



**RICARDO BARROSO LEITE**

**PROCEDIMENTOS AVANÇADOS EM CODIFICAÇÃO WAVELET ADAPTADA À  
GEOMETRIA PARA TRATAMENTO E COMPRESSÃO DE IMAGENS**

**CAMPINAS – SP**

**2014**



**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO**

**Ricardo Barroso Leite**

**Procedimentos Avançados em Codificação *Wavelet* Adaptada à  
Geometria para Tratamento e Compressão de Imagens**

**Orientador: Yuzo Iano**

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Telecomunicações e Telemática

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA TESE  
DEFENDIDA PELO ALUNO RICARDO BARROSO LEITE  
E ORIENTADO PELO PROF. DR. YUZO IANO

Campinas, SP  
2014

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca da Área de Engenharia e Arquitetura  
Rose Meire da Silva - CRB 8/5974

L536p Leite, Ricardo Barroso, 1984-  
Procedimentos avançados em codificação *wavelet* adaptada à geometria para  
tratamento e compressão de imagens / Ricardo Barroso Leite. – Campinas, SP :  
[s.n.], 2014.

Orientador: Yuzo Iano.  
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de  
Engenharia Elétrica e de Computação.

1. Teoria da codificação. 2. Compressão de imagens. 3. Wavelets  
(Matemática). 4. Televisão digital. I. Iano, Yuzo, 1950-. II. Universidade Estadual  
de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Advanced procedures in geometry adapted wavelet coding for image  
processing and compression

**Palavras-chave em inglês:**

Coding theory

Image compression

Wavelets (Mathematics)

Digital television

**Área de concentração:** Telecomunicações e Telemática

**Titulação:** Doutor em Engenharia Elétrica

**Banca examinadora:**

Yuzo Iano [Orientador]

David Bianchini

Luiz César Martini

Luís Geraldo Pedroso Meloni

Osamu Saotome

**Data de defesa:** 03-07-2014

**Programa de Pós-Graduação:** Engenharia Elétrica

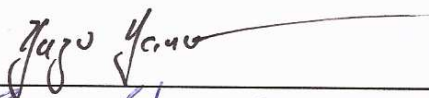
## COMISSÃO JULGADORA - TESE DE DOUTORADO

**Candidato:** Ricardo Barroso Leite

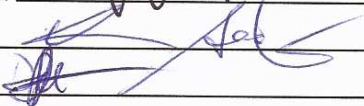
**Data da Defesa:** 3 de julho de 2014

**Título da Tese:** "Procedimentos Avançados em Codificação Wavelet Adaptada à Geometria para Tratamento e Compressão de Imagens"

Prof. Dr. Yuzo Iano (Presidente):



Prof. Dr. Osamu Saotome:



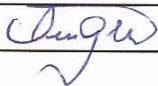
Prof. Dr. David Bianchini:



Prof. Dr. Luiz César Martini:



Prof. Dr. Luís Geraldo Pedroso Meloni:





# Resumo

Muitas áreas de pesquisa utilizam imagens digitais e outros arranjos multidimensionais de dados, que para serem transmitidos e armazenados de forma mais eficiente passam por um processo de filtragem e compressão. A transformada *wavelet* isotrópica é tradicionalmente usada e considerada um método rápido e eficiente para compressão. Por agregar as vantagens de representação multirresolução e a localização dos contornos, as *bandelelets* têm sido consideradas estado-da-arte em várias aplicações de processamento de imagens. Neste trabalho é apresentado um novo método para processamento e compressão de imagens baseado na transformada *bandelet*. Em nosso método, uma estimativa é feita de forma a reduzir o espaço de busca e tornar o processamento da imagem assintoticamente mais rápido. Os resultados mostram que pode ser feito um compromisso entre qualidade da imagem e tempo computacional, tornando o esquema mais atrativo para uma ampla gama de aplicações. Dentre as áreas beneficiadas por esse método estão transmissão de imagens e vídeo (TV digital e dispositivos móveis), imagens médicas e modelagem 3D.

**Palavras-chave:** Teoria da codificação, Compressão de imagens, Wavelets, Televisão digital.

# Abstract

Many research areas use digital images and other multi-dimensional arrays of data which to be transmitted and stored more efficiently pass through a filtering process and compression. The isotropic wavelet transform is traditionally used and considered a fast and efficient method for compression. By adding the advantages of multi-resolution representation and location of edges, the *bandelelets* have been considered state of the art in several image processing applications. This work presents a new method for processing and compression based on *bandelet* transform images. In our method, an estimation is made to reduce the search space and make the image processing asymptotically faster. The results show that a compromise can be made between image quality and computational time, which makes it a more attractive scheme for a wide range of applications. Among the areas benefited by this method are image and video transmission (digital TV and mobile devices), medical imaging and 3D modeling.

**Keywords:** Coding theory, Image Compression, Wavelets, Digital TV.





# Sumário

<b>Lista de Figuras</b>	<b>xvii</b>
<b>Lista de Tabelas</b>	<b>xxi</b>
<b>Glossário</b>	<b>xxiii</b>
<b>Trabalhos Publicados Pelo Autor</b>	<b>xxv</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contexto . . . . .	1
1.2 Contribuições e Organização do Trabalho . . . . .	2
1.3 Organização da Tese . . . . .	3
<b>2 Processamento e Compressão de Imagens Digitais</b>	<b>5</b>
2.1 Codificação por Transformadas Ortogonais . . . . .	5
2.2 A Transformada <i>Wavelet</i> . . . . .	8
2.2.1 Algoritmo de Mallat . . . . .	10
2.3 Bases <i>Wavelet</i> Separáveis . . . . .	11
2.4 Aproximações de Imagens . . . . .	13
2.4.1 Aproximações Lineares . . . . .	14
2.4.2 Aproximações Não-Lineares . . . . .	17
2.5 O Problema das Transformadas Isotrópicas . . . . .	18
<b>3 Novos Métodos para Processamento de Geometria</b>	<b>23</b>
3.1 <i>Directionlets</i> . . . . .	24
3.2 <i>Curvelets</i> . . . . .	26
3.3 <i>Wedgelets</i> . . . . .	27
3.4 <i>Arclets</i> . . . . .	33
3.5 <i>Wedgeprints</i> . . . . .	35
3.6 <i>Surflets</i> e <i>Surfprints</i> . . . . .	35
3.7 <i>Platelets</i> . . . . .	39
3.8 <i>Bandelets</i> de 1ª Geração . . . . .	42
3.9 <i>Bandelets</i> de 2ª Geração . . . . .	43

<b>4</b>	<b>Método rápido para obtenção da orientação</b>	<b>47</b>
4.1	Estado-da-arte e Método Proposto . . . . .	47
4.1.1	Cálculo dos Coeficientes de Aproximação em Cada Segmento de Imagem . .	49
4.1.2	Obtenção da Melhor Árvore <i>Quadtree</i> . . . . .	50
4.1.3	Novas Funções de Custo . . . . .	51
4.1.4	Codificação . . . . .	51
4.2	Método de Momentos . . . . .	52
4.2.1	Momentos Geométricos . . . . .	52
4.2.2	Momentos Geométricos Customizados . . . . .	53
4.3	Operadores de Detecção de Contornos . . . . .	54
4.4	Da Transformada <i>Bandelet</i> para a Transformada <i>Bandelet</i> Rápida . . . . .	56
<b>5</b>	<b>Resultados</b>	<b>59</b>
5.1	Introdução e Observações sobre as Simulações . . . . .	59
5.2	Estimação da Orientação . . . . .	60
5.3	<i>Bandelet</i> Rápida . . . . .	64
5.4	<i>Bandelet</i> com espaço de busca limitado . . . . .	66
<b>6</b>	<b>Aplicações da Presente Proposta</b>	<b>71</b>
6.1	Compressão de imagens e vídeo de alta resolução . . . . .	71
6.1.1	Super-Resolução . . . . .	72
6.1.2	<i>Inpainting</i> . . . . .	73
6.1.3	Sensoriamento Comprimido . . . . .	73
6.2	Mapas de Profundidade . . . . .	73
6.3	Modelagem 3D . . . . .	77
6.4	Segurança . . . . .	77
6.5	Sensoriamento Remoto . . . . .	78
6.6	Diagnósticos em Medicina . . . . .	78
<b>7</b>	<b>Conclusões</b>	<b>81</b>
7.1	Comentários sobre o Contexto e Contribuições . . . . .	81
7.2	Comentários sobre Simulações e Resultados . . . . .	82
7.3	Sugestões para Trabalhos Futuros . . . . .	82
	<b>Referências Bibliográficas</b>	<b>83</b>

*Ao meu pequeno Guilherme.*



# Agradecimentos

Após mais uma jornada concluída, depois de dedicar bastante tempo a este trabalho e tendo aprendido muito durante o programa de doutorado, sinto-me cada vez mais motivado para a carreira de pesquisador. Por isso, não poderia deixar de agradecer a todos que me apoiaram, me incentivaram e me inspiraram.

A meu orientador, prof. Dr. Yuzo Iano, que desde a graduação me deu a oportunidade de trabalhar com ele e que sempre me deu muita liberdade criativa e teve confiança em meu trabalho. Suas reuniões sempre estimularam nosso grupo de pesquisa e acredito que todos nós orientados somos muito gratos. Aprendi muito em cada projeto desenvolvido em seu laboratório (ITASAT, CAPES-RHTVD, RNP-STB-SCAN, RNP-Spaces4D).

Aos professores Dr. Osamu Saotome, Dr. David Bianchini, Dr. Luiz César Martini e Dr. Luís Geraldo P. Meloni por participarem da defesa da tese, aos quais não conhecia profundamente, mas com os quais espero ter outras oportunidades de colaboração acadêmica.

À Faculdade de Engenharia Elétrica e de Computação da Unicamp, na qual obtive minha graduação, mestrado e agora concluo o doutorado e cujo nível de excelência acadêmica sempre ficará marcado em minha formação. Ao CNPq, CAPES e FAPESP por terem me dado apoio financeiro para realizar as pesquisas por meio de bolsas de Iniciação Científica, Mestrado e Doutorado, respectivamente.

À minha família pelo apoio durante esta jornada, em especial minha mãe, Maria Cláudia, por todo suporte que só uma mãe sabe dar.

À minha esposa Angela e ao meu filho Guilherme, por iluminarem meus dias e darem um sentido à minha vida.

Aos meus colegas de pós-graduação do Laboratório de Comunicações Visuais que têm sido grandes amigos além de terem me ajudado quando a pesquisa se tornava difícil. Agradeço também aos meus colegas de graduação, que embora em caminhos distintos, têm sempre mantido o contato e a amizade.

A Deus por iluminar meu caminho e não me deixar fraquejar. Maior do que o orgulho de chegar até aqui é a fé em continuar lutando.

# Agradecimentos

Agradeço à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pelo apoio financeiro e pelo incentivo acadêmico para a realização deste trabalho.

# **Agradecimentos**

Agradecimentos especiais ao programa CAPES RH-TVD da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior pelo incentivo acadêmico para que este trabalho pudesse ser realizado.





# Lista de Figuras

2.1	Funções base da DCT 2D. . . . .	7
2.2	Efeito de bloco em imagem codificada DCT (a) original (b) recuperada. . . . .	7
2.3	Esquema de codificação por sub-banda. . . . .	9
2.4	Codificação por sub-banda de duas bandas e reconstrução. . . . .	10
2.5	Algoritmo de Mallat para decomposição <i>wavelet</i> . . . . .	11
2.6	Algoritmo de Mallat para reconstrução <i>wavelet</i> . . . . .	11
2.7	Exemplo da aplicação do algoritmo de Mallat: (a) Sinal original (b) Aproximações multiresolução discretas nas escalas $2^{-j}$ calculadas com <i>wavelets splines</i> cúbicas. . . . .	12
2.8	Decomposição <i>wavelet</i> ortogonal 2D para $J = 3$ . . . . .	13
2.9	Exemplo de decomposição <i>wavelet</i> : (a) Imagem original (b) Decomposição <i>wavelet</i> Daubechies separável (passa-alta foi ressaltada para comparação). . . . .	14
2.10	Diagrama em blocos para imagem de: (a) Um estágio de decomposição <i>wavelet</i> (b) Um estágio de reconstrução <i>wavelet</i> . . . . .	15
2.11	Reconstrução de um sinal unidimensional de variação limitada (a) Sinal original (b) Aproximação Fourier (c) Aproximação por <i>wavelet</i> Daubechies 4. . . . .	17
2.12	Reconstrução de um sinal unidimensional (a) Sinal original (b) Decomposição <i>wavelet</i> , com os $N/6$ maiores coeficientes <i>wavelet</i> Symmlet 4 (c) Aproximação não-linear. . . . .	18
2.13	Esquemas de aproximação (a) Imagem original ( $N = 2562$ ) (b) Decomposição <i>wavelet</i> , pontos escuros correspondem a $ \langle f, \psi_{j,n}^k \rangle  > T$ (c) Aproximação linear com $N/16$ (d) Aproximação não-linear com $N/16$ . . . . .	19
2.14	Ineficiência da <i>wavelet</i> isotrópica em representar a geometria de uma imagem. (a) Imagem original (b) 13 (c) 26 (d) 52. . . . .	20
2.15	Projeção <i>wavelet</i> (a) Imagem $f = C \cdot \mathbf{1}_\Omega$ (b) Na escala $2^j$ , as <i>wavelets</i> $\psi_{j,n}^l$ têm um suporte quadrado de largura proporcional a $2^j$ . Esse suporte corresponde a uma grade de intervalo $2^j$ , indicada pelos pequenos pontos. Pontos maiores indicam <i>wavelets</i> com suporte que cruza a fronteira de $\Omega$ para $\langle f, \psi_{j,n}^l \rangle \neq 0$ . . . . .	21
2.16	<i>Ringing</i> e embaçamento ( <i>blur</i> ) em uma descontinuidade unidimensional. . . . .	21
2.17	Exemplo de <i>ringing</i> em imagem codificada por TW (a) original (b) após a decodificação. . . . .	22
3.1	Geometrias de aproximação (a) Imagem original (b) Aproximação com triangulação (c) aproximação com uma base <i>wavelet</i> . . . . .	23

3.2	Funções base para a decomposição <i>wavelet</i> usual e anisotrópica, para diferentes direções e bancos de filtros. (a) <i>wavelet</i> Haar isotrópica usual (b) <i>Wavelet</i> Haar isotrópica inclinada com orientação de $45^\circ$ (c) <i>Wavelet</i> Haar anisotrópica inclinada com orientação de $45^\circ$ (d) <i>Wavelet</i> biortogonal 9-7 isotrópica usual (e) <i>Wavelet</i> biortogonal isotrópica inclinada com orientação de $45^\circ$ (f) <i>wavelet</i> biortogonal 9-7 anisotrópica inclinada com orientação de $45^\circ$ . . . . .	25
3.3	Árvores de descendência dos coeficientes <i>wavelet</i> (a) <i>Wavelet</i> 2D usual (b) <i>Wavelet</i> anisotrópica inclinada na direção de $45^\circ$ . . . . .	25
3.4	Aproximações em blocos não retangulares. (a) Triângulo adaptado para aproximar uma curva (b) Elemento <i>curvelet</i> de aproximação. . . . .	26
3.5	Plano de frequências 2D da decomposição <i>curvelet</i> . . . . .	27
3.6	Banco de filtros da transformação <i>contourlet</i> , evidenciando a divisão em frequências discretizadas 2D. . . . .	28
3.7	Divisão em frequências das <i>curvelets</i> 3D. . . . .	28
3.8	Estrutura <i>quadtree</i> . . . . .	28
3.9	Decomposição <i>wedgelet</i> (a) Imagem artificial (b) decomposição <i>wedgelet</i> multiescala (c) parametrização de um <i>wedgelet</i> em um quadrado $N \times N$ diádico: $\theta$ é a orientação angular do contorno, $d$ é a distância do contorno ao centro do bloco, $m_1$ e $m_2$ são as intensidades em cada lado do contorno. . . . .	29
3.10	Decomposição <i>wedgelet</i> de um trecho de imagem (a) escala única (tamanho dos blocos fixo) (a) multiescala. . . . .	29
3.11	Modelo de cadeia de Markov para representar as transições na <i>quadtree</i> . . . . .	31
3.12	Estruturas geométricas mais suaves apresentam uma entropia menor do que estruturas mais complexas. . . . .	32
3.13	Aproximações em <i>quadtree</i> (a) <i>Quadtree</i> com 967 nós de aproximação constante, PSNR: 33,69 dB. (b) <i>Quadtree</i> com 248 nós <i>wedgelet</i> (496 coeficientes <i>wedgelet</i> ), PSNR: 33,71 dB. . . . .	33
3.14	<i>Wedgelets</i> de segunda-ordem ou <i>arcllet</i> . . . . .	34
3.15	<i>Wedgelets</i> no domínio <i>wavelet</i> (a) Porção de uma imagem contendo um <i>wedgelet</i> em um bloco (b) <i>Wedgeprint</i> obtido pela projeção do <i>wedgelet</i> na sub-banda vertical do domínio <i>wavelet</i> . . . . .	36
3.16	Obtenção de um <i>wedgeprint</i> : (a) Crie uma imagem temporária $Y$ usando extensões lineares nas bordas de $S_{j,k}$ . (b) Obtenha a transformada <i>wavelet</i> de $Y$ e extraia a subárvore de coeficientes <i>wavelet</i> com raiz no nó $(j, k)$ (escolhido na sub-banda horizontal, neste exemplo). (c) Um procedimento similar é usado para projetar um agrupamento <i>wedgelet</i> . . . . .	36
3.17	Exemplos de surflets: (a) $M = 2$ , $Kd = (1,2)$ (b) $M = 2$ , $Kd = (2,3)$ (c) $M = 3$ , $Kd = (1,2)$ (d) $M = 3$ , $Kd = (2,3)$ . . . . .	37
3.18	Exemplos de aproximações por árvores <i>surflet</i> : (a) Cúbica por partes com $M = 2$ (b) Linear por partes com $M = 3$ . . . . .	38
3.19	<i>Surflets</i> no domínio <i>wavelet</i> : (a) Porção de uma imagem contendo um <i>surflet</i> em um bloco (b) <i>Surfprint</i> obtido pela projeção do <i>surflet</i> na sub-banda vertical do domínio <i>wavelet</i> . . . . .	38
3.20	Elemento do conjunto <i>platelet</i> . . . . .	39

3.21	Aproximações de uma imagem suave por partes: (a) Imagem original (b) <i>wavelet</i> Haar (c) <i>Wedgelets</i> (d) <i>Platelets</i> em regiões quadradas (e) <i>Platelets</i> em regiões definidas por <i>wedgelets</i> . . . . .	40
3.22	Erro de aproximação em relação ao número de parâmetros total necessário. . . . .	41
3.23	Aproximação <i>smoothlet</i> (a) Imagem <i>horizon</i> (b) Imagem <i>horizon</i> borrada ( <i>blurred</i> ), de acordo com a curvatura. . . . .	41
3.24	Exemplo de partição de uma imagem em quadrados diádicos de acordo com o fluxo geométrico. . . . .	42
3.25	Contornos em uma imagem: (a) Coeficientes <i>wavelet</i> na escala $2^j$ são amostras da função $f * \overline{\psi}_j^k(x)$ (b) A imagem filtrada varia regularmente ao longo de um contorno $\gamma$ . . . . .	44
3.26	Decomposição <i>bandelet</i> : (a) Coeficientes <i>wavelet</i> de uma imagem. (b) Exemplo de segmentação de uma escala da transformada <i>wavelet</i> em quadrados diádicos de diferentes tamanhos. . . . .	45
3.27	Determinação do fluxo geométrico da decomposição <i>bandelet</i> : (a) Partição quadrada de coeficientes <i>wavelet</i> contendo um contorno. O fluxo geométrico paralelo ao contorno é mostrado com setas. (b) Uma distorção ( <i>warping</i> ) $w$ mapeia o fluxo original para um fluxo horizontal. (c) Suporte das <i>wavelets</i> direcionais de comprimento $2^i$ e largura $2^l$ no domínio distorcido. . . . .	45
3.28	Decomposição <i>wavelet</i> direcional: (a) Decomposição <i>wavelet</i> (b)-(d) <i>wavelets</i> direcionais Alpert de diferentes comprimento e largura. (e)-(g) Funções <i>bandelet</i> calculadas a partir das <i>wavelets</i> direcionais de (b)-(d) correspondentes aos quadrados $S_1, S_2$ e $S_3$ , respectivamente, de (a). . . . .	46
4.1	Diagrama de blocos geral da compressão utilizando métodos geométricos. . . . .	47
4.2	Diagrama de blocos da compressão utilizando decomposição <i>bandelet</i> de 2º ordem. . . . .	49
4.3	Somas parciais para o cálculo da média em uma região com contorno bem definido. . . . .	50
4.4	Estimativa do parâmetro angular do <i>wedgelet</i> a partir de níveis superiores. . . . .	52
4.5	Magnitude do gradiente por diferentes operadores (a) Imagem original (b) Prewitt (c) Sobel. . . . .	55
5.1	Aproximação do caso de descontinuidade reta (a) Imagem original (b) Reconstrução pelo operador de Prewitt (c) Reconstrução pelo operador de Sobel (d) Reconstrução pelo cálculo de momentos. . . . .	60
5.2	Aproximação do caso de descontinuidade curva (a) Imagem original (b) Reconstrução pelo operador de Prewitt (c) Reconstrução pelo operador de Sobel (d) Reconstrução pelo cálculo de momentos. . . . .	61
5.3	Aproximação do caso de descontinuidade reta e uma região suave (a) Imagem original (b) Reconstrução pelo operador de Prewitt (c) Reconstrução pelo operador de Sobel (d) Reconstrução pelo cálculo de momentos. . . . .	62
5.4	Aproximação do caso de descontinuidade reta e duas regiões suaves (a) Imagem original (b) Reconstrução pelo operador de Prewitt (c) Reconstrução pelo operador de Sobel (d) Reconstrução pelo cálculo de momentos. . . . .	63

5.5	Estimação de um círculo maior (“floresta”) composto por pequenos círculos menores (“árvores”) (a) Imagem original (b) Solução encontrada por operadores locais (c) Solução encontrada por um método baseado em momentos. . . . .	64
5.6	Tempo computacional medido e estimado para imagens de diferentes tamanhos: (a) BT, (b) TW e FBT. . . . .	65
5.7	Qualidade da imagem para diferentes tamanhos do espaço de procura: (a) Momentos, (b) Prewitt e (c) Sobel. . . . .	67
5.8	Qualidade da imagem versus tempo computacional para diferentes tamanhos de procura com tamanho da imagem $512 \times 512$ e taxa de <i>bits</i> 1,0 bpp. . . . .	68
5.9	Comparação da imagem “Barbara” comprimida em 0,28 bpp (a) Imagem original, (b) Imagem original ( <i>zoom</i> ), (c) TW, (d) FBT, (e) FBT+4, (f) BT. . . . .	69
5.10	Comparação da imagem “Lena” comprimida em 0,28 bpp (a) Imagem original, (b) Imagem original ( <i>zoom</i> ), (c) TW, (d) FBT, (e) FBT+4, (f) BT. . . . .	70
6.1	Exemplo de <i>inpainting</i> (a) Imagem original (b) Imagem processada, com objeto removido. . . . .	74
6.2	Exemplos de mapa de profundidade: (a) Estrutura cúbica, (b) Superfícies próximas estão mais escuras (c) Superfícies próximas do plano focal estão mais escuras. . . . .	75
6.3	Compressão de imagens de mapa de profundidade usando <i>platelets</i> (a) Projeção 2D de uma cena (imagem “Teddy”) (b) Mapa de profundidades original (c) Divisão <i>quadtree</i> (d) Correspondente mapa de profundidade reconstruído, com taxa de <i>bits</i> 0,12 bpp e PSNR de 36,1 dB. . . . .	76
6.4	Exemplo de codificação de superfícies: (a) Modelo original (b) Trecho original (c) Codificação <i>wavelet</i> com 0,2 bpv (d) Codificação <i>bandelet</i> com 0,2 bpp . . . . .	77
6.5	Exemplo de aplicação de <i>platelets</i> em imagens médicas: (a) Imagem de medicina nuclear com ruído (exemplo cardíaco), (b) Aproximação por <i>wavelet</i> de Haar (c) Aproximação por <i>platelets</i> . . . . .	79

# Lista de Tabelas

3.1	Comparação da complexidade computacional de decomposições geométricas em uma imagem $N \times N$ . . . . .	24
4.1	Técnicas inovadoras propostas no presente trabalho para otimização da decomposição <i>bandelet</i> . . . . .	48
5.1	Orientação estimada por cada método. . . . .	60
5.2	Tempo computacional (segundos) para a imagem “Barbara”. . . . .	65



# Glossário

bpp - *bits por pixel*

BT - *Bandelet Transform*

DCT - *Discrete Cosine Transform*

EBCOT - *Embedded Block Coding with Optimized Truncation*

FBT - *Fast Bandelet Transform*

FPGA - *Field-programmable Gate Array*

GPU - *Graphic Processing Unit*

HDTV - *High Definition Television*

HEVC - *High Efficiency Video Coding*

IDCT - *Inverse Discrete Cosine Transform*

ISDB-Tb - *Integrated Services Digital Broadcasting, Terrestrial, Brazilian version*

JPEG - *Joint Photographs Expert Group*

KLT - *Karhunen-Loève Transform*

MPEG - *Moving Pictures Expert Group*

MSE - *Mean Square Error*

PET - *Positron Emission Tomography*

PSNR - *Peak Signal to Noise Ratio*

SAR - *Synthetic Aperture Radar*

SD - *Standard Definition*

SFQ - *Space Frequency Quantization*

SPECT - *Single Photon Emission Computed Tomography*

SPIHT - *Set Partitioning In Hierarchical Trees*

SR - Super Resolução

SSIM - *Structural SIMilarity*

TW - Transformada *Wavelet*

UHDTV - *Ultra High Definition Television*

VCEG - *Video Coding Experts Group*



# Trabalhos Publicados Pelo Autor

1. R. S. Higa, R. F. Larico, R. B. Leite, R. Arthur, Y. Iano. “Plenoptic image compression comparison between JPEG, JPEG2000 and SPITH.” *Cyber Journals: Multidisciplinary Journals in Science and Technology*, v. 3, p. 1, 2013.
2. R. B. Leite, A. L. M. SILVA, F. S. SILVA, Y. IANO. “Codificador de Imagens Wedgelet-Wavelet por Blocos.” *Revista Telecomunicações*, v. 14, p. 1, 2012.
3. R. S. Higa, Y. Iano, R. B. Leite, R. F. Larico. “The use of light field cameras in surveillance applications.” *Collaborative Conference on 3D Research (CC3DR) 2013*, 2013, Jeju. CC3DMR 2013 Program and Abstract, 2013.
4. R. S. Higa, Y. Iano, R. B. Leite, R. F. Larico, R. Arthur. “Challenges in using a light field camera for surveillance applications.” *Latin Display 2012*, 2012, São Paulo - SP - Brazil. Proceedings of LatinDisplay 2012/IDRC 2012, 2012.
5. R. B. Leite, Y. Iano. “Fast Bandelet-Based Image Compression Using Local Edge Estimations.” *Collaborative Conference on 3D Research (CC3DR) 2014*, 2014. Incheon/Seoul. CC3DMR 2014 Program and Abstract, 2014.



# Capítulo 1

## Introdução

### 1.1 Contexto

As duas últimas décadas assistiram a uma grande atividade no desenvolvimento de novas ferramentas matemáticas e computacionais baseadas em ideias multiescala e multiresolução em muitos campos da ciência e tecnologia. Em tecnologia da informação e principalmente processamento de sinais, o desenvolvimento de *wavelets* e ideias relacionadas levaram a métodos convenientes para processamento e visualização de grandes bases de dados, a transmitir dados rapidamente por meio de compressão, a remover ruído de sinais e imagens e a identificar características cruciais em tais dados. Em ciências computacionais, *wavelets* e outros métodos multiescala permitiram a aceleração de cálculos fundamentais tais como a solução de equações parciais diferenciais.

Apesar do considerável sucesso, pesquisas intensas nos últimos anos têm mostrado que ideias multiresolução clássicas estão longe de serem universalmente eficientes. [1] [2] [3] De fato, assim como as pessoas reconheceram que os métodos baseados na análise de Fourier não eram bons para todos os propósitos - e conseqüentemente foram introduzidos novos sistemas, como as *wavelets* - pesquisadores têm buscado alternativas a análise *wavelet*. Em processamento de sinais, por exemplo, é necessário lidar com o fenômeno que ocorre ao longo de curvas, como os contornos em uma imagem bidimensional. Enquanto *wavelets* são certamente adequadas para tratar objetos em que singularidades estão associadas a pontos isolados, elas não são adaptadas para detectar, organizar e prover representação compacta para estruturas dimensionais complexas. [1] Dada a importância de tais fenômenos, tem havido um grande esforço de pesquisadores em obter alternativas mais bem adaptadas combinando ideias da geometria com ideias da análise multiresolução tradicional. O desenvolvimento e a adoção dos métodos geométricos resultam da sua similaridade com a visão humana. De fato, eles podem detectar mudanças de localização, escala e também orientação.

Os métodos de *wavelets* geométricas podem ser divididos em dois grupos. O primeiro grupo baseia-se na teoria de frames, utilizando novas transformadas parametrizadas para diferentes localizações, escalas e orientações, expandindo o domínio de cálculo e permitindo maior adaptação a diferentes imagens. Nesse grupo enquadram-se as *directionlets* [4], *curvelets* [5], dentre outros. O segundo grupo consiste em dicionários de elementos discretos, normalmente definidos sobre o domínio espacial, embora com suporte multiescala. Nesse grupo estão as *wedgelets* [6], *arcllets* [7] e *platelets* [8]. Embora possam ser mais adaptados a diferentes aspectos das imagens com menor número de coeficientes, esses métodos são muito mais lentos para serem processados, envolvendo

algoritmos de complexidade exponencial quanto ao número de elementos no dicionário e ao tamanho das imagens. [4][9]

As *bandelelets*, embora se encaixem no primeiro grupo, conseguem uma aproximação bastante compacta e aproveitam o legado dos métodos de decomposição *wavelet* já presentes, tais como *lifting* e o algoritmo de Mallat [10]. Assim como os métodos do segundo grupo, as *bandelelets* também necessitam de algoritmos exaustivos para determinação da orientação local, que irá definir a geometria, inviabilizando sua adoção em muitas aplicações.

## 1.2 Contribuições e Organização do Trabalho

Neste trabalho serão apresentados os principais métodos atualmente considerados estado-da-arte e apresentadas suas eficiências e deficiências frente aos outros métodos menos eficientes, principalmente na tarefa de compressão de imagens. Devido a similaridade das *bandelelets* com as *wavelets* e a existência de referências teóricas e práticas para sua implementação, além de outras vantagens que serão detalhadas no decorrer do texto, esse método foi escolhido para a implementação de um algoritmo novo. Este algoritmo tem como objetivo principal reproduzir o desempenho das *bandelelets* na compressão de imagens (considerando a qualidade de imagem objetiva frente a taxa de *bits* medida aqui por codificadores de entropia) frente aos métodos *wavelet* tradicionais e outros, como atingido nas publicações deste método em estado-da-arte, mas com menor tempo de processamento assintótico, demonstrado por uma análise destes algoritmos e pela medida de tempo computacional para um conjunto de simulações.

Para auxiliar na tarefa de otimização do algoritmo de compressão por *bandelelets* quanto ao tempo computacional, serão usados dois conjuntos de ferramentas matemáticas que já têm sucesso considerável em outras atividades de processamento de imagens e até em aplicações de compressão [11]. São elas os momentos geométricos [12] [13] [14] [15] [16], neste caso em uma forma customizada, e os detectores de bordas [17], que serão usados exclusivamente para determinação da orientação. Estes métodos serão detalhados no texto e comparados na seção de resultados. Espera-se com essa comparação apresentar ao leitor as vantagens e deficiências de se usar um ou outro método já bem estabelecido na área de processamento de imagens para auxiliar na compressão.

Todo algoritmo novo tem suas vantagens e desvantagens quanto aos métodos previamente estabelecidos. O algoritmo que será apresentado aqui não é exceção e o ganho em tempo computacional vem associado a uma queda no desempenho da compressão. Felizmente essa queda não é tão grande frente ao ganho conseguido em termos de tempo computacional, como será mostrado no capítulo 5. Além disso, é proposto também um método adaptativo, em que é possível alterar a precisão desejada e oscilar entre um alto custo computacional e melhor desempenho ou um menor custo computacional e desempenho inferior, mas ainda superior à abordagem tradicional por *wavelets*. Na seção de resultados essa comparação será mostrada em detalhes.

Em resumo, propõem-se como contribuições deste trabalho:

- Apresentação dos métodos *wavelet* geométricos e a comparação teórica quanto às vantagens e desvantagens para a compressão de imagens.
- Desenvolvimento de um algoritmo novo, baseado no método de aproximação por *bandelelets*, com o objetivo principal de diminuir o tempo computacional.

- Desenvolvimento de um algoritmo novo com controle de precisão, de forma a diminuir o tempo computacional em detrimento da qualidade ou o inverso.

## 1.3 Organização da Tese

A tese está organizada de acordo com os seguintes capítulos:

**Capítulo 2 Processamento e Compressão de Imagens Digitais:** Revisão das técnicas tradicionalmente usadas para processamento e compressão de imagens, em especial as transformadas *wavelet* 2D isotrópicas.

**Capítulo 3 Novos Métodos para Processamento de Geometria:** Apresentação dos métodos de processamento de imagens envolvendo *wavelet* geométricas.

**Capítulo 4 Método rápido para obtenção da orientação:** Desenvolvimento de um novo método baseado em transformada *bandelet* e apresentação das ferramentas matemáticas envolvidas.

**Capítulo 5 Resultados:** Descrição do ambiente computacional e das imagens utilizadas para os testes, apresentação de resultados e comparações dos envolvendo diferentes parâmetros.

**Capítulo 6 Aplicações da Presente Proposta:** Sugestões de novas áreas de aplicação que podem ser beneficiadas pelo algoritmo proposto ou por outros métodos *wavelet* geométricos.

**Capítulo 7 Conclusões:** Comentários e conclusões acerca dos resultados obtidos, analisando-se as perspectivas de melhorias e extensões do procedimento proposto.

Espera-se, por meio deste trabalho e das sugestões aqui apresentadas, estimular o uso de métodos *wavelet* geométricos, em especial com a inclusão de ferramentas matemáticas que evitem o alto custo computacional tradicionalmente presente.



