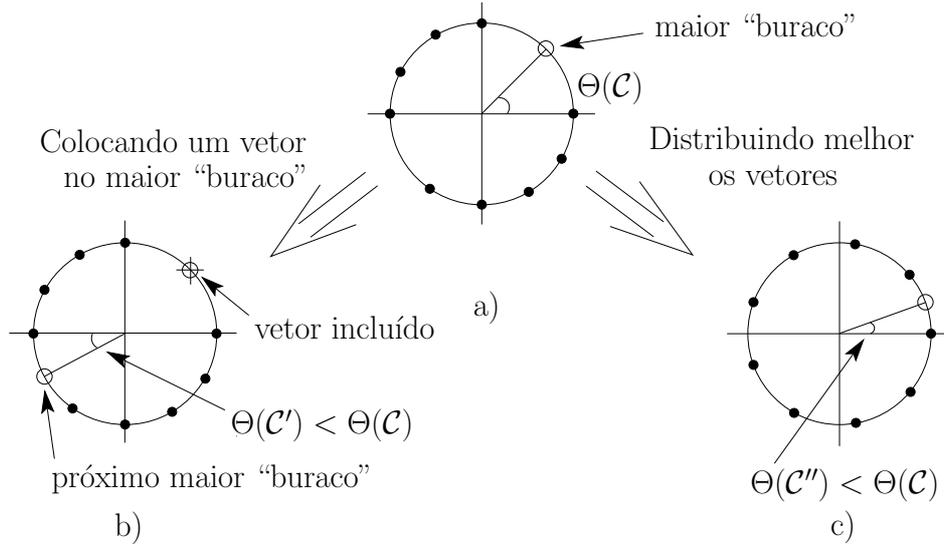


Figura 3.3: $\Theta(\mathcal{C})$ em a) pode ser reduzido: b) adicionando um vetor extra no maior buraco do dicionário \mathcal{C} no \mathbb{R}^2 ; c) distribuindo melhor os vetores em \mathbb{R}^2 .

valor de $\|\mathbf{x} - \alpha^{km} \mathbf{v}_{r_m}\|$. Desta forma, dentre todos os dicionários com o mesmo $\Theta(\mathcal{C})$, deve-se obter aquele cujas características sejam as mais próximas daquelas do sinal.

Assim, esta dissertação é orientada à busca de bons dicionários para o algoritmo MPGBP, os quais devem apresentar:

- (i) pequenos valores de $\Theta(\mathcal{C})$ e $q(\mathcal{C})$;
- (ii) átomos os mais parecidos possível com o sinal.

No capítulo 5 veremos que dicionários foram concebidos segundo esta análise e que resultados concretos obtivemos com seu uso no codificador de vídeo baseado no algoritmo MPGBP descrito a seguir. Estes dicionários foram baseados num novo conjunto de funções 2-D que representam as singularidades contidas em sinais bidimensionais de forma bem mais eficiente do que representações mais convencionais, tais como wavelet e DCT, por exemplo. Tais funções são chamadas de *ridgelet* [12] e sua descrição é dada na Seção 4.1.

3.3 Codificador de Vídeo baseado no MPGBP

Devido à similaridade existente entre o MPGBP e o MP, foram realizados alguns testes comparativos de forma a avaliar a diferença de performance entre estes

dois algoritmos, averiguando o quão eficiente é a decomposição em planos de bits generalizados. O contexto de avaliação utilizado foi o de compressão de vídeo, de maneira que um codificador de vídeo baseado em MPGBP [10, 40, 39] foi desenvolvido e seus resultados comparados com o codificador MP de Neff e Zakhor [6]. A seguir veremos a descrição do codificador de vídeo MPGBP utilizado, o qual é a base do codificador de vídeo proposto nesta Tese.

Como dito anteriormente, na Seção 2.2, algumas das considerações e procedimentos adotados por Neff e Zakhor foram utilizados na confecção do compressor de vídeo MPGBP em [10, 40]. De fato, a única alteração imposta neste codificador está relacionada à codificação da imagem residual, onde o algoritmo MP foi substituído pelo algoritmo MPGBP. Assim, durante o procedimento de redução da redundância temporal o compressor de vídeo baseado em MPGBP também utiliza como modelo de estimação/compensação de movimento o TMN [23].

Esta alteração advém do fato do algoritmo MPGBP decompor o valor da projeção p_n nos planos de bits generalizados k_m . Embora todo o procedimento de busca, de casamento e de codificação da posição/forma do átomo seja realizado como no codificador de vídeo MP, o algoritmo MPGBP não mais codifica os valores quantizados do produto interno, mas sim os índices k_m correspondentes aos planos de bits j nos quais os produtos internos entre \mathbf{x} e os vetores \mathbf{v}_{r_m} são mapeados. Como consequência direta desta modificação, há uma alteração no tipo de codificador utilizado para codificar as informações de projeção.

Conforme descrito no final da Subseção 2.2.2, no processo de codificação dos produtos internos quantizados é utilizado um codificador de Huffman de comprimento fixo. Entretanto, este mesmo codificador já não pode ser utilizado na codificação dos índices k_m , uma vez que este codificador foi projetado para codificar símbolos cujos valores decresçam juntamente com a energia dos resíduos. Como os valores dos índices k_m crescem com o decréscimo da energia dos resíduos, foi proposto um codificador que melhor se adapte ao comportamento destes índices. Trata-se de uma versão modificada do codificador aritmético adaptativo [43].

Como o codificador aritmético realiza a codificação através de sucessivos refinamentos dentro de um intervalo de valores conhecidos, e como o valor máximo de k_m é, a princípio, desconhecido, uma pequena modificação se fez necessária. Inici-

almente, considera-se que k_m possui dois possíveis valores ($k_m = 1$ e $k_m = 2$), mais um código de escape. Sendo necessária a transmissão de $k_m = 3$, primeiramente transmite-se o código de escape para indicar um aumento no número de símbolos e então transmite-se o código de $k_m = 3$. Agora, os possíveis símbolos são $k_m = 1, 2, 3$ e o código de escape incrementado. Este processo é repetido para cada novo valor de k_m que esteja fora do intervalo atual.

Uma vez que estes resultados tiveram como intuito avaliar a diferença de performance entre os algoritmos MP e MPGBP no tocante à estratégia de decomposição em plano de bits generalizados, o dicionário utilizado nos casamentos e o controle de taxa empregado na codificação dos átomos do codificador MPGBP foram os mesmo utilizados na implementação do codificador MP em [6] (ver Subseções 2.2.1 e 2.2.3). E, de fato, a partir da visualização destes resultados, observa-se um ganho de desempenho devido da utilização de tal estratégia.

Para maiores detalhes desta implementação, ver [10, 40]

3.4 Dicionário proposto

A análise do Teorema 3.1 e da Equação (3.7) levantada previamente na Seção 3.2, indica que o dicionário tem um papel fundamental no desempenho do algoritmo MPGBP. Apesar do sucesso da implementação deste algoritmo no contexto de vídeo [10], seu objetivo, como mencionado anteriormente, foi de avaliar apenas a eficiência da decomposição em planos de bits generalizados. Desta forma, não se pôde avaliar a escolha de diferentes dicionários, uma vez que o dicionário utilizado foi o mesmo da implementação do codificador MP. Assim, uma importante questão ficou sem resposta: existem outros dicionários cuja utilização resulta em uma melhor performance do algoritmo MPGBP?

Um levantamento bibliográfico realizado em busca de candidatos para desempenhar o papel de átomos do dicionário na representação MPGBP apontou para um novo conjunto de funções que têm despertado o interesse da comunidade de processamento de imagem/vídeo, as Ridegelets [12] (assim como curveletes [8] e contourlets [17]). Estas funções representam de maneira bastante eficiente sinais cujas descontinuidades estão dispostas ao longo de linhas. Como os contornos de

uma imagem podem ser vistos como descontinuidades ao longo de linhas, elas representam uma alternativa efetiva para a decomposição de imagens [12]. Isto é especialmente relevante no caso de vídeo, pois as dfds são compostas essencialmente de contornos. Desta forma, dicionários cuja construção se baseia nestas funções têm o potencial de satisfazer à condição (ii) acima mencionada.

De forma a ilustrar na prática as conseqüências de boas escolhas de dicionário, nesta Tese nós investigamos o desempenho do codificador de vídeo MPGBP descrito na Seção 3.3, usando novos dicionários supercompletos desenvolvidos a partir das ridgelets que satisfazem bem os critérios (i) e (ii) acima. No Capítulo 5 apresentamos e analisamos resultados de simulação.

Capítulo 4

X-Lets – Dicionários direcionais

Para sinais unidimensionais, os quais apresentam descontinuidades pontuais, as funções wavelet se estabeleceram como a grande ferramenta de decomposição por proverem uma boa representação (esparsa e bem localizada no tempo e na frequência) e possibilitarem a construção de algoritmos eficientes. No domínio bidimensional, as wavelets separáveis são muito eficientes para representar descontinuidades, tais como arestas, dispostas nas direções horizontal e vertical. Em particular elas possuem transformadas rápidas através de estruturas em árvore (EZW [41], SPIHT [42]). Dessa forma, as wavelets obtiveram grande sucesso em aplicações de processamento de sinais e comunicações, sendo, por exemplo, utilizadas na composição do modelo de compressão de imagem JPEG-2000 [44].

Entretanto, imagens naturais não são compostas simplesmente de elementos dispostos na horizontal e/ou vertical. Suas descontinuidades são tipicamente dispostas em curvas suaves, isto é, contornos, representando as bordas dos objetos. Como resultado da expansão de bases 1-D separáveis, as wavelets 2-D são boas para isolar descontinuidades horizontais e verticais, mas não “enxergam” bem descontinuidades suaves ao longo de contornos. Ou seja, wavelets separáveis só capturam um número limitado de informações direcionais, as quais são uma importante característica de sinais multidimensionais [5]. Isso indica que se necessita de representações alternativas para sinais multidimensionais.

Uma forma de ver como se pode melhorar o desempenho das wavelets 2-D separáveis parte da observação da Figura 4.1. Esta figura ilustra a representação de uma curva suave no domínio bidimensional com o uso de duas representações

distintas. Em ambos os casos a curva é aproximada com uma combinação de funções do respectivo dicionário. A medida de eficiência do dicionário usado é dada pela menor quantidade M de elementos significativos utilizados para representar a curva com uma determinada distorção $\|\mathbf{x}^{(M)} - \mathbf{x}\|^2$.

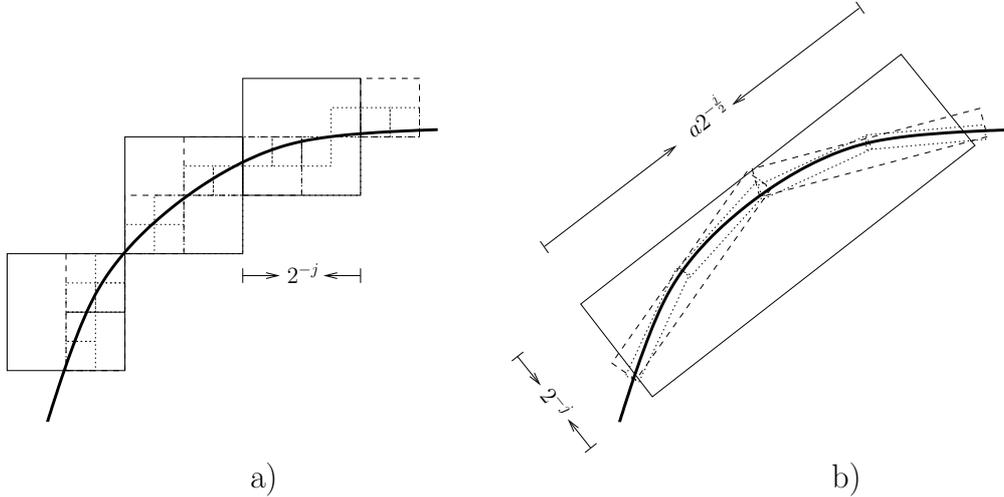


Figura 4.1: Representação de um contorno suave através do uso a) das wavelets 2-D separáveis e b) de funções retangulares rotacionadas.

Na Figura 4.1a é empregada a Transformada Wavelet 2-D separável, ou seja, o dicionário é composto por funções wavelet separáveis. Note, então, que este dicionário representa a curva através de bases com região de suporte quadrada e de dimensões diádicas (potência de 2). Isto se deve ao fato de as funções wavelet 2-D separáveis serem geradas a partir de produtos externos de wavelets 1-D de mesma resolução j . Assim, os quadrados maiores representam as bases de menor frequência e os quadrados menores representam as bases de maior frequência, ou seja, com um maior nível de detalhamento.

Para a escala 2^{-j} existem $M = O(2^j)$ wavelets significativas e a taxa de decaimento da distorção para estes M elementos é da ordem de M^{-1} ; já no caso de funções 1-D, de singularidade 0-D, esta taxa decai mais rapidamente com M [12]. Por outro lado, se dispuséssemos de funções base com região de suporte retangular e com orientações arbitrárias, o número de funções significativas na representação desta curva cairia significativamente. Em [12], mostra-se que a quantidade de coeficientes significativos pode chegar a $O(2^{\frac{j}{2}})$.

De fato, esta intuição foi previamente formalizada por Candès e Donoho na

metade final dos anos 90, através da construção de um novo conceito de decomposição não-adaptativo de funções bidimensionais, as curvelets [8]. Como o nome sugere, esta decomposição é especializada em detectar singularidades curvilíneas, de modo que, no contexto de aproximação da Figura 4.1, a taxa de decaimento para os M primeiros coeficientes significativos é da ordem de $(\log M)^3 M^{-2}$, a qual é próxima de $O(M^{-2})$ [8]. Tal acontecimento quebrou uma forte conjuntura (Folk-Conjecture/Folk-Theorem), a qual afirma que o valor $O(M^{-2})$ seria somente alcançado por métodos adaptativos e excederia de longe o limite atingível por qualquer representação não-adaptativa, causando grande surpresa [8].

Note que os princípios de multidirecionamento e anisotropia embutidos nestas funções concede um benefício para a decomposição de sinais multidimensionais equiparável ao benefício trazido pelas wavelets ao introduzir o conceito de multiresolução e localização tempo-freqüência (utilizando transformadas tais como Fourier 2-D e DCT, a taxa de decaimento da distorção $\|\mathbf{x}^{(M)-\mathbf{x}}\|$ para os M primeiros termos é da ordem de $M^{-\frac{1}{2}}$ [8]).

No entanto, as curvelets são uma evolução de uma categoria de funções especialmente construídas para a detecção de singularidades retilíneas, as ridgelets [45]. Como os dicionários desenvolvidos e expostos nesta tese são baseados nestas funções ridgelet, nas próximas seções descreveremos mais detalhadamente a sua formação e algumas de suas derivações, quais sejam: curvelets [8] e contourlets [46]. A partir disso, justificaremos a nossa escolha pelas ridgelets assim como proporemos algumas modificações visando um melhor desempenho no contexto do codificador de vídeo MPGBP.

4.1 Ridgelets

A Transformada Wavelet se mostrou muito eficiente na decomposição de sinais 1-D [5]. A esparsidade atingida por seus coeficientes no tratamento de singularidades 0-D superou sobremaneira àquela alcançada pelos coeficientes de transformadas clássicas – Fourier [47] e DCT [4] – causando grande impacto na comunidade de processamento de sinais. Entretanto, tal transformada foi superestimada, sendo utilizada em aplicações fora de sua especialidade, demonstrando algumas limitações.

Em particular, sua utilização no contexto de compressão de imagens e vídeo mostrou que esta transformada não lida tão bem com descontinuidades lineares tanto quanto com descontinuidades pontuais (ver Figura 4.1).

Assim, devido à falta de funções mais adaptadas ao domínio multidimensional, vários pesquisadores procuraram criar alternativas para contornar estas limitações das wavelets. Em particular, Candès [45] introduziu um conjunto de funções melhor adaptável ao domínio multidimensional que as wavelets e que apresenta uma solução estável e implementável para aproximação de funções n -dimensionais através do uso das *ridges* [15]. Estas funções são chamadas de *funções ridgelet* e sua descrição é dada nesta seção.

Ao realizar pesquisas na área de redes neurais, Candès [13] trabalhou com um conceito de funções criado na área de tomografia, as funções *ridge* [15, 48]. As *ridges* são funções multivariáveis $f : \mathbb{R}^n \rightarrow \mathbb{R}$ que obedecem à seguinte equação:

$$f(x_1, x_2, \dots, x_n) = r(a_1x_1 + a_2x_2 + \dots + a_nx_n) = r(\mathbf{x} \cdot \mathbf{a}) \quad (4.1)$$

onde $r : \mathbb{R} \rightarrow \mathbb{R}$ e $\mathbf{a} = (a_1, a_2, \dots, a_n) \in \mathbb{R}^n \setminus \{0\}$. Ou seja, são funções multivariáveis constantes ao longo do hiperplano $a_1x_1 + a_2x_2 + \dots + a_nx_n = c$, $c \in \mathbb{R}$.

Desde o seu surgimento [15], datado de 1975, pesquisadores têm usado a superposição das funções *ridge* como alternativa aos métodos padrão daquela época para a aproximação de funções multivariáveis. Como exemplo, essa técnica de aproximação teve grande influência no surgimento do método de Projection Pursuits, um antecessor do Matching Pursuits, com Friedman [49], Huber [50] e Donoho [51].

Alguns resultados clássicos de pesquisas em outras áreas, como por exemplo [52, 53], onde a sua aplicação fora difundida, garantem que é possível aproximar qualquer função com n -variáveis com tais superposições, mas não chegam, de fato, a conceber uma seqüência de aproximação matematicamente concreta. Mesmo outros resultados que exibem tais seqüências, falham no que diz respeito à estabilidade [13]. De modo a resolver esta problemática, Candès concebeu um sistema de *ridges* estável e implementável usando como função $r : \mathbb{R} \rightarrow \mathbb{R}$ (ver Equação 4.1) funções oscilatórias $\psi : \mathbb{R} \rightarrow \mathbb{R}$. Acreditando que as características de multiresolução e localização tempo-frequência das wavelets são de extrema importância na representação de sinais, compôs um novo dicionário onde tais funções oscilatórias ψ são funções wavelet. Assim, este novo dicionário $\mathcal{D}_{Ridgelet} = \{\psi_\gamma : \gamma \in \Gamma\}$ é composto

pelas ridgelets:

$$\psi_\gamma(\mathbf{x}) = a^{\frac{1}{2}} \psi \left(\frac{\mathbf{u} \cdot \mathbf{x} - b}{a} \right) \quad (4.2)$$

onde o parâmetro $\gamma = (a, b, \mathbf{u})$ é um dos elementos do conjunto $\Gamma \equiv \{(a, b, \mathbf{u}) : a, b \in \mathbb{R}, a > 0, \mathbf{u} \in \mathbf{S}^{n-1}\}$, com \mathbf{S}^{n-1} denotando a hipersfera unitária no \mathbb{R}^n .

Disto, definiu-se no, domínio 2-D, as funções ridgelet bivariáveis $\psi_{\gamma=(a,b,\theta)} : \mathbb{R}^2 \rightarrow \mathbb{R}$ como sendo:

$$\psi_\gamma(\mathbf{x}) = a^{\frac{1}{2}} \psi \left(\frac{\cos(\theta)x_1 + \sin(\theta)x_2 - b}{a} \right) \quad (4.3)$$

de modo que assim como a linha $\cos(\theta)x_1 + \sin(\theta)x_2 = c$ é um conjunto de pontos alinhados na direção θ , a ridgelet representa uma superposição de wavelets ao longo da ridge $\cos(\theta)x_1 + \sin(\theta)x_2 = c$ (ver Figura 4.2). Daí o nome é *ridgelet*.

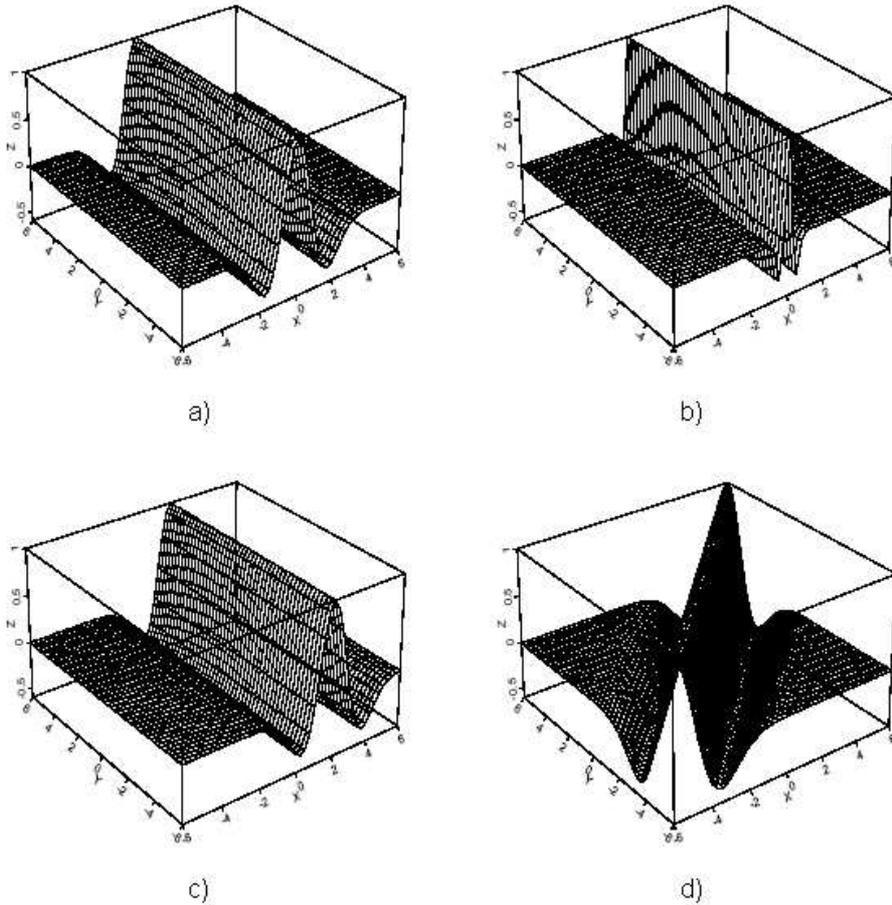


Figura 4.2: Funções ridgelet: a) original; b) escalonada; c) transladada; d) rotacionada.

Assim, dada uma função bidimensional integrável $f(\mathbf{x})$, os coeficientes ridge-

let são dados por [45]:

$$\mathcal{R}_f(a, b, \theta) = \int \bar{\psi}_{a,b,\theta}(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \quad (4.4)$$

onde $\bar{\psi}$ representa a versão normalizada de ψ . Candès [45] também chegou a uma fórmula de reconstrução perfeita, dada por:

$$f(\mathbf{x}) = \int_0^{2\pi} \int_{-\infty}^{\infty} \int_0^{\infty} \mathcal{R}_f(a, b, \theta) \psi_{a,b,\theta}(\mathbf{x}) \frac{da}{a^3} db \frac{d\theta}{4\pi} \quad (4.5)$$

mostrando que qualquer integrável função pode ser escrita como uma superposição de funções *ridge*. Vale lembrar que estas duas integrais também foram descobertas por Murata [14] na mesma época e de forma independente. Entretanto, Cadès também mostrou que sua aproximação por ridges é estável, obedecendo à relação de Parseval, como segue:

$$\int |f(\mathbf{x})|^2 d\mathbf{x} = \int_0^{2\pi} \int_{-\infty}^{\infty} \int_0^{\infty} |\mathcal{R}_f(a, b, \theta)|^2 \frac{da}{a^3} db \frac{d\theta}{4\pi} \quad (4.6)$$

o que não foi mostrado no trabalho do Murata [14].

Entretanto, da forma que as ridgelets foram concebidas, apresentaram algumas dificuldades na sua transposição para o domínio discreto, algo particularmente grave no tocante ao processo de compressão de imagens e vídeo, processo este preponderantemente digital. O maior problema enfrentado é o fato das ridgelets não pertencerem ao domínio de energia finita $L^2(\mathbb{R}^2)$, justamente por serem essencialmente funções *ridge*, as quais não possuem limitação de energia. Ciente disto, Candès propôs uma decomposição em frames [13] de forma a superar esta dificuldade e ligar suas funções ridgelet ao domínio discreto. No entanto, sua formulação não foi completamente clara, dando margem a dualidades/ambigüidade sobretudo com relação à construção ao seu sistema de síntese. Logo, os “frames de ridgelet” pecaram pelo mesmo motivo que as ridges originais: falta de uma fórmula fechada para a seqüência de decomposição [13, 12].

Donoho, então, propôs uma solução para este problema utilizando uma outra interpretação das ridgelets de Candès. Segundo esta outra interpretação, as ridgelets são wavelets aplicadas ao domínio Radon [16]. Isto porque, assim como uma linha no domínio espacial é mapeada em um ponto no domínio Radon [54], as singularidades lineares são transformadas em singularidades pontuais. Então, às estas singularidades pontuais se aplica as wavelets, que são especialistas na sua detecção.

Assim, seja $\{\psi_{j,k}(t) : j, k \in \mathbb{Z}\}$ uma base ortonormal de wavelets de Meyer pertencente a $L^2(\mathbb{R})$ e seja $\{w_{i_0,l}^0(\theta), l = 0, \dots, 2^{i_0}-1; w_{i,l}^1(\theta), i \geq i_0, l = 0, \dots, 2^i-1\}$ uma base ortonormal para $L^2[0, 2\pi)$ composta pela scaling function periódica de Lemarié $w_{i_0,l}^0$ no nível i_0 e pela wavelet periódica de Meyer $w_{i,l}^1$ para os níveis $i \geq i_0$, devidamente normalizadas, e seja $\hat{\psi}_{j,k}(\omega)$ a transformada de Fourier de $\psi_{j,k}(t)$, as ridgelets ortonormais $\rho_\lambda(\mathbf{x}), \lambda = (j, k; i, l, \varepsilon)$ são funções de $\mathbf{x} \in \mathbb{R}^2$ definidas no domínio freqüencial como [12]:

$$\hat{\rho}_\lambda(\xi) = |\xi|^{-\frac{1}{2}} \frac{\hat{\psi}_{j,k}(|\xi|)w_{i,l}^\varepsilon(\theta) + \hat{\psi}_{j,k}(-|\xi|)w_{i,l}^\varepsilon(\theta + \pi)}{2} \quad (4.7)$$

Os índices obedecem às seguintes relações: $j, k \in \mathbb{Z}, l = 0, \dots, 2^i - 1; i \geq i_0$, e se $\varepsilon = 0, i = \max(i_0, j)$, enquanto que se $\varepsilon = 1, i \geq \max(i_0, j)$, onde j representa a escala da *ridge*, i representa a escala angular, k representa a posição da *ridge* e l representa a posição angular.

A wavelet de Meyer é dada por:

$$\hat{\psi}(\omega) = \begin{cases} \frac{1}{\sqrt{2\pi}} e^{i\omega/2} \operatorname{sen} \left(\frac{\pi}{2} \nu \left(\frac{3}{2\pi} |\omega| - 1 \right) \right), & \frac{2\pi}{3} \leq |\omega| \leq \frac{4\pi}{3} \\ \frac{1}{\sqrt{2\pi}} e^{i\omega/2} \cos \left(\frac{\pi}{2} \nu \left(\frac{3}{4\pi} |\omega| - 1 \right) \right), & \frac{4\pi}{3} \leq |\omega| \leq \frac{8\pi}{3} \\ 0, & |\omega| \notin \left[\frac{2\pi}{3}, \frac{8\pi}{3} \right] \end{cases} \quad (4.8)$$

com $\nu(a) = a^4(35 - 84a + 70a^2 - 20a^3), a \in [0, 1]$. A scaling function de Lemarié é dada por:

$$\hat{\phi}(\omega) = \begin{cases} (2\pi)^{-\frac{1}{2}} \left[\frac{\operatorname{sen} \left(\frac{\omega}{2} \right)}{\frac{\omega}{2}} \right]^{N+1}, & \text{se } N \text{ for ímpar} \\ (2\pi)^{-\frac{1}{2}} e^{-i\frac{\omega}{2}} \left[\frac{\operatorname{sen} \left(\frac{\omega}{2} \right)}{\frac{\omega}{2}} \right]^{N+1}, & \text{se } N \text{ for par} \end{cases} \quad (4.9)$$

Já a wavelet periódica de Meyer e a scaling function periódica de Lemarié são periodizações destas e são dadas, respectivamente, como [12]:

$$w_{i,l}^1(\theta) = \sum_{h=-\infty}^{\infty} \psi_{i,l+h2^i} \left(\frac{\theta}{2\pi} \right), \quad i \geq i_0 > 0, l = 0, \dots, 2^i; \quad (4.10)$$

e

$$w_{i_0,l}^0(\theta) = \sum_{h=-\infty}^{\infty} \phi_{i_0,l+h2^{i_0}} \left(\frac{\theta}{2\pi} \right), \quad l = 0, \dots, 2^{i_0}; \quad (4.11)$$

Embora as ridgelets ortonormais não sejam efetivamente construídas a partir de um somatório de funções *ridge*, pode-se mostrar que elas são uma espécie de

média de funções *ridge* [12]. Assim, elas podem ser vistas como substitutas em L^2 das ridgelets originais, desempenhando muitas das mesmas tarefas destas. Além disso, apesar de ter uma definição mais complexa que as ridgelets originais, sua implementação é mais fácil de ser realizada, além de adicionar algumas propriedades de localização freqüencial–angular [12].

Com a construção de uma versão $L^2(\mathbb{R}^2)$ para as ridgelets, tornou-se possível sua implementação no domínio discreto. Surgiram, então, alguns trabalhos nesta direção [18, 55, 56, 57]. Um, entretanto, nos chamou a atenção por se adaptar melhor ao nosso contexto de vídeo. Para a construção dos nossos dicionários baseados em ridgelets, partimos da versão digital de Donoho e Flésia [18] por ser igual ou mais eficiente que as outras e mais simples de implementar em termos de dicionário para o MPGBP. Assim, descreveremos somente esta versão, por se tratar da de nosso interesse mais direto.

Para Donoho e Flésia, as imagens digitais são matrizes $n \times n$ indexadas pelas coordenadas (u, v) e demarcadas pelo quadrado $-\frac{n}{2} \leq u, v < \frac{n}{2}$ centradas no ponto $(0,0)$. Sendo $\tan(\theta_{l,n}^1) = \frac{2l}{n}$, $-\frac{n}{2} \leq l < \frac{n}{2}$; $\cotan(\theta_{l,n}^2) = \frac{2l}{n}$, $-\frac{n}{2} \leq l < \frac{n}{2}$, a *ridgelet digital* $\rho_{j,k,s,l}$ é uma matriz $n \times n$ formulada como uma função *ridge* derivadas da wavelet de Meyer através das seguintes equações:

$$\rho_{j,k,s,l}(u, v) = \psi_{j,k}(u + \tan(\theta_l^s)v), \quad s = 1 \quad (4.12)$$

e

$$\rho_{j,k,s,l}(u, v) = \psi_{j,k}(v + \cotan(\theta_l^s)v), \quad s = 2 \quad (4.13)$$

As ridgelets mostram sensíveis melhoras com relação às wavelets no tratamento de singularidades dispostas em linhas retas. Ao contrário, porém, das singularidades pontuais, que só se apresentam de uma forma, pontos, as singularidades lineares não são somente do tipo linha reta, mas se apresentam também através de curvas suaves. Para este caso, as ridgelets não apresentam uma representação esparsa, pois a Transformada Radon de tais descontinuidades também apresenta singularidades curvilíneas e não pontuais [12]. Contudo, fazendo uso de uma localização multiresolucional apropriada, as ridgelets podem realizar representações esparsas de uma singularidade curvilínea, apresentando sensíveis melhoras com relação às wavelets.