## Capítulo 2

# Processamento e Compressão de Imagens Digitais

Neste capítulo será feita uma revisão dos métodos usados tradicionalmente, em especial a transformada *wavelet* 2D isotrópica. Em seguida, será demonstrada a ineficiência dos métodos tradicionais no processamento de estruturas geométricas em imagens, servindo como motivação para este trabalho.

### 2.1 Codificação por Transformadas Ortogonais

Em processamento ou compressão de imagens digitais, é fundamental converter o sinal original para uma nova forma de representação, que destaque suas características e isole componentes a serem processados separadamente. Em geral, deseja-se obter uma boa representação de um sinal com relação a um determinado funcional ou função de custo, que pondere a qualidade do sinal com o número de componentes necessários para sua reconstrução. Representações esparsas são aquelas em que a maior parte dos componentes é nula ou insignificante. Esses componentes podem ser desprezados na reconstrução sem grandes perdas.

Uma forma de mensurar o conceito de “esparsidade” é pelo número de componentes não-nulas em uma representação, definido matematicamente como a norma  $\ell^0$ : [3]

$$\|x\|_0 = \lim_{p \rightarrow 0} \|x\|_p = \lim_{p \rightarrow 0} (|x[1]|^p + |x[2]|^p + \dots + |x[N]|^p)^{\frac{1}{p}} \quad (2.1)$$

Para compressão de imagens com mínimas perdas (*lossless*), são desejáveis aproximações que representem bem aspectos visuais e que consigam boa reconstrução em termos da norma  $\ell^2$ , também chamada de erro quadrático médio (MSE - *Mean Square Error*): [3]

$$\|f\|^2 = \sum_{n=1}^N |f[n]|^2 \quad (2.2)$$

$$MSE = E(\|error\|^2) = \frac{\|f - \hat{f}\|^2}{N} = \frac{1}{N} \sum_{n=1}^N |f[n] - \hat{f}[n]|^2 \quad (2.3)$$

Onde  $\hat{f}[n]$  representam as amostras do sinal  $f$ .

É importante destacar que, para qualquer função  $f$ , sempre existe uma representação adaptativa que se encaixa de forma ótima (com máxima “esparsidade”, ou seja, um elemento apenas para representação exata), a base ortonormal: [5]

$$\psi = \{\psi_0, \psi_1, \dots\} \quad \text{com} \quad \psi_0 = \frac{f}{\|f\|^2} \quad (2.4)$$

No entanto, essa base não tem utilidade, pois os coeficientes não apresentam nenhuma informação sobre o sinal. Além disso, para que seja aplicada de forma genérica, uma base deve conter elementos que sejam adequados a qualquer sinal (ou a uma classe significativa). Como proposto por [18], uma base é adequada quando é possível encontrar elementos para representar qualquer sinal de entrada em tempo polinomial em relação ao tamanho do sinal original. A base trivial mostrada acima não apresenta essa propriedade, pois para que seja adaptável a todos os sinais seria necessário um conjunto de funções base de ordem no mínimo exponencial ( $\Omega(2^N)$ ).

Dentre as transformadas usadas para a codificação de imagens, destacam-se a DCT (*Discrete Cosine Transform*) e as transformadas *wavelet*. A DCT é uma base ortonormal formada por cossenos em diferentes fases. Ela provem de um caso particular da transformada de Fourier discreta, considerando os dados como periódicos, reais e positivos, de forma que os elementos em quadratura (senos) são nulos. A forma mais usada na compressão de imagens é a DCT-II em 2 dimensões, definida por: [19]

$$F[x, y] = \frac{C(x)C(y)}{\sqrt{MN}} \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} f[j, k] \cos\left(\frac{(2j+1)\pi x}{2\sqrt{MN}}\right) \cos\left(\frac{(2k+1)\pi y}{2\sqrt{MN}}\right) \quad (2.5)$$

Onde  $F$  representa o sinal no domínio transformado. Os coeficientes  $C(n)$  são dados por:

$$C(n) = \begin{cases} 1, & n = 0 \\ \sqrt{2}, & n \neq 0 \end{cases} \quad (2.6)$$

As funções base dessa transformação são mostradas na Figura 2.1. Nessa figura, cada um dos  $8 \times 8$  quadrados é uma imagem formada pelo produto de cossenóides 2D isotrópicas em uma das possíveis frequências.

A imagem é recuperada através da transformação inversa IDCT (*Inverse Discrete Cosine Transform*), definida por (também chamada de DCT-III): [19]

$$f[x, y] = \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} \frac{C(x)C(y)}{\sqrt{MN}} F[j, k] \cos\left(\frac{(2j+1)\pi x}{2\sqrt{MN}}\right) \cos\left(\frac{(2k+1)\pi y}{2\sqrt{MN}}\right) \quad (2.7)$$

Os coeficientes  $C(n)$  são os mesmos anteriormente definidos.

No codificador, o estágio de transformadas é usualmente seguido por um estágio de quantização, que remove informações sem influência significativa na imagem reconstruída, tornando a reconstrução da imagem inexata. O nível de compressão e a perda na qualidade dependem do número de níveis discretos produzidos pela quantização.



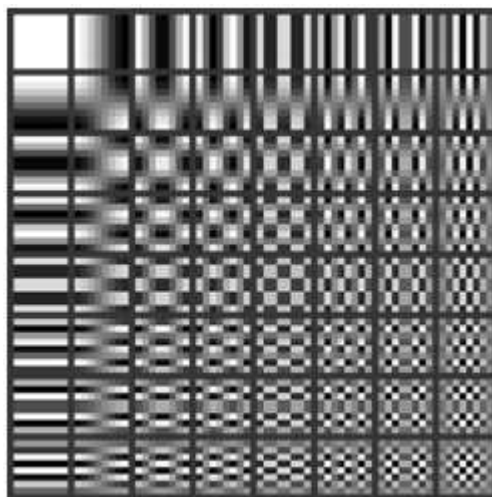


Fig. 2.1: Funções base da DCT 2D.

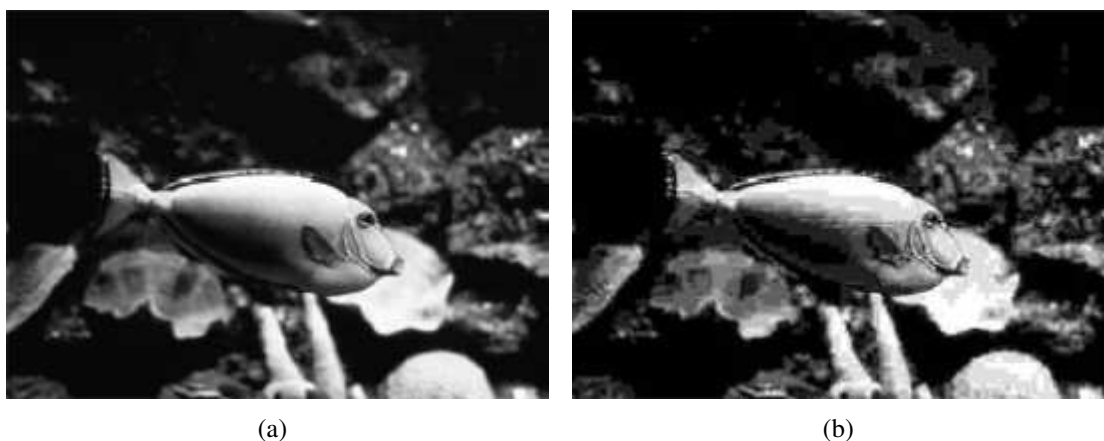


Fig. 2.2: Efeito de bloco em imagem codificada DCT (a) original (b) recuperada. [21]

A degradação mais conhecida é o efeito de bloco, mostrado na Figura 2.2, caracterizado por padrões quadrados, mais perceptíveis em áreas planas da imagem. Nas aplicações, a DCT é normalmente operada sobre blocos de imagem de tamanho fixo  $8 \times 8$  ou  $4 \times 4$ . Essa divisão em blocos permite que os dados de entrada possuam uma maior correlação interna, facilitando a codificação de entropia e eliminação de redundâncias, mas as descontinuidades se tornam visíveis quando a quantização é pouco refinada (baixa taxa de *bits*). [20]

O padrão JPEG (*Joint Photographs Expert Group*) foi criado em 1987 [22] e ainda é muito utilizado para a compressão de imagens estáticas, incluído em muitas câmeras digitais comerciais, juntamente com as variações do MPEG (*Moving Pictures Expert Group*) para vídeo. Ambos os esquemas se utilizam da DCT como bloco principal. Padrões mais modernos têm tentado resolver os problemas da transformas e garantir melhor qualidade visual em relação à taxa de *bits* (*bitrate*).

O H.264 ou MPEG-4 Part 10 AVC [23] é outro padrão de compressão de vídeo bastante usado atualmente, tendo sido incluído no padrão de televisão digital brasileiro ISDB-Tb (*Integrated Services Digital Broadcasting, Terrestrial, Brazilian version*) [24], e que consegue taxas de compressão

muito maiores do que os padrões MPEG anteriores. Além da DCT, o H.264 utiliza também estágios de predição de movimento, predição entre blocos dentro de um mesmo quadro (predição intra) e filtragem para remover blocos. A DCT utilizada é uma versão modificada, que realiza quantização de forma integrada e otimiza o cálculo com estágios aritméticos usando números inteiros. O padrão completo exige um sistema de alta capacidade computacional e por isso, a maioria das aplicações utiliza apenas as versões mais simples do padrão. Devido aos problemas de artefatos visuais da DCT e ao peso computacional, o H.264 completo é mais indicado para vídeos em resolução HDTV (*High Definition Television*), com altas taxas de *bits*, inviável para aplicações móveis.[25]

O HEVC (*High Efficiency Video Coding*), também chamado de H.265 e de MPEG-H Part 2, é o mais novo padrão de compressão de vídeo e foi, assim como o H.264, foi idealizado por uma colaboração entre os grupos de pesquisa MPEG e VCEG (*Video Coding Experts Group*). Sua especificação terminou de ser formalizada em abril de 2013. As principais diferenças são em relação ao número de modos de predição intra e no modo como a DCT é calculada. De acordo com os estudos iniciais [26], o HEVC consegue ganhos de até 50% mais compressão em relação ao H.264 para aplicações de vídeo sob demanda, ou seja, mesma qualidade com metade da taxa de *bits*. Para codificação em tempo real, as expectativas são de uma redução de 35% da largura de banda a níveis de qualidade similares. De forma alternativa, o HEVC pode permitir filmes em maior resolução (UHDTV - *Ultra High Definition Television*), tais como 2K e 4K. Várias empresas já anunciaram decodificadores de software, mas não está claro o quanto de potência será necessário para conduzi-los. As metas originais para HEVC foram  $10\times$  complexidade de codificação e  $2\times$  a  $3\times$  complexidade de decodificação, em comparação com H.264. No entanto, a reprodução H.264 é acelerada por hardware na maioria das plataformas, incluindo o uso de GPU em computadores. Além disso, várias empresas têm patentes relacionadas à tecnologia HEVC e planejam cobrar *royalties* dos usuários.

Um problema encontrado na aplicação da DCT é que, no caso de dados computacionais como as imagens, tenta-se aproximar uma função de domínio limitado a um conjunto de funções de domínio infinito (estacionárias), como demonstrado em [21]. Muitos sinais requerem uma aproximação mais flexível, onde seja possível variar o tamanho da janela a fim de determinar mais precisamente tempo e frequência. A análise *wavelet* representa um passo nessa direção: uma técnica de janelamento com regiões de tamanho variável. A análise *wavelet* permite o uso de intervalos longos de dados onde existe informação de baixa frequência e intervalos menores de dados para informação de alta frequência, pois suas funções-base são limitadas. Com isso, elementos transientes de um sinal podem ser representados mais eficientemente por uma quantidade de informação menor.

## 2.2 A Transformada *Wavelet*

As Transformadas *Wavelets* (TW) foram desenvolvidas como fruto das contribuições de pesquisadores das mais diversas áreas (matemática, física, estatística e engenharia), tendo como destaque os inúmeros trabalhos de I. Daubechies [27] e S. Mallat [3] e apresentam desempenho custo-benefício consideravelmente superior ao desempenho das demais transformadas. Foram desenvolvidas como uma evolução da decomposição em sub-bandas e da análise multirresolução de funções. A transformada *wavelet* apresenta algoritmos rápidos, boa capacidade de concentrar a energia do sinal em poucos coeficientes, transmissão progressiva, além de não introduzir artefatos de blocagem.

A codificação por sub-banda realiza uma análise tempo-frequência do sinal na qual o sinal é

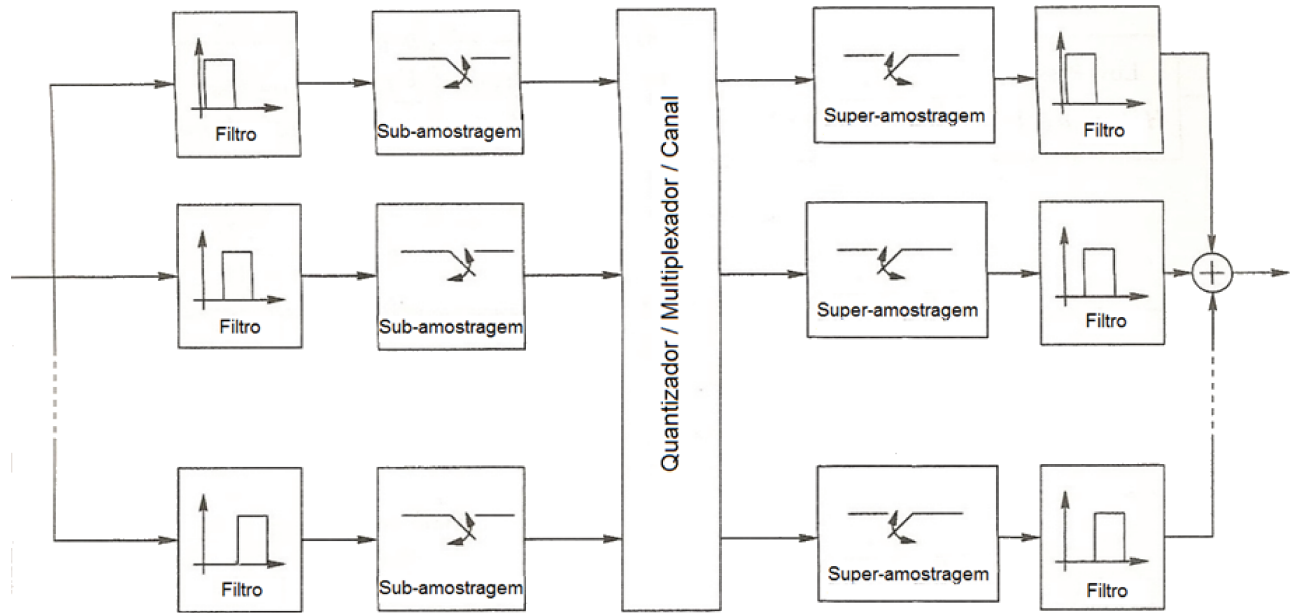


Fig. 2.3: Esquema de codificação por sub-banda. [28]

particionado em um número de diferentes bandas de frequência. Podem ser adotados  $M$  intervalos disjuntos de comprimento  $2s_N/M$  produzindo  $M$  sinais com sub-banda  $s_N/M$  cada, onde para um sinal limitado em banda com componente frequencial nula  $F(s) = 0$  e  $|s| \geq s_N$ :

$$s_{Nyquist} = \frac{1}{\Delta x} = 2 \cdot s_n \quad (2.8)$$

Essas sub-bandas podem ainda ser quantizadas independentemente, de acordo com a aplicação. Tal procedimento permite que codificadores identifiquem as sub-bandas de maior concentração de energia, facilitando uma alocação de *bits* apropriada.

Considerando um sinal unidimensional e supondo dois canais de sub-banda, o intervalo de frequências é particionado no ponto  $s_N/2$  submetendo o sinal a dois filtros que geram dois sinais de sub-banda: um sinal meia-banda baixa e um sinal meia-banda alta. A Figura 2.3 mostra o esquema de codificação por sub-banda de forma geral.

Na Figura 2.4, os passos de subamostragem-filtragem e superamostragem-filtragem nos lados do codificador e decodificador, respectivamente, são mostrados para um sinal unidimensional  $f(x)$ . Sabendo que  $y_1(i\Delta x)$  contém exatamente a informação de alta frequência que foi eliminada de  $f(i\Delta x)$  durante a geração de  $y_0(i\Delta x)$ , pode-se dizer que  $y_1(i\Delta x)$  e  $y_0(i\Delta x)$  contêm juntos toda a informação presente no sinal original  $f(i\Delta x)$  [29]. Assim, tem-se: [30]

$$f(i\Delta x) = y_0(i\Delta x) + y_1(i\Delta x) = f(i\Delta x) * h(i\Delta x) + f(i\Delta x) * g(i\Delta x) \quad (2.9)$$

pois:

$$H(s) + G(s) = 1 \quad (2.10)$$

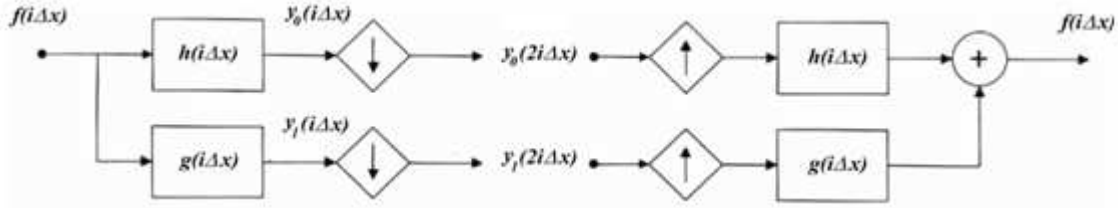


Fig. 2.4: Codificação por sub-banda de duas bandas e reconstrução. [30]

Assim, o sinal original  $f(i\Delta x)$  de  $N$  pontos pode ser representado sem redundância em dois sinais de  $N/2$  pontos. Como  $y_0(i\Delta x)$  e  $y_1(i\Delta x)$  podem ser completamente recuperados a partir dos sinais codificados em sub-banda,  $f(i\Delta x)$  pode ser reconstruído sem erros no decodificador.

Portanto, a codificação por sub-banda em dois canais, do lado do codificador, requer somente a filtragem de  $f(i\Delta x)$  com  $h(i\Delta x)$  e  $g(i\Delta x)$ , seguidas pela subamostragem de cada saída: [30]

$$y_0(2i\Delta x) = \sum_k f(k\Delta x) \cdot h((-k + 2i)\Delta x) \quad (2.11)$$

$$y_1(2i\Delta x) = \sum_k f(k\Delta x) \cdot g((-k + 2i)\Delta x) \quad (2.12)$$

Para a reconstrução do sinal original, basta superamostrar os dois sinais codificados em sub-banda, interpolá-los com  $2h(i\Delta x)$  e  $2g(i\Delta x)$ , e em seguida, somá-los, ou seja: [30]

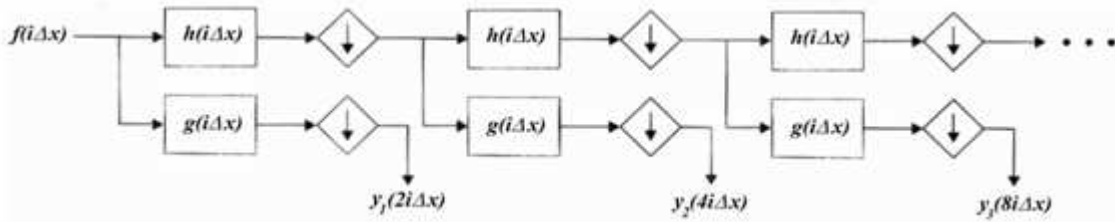
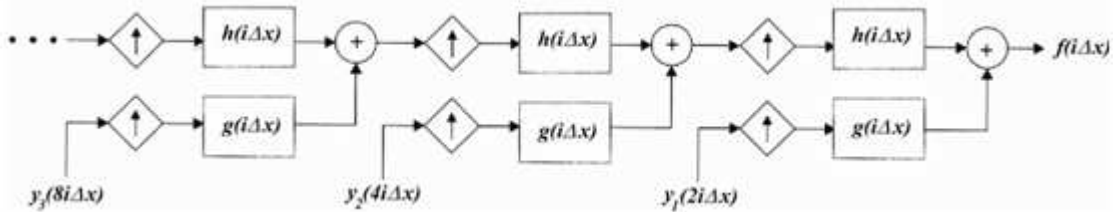
$$f(i\Delta x) = 2 \sum_k [y_0(2k\Delta x) \cdot h((-i + 2k)\Delta x) + y_1(2k\Delta x) \cdot g((-i + 2k)\Delta x)] \quad (2.13)$$

O projeto dos filtros adotados deve ser feito de forma a evitar que os trechos do espectro sobrepostos resultem na distorção bem conhecida como *aliasing* (sobreposição), que não permite recuperação.

### 2.2.1 Algoritmo de Mallat

Mallat [10] definiu um algoritmo iterativo para a TW, mais eficiente do que calcular o conjunto completo de produtos internos. Em seu algoritmo, após aplicar a codificação por sub-banda em dois canais ao sinal  $f(i\Delta x)$ , o sinal meia-banda baixa  $y_0(i\Delta x)$  é novamente sujeito à mesma codificação por sub-banda em dois canais. Desse procedimento derivam um sinal meia-banda alta com  $N/2$  coeficientes e dois sinais sub-banda com  $N/4$  coeficientes. A cada passo é retido o sinal meia-banda alta e processado o sinal meia-banda baixa, até que reste apenas um único coeficiente (Figura 2.5). Os coeficientes transformados são então o coeficiente banda-baixa e o conjunto de coeficientes meia-banda alta codificados. No final, tem-se um total de  $N$  coeficientes.

Cada conjunto de coeficientes transformados é obtido por meio da convolução de  $f(i\Delta x)$  repetidamente com  $h(i\Delta x)$  e somente uma vez com  $g(i\Delta x)$ . Daí, as funções-base da TW,  $\psi_{j,n}(x)$  serem a reflexão de  $g(i\Delta x)$  e de outras funções derivadas da convolução de  $g(i\Delta x)$  repetidamente com  $h(i\Delta x)$ . [30]. O processo de reconstrução de  $f(i\Delta x)$  encontra-se esquematizado na Figura 2.6.

Fig. 2.5: Algoritmo de Mallat para decomposição *wavelet*. [30]Fig. 2.6: Algoritmo de Mallat para reconstrução *wavelet*. [30]

A Figura 2.7 apresenta um exemplo da aplicação do algoritmo de Mallat, gerando os coeficientes da aproximação para várias escalas diferentes a partir de um mesmo sinal contínuo. Essas aproximações podem ser calculadas diretamente a partir de  $f(x)$ , ou uma a partir da outra (sempre no sentido da maior para a menor resolução).

## 2.3 Bases *Wavelet* Separáveis

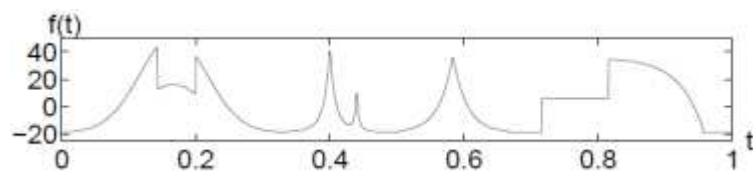
Implementações da transformada *wavelet* em 2 dimensões são normalmente feitas de forma isotrópica, isto é, com distribuição independente entre as duas dimensões. São chamadas também de *wavelets* separáveis, pois podem ser escritas como um produto tensorial, similar ao tradicionalmente feito para outras bases 2D como a de Fourier.

A qualquer base *wavelet* ortonormal  $\{\psi_{j,n}(x,y)\}_{(j,n) \in \mathbb{Z}^2}$  de  $L^2(\mathbb{R})$ , pode-se associar uma base *wavelet* ortonormal separável de  $L^2(\mathbb{R}^2)$ :

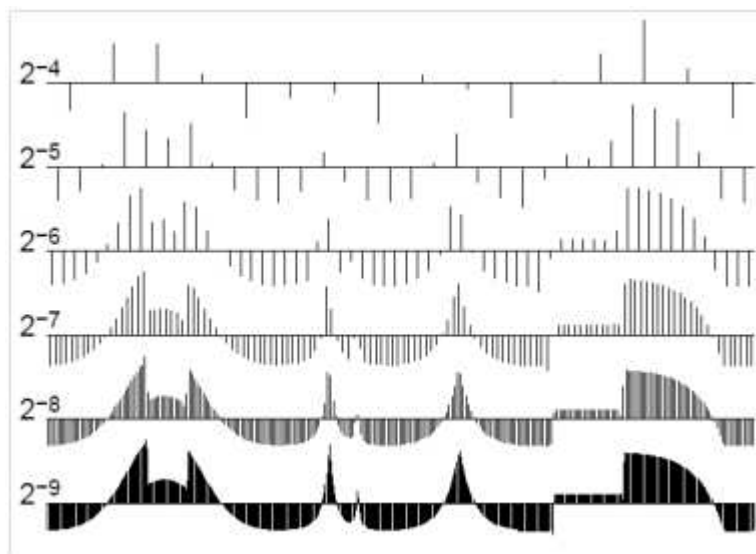
$$\{\psi_{j_1;n_x}(x) \psi_{j_2;n_y}(y)\}_{(j_1,j_2;n_1,n_2) \in \mathbb{Z}^4} \quad (2.14)$$

Decomposições do sinal em bases *wavelet* bidimensionais são calculadas com uma extensão separável de algoritmos unidimensionais de Mallat. A representação *wavelet* ortogonal 2D de uma imagem original pode ser calculada por meio de sucessivas decomposições de  $A_{2^{j+1}}^d \cdot f$  em  $A_{2^j}^d \cdot f$ ,  $D_{2^j}^1 \cdot f$ ,  $D_{2^j}^2 \cdot f$  e  $D_{2^j}^3 \cdot f$ , sendo  $-J \leq j \leq -1$ . Um estágio completo desse algoritmo encontra-se na Figura 2.8. A representação *wavelet* 2D pode, então, ser calculada por meio de dois algoritmos piramidais 1D em sequência: o primeiro aplicado nas linhas da imagem, e o segundo aplicado às colunas [30].

As funções  $\psi_{j_1;n_x}(x) \psi_{j_2;n_y}(y)$  misturam informações em duas escalas diferentes  $2^{j_1}$  e  $2^{j_2}$  ao longo de  $x$  e  $y$ , aspecto frequentemente desejável de ser evitado [3]. A multiresolução separável leva

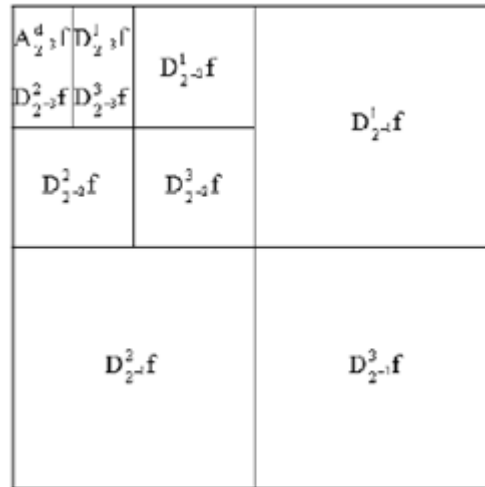


(a)



(b)

Fig. 2.7: Exemplo da aplicação do algoritmo de Mallat: (a) Sinal original (b) Aproximações multiresolução discretas nas escalas  $2^{-j}$  calculadas com *wavelets splines* cúbicas. [3]

Fig. 2.8: Decomposição *wavelet* ortogonal 2D para  $J = 3$ . [30]

a outra construção de bases *wavelet* separáveis com *wavelets* que são produtos de funções dilatadas de mesma escala. O tratamento aparentemente independente entre os dois termos da análise multi-resolução separável pressupõe uma propriedade de isotropia nas funções *wavelet*, isto é, distribuição equivalente em frequência para todas as direções.

A imagem  $A_{2^{-j}}^d \cdot f$  é a aproximação de  $A_1^d \cdot f$  com nível de resolução  $2^{-j}$ , correspondendo às frequências mais baixas de  $f$ .  $\{D_{2^j}^i \cdot f\}_{1 \leq i \leq 3}$  são as imagens de detalhamento para diferentes níveis de resoluções onde:  $D_{2^j}^1 \cdot f$  corresponde às frequências verticais altas (linhas horizontais na imagem),  $D_{2^j}^2 \cdot f$  corresponde às frequências horizontais altas (linhas verticais na imagem) e  $D_{2^j}^3 \cdot f$  corresponde às frequências altas em ambas as direções. Se a imagem original possui  $N$  pixels, cada imagem  $A_{2^{-j}}^d \cdot f$ ,  $D_{2^j}^1 \cdot f$ ,  $D_{2^j}^2 \cdot f$  e  $D_{2^j}^3 \cdot f$  da representação possui  $2^{2j} \cdot N$  pixels, totalizando os mesmos  $N$  pixels da imagem original [30]. Na Figura 2.9 é apresentado um exemplo evidenciando as frequências verticais, horizontais e diagonais aparecendo nas subimagens separadamente, conforme a disposição já apontada na Figura 2.8

O nível de aproximação  $A_{2^{j+1}}^d \cdot f$  pode ser reconstruído fazendo a inserção de uma coluna de zeros entre cada coluna de  $A_{2^j}^d \cdot f$ ,  $D_{2^j}^1 \cdot f$ ,  $D_{2^j}^2 \cdot f$  e  $D_{2^j}^3 \cdot f$ , seguida pela convolução das linhas resultantes com um filtro 1D. A seguir, há a inserção de uma linha de zeros entre cada linha, seguida pela convolução das colunas resultantes com outro filtro 1D. Cada coeficiente de  $A_{2^{j+1}}^d \cdot f$  é formado por todos os coeficientes de  $A_{2^j}^d \cdot f$ ,  $D_{2^j}^1 \cdot f$ ,  $D_{2^j}^2 \cdot f$  e  $D_{2^j}^3 \cdot f$ . A imagem  $f$  é reconstruída repetindo-se esse processo para  $-1 \leq j \leq -J$ , conforme mostrado na Figura 2.10.

## 2.4 Aproximações de Imagens

As transformadas ortogonais têm limitações na capacidade de concentrar dados em poucas componentes para sinais com descontinuidades. O erro de aproximação depende da regularidade do sinal tratado. Para sinais suaves, aproximações lineares e não-lineares têm desempenho similar [3]. Quando a regularidade do sinal não é uniforme, aproximações não-lineares em bases *wavelet* são mais apropriadas. Para aproximar funções suaves por partes e com geometria não-separável, novas

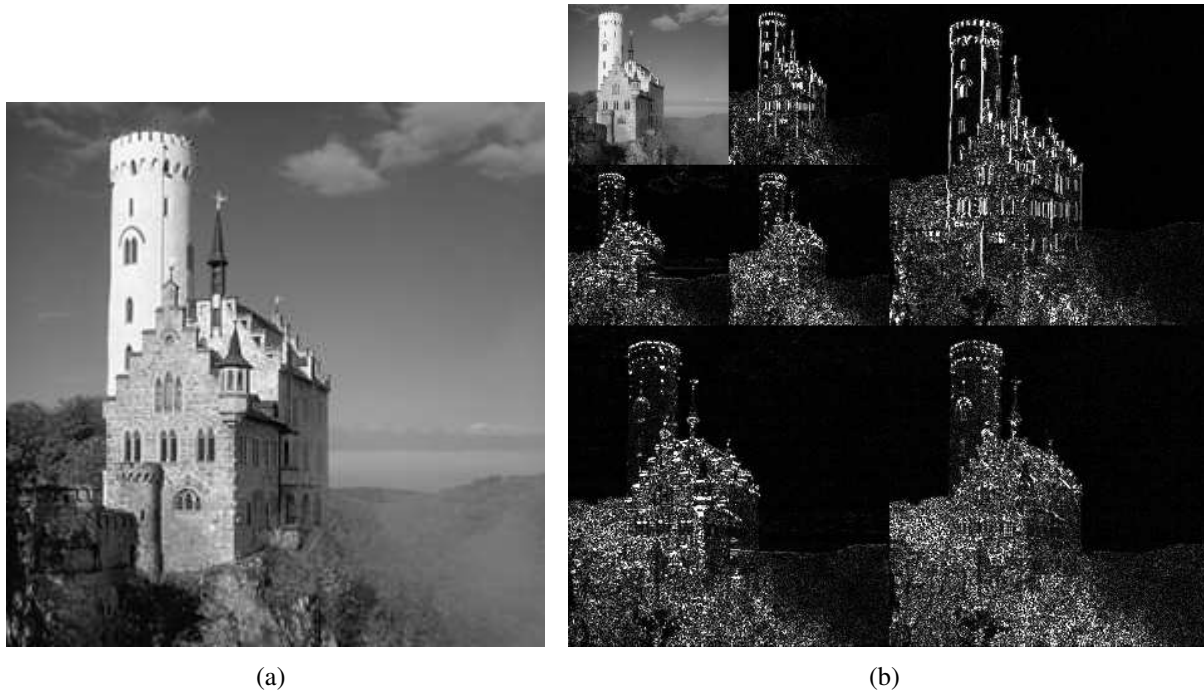


Fig. 2.9: Exemplo de decomposição *wavelet*: (a) Imagem original (b) Decomposição *wavelet* Daubechies separável (passa-alta foi ressaltada para comparação).

transformadas e dicionários redundantes, que garantem representação mais esparsa, foram propostos na literatura e serão vistos no Capítulo 3. A seguir uma comparação das duas formas de aproximação em usando transformadas para imagens.

### 2.4.1 Aproximações Lineares

Seja  $\{\varphi_n(x) = \varphi_s(x - ns)\}_{0 \leq n < N}$  uma base ortonormal de um espaço  $A_N \subset L^2[0, 1]$ . Considerando um sinal analógico  $f(x)$  de suporte compacto normalizado em  $[0, 1]$ , a melhor aproximação de  $f$  em  $A_N$  é a projeção ortogonal  $f_N$  de  $f$  em  $A_N$ , recuperado pela base  $\{\varphi_n(x)\}_{0 \leq n < N}$ .

$$f_N(x) = \sum_{n=0}^{N-1} \langle f, \varphi_n \rangle \varphi_n(x) \quad (2.15)$$

A questão central é determinar quantos coeficientes *wavelet* são necessários para representar um determinado sinal até um dado erro de aproximação tolerado. Em aproximações lineares, as  $N$  *wavelets* em  $A_N$  são escolhidas a priori representando a  $N$ -ésima escala  $A_N = \text{span} \{\varphi_{n,-N}\}_{n=0}^{N-1}$ .