Esta localização multiresolucional foi concebida por Candès e Donoho através da criação das *Curvelets* [8]. Na próxima seção descrevemos sua formulação.

4.2 Curvelets

Infelizmente, a tarefa incumbida às ridgelets é, de certa forma, mais difícil de ser executada do que aquela incumbida às wavelets, uma vez que singularidades zero-dimensionais tem sempre a mesma forma, ponto, enquanto que a forma da singularidade unidimensional pode ser variada, linhas retas ou curvas. Como as ridgelets representam bem apenas as singularidades retilíneas, descontinuidades curvilíneas limitam a ação das ridgelets na representação de singularidades unidimensionais.

Como saída para esta limitação, Candès, Duncan e Donoho [8, 58] concederam às ridgelets uma nova propriedade de localização espaço-freqüencial, a exemplo da propriedade de localização tempo-freqüencial das wavelets. O domínio em questão é dividido suavemente em “quadrados” e então se posiciona fragmentos da curva nestes “quadrados”. A idéia é que para uma escala s suficientemente fina a singularidade curvilínea seja retalhada em vários fragmentos retilíneos, de modo que se possa aplicar as ridgelets, localizadas apropriadamente.

Além da presença da curvatura nas descontinuidades 1-D, outro fato impede a aplicação direta das ridgelets às imagens naturais. Numa imagem natural as descontinuidades são apresentadas de forma mascarada por causa da presença do preenchimento (textura) dos objetos da cena. Assim, para melhor representar estas descontinuidades, é mais vantajoso retirar as informações de baixa freqüência contidas na imagem, deixando apenas os detalhes de alta freqüência que revelam as descontinuidades dos objetos.

Com a união destes dois aspectos, Candès, Duncan e Donoho criaram as curvelets [8, 58] cuja implementação é dada segundo os seguintes passos:

1. *Decomposição em subbandas.* Definindo um banco de filtros $P_0, \Delta_s, s \geq 0$, a imagem f é filtrada em subbandas:

$$f \mapsto (P_0 f, \Delta_1 f, \Delta_2 f, \dots)$$

As diferentes subbandas $\Delta_s f$ contêm detalhes de cerca de 2^{-2s} de largura.

Esta decomposição em subbandas pode ser realizada, por exemplo, com o uso de um sistema de wavelets, e tem como objetivo expor as arestas da imagem original, visíveis somente nas informações de mais alta frequência (subbandas $\Delta_i f$). Assim, se torna viável a aplicação das ridgelets. Já a subbanda de baixa-frequência não é representada pelas ridgelets, sendo representada por um outro método mais adequado, como por exemplo, o Differential Pulse Code Modulation (DPCM) [59].

2. *Particionamento suave.* Após a etapa de decomposição em subbandas, divide-se cada subbanda $\Delta_i f$ em quadrados diádicos dados por:

$$Q = \left[\frac{k_1}{2^s}, \frac{k_1 + 1}{2^s} \right) \times \left[\frac{k_2}{2^s}, \frac{k_2 + 1}{2^s} \right)$$

Definindo um conjunto de janelas suaves $w_Q(x_1, x_2)$ localizadas em torno de Q , multiplica-se cada quadrado pela janela w_Q correspondente produzindo um resultado localizado próximo a Q . Fazendo isso para todos os Q em uma dada escala, ou seja, para todo $Q = Q(s, k_1, k_2)$ com k_1 e k_2 variando, mas s fixo, impõe uma partição suave da função em “quadrados”. Neste estágio do processo, aplica-se esse janelamento a cada uma das subbandas isoladas resultante do estágio anterior:

$$\Delta_s f \mapsto (w_Q \Delta_s f)_{Q \in \mathcal{Q}_s}$$

Na verdade estes “quadrados” não são quadrados, pois: (1) ao longo das descontinuidades, as imagens resultantes da filtragem $\Delta_s f$ exibem *ridges* de largura $\approx 2^{-2s}$ (a largura da banda passante do filtro); (2) as imagens filtradas são divididas em quadrados Q de lados $2^{-s} \times 2^{-s}$. Logo, os “quadrados” que possuem alguma informação (pois nem todos quadrados contêm informação) na verdade exibem fragmentos finos de *ridge* de dimensões $2^{-s} \times 2^{-2s}$. Assim, se estabelece uma importante relação de anisotropia, dada por:

$$\text{largura} \approx \text{comprimento}^2$$

3. *Renormalização.* Para um determinado quadrado diádico Q define-se:

$$(T_Q f)(x_1, x_2) = 2^s f(2^s x_1 - k_1, 2^s x_2 - k_2)$$

como sendo o operador que renormaliza a região de suporte localizada próxima a Q para uma região de suporte próxima a $[0, 1]^2$.

Neste estágio do processo, cada “quadrado” resultante do estágio anterior é renormalizado para a escala unitária:

$$g_Q = (T_Q)^{-1}(w_Q \Delta_s f), \quad Q \in \mathcal{Q}_s$$

4. *Aplicação das Ridgelets.* Cada “quadrado” renormalizado é “analisado” pela ridgelet ortonormal, que é um sistema de elementos base ρ_λ ortonormais em $L^2(\mathbb{R}^2)$. Assim, temos como resultado a geração dos coeficientes α_μ dados como:

$$\alpha_\mu = \langle g_Q, \rho_\lambda \rangle, \quad \mu = (Q, \lambda)$$

De forma a ilustrar o processo de construção das curvelets, observe as Figuras 4.3 e 4.4.

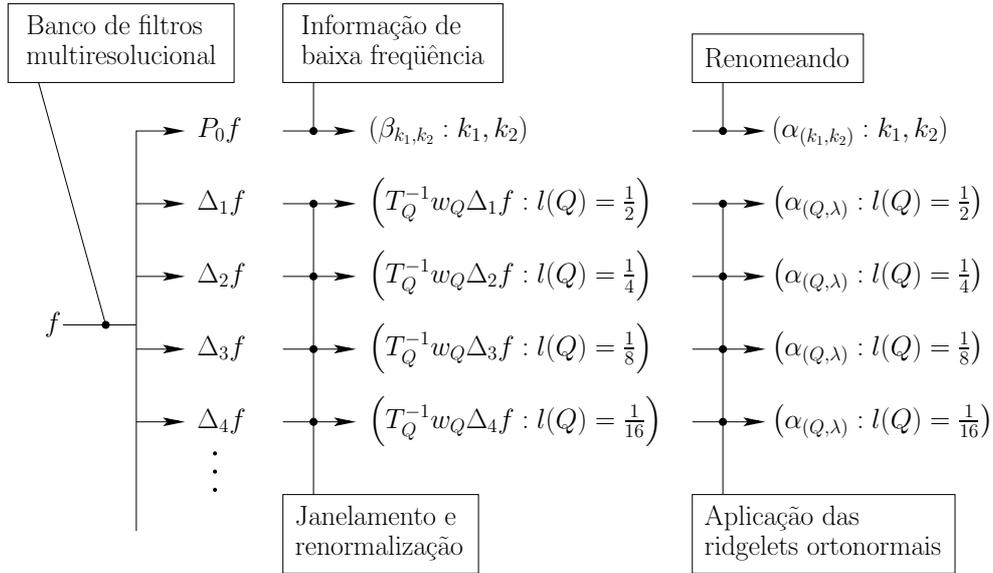


Figura 4.3: Visão geral da decomposição por curvelet.

Para maiores detalhes, ver [8, 58].

Candès e Donoho [60] também propuseram uma ‘segunda geração de curvelets’. Mas desta feita, utilizaram um particionamento freqüencial sem fazer uso das ridgelets. Assim, devido a não utilização das ridgelets, sua descrição não cabe nesta tese.

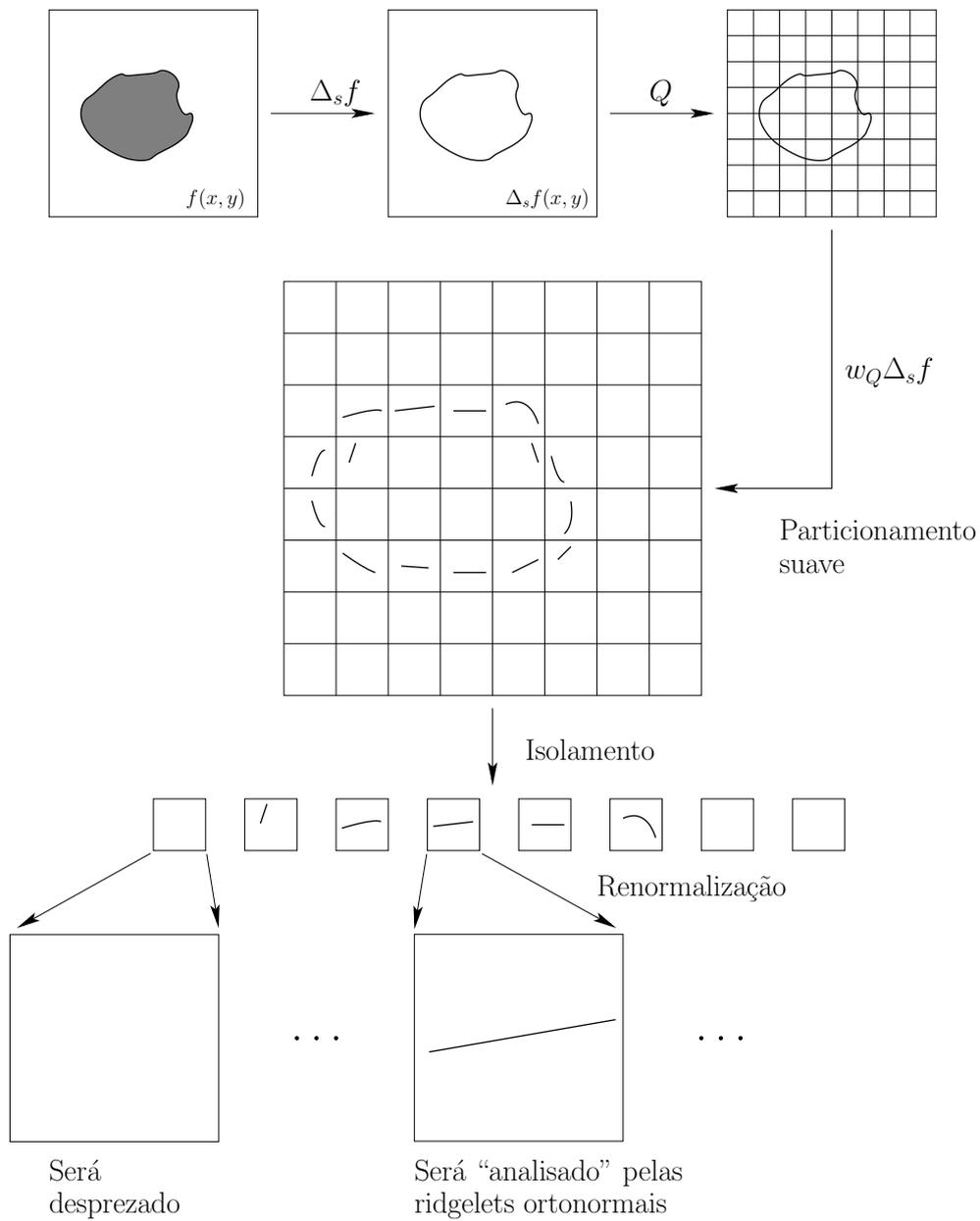


Figura 4.4: Decomposição espacial de uma dada subbanda.

4.3 Contourlets

Assim como as ridgelets apresentaram dificuldades na implementação no domínio discreto, as curvelets também apresentaram.

Um dos problemas recai sobre o fato de a curvelet ser uma transformada baseada numa divisão em blocos, à semelhança da DCT 2-D. Com isso, ou as imagens aproximadas apresentam efeitos de blocagem ou se é necessário utilizar sobreposição de janelas, aumentando a redundância. Outro problema diz respeito ao uso das ridgelets. Como elas são definidas em coordenadas polares há uma certa dificuldade em aplicar a curvelet em imagens discretas – amostradas num grid retangular – sobretudo quando se deseja trabalhar em um sistema que envolva amostragem crítica.

De modo a resolver estes problemas, Do e Vetterli propuseram um novo esquema de representação de imagens bidimensionais chamado de *contourlets* [17, 46]. Eles utilizaram o mesmo princípio adotado por Candès e Donoho, ou seja, criaram um sistema de decomposição de duas etapas: (1) decomposição da imagem em subbandas de forma a evidenciar os contornos e (2) utilização de um sistema multidirecional para a captura de tais contornos. Entretanto, para executar estas tarefas foi utilizado um sistema de banco de filtro de dois estágios chamado de *Banco de Filtros Piramidal Direcional* (PDFB), descrito a seguir.

A primeira etapa deste banco de filtros, a etapa multiescala, é executada através da *Pirâmide de Laplace* (LP). Este banco de filtro, primeiramente desenvolvido por Burt e Adelson no ano de 1983 [61] e adaptado por Do e Vetterli em 2003 [62], realiza uma decomposição em oitavas através da estrutura mostrada na Figura 4.5.

Ao final desta primeira etapa, a imagem é dividida nas componentes c e d . A componente c é a componente de baixa definição que, dependendo do número de oitavas pré-determinado será ou não novamente dividida por outra estrutura LP de análise. Já a componente d é passada à segunda etapa, a multidirecional. Nesta etapa é utilizado o *Banco de Filtros 2-D Direcional* (DFB) criado por Bamberg e Smith [63] em 1993. O DBF é um banco de filtro maximamente decimado que permite reconstrução perfeita. Ele, como o nome sugere, subdivide a banda freqüencial em subbandas direcionais, como mostra a Figura 4.6a), e é eficientemente implementado via decomposição em árvore binária de l -níveis. A exemplo da LP, Do e

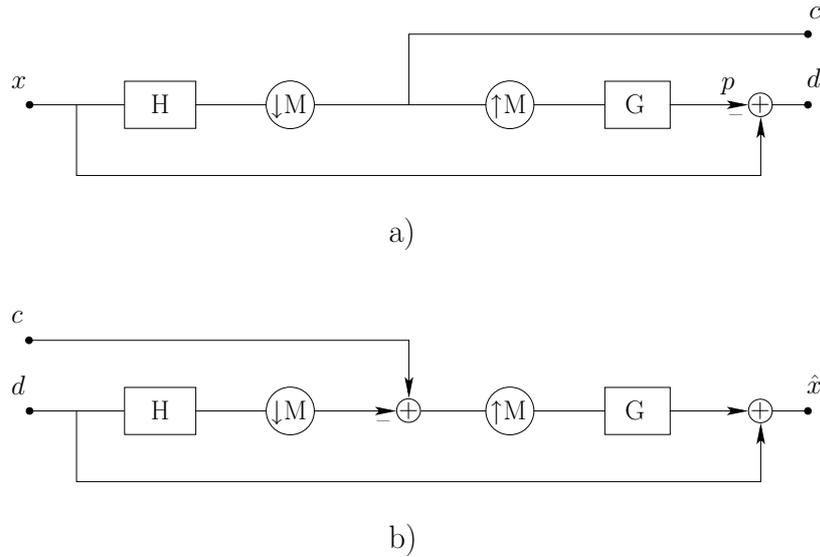


Figura 4.5: Esquema da Pirâmide de Laplace: a) estrutura de análise: as saídas são uma imagem de baixa definição c e uma imagem diferença d composta pela diferença entre o sinal original e uma previsão; b) estrutura de síntese

Vetterli também propuseram em 2001 uma modificação no DBF [64].