Para estimar o erro de aproximação  $\|f - f_N\|$ , considere uma base ortonormal  $B = \{g_m\}_{m \in \mathbb{N}}$  de  $L^2[0, 1]$  com  $N$  vetores  $\{g_m\}_{0 \leq m < N}$  formando uma base ortogonal do mesmo espaço  $A_N$ . Como mostrado em [3], aproximações utilizando bases de Fourier ou *wavelets* são ótimas para sinais que pertençam a espaços definidos por:



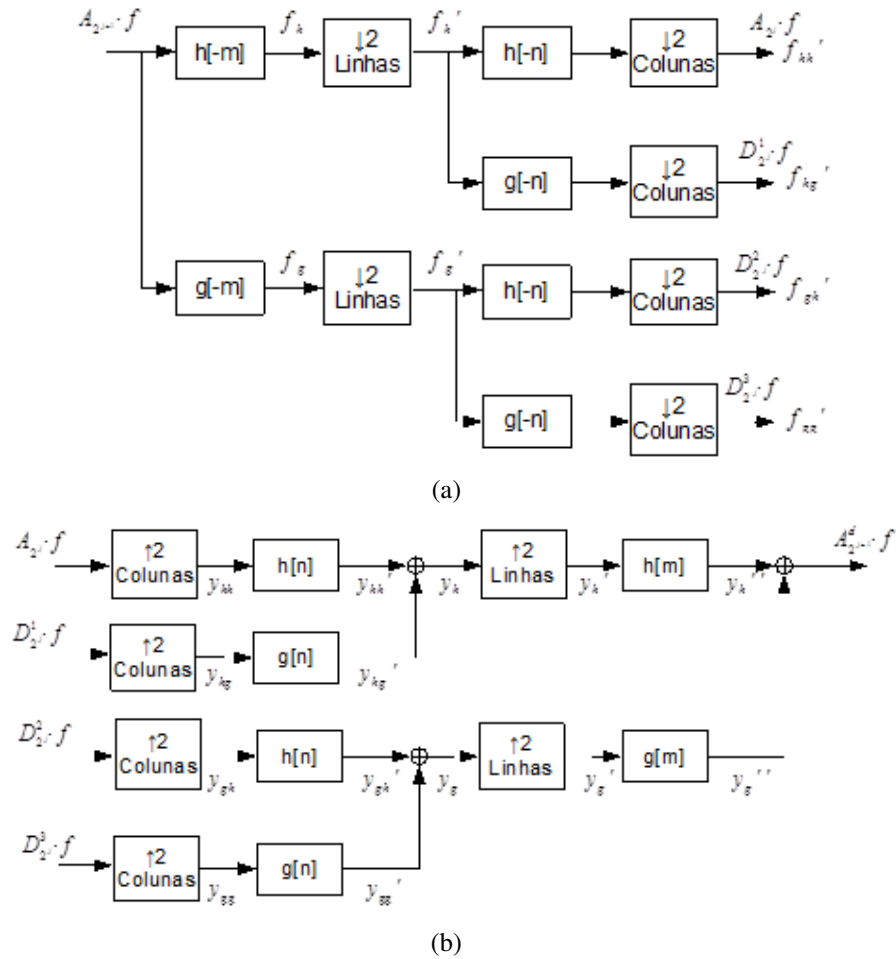


Fig. 2.10: Diagrama em blocos para imagem de: (a) Um estágio de decomposição *wavelet* (b) Um estágio de reconstrução *wavelet*. [30]

$$\mathbf{W}_{B,s} = \left\{ f \in \mathbf{H} : \sum_{m=0}^{+\infty} m^{2s} |\langle f, g_m \rangle|^2 < +\infty \right\} \quad (2.16)$$

Quando  $B$  é uma base de Fourier ou *wavelets*, a equação acima define  $\mathbf{W}_{B,s}$  como espaços de Sobolev:

$$\mathbf{W}_{B,s} = \left\{ f \in \mathbf{H} : \sum_{m=0}^{+\infty} m^{2s} \left| \frac{\partial^k f}{\partial x^k} \right|^2 < +\infty, 0 \leq k \leq s \right\} \quad (2.17)$$

Como mostrado no Teorema a seguir [3], o erro provocado por essas aproximações segue  $\epsilon_l(N, f) = O(N^{-2s})$ , onde  $N$  é o número de componentes utilizadas.

**Teorema 1.** Para algum  $s > 1/2$ , existem  $A, B > 0$  tais que  $\sum_{m=0}^{+\infty} |m|^{2s} |\langle f, g_m \rangle|^2 < +\infty$ , então:

$$A \sum_{m=0}^{+\infty} |m|^{2s} |\langle f, g_m \rangle|^2 \sum_{N=0}^{+\infty} \epsilon_l(N, f) \leq B \leq \sum_{m=0}^{+\infty} |m|^{2s} |\langle f, g_m \rangle|^2 \quad (2.18)$$

Para sinais mais complexos, que possuam descontinuidades bem localizadas, a aproximação linear de  $f$  pelas primeiras  $N$  componentes não é necessariamente a melhor aproximação. Nesse caso,  $f \notin \mathbf{W}^2[0, 1]$  para qualquer  $s > 1/2$ , portanto  $\epsilon_l(N, f)$  decai com  $O(N^{-1})$ . A Figura 2.11 [3] mostra um exemplo de sinal descontínuo com variação limitada aproximado pelas componentes de alta escala da decomposição *wavelet*. Nota-se a existência de erros nas proximidades das descontinuidades, devido à insuficiência de componentes para representá-las. Na Figura 2.11(a), a aproximação é dada pelas  $N = 128$  componentes Fourier de mais baixa frequência com relação sinal-ruído de  $8,63 \times 10^{-2}$ . Na Figura 2.11(b), a aproximação é dada pelos coeficientes *wavelet* Daubechies 4 de maior escala com  $N = 128$  e a relação sinal-ruído é de  $8,58 \times 10^{-2}$ .

Para sinais estacionários, modelados por um processo aleatório de média nula, a base que minimiza o erro quadrático médio para aproximações lineares é a transformada de Karhunen-Loève (KLT - Karhunen-Loève Transform). Caso o sinal  $f$  não tenha média nula, subtrai-se o valor da média amostral  $E[f]$  antes da transformação e o adiciona novamente após a transformação inversa. Essa transformada concentra toda a energia de um sinal de dimensões finitas em poucos coeficientes bem localizados (correspondente a uma matriz diagonal), de forma a diagonalizar a matriz de covariância do sinal.

A transformação  $\Gamma$  que satisfaz a Equação 2.19 é a transformada cujos vetores-base (amostras da função-base) são os autovetores da matriz de covariância e, por consequência, os elementos da diagonal da matriz de covariância transformada são os autovalores da matriz de dados original.

$$Cov(F) = \Gamma^{*T} Cov(f) \Gamma^* \quad (2.19)$$

Onde  $()^T$  indica a matriz transposta e  $()^*$  indica o operador complexo conjugado.

Em termos práticos, essa transformada é difícil de ser implementada, pois necessita que sejam calculados os autovetores da matriz  $Cov(f)$  a cada nova amostra de imagem e, dependendo da dimensão dos dados, pode consumir muito tempo computacional. Transformadas com coeficientes fixos (como

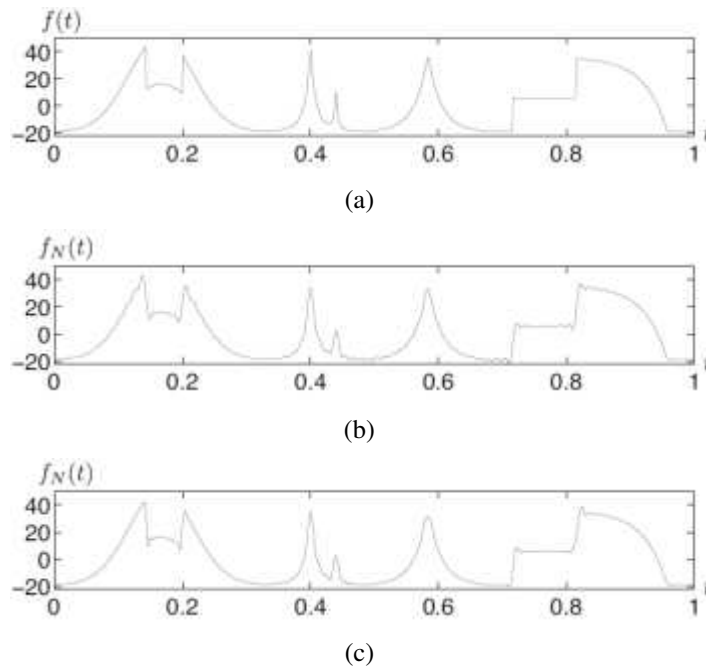


Fig. 2.11: Reconstrução de um sinal unidimensional de variação limitada (a) Sinal original (b) Aproximação Fourier (c) Aproximação por *wavelet* Daubechies 4. [3]

a DCT) são muito mais fáceis de serem implementadas e atingem resultados muito próximos à KLT. Além disso, quando  $f$  não é um processo estacionário gaussiano (ou não o é em todo domínio e possui descontinuidades), esse tipo de transformada (bem como outras transformadas baseadas em funções estacionárias como as bases de Fourier) não é eficiente, pois os autovetores da matriz de covariância têm a mesma ordem de magnitude e  $f$  não terá a energia concentrada em poucos coeficientes.

### 2.4.2 Aproximações Não-Lineares

Para sinais mais complexos, que possuem descontinuidades bem localizadas, a aproximação linear de  $f$  pelas  $N$  primeiras componentes não é necessariamente a melhor aproximação. Usando aproximações não-lineares,  $A_N$  consiste de  $N$  *wavelets* em diferentes escalas e localizações. O erro de uma aproximação não-linear é calculado a partir dos coeficientes não utilizados na aproximação:

$$\hat{f}[n] = \sum_{m \in A_n} \langle f, g_m \rangle g_m[n] \quad (2.20)$$

$$\|f - \hat{f}\|^2 = \sum_{m \notin A_n} |\langle f, g_m \rangle|^2 \quad (2.21)$$

Como descrito na Equação 2.20, a não-linearidade está no método de escolha dos coeficientes *wavelet* pelos maiores, sendo portanto adaptativo ao sinal analisado. Modelando o sinal como uma função suave por partes, pode-se mostrar que *wavelets* oferecem a melhor taxa de aproximações não-lineares. Quando o sinal  $f$  contém descontinuidades, o erro assintótico segue  $N^{-1}$  e  $N^{-2}$  para a aproximação linear e não-linear, respectivamente.

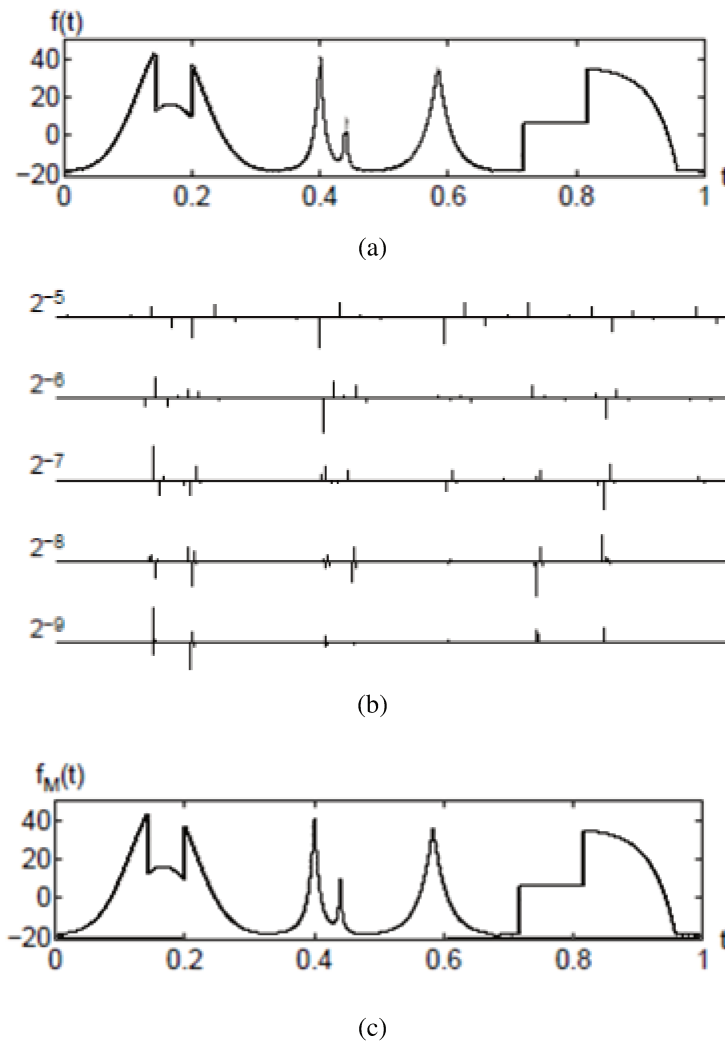


Fig. 2.12: Reconstrução de um sinal unidimensional (a) Sinal original (b) Decomposição *wavelet*, com os  $N/6$  maiores coeficientes *wavelet* Symmlet 4 (c) Aproximação não-linear. [3]

A Figura 2.12 mostra a aproximação não-linear *wavelet* de um sinal regular por partes. Impulsos de Dirac representam coeficientes *wavelet* com coeficientes positivos ou negativos de amplitude acima de um dado limiar  $T$ . Os maiores coeficientes estão nas proximidades das discontinuidades. Nesse caso, o erro de aproximação é 17 vezes menor do que para aproximações lineares. [3] Na Figura 2.13, é mostrada uma comparação entre aproximações lineares e não-lineares para um sinal de imagem.

## 2.5 O Problema das Transformadas Isotrópicas

A aproximação por transformadas *wavelet* consegue bons resultados quando são tratados sinais unidimensionais, mas problemas surgem na aplicação para imagens devido à importância da geometria na qualidade visual. Embora as bases *wavelet* se adaptem a discontinuidades pontuais, como *wavelets*  $d$ -dimensionais são normalmente construídas com  $2^d - 1$  produtos tensoriais de *wavelets*

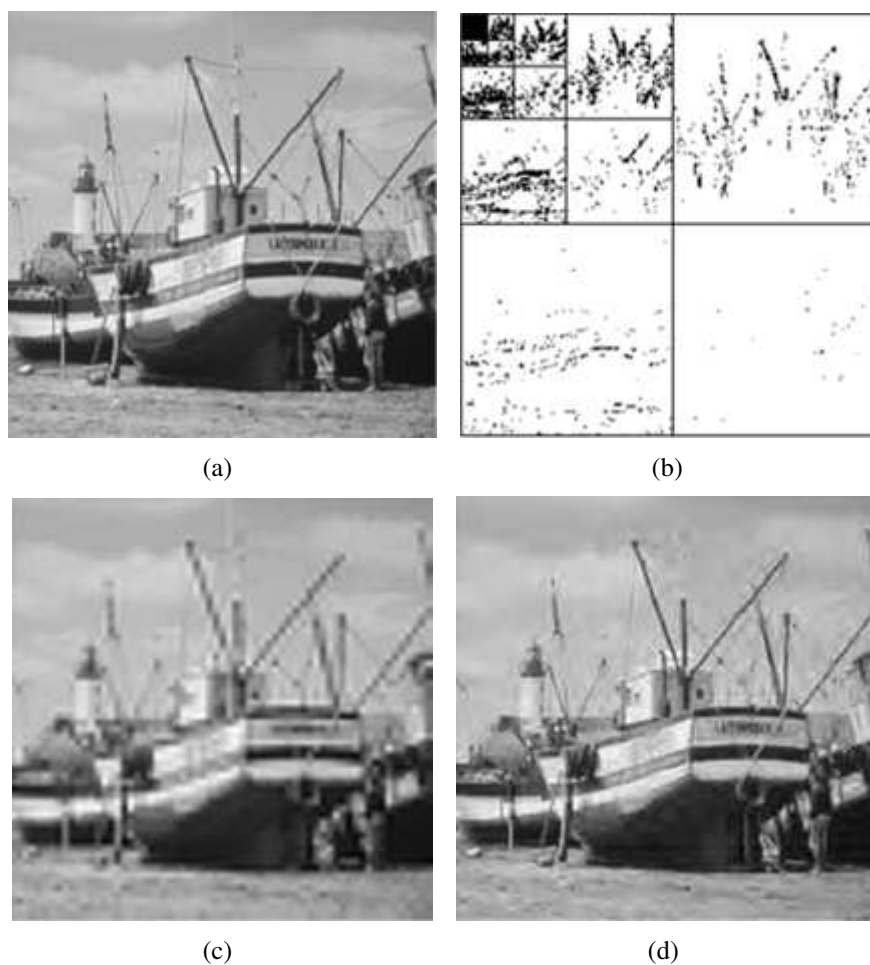


Fig. 2.13: Esquemas de aproximação (a) Imagem original ( $N = 2562$ ) (b) Decomposição *wavelet*, pontos escuros correspondem a  $|\langle f, \psi_{j,n}^k \rangle| > T$  (c) Aproximação linear com  $N/16$  (d) Aproximação não-linear com  $N/16$ . [3]

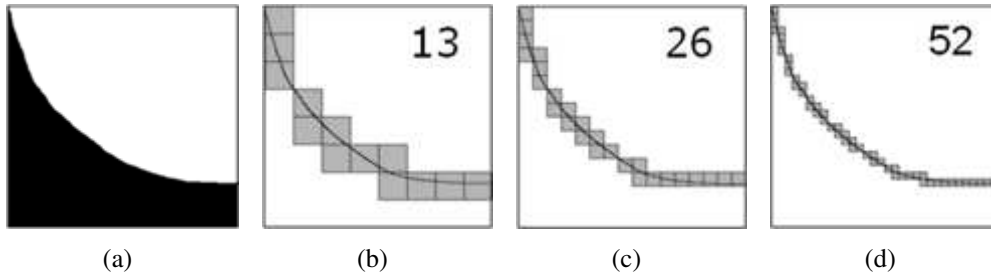


Fig. 2.14: Ineficiência da *wavelet* isotrópica em representar a geometria de uma imagem. (a) Imagem original (b) 13 (c) 26 (d) 52. [31]

unidimensionais (separáveis), descontinuidades em domínios de maior dimensão (como contornos, no caso de imagens) podem estar organizadas ao longo de subconjuntos de acordo com a geometria. A Figura 2.14 mostra a ineficiência da decomposição *wavelet* em representar contornos por não explorar a regularidade dos mesmos. Muitos coeficientes *wavelet* são necessários para aproximar a descontinuidade, que se espalham por todos os níveis de decomposição, mas que poderiam ser reduzidos conhecendo-se a geometria.

Para melhor ilustrar a baixa taxa de decaimento do erro de aproximações em sinais bidimensionais, tem-se na Figura 2.15(a) uma imagem simples de variação limitada. A imagem é definida como  $f = C \cdot \mathbf{1}_\Omega$  e a parte escura identifica um subconjunto  $\Omega$ . A Figura 2.15(b) mostra os suportes das bases *wavelet* quando é feita a projeção da imagem, com o tamanho dos pontos de acordo com a magnitude dos respectivos coeficientes *wavelet*. Se o suporte da *wavelet*  $\psi_{j,n}^l$  não intercepta a borda  $\partial\Omega$ , então  $\langle f, \psi_{j,n}^l \rangle = 0$ . As *wavelets*  $\psi_{j,n}^l$  têm um suporte quadrado de tamanho proporcional a  $2^j$ , que é mostrado na Figura 2.15(b) por uma grade. A borda  $\partial\Omega$  tem comprimento finito  $L$  e portanto existem da ordem de  $L \cdot 2^{-j}$  *wavelets* interceptando  $\partial\Omega$ . Como  $f$  tem variação limitada,  $\langle f, \psi_{j,n}^l \rangle = O(C \cdot 2^j)$ , sendo  $\langle f, \psi_{j,n}^l \rangle \approx C \cdot 2^j$  ao longo da borda. Assim, os  $N$  maiores coeficientes estão nas escalas  $2^j \geq L/N$ , resultando em um erro de aproximação não-linear de:

$$\epsilon_{nl}(N, f) \approx \sum_{j=-\inf}^{\log_2(L/N)-1} L 2^{-j} C^2 2^{2j} = (CL)^2 N^{-1} \quad (2.22)$$

Assim, o número excessivo de coeficientes *wavelet* produzidos ao longo dos contornos limita o decaimento do erro em  $N^{-1}$ .

Métodos baseados em transformadas *wavelet* não apresentam efeito de blocos (os erros de quantização de um coeficiente não se espalham, pois advêm de uma representação compacta, não-estacionária), mas introduzem outros artefatos como *ringing* e embaçamento (*blurring*). O efeito de *ringing*, também chamado de efeito de Gibbs, é causado pela quantização ou truncamento dos coeficientes de alta frequência necessários à representação de um sinal com descontinuidades. São bastante perceptíveis na vizinhança de contornos da imagem. Em geral, artefatos de *ringing* são mais difíceis de caracterizar do que efeito de blocos. O embaçamento é um artefato que também surge pela falta de componentes de alta frequência, do lado oposto ao *ringing* (Figura 2.16). A Figura 2.17 mostra o efeito de *ringing* para a imagem “Lena” quando é feita a codificação TW com poucos coeficientes.

Por essas razões e para tratar a deficiência da TW pura em explorar a geometria, outros tipos de representação de sinais, incluindo dicionários redundantes, tem sido o foco de novos avanços em

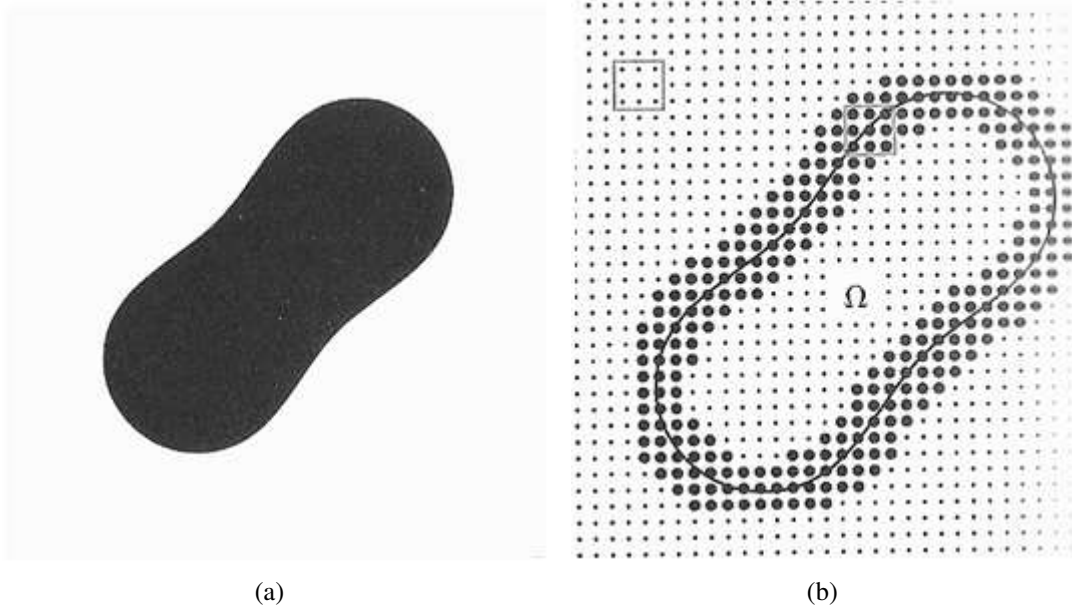


Fig. 2.15: Projeção *wavelet* (a) Imagem  $f = C \cdot \mathbf{1}_\Omega$  (b) Na escala  $2^j$ , as *wavelets*  $\psi_{j,n}^l$  têm um suporte quadrado de largura proporcional a  $2^j$ . Esse suporte corresponde a uma grade de intervalo  $2^j$ , indicada pelos pequenos pontos. Pontos maiores indicam *wavelets* com suporte que cruza a fronteira de  $\Omega$  para  $\langle f, \psi_{j,n}^l \rangle \neq 0$ . [3]

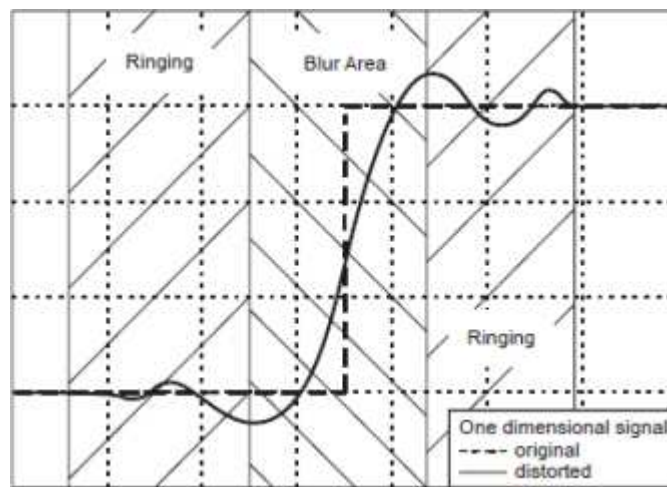


Fig. 2.16: *Ringling* e embaçamento (*blur*) em uma descontinuidade unidimensional. [32]



Fig. 2.17: Exemplo de ringing em imagem codificada por TW (a) original (b) após a decodificação.

codificação de imagem, além do desempenho das *wavelets* ortogonais. Esses avanços se justificam quando são aplicadas etapas severas de quantização (necessárias para altas taxas de compressão) que normalmente geram artefatos pelos algoritmos tradicionais. No Capítulo 3 os principais métodos envolvendo a geometria para compressão e aproximação de imagens serão apresentados.



## Capítulo 3

# Novos Métodos para Processamento de Geometria

Anisotropia é a propriedade de um sistema de transformadas em apresentar variação com a direcionalidade do sinal analisado. Tradicionalmente, em algoritmos de codificação de imagens tem-se usado um produto tensorial entre duas bases unidimensionais para analisar a direção vertical e a horizontal da imagem. Trabalhos mais recentes têm explorado a característica taxa-distorção de sistemas anisotrópicos, que consideram bases variantes com a direção e portanto não podem ser decompostas, de forma a representar imagens com contornos bem definidos.[33][2]

Assim como a *wavelet* tradicional isotrópica analisa blocos quadrados de tamanho diádico (Figura 3.1(c)), novas transformadas anisotrópicas foram definidas em domínios diferentes. Para imagens suaves por partes, é provado que aproximações lineares sobre partições de  $N$  triângulos, como na Figura 3.1(b) [2], podem atingir taxa de decaimento de erro ótima  $O(N^{-2})$  como se a imagem não tivesse descontinuidades. No entanto, não existe um algoritmo que garanta a melhor triangulação para todas as imagens suaves por partes, porque a geometria dos contornos e o contraste de embaçamento (*blur*) são desconhecidos. Algoritmos propostos na literatura constroem iterativamente a triangulação de forma a reduzir progressivamente o erro com o aumento do número de triângulos, ou utilizam triangulações densas que podem ser decimadas (reduzidas por fator de dois).

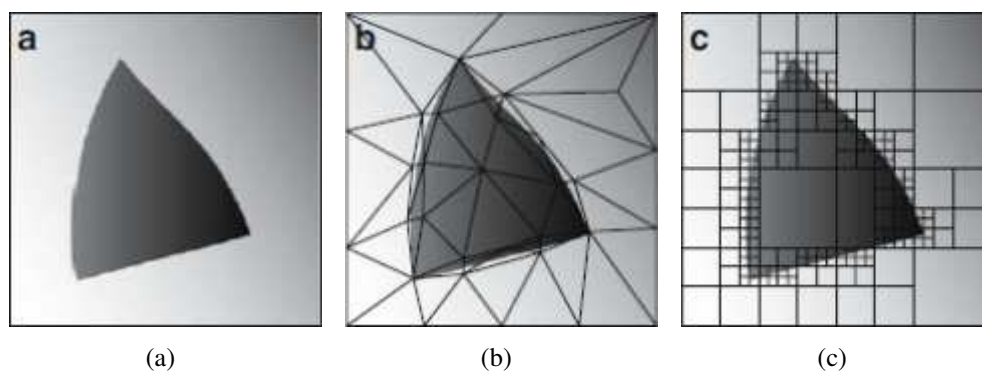


Fig. 3.1: Geometrias de aproximação (a) Imagem original (b) Aproximação com triangulação (c) aproximação com uma base *wavelet*. [2]

Tab. 3.1: Comparação da complexidade computacional de decomposições geométricas em uma imagem  $N \times N$ . [4]

<i>Directionlets</i>	$O(LN^2)$
<i>Curvelets</i>	$O(N^2 (\log_2 N)^2)$
<i>Contourlets</i>	$O(L_1 L_2 N^2)$
<i>Wedgelets</i>	$O(N^2 \log_2 N)$
<i>Wedgeprints</i>	
<i>Bandelets</i>	$O(N^2 (\log_2 N)^2)$

Muitos métodos geométricos foram propostos na literatura de forma a especificar como são feitas as partições, como aproximar o interior de cada partição e como codificar os símbolos. Alguns métodos são baseados em novas transformadas e outros são baseados em dicionários redundantes, como será mostrado a seguir. Apesar de melhor adaptados para a análise de dados geométricos, esses métodos em geral requerem algoritmos mais complexos do que as transformadas isotrópicas. O paradigma de combinar transformadas para representar de forma mais eficiente, diferentes conteúdos de um sinal foi preconizada por Coifman no início dos anos 90 [34]. A Tabela 3.1 mostra a comparação da complexidade computacional assintótica de alguns desses métodos. Neste capítulo serão mostrados em detalhes cada um deles.

### 3.1 *Directionlets*

Em [4], são apresentadas transformadas com reconstrução perfeita com momentos nulos direcionais, impostas por funções base bidimensionais ao longo de diferentes direções, batizadas de *directionlets* (Figura 3.2). Esse esquema segue as mesmas operações de filtragem e subamostragem do caso unidimensional, mas são obtidas mais componentes de detalhe para uma determinada direção, com a mesma complexidade da decomposição *wavelet* 2D separável. Para que essa aproximação seja efetiva, é necessário um bom algoritmo de codificação que agrupe de forma adequada os coeficientes gerados nas diferentes resoluções e direções e que preserve a regularidade.

Como as *directionlets* são anisotrópicas e orientadas em diferentes direções, as árvores de descendência são agrupadas em conjuntos anisotrópicos orientados, como mostrado na Figura 3.3(b). A transformada ainda mantém a característica separável da filtragem e da subamostragem presente na análise *wavelet* tradicional.

Direcionalidade é uma característica local, definida em relação a uma pequena vizinhança. Isso implica que as *directionlets* sejam aplicadas após segmentações espaciais, particionando a imagem de forma diádica em segmentos menores, com direções dominantes por segmento, como apresentado em [4]. Em [36] é mostrado que as *directionlets* com segmentação espacial conseguem decaimento de erro assintótico  $O(N^{-1,55})$ .

As *directionlets* conseguem boa aproximação de imagens constantes por partes ou polinomiais por partes, mas seu desempenho é sensivelmente menor para imagens naturais. Em [35] é apontado que elas possam ser usadas como ferramenta de remoção de ruído. Em [37] é apresentada uma modificação no algoritmo central do codificador de imagens Dirac (originalmente baseado em *wavelets*) para fazer a codificação de imagens e de vídeo usando *directionlets*. Os resultados mostram ganhos

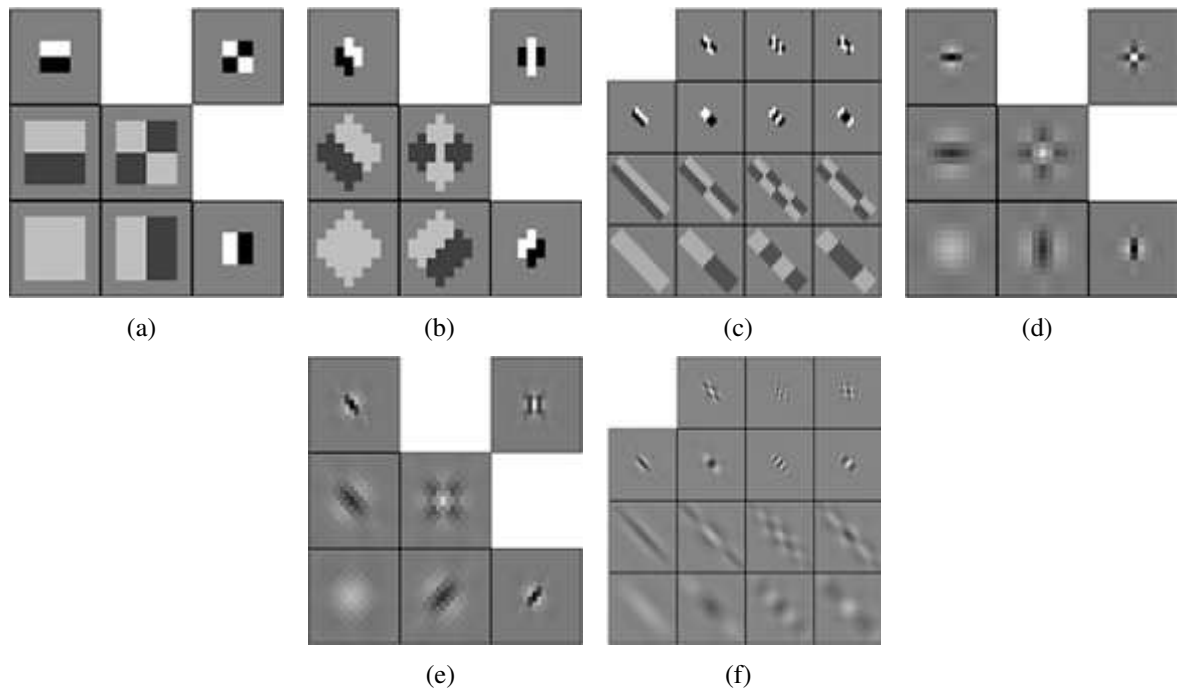


Fig. 3.2: Funções base para a decomposição *wavelet* usual e anisotrópica, para diferentes direções e bancos de filtros. (a) *wavelet* Haar isotrópica usual (b) *Wavelet* Haar isotrópica inclinada com orientação de  $45^\circ$  (c) *Wavelet* Haar anisotrópica inclinada com orientação de  $45^\circ$  (d) *Wavelet* biortogonal 9-7 isotrópica usual (e) *Wavelet* biortogonal isotrópica inclinada com orientação de  $45^\circ$  (f) *wavelet* biortogonal 9-7 anisotrópica inclinada com orientação de  $45^\circ$ . [35]

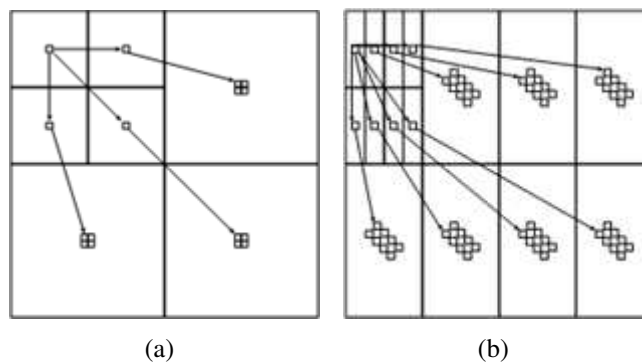


Fig. 3.3: Árvore de descendência dos coeficientes *wavelet* (a) *Wavelet* 2D usual (b) *Wavelet* anisotrópica inclinada na direção de  $45^\circ$ . [4]

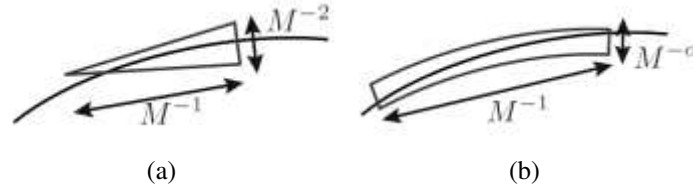


Fig. 3.4: Aproximações em blocos não retangulares. (a) Triângulo adaptado para aproximar uma curva (b) Elemento *curvelet* de aproximação. [2]

de até 0,6 dB em média, mantendo a mesma complexidade computacional assintótica.