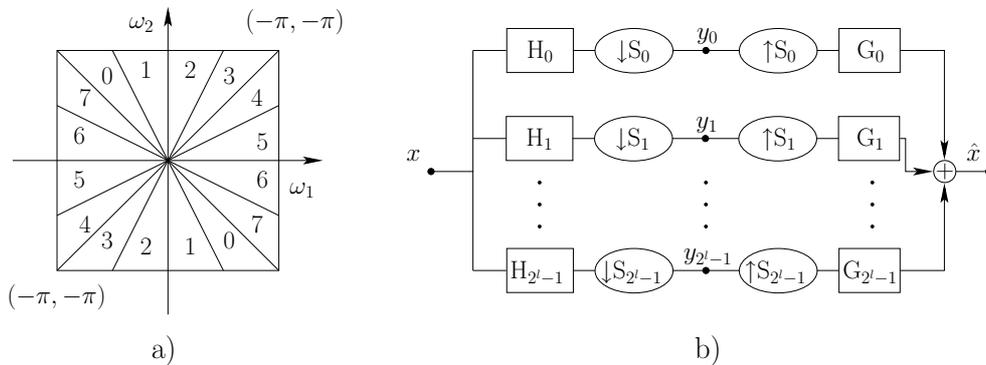


Figura 4.6: a) partição freqüencial da promovida pelo DFB onde $l = 3$. Assim, existem $2^3 = 8$ bandas as quais são divididas em bandas mais próximas da horizontal (0–3) e mais próximas da vertical (4–7); b) visão multicanal da estrutura em árvore do DBF.

Assim, a estrutura final do PDFB pode ser ilustrada como na Figura 4.7.

Note que a medida que se passa de uma oitava para outra, o número de direções do banco de filtro dobra de modo a satisfazer a condição de anisotropia $largura \approx comprimento^2$ originalmente estabelecida pelas curvelets.

Para maior detalhes, veja [17, 46].

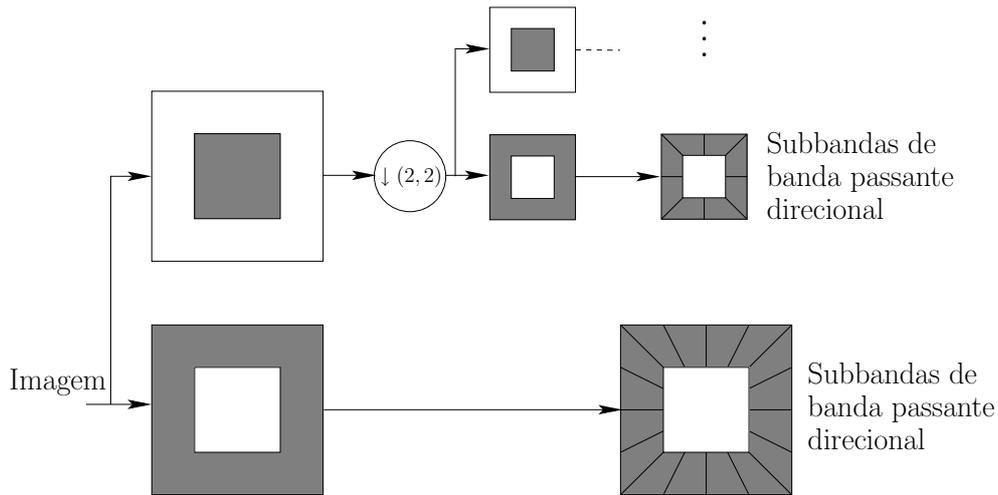


Figura 4.7: Estrutura do banco de filtro piramidal direcional.

4.4 Nossa proposição

Recapitulando, o contexto em que esta tese se encontra é o da compressão de vídeo. Em particular procuramos desenvolver bons dicionários para serem usados no codificador de vídeo baseado no método de decomposição de sinais Matching Pursuits com Planos de Bits generalizados, desenvolvido previamente por Caetano e Eduardo [10, 40].

Particularmente, encontramos uma semelhança muito forte entre as curvelets e o contexto onde estamos inseridos. Note que o primeiro passo das curvelets tem como objetivo realçar os contornos para que possam ser melhor representados pela transformada multidirecional. Como segundo passo importante, os contornos realçados são particionados em “quadrados” anisotrópicos ($largura \approx comprimento^2$), os quais são pequeno o bastante para que cada “quadrado” que contenha alguma energia presente, de modo geral, trechos retilíneos do contorno. Como passo final, aplica-se as ridgelets, as quais são especialistas em representar discontinuidades 1-D retilíneas.

No nosso contexto, o passo de estimação/compensação de movimento realiza a “mesma” tarefa da decomposição em subbandas, no sentido de que a imagem estimada/compensada (dfd) também é composta por contornos. O segundo passo das curvelets (localização espaço-freqüencial) também é implementado pelo algoritmo MPGBP, que de modo a realizar uma busca pelos melhores encaixes de maneira mais eficiente, divide a imagem dfd em quadrados, macroblocos, e localiza os elementos

do dicionário pixel a pixel no seu interior. A diferença é que o MPGBP proporciona uma localização muito mais apurada, uma vez que os átomos do dicionário são posicionados pixel a pixel dentro do macrobloco. Contudo, a realização da anisotropia é deixada a cargo do dicionário utilizado.

Assim, a nossa proposta consiste em utilizar as ridgelets digitais, mas não diretamente da forma como foram apresentadas nas Equações (4.12) e (4.13). Como vimos anteriormente, as versões digitais das ridgelets foram construídas a partir da versão ortonormal contínua (Equação (4.7)). Como discutimos no decorrer da Seção 3.2, o uso de dicionários \mathcal{C} ortonormais não nos é interessante, pois em termos de $\Theta(\mathcal{C})$ apresentam mesmo desempenho de qualquer outro dicionário ortonormal. Propomos, então, criação de ridgelets redundantes de modo a melhor satisfazer a condição (i) – pequenos valores de $\Theta(\mathcal{C})$ e $q(\mathcal{C})$. Além disso, outra modificação importante que propomos é a anisotropificação das ridgelets digitais, as quais possuem originalmente região de suporte quadrada, como definidas nas Equações (4.12) e (4.13). Com isso, esperamos adaptar as ridgelets ainda mais ao formato dos trechos dos contornos de modo a satisfazer ainda mais eficientemente a condição (ii) – átomos os mais parecidos possível com as regiões da imagem. Como terceira modificação, propomos a utilização de outras funções, além das wavelets, na composição das nossas ridgelets, concedendo, desta maneira, mais graus de liberdade na sua construção.

No próximo capítulo apresentaremos a implementação destas modificações e os resultados alcançados.

Capítulo 5

Dicionários para o MPGBP

Como visto na Seção 3.2, procuramos por dicionários cujos valores de $\Theta(\mathcal{C})$ e de $q(\mathcal{C})$ sejam os menores possíveis. Uma vez obtido dicionários com um bom compromisso $\Theta(\mathcal{C})$ - $q(\mathcal{C})$ deve-se levar em conta que qualquer dicionário \mathcal{C}' obtido por rotação dos elementos de \mathcal{C} , encarados como vetores em \mathbb{R}^N , possui $\Theta(\mathcal{C}') = \Theta(\mathcal{C})$, o que, através do Teorema 3.1, equivaleria dizer que ele teria o mesmo desempenho do dicionário \mathcal{C} quando utilizado no codificador MPGBP. Contudo, durante a primeira iteração do algoritmo MPGBP, a redução do erro de aproximação só depende das formas de onda dos átomos do dicionário. Quanto mais elas forem parecidas com o sinal \mathbf{x} , maior será o valor do produto interno entre eles e, por conseguinte, menor será o valor de $\|\mathbf{x} - \alpha^{k_m} \mathbf{v}_{r_m}\|$. Desta forma, dentre todos os dicionários com os mesmos valores de $\Theta(\mathcal{C})$ e $q(\mathcal{C})$, deve-se obter aquele cujas características sejam as mais próximas daquelas do sinal em questão. Com isso, vimos que um bom dicionário deve ter:

- (i) pequenos valores de $\Theta(\mathcal{C})$ e $q(\mathcal{C})$;
- (ii) átomos os mais parecidos possível com o sinal.

No contexto de codificação de vídeo, a imagem alvo é representada pelo quadro diferença (dfd) originado por um processo de estimação/compensação de movimento. O dfd é essencialmente composto por contornos de diversas larguras e orientados nas mais variadas direções. Da análise levantada no capítulo anterior, vemos, então, que átomos baseados nas funções ridgelet são bons candidatos para satisfazer a condição (ii) enunciada acima.

No entanto, para que a condição (i) seja melhor atendida devemos desconsiderar o uso da Transformada Ridgelet Ortonormal. Isto porque, sendo uma transformada ortonormal uma mera rotação de qualquer outra transformada ortonormal, como a Transformada Wavelet ou a DCT, por exemplo, todas as transformadas ortonormais de mesma dimensão têm um mesmo valor de $\Theta(\mathcal{C})$. Por isso, se faz necessário o uso de dicionários ridgelets supercompletos para a redução dos valores de $\Theta(\mathcal{C})$. Observe que este procedimento é retratado através da Figura 3.3 (aqui repetida como Figura 5.1 para maior conveniência), sendo o equivalente à adição de novos átomos visando o preenchimento dos “buracos” no dicionário.

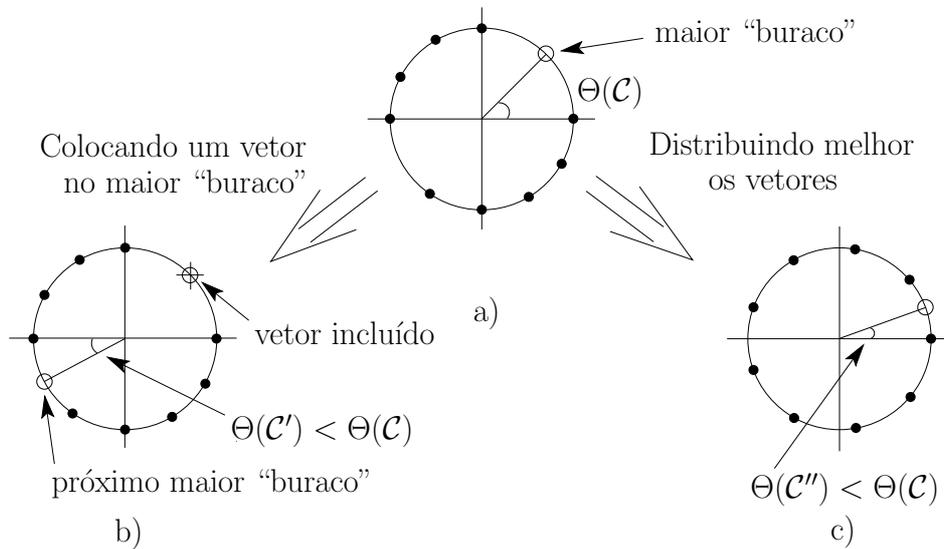


Figura 5.1: $\Theta(\mathcal{C})$ em a) pode ser reduzido: b) adicionando um vetor extra no maior “buraco” do dicionário \mathcal{C} no \mathbb{R}^2 ; c) distribuindo melhor os vetores em \mathbb{R}^2 .

De modo a transformar a ridgelet ortonormal em uma ridgelet supercompleta utilizamos um número maior de escalas na construção dos átomos, quebrando a propriedade de ortonormalidade desta transformada multiresolucional. Além disso, a utilização das ridgelets em si significa a introdução de átomos multidirecionais, estabelecendo um novo grau de liberdade, sobretudo se comparado a dicionários separáveis, caso muito freqüente nos algoritmos de codificação de imagens e vídeo (vide, por exemplo, o EZW e o SPIHT, que usam wavelets 2-D separáveis e mesmo o codificador MP, que usa bases 2-D separáveis de Gabor).

Ademais, de forma a satisfazer mais eficientemente à condição (ii) uma outra importante alteração deve ser feita nas ridgelets digitais. Dado um fragmento reto

de um contorno, pode-se entendê-lo como uma função do \mathbb{R}^2 de região de suporte retangular. Assim, para que as ridgelets possuam um melhor casamento com estes segmentos, devemos converter sua região de suporte tipicamente quadrada [18] em uma região de suporte retangular. Em outras palavras, devemos torná-las anisotrópicas. Isto é feito “encolhendo” ou “esticando” uma ridgelet $\rho(u, v)$ nas duas direções v e u , criando uma função $g(u, v) = \rho(au, bv)$, com $a \neq b \neq 0$.

Assim, visando a generalização do conceito das ridgelets digitais, definamos, a partir das Equações (4.12) e (4.13), as *ridgelets digitais infinitas*, cuja *ridge* possui um ângulo $\varpi \in [0, \pi)$ com o eixo horizontal de acordo com:

$$\rho_{\gamma, c, \varpi}^{(f)}(u, v) = f_{\gamma}(u \cdot \tan(\varpi) + v) \cdot W_c(u \cdot \tan(\varpi) + v) \quad (5.1)$$

onde $u, v \in \mathbb{Z}$ e $-\infty \leq u, v < \infty$ e γ representa um vetor de parâmetros de uma função f oscilatória e multiresolucional qualquer, muito embora tradicionalmente seja uma função wavelet: $f_{\gamma}(n) = f_{j,k}(n) = 2^{\frac{j}{2}}\psi(2^j n - k)$, onde $j, k \in \mathbb{R}$, com j representando a escala e k a translação. W_c é uma janela 1-D retangular e unitária de comprimento c , na direção da *ridge* definida como:

$$W_c(v) = \begin{cases} 1, & v \in [-\frac{c-1}{2}, \frac{c-1}{2}] \\ 0, & v \notin [-\frac{c-1}{2}, \frac{c-1}{2}] \end{cases} \quad (5.2)$$

Note que para limitar a extensão desta ridgelet infinita necessitamos de um outro janelamento na direção ortogonal à *ridge*. Logo, podemos observar que as ridgelets digitais definidas anteriormente na Seção 4.1 são um caso particular desta nossa definição, onde este outro janelamento seria realizado novamente com a janela W_c .

No entanto, como o interesse por trás desta nossa perspectiva “infinita” das ridgelets digitais é a introdução da anisotropia, o janelamento deve ser realizado por uma outra janela qualquer de comprimento c' . De modo a realizar um janelamento suave optamos, então, pela utilização de uma outra função ridgelet infinita ao invés da janela W_c , gerando, assim, um conjunto de ridgelets anisotrópicas e supercompletas dadas por:

$$g_{\gamma_1, \gamma_2, c_1, c_2, \varpi}^{(f_1, f_2)}(u, v) = \rho_{\gamma_1, c_1, \varpi}^{(f_1)}(u, v) \cdot \rho_{\gamma_2, c_2, \frac{\pi}{2} - \varpi}^{(f_2)}(u, v) \quad (5.3)$$

Observe que os átomos gerados de acordo com a Equação (5.3) têm uma região de suporte de área retangular finita de dimensões $c_1 \times c_2$. Isto se deve ao fato de cada ridgelet infinita ter comprimento finito na direção ortogonal à sua ridge devido ao janelamento por W_c (ver Equações (5.1) e (5.2)).

É importante notar que os átomos gerados pela Equação (5.3) são, em geral, não-separáveis. O uso de dicionários não-separáveis pode acarretar em um grande crescimento da complexidade computacional do codificador. Entretanto, vários métodos de aceleração do processo de busca dos átomos foram propostos na literatura. Um bom exemplo de tais métodos é demonstrado em [65].