## 3.2 Curvelets

Candes e Donoho [5] propuseram as *curvelets*, definidas como bases (ou *frames*) em  $L^2(\mathbb{R}^2)$  com suporte multi-escala, alongável e rotacionável, garantindo uma taxa de aproximação não-linear  $N^{-2} \log(N)$ , melhor do que as *wavelets*, para imagens com descontinuidades ao longo de curvas  $C^2$ .

Da mesma forma que as *wavelets* direcionais, *curvelets* são obtidas pela rotação, dilatação e translação de formas de onda elementares (bases). No entanto, *curvelet* possui suporte mais alongado, obtido pelo escalonamento dos parâmetros de largura e altura (Figura 3.4).

A decomposição em *curvelets* é obtida de formas diferentes de acordo com a escala. Em baixa escala (aproximação), é feita a filtragem pela função de escala da decomposição *wavelet*. Considerando-se  $\psi_j$  uma *curvelet* “mãe”, as outras *curvelets* em escalas  $2^{-j}$  são obtidas por rotações e translações de  $\psi_j$ . Os ângulos de rotação são parametrizados por  $\theta_l = 2\pi \cdot 2^{-j/2} \cdot l$ , com  $l = 0, 1, \dots$ , tal que  $0 < \theta_l < 2\pi$  e  $2^{-j}$  indica a escala. As translações são parametrizadas por  $k = (k_1, k_2) \in \mathbb{Z}^2$  e as *curvelets* são definidas em escalas  $2^{-j}$ , orientação  $\theta_l$  e posição  $x_k^{(j,l)} = R_\theta^{-1}(k_1 \cdot 2^{-j}, k_2 \cdot 2^{-j})$ , como: [38]

$$\psi_{j,l,k}(x) = \psi_j \left( R_{\theta,l} \left( x - x_k^{(j,l)} \right) \right) \quad (3.1)$$

Onde  $R_\theta$  indica uma rotação pelo ângulo  $\theta$ . Um coeficiente *curvelet* de detalhe é obtido pela projeção do sinal  $f$  na *curvelet*  $\psi_{j,l,k}$ :

$$c(j, l, k) = \langle f, \psi_{j,l,k} \rangle = \int_{\mathbb{R}^2} f(x) \psi_{j,l,k}(x) dx \quad (3.2)$$

A Figura 3.5 mostra o plano de frequências 2D correspondente à decomposição *curvelet*.

De acordo com [5], o erro da aproximação não-linear *curvelet* é da ordem  $O(N^{-2}(\log N)^3)$ . Duas formas de algoritmos rápidos para o cálculo da decomposição *curvelet* são descritos em [1] com complexidade assintótica  $O(N \log N)$ . Ambos foram incluídos no software CurveLab. Um deles se baseia na formulação da transformada de Fourier polar e o outro se baseia em uma reordenação dos coeficientes (chamada no artigo original de *wrapping*), de forma similar ao algoritmo usado para a decomposição *bandelet*. Algumas extensões para a aproximação *curvelet* incluem:

- **Countorlets:** Trabalhando diretamente no domínio discreto e depois estendendo para domínios contínuos, as *countorlets* baseiam-se nas mesmas ideias das *curvelets* usando filtros não sepa-

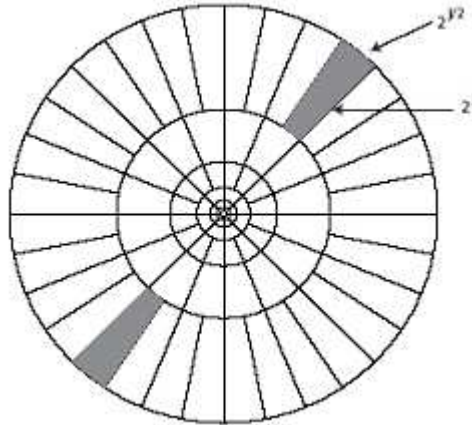


Fig. 3.5: Plano de frequências 2D da decomposição *curvelet*. [38]

ráveis. Essa construção resulta em uma flexibilidade para a expansão em diferentes níveis de multirresolução, localizações e direções. Na Figura 3.6 é mostrado espectro de frequências 2D resultante para um conjunto *countorlet*.

- **Ridgelets:** Subdividindo cada anel diádico de frequências em  $C \cdot 2^j$  ângulos a cada aumento de escala  $j$ , tem-se a transformada *ridgelet*.
- **Gabor:** Uma análise de Gabor é obtida usando a mesma largura de banda para todos os anéis de frequência da Figura 3.5, ao invés de considerar aumentos de potências de 2 como é feito nas *curvelets*.
- **Ridgelet packets:** De forma mais geral, pode-se particionar o plano de frequências de forma adaptativa de acordo com a imagem a ser analisada, consistindo nas *ridgelets packets*, apresentadas em [40].
- **Curvelets tridimensionais**, como mostradas na Figura 3.7, que se adaptam melhor a domínios maiores de dados.[1]

### 3.3 Wedgelets

Trabalhando diretamente no domínio espacial, Donoho [6] propôs o dicionário de *wedgelets*, que é redundante, mas apresenta propriedades multirresolução. Um *wedgelet* é definido como uma função constante por partes em um quadrado diádico  $S$  contendo uma descontinuidade ao longo de uma linha com orientação  $\theta$  e deslocamento  $d$ , como mostrado na Figura 3.9(c). Uma representação *wedgelet* de uma imagem pode ser realizada em blocos de tamanho pré-determinado (Figura 3.10(a)) ou iterativamente em blocos de diferentes resoluções, organizados na forma de uma árvore *quadtrees* (Figuras 3.9(b) e 3.10(b)). Os nós da *quadtrees* definem a partição diádica. Nas folhas da *quadtrees* são armazenados os parâmetros do *wedgelet* que aproxima o segmento de imagem no correspondente quadrado diádico. A Figura 3.8 ilustra um exemplo de estrutura *quadtrees*.

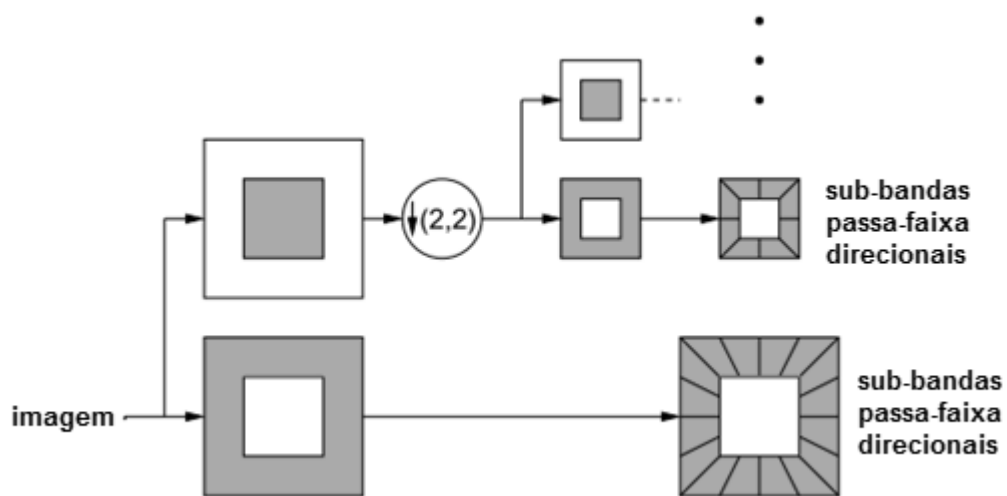


Fig. 3.6: Banco de filtros da transformação *contourlet*, evidenciando a divisão em frequências discretizadas 2D. [39]

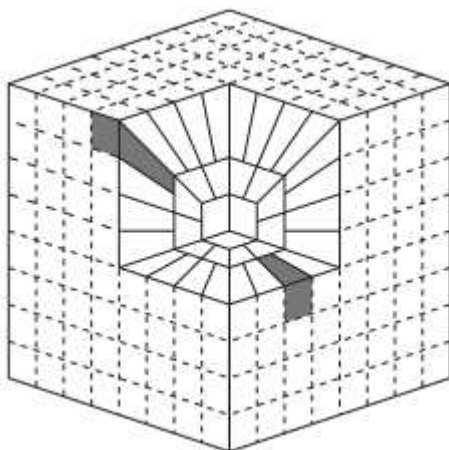


Fig. 3.7: Divisão em frequências das *curvelets* 3D. [1]

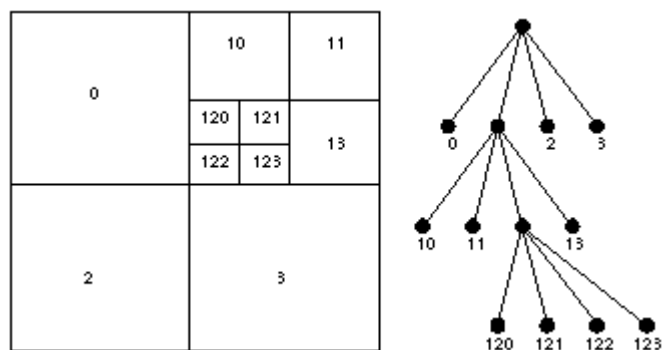


Fig. 3.8: Estrutura *quadtree*.

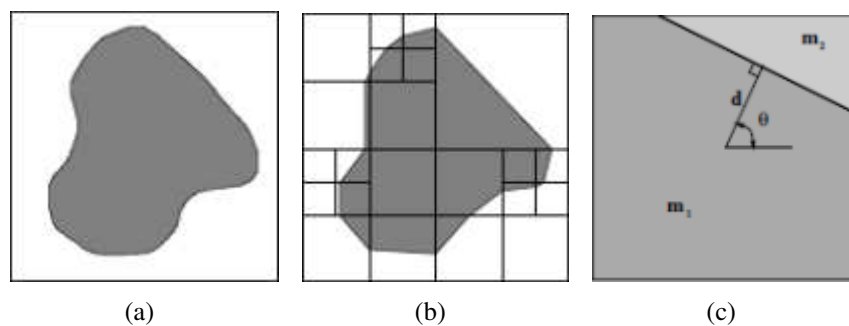


Fig. 3.9: Decomposição *wedgelet* (a) Imagem artificial (b) decomposição *wedgelet* multiescala [41] (c) parametrização de um *wedgelet* em um quadrado  $N \times N$  diádico:  $\theta$  é a orientação angular do contorno,  $d$  é a distância do contorno ao centro do bloco,  $m_1$  e  $m_2$  são as intensidades em cada lado do contorno. [42]

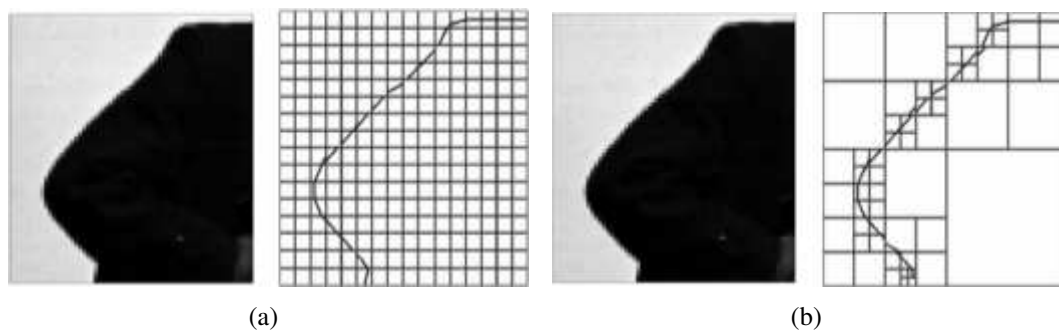


Fig. 3.10: Decomposição *wedgelet* de um trecho de imagem (a) escala única (tamanho dos blocos fixo) (a) multiescala. [43]

Da mesma forma que nos algoritmos de aproximações não-lineares em *quadtree wavelet* pode-se cortar a descendência da árvore em escalas mais baixas em regiões onde a imagem é suave na *quadtree wedgelet* pode-se cortar a árvore em regiões onde a imagem é bem aproximada por um *wedgelet* em termos de erro quadrático. Assim, imagens suaves por partes podem ser bem aproximadas usando um pequeno número de componentes.

Para determinar a taxa de erro de aproximação, Donoho[6] introduziu o conceito de funções *horizon* (horizonte), para descrever um tipo de sinal hipotético que seria bem representado usando *wedgelets* (Figura 3.9(a)). Considerando  $H(x)$  uma função unidimensional chamada de *horizon*, definida no intervalo  $[0, 1]$  e uma imagem  $f$  da forma:

$$f(x_1, x_2) = \mathbf{1}_{\{x_2 \geq H(x_1)\}}, 0 \leq x_1, x_2 \leq 1 \quad (3.3)$$

Nesse modelo, tem-se uma imagem que é “branca” ( $f = 1$ ) para coordenadas acima do *horizon* e “preta” ( $f = 0$ ) para coordenadas abaixo. Deseja-se estudar casos em que a função *horizon* é regular, ou seja, segue a condição de Hölder:

$$|f(x) - f(y)| \leq C |x - y|^\alpha \quad (3.4)$$

Nesses casos, a função *horizon* é denotada por  $Hor^\alpha([0, 1]^2)$ . Considerando apenas contornos não degenerados, isto é, que não estejam na fronteira de um quadrado, define-se o *wedgelet*  $w(x_1, x_2)$  da partição como:

$$w(x_1, x_2) = m_1 \cdot H_c + m_2 \cdot (1 - H_c) \quad (3.5)$$

Onde

$$H_c = \mathbf{1}_{\{x_2 \leq b(x_1)\}}, (x_1, x_2) \in S \quad (3.6)$$

Para a escolha dos elementos do dicionário *wedgelet* existem muitas formas propostas na literatura, sendo a mais direta com uma busca exaustiva. Definido um conjunto discreto de possíveis orientações e distâncias  $(\theta, d)$ , é calculado o erro quadrático de cada aproximação possível  $w_{\theta,d}(x_1, x_2)$  e armazena o mínimo valor encontrado. Os valores de  $m_1$  e  $m_2$  para cada *wedgelet* são tomados como a média do segmento da imagem na região acima ou abaixo da linha.

$$m_1 = E[f(x_1, x_2) \cdot H_c], m_2 = E[f(x_1, x_2) \cdot (1 - H_c)] w(x_1, x_2) = \min_{\theta,d} \|f - f_w\|^2 \quad (3.7)$$

Onde  $f$  denota a imagem original e  $f_w$  é a aproximação *wedgelet*.

Muitos outros métodos poderiam ser usados para a obtenção da melhor projeção *wedgelet* de um bloco de imagem  $N \times N$  (que minimize o MSE). A busca exaustiva no espaço  $\mathbb{R}^4$  para todos os parâmetros  $\theta, d, m_1$  e  $m_2$  é o método mais pesado computacionalmente.

O dicionário *wedgelet* é definido como o conjunto

$$W = \{w_{i,j,m} : i = 0, \dots, \log_2 N, j = 0, \dots, 4^i - 1, m = 0, \dots, M_W(S_{i,j}) - 1\} \quad (3.8)$$

A representação *wedgelet* é definida pela seguinte fórmula:



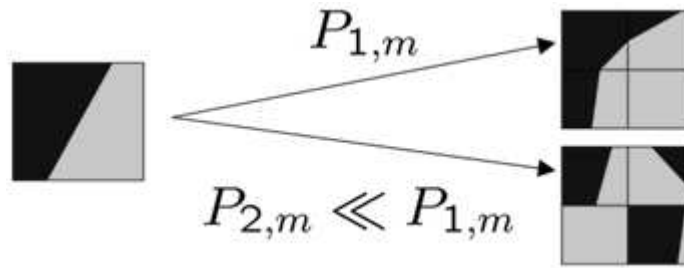


Fig. 3.11: Modelo de cadeia de Markov para representar as transições na *quadtree*. [31]

$$f_w(x_1, x_2) = \sum_{i,j,m} w_{i,j,m}(x_1, x_2) \quad (3.9)$$

Como  $W$  é um dicionário redundante, não uma base, alguns coeficientes  $w_{i,j,m}$  da representação acima podem ser nulos. A decomposição é feita basicamente em dois passos. No primeiro passo é feita uma decomposição completa da imagem, isto é, são encontradas as melhores representações *wedgelet* para todos os níveis de resolução em cada quadrado  $S_{i,j}$ ,  $0 \leq j \leq 4^i$ ,  $0 \leq i \leq \log_2 N$ , de acordo com o erro quadrático. Todos os coeficientes *wedgelet* são armazenados como nós da *quadtree*. Em seguida, no segundo passo é aplicado um algoritmo de otimização, removendo nós em níveis avançados da árvore, de forma a obter um número mínimo de nós na representação, mas garantindo a qualidade desejada. O algoritmo busca minimizar a seguinte função de custo [33]:

$$R = \min_P \{ \|f - f_w\|^2 + \lambda \cdot |P| \cdot K \} \quad (3.10)$$

Onde  $P$  denota a partição da imagem (nós armazenados na *quadtree* durante o primeiro passo),  $K$  é o número de *bits* necessário para codificar os nós e  $\lambda$  é o parâmetro de taxa-distorção, também chamado de fator de penalização. Para reconstrução exata,  $\lambda = 0$  e a aproximação *wedgelet* é exatamente a imagem original. Usualmente, a minimização é feita por um algoritmo de corte de baixo para cima na *quadtree* (*bottom-up*), cortando da árvore os nós com um custo maior do que o custo do nó pai.

Para a otimização, a *quadtree* é considerada como um modelo de cadeia de Markov de 1ª ordem, onde cada estado representa a orientação de um nó e cada transição (descida na *quadtree*) tem uma probabilidade de alterar essa orientação. São esperadas transições mais suaves, ou seja, que os nós filhos de um mesmo nó pai tenham orientações similares a orientação do nó pai, como mostrado na Figura 3.11:

$$P_{m,n} = P(\text{orientação do filho} = n | \text{orientação do pai} = m) \quad (3.11)$$

Assim, um limite inferior para determinar o gasto em *bits* para a representação da *quadtree* é entropia das probabilidades de transição conjuntas na cadeia de Markov explicitada acima. Espera-se então que, para imagens naturais com transições suaves entre os contornos a entropia seja menor do que estruturas geométricas, como mostrado na Figura 3.12. Essa propriedade pode ser usada para o desenvolvimento de algoritmos de compressão bem como para remoção de ruídos (considerando os ruídos como estruturas mais complexas e não naturais).

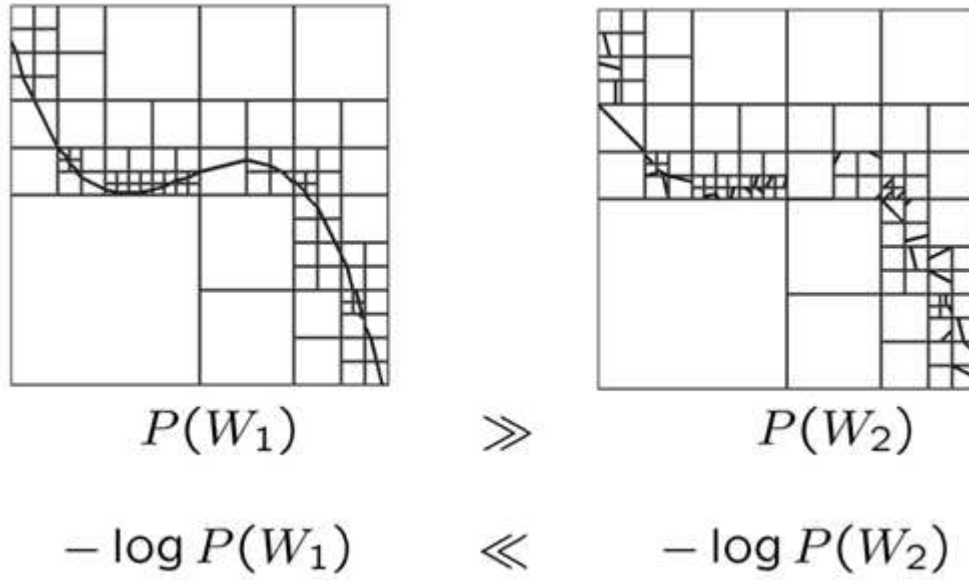


Fig. 3.12: Estruturas geométricas mais suaves apresentam uma entropia menor do que estruturas mais complexas. [31]

Para a análise do erro produzido por aproximações *wedgelet*, define-se o conjunto  $\mathfrak{M}$  de todas as possíveis partições com um seguimento de contorno (*wedge*) em um quadrado unitário. Para uma dada partição *wedgelet*  $W \in \mathfrak{M}$ , seja  $V(W)$  o espaço linear de funções lineares por partes, relativo a  $W$ :

$$V(W) := \left\{ f \mid f(x) = \sum_{W \in \mathfrak{M}} P_W(x) x(x_1, x_2), \text{ para alguma } P_W \in \Pi_1 \right\} \quad (3.12)$$

Onde  $\Pi_1$  denota o conjunto de polinômios de duas variáveis de grau no máximo 1. Considerando um sinal  $f \in L^2[0, 1]^2$  e denotando  $\hat{f}(W)$  a projeção ortogonal de  $f$  no espaço  $V(W)$ ,  $\hat{f}(W)$  é a única função que minimiza o erro quadrático  $L^2$ :

$$\|f - \hat{f}(W)\|_{L^2[0,1]^2} = \inf_{g \in V(W)} \|f - g\|_{L^2[0,1]^2} \quad (3.13)$$

A partir disso, pode-se definir uma aproximação não-linear de  $n$  termos como nos casos anteriores escolhendo uma partição  $W$  tal que  $|W| = n$ :

$$\|f - \hat{f}(W_n)\|_{L^2[0,1]^2} = \inf_{W \in \mathfrak{M}_n} \|f - \hat{f}(W)\|_{L^2[0,1]^2} \quad (3.14)$$

Onde  $\mathfrak{M}_n := \{W \in \mathfrak{M}, |W| \leq n\}$ .

Para determinar a eficiência dessa aproximação, define-se o conjunto  $W_\alpha$  e uma métrica  $|f|_{W,\alpha}$  associada:

$$W_\alpha = \left\{ f \in L^2[0, 1]^2, |f|_{W,\alpha} < +\infty \right\} \quad (3.15)$$

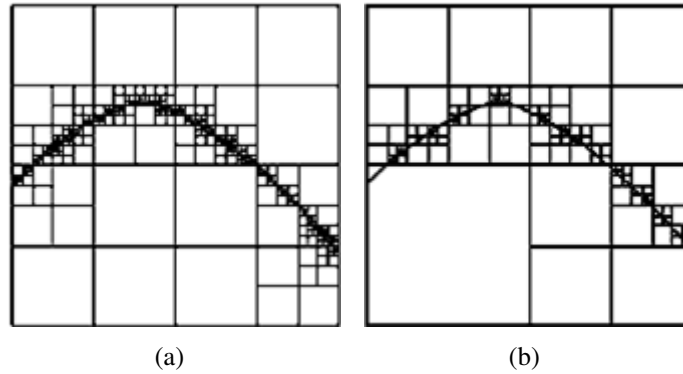


Fig. 3.13: Aproximações em *quadtree* (a) *Quadtree* com 967 nós de aproximação constante, PSNR: 33,69 dB. (b) *Quadtree* com 248 nós *wedgelet* (496 coeficientes *wedgelet*), PSNR: 33,71 dB. [44]

$$\|f\|_{W,\alpha} := \sup_{n \in \mathbb{N}} \left( n^{\frac{\alpha}{2}} \left\| f - \hat{f}(W_n) \right\|_{L^2[0,1]^2} \right) \quad (3.16)$$

O conjunto  $W_\alpha$  é chamado classe de aproximação de ordem  $\alpha$  e atua de forma similar à norma do espaço de Sobolev. Um resultado interessante é quando o sinal possui descontinuidades e  $1 \leq \alpha \leq 2$ , onde a aproximação por *wedgelets* tem uma taxa de decaimento mais rápida do que a aproximação não-linear por coeficientes *wavelet*, como apontado pelo Teorema 2 a seguir: [6]

**Teorema 2.** *Seja  $\alpha \in [1, 2]$ ,  $f \in H^{or^\alpha}[0, 1]^2$  e  $h$  a função que define o contorno de  $f$ . Existem constantes  $B, C$  independentes de  $n$  e para cada  $n$  existe uma partição *wedgelet*  $W_n$  sobre o quadrado unitário  $[0, 1]^2$ , tal que  $|W_n| \leq A \cdot n$  e satisfaz:*

$$\left\| f - \hat{f}_n(W_n) \right\|_{L^2[0,1]^2} \leq \frac{C}{n^{\frac{\alpha}{2}}} \quad (3.17)$$

Onde  $\hat{f}_n := \hat{f}_n(W_n)$  é a projeção  $L^2$  de  $f$  no espaço de funções *wedgelets* sobre a partição  $W_n$ .

Para imagens *horizon*, a aproximação não-linear por *wavelets* tem uma taxa de decaimento de erro mais lenta:

$$\left\| f - \hat{f}_n \right\|_{L^2[0,1]^2} = O\left(\frac{C}{n^{\frac{1}{2}}}\right) \quad (3.18)$$

A Figura 3.13 mostra uma comparação simplificada da aproximação de uma imagem tipo *horizon* por árvores *quadtree* compostas por nós quadrados de aproximação constante e por nós *wedgelet*. Para uma reconstrução com qualidade comparável a aproximação *wedgelet* requer aproximadamente metade do número de componentes em relação à aproximação por *quadtrees* constantes.

### 3.4 Arclets

Como mostrado em [45], o aumento de um parâmetro na representação *wedgelet* permite melhor aproximação com um número comparável de *bits*. Foram definidas *wedgelets* de 2ª ordem, ou *arclets*

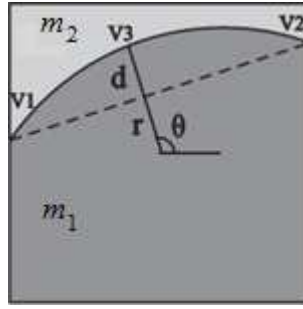


Fig. 3.14: *Wedgelets* de segunda-ordem ou *arcllet*. [45]

[7], que possuem um parâmetro adicional para indicar linhas curvas na fronteira entre as regiões *wedgelet*. A parametrização é feita por funções cônicas, como parábolas, elipses e hipérbolas, alinhadas segundo um ângulo e com apenas um parâmetro a mais, que indica a abertura da curva (Figura 3.14).

Usando a parametrização mostrada na Figura 3.14, a curva de descontinuidade do *wedgelet* de segunda-ordem pode ser côncava ou convexa. Em [45] é assumido que o parâmetro  $d$  é negativo para o caso convexo. Além disso, esse parâmetro pode ser quantizado usando um número de *bits* fixo ou dependente da escala.

Outra forma de parametrização é considerar a equação geral de uma função quadrática bidimensional:

$$ax_1^2 + 2bx_1x_2 + cx_2^2 + 2dx_1 + 2ex_2 + f = 0 \quad (3.19)$$

Na forma matricial, tem-se:

$$\underbrace{\begin{bmatrix} x_1 & x_2 & 1 \end{bmatrix}}_X \cdot \underbrace{\begin{bmatrix} a & b & d \\ b & c & e \\ d & e & f \end{bmatrix}}_Q \cdot \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}}_{X^T} = 0 \quad (3.20)$$

Onde  $a, b$  e  $c$  não são simultaneamente nulos e a matriz  $Q$  é não singular:

$$\begin{vmatrix} a & b & d \\ b & c & e \\ d & e & f \end{vmatrix} \neq 0, acf + 2bde - cd^2 - ae^2 - b^2f \neq 0, a, b, c, d, e, f \in \mathbb{R} \quad (3.21)$$

Usando a mesma notação dos *wedgelets* tradicionais, um *arcllet* ou *wedgelet* de segunda-ordem é definido em um quadrado diádico  $S$  como:

$$\hat{w}(x_1, x_2) = \mathbf{1} \left\{ x_2 \leq \hat{b}(x_1) \right\}, (x_1, x_2) \in S \quad (3.22)$$

Onde  $\hat{b}$  é a curva quadrática definida pelas equações anteriores.

Em [45] a decomposição em *arcllets* é feita a partir da decomposição em *wedgelets* de primeira ordem. Após encontrar os melhores parâmetros  $m_1, m_2, \theta$  e  $d$ , é feita uma busca pelo melhor parâmetro  $r$ . Como nem todos os possíveis *arcllets* são considerados, o cálculo é mais rápido que uma busca exaustiva e os resultados ainda são satisfatórios.

Em [46] é demonstrado que os *arcllets* resultam em menores distorções do que os de primeira-ordem, mas com aumento do tempo computacional (de acordo com o número de *bits* de quantização do parâmetro  $r$ ).

### 3.5 *Wedgeprints*

Os modelos *wedgelet* e *arcllet* garantem uma reconstrução da estrutura geométrica (contornos) com taxa de aproximação superior às transformadas *wavelets* e outros modelos ortogonais e têm reconstrução perfeita para sinais do tipo *horizon*, desde que o modelo seja bem parametrizado a priori (daí, a grande variedade de *wedgelets* disponível, além da parametrização dos ângulos e do processo de decomposição em árvores). Para imagens naturais no entanto, modelos *horizon* não representam com perfeição, mesmo que a informação primordial de contornos seja preservada. Para garantir a reconstrução perfeita é preciso completar o modelo com informação não geométrica.

Em [47], Wakin et al. propõem um método de codificação híbrida *wedgelets-wavelet*, que otimiza a relação taxa-distorção para cada subregião da imagem, ao invés de usar o resíduo. Basicamente o método é uma modificação do algoritmo SFQ (*Space Frequency Quantization*), que codifica os coeficientes *wavelet* buscando minimizar o erro da aproximação não-linear a cada passo e a cada resolução observada. Trata-se de um algoritmo mais complexo que o SPIHT (*Set Partitioning In Hierarchical Trees*) pois pode fazer várias varreduras em coeficientes *wavelet* em um mesmo passo, até encontrar a melhor escolha, e por isso obtém melhores resultados.

Na modificação proposta em [47], chamada de W-FSQ, os coeficientes *wavelet* são comparados um a um a fim de determinar a função de custo taxa $\times$ distorção (número de *bits* necessários para codificá-lo e erro produzido ao não codificar). O coeficiente pode ser classificado como não-significativo (região suave, representada com um coeficiente nulo na decomposição *wavelet*), textura (região oscilatória, o coeficiente é quantizado e codificado e a árvore de descendência entra na codificação) ou contorno. Nesse último caso, são codificados os parâmetros *wedgelet* ( $d, \theta, m_1, m_2$ ) que representam o contorno naquela região. Para a decodificação, a região é preenchida no domínio *wavelet* com os coeficientes correspondentes à geometria, ou mais especificamente, ao *wedgelet*. Isso é feito usando *wedgeprints*, projeções dos *wedgelets* no domínio *wavelet* (Figura 3.15).

Para gerar os *wedgeprints*, é utilizada uma amostra do *wedgelet* com tamanho maior que o suporte da base *wavelet* escolhida. Caso as bordas não sejam tratadas de forma adequada para a decomposição, com algum tipo de extensão periódico ou espelhamento, a reconstrução do *wedgelet* a partir do *wedgeprint* pode gerar efeitos de blocagem intensos. A Figura 3.16 ilustra esse processo.

### 3.6 *Surflets e Surfprints*

Em [49] o conceito de *wedgelets* é generalizado pelo dicionário de *surflets*. Consideram-se funções  $f(M, K_d, K_s)$   $M$ -dimensionais, constantes por partes ou suaves por partes, que contenham uma descontinuidade  $C^{K_d}$ -suave  $(M - 1)$ -dimensional, separando duas regiões  $C^{K_s}$ -suaves em  $M$  dimensões. Para essa notação, um *wedgelet* de primeira ordem é um *surflet* com  $K_d = 2$  e  $M = 2$  e um *arcllet* é um *surflet* com  $K_d = 3$  e  $M = 2$ . A Figura 3.17 mostra esses e outros exemplos.