Ainda dentro da visão de generalização das ridgelets, dada qualquer função 1-D podemos gerar átomos segundo a Equação (5.3), nos dando mais graus de liberdade em relação às ridgelets digitais originais. Assim, de modo a tirar proveito deste outro grau de liberdade, nós testamos uma outra categoria de funções que, além de serem formadas por translações e escalonamentos, usam também modulações de uma mesma função mãe, a gaussiana $g(t) = \sqrt[4]{2}e^{-\pi t^2}$. Estas funções são as funções de Gabor, definidas na Equação (2.14) como:

$$g_{s,\xi,\phi}(i) = K_{s,\xi,\phi} \cdot g\left(\frac{i - \frac{N-1}{2}}{s}\right) \cdot \cos\left(\frac{2\pi\xi(i - \frac{N-1}{2})}{N} + \phi\right) \quad (5.4)$$

Assim, nesta Tese, f é representado tanto pelas wavelets e scaling functions de Meyer [18] quanto pelas funções de Gabor, onde $\gamma = (s, \xi, \phi)$ (ver Tabela 2.1). A wavelet de Meyer, como definida na Equação (4.8), é dada no domínio freqüencial como:

$$\hat{\psi}(\omega) = \begin{cases} \frac{1}{\sqrt{2\pi}} e^{i\omega/2} \text{sen}\left(\frac{\pi}{2}\nu\left(\frac{3}{2\pi}|\omega| - 1\right)\right), & \frac{2\pi}{3} \leq |\omega| \leq \frac{4\pi}{3} \\ \frac{1}{\sqrt{2\pi}} e^{i\omega/2} \text{cos}\left(\frac{\pi}{2}\nu\left(\frac{3}{4\pi}|\omega| - 1\right)\right), & \frac{4\pi}{3} \leq |\omega| \leq \frac{8\pi}{3} \\ 0, & |\omega| \notin \left[\frac{2\pi}{3}, \frac{8\pi}{3}\right] \end{cases} \quad (5.5)$$

e sua scaling function como:

$$\hat{\phi}(\omega) = \begin{cases} \frac{1}{\sqrt{2\pi}}, & |\omega| \leq \frac{2\pi}{3} \\ \frac{1}{\sqrt{2\pi}} \text{cos}\left(\frac{\pi}{2}\nu\left(\frac{3}{2\pi}|\omega| - 1\right)\right), & \frac{2\pi}{3} \leq |\omega| \leq \frac{4\pi}{3} \\ 0, & |\omega| \geq \frac{4\pi}{3} \end{cases} \quad (5.6)$$

onde $\nu(a) = a^4(35 - 84a + 70a^2 - 20a^3)$, $a \in [0, 1]$. De forma a utilizar estas funções no domínio espacial fizemos o uso da Transformada Rápida Inversa de Fourier

(*Inverse Fast Fourier Transform* – IFFT) através da função *instdfft* do programa MATLAB® [66].

De forma a avaliar o desempenho destas novas ridgelets digitais, supercompletas e anisotrópicas, adaptamos seu uso ao codificador de vídeo MPGBP, descrito na Seção 3.3, compondo dicionários supercompletos como especificado na Seção 5.2. Os resultados alcançados foram submetidos a comparações com os resultados alcançados pelo dicionário separável utilizado por Neff e Zakhor em [6], descrito na Subseção 2.2.1.

Entretanto, algumas mudanças foram realizadas no algoritmo MPGBP de [10, 39], de modo a facilitar sua implementação com os novos dicionários assim como permitir uma comparação justa entre os resultados alcançados com os nossos dicionários e com o dicionário de Neff e Zakhor. Estas mudanças foram propostas em [40] e estão descritas na próxima seção.

5.1 Codificador Utilizado

O codificador aqui utilizado é, em grande parte, o mesmo descrito em [10, 39]. A única diferença entre ele e o usado neste trabalho é a inserção de um mecanismo de controle de taxa que, além de ser simples, provê um controle de taxa preciso. Sua razão de ser está ligada ao fato de que uma comparação justa entre dicionários requer a utilização de rigorosamente as mesmas taxas reais.

Essencialmente este mecanismo calcula a taxa gasta por cada novo átomo codificado e interrompe a adição de átomos quando a taxa de bits desejada é alcançada. Note que adicionar um átomo é equivalente a adicionar um par de índices (k_m, r_m) . O átomo indexado é definido tanto por sua forma (dimensões, escala, rotação, etc) quanto por sua posição. Sua forma, assim como o expoente k_m , são codificados exatamente como em [10, 39]. Já a sua posição dentro de um macrobloco 16×16 é codificada diferencialmente (ver [6, 10]). Assim, para que haja um cálculo preciso da taxa de bits devemos recalcular, a cada átomo codificado adicionado, a taxa de bits gasta para codificar a posição diferencial de todos os átomos codificados no macrobloco. Para maiores detalhes, consultar [40].

Apenas uma simples adaptação adicional foi realizada devido à mudança da

região de suporte dos átomos do dicionário. Isto porque, com a rotação dos átomos, passamos a ter uma região de alterada, como ilustrado na Figura 5.2.

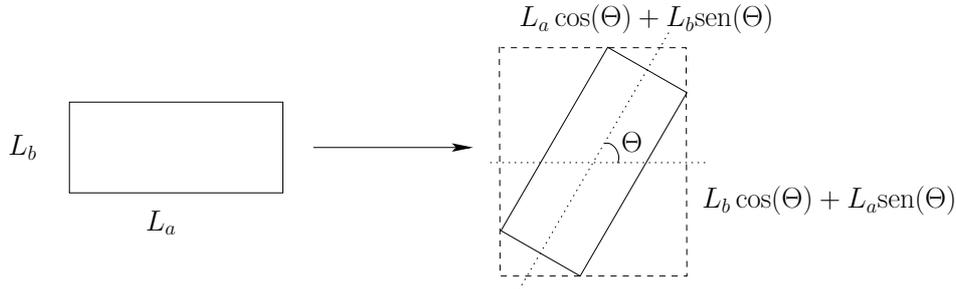


Figura 5.2: Alteração na região de suporte dos átomos do dicionário.

Para maiores detalhes, ver [40].

5.2 Resultados Experimentais

O leitor deve ter em mente que o foco desta tese está em demonstrar que o desempenho da decomposição de sinais MPGBP pode ser melhorado com o uso de dicionários adequados. No Capítulo 3 vimos a descrição deste método de representação de sinais assim como os fatores que são importantes para o seu bom desempenho, de forma que concluímos que o dicionário é uma peça de vital importância para o seu bom desempenho.

No entanto, naquela ocasião chegamos apenas à conclusão de que é possível termos bons dicionários. Levantamos teoricamente duas características que os dicionários devem possuir para promoverem bons desempenhos para o algoritmo MPGBP, mas quantitativamente não sabemos de ante-mão que resultados podem ser alcançados na prática.

Assim, de forma a revelar resultados práticos para toda esta teoria utilizamos novamente o contexto de compressão de vídeo, pelo fato de já apresentar resultados experimentais do algoritmo MPGBP com o uso de outro dicionário que não os baseados em ridgelets [10, 40]. Este outro dicionário é o dicionário 2-D separável construído a partir das funções 1-D de Gabor, descritas no Capítulo 2. Vimos, neste mesmo capítulo, que Mallat e Zhang [9] propuseram o uso destas funções de forma a dar uma maior eficiência à decomposição matching pursuits.

De fato, a aplicação da decomposição MP em vídeo, implementada por Neff e Zakhor [6] revelou o dicionário baseado nas funções de Gabor como um dicionário de bom desempenho, levando o codificador de vídeo MP a superar o desempenho dos compressores de vídeo padrão, baseados na DCT [6].

Assim, como a implementação do algoritmo MPGBP em vídeo utilizou o mesmo dicionário de Neff e Zakhor, nosso contexto de avaliação é dado por comparações entre os resultados gerados com o uso do algoritmo MPGBP descrito no Capítulo 3 (juntamente com a alteração descrita na seção anterior) utilizando os dicionários aqui desenvolvidos e este dicionário separável 2-D de Gabor. Estas comparações se dão numericamente através de 3 parâmetros quantitativos distintos: o valor de PSNR, o valor de $\Theta(\mathcal{C})$ e o valor $\bar{\Theta}(\mathcal{C})$.

O valor de PSNR (Peak Signal to Noise Ratio) é obtido segundo uma média aritmética dos valores de PSNR alcançados após a codificação de cada quadro da seqüência. Este parâmetro indica a razão entre o valor de pico da energia do quadro aproximado e a energia do ruído de aproximação ou erro de aproximação na escala logarítmica (em decibel – dB). Assim, quanto maior o valor de PSNR mais próxima é a imagem aproximada da imagem original, indicando uma melhor representação. Seu valor é dado por:

$$\text{PSNR} = \frac{1}{T} \sum_{i=1}^T \text{PSNR}_i = \frac{1}{T} \sum_{i=1}^T 10 \log_{10} \frac{255^2}{\text{MSE}_i} \quad (5.7)$$

onde T é o número total de quadros $M \times N$ codificados e MSE é o erro médio quadrático dado por:

$$\text{MSE} = \frac{\|\hat{\mathbf{x}} - \mathbf{x}\|^2}{M \cdot N} \quad (5.8)$$

O valor de $\Theta(\mathcal{C})$ representa o ângulo máximo da projeção (produto interno) do vetor $\mathbf{y} \in \mathbb{R}^N$ no átomo \mathbf{v}_j do dicionário \mathcal{C} mais próximo. Como visto no Capítulo 3, quanto menor seu valor melhor é o desempenho do dicionário. Seu valor é dado por:

$$\Theta(\mathcal{C}) = \max_{\mathbf{y} \in \mathbb{R}^N} \left\{ \min_{\mathbf{v}_j \in \mathcal{C}} \left[\arccos \frac{\langle \mathbf{y}, \mathbf{v}_j \rangle}{\|\mathbf{y}\| \cdot \|\mathbf{v}_j\|} = \arccos \frac{\langle \mathbf{y}, \mathbf{v}_j \rangle}{\|\mathbf{y}\|} \right] \right\} \quad (5.9)$$

Já o valor de $\bar{\Theta}(\mathcal{C})$ não é uma mera média aritmética dos valores de θ de cada átomo. Na verdade nós definimos para cada passo i um valor de $\beta^{(i)}$ tal que o módulo do resíduo no passo $i + 1$, $\|r^{(i+1)}\|$, seja igual a $\beta^{(i)} \|r^{(i)}\|$. Com isso, também definimos $\bar{\beta}$ como sendo o valor médio de todos os $\beta^{(i)}$ da seqüência completa de

quadros, ou seja:

$$\bar{\beta} = \frac{1}{N_{\beta}} \sum_{i=1}^{N_{\beta}} \beta^{(i)} \quad (5.10)$$

onde N_{β} representa o número total de $\beta^{(i)}$ para a toda a seqüência de quadros codificados. Dessa forma, o valor de $\bar{\Theta}(\mathcal{C})$ é dado como uma derivação direta do Teorema 3.1, ou seja:

$$\bar{\Theta} = \arccos \left(\sqrt{\frac{1 - \bar{\beta}^2}{2\alpha - \alpha^2}} \right) \quad (5.11)$$

Como é desejado o máximo de redução dos resíduos a cada passo do algoritmo, é necessário que $\bar{\beta}$ seja o menor possível. Tal fato implica que bons dicionários devem ter valores de $\bar{\Theta}(\mathcal{C})$ os menores possíveis.

Portanto, definidos os parâmetros de avaliação de desempenho dos dicionários no contexto do codificador de vídeo baseado no algoritmo MPGBP, retomemos nosso contexto de avaliação. Como explicado anteriormente, as comparações dos nossos dicionários são realizadas diretamente com o dicionário separável 2-D de Gabor utilizado por Neff e Zakhor [6]. Mas, para que as comparações tenham alguma validade, é necessário que estes dicionários sejam utilizados exatamente nas mesmas condições de simulação.

Assim, nesta tese, todos os dicionários utilizados são aplicados e comparados utilizando-se as mesmas seqüências de imagem, mesma taxa de compressão, mesma resolução de quadros e mesma taxa de quadros por segundo (fps). No entanto, para que os resultados possam ser representativos, utilizamos um número razoável de diferentes tipos de imagens e taxas de codificação. As seqüências de imagens utilizadas nas simulações, a exemplo das utilizadas no codificador MP e no MPGBP descritos anteriormente, são todas do tipo QCIF, cuja resolução é 176×144 para a componente de luminância Y e 88×72 para as duas componentes de crominância Cb e Cr. Contudo, como previsto no início do Capítulo 2, consideraremos apenas a componente de luminância, de modo que a seqüência original é composta por 300 quadros de resolução 176×144 e com 8bits/pixel cada, amostrados a uma taxa de 30fps, resultando em uma taxa de aproximadamente 6082kbps (kilobits por segundo). Foram utilizadas 7 tipos diferentes de seqüências de imagens, quais sejam: *coast-guard*, *container*, *foreman*, *hall-monitor*, *mother-and-daughter*, *silent-voice* e *weather*. Alguns dos quadros destas seqüências são exibidos no Apêndice A.

As taxas de quadros por segundo e de bits por segundo utilizadas na codificação das seqüências estão relacionadas entre si da seguinte maneira: nas simulações que envolveram taxas de até 20kbps houve subamostragem para uma taxa de 7,5fps (equivalendo a codificação total de 75 quadros); nas simulações com taxas entre 20kbps e 100kbps, as seqüências foram subamostradas e codificadas a uma taxa de 10fps (equivalendo a codificação efetiva de 100 quadros); já nas simulações acima de 100kbps não houve subamostragem.

Acerca do valor do fator de aproximação de escala α utilizamos o mesmo valor $\alpha = 0,56$ usado anteriormente por Caetano e Eduardo [10, 40] ao implementar seu codificador de vídeo MPGBP com o dicionário de Neff e Zakhor [6]. Isto não se trata de uma escolha ao acaso, mas sim uma escolha fruto de alguns testes de simulação. De fato, realizamos os mesmos testes que Caetano e Eduardo para a escolha do valor 0,56, ou seja, para algumas seqüências de imagem e alguns dicionários variamos os valores de α de 0 a 1 e agrupamos valores de PSNR finais em gráficos de $PSNR \times \alpha$. Na Figura 5.3 apresentamos um exemplo destes gráficos para o dicionário GR descrito no decorrer desta seção.

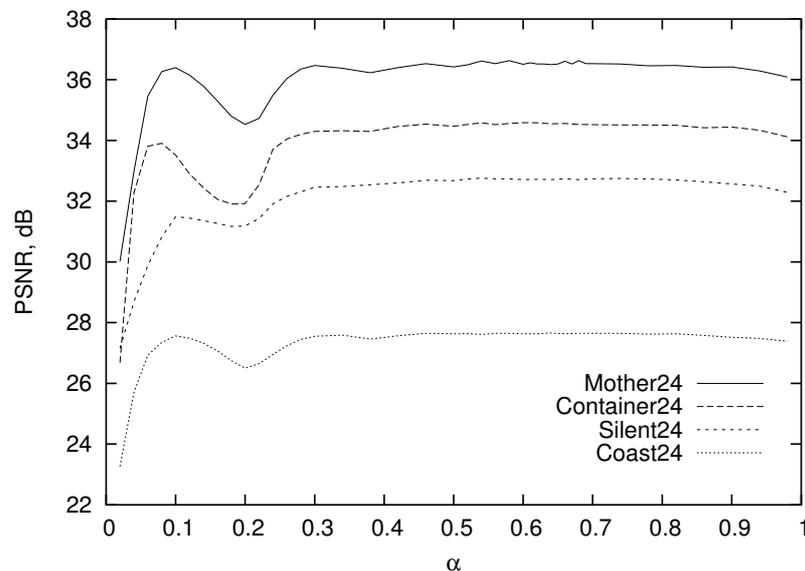


Figura 5.3: Variação do PSNR médio com o parâmetro α para as seqüências de imagem mother, container, silent e coast para a taxa de 24kbps para o dicionário GR.

O comportamento da curva $PSNR \times \alpha$ é semelhante para todos os dicionários usados, apresentando a mesma característica aproximadamente plana para

$\alpha \in [0,4; 0,85]$. Isto nos permitiu manter o mesmo valor de $\alpha = 0,56$.

Uma vez especificado toda a estrutura de simulação do codificador de vídeo MPGBP, especificaremos, no decorrer desta seção, que dicionários construímos e que valores de PSNR, $\Theta(\mathcal{C})$ e $\bar{\Theta}$ foram alcançados com sua implementação no algoritmo MPGBP.

Primeiramente nós usamos o dicionário empregado por Neff e Zakhor em [6] (ver Subseção 2.2.1), que daqui por diante será referido como dicionário NZ – Neff e Zakhor. Vale lembrar que este dicionário é supercompleto e seus átomos foram escolhidos através de um processo de treinamento que envolveu 5 seqüências de imagens, quais sejam: *akiyo*, *container*, *hall-monitor*, *mother-and-daughter* e *sean*, para taxas de 10kbps e 24kbps. Os resultados obtidos com o uso deste dicionário no algoritmo MPGBP modificado (ver Seção 5.1) são mostrados na Tabela 5.1 para maior conveniência. A cardinalidade do dicionário NZ é $q(\mathcal{C}) = 400$.

Com o intuito de diminuir os valores de $\Theta(\mathcal{C})$ apresentados pelo dicionário NZ de modo a satisfazer mais eficientemente a condição (i) mencionada no início deste capítulo, procuramos preencher os “buracos” deste dicionário NZ com a adição de novos vetores. No entanto, para de fato satisfazer a condição (i), esses novos vetores adicionados devem ser *quantitativa* e *qualitativamente* bem posicionados. Quantitativa significa que o número de átomos adicionados deve ser o menor possível para que a cardinalidade $q(\mathcal{C})$ não cresça desnecessariamente. E qualitativamente significa que os átomos a serem adicionados devem ser distribuídos o mais eficientemente possível no preenchimento dos “buracos”. Assim, nos propomos a tapar os “buracos” usando os dois métodos descritos através da Figura 5.1, ou seja, colocando novos átomos nos “buracos” e os colocando de forma bem distribuída.

Para a realização desta tarefa, acreditamos que as funções ridgelet são as funções certas para a realização desta tarefa por suas características demonstradas através da Figura 4.1. Contudo, o entendimento que nos levou à formulação das ridgelets proposta nesta Tese (ver Equação (5.3)) só foi atingido após um longo processo de análise bibliográfica e de testes e avaliações dos resultados de simulação.

Como primeiro passo deste processo, testamos as funções ridgelet desenvolvidas por Donoho e Flésia em [18] (ver Equações (4.12) e (4.13)). Sua utilização envolve escolhas a respeito do tamanho n da janela quadrada, da escala j e do

Tabela 5.1: Valores de PSNR, $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$ obtidos com o dicionário NZ aplicado ao codificador MPGBP.

Seq+Taxa (kbps)	NZ		
	PSNR (dB)	$\Theta(\mathcal{C})$	$\bar{\Theta}(\mathcal{C})$
Mother24	36,19	86,17°	82,29°
Weather24	31,75	87,52°	82,61°
Silent24	32,59	87,08°	80,19°
Hall24	36.41	87.28°	81.62°
Coast24	27.71	86.79°	81.45°
Container24	34,58	87,04°	83,17°
Mother64	40,38	87,19°	83,51°
Weather64	37,56	88,26°	83,60°
Silent64	37,72	87,81°	82,25°
Hall64	39.87	87.69°	83.01°
Coast64	31.33	87.32°	82.92°
Container64	37,83	87,70°	84,16°
Foreman64	33,45	86,80°	80,77°
Foreman96	35,52	87,07°	81,60°
Weather96	40,38	88,48°	84,00°

parâmetro de translação k . Para o parâmetro n fizemos escolhas parecidas com a do dicionário NZ, ou seja, escolhemos valores ímpares variando de 3 a 35. Para tanto, foi necessária uma simples modificação na região de suporte das ridgelets de Donoho e Flésia. Isto porque sua região de suporte prevê apenas dimensões pares, haja visto sua formulação, dada por $-\frac{n}{2} \leq u, v < \frac{n}{2}$. No seu lugar, utilizamos a seguinte formulação: $-\frac{n-1}{2} \leq u, v \leq \frac{n-1}{2}$, de forma a permitir valores ímpares para a região de suporte. Novamente a exemplo do dicionário NZ, escolhemos o valor do parâmetro k de modo a posicionar as funções ridgelet sempre passando pelo ponto central pois, como vimos no Capítulo 3, o algoritmo MPGBP simula a translação do átomo no seu processo de busca pelos melhores átomos. Assim, como usamos tanto as wavelets quanto as scaling functions de Meyer, k obedece a relação $k = 2^{j-1}$ para

wavelets (cuja simetria não é em $x = 0$, mas sim em $x = -2^{j-1}$) e $k = 0$ para as scaling functions (que são simétricas em $x = 0$). Por último, utilizamos os valores de escala j relacionados ao valor do parâmetro n como na Tabela 5.2. Esta escolha foi fruto de breves observações do comportamento destas ridgelets nos domínios espacial e freqüencial. Como resultado, obtivemos um dicionário, aqui chamado de RO (ridglet ortonormal), cujos valores de PSNR, $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$ obtidos com o algoritmo MPGBP são expostos na Tabela 5.3. Gerando 4865 novos átomos, este dicionário tem cardinalidade $q(\mathcal{C}) = 5265$.

Tabela 5.2: Comprimentos (n) e escalas (j) dos átomos utilizados na composição do dicionário RO.

n	j
3	1
5, 7 e 9	1 e 2
11, 13, 15, 17, 19 e 21	1, 2 e 3
23, 25 e 27	1, 2, 3 e 4
29, 31, 33 e 35	1, 2, 3, 4 e 5

Tabela 5.3: Valores de PSNR, $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$ obtidos com o dicionário RO aplicado ao codificador MPGBP.

Seq+Taxa (kbps)	RO			RO – NZ	
	PSNR (dB)	$\Theta(\mathcal{C})$	$\bar{\Theta}(\mathcal{C})$	Δ PSNR (dB)	$\Delta\bar{\Theta}(\mathcal{C})$
Mother24	36,18	85,66°	81,44°	-0,01	-0,85°
Weather24	31,32	86,64°	81,99°	-0,43	-0,62°
Silent24	32,39	85,91°	79,42°	-0,20	-0,77°

Note que, neste caso o aumento da cardinalidade não refletiu na redução satisfatória dos valores de $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$, tendo como consequência um decréscimo dos valores de PSNR, denunciando a realização de um mal compromisso entre quantidade e qualidade dos átomos adicionado. Vale ressaltar ainda que a queda sofrida pelos

valores de PSNR mostra que a ortonormalidade presente em 88,96% dos átomos deste dicionário leva o seu desempenho de volta, comparativamente, à marca do desempenho alcançado pelo modelo H.263, que usa como dicionário as funções base da DCT, funções estas também ortonormais. Entendemos, então, que somente a introdução da característica de multidirecionamento não é suficiente para preencher adequadamente os “buracos”. Portanto, também é necessário tornar supercompleto o dicionário ortonormal das ridgelets. Para tanto, populamos as ridgelets ortonormais com o uso de escalas j intermediárias.

Assim, a escolha das novas escalas se deu como descrito na Tabela 5.4. Como resultado, obtivemos um dicionário, aqui chamado de RS (ridglet supercompleta), cujos valores de PSNR, $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$ obtidos com o algoritmo MPGBP estão na Tabela 5.5. Gerando 7512 novos átomos, este dicionário tem cardinalidade $q(\mathcal{C}) = 7912$.

Tabela 5.4: Comprimentos (n) e escalas (j) dos átomos utilizados na composição do dicionário RS.

n	j
3	1
5, 7 e 9	1; 1,5 e 2
11, 13, 15, 17, 19 e 21	1; 1,5; 2; 2,5 e 3
23, 25, 27 29, 31, 33 e 35	1; 1,5; 2; 2,5; 3; 3,5 e 4

Tabela 5.5: PSNR, $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$ obtidos com o dicionário RS aplicado ao codificador MPGBP.

Seq+Taxa (kbps)	RS			RS – NZ	
	PSNR (dB)	$\Theta(\mathcal{C})$	$\bar{\Theta}(\mathcal{C})$	Δ PSNR (dB)	$\Delta\bar{\Theta}(\mathcal{C})$
Mother24	36,33	85,36°	81,28°	0,14	-1,01°
Weather24	31,47	86,75°	81,82°	-0,28	-0,79°
Silent24	32,48	85,90°	79,29°	-0,11	-0,9°

Estes resultados mostram uma melhora com relação aos resultados alcançados

pelo dicionário RO, ratificando nossas expectativas. Observe que a redução dos valores de $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$ é mais expressiva, de modo que, mesmo com um aumento ainda maior na cardinalidade, os valores de PSNR são mais elevados. Assim, passamos a satisfazer a condição (i) mais convenientemente. Contudo, para algumas seqüências, a performance deste dicionário deixou a desejar.

À procura de uma justificativa para este fato, observamos que a condição (ii) não estava sendo satisfeita adequadamente. Apesar de estarmos usando átomos multidirecionais e multiescalares, a região de suporte quadrada limita os casamentos possíveis, pois de um modo geral os contornos apresentam região de suporte retangulares, anisotrópicas. Assim, surgiu a necessidade de adicionarmos átomos que copiassem essa característica dos contornos do quadro diferença, a exemplo dos átomos do dicionário NZ.

A partir de então, definimos a ridgelet digital infinita supercompleta como uma precursora da ridgelet digital anisotrópica supercompleta, definida matematicamente como na Equação (5.3). Com isso passamos a janelar as ridgelets supercompletas com outras ridgelets supercompletas segundo a Equação (5.3). Contudo, realizar esse janelamento para todas as ridgelets definidas como na Tabela 5.5 geraria um dicionário de cardinalidade extraordinariamente elevada. De forma a não permitir um descontrole sobre a cardinalidade do dicionário, tivemos que usar poucos valores de escala j e comprimento c das ridgelets. Estes valores estão expostos na Tabela 5.6. Eles foram baseados nas escolhas de Neff e Zakhor para a construção dos seus átomos de Gabor.

Já os ângulos de rotação ϖ de cada átomo de dimensões c_1 e c_2 foram escolhidos de acordo com o critério sugerido por Donoho e Flesia em [18], sendo dados por:

$$\varpi_b = \arctan\left(\frac{2b}{L-1}\right), \quad -\frac{L-1}{2} \leq b \leq \frac{L-1}{2} \quad (5.12)$$

onde $L = \max\{c_1, c_2\}$.

Assim, utilizando as wavelets e scaling functions de Meyer para figurarem tanto com f_{j_1, k_1} quanto como f_{j_2, k_2} , geramos um dicionário, aqui chamado de MEYER, que adicionou 24217 novos átomos ao dicionário NZ, resultando em uma cardinalidade $q(\mathcal{C}) = 24617$ vetores, a qual é cerca de 60 vezes a cardinalidade do dicionário NZ. Os resultados alcançados por este dicionário pode ser visto na Tabela 5.7

Tabela 5.6: Comprimentos (c) e escalas (j) dos átomos utilizados na composição do dicionário MEYER.

c	j
1	0
3	1
5	2
7	1,5
7	2
9	2
9	2,5
11	3
13	3
15	3
21	3
21	3,5
23	4
27	4
29	4
35	4,5
35	5

Pode-se ver que atingimos uma redução nos valores de $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$ em todos os casos, acompanhado por um crescimento no valor de PSNR, quando comparamos com os resultados obtidos com o dicionário NZ. É importante observar que esta melhora foi atingida em quase todas as seqüências e taxas, mesmo tendo ocorrido um aumento bastante significativo na cardinalidade $q(\mathcal{C})$. Isto confirma a conjectura, baseada no Teorema 3.1, a qual afirma que é possível se obter bons dicionários para o algoritmo MPGBP diminuindo-se $\Theta(\mathcal{C})$ a despeito do custo do aumento da cardinalidade. Isto é, o aumento de taxa produzido pelo aumento em $q(\mathcal{C})$ foi compensado pela diminuição da distorção causada pela redução em $\Theta(\mathcal{C})$. No entanto, como não utilizamos nenhuma técnica de treinamento, os resultados apresentados

Tabela 5.7: PSNR, $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$ obtidos com o dicionário MEYER aplicado ao codificador MPGBP.

Seq+Taxa (kbps)	MEYER			MEYER – NZ	
	PSNR (dB)	$\Theta(\mathcal{C})$	$\bar{\Theta}(\mathcal{C})$	Δ PSNR (dB)	$\Delta\bar{\Theta}(\mathcal{C})$
Mother24	36,46	85,36°	80,81°	0,27	-1,48°
Weather24	31,85	86,46°	81,23°	0,10	-1,38°
Silent24	32,63	85,86°	78,64°	0,04	-1,55°
Mother64	40,51	86,12°	82,25°	0,13	-1,26°
Weather64	37,80	86,91°	82,37°	0,24	-1,23°
Silent64	37,65	86,56°	80,98°	-0,07	-1,27°
Foreman64	33,83	85,84°	78,97°	0,38	-1,73°

poderiam ser melhores se não tivéssemos uma cardinalidade tão elevada.

Por último, como relatado anteriormente, utilizamos as funções 1-D de Gabor na Equação (5.3), de forma a compor novos átomos a ser utilizados no preenchimento dos “buracos” do dicionário NZ. As funções que figuraram no papel das funções f_{γ_1} e f_{γ_2} da Equação (5.3) foram as 20 funções de Gabor definidas como na Tabela 2.1. No tocante às rotações, escolhemos os valores de ϖ de duas maneiras distintas para efeito de teste. Uma é a aquela adotada para a composição do dicionário MEYER acima descrito, a qual toma valores de ϖ a partir das dimensões iniciais do átomo segundo a Equação 5.12, de modo quanto maior a região de suporte inicial do átomo, mais ângulos são tomados para a realização das rotações. Da outra maneira, escolhemos valores de ϖ igualmente espaçados de $2,5^\circ$, independente das dimensões iniciais do átomo. Assim, construímos 2 dicionários diferentes, o primeiro (*Dic1*) com 8090 átomos e o último (*Dic2*) com 14764 átomos. Alguns testes preliminares mostraram seus desempenhos em termos de *PSNR*, $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$ como na Tabela 5.8.

Note que a quantidade de átomos a mais que o *Dic2* possui não provoca, em relação ao dicionário *Dic1*, uma redução suficientemente grande de $\Theta(\mathcal{C})$ para causar um aumento consistente do valor de *PSNR*. Assim, este dicionário *Dic1* de cardinalidade $q(\mathcal{C}) = 8090$ (cerca de 20 vezes a do NZ), agora chamado de dicionário GR (Gabor Ridgelet), foi escolhido para realizar outros testes de simulação. Os

Tabela 5.8: PSNR, $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$ obtidos com os dicionários *Dic1* e *Dic2* aplicados ao codificador MPGBP.

Seq+Taxa (kbps)	<i>Dic1</i>			<i>Dic2</i>		
	PSNR (dB)	$\Theta(\mathcal{C})$	$\bar{\Theta}$	PSNR (dB)	$\Theta(\mathcal{C})$	$\bar{\Theta}$
Mother24	36,53	85,69°	81,03°	36,58	85,36°	80,91°
Weather24	32,25	86,62°	81,27°	32,20	86,47°	81,13°
Foreman64	33,93	85,94°	79,24°	33,86	86,06°	79,09°

resultados obtidos com outros testes por este dicionário estão expostos na Tabela 5.9.