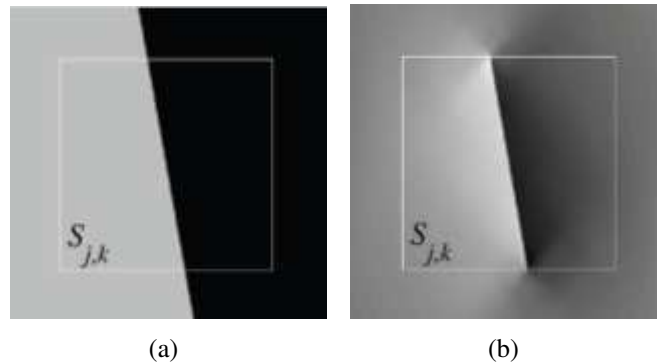


Fig. 3.15: *Wedgelets* no domínio *wavelet* (a) Porção de uma imagem contendo um *wedgelet* em um bloco (b) *Wedgeprint* obtido pela projeção do *wedgelet* na sub-banda vertical do domínio *wavelet*. [48]

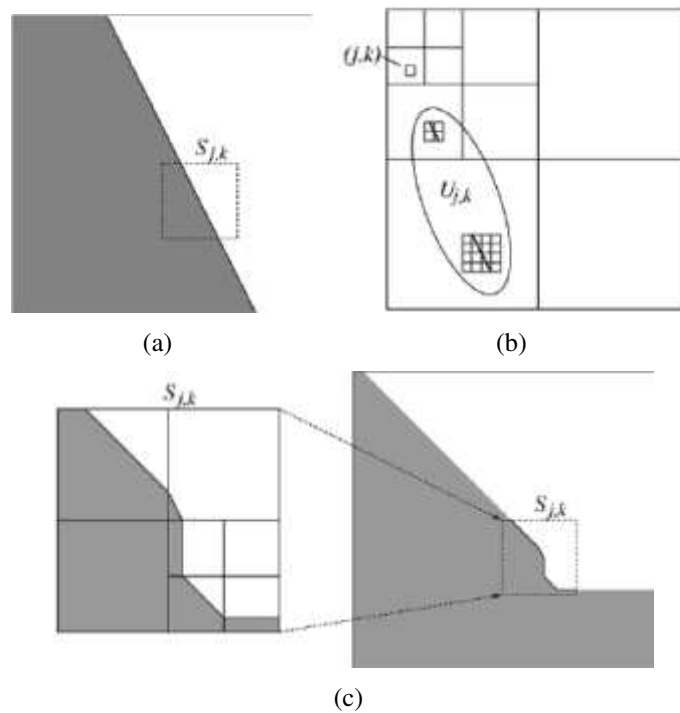


Fig. 3.16: Obtenção de um *wedgeprint*: (a) Crie uma imagem temporária  $Y$  usando extensões lineares nas bordas de  $S_{j,k}$ . (b) Obtenha a transformada *wavelet* de  $Y$  e extraia a subárvore de coeficientes *wavelet* com raiz no nó  $(j, k)$  (escolhido na sub-banda horizontal, neste exemplo). (c) Um procedimento similar é usado para projetar um agrupamento *wedgelet*. [48]

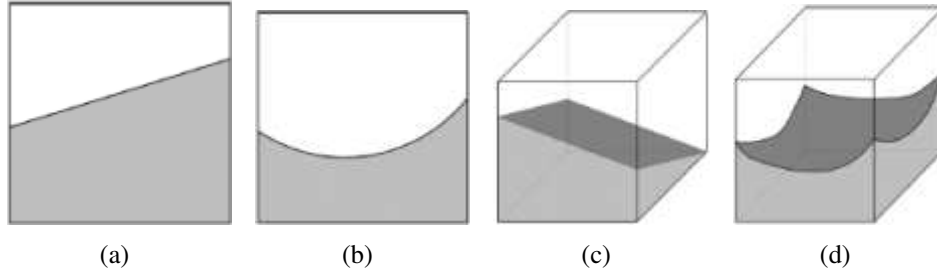


Fig. 3.17: Exemplos de surflets: (a)  $M = 2, K_d = [1, 2]$  (b)  $M = 2, K_d = [2, 3]$  (c)  $M = 3, K_d = [1, 2]$  (d)  $M = 3, K_d = [2, 3]$ . [49]

A decomposição em *surflets* é motivada pela teoria de aproximações em séries de Taylor. Considerando uma função  $f$  de  $D$  variáveis em  $C^K$  com  $K = r + \alpha$ ,  $r$  um inteiro positivo e  $\alpha \in (0, 1]$ , então  $f$  pode ser aproximada em um ponto  $\mathbf{z} + \mathbf{h}$  pela combinação das derivadas parciais no ponto  $\mathbf{z}$ :

$$f(\mathbf{z} + \mathbf{h}) = f(\mathbf{z}) + \frac{1}{1!} \sum_{i_1=1}^D f_{z_{i_1}}(\mathbf{z}) h_{i_1} + \frac{1}{2!} \sum_{i_1, i_2=1}^D f_{z_{i_1}, z_{i_2}}(\mathbf{z}) h_{i_1} h_{i_2} + \cdots + \frac{1}{r!} \sum_{i_1, \dots, i_r=1}^D f_{z_{i_1}, \dots, z_{i_r}}(\mathbf{z}) h_{i_1} \cdots h_{i_r} + O(\|\mathbf{h}\|^K) \quad (3.23)$$

Onde  $f_{z_{i_1}, \dots, z_{i_r}}$  indica as derivadas parciais de  $f$  com relação a  $z_{i_1}, \dots, z_{i_r}$ . Assim, em um domínio pequeno em torno de  $\mathbf{z}$ ,  $f$  é aproximada por um polinômio de ordem  $r$ .

Assim, parametrizando as coordenadas  $x \in S_M$  do domínio  $M$ -dimensional pelo vetor  $\mathbf{y} = [x_1, x_2, \dots, x_{M-1}]$ , um *surflet* é definido pelo polinômio  $(M - 1)$ -dimensional  $p$  correspondente:

$$s(\mathbf{x}) = \begin{cases} 1, & x_M \geq p(\mathbf{y}) \\ 0, & \text{caso contrário} \end{cases} \quad (3.24)$$

O polinômio  $p(\mathbf{y})$  é definido como:

$$p(\mathbf{y}) = p_0 + \sum_{i_1=1}^{M-1} p_{1, i_1} y_{i_1} + \sum_{i_1, i_2=1}^{M-1} p_{2, i_1, i_2} y_{i_1} y_{i_2} + \cdots + \sum_{i_1, \dots, i_r=1}^{M-1} p_{r, i_1, i_2, \dots, i_r} y_{i_1} \cdots y_{i_r} \quad (3.25)$$

Pela aproximação de Taylor é possível demonstrar que qualquer função suave definindo a descontinuidade da imagem pode ser aproximada por uma representação polinomial.

Assim como na decomposição *wedgelet*, uma função  $M$ -dimensional pode ser aproximada por partes por uma coleção de *surflets* de diferentes tamanhos e em diferentes localizações, em uma estrutura de árvore, como mostrado na Figura 3.18.

Em [50] é mostrada uma abordagem para encontrar os melhores coeficientes *surflets* como um problema de otimização no espaço contínuo correspondente, de forma a minimizar o erro quadrático. Para evitar instabilidade em regiões descontínuas, é utilizada uma aproximação por método de Newton iterativo. O sucesso dessa abordagem depende de uma boa escolha dos parâmetros iniciais, mas

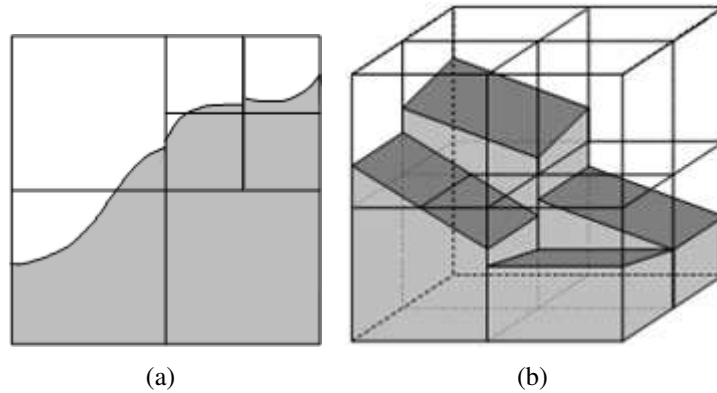


Fig. 3.18: Exemplos de aproximações por árvores *surflet*: (a) Cúbica por partes com  $M = 2$  (b) Linear por partes com  $M = 3$ . [49]

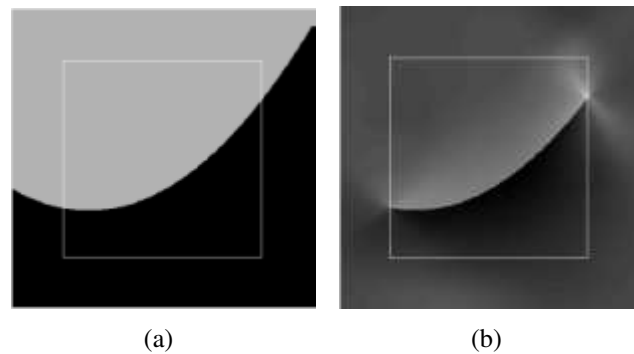


Fig. 3.19: *Surflets* no domínio *wavelet*: (a) Porção de uma imagem contendo um *surflet* em um bloco (b) *Surfprint* obtido pela projeção do *surflet* na sub-banda vertical do domínio *wavelet*. [49]

em [50] é conseguida a convergência em até 10 iterações, com resultados melhores do que a busca exaustiva no domínio discreto.

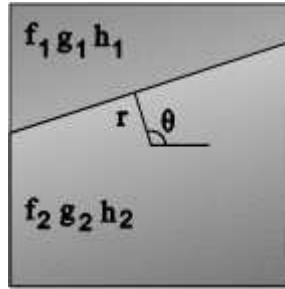
A partir do trabalho de Cohen et al. [51], obtém-se que a taxa de aproximação da decomposição *surflet* em relação ao número  $N$  de elementos usados para classes de dados com descontinuidades definidas por funções  $(M - 1)$ -dimensionais  $C^{K_d}$  é:

$$\|f - \hat{f}_N\|^2 = O\left(\left(\frac{1}{N}\right)^{\frac{K_d}{M-1}}\right) \quad (3.26)$$

Da mesma forma que as *wedgelets* foram estendidas para as *surflets*, em [49] as *wedgeprints* são estendidas para as *surfprints*, projeções das *surflets* no domínio *wavelet* (Figura 3.19).

Considerando a classe de funções  $F_S(M, K_d, K_s)$  consistindo de funções  $M$ -dimensionais suaves por partes que contém descontinuidades  $C^{K_d}$ -suaves  $(M - 1)$ -dimensionais separando duas regiões  $C^{K_s}$ -suaves em  $M$  dimensões definida no domínio  $S$ , é mostrado que a taxa de aproximação pela combinação de *surflets* e *surfprints* é da forma:



Fig. 3.20: Elemento do conjunto *platelet*. [8]

$$\|f - \hat{f}_N\|^2 = O\left(\left(\frac{1}{N}\right)^{\min\left(\frac{K_d}{M-1}, \frac{2K_s}{M}\right)}\right) \quad (3.27)$$

Em [49] é mostrado que, usando codificação baseada em árvore, consegue-se uma relação logarítmica para o erro de aproximação em relação à taxa de *bits*  $R$ :

$$\|f - \hat{f}_R\|^2 = O\left(\left(\frac{\log R}{R}\right)^{\min\left(\frac{K_d}{M-1}, \frac{2K_s}{M}\right)}\right) \quad (3.28)$$

Apesar do resultado subótimo para essa abordagem de codificação usando *surflets* e *surfprints*, algoritmos que atinjam esse desempenho ainda não foram propostos.

### 3.7 Platelets

Aproximações por regiões suaves, mas não constantes, são propostas em [8]. São definidas aproximações chamadas de *platelets*, que consistem de *wedgelets* contendo regiões com valores linearmente variáveis, divididas por uma linha reta, como mostrado na Figura 3.20. A aproximação por *platelets* é mais recomendada em algumas aplicações envolvendo imagens médicas em que o gradiente da imagem corresponde a informação sobre emissão de luz ou refletividade de um material.

A aproximação de uma imagem em um quadrado diádico  $S$  por uma *platelet* é definida como:

$$f_S(x, y) = (A_S x + B_S y + C_S) \cdot I_S(x, y) \quad (3.29)$$

Onde  $A_S, B_S, C_S \in \mathbb{R}$  e  $I_S$  denota o indicador da função em  $S$ . Cada região *platelet* precisa de 3 parâmetros em comparação com um único parâmetro da aproximação constante. Apesar de precisar de dois parâmetros adicionais, para imagens com regiões suficientemente suaves a aproximação por *platelets* requer menos elementos para um mesmo erro. [8]

Na Figura 3.21 tem-se a comparação da aproximação de uma imagem de teste suave por partes (quadrática) por quadrados constantes (correspondente à transformada *wavelet* de Haar), *wedgelets* constantes por partes, *platelets* em regiões quadradas e *platelets* em regiões definidas por *wedgelets*. Visualmente, é notável a melhoria proporcionada pela aproximação linear por partes, principalmente distante da descontinuidade, no centro e fundo da imagem. Apesar de precisar de mais parâmetros por

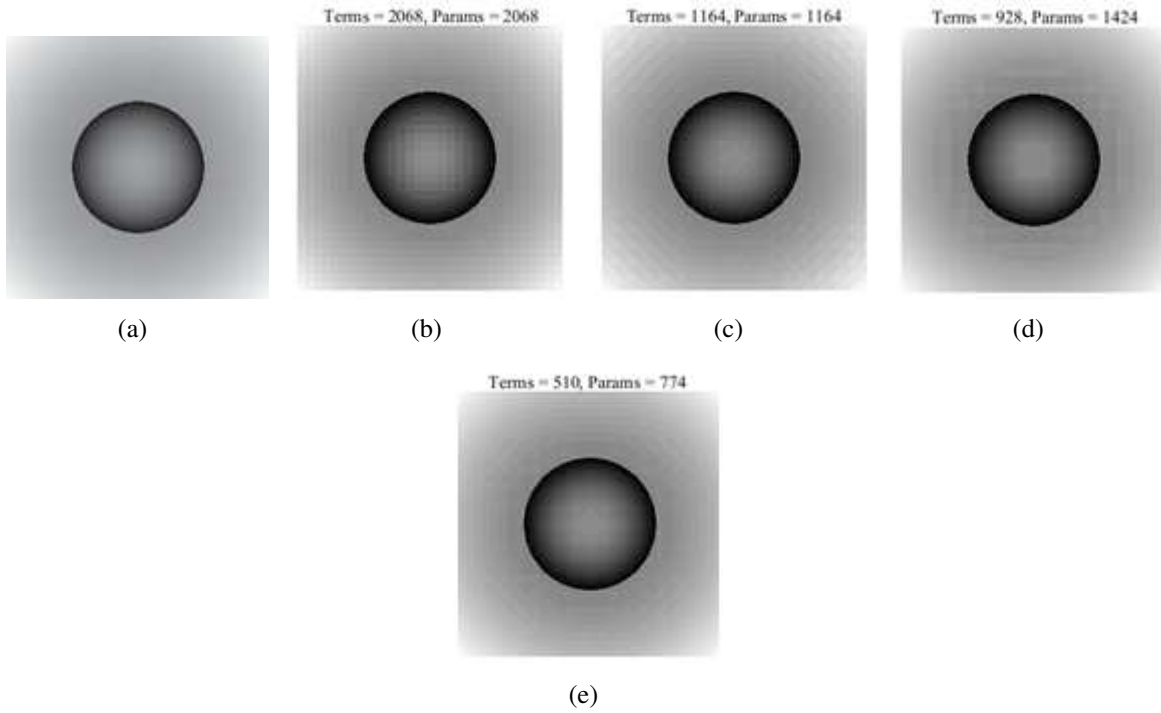


Fig. 3.21: Aproximações de uma imagem suave por partes: (a) Imagem original (b) *wavelet* Haar (c) *Wedgelets* (d) *Platelets* em regiões quadradas (e) *Platelets* em regiões definidas por *wedgelets*. [8]

elemento para a codificação, o número de elementos necessários é menor com a aproximação *platelet*, fazendo com que o número de parâmetros total seja menor, para um mesmo ou menor erro na imagem. A Figura 3.22 mostra a comparação em termos de erro quadrático pelo número de parâmetros total (taxa  $\times$  distorção).

Considerando funções  $f(x, y)$  suaves, isto é,

$$|f(\mathbf{y}) - p_{\mathbf{x}}(\mathbf{y})| \leq C_{\beta} |\mathbf{y} - \mathbf{x}|^{\beta} \quad (3.30)$$

Onde  $\mathbf{x}, \mathbf{y} \in [0, 1]^2$  e  $p_{\mathbf{x}}(\mathbf{y})$  é o polinômio de Taylor de ordem  $k$  para  $f(x, y)$  no ponto  $\mathbf{x}$  e  $\beta \in (1, 2]$ . Em [8] é mostrado que, para uma imagem do tipo:

$$f(x, y) = f_1(x, y) \cdot I_{\{y \geq H(x)\}} + f_2(x, y) \cdot (1 - I_{\{y \geq H(x)\}}), \quad H \in \text{Höder}^{\alpha}, \alpha \in (1, 2] \quad (3.31)$$

Ou seja,  $f(x, y)$  consiste de regiões  $f_1(x, y)$  e  $f_2(x, y)$  suaves separadas por contornos suaves, *platelets* permitem aproximações melhores do que por Fourier, *wavelet* ou *wedgelet*. Por exemplo, para contornos do tipo Lipschitz ( $\alpha, \beta = 2$ ), em baixas taxas ( $N$  pequeno) a aproximação por *platelets* resulta em erros  $O(N^{-2})$ , enquanto aproximações em bases de Fourier resultam em  $O(N^{-1/2})$  e *wavelet* e *wedgelet* resultam em  $O(N^{-1})$ .

Para aproximar classes com maior grau de suavidade ( $\alpha, \beta > 2$ ) são necessárias funções de maior ordem para que se obtenha o mesmo resultado assintótico. Em [8] é argumentado que essa abordagem, embora de análise teórica simples, pode se tornar computacionalmente inviável devido

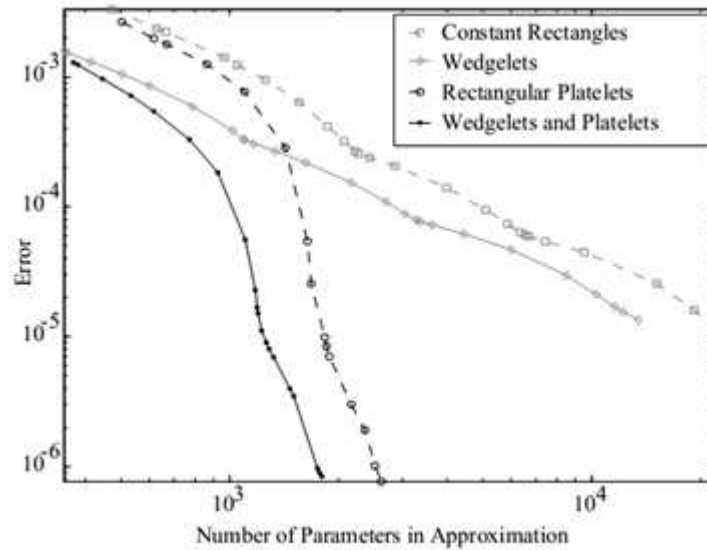


Fig. 3.22: Erro de aproximação em relação ao número de parâmetros total necessário. [8]

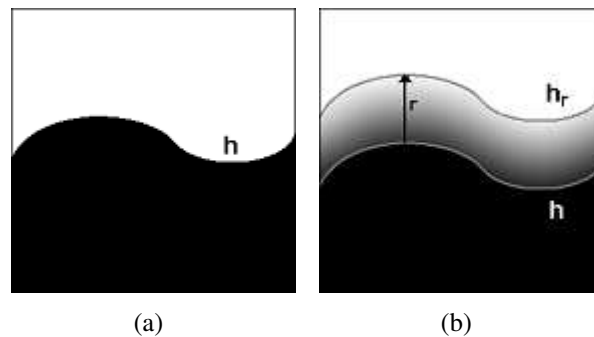


Fig. 3.23: Aproximação *smoothlet* (a) Imagem *horizon* (b) Imagem *horizon* borrada (*blurred*), de acordo com a curvatura. [52]

ao aumento dos graus de liberdade do problema. A busca exaustiva se torna proibitiva, mas métodos mais rápidos para esses casos ainda não foram propostos.

Lisowska [52] propôs uma nova variação para o dicionário *wedgelet*, considerando regiões constantes separadas por uma faixa de variação suave, como mostra a Figura 3.23. Chamadas de *smoothlets*, essas estruturas podem se adaptar à localização, tamanho, rotação, curvatura e também à suavidade. O algoritmo é muito mais complexo que a decomposição *wedgelet* usual e mais parâmetros são necessários para codificar uma imagem, mas são conseguidos ganhos da ordem de 3,8% de PSNR (*Peak Signal to Noise Ratio*) para uma mesma taxa de *bits* em relação ao método com *wedgelets* tradicionais. O ganho efetivo no entanto está relacionado com a suavidade das bordas da imagem. Em [52] é apontada uma ordem de aproximação  $O(N^{-3})$ .

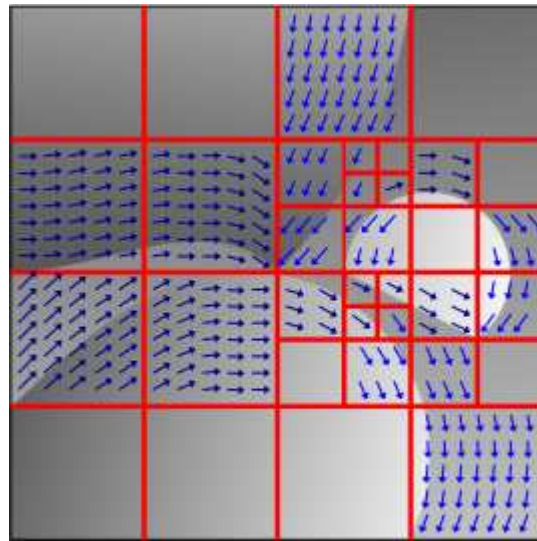


Fig. 3.24: Exemplo de partição de uma imagem em quadrados diádicos de acordo com o fluxo geométrico. [56]

### 3.8 *Bandelets* de 1ª Geração

Le Pennec junto com outros pesquisadores em [53], [54], [55] e [56], propuseram o dicionário de bases *bandelet* ortogonais, tomando vantagem da regularidade geométrica. Ao invés de descrever a geometria da imagem por meio de contornos, que são na maioria das vezes mal definidos, a geometria de imagens é caracterizada por fluxos geométricos.

Em uma região  $\Omega$ , um fluxo geométrico é um campo vetorial  $\tau(x_1, x_2)$  definido para cada  $(x_1, x_2) \in \Omega$ , que indica localmente a direção em que a imagem  $f$  tem variações suaves. Se a imagem for uniformemente regular nessa região (região homogênea), então essa direção não é unicamente definida. Para construir bases ortogonais, uma condição imposta é que o fluxo seja paralelo verticalmente ( $\tau(x_1, x_2) = \tau(x_1)$ ) ou horizontalmente ( $\tau(x_1, x_2) = \tau(x_2)$ ). Para manter uma certa flexibilidade, a condição de paralelismo é imposta apenas em regiões  $\Omega_i$  contidas no suporte da imagem. O suporte da imagem  $S$  é particionado em regiões  $S = \bigcup_i \Omega_i$  e em cada  $\Omega_i$  o fluxo é paralelo verticalmente ou horizontalmente. Para regiões homogêneas esse fluxo geométrico não é definido.

A Figura 3.24 mostra um exemplo de imagem particionada em regiões quadradas de tamanho suficiente para que cada região  $\Omega_i$  inclua pelo menos um contorno. Em cada região o fluxo é paralelo às curvas tangentes aos contornos. *Bandelets* são construídas nessas regiões distorcendo as bases *wavelet* separáveis de forma a seguirem as linhas do fluxo e aplicando um processo de “bandelização” que aproveita a vantagem da regularidade da imagem ao longo do fluxo geométrico.

O método de decomposição por *bandelets* retransforma os coeficientes *wavelet* de forma direcional, criando novos momentos nulos nas direções apropriadas, com decaimento do erro em  $O(N^{-\alpha})$ , onde  $\alpha$  é o parâmetro que indica a regularidade (expoente de Hölder). De forma distinta das *directionlets* e *curvelets*, que são novas bases de decomposição, para a decomposição em *bandelets* são utilizadas as bases *wavelet* aplicadas em direções diferentes.

Se não existir um fluxo geométrico sobre a região  $\Omega_i$ , o que indica regularidade isotrópica, é

aplicada a *wavelet* separável, como mostrado na Seção 2.3. Se existir um fluxo geométrico, a base *wavelet* é substituída pela *bandelet* correspondente.

Para melhor explicar a construção de uma *bandelet*, considere primeiro um fluxo paralelo em uma direção vertical:  $\tau(x_1, x_2) = \tau(x_1)$ . Normalizando os vetores do fluxo, escreve-se:  $\tau(x_1) = (1, c'(x_1))$ . Seja  $x_{min} = \inf_{x_1} \{(x_1, x_2) \in \Omega\}$ . Uma linha de fluxo é definida como a integral do fluxo, cuja tangente é paralela à  $\tau(x_1)$ . Como o fluxo é paralelo verticalmente, uma linha de fluxo é um conjunto de pontos  $(x_1, x_2) \in \Omega$  que satisfazem  $x_2 = c(x_1) + c_0$ , onde o parâmetro  $c_0$  indica translação e

$$c(x_1) = \int_{x_1}^{x_{min}} c'(x_1) dx_1 \quad (3.32)$$

Pela construção do fluxo geométrico, os níveis de cinza da imagem tem variação regular ao longo das linhas de fluxo. Para tirar vantagem dessa regularidade com *wavelets*, as *wavelets* separáveis são distorcidas com um operador  $W$  e fazendo translações ao longo de  $x_2$ . Para evitar o surgimento de efeitos de borda, é usada uma transformada *wavelet* distorcida discreta que se estende além dos limites da imagem e garante propriedades de reconstrução perfeita.

A direção é parametrizada verticalmente ou horizontalmente, dependendo da geometria. Considerando um bloco de imagem  $f$  com direção parametrizada na vertical ( $x_1 = c(x_2)$ ), este pode ser recuperado pelos coeficientes *bandelet*:

$$w_{j,k}^c(x_2) = \langle f(x_1, x_2), \psi_{j,c(x_2)}^k(x_1) \rangle \quad (3.33)$$

Se o bloco de imagem contém uma borda regular (linha reta), é verificado que, para  $j$  e  $k$  fixos, os coeficientes variam de forma suave como uma função de  $x_2$ . Esses coeficientes são comprimidos de forma eficiente por uma transformada *wavelet* unidimensional:

$$\{\psi_{l,n}(t) = 2^{-l/2} \psi(2^{-l}t - n)\}_{l,n} \quad (3.34)$$

### 3.9 *Bandelets* de 2ª Geração

As *bandelets* de segunda geração [57] avançaram ainda mais na representação de imagens em relação às *bandelets* de primeira geração, partindo dos coeficientes de uma transformação *wavelet* ao invés de transformar a imagem. Com isso, o efeito de bloco (comum nas *bandelets* de 1ª geração) é atenuado pelo espalhamento na reconstrução *wavelet*. Dessa forma, as *bandelet* aproveitam a regularidade capturada pela transformada *wavelet*, gerando novos momentos nulos.

Os coeficientes da decomposição *wavelet* da imagem  $f$  podem ser escritos como:

$$\langle f, \psi_{j,n}^k \rangle = f * \bar{\psi}_j^k(2^j n) \quad (3.35)$$

Com

$$\bar{\psi}_j^k(x) = 2^{-j} \psi^k(-2^{-j}x) \quad \text{para } x = (x_1, x_2) \quad \text{e } n = (n_1, n_2) \quad (3.36)$$

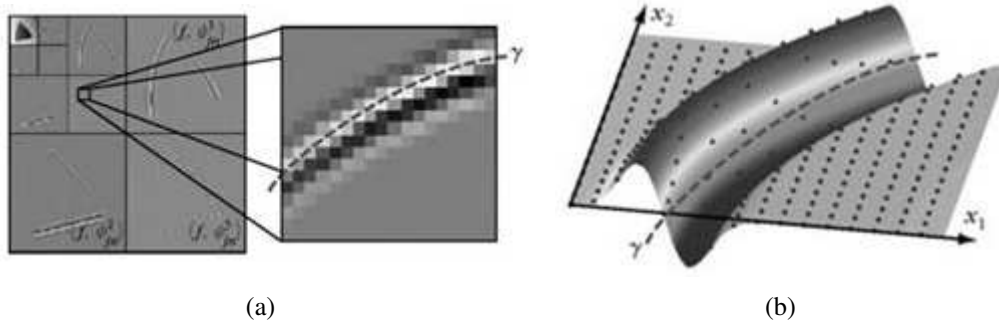


Fig. 3.25: Contornos em uma imagem: (a) Coeficientes *wavelet* na escala  $2^j$  são amostras da função  $f * \bar{\psi}_j^k(x)$  (b) A imagem filtrada varia regularmente ao longo de um contorno  $\gamma$ . [3]

A função  $f * \bar{\psi}_j^k(x)$  tem a mesma regularidade direcional de  $f$ , por exemplo ao longo de um contorno, e é regularizada pela convolução com  $\bar{\phi}_j^k(x)$ . A Figura 3.25 mostra os coeficientes *wavelet* próximos a um contorno.

As *bandeleets* resultantes são escritas como:

$$\varphi_p(x) = \sum_n \tilde{\psi}_{i,l,m}[n] \psi_{j,n}^k(x) \quad (3.37)$$

Onde  $\tilde{\psi}_{i,l,m}[n]$  é uma *wavelet* direcional de comprimento  $2^i$  e largura  $2^l$ , com posição  $m$  e  $p = (k, j, l, i, m)$  é a parametrização em posição, escala e tamanho do *bandelet*.

A função *bandelet*  $\varphi_p(x)$  é uma combinação linear de *wavelets*  $\psi_{j,n}^k(x)$  e desta forma apresenta a mesma regularidade dessas *wavelets*. Como  $\psi_{j,n}^k(x)$  tem um suporte quadrado de largura proporcional a  $2^j$ , a *bandelet*  $\varphi_p$  tem um suporte com comprimento proporcional a  $2^{j+i}$  e largura proporcional a  $2^{j+l}$ .

Se o expoente de regularidade  $\alpha$  na vizinhança de um contorno for conhecido, requer-se uma *bandelet* com suporte alongado de forma que  $2^{j+l} = (2^{j+i})^\alpha$  e portanto  $l = \alpha i + (\alpha - 1)j$ . [3] *Curvelets* satisfazem essa propriedade para  $\alpha = 2$ . No entanto, como  $\alpha$  não é conhecido a priori, os parâmetros  $i$  e  $l$  precisam ser ajustados adaptativamente.

Os coeficientes *bandelet* de um sinal  $f$  são descritos como:

$$\langle f, \varphi_p \rangle = \sum_n \tilde{\psi}_{i,l,m}[n] \langle f, \psi_{j,n}^k \rangle \quad (3.38)$$

Na prática, a implementação é feita reorganizando os coeficientes da *wavelet* 2D em uma estrutura unidimensional, ao longo da direção especificada, e aplicando uma transformada *wavelet* unidimensional. Esse processo de reordenação dos coeficientes *wavelet* (chamado de “bandeletização”) remove a correlação existente nos coeficientes *wavelet* em torno de uma singularidade (contornos). O caso mais simples é quando os dados estão alinhados em uma linha reta com direção bem definida. Outras formas de alinhamento podem existir, mas são mais difíceis de representar.

A Figura 3.26 mostra um exemplo de partição no domínio *wavelet*, de forma a ser usada com uma transformada *bandelet* de segunda geração. Em regiões suaves, os coeficientes *wavelet* têm valores pequenos e não precisam ser retransformados. Em junções de contornos, ocorrem irregularidades em

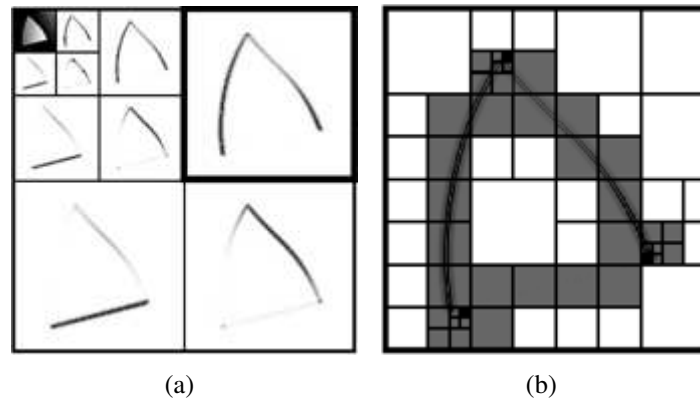


Fig. 3.26: Decomposição *bandelet*: (a) Coeficientes *wavelet* de uma imagem. (b) Exemplo de segmentação de uma escala da transformada *wavelet* em quadrados diádicos de diferentes tamanhos. [3]

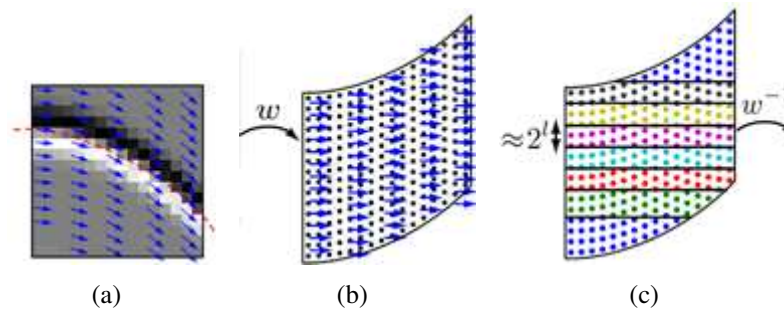


Fig. 3.27: Determinação do fluxo geométrico da decomposição *bandelet*: (a) Partição quadrada de coeficientes *wavelet* contendo um contorno. O fluxo geométrico paralelo ao contorno é mostrado com setas. (b) Uma distorção (*warping*)  $w$  mapeia o fluxo original para um fluxo horizontal. (c) Suporte das *wavelets* direcionais  $\tilde{\psi}_{i,l,m}(w[n])$  de comprimento  $2^i$  e largura  $2^l$  no domínio distorcido. [3]

todas as direções e os coeficientes também não precisam ser retransformados. Apenas nos contornos e em sua vizinhança é que existe informação geométrica que pode ser retransformada para uma melhor representação esparsa.

Fluxos geométricos são calculados para cada partição quadrada em torno dos coeficientes *wavelet* de um contorno, como mostrado na Figura 3.27(a), que indica as direções em que a projeção  $\langle f, \psi_j^k(w(x)) \rangle$  é mais regular. Uma *wavelet* direcional distorcida é aplicada de forma a aproveitar a regularidade horizontal dos coeficientes, como mostra a Figura 3.27(b). *Wavelets* direcionais podem ser construídas com *wavelets* polinomiais do tipo Alpert [58], mostradas na Figura 3.28.

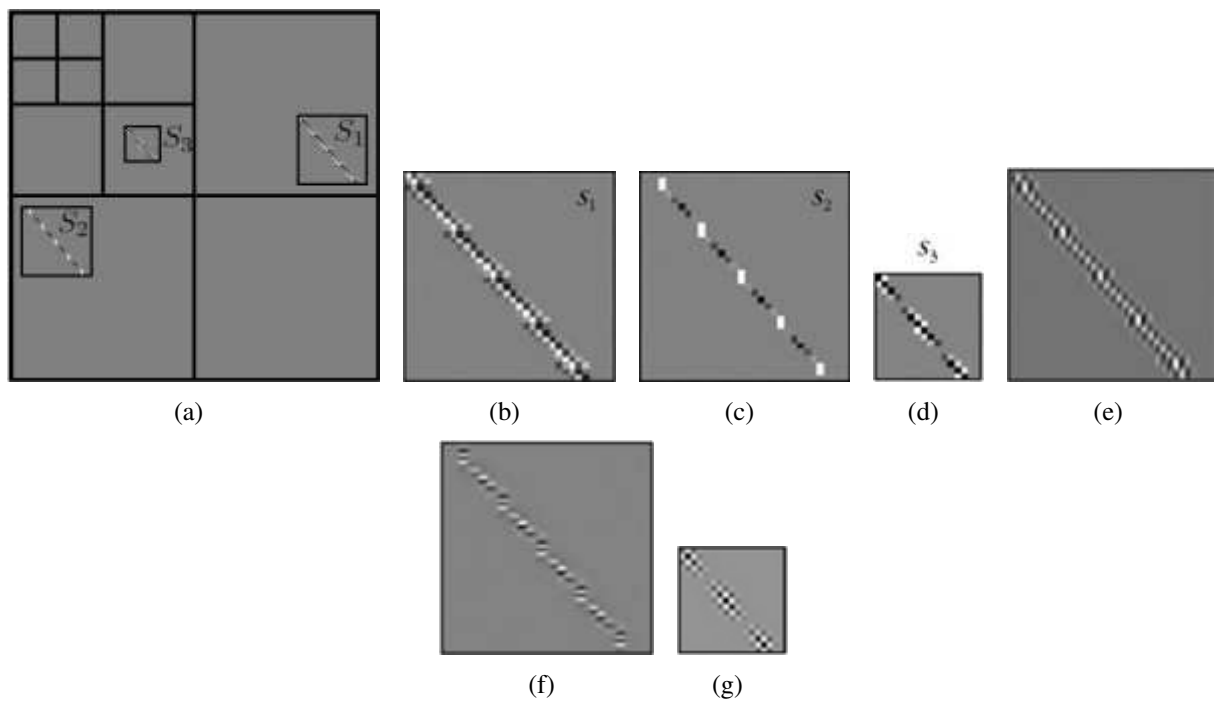


Fig. 3.28: Decomposição *wavelet* direcional: (a) Decomposição *wavelet* (b)-(d) *wavelets* direcionais Alpert de diferentes comprimento e largura. (e)-(g) Funções *bandelet* calculadas a partir das *wavelets* direcionais de (b)-(d) correspondentes aos quadrados  $S_1$ ,  $S_2$  e  $S_3$ , respectivamente, de (a). [3]



# Capítulo 4

## Método rápido para obtenção da orientação

### 4.1 Estado-da-arte e Método Proposto

Na Figura 4.1 é mostrado um diagrama de blocos que pode ser considerado geral para muitos métodos de compressão de imagens baseados em aproximação geométrica. Por agregar as vantagens de representação multirresolução das transformadas *wavelet* isotrópicas e a localização dos contornos das *wavelets* direcionais, as *bandelelets* têm sido consideradas estado-da-arte em várias aplicações de processamento de imagens [53][54][55][56][57]. No entanto, para aplicações de compressão ainda não tem desempenho satisfatório quando comparadas com algoritmos padronizados como o JPEG 2000, devido ao alto custo computacional e à taxa adicional necessária para codificar os parâmetros do dicionário. [3] Os procedimentos computacionais tradicionalmente envolvidos na decomposição e codificação *bandelet* de 2º ordem estão resumidos na Figura 4.2.

O presente trabalho propõe melhorias inovadoras a aplicação das *bandelet* na codificação de imagens, buscando otimizar os blocos representados na Figura 4.2, com objetivo de redução da complexidade de processamento e de melhorar o desempenho da codificação *wedgelet* quanto a qualidade da imagem. A Tabela 4.1 resume as técnicas possíveis de serem usadas em cada etapa do processo e nas próximas seções serão detalhadas essas etapas.

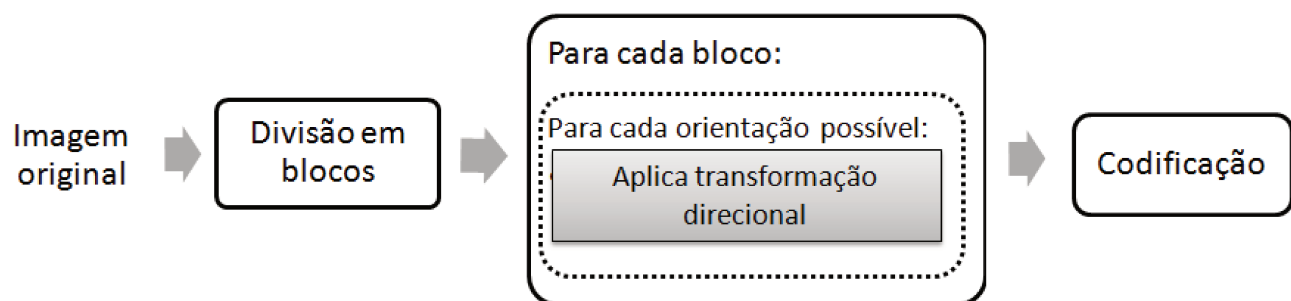


Fig. 4.1: Diagrama de blocos geral da compressão utilizando métodos geométricos.

<b>Etapa do processo de compressão</b>	<b>Técnicas para melhorar a qualidade da imagem</b>	<b>Técnicas para redução da complexidade computacional</b>
Cálculo dos coeficientes de aproximação em cada segmento de imagem	Dicionários ( <i>wedgelets, arclets, platelets, surflets...</i> ) ou transformadas direcionais ( <i>directionlets, curvelets, contourlets, bandelets, ...</i> )	Teorema de Green, somas parciais, processamento em blocos, momentos geométricos, detectores de contorno
Cálculo da árvore <i>quadtree</i>	Árvores não diádicas, partições H-V	Modelo geométrico multiescala com dependência entre as orientações de diferentes escalas e/ou localidades, por meio de uma cadeia de Markov
Obtenção da melhor árvore para uma determinada relação taxa-distorção	Novas funções de custo envolvendo estimadores de entropia e outras medidas de qualidade além do erro quadrático, como o SSIM [59]	Cálculos prévios que evitem cortes desnecessários na árvore ou permitam junções de nós filhos ( <i>join-prune</i> )
Codificação dos parâmetros	Codificação diferencial com modelos preditivos a partir do nó pai e entropia de ordem elevada para os parâmetros de orientação e árvore <i>quadtree</i>	Estabelecimento a priori de símbolos, de acordo com estudos feitos em imagens teste

Tab. 4.1: Técnicas inovadoras propostas no presente trabalho para otimização da decomposição *bandelet*.

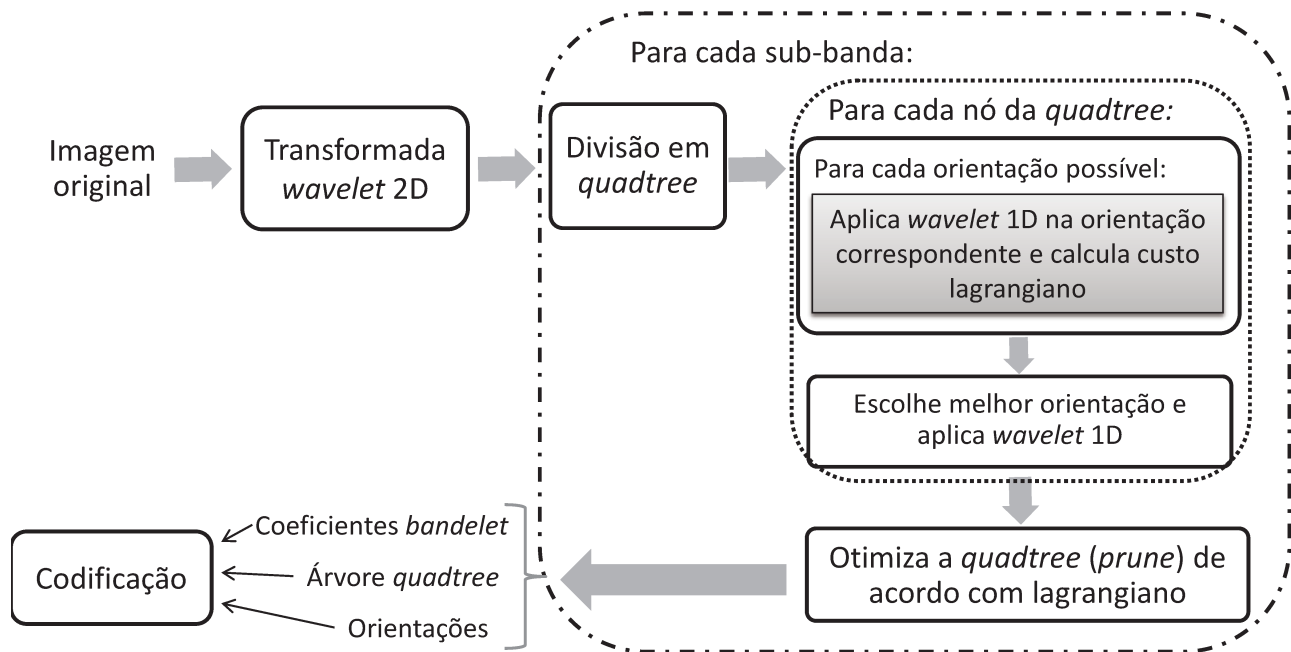


Fig. 4.2: Diagrama de blocos da compressão utilizando decomposição *bandelet* de 2º ordem.

#### 4.1.1 Cálculo dos Coeficientes de Aproximação em Cada Segmento de Imagem

A maior dificuldade em aplicar os novos métodos de representação apresentados é na determinação de uma geometria apropriada a partir de uma imagem. Para a decomposição *bandelet*, a geometria corresponde à partição da imagem em regiões  $\Omega_i$  e pelo fluxo geométrico presente em cada região. O conceito de uma geometria ótima pode ser bem definido no contexto de uma aplicação específica, como presente no caso em que será concentrado na aplicação de compressão de imagens.

Como já foi mostrado no Capítulo 3, dentre os métodos de aproximação geométrica as transformadas *bandelet* são as que conseguem melhor desempenho, por aliam as propriedades da aproximação *wavelet* com a preservação da geometria. A maior desvantagem ainda não tratada dessa técnica é o alto custo computacional (conforme Tabela 3.1). Neste trabalho um novo método para redução da complexidade computacional da decomposição *bandelet* é desenvolvido, especificamente no que tange à busca pela melhor orientação em cada bloco de imagem. Enquanto o trabalho original de Peyre e Le Pennec [55] realiza uma busca exaustiva por todas as possíveis orientações, propõe-se aqui uma estimação prévia das orientações, similar ao que foi feito por Popovici [60] e por Lisowska [9] para a decomposição *wedgelet*.

Para a determinação da melhor orientação em cada região da imagem e atingir a geometria ótima, Lisowska [9] propôs um método de cálculo dos *wedgelets* que substitui as comparações excessivas por cálculo de momentos direcionais da imagem. Com esse método consegue-se complexidade  $O(N^2 \log N)$  para imagens de tamanho  $N \times N$ , enquanto que o método tradicional resulta em complexidade  $O(N^4 \log N)$ . Apesar de possíveis desvios da escolha ótima poderem existir, a principal vantagem do método proposto é o cálculo rápido. Portanto, é possível relacionar tempo de processamento com a qualidade do resultado. Para se atingir mais qualidade, maior tempo computacional é necessário.

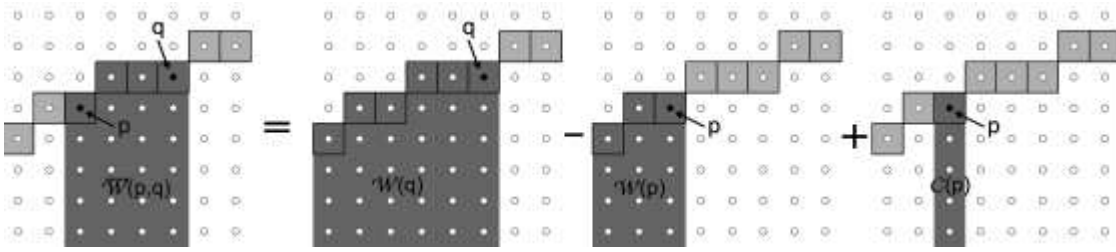


Fig. 4.3: Somas parciais para o cálculo da média em uma região com contorno bem definido. [61]

Na Seção 4.2 os momentos geométricos serão detalhados e relacionados com o presente trabalho. Além da determinação da geometria por meio de momentos geométricos customizados, na Seção 4.3 avaliaremos também o uso de operadores de detecção de contorno bastante conhecidos na área de processamento de imagens.

Uma outra abordagem para a determinação da orientação, que promete ser mais rápida que a usual mas que ainda é baseada em buscas exaustivas, é mostrada em [61], para o caso da decomposição *wedgelet*. Considerando especificidades do cálculo de momentos em domínios discretos, considerando o teorema de Green, obtém-se uma complexidade  $O(|\Theta| \cdot |S|)$ , onde  $|\Theta|$  indica o tamanho do conjunto de orientações possíveis e  $|S|$  é o número de pontos no quadrado diádico. Na Figura 4.3 é caracterizada a idéia principal do referido trabalho, que consiste em calcular parâmetros como médias através de somas parciais e aproveitá-las para o cálculo em diferentes orientações.

#### 4.1.2 Obtenção da Melhor Árvore *Quadtree*

A forma usual de obter decomposições *bandelet* é através de uma abordagem gulosa (*greedy*), obtendo aproximações ótimas para cada partição da imagem, em uma estrutura *quadtree*. Para obter a aproximação *bandelet* em uma determinada partição usualmente é feita uma busca exaustiva considerando todas as possíveis orientações, de forma a minimizar a seguinte função lagrangeana:

$$L = \left\| \hat{f}_w - f_w \right\|^2 + \lambda \cdot \left\| \hat{f}_w \right\|_0 \cdot T^2 \quad (4.1)$$

Onde  $f_w$  e  $\hat{f}_w$  representam os coeficientes *bandelet* originais e quantizados, respectivamente,  $T$  é o fator de quantização e  $\lambda$  é um multiplicador de Lagrange, definido como  $\lambda = 3/28$ , como sugerido em [55]. O termo  $\left\| \hat{f}_w \right\|_0$  representa a norma  $\ell^0$ , ou seja, o número de coeficientes não nulos na representação quantizada.

Para uma imagem de  $N$  pixels, a abordagem gulosa leva a uma complexidade computacional assintótica  $O(N^2 \log N \cdot |\Theta|)$ , onde  $|\Theta|$  indica o número de possíveis orientações. No domínio discretizado das imagens digitais nem todos os ângulos são possíveis dependendo da escala, ou podem ser quantizados para os parâmetros mais próximos. Mesmo assim, para uma busca exaustiva ideal,  $|\Theta| = O(N)$  e, portanto, a complexidade é  $O(N^2 \log N)$ , ou seja, quadrática com relação ao número de pixels.

A utilização de árvores não-diádicas ou com partições do tipo H-V (divisões horizontal e vertical), além de apresentar um custo computacional excessivamente maior (pois a melhor divisão não é determinística e uma busca exaustiva é preciso ser feita), não apresenta ganhos consideráveis para

compressão, pois novos parâmetros precisam ser codificados para identificar as partições mais complexas.

### 4.1.3 Novas Funções de Custo

A função de custo envolvida no cálculo do lagrangeano (Equação 4.1) para a otimização das árvores *quadtree* é tradicionalmente construída como uma ponderação das métricas  $\ell^2$  e  $\ell^0$ , que caracterizam a perda de qualidade (mais especificamente o erro quadrático médio) e o número de coeficientes necessários para a reconstrução, respectivamente. Essas métricas podem ser alteradas de forma a garantir um melhor desempenho, seja em termos objetivos ou subjetivos.

Para a medida de perda de qualidade, muitos métodos têm sido propostos na literatura como forma de superar as deficiências da métrica  $\ell^2$  e do PSNR (advindo da relação logarítmica da métrica  $\ell^2$ ) em quantificar a qualidade da imagem. Como mostrado em [62], o PSNR não distingue problemas espalhados (provenientes por exemplo de ruído aditivo) de problemas localizados em estruturas geométricas, tais como efeito de bloco (Figura 2.2) e *ringing* (Figura 2.16).

A proposta mais aceita atualmente para métrica objetiva de qualidade como alternativa ao PSNR é o SSIM (*Structural SIMilarity*) proposta por Wang e outros [59] e tem como principal paradigma a separação da percepção de qualidade de imagem nos aspectos de luminância, contraste e estrutura. Embora esse método funcione bem para avaliar as imagens processadas de forma comparativa, ele não é multirresolução, o que impede sua aplicação no domínio *wavelet*, ou mesmo *bandelet*. Pela sua estrutura não-linear, envolvendo estatística de primeira e segunda ordens dos *pixels* de uma vizinhança, sua transposição para o domínio *wavelet* não é simples e nem garante uma correlação de qualidade com a aplicação no domínio original da imagem após a decodificação.

Para este trabalho foi mantida a métrica  $\ell^2$  de forma a evitar cálculos adicionais que atrapalhariam o objetivo principal que é reduzir a complexidade computacional. Além disso, para comparação com outros métodos da literatura, a medida de PSNR ainda é a mais aceita e disponível.

No caso do número de coeficientes, uma métrica alternativa que poderia representar melhor a taxa de *bits* após a compressão seria a entropia. No entanto, trata-se também de uma medida não-linear, que depende da estatística da imagem toda (ou dos coeficientes após a transformação) e que pode mudar sensivelmente após cada variação na *quadtree*. Por isso, a métrica  $\ell^0$  também foi mantida no cálculo do lagrangeano da Equação 4.1.

### 4.1.4 Codificação

Para a codificação, a forma mais simples consiste em usar o mesmo número de *bits* em cada nó e armazenar a orientação e a estrutura da árvore *quadtree*. Uma forma mais sofisticada é armazenar apenas as diferenças de parâmetros em relação a parâmetros estimados pelos níveis superiores da árvore. Para imagens com geometria regular, pode-se estimar a orientação em um nível de resolução mais detalhado a partir do nó pai, com um pequeno desvio máximo permitido, codificando apenas as diferenças (Figura 4.4). A diferença da orientação é codificada por meio de um dos métodos em estado-da-arte de codificação de entropia, de acordo com as probabilidades de ocorrência.

Embora a abordagem preditiva a partir do nó pai favoreça contornos com geometria regular, sem mudança abrupta na orientação da descontinuidade, para as imagens testadas não houve ganho significativo e a complexidade computacional foi maior.

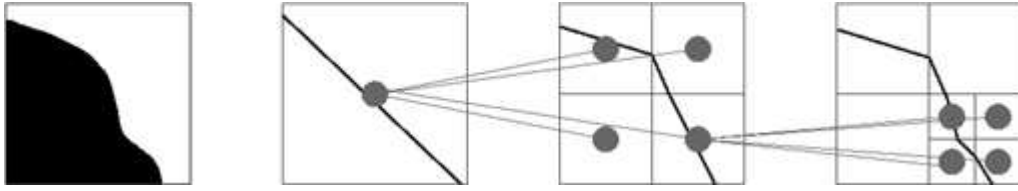


Fig. 4.4: Estimativa do parâmetro angular do *wedgelet* a partir de níveis superiores. [31]

## 4.2 Método de Momentos

Na literatura de processamento digital de sinais existem muitos algoritmos de otimização para o cálculo de momentos em imagens digitais. Em [12] são mostrados vários esquemas usados em reconhecimento de padrões e detecção de objetos. É importante notar que o melhor método para cada caso varia de acordo com os requisitos de *hardware* disponíveis e com o tipo de imagem a se tratar.

### 4.2.1 Momentos Geométricos

Momentos geométricos são uma metodologia de análise de imagens muito usada em diversas áreas, como identificação de objetos, correspondência entre cenas, análise de formas, reconhecimento de caracteres, caracterização de texturas, dentre muitas outras, como mostrado em [13],[14],[15] e [16].

Considerando uma função bidimensional  $f(x, y)$  como uma função densidade, os momentos de ordem  $(p + q)$ , denotados como  $M_{p,q}$ , são definidos como: [15]

$$M_{p,q} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad (4.2)$$

Por razões práticas, os momentos geométricos são muitas vezes definidos na forma discreta. Considerando  $f[x, y]$  definida no domínio discreto  $x = 0, 1, \dots, M$  e  $y = 0, 1, \dots, N$ : [15]

$$M_{p,q} = \sum_{y=0}^N \sum_{x=0}^M x^p y^q f[x, y] \quad (4.3)$$

Onde  $p$  e  $q$  são inteiros não negativos. Os momentos geométricos podem assumir valores muito altos com o aumento da ordem, levando a instabilidade numérica e a alta sensibilidade a ruídos.

Alguns momentos geométricos são amplamente usados em análise estrutural. O momento de ordem zero,  $M_{0,0}$ , representa a massa total de uma distribuição  $f(x, y)$ : [13]

$$M_{0,0} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy \quad (4.4)$$

Quando  $f(x, y)$  é uma imagem binária,  $M_{0,0}$  representa sua área. As razões entre os momentos geométricos de primeira ordem e o momento de ordem zero determinam as coordenadas do centro de massa: [13]

$$M_{0,1} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y f(x, y) dx dy \quad (4.5)$$

$$M_{1,0} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x f(x, y) dx dy \quad (4.6)$$

$$\bar{x} = M_{1,0}/M_{0,0}, \bar{y} = M_{0,1}/M_{0,0} \quad (4.7)$$

Os momentos de segunda ordem ( $M_{0,2}$ ,  $M_{1,1}$ ,  $M_{2,0}$ ), conhecidos como momentos de inércia, podem ser usados para determinar a orientação da imagem, ou direção dos eixos principais. [13]

### 4.2.2 Momentos Geométricos Customizados

O método de momentos geométricos customizados (*custom-built*) procura por uma parametrização em uma região quadrada limitada por um contorno definido por uma linha reta, definido pelo seguinte polinômio geral:

$$ax + by = c \quad (4.8)$$

A direção é dada pelo arco-tangente deste contorno:

$$\theta = -\arctan\left(\frac{b}{a}\right) \quad (4.9)$$

Para o cálculo dos momentos são usados polinômios definidos pelas derivadas parciais da seguinte função:

$$F(x, y) = \begin{cases} (1 - x^2)(1 - y^2) & \text{se } (x, y) \in [-1, 1] \times [-1, 1] \\ 0 & \text{caso contrário} \end{cases} \quad (4.10)$$

Esta função é construída com suporte em  $S = [-1, 1] \times [-1, 1]$ , de modo a apresentar valor nulo fora de  $S$ . As derivadas de interesse são:

$$A = \frac{\partial F}{\partial x} = -2x(1 - y^2), \quad B = \frac{\partial F}{\partial y} = -2y(1 - x^2) \quad (4.11)$$

Os parâmetros  $a$  e  $b$  do polinômio que define a região (Equação 4.8) são os momentos obtidos pelas integrais ponderadas pelas derivadas acima definidas:

$$a = \int \int_S A \cdot I dx dy = 2 \int \int_S x(y^2 - 1) \cdot I dx dy, \quad b = \int \int_S B \cdot I dx dy = 2 \int \int_S y(x^2 - 1) \cdot I dx dy \quad (4.12)$$

Esses momentos também podem ser escritos como combinações lineares de momentos geométricos usuais de ordem menor que 4:

$$a = 2(M_{1,2} - M_{1,0}), \quad b = 2(M_{2,1} - M_{0,1}) \quad (4.13)$$

Os cálculos dos momentos gastam tempo linear em relação ao número de *pixels* no bloco de imagem. Considerando uma imagem de  $N$  *pixels*, irá levar  $O(N)$  para o cálculo da direção na imagem toda. Em uma decomposição em diferentes escalas, o número total de *pixels* será  $O(N \log N)$ . Dessa forma, esse método pode ser usado para estimar a direção da transformada *bandelet* (assim como das *wedgelets*) em cada bloco de imagem, em tempo assintoticamente menor.

### 4.3 Operadores de Detecção de Contornos

Detecção de contornos é o nome dado a um conjunto de métodos matemáticos que têm como objetivo identificar pontos em uma imagem digital em que o brilho muda abruptamente, ou mais formalmente, que a função correspondente à imagem apresente uma descontinuidade. Os pontos em que o brilho da imagem muda abruptamente estão normalmente arranjados em um conjunto de segmentos de curva denominados contornos. A detecção de contornos é uma ferramenta essencial em processamento de imagens, visão de máquina e visão computacional, particularmente em aplicações de detecção e extração de características. [11]

Uma classe bastante conhecida de métodos de detecção de contornos em imagens são os operadores diferenciais discretos, que calculam a aproximação do gradiente da função de intensidade da imagem. Os métodos de Prewitt [63] e Sobel [64] são baseados na convolução da imagem com filtros pequenos, separáveis e de valores inteiros, nas direções horizontal e vertical e portanto são relativamente econômicos em termos de cálculo computacional. Esses métodos tentam aproximar a seguinte operação do domínio contínuo:

$$G = \int \int (\nabla f) dx dy \quad (4.14)$$

$$G_y = \int \int \left( \frac{\partial f}{\partial y} \right) dx dy, \quad G_x = \int \int \left( \frac{\partial f}{\partial x} \right) dx dy \quad (4.15)$$

Para o operador de Prewitt, a discretização adquire a seguinte forma:

$$G_y = \sum_y \sum_x f[x, y + 1] - f[x, y - 1], \quad G_x = \sum_y \sum_x f[x + 1, y] - f[x - 1, y] \quad (4.16)$$

Essas equações levam às seguintes matrizes  $3 \times 3$  que são convoluídas com a imagem:

$$\mathbf{G}_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} * f, \quad \mathbf{G}_x = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * f \quad (4.17)$$

Para o operador de Sobel a discretização adquire a seguinte forma:

$$G_y = \sum_y w[y] \sum_x f[x, y + 1] - f[x, y - 1], \quad G_x = \sum_x w[x] \sum_y f[x + 1, y] - f[x - 1, y] \quad (4.18)$$



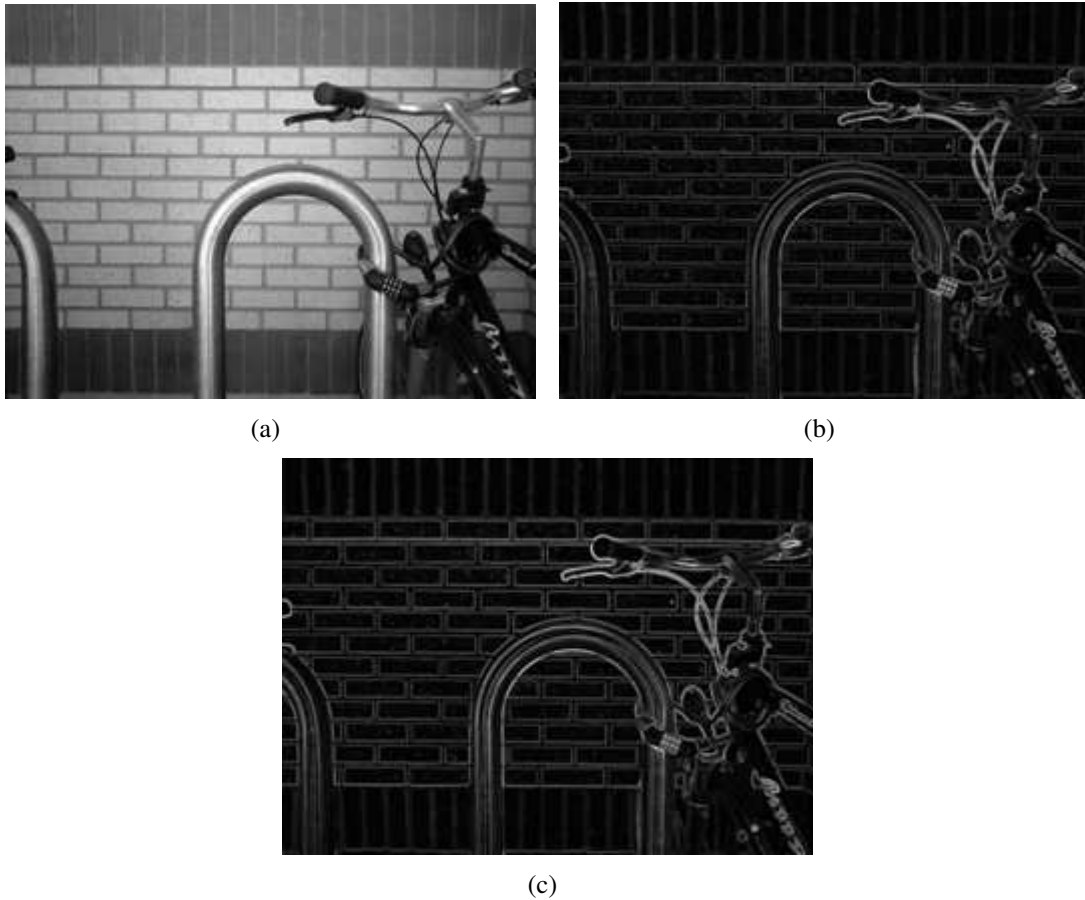


Fig. 4.5: Magnitude do gradiente por diferentes operadores (a) Imagem original (b) Prewitt (c) Sobel.

Onde  $w[\cdot]$  denota o vetor de pesos  $w = [1 \ 2 \ 1]$ . Essas equações levam às seguintes matrizes  $3 \times 3$  que são convoluídas com a imagem:

$$\mathbf{G}_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * f, \quad \mathbf{G}_x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * f \quad (4.19)$$

Após a convolução com os operadores, a magnitude do gradiente é obtida por:

$$|G| = \sqrt{\mathbf{G}_y^2 + \mathbf{G}_x^2} \quad (4.20)$$

A Figura 4.5 mostra uma comparação visual do desempenho dos três operadores para estimar a magnitude do gradiente de uma imagem.

As estimativas de orientação local de cada trecho da imagem podem ser obtidas pela direção do gradiente:

$$\theta = \arctan 2(\mathbf{G}_y, \mathbf{G}_x) \quad (4.21)$$

Na Expressão ?? o termo  $\arctan 2$  indica uma função arco-tangente de dois argumentos. O propósito de usar dois argumentos ao invés de um é para agregar informação dos sinais das entradas de

forma a retornar o quadrante apropriado do ângulo calculado, que não é possível de ser obtido com a função arco-tangente de um único argumento.

Utilizando esses métodos, o cálculo das orientações tem uma complexidade de tempo assintoticamente linear em relação ao número de *pixels* do trecho de imagem em que se está sendo calculado. É possível ainda estender os filtros usados de forma a superar deficiências em imagens ruidosas com um tempo computacional maior mas com a mesma complexidade assintótica.

Contornos extraídos de imagens naturais complexas normalmente estão contaminados por fragmentação, ou seja, as curvas dos contornos não estão conectadas, faltando segmentos do contorno ou com contornos falsos não correspondendo a objetos ou outros fatores de interesse, complicando a tarefa de interpretação dos dados da imagem. [17]

## 4.4 Da Transformada *Bandelet* para a Transformada *Bandelet* Rápida

O algoritmo da Transformada *Bandelet* (BT - *Bandelet Transform*) como foi proposto originalmente por [55] é baseado em buscas exaustivas e escolhas gulosas. A seguir a formalização deste algoritmo com sua complexidade assintótica, conforme já esquematizado na Figura 4.2.

**Entrada:** Imagem original em escala de cinza (conjunto de dados 2D diádico)

**Resultado:** *stream* de bits

Aplica decomposição *wavelet* 2D multinível (algoritmo de Mallat);

**para cada sub-banda faça**

**para cada nó da quadtree faça**

**para cada orientação possível faça**

            Aplica decomposição *wavelet* 1D de 1 nível (*lifting*) na orientação correspondente;

            Aplica determinação da função de custo lagrangeana;

**fim**

        Escolhe orientação com menor custo;

        Aplica decomposição *wavelet* 1D de 1 nível (*lifting*) na orientação correspondente;

**fim**

    Escolhe entre manter o nó atual ou os nós-filhos;

**fim**

Codifica coeficientes *bandelet*, árvore *quadtree* e orientações;

**Complexidade:**  $O(N^2 \log N)$

**Algoritmo 1:** Compressão de imagens utilizando *bandeleets* - abordagem tradicional (BT).

O algoritmo da Transformada *Bandelet* Rápida (FBT - *Fast Bandelet Transform*) proposto neste trabalho é apresentado a seguir:

**Entrada:** Imagem original em escala de cinza (conjunto de dados 2D diádico)

**Resultado:** fluxo de *bits*

**para cada bloco de imagem faça**

    | Calcula estimativa da orientação local do bloco de imagem;

**fim**

Aplica decomposição *wavelet* 2D multinível (algoritmo de Mallat);

**para cada sub-banda faça**

    | **para cada nó da quadtree faça**

        | Aplica decomposição *wavelet* 1D de 1 nível (*lifting*) na orientação estimada  
        | previamente;

    | **fim**

    | Escolhe entre manter o nó atual ou os nós-filhos;

**fim**

Codifica coeficientes *bandelet*, árvore *quadtree* e orientações;

**Complexidade:**  $O(N \log N)$

**Algoritmo 2:** Compressão de imagens utilizando *bandelelets* - método rápido proposto (FBT).

Embora o método de momentos preveja muito bem a direcionalidade em blocos de imagens simples, não garante sempre bons resultados na relação taxa-distorção. Isso se deve ao cálculo ser feito em imagens com muito mais detalhes, em que a geometria não segue linhas retas no qual os momentos foram definidos. Além disso, a transformada *bandelet* de segunda geração é aplicada no domínio *wavelet*, mas a estimativa de direção é feita no domínio da imagem. Devido às subamostragens, não é possível garantir a mesma direção para todas as escalas. Por isso, as estimativas de direção obtidas por momentos foram usadas como forma de reduzir o espaço de procura do algoritmo de transformada *bandelet* original, fazendo um compromisso em termos de eficiência de compressão e tempo computacional, similar ao que foi feito em [9] para o caso de *wedgelets*. A busca é estendida em  $R$  ângulos centrados na orientação estimada.