Novamente nota-se que obtivemos um aumento substancial nos valores de PSNR para a maioria das seqüências e taxas juntamente com uma redução nos valores de $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$. De fato, o dicionário GR apresenta um desempenho melhor que o do dicionário MEYER. Entretanto tal fato não era inesperado, uma vez que as funções de Gabor em [6] foram geradas através de um processo de treinamento intencionando a obtenção de funções 2-D separáveis com características similares às características apresentadas pelas imagens dfds (isto é, elas foram projetadas especificamente para satisfazer a condição (ii)), enquanto que não houve qualquer treinamento na geração dos átomos adicionais que compõem o dicionário MEYER (por isso a diferença de cerca de 3 vezes entre suas cardinalidades).

Particularmente, neste caso criamos um outro dicionário NZ generalizado, pois para rotações de ângulo 0° reproduzimos o dicionário separável 2-D de Gabor. Portanto, podemos entender que temos um dicionário que é uma generalização dos dicionários 2-D separáveis, baseados em produtos externos de funções 1-D. Note que as mesmas funções 1-D de Gabor que Neff e Zakhor usou, nós também utilizamos, mas produzindo um ganho muito superior. Isso mostra que nossa maneira de usar as funções 1-D são muito mais eficientes, em termos de PSNR, do que anteriormente usado. Nesse caso vemos como a ferramenta do multidirecionamento é importante, pois NZ já utiliza valores diversos de escalas, já é anisotrópico e supercompleto. E, pensando em termos de treinamento, conjecturamos que resultados ainda mais expressivos podem ser alcançados com treinamentos nos ângulos utilizados.

Para uma melhor visualização do ganho de desempenho promovido pelo dicionário GR face ao dicionário NZ, expomos, por meio de gráficos comparativos,

Tabela 5.9: PSNR, $\Theta(\mathcal{C})$ e $\bar{\Theta}(\mathcal{C})$ obtidos com o dicionário GR aplicado ao codificador MPGBP.

Seq+Taxa (kbps)	GR			GR – NZ	
	PSNR (dB)	$\Theta(\mathcal{C})$	$\bar{\Theta}(\mathcal{C})$	Δ PSNR (dB)	$\Delta\bar{\Theta}(\mathcal{C})$
Mother24	36,53	85,69°	81,03°	0,34	-1,26°
Weather24	32,25	86,62°	81,27°	0,50	-1,34°
Silent24	32,74	86,29°	78,90°	0,15	-1,29°
Hall24	36,22	86,64°	80,68°	-0,19	-0,94°
Coast24	27,65	85,87°	80,51°	-0,06	-0,94°
Container24	34,52	85,94°	82,14°	-0,06	-1,03°
Mother64	40,72	86,51°	82,35°	0,34	-1,16°
Weather64	38,39	87,20°	82,34°	0,83	-1,26°
Silent64	37,96	86,92°	81,14°	0,24	-1,11°
Hall64	39,85	86,97°	82,13°	-0,02	-0,88°
Coast64	31,36	86,29°	82,01°	0,03	-0,91°
Container64	37,91	86,59°	83,11°	0,08	-1,05°
Foreman64	33,93	85,94°	79,25°	0,48	-1,52°
Foreman96	36,01	86,34°	80,30°	0,49	-1,30°
Weather96	41,41	87,30°	82,72°	1,03	-1,28°

alguns resultados de simulações obtidos através do uso do codificador MPGBP com ambos os dicionários. Um conjunto maior de taxas foi utilizado, indo desde taxas muito baixas (da ordem de 10kbps) até taxas altas (da ordem de 1000kbps).

Nos gráficos das Figuras 5.4 e 5.5 comparamos os valores de PSNR obtidos com a codificação dos vídeos Mother e Weather, respectivamente. A escolha destas seqüências em particular é devida aos abundantes resultados na literatura para as mesmas, comumente apresentando valores de PSNR para taxas compreendidas entre 10kbps e 100kbps. Note que o uso do dicionário GR garante ao algoritmo MPGBP uma melhora consistente de desempenho ao longo de toda esta faixa.

Como comparação final, codificamos a seqüência de imagens Foreman utilizando o codificador de vídeo MPGBP com os dicionários GR e NZ para taxas de

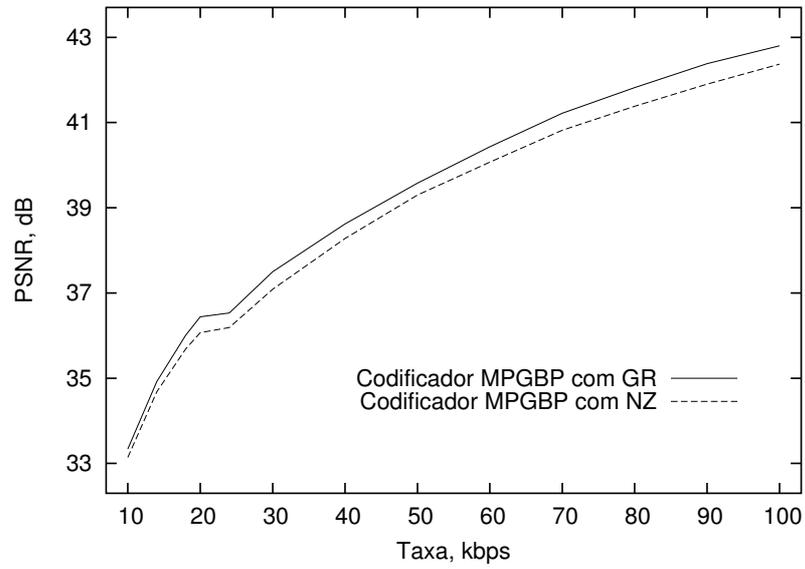


Figura 5.4: Variação do valor de PSNR em função da taxa utilizando o codificador de vídeo MPGBP com os dicionários GR e NZ para a seqüência Mother.

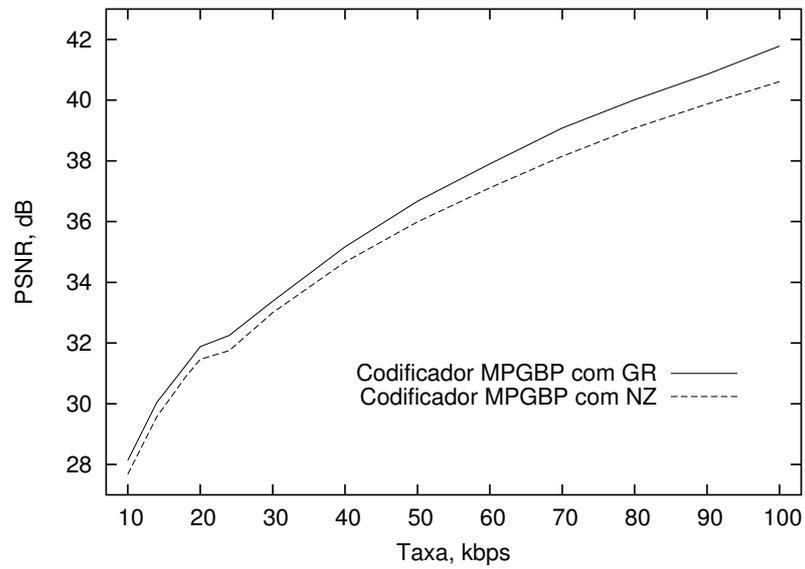


Figura 5.5: Variação do valor de PSNR em função da taxa utilizando o codificador de vídeo MPGBP com os dicionários GR e NZ para a seqüência Weather.

bits compreendidas entre 100kbps e 1047kbps. Os valores de PSNR obtidos estão expostos no gráfico comparativo da Figura 5.2. Esta comparação foi motivada por recentes resultados de pesquisa em codificadores de vídeo baseados no algoritmo MP [38] com o dicionário NZ, os quais também são expostos neste mesmo gráfico.

Observamos que mesmo em médias e altas taxas (100kbps – 1047kbps) o dicionário GR continua permitindo ao algoritmo MPGBP uma codificação mais eficiente dos quadros diferença, proporcionando um ganho de aproximadamente 1 dB de PSNR em toda a extensão desta banda.

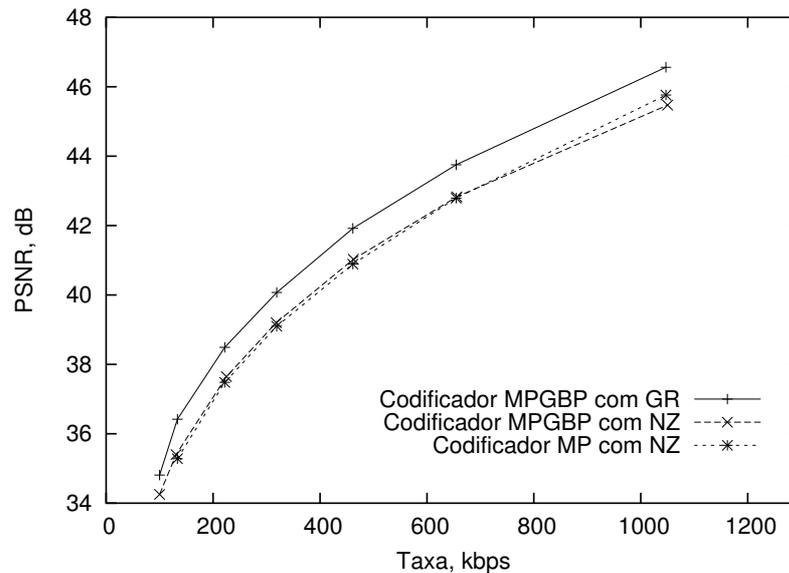


Figura 5.6: Variação do PSNR em função da taxa utilizando o codificador MPGBP com os dicionários GR e NZ e utilizando a versão mais recente do codificador MP [38] com o dicionário NZ para a seqüência Foreman.

Esta figura nos permite ainda observar que o codificador MPGBP possui desempenho comparável ao do estado da arte em codificação de vídeo utilizando o algoritmo MP [38].

A partir da observação destes gráficos podemos notar que ao passo que a taxa de compressão aumenta, o ganho proporcionado ao algoritmo MPGBP com o uso do dicionário GR torna-se maior com relação ao uso do dicionário NZ. Tal comportamento é mais visível para seqüências de imagens mais difíceis, ou seja, aquelas cuja compressão resulta em menores valores de PSNR, tais como a seqüência *foreman* e a *weather*. Esta alta no ganho se deve ao fato de um aumento na taxa de codificação implicar em um aumento no número de átomos utilizados na codificação da imagem

diferença. Assim, quanto mais iterações do algoritmo MPGBP são realizadas, mais o resíduo perde a característica inicial da imagem diferença, de modo que ele se torna mais e mais uniformemente distribuído na hiperesfera. Logo, o valor de $\Theta(\mathcal{C})$ tende a diminuir levando a um melhor desempenho do codificador.

Capítulo 6

Conclusão

Nesta Tese, nós investigamos bons dicionários para o codificador de vídeo baseado no algoritmo MPGBP [10, 40, 39].

Analizamos, no Capítulo 2, a proposta do algoritmo de decomposição de sinais matching pursuits face aos métodos de representação de sinais existentes na época do seu surgimento. A partir da sua formulação e da sua implementação no contexto de vídeo, observamos que o dicionário é uma parte importante da sua estrutura de decomposição.

No capítulo seguinte, expomos toda a teoria por trás do codificador de vídeo usado nesta Tese. Vimos que, à exemplo do algoritmo MP, o algoritmo MPGBP realiza uma decomposição voraz do sinal em questão, projetando-o em um dicionário supercompleto através de uma intensiva pelos melhores casamentos entre o sinal e os átomos do dicionário. Definimos, neste mesmo capítulo, condições a serem satisfeitas por estes dicionários de forma a proporcionar um melhor desempenho ao algoritmo, quais sejam: (i) apresentar pequenos valores de $\Theta(\mathcal{C})$ e de $q(\mathcal{C})$ e (ii) possuir características as mais próximas possível das características do sinal.

Nos propusemos, então, a criar dicionários que satisfizessem tais condições para o algoritmo de codificação de vídeo baseado no MPGBP. Para tanto, realizamos intenso levantamento bibliográfico que culminou com a descoberta de uma classe de funções bidimensionais multidirecionais: ridgelets, curvelets e contourlets. Assim, dedicamos o Capítulo 4 à exposição de toda a teoria por trás de tais funções e fizemos um paralelo com o nosso contexto de codificação de vídeo. Conjecturamos, então, que dicionários baseados nas funções ridgelet seriam bons candidatos para satisfazer

as condições (i) e (ii).

Por fim, no Capítulo 5 apresentamos alguns dicionários desenvolvidos durante os trabalhos de tese. Tais dicionários foram elaborados através da adição de átomos baseados em ridgelets ao dicionário de Neff e Zakhor [6]. Propusemos, de fato, uma nova forma de construção das ridgelets digitais que melhor se adaptasse ao nosso contexto de aplicação. Esta nova forma passou a ser mais flexível em muitos aspectos em relação às ridgelets originais. Mostramos que tal flexibilidade trouxe bons resultados através da construção de dois dicionários supercompletos e anisotrópicos propostos a partir de dois tipos de funções 1-D distintas, quais sejam: as wavelets e scaling functions de Meyer e as funções de Gabor. Apesar do grande crescimento na cardinalidade (e assim, da taxa para codificar cada átomo), atingimos uma redução em $\Theta(\mathcal{C})$ que compensou o crescimento da taxa, possibilitando um aumento significativo nos valores de PSNR para as seqüências e taxas aqui apresentadas, sobretudo para codificação utilizando taxas mais elevadas em imagens mais difíceis. Desta forma, demonstramos que realmente há como projetar dicionários melhores que o de Neff e Zakhor, com um bom compromisso entre $\Theta(\mathcal{C})$ e a cardinalidade $q(\mathcal{C})$.

Como trabalhos futuros, visamos um maior desenvolvimento desta nossa proposta de ridgelet. Como ponto de partida, avaliaremos as conseqüências do uso de janelamentos não-ortogonais entre as funções ridgelets na Equação (5.3). Em paralelo, realizaremos pesquisas bibliográficas com o intuito de encontrar outras funções unidimensionais que tragam algum benefício à nossa formulação.

Posteriormente procuraremos utilizar esta formulação de ridgelets em outros contextos de aplicação bidimensional, como, por exemplo, na codificação de imagens estáticas.