Deve-se notar que, embora haja mais cálculos para serem feitos na busca, assintoticamente o tempo computacional permanece o mesmo pelo fato de que as operações adicionais serem feitas um número constante de vezes. No entanto, é possível aumentar tanto o tempo computacional de forma a fazer a busca com todos os ângulos possíveis e calcular realmente a *bandelet* por meio de buscas exaustivas. Lembrando que o número de ângulos possíveis é finito, pois estamos trabalhando em um quadrado diádico discreto, com extensão que depende do tamanho do bloco analisado. A variação não é linear, pois alguns ângulos utilizam a mesma ordem de *pixels* e para fins de cálculo da transformada são equivalentes.



# Capítulo 5

## Resultados

Neste capítulo serão apresentados os resultados experimentais obtidos com o método proposto. Primeiramente é feita uma comparação dos métodos usados para estimação de orientação para casos simples presentes em processamento de imagens (Seção 5.2). Embora nem todos os métodos sejam bem sucedidos nessa tarefa, ainda assim eles serão usados para a compressão das imagens como forma de validar nossa análise. Em seguida, é demonstrado o desempenho do método de transformada *bandelet* rápida (Seção 5.3) e do método com variação do espaço de busca de tamanho variável (Seção 5.4).

### 5.1 Introdução e Observações sobre as Simulações

Os algoritmos foram implementados em ambiente de programação Matlab 7.11.0 (R2010b) em um computador PC Intel Core 2-Duo 2.0 GHz, 3.0 GB de RAM Windows 7, isolado de outros processos externos. Os resultados obtidos são analisados com base no tempo de processamento, na qualidade visual das imagens recuperadas (análise subjetiva), na medida de PSNR, no número de componentes utilizados para a aproximação e no número de *bits* por *pixel* (bpp) utilizado na codificação. Para a medida de tempo foi usada a função *timeit* do Matlab. Para a transformada *wavelet*, foi usada CDF 9/7 com 3 níveis de decomposição em um esquema *lifting*.

A função *timeit* trata automaticamente os procedimentos de *benchmarking* usuais de “aquecimento” do algoritmo a ser avaliado, descobrindo quantas vezes cada função irá repetir em um laço, etc. *Timeit* compensa também a sobrecarga de tempo associada com medições do tipo tic/toc e com a chamada de funções. O valor retornado é a mediana de uma série de várias medições.

O ambiente Matlab foi escolhido por ser uma ferramenta padronizada na área de processamento de imagens (dentre outras), possuir uma linguagem de fácil implementação e já ter incluídas operações rotineiras em suas bibliotecas e funções, permitindo-nos concentrar apenas no desenvolvimento dos algoritmos. Assim, é de se esperar que a velocidade de processamento e o consumo de memória não sejam otimizados e os resultados apresentados aqui são de caráter apenas qualitativo, necessitando ainda de estudos mais específicos para a implementação em outros ambientes.

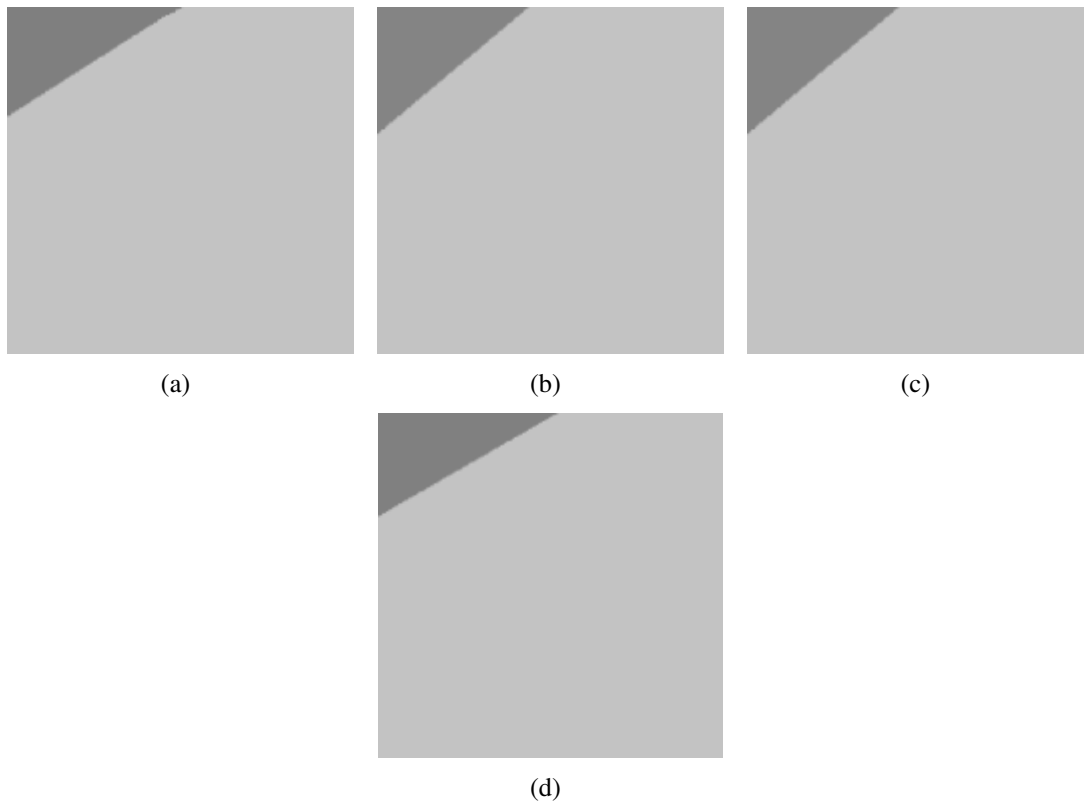


Fig. 5.1: Aproximação do caso de descontinuidade reta (a) Imagem original (b) Reconstrução pelo operador de Prewitt (c) Reconstrução pelo operador de Sobel (d) Reconstrução pelo cálculo de momentos.

## 5.2 Estimação da Orientação

A fim de validar os métodos de estimação de orientação, casos simples de imagens sintetizadas foram testados. Nas Figuras 5.1, 5.2, 5.3 e 5.4 são mostrados resultados da obtenção da orientação usando os momentos geométricos customizados e os operadores de contornos. São considerados 4 casos de testes: descontinuidade reta com regiões constantes (Figura 5.1(a)), descontinuidade curva (Figura 5.2(a)), uma região suave (Figura 5.3(a)) e ambas as regiões suaves (Figura 5.4(a)). Na Tabela 5.1 são mostradas as orientações obtidas com cada método.

A análise desses resultados revela a vantagem do método de momentos sobre outros estimadores de orientação. Embora consigam resultados similares no caso da descontinuidade reta, os operadores

	Prewitt	Sobel	Momentos
Descontinuidade reta (Figura 5.1)	0,6981	0,6981	0,5236
Descontinuidade curva (Figura 5.2)	0	0	0,3491
Uma região suave (Figura 5.3)	1,5708	1,5708	0,6981
Duas regiões suaves (Figura 5.4)	-1,5708	-1,5708	0,5236

Tab. 5.1: Orientação estimada por cada método.

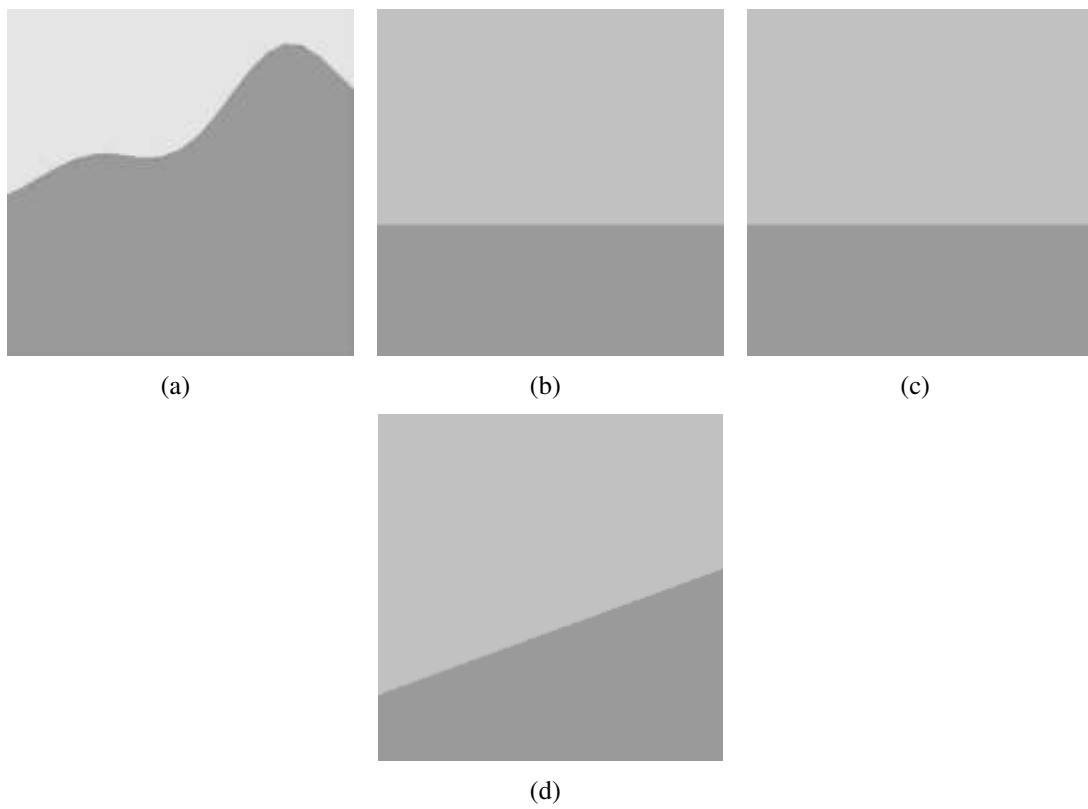


Fig. 5.2: Aproximação do caso de descontinuidade curva (a) Imagem original (b) Reconstrução pelo operador de Prewitt (c) Reconstrução pelo operador de Sobel (d) Reconstrução pelo cálculo de momentos.

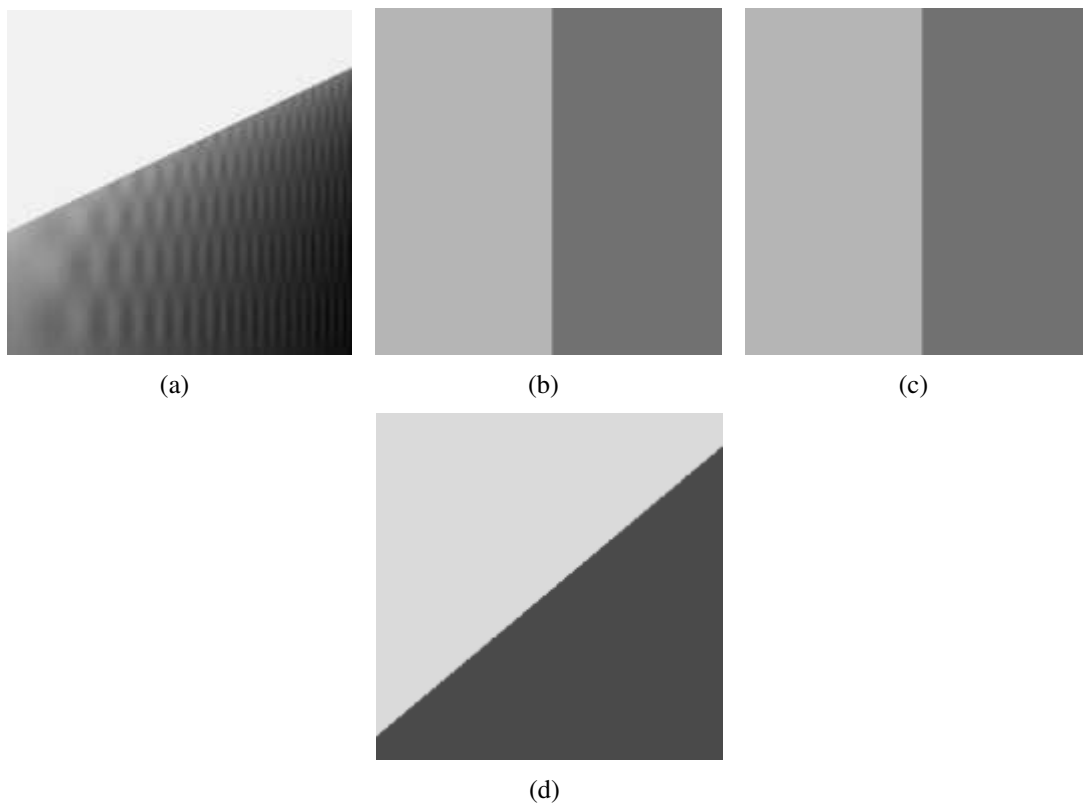


Fig. 5.3: Aproximação do caso de descontinuidade reta e uma região suave (a) Imagem original (b) Reconstrução pelo operador de Prewitt (c) Reconstrução pelo operador de Sobel (d) Reconstrução pelo cálculo de momentos.



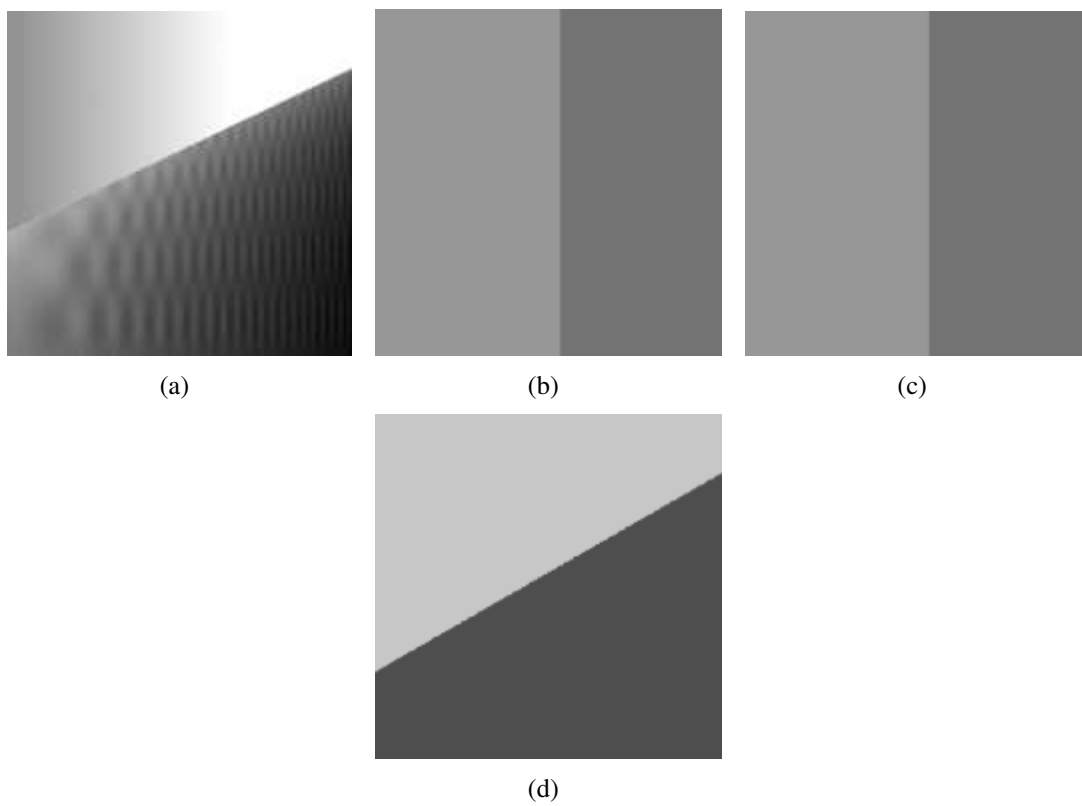


Fig. 5.4: Aproximação do caso de descontinuidade reta e duas regiões suaves (a) Imagem original (b) Reconstrução pelo operador de Prewitt (c) Reconstrução pelo operador de Sobel (d) Reconstrução pelo cálculo de momentos.

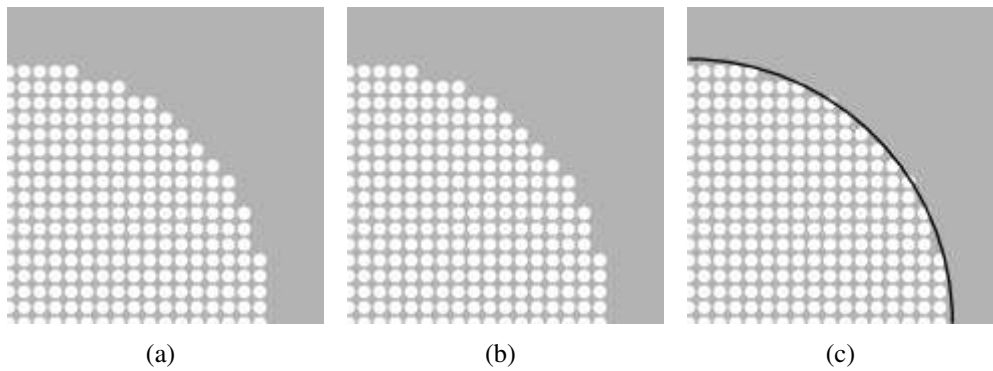


Fig. 5.5: Estimação de um círculo maior (“floresta”) composto por pequenos círculos menores (“árvores”) (a) Imagem original (b) Solução encontrada por operadores locais (c) Solução encontrada por um método baseado em momentos. [65]

de Prewitt e Sobel tiveram resultados desastrosos para imagens mais complexas. Isso ocorre porque esses operadores são muito localizados e não levam em conta o bloco como um todo. O método de cálculo por momentos procura por uma orientação levando em conta todo o bloco de imagem de uma vez. Na literatura de processamento de imagens diz-se que esse método busca a “floresta” e não as “árvores”, como ilustrado na Figura 5.5 [65]. Métodos baseados em análise frequencial ou em transformada Hough também buscam pelas “árvores”.

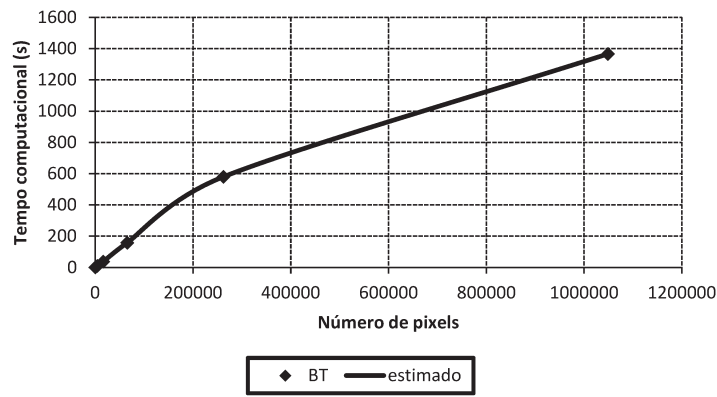
### 5.3 *Bandelet Rápida*

A Tabela 5.2 apresenta o tempo computacional para o método FBT usando os três estimadores de orientação e do método BT, para diferentes tamanhos de imagem. Para essa análise, apenas as etapas de determinação da decomposição *bandelet* e da codificação dos símbolos foram consideradas para a avaliação do tempo. Os símbolos foram codificados usando entropia de primeira ordem (código de Huffman), considerando estatísticas dos coeficientes *bandelet* e *wavelet*, da árvore de particionamento *quadtree* e dos parâmetros de orientação dos blocos (ângulos). O processo de decodificação, que é bem mais rápido e independente do método de estimação da orientação, não é levado em conta. A transformada *wavelet* sozinha também é mostrada para comparação.

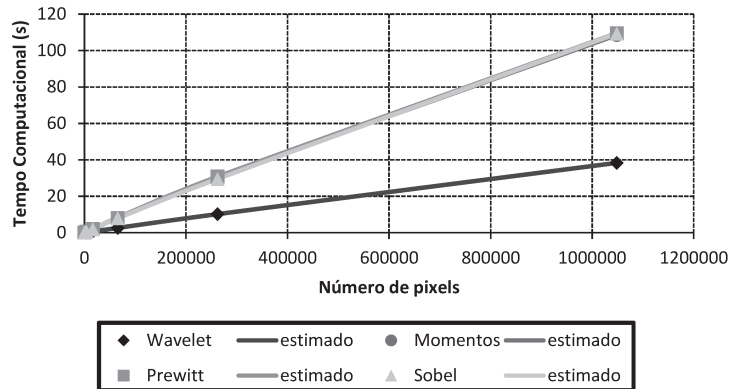
Pode ser observado pela Tabela 5.2 que o tempo computacional da FBT cai drasticamente quando comparado com a BT. Além disso, na Figura 5.6, os tempos estimados e medidas são apresentados. Os pontos representam medidas experimentais e as linhas sólidas constituem as estimativas por função de ordem  $O(N \log N)$  e  $O(N^2 \log N)$  para FBT e BT, respectivamente. Devido à grande diferença de desempenho entre o método tradicional e os métodos rápidos, os gráficos não são comparáveis entre si diretamente pois os eixos estão em escalas diferentes. Os três estimadores de orientação apresentaram desempenho similar, com pequena vantagem para o método de momentos geométricos customizados.

Tab. 5.2: Tempo computacional (segundos) para a imagem “Barbara”.

Número de <i>pixels</i>	TW	BT	FBT		
			Momentos	Prewitt	Sobel
64 (8 × 8)	0,0018	0,1234	0,0054	0,0053	0,0054
256 (16 × 16)	0,0088	0,4293	0,0287	0,0293	0,0292
1024 (32 × 32)	0,0368	2,3719	0,1151	0,1207	0,1206
4096 (64 × 64)	0,1550	9,1913	0,4710	0,5342	0,5459
16384 (128 × 128)	0,6267	36,9094	1,8931	1,9571	1,9459
65536 (256 × 256)	2,5659	157,6806	7,7951	7,9780	7,8898
262144 (512 × 512)	10,2086	579,5594	30,0524	30,8001	29,4732
1048576 (1024 × 1024)	38,2686	1365,7377	108,4803	109,4236	109,2457



(a)



(b)

Fig. 5.6: Tempo computacional medido e estimado para imagens de diferentes tamanhos: (a) BT, (b) TW e FBT.

## 5.4 *Bandelet* com espaço de busca limitado

Nesta seção será mostrado como uma estimativa a priori da orientação em cada bloco pode acelerar o algoritmo de decomposição *bandelet* e ainda obter resultados significativos em termos de qualidade de imagem. Centralizando a procura na orientação estimada, variamos o espaço de busca e obtemos resultados progressivamente melhores, embora com maior tempo computacional, até igualarmos ao conseguido com a busca exaustiva.

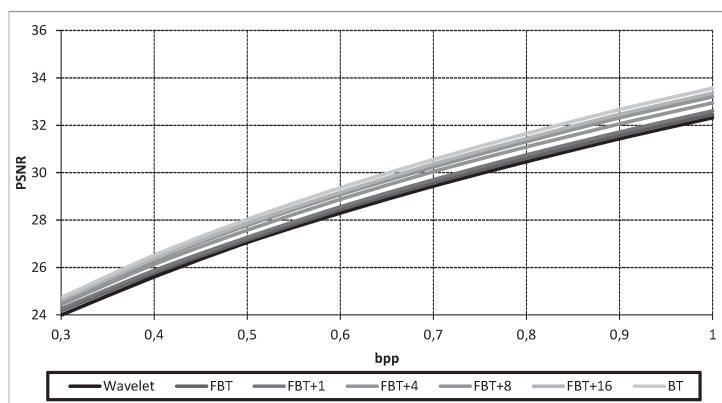
Na Figura 5.7 é apresentada a qualidade de imagem em termos de PSNR para diferentes taxas de *bits* (variando através do fator de quantização), diferentes tamanhos de espaço de procura e tamanho de imagem fixado em  $512 \times 512$ . O tamanho do espaço de busca varia entre 0 (FBT) até 16 e também infinito (BT). A transformada *wavelet* sozinha também é mostrada para comparação. De forma geral, consegue-se mais de 1 dB de ganho ao longo da curva e o maior ganho conseguido é quando se passa da TW para a FBT.

Na Figura 5.8 as curvas para diferentes tamanhos de espaço de procura com tamanho de imagem fixo  $512 \times 512$  e taxa de *bits* 1,0 bpp são combinadas, mostrando a relação entre qualidade de imagem e tempo computacional. O operador Prewitt de estimação de orientação foi o que obteve a melhor curva. Este comportamento foi observado mesmo para outras taxas de *bits* e tamanhos. Essa figura confirma que a transformada *bandelet* pode ser acelerada com perdas controláveis.

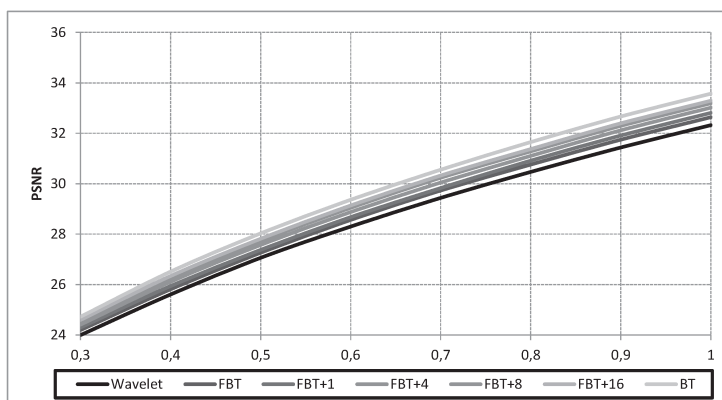
A codificação baseada em *bandelelets* superou a codificação baseada em *wavelets* em aproximadamente 0,5 dB para a imagem “Lena” e 1,5 dB para a imagem “Barbara”. É importante observar que a vantagem se mantém com outras imagens e tamanhos, principalmente em regiões com forte conteúdo geométrico, como contornos e alguns tipos de texturas.

A Figura 5.9 mostra um *zoom* em uma estrutura geométrica da imagem “Barbara”, comparando a imagem comprimida obtida com uma taxa aproximada de 0,2 bpp para diferentes métodos, similar ao resultado apresentado em [66]. A distorção é menor com a decomposição por *bandelelets* do que por *wavelets*, mesmo com o método rápido. Os padrões geométricos presentes nas linhas da calça do modelo não podem ser reconstruídos adequadamente pelos métodos baseados unicamente em *wavelets* isotrópicas quando em baixas taxas, ao contrário dos métodos com *wavelets* geométricas, que mesmo com o espaço de procura reduzido conseguiram um desempenho superior.

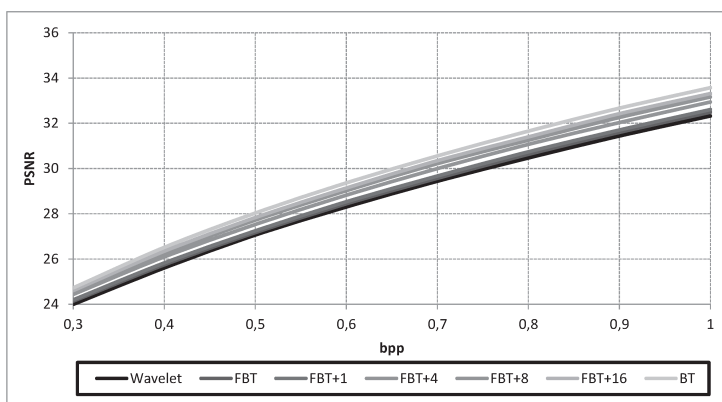
Na Figura 5.10 é mostrado um *zoom* da imagem “Lena” para comparação visual. Embora essa imagem apresente menor ênfase nas características geométricas (muitas regiões planas e fundo se confunde com a modelo), observa-se que na região dos lábios, quando utilizando apenas *wavelets* isotrópicas, ocorre uma suavização que inibe os detalhes e ao mesmo tempo existe a presença de *ringing* em outras regiões. Com as *wavelets* geométricas, embora o *ringing* não tenha sido totalmente limitado devido à baixa taxa e ao tamanho reduzido da imagem e das sub-bandas *wavelet*, a região dos lábios tem uma reprodução mais fiel à imagem original, mesmo com espaço de procura reduzido.



(a)



(b)



(c)

Fig. 5.7: Qualidade da imagem para diferentes tamanhos do espaço de procura: (a) Momentos, (b) Prewitt e (c) Sobel.

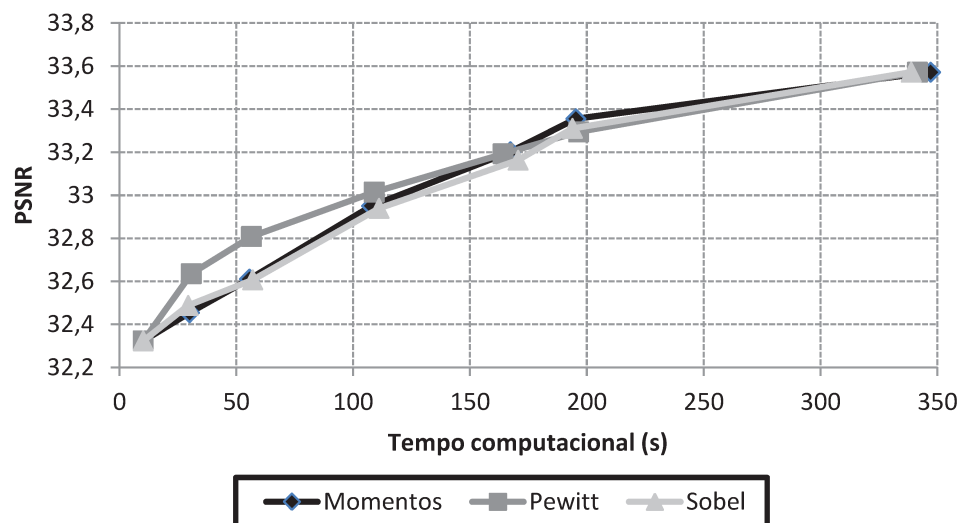


Fig. 5.8: Qualidade da imagem versus tempo computacional para diferentes tamanhos de procura com tamanho da imagem  $512 \times 512$  e taxa de *bits* 1,0 bpp.

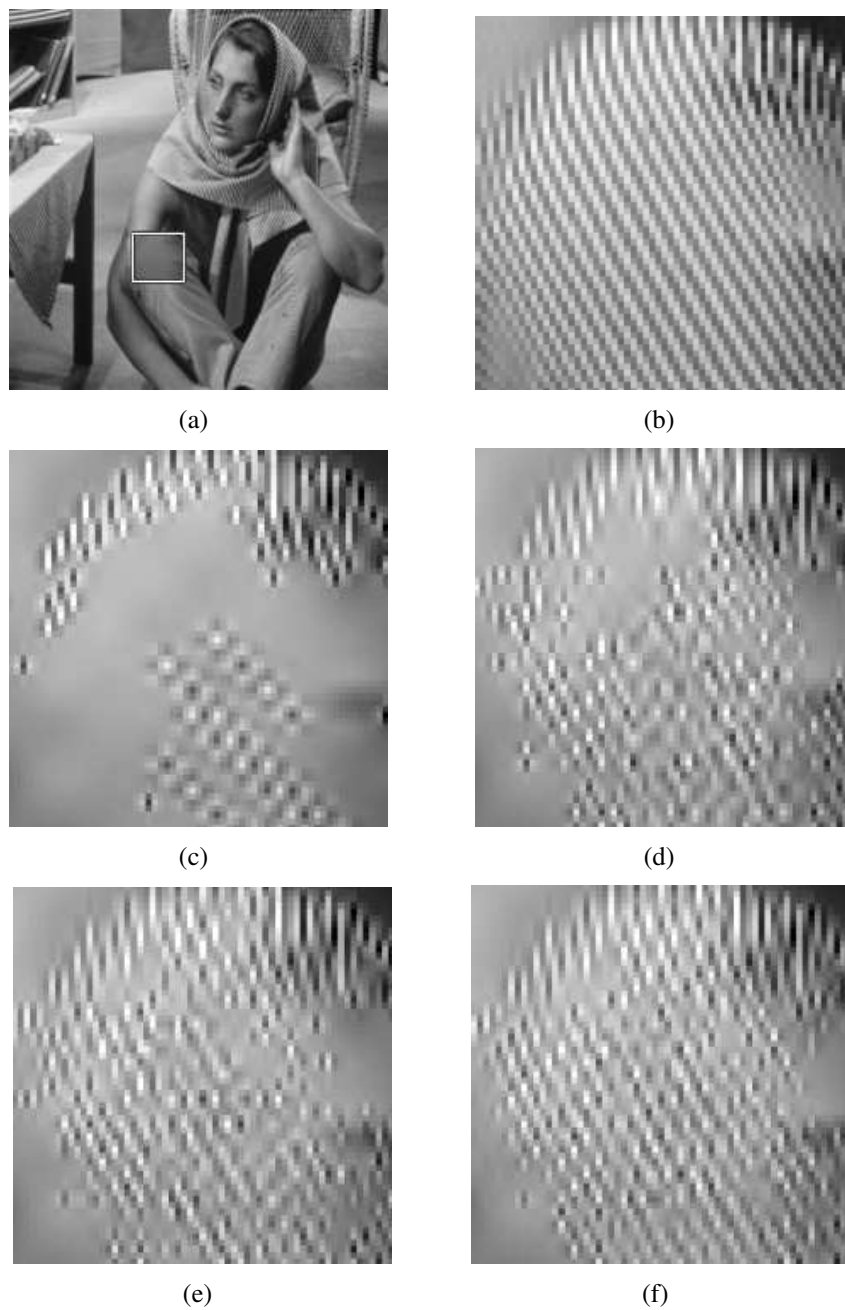


Fig. 5.9: Comparação da imagem “Barbara” comprimida em 0,28 bpp (a) Imagem original, (b) Imagem original (*zoom*), (c) TW, (d) FBT, (e) FBT+4, (f) BT.

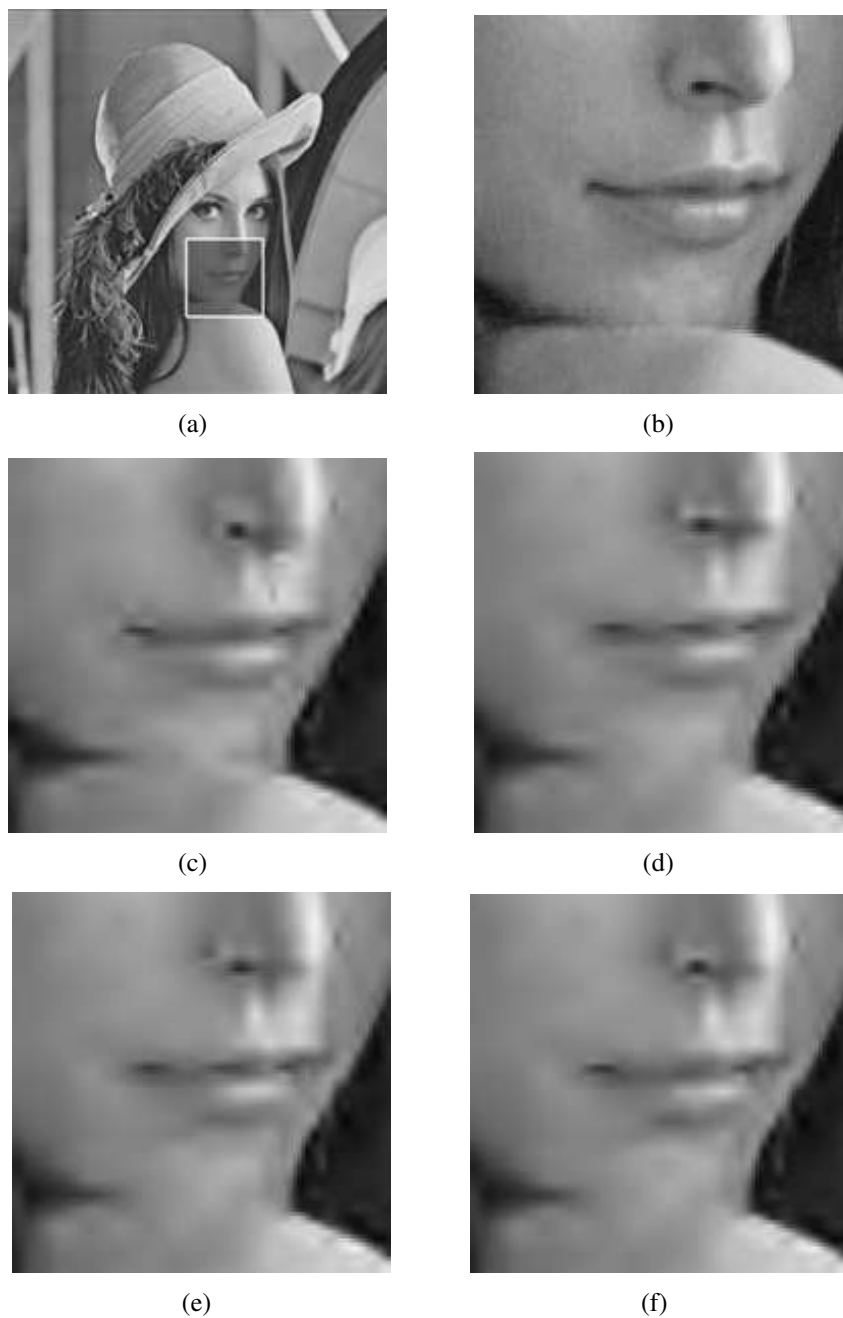


Fig. 5.10: Comparação da imagem “Lena” comprimida em 0,28 bpp (a) Imagem original, (b) Imagem original (*zoom*), (c) TW, (d) FBT, (e) FBT+4, (f) BT.



# Capítulo 6

## Aplicações da Presente Proposta

Os avanços na área de compressão de imagem e vídeo têm gerado uma ampla gama de aplicações para indústrias e serviços nas mais diversas áreas de atuação humana. Novas técnicas que sirvam como “ferramentas” para tratamento e compressão de imagem e vídeo rapidamente ganham interesse acadêmico e comercial. Neste capítulo iremos demonstrar algumas das áreas de pesquisa mais promissoras em processamento de imagens digitais e que podem ser beneficiadas pelos métodos baseados em geometria, em especial os métodos com processamento mais rápido, que são o foco deste trabalho.

### 6.1 Compressão de imagens e vídeo de alta resolução

A televisão é um dos bens de consumo mais presentes nos domicílios brasileiros. Cerca de 95,1% da população possui um aparelho de TV em casa [67]. A programação transmitida pelas emissoras tem como objetivo informar, entreter, definir padrões de consumo e até mesmo orientar a agenda dos expectadores. Por ser um meio de comunicação passivo e simples, é usado por todas as faixas etárias e classes sociais. No contexto comercial não há como ignorar o alcance das chamadas publicitárias na venda de produtos e difusão de marcas. Além de um meio de comunicação e cultura estratégico, a televisão é também um grande propulsor de inovação tecnológica e desenvolvimento industrial.

Atualmente a televisão está em um processo de intensa evolução. A digitalização, que possibilitou melhor qualidade das imagens, maior resolução e também maior cobertura para as operadoras, iniciou-se em 2011 e tem prazo para terminar em 2016. Espera-se que em 2014 65% dos domicílios recebam a TV digital. [68]

O sistema de imagem chamado de HDTV é o que apresenta resoluções maiores do que o sistema de imagem tradicional (SD - *Standard Definition*). Com exceção de formatos analógicos adotados na Europa e Japão, o HDTV é transmitido digitalmente e por isso sua implementação geralmente coincide com a introdução da televisão digital. Tradicionalmente, o sistema *Half HD* possuía resolução de 720p (progressivo), sendo 1280 colunas e 720 linhas, ocupando 921.600 *pixels* por quadro, ou resolução de 1080i (entrelaçado), sendo 1920 ou 1440 colunas e 1080 linhas, ocupando 1.036.800 ou 777.600 *pixels*, respectivamente. Esse é o formato disponível para a televisão digital no Brasil e no mundo e também em mídias como o Blu-Ray. Vale lembrar que, além de maior resolução, os sistemas modernos possuem melhor qualidade de imagem pelos algoritmos envolvidos no processamento

e armazenamento dos dados.

Durante a década de 2000, pesquisadores japoneses da NHK propuseram formatos com ainda maior resolução, chamados de Ultra High Definition (UHDTV): sistema 4K (3840 colunas e 2160 linhas) e sistema 8K (7680 colunas e 4320 linhas). Para criar o conteúdo, a NHK combinou 16 câmeras HDTV para criar um único clipe de 30 minutos UHDTV. Em 2005, um programa UHDTV foi colocado no ar e transmitido a mais de 240km por meio da rede de fibras ópticas. No começo de 2012, depois de vários avanços da tecnologia, a NHK finalmente demonstrou a primeira câmera de vídeo de montagem no ombro UHDTV. Atualmente, a emissora japonesa transmite com sucesso UHDTV por ondas de rádio UHF convencionais a um raio de 4,2 km.

Além de maior resolução, o sistema UHDTV sistemas também apresenta mais cores do que o usual e até mesmo mais do que cinema. Para o áudio, os sistemas UHDTV possuem 22.2 canais em comparação com os 5.1 canais dos sistemas HDTV. Aos poucos, algumas marcas lançam equipamentos compatíveis com formatos UHDTV, como vídeo games e visualização pela internet. Transmissão em tempo real ainda é algo distante.

Considerando a taxa máxima de 120 quadros por segundo, um vídeo sem compressão na resolução UHDTV precisa de uma taxa de *bits* de 48 Gbps. O áudio 22.2 ocupa uma banda adicional de 50 Mbps. Após a compressão (utilizando um codec especial, chamado Super Hi-Vision), a taxa total chega a 500 Mbps. Para comparar, um sinal HDTV 1080p tem cerca de 10 Mbps.

As *bandelets*, bem como as outras transformadas de imagens, têm um papel fundamental na compressão de imagens e na melhoria de qualidade visual para a aplicação de televisão. Atualmente os padrões utilizam apenas a DCT e, em alguns casos, as *wavelets* (por exemplo, o Dirac [69]). Como os resultados da compressão utilizando *bandelets* são superiores às *wavelets* e DCT em todas as implementações, o único empecilho para a adoção em sistemas de TV ainda é o alto custo computacional. Deve-se levar em conta que os codificadores deixaram de serem vistos como tecnologia de alto custo e são focados nos usuários finais, em câmeras portáteis, *tablets*, celulares e PCs. Nas aplicações envolvendo tempo-real, como em programas ao vivo ou vídeo conferência, o alto custo computacional é proibitivo. A presente proposta pode então pode ser de grande ajuda para que os métodos de aproximação geométrica tenham apelo comercial maior na área de TV.

### 6.1.1 Super-Resolução

Uma área de pesquisa muito ativa desde o surgimento do HDTV é a Super-Resolução (SR). Com essa técnica, é possível transformar um conteúdo originalmente gerado por um sensor de baixa resolução em um conteúdo HDTV ou mesmo UHDTV. Existem limites para essa transformação no entanto, mas algoritmos cada vez mais complexos de estimação e interpolação são desenvolvidos de forma a tornar as imagens mais agradáveis ao olho humano.

Existem duas abordagens para a SR: de um único quadro ou de múltiplos quadros. A abordagem com múltiplos quadros usa deslocamentos *sub-pixel* entre várias imagens de baixa resolução de uma mesma cena, criando uma imagem com resolução aumentada pela fusão da informação. SR de um único quadro tenta aumentar a imagem sem adicionar embaçamento, usando partes diferentes de uma mesma imagem para estimar como a imagem de alta resolução deva ser.

Os algoritmos podem ser divididos em baseados no domínio do tempo e no domínio da frequência. Originalmente, métodos SR funcionavam bem apenas com imagens em escala de cinza [70], mas pesquisadores já propuseram métodos para imagens coloridas [71] e imagens 3D. [72]

*Bandelets* e outros métodos geométricos podem ser usados em algoritmos de super-resolução e garantir a preservação na geometria da imagem mesmo com a interpolação. Em [73] Mallat descreve o sistema desenvolvido por sua empresa Let It Wave para um conversor de vídeos em resolução SD para HDTV, utilizando interpolação temporal e *bandelets* para a interpolação intra-frame. Como todos os métodos baseados em geometria, o produto sofre com a alta complexidade envolvida, por isso a adaptação de algoritmos como o apresentado neste trabalho poderiam beneficiá-lo. Em 2008, a empresa Let It Wave foi comprada pela Zoran Corporation, mas os produtos da empresa aparentemente ainda não conseguiram sucesso comercial.

### 6.1.2 *Inpainting*

Um problema de interpolação similar à super-resolução é o processo de *inpainting*. Esse processo consiste em reconstruir partes de uma imagem ou de um vídeo, mesmo com partes perdidas ou deterioradas arbitrariamente. Em fotografia e cinema, é usada para a restauração de mídias baseadas em filme, para reverter a deterioração (rachaduras em fotografias ou arranhões e manchas de poeira no cinema). Também pode ser usado para remover olhos vermelhos, logotipos, datas e textos em uma fotografia, legendas ou mesmo objetos. Pode estar aliada a transmissão de vídeo, substituindo blocos perdidos. Um exemplo de *inpainting* é ilustrado na Figura 6.1.

Em [75] e [76] são mostrados exemplos de *inpainting* usando *bandelets*.

### 6.1.3 Sensoriamento Comprimido

No caso extremo, o processo de *inpainting* pode ser considerado um caso de sensoriamento comprimido (*compressed sensing*), isto é, quando se deseja recuperar a informação de um sinal com um número muito baixo de amostras (menor do que o limite de Nyquist-Shannon). É conhecido que um sinal pode ser perfeitamente reconstruído por suas amostras se a maior frequência presente no sinal for menor ou igual a metade da taxa de amostragem. A ideia principal é que, com conhecimento a priori da máxima frequência do sinal, menos amostras são necessárias para reconstruir o sinal. Em 2004, Emmanuel Candès, Terence Tao e David Donoho provaram que, com o conhecimento sobre a “esparsidade” de um sinal, este pode ser reconstruído com menos amostras.[77][78] A efetiva implementação dessa idéia, depende do projeto de um amostrador adaptativo e que seja eficiente para o tipo de sinal que esta sob análise (e para quais fins ele será usado). No caso de imagens com geometria bem definida, as bases *bandelet* apresentam-se como um bom modelo para esse amostrador.

## 6.2 Mapas de Profundidade

Na última década, muito progresso tem ocorrido em pesquisas envolvendo imagens tridimensionais. Como resultado, métodos envolvendo dados e imagens em 3D estão sendo empregados em várias aplicações, incluindo televisão, robótica, medicina e visão estereográfica. Em processamento de imagens, mapa de profundidade indica uma imagem, ou um canal presente em uma imagem, que contém informações a respeito das distâncias das superfícies em relação aos objetos a partir de um ponto de vista. [79]



(a)



(b)

Fig. 6.1: Exemplo de *inpainting* (a) Imagem original (b) Imagem processada, com objeto removido (reproduzido de [74]).

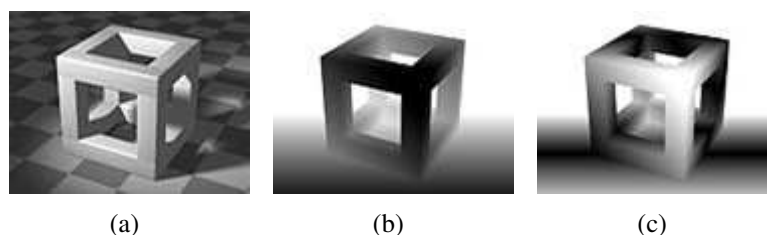


Fig. 6.2: Exemplos de mapa de profundidade: (a) Estrutura cúbica, (b) Superfícies próximas estão mais escuras (c) Superfícies próximas do plano focal estão mais escuras. [80]

Na Figura 6.2 são mostrados dois exemplos de mapa de profundidade. O primeiro (Figura 6.2(b)) mostra iluminação proporcional à distância da câmera, de forma que superfícies próximas estão mais escuras. O segundo (Figura 6.2(c)) mostra iluminação proporcional a distância de um determinado plano focal, de forma que as superfícies próximas a esse plano focal estão mais escuras.

Os maps de profundidade são usados em muitas aplicações, dentre elas:

- Simular o efeito de meios semi-transparentes uniformemente densos dentro de uma cena - como neblina, fumaça ou grandes volumes de água.
- Simular profundidade de campo rasa, onde algumas partes de uma cena parecem estar fora de foco. Mapas de profundidade podem ser usadas seletivamente para desfocar a imagem em diferentes graus. A profundidade de campo de uma lente específica é o intervalo de foco aceitável entre a lente principal e a objetiva. Aberturas maiores diminuem a profundidade de campo, enquanto aberturas menores aumentam-na. Técnicas de fotografia computacional processam a profundidade de campo digitalmente com o objetivo de modificar o foco para fins artísticos ou para se obter imagens com toda a cena igualmente em foco. [80]
- *Z-buffering* e *z-culling*, técnicas que auxiliam na renderização de cenas em 3D. Podem ser usadas para identificar objetos escondidos da vista e que seriam ignorados por alguns métodos de renderização. Torna-se particularmente importante para aplicações em tempo real como jogos, em que muitas renderizações precisam estar disponíveis e serem mostradas rapidamente.
- Mapeamento de sombra, usando os mapas de profundidade calculados a partir da perspectiva das luzes, não do observador.
- Prover informação de distância necessária para criação de autoestereogramas e outras aplicações relacionadas com a visualização 3D por stereoscopia. Em [81] e [82] são mostrados exemplos de abordagens para TV em 3D usando mapas de profundidade.
- Dispersão de superfície, uma técnica usada para aumentar o realismo simulando propriedades semitransparentes de materiais como a pele humana.

Alguns trabalhos mais tradicionais em codificação de imagens de mapa de profundidade tem usado algoritmos baseados em transformada derivados do JPEG-2000 [83] e do MPEG. A vantagem de usar padrões bem estabelecidos de codificação de vídeo para comprimir imagens de mapa de profundidade é a compatibilidade com tecnologias já existentes. No entanto, esses codificadores

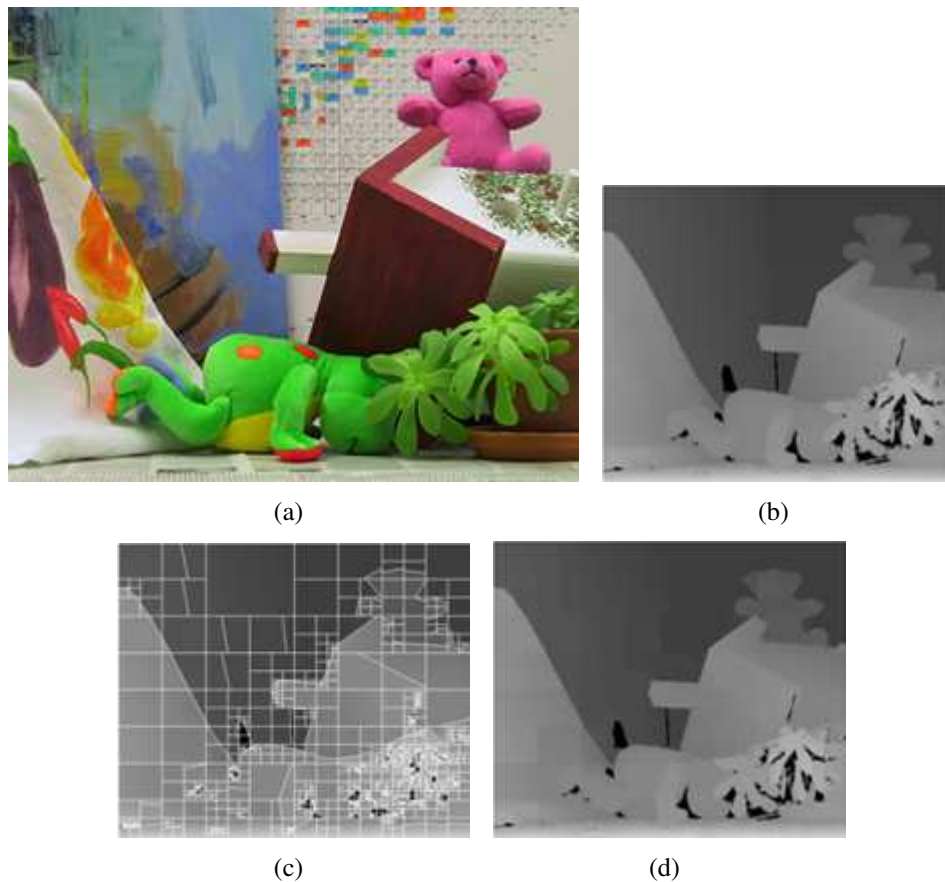


Fig. 6.3: Compressão de imagens de mapa de profundidade usando *platelets* (a) Projeção 2D de uma cena (imagem “Teddy”) (b) Mapa de profundidades original (c) Divisão *quadtree* (d) Correspondente mapa de profundidade reconstruído, com taxa de *bits* 0,12 bpp e PSNR de 36,1 dB. [86]

têm mostrado deficiências em representar os contornos sem deterioração em baixas taxas de *bit*. Visualmente nota-se artefatos como *ringing* ao longo dos contornos, que aparecem como sinais de profundidade borrados ao longo das bordas dos objetos. Uma abordagem alternativa é utilizar malhas triangulares [84], [85].

Em [86] é mostrado um método para codificação dos mapas de profundidade em um sistema de imagens 3D *multiview*. Na Figura 6.3 é mostrado o resultado dessa codificação. Observa-se que, como os objetos e superfícies aparecem como regiões suaves ou constantes nas imagens de mapa de profundidade, a representação por *platelets* é bastante adequada. A abordagem por *bandelelets* também pode ser facilmente adaptada, pois funciona bem com regiões de suavidades de diversos níveis.

Em [87] e [88] são mostradas aplicações usando *bandelelets* para avaliar a qualidade de imagens estereográficas.

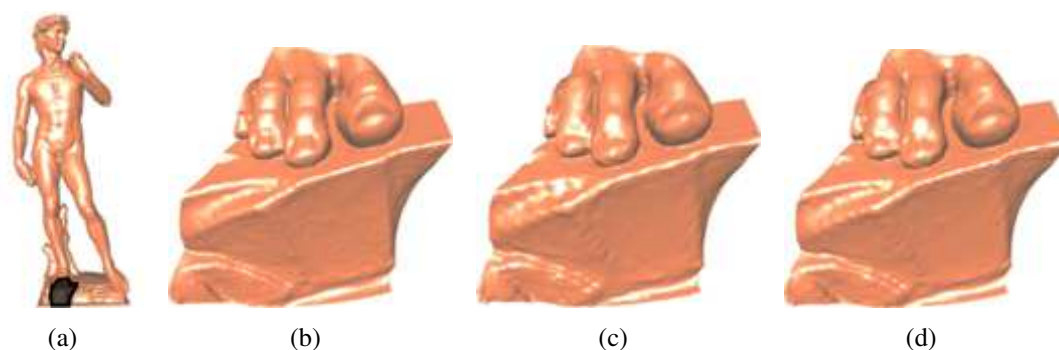


Fig. 6.4: Exemplo de codificação de superfícies: (a) Modelo original (b) Trecho original (c) Codificação *wavelet* com 0,2 bpv (d) Codificação *bandelet* com 0,2 bpp. [57]

### 6.3 Modelagem 3D

Embora os métodos de aproximação por geometria apresentados sejam baseados em estruturas bidimensionais, eles podem ser estendidos para 3 ou mais dimensões, de acordo com a dimensionalidade dos dados a serem analisados.

Em busca de um modelo de aproximações para superfícies tridimensionais, em [89] é apresentado um método baseado em *wavelets* direcionais, similar às *directionlets*. Dependendo da aplicação, o modelo por *surflets* (Seção 3.6) descrito em [90], ou algum método híbrido, pode ser mais indicado. Em [91] é mostrado um modelo mais simples, usando projeções bidimensionais e *wedgelets*. Em [57] é mostrado o desempenho da compressão de dados tridimensionais (superfícies) por meio de *bandelelets*. A Figura 6.4 mostra um exemplo da codificação por *bandelelets* para a modelagem 3D.

Todos esses métodos encaixam-se bem em problemas de modelagem de terrenos, em aplicações de sensoriamento remoto, ou em modelagem de objetos 3D, em aplicações de renderização para cinema, indústria e medicina. Por trabalharem com quantidades muito grandes de dados, podem ser beneficiados por métodos que tornem mais rápido o cálculo por meio de estimativas.

Operadores de detecção de contornos também foram propostos de maneira precisa para o caso 3D em [92] e [93], enquanto o método de momentos customizados de [65] pode ser adaptado para o caso 3D também, utilizando integrais triplas.

### 6.4 Segurança

Aplicações de segurança envolvendo *bandelelets* e outros métodos geométricos vão desde a detecção de pessoas por câmeras de vigilância [94] a identificação por biometria (seja por impressão digital [95] ou pela íris [96]). Em comum essas aplicações lidam com imagens de características geométricas específicas, que normalmente são perdidas quando são usados métodos baseados somente em características espaciais ou frequenciais.

## 6.5 Sensoriamento Remoto

Muitas outras aplicações industriais podem ser beneficiadas pelo aumento posterior de resolução quando a aquisição dos dados for difícil ou custosa. Em geofísica, a maior resolução possível deve ser recuperada de relativamente poucas medidas obtidas pela reflexão de ondas no subsolo ou no mar. Em radiografia, o tempo de exposição à radiação de um paciente e portanto a aquisição dos dados também são limitados. Para aplicações de sensoriamento remoto, aumentar a resolução envolve a colocação em órbita de novos satélites, um projeto bastante custoso.

O trabalho [97] apresenta o uso de *bandelelets* na área de sensoriamento remoto por imagens de satélite. O trabalho [98] apresenta um método similar de transformada geométrica mas usando transformada de Fourier no lugar da TW. Um caso mais específico é apresentado em [99], em que as *bandelelets* são usadas para remoção de nuvens das imagens, aplicando *inpaint*. Para imagens SAR (*Synthetic Aperture Radar*), têm-se os trabalhos: [100], [101], [102], [103] e [104].

## 6.6 Diagnósticos em Medicina

Em diagnósticos da medicina existem diferentes tipos de imagens, todos eles com propriedades geométricas importantes. Em geral, deseja-se reconhecer um determinado órgão ou tecido e, em caso de anomalia, poder visualizá-lo com clareza sem perigo de confundir com um possível ruído. Por isso, métodos de compressão devem ser usados sem perdas (*lossless*). Um método para compressão de imagens médicas usando *bandelelets* é apresentado em [105].

Em [8] é mostrado o uso de *platelets* para representar imagens de tomografia (PET, SPECT e microscopia confocal e infravermelho) e em [106] as *bandelelets* são comparadas a outros métodos para compressão de imagens de cromossomos. Para imagens de ressonância magnética, em [107] é mostrado o uso de *bandelelets* com um limitador *fuzzy*. Um dos resultados é mostrado na Figura 6.5.

Em [108] é mostrado o uso de *bandelelets* para melhoramento de imagens de ultrassom, especificamente para tirar ruído (*denoising*). A melhoria da qualidade visual para as imagens biomédicas, quando utilizadas cores, é o objetivo de [109].



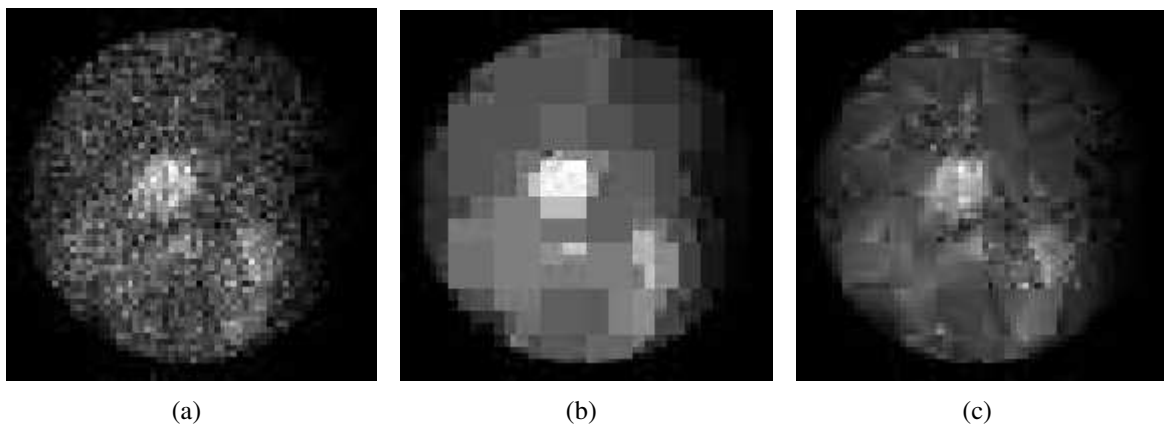


Fig. 6.5: Exemplo de aplicação de *platelets* em imagens médicas: (a) Imagem de medicina nuclear com ruído (exemplo cardíaco), (b) Aproximação por *wavelet* de Haar (c) Aproximação por *platelets*. [8]



# Capítulo 7

## Conclusões

Neste capítulo serão retomados os conceitos abordados no presente trabalho e as principais contribuições.

### 7.1 Comentários sobre o Contexto e Contribuições

A área de processamento de imagens encontra cada vez mais aplicações, deixando de ser objeto de estudo de apenas um caso particular de processamento de sinais para definir novos rumos de pesquisa e focos de atuação. Aplicações como as citadas no Capítulo 6 são só alguns dos campos de pesquisa mais recentes em que imagens digitais não são só um complemento mas um conjunto de dados essencial. Em todas essas aplicações a velocidade de processamento e de transmissão, bem como a compactação dos dados e a identificação da geometria, são características fundamentais a serem consideradas. Este trabalho foi realizado considerando essas questões.

A codificação por transformadas ortogonais, em especial as transformadas *wavelet*, é considerada atualmente o estado-da-arte. Com uma formulação matemática bastante complexa, as *wavelets* são aplicadas para diferentes objetivos em processamento de sinais, seja identificação de características, remoção de ruído, ou compressão de dados. No entanto, as *wavelets* separáveis falham na representação de descontinuidades como os contornos da imagem, espalhando a energia para muitos coeficientes e não provendo representação esparsa. Neste trabalho foi descrita a teoria básica das *wavelets* e onde essas falham.

Dentre os métodos propostos mais recentemente para aproximar imagens de forma esparsa considerando a geometria, a decomposição por *bandelelets* de 2ª geração, transformadas não isotrópicas construídas sobre os coeficientes da transformada *wavelet* 2D, foi escolhida. Essa transformada consegue aliar a capacidade de representar regularidades em imagens, captadas pelas *wavelets*, bem como a geometria dos contornos, ou mais especificamente dos fluxos geométricos, alinhamentos orientados de dados no domínio multirresolução. A obtenção da orientação ótima, que é a principal desvantagem desse método e tem inviabilizado sua adoção em esquemas comerciais, pôde ser realizada de forma mais rápida utilizando abordagens inéditas aqui neste trabalho, os momentos geométricos customizados e os detectores de contorno. Apesar de essas técnicas serem bem conhecidas pelos especialistas em processamento de imagens e já existirem muitas implementações em hardware e software, está é a primeira vez em que são usadas para o processamento de transformadas *bandelet*.

## 7.2 Comentários sobre Simulações e Resultados

Os resultados apresentados dos algoritmos propostos tiveram caráter apenas qualitativo, visto que a implementação foi feita em Matlab, sem quaisquer otimizações computacionais que estariam presentes em implementações comerciais. Ainda assim, os resultados apresentados foram comparáveis aos disponíveis na literatura especializada e comprovam o potencial do algoritmo proposto em tornar mais rápido o processo de aproximação de imagens por *bandelelets* sem perda significativa de qualidade. O ajuste de precisão se adequa a diferentes necessidades de acordo com as aplicações ou o ambiente em que for aplicado o algoritmo proposto.

Embora só tenham sido feitos testes com imagens naturais (fotografias), é esperado que o algoritmo tenha um desempenho ainda melhor com imagens sintetizadas ou imagens médicas. Como visto pelos resultados, o principal ganho é nas regiões em que a geometria é pronunciada, como em contornos e bordas de objetos, uma característica comum também nessas imagens.

O desempenho da compressão das imagens poderia ter sido ainda melhor, caso fosse feito um estudo aprofundado da codificação dos coeficientes produzidos pela transformada *bandelet* e dos parâmetros de orientação envolvidos. Uma codificação que leve em conta vizinhanças e contextos, como no caso do codificador EBCOT (*Embedded Block Coding with Optimized Truncation*), presente no padrão JPEG 2000 [110], poderia ser mais eficiente do que a entropia de primeira ordem, embora provavelmente seja computacionalmente mais lento. Esse estudo não foi abordado no trabalho pois fugiria do escopo de acelerar os métodos envolvidos, visto que a superioridade das *bandelelets* frente às *wavelets* já é conhecida.

## 7.3 Sugestões para Trabalhos Futuros

Com a evolução das unidades de processamento gráfico (GPU - *Graphic Processing Unit*) e a integração destas em dispositivos móveis, torna-se possível o uso de algoritmos que executem cálculos em paralelo (*pipeline*) [111]. Em trabalhos futuros, algoritmos de computação paralela que beneficiem todas as etapas do processo podem ser estudados, caso a arquitetura dos sistemas permita. Como uma primeira abordagem, em [112] é mostrado uma implementação em hardware usando uma FPGA (*Field-Programmable Gate Array*) e o processador Nios II da transformada *bandelet*, mas sem o algoritmo proposto neste trabalho. É possível que esse e mais outros trabalhos envolvendo *bandelelets* possam ser beneficiados pela presente proposta.

# Referências Bibliográficas

- [1] E.J. Candés, Laurent Demanet, and Ying D.L. Donoho. Fast discrete curvelet transforms. *Multiscale Modeling & Simulation*, 2005.
- [2] Stéphane Mallat and Gabriel Peyré. A review of Bandlet methods for geometrical image representation. *Numerical Algorithms*, 2007.
- [3] Stéphane Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 3rd edition, 2008.
- [4] V. Velisavljevic, B. Beferull-Lozano, M. Vetterli, and P. L. Dragotti. Low-rate reduced complexity image compression using directionlets. *Image Processing*, 2006.
- [5] Emmanuel J. Candés and David L. Donoho. Curvelets: a surprisingly effective nonadaptive representation of objects with edges. *Curve And Surface Fitting: Saint-Malo*, 2000.
- [6] David L Donoho. Wedgelets: Nearly minimax estimation of edges. *The Annals of Statistics*, 1999.
- [7] Technische Universitat München, Piyanut Pongpiyapaiboon, Piyanut Pongpiyapaiboon, and Piyanut Pongpiyapaiboon. *Development of Efficient Algorithm for Geometrical Representation based on Arclet Decomposition*. Master's thesis, Technische Universitat München, 2005.
- [8] RM Willett and RD Nowak. Platelets: a multiscale approach for recovering edges and surfaces in photon-limited medical imaging. *IEEE Transactions on Medical Imaging*, 2003.
- [9] Agnieszka Lisowska. Moments-Based Fast Wedgelet Transform. *Journal of Mathematical Imaging and Vision*, 2010.
- [10] G. Mallat. A theory for multiresolution signal decomposition : the wavelet representation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 1989.
- [11] Scott E. Umbaugh. *Digital Image Processing and Analysis: Human and Computer Vision Applications with CVIPtools, Second Edition*. CRC Press, Inc., 2nd edition, 2010.
- [12] A. L. I. A. Dalhoum. A comparative survey on the fast computation of geometric moments. *European Journal of Scientific Research*, 2008.
- [13] S. X. Liao. *Image Analysis by Moments*. PhD thesis, University of Manitoba, 1993.

- [14] P.T. Yap, R. Paramesran, and S.H. Ong. Image analysis using hahn moments. *IEEE transactions on pattern analysis and machine intelligence*, 2007.
- [15] L. Kotoulas and I. Andreadis. Image analysis using moments. *Information Systems