Apêndice A

Seqüências Originais

Neste apêndice nós mostramos a componente de luminância dos quadros 000, 020, 040, 060, 080, 100, 120, 140, 160, 180, 200, 220, 240, 260, 280 e 299 das seqüências originais: *coast-guard*, *container*, *foreman*, *hall-monitor*, *mother-and-daughter*, *silent-voice* e *weather*. Estas seqüências são compostas de 300 quadros QCIF (176×144) com 8bits/pixel.



coast 000



coast 020



coast 040



coast 060



coast 080



coast 100

Figura A.1: Quadros 000, 020, 040, 060, 080 e 100 da seqüência *coast-guard* original



coast 120



coast 140



coast 160



coast 180



coast 200



coast 220

Figura A.2: Quadros 120, 140, 160, 180, 200 e 220 da seqüência *coast-guard* original



coast 240



coast 260



coast 280



coast 299

Figura A.3: Quadros 240, 260, 280 e 299 da seqüência *coast-guard* original



container 000



container 020



container 040



container 060



container 080



container 100

Figura A.4: Quadros 000, 020, 040, 060, 080 e 100 da seqüência *container* original



container 120



container 140



container 160



container 180



container 200



container 220

Figura A.5: Quadros 120, 140, 160, 180, 200 e 220 da seqüência *container* original



container 240



container 260



container 280



container 299

Figura A.6: Quadros 240, 260, 280 e 299 da seqüência *container* original



foreman 000



foreman 020



foreman 040



foreman 060



foreman 080



foreman 100

Figura A.7: Quadros 000, 020, 040, 060, 080 e 100 da seqüência *foreman* original



foreman 120



foreman 140



foreman 160



foreman 180



foreman 200



foreman 220

Figura A.8: Quadros 120, 140, 160, 180, 200 e 220 da seqüência *foreman* original



foreman 240



foreman 260



foreman 280



foreman 299

Figura A.9: Quadros 240, 260, 280 e 299 da seqüência *foreman* original



hall 000



hall 020



hall 040



hall 060



hall 080



hall 100

Figura A.10: Quadros 000, 020, 040, 060, 080 e 100 da seqüência *hall-monitor* original



hall 120



hall 140



hall 160



hall 180



hall 200



hall 220

Figura A.11: Quadros 120, 140, 160, 180, 200 e 220 da seqüência *hall-monitor* original



hall 240



hall 260



hall 280



hall 299

Figura A.12: Quadros 240, 260, 280 e 299 da seqüência *hall-monitor* original



mother 000



mother 020



mother 040



mother 060



mother 080



mother 100

Figura A.13: Quadros 000, 020, 040, 060, 080 e 100 da seqüência *mother-and-daughter* original



mother 120



mother 140



mother 160



mother 180



mother 200



mother 220

Figura A.14: Quadros 120, 140, 160, 180, 200 e 220 da seqüência *mother-and-daughter* original



mother 240



mother 260



mother 280



mother 299

Figura A.15: Quadros 240, 260, 280 e 299 da seqüência *mother-and-daughter* original



silent 000



silent 020



silent 040



silent 060



silent 080



silent 100

Figura A.16: Quadros 000, 020, 040, 060, 080 e 100 da seqüência *silent-voice* original



silent 120



silent 140



silent 160



silent 180



silent 200



silent 220

Figura A.17: Quadros 120, 140, 160, 180, 200 e 220 da seqüência *silent-voice* original



silent 240



silent 260



silent 280



silent 299

Figura A.18: Quadros 240, 260, 280 e 299 da seqüência *silent-voice* original



weather 000



weather 020



weather 040



weather 060



weather 080



weather 100

Figura A.19: Quadros 000, 020, 040, 060, 080 e 100 da seqüência *weather* original



weather 120



weather 140



weather 160



weather 180



weather 200



weather 220

Figura A.20: Quadros 120, 140, 160, 180, 200 e 220 da seqüência *weather* original



weather 240



weather 260



weather 280



weather 299

Figura A.21: Quadros 240, 260, 280 e 299 da seqüência *weather* original

Referências Bibliográficas

- [1] MPEG, “Moving Picture Experts Group”.
<http://www.mpeg.org>.
- [2] “MPEG-2, test model 5”, Test Model Editing Committee, ISO/IEC/JTC1/SC29/WG11/93-225B, Abril de 1993.
<http://www.mpeg.org/MPEG/MSSG/tm5/>.
- [3] MPEG-4 standard MPEG97/N1796-ISO/IEC JTC1/SC29/WG11, Julho de 1997.
<http://www.m4if.org/>,
<http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm>.
- [4] AHMED, N., NATARAJAN, T., RAO, K. R., “Discrete Cosine Transform”, *IEEE Transactions on Computers*, v. **C-23**, pp. 90–93, 1974.
- [5] MALLAT, S. G., *A Wavelet Tour of Signal Processing*. 2 ed. Academic Press, 1999. ISBN : 0-12-466606-X.
- [6] NEFF, R., ZAKHOR, A., “Very Low Bit Rate Video Coding Based in Matching Pursuits”, *IEEE Transactions Circuits and Systems*, v. 7, n. 1, pp. 158–171, Fevereiro 1997.
- [7] KIM, B. J., PEARLMAN, W. A., “An Embedded Wavelet Video Coder Using Three-Dimensional Set Partitioning in Hierarchical Trees (SPIHT)”, *IEEE Data Compression Conference*, pp. 251–260, 1997.
- [8] CANDÈS, E. J., DONOHO, D. L., *Curve and Surface Fitting*, chapter Curvelets - A Surprisingly Effective Nonadaptive Representation For Objects with Edges,

- A. Cohen, C. Rabut and L. L. Schumaker (ed.), Vanderbilt University Press, Saint-Malo, pp. 105–120, 1999.
- [9] MALLAT, S. G., ZHANG, Z., “Matching Pursuits with Time-Frequency Dictionaries”, *IEEE Transactions on Signal Processing*, v. 41, n. 12, pp. 3397–3415, Dezembro 1993.
- [10] CAETANO, R., da SILVA, E. A. B., CIANCIO, A. G., “Matching pursuits video coding using generalized bit-planes”, *International Conference on Image Processing*, v. 3, n. 24–28, pp. 677–680, Junho 2002.
- [11] CRAIZER, M., da SILVA, E. A. B., RAMOS, E. G., “Convergent algorithm for successive approximation vector quantisation with applications to wavelet image compression”, *IEE Vis. Image Signal Processing*, v. 146, n. 3, pp. 159–164, 1999.
- [12] DONOHO, D. L., *Orthonormal Ridgelets and Linear Singularities*, Report, Stanford University, 1998.
<http://www-stat.stanford.edu/~donoho/Reports/1998/ridge-lin-sing.ps>.
- [13] CANDÈS, E., *Harmonic Analysis of Neural Nets.*, Report, Stanford University, 1997.
- [14] MURATA, N., “An integral representation of functions using three-layered networks and their approximation bounds”, *Neural Networks*, v. 9, pp. 947–956, 1996.
- [15] LOGAN, B. F., SHEPP, L. A., “Optimal reconstruction of a function from its projections”, *Duke Math*, v. 42, n. 4, pp. 645–659, 1975.
- [16] DEANS, S. R., *The Radon Transform and Some of Its Applications*. revised ed. Malabar, FL, Krieger Publishing Company, 1993.
- [17] DO, M. N., VETTERLI, M., “Contourlets: a directional multiresolution image representation”, *International Conference on Image Processing*, v. 1, pp. I–357 – I–360, Setembro 2002.

- [18] DONOHO, D. L., FLESIA, A. G., *Beyond Wavelets*, chapter Digital Ridgelet Transform based on True Ridge Functions, Grant Welland (ed.), Academic Press, pp. 1–30, Setembro de 2003.
- [19] JAIN, A. K., *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc., 1989. ISBN : 0-13-336165-9.
- [20] WATKINSON, J., *The Engineer’s Guide to Motion Compensation*. Snell & Wilcox Ltd.
<http://www.snellwilcox.com/knowledgecenter/books/books/emotion.pdf>, 1994.
- [21] CLARKE, R. J., *Digital Compression Of Still Images And Video*. Academic Press, 1995. ISBN : 012175720X.
- [22] SASTRE, J., FERRERAS, A., HERNÁNDEZ-GIL, J., “Motion Vector Size-Compensation Based Method For Very Low Bit Rate Video Coding”, *IEEE Transactions on Circuits and Systems for Video Technology*, v. 10, n. 7, pp. 1192–1197, Outubro de 2000.
- [23] IUT Telecommunications Standardization Sector LBC-95, “Video Codec Test Model TMN5” Disponível em Telenor Research, <http://www.nta.no/brukere/DVC>.
- [24] IUT-T Study Group 15, Working Party 15/1, Draft Recommendation H.263, Dezembro de 1995.
- [25] DINIZ, P. S. R., da SILVA, E. A. B., NETO, S. L., *Digital Signal Processing: System Analysis and Design*. Cambridge University Press, 2002. ISBN 0127432736.
- [26] CHEN, S. S., DONOHO, D. L., SAUNDERS, M. A., “Atomic Decomposition by Basis Pursuit”, *SIAM Journal on Scientific Computing*, v. 20, n. 1, pp. 33–61, 1999.
<http://www-stat.stanford.edu/~donoho/Reports/1995/30401.pdf>.

- [27] COIFMAN, R. R., WICKERHAUSER, M. V., “Entropy-Based Algorithms for Best Basis Selection”, *IEEE Transactions on Information Theory*, v. 38, n. 2, pp. 713–718, March 1992.
- [28] DAUBECHIES, I., “Time-frequency Localization Operators: A Geometric Phase Space Approach”, *IEEE Transactions on Information Theory*, v. 34, pp. 605–612, April 1988.
- [29] WICKERHAUSER, M. V., *Adapted Wavelet Analysis from Theory to Software*. A. K. Peters, Ltd, 1994.
- [30] GRIBONVAL, R., BACRY, E., “Harmonic decomposition of audio signals with matching pursuit”, *IEEE Transactions on Signal Processing*, v. 51, n. 1, pp. 101–111, Janeiro de 2003.
- [31] ETEMOGLU, A. O., CUPERMAN, V., “Matching Pursuits Sinusoidal Speech Coding”, *IEEE Transactions on Speech and Audio Processing*, v. 11, n. 5, pp. 413–424, Setembro de 2003.
- [32] AIAZZI, B., BARONTI, S., ALPARONE, L., “Blind image estimation through fuzzy matching pursuits”, *IEEE International Conference on Image Processing*, v. 1, pp. 241–244, 2001.
- [33] KIM, S., ILTIS, R. A., “A Matching-Pursuit/GSIC-Based Algorithm for DS-CDMA Sparse-Channel Estimation”, *IEEE Signal Processing Letters*, v. 11, n. 1, pp. 12–15, Janeiro de 2004.
- [34] ITU-T, “International Telecommunication Union, Telecom Standardization”.
<http://www.itu.int/ITU-T/>.
- [35] CHEUNG, S.-C. S., ZAKHOR, A., “Matching Pursuits Experimental Video Codec”.
<http://www-video.eecs.berkeley.edu/~cheungsc/work/docs/mpsoftware.pdf>.
- [36] NEFF, R., ZAKHOR, A., “Adaptive Modulus Quantizer Design for Matching Pursuits Video Coding”, *IEEE International Conference on Image Processing*, v. 2, pp. 81–85, 1999.

- [37] NEFF, R., ZAKHOR, A., “Modulus Quantization for Matching Pursuits Video Coding”, *IEEE Transactions Circuits and Systems for Video Technology*, v. 10, pp. 895–912, 2000.
- [38] VLEESCHOUWER, C. D., ZAKHOR, A., “In loop atom modulus quantization for matching pursuit and its applications to video coding”, *IEEE Transactions on Image Processing*, v. 12, n. 10, pp. 1226–1242, outubro de 2003.
- [39] CAETANO, R., da SILVA, E. A. B., CIANCIO, A. G., “Video Coding Using Greedy Decomposition on Generalized Bit-Planes”, *Electronics Letters*, v. 38, n. 11, pp. 507–508, maio de 2002.
- [40] CAETANO, R., *Codificação de Vídeo Usando Planos de Bits Generalizados*. Ph.D. dissertation, Universidade Federal do Rio de Janeiro, 2004.
- [41] SHAPIRO, J. M., “Embedded image coding using zerotrees of wavelet coefficients”, *IEEE Transactions on Signal Processing*, v. 41, n. 12, pp. 3445–3462, Dezembro de 1993.
- [42] SAID, A., PEARLMAN, W. A., “A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees”, *IEEE Transactions on Circuits and Systems for Video Technology*, v. 6, n. 3, pp. 243–250, Junho de 1996.
- [43] BELL, T. C., CLEARY, J. G., WITTEN, I. H., *Text Compression*. Englewood Cliffs, NJ, Prentice Hall, 1990.
- [44] SKODRAS, A., CHRISTOPOULOS, C., EBRAHIMI, T., “The JPEG 2000 still image compression standard”, *IEEE Signal Processing Magazine*, v. 18, pp. 36–58, Setembro de 2001.
- [45] CANDÈS, E. J., DONOHO, D. L., “Ridgelets: a key to higher-dimensional intermittency?”, *Phil. Trans. R. Soc. Lond. A.*, v. 357, pp. 2495–2509, Junho 1999.
- [46] DO, M. N., VETTERLI, M., *Beyond Wavelets*, chapter Contourlet, Grant Welland (ed.), Academic Press, pp. 83–106, Setembro de 2003.

- [47] RABBANI, M., JONES, P. W., *Introduction to the Theory of Fourier Integrals*. Society of Photo-Optical Instrumentation Engineers, 1991. ISBN : 0819406481.
- [48] PINKUS, A., *Surface Fitting and Multiresolution Methods*, chapter Approximating by Ridge Functions, A. Le Mehaute, C. Rabut and L. L. Schumaker (ed.), Vanderbilt University press, Nashville, pp. 279–292, 1997.
- [49] FRIEDMAN, J., STUETZLE, W., “Projection pursuit regression”, *J. Amer. Statist. Assoc.*, v. 76, pp. 817–823, 1981.
- [50] HUBER, P. J., “Projection pursuit”, *Ann. Statist.*, v. 13, pp. 435–475, 1985.
- [51] DONOHO, D. L., JOHNSTONE, I. M., “Projection-based approximation and a duality method with kernel methods”, *Ann. Statist.*, v. 17, pp. 58–106, 1989.
- [52] CYBENKO, G., “Approximation by superpositions of a sigmoidal function”, *Math. Control, Signals and Systems*, v. 2, n. 4, pp. 303–314, 1989.
- [53] LESHNO, M., LIN, V., PINKUS, A., *et al.*, “Multilayer feedforward networks with a nonpolynomial activation function can approximate any function”, *Neural Networks*, v. 6, pp. 861–867, 1993.
- [54] DEANS, S. R., *Mathematical Analysis of Physical Systems*, chapter The Radon Transform, Van Nostrand Reinhold, New York, pp. 81–133, 1985.
- [55] FLESIA, A. G., HEL-OR, H., AVERBUCH, A., *et al.*, *Beyond Wavelets*, chapter Digital Implementation of Ridgelet Packets, Grant Welland (ed.), Academic Press, pp. 31–60, Settembre de 2003.
- [56] DO, M. N., VETTERLI, M., “Orthonormal finite ridgelet transform for image compression”, *International Conference on Image Processing*, v. 2, pp. 367–370, Settembre 2000.
- [57] DO, M. N., VETTERLI, M., “The finite ridgelet transform for image representation.”, *IEEE Transactions on Image Processing*, to appear.
- [58] DONOHO, D., DUNCAN, M., “Digital Curvelet Transform: Strategy, Implementation, Experiments”, Novembre de 1999.
<http://www-stat.stanford.edu/~donoho/Reports/1999/DCvT.pdf>.

- [59] TITCHMARSH, E. C., *Digital Image Compression Techniques*. Clarendon Press, 1948.
- [60] CANDÈS, E. J., DONOHO, D. L., *New tight frames of curvelets and optimal representations of objects with smooth singularities*, Report, Stanford University, Novembro de 2002.
- [61] BURT, P. J., ADELSON, E. H., “The Laplacian pyramid as a compact image code”, *IEEE Transactions on Communications*, v. 31, n. 4, pp. 532–540, Abril de 1983.
- [62] DO, M. N., VETTERLI, M., “Framing pyramids”, *IEEE Transactions on Signal Processing*, v. 51, n. 9, pp. 2329–2342, Setembro de 2003.
- [63] BAMBERGER, R. H., SMITH, M. J. T., “A filter bank for the directional decomposition of images: Theory and design”, *IEEE Transactions on Signal Processing*, v. 40, n. 4, pp. 882–893, Abril de 1992.
- [64] DO, M. N., *Directional multiresolution image representations*. Ph.D. dissertation, Swiss Federal Institute of Technology, Dezembro de 2001.
- [65] NEFF, R., ZAKHOR, A., “Matching Pursuits Video Coding - Part 1: Dictionary Approximation”, *IEEE Transactions on Circuits and Systems for Video Technology*, v. 12, n. 1, pp. 13–26, Janeiro 2002.
- [66] MATHWORKS, “MATLAB”.
<http://www.mathworks.com/products/matlab/>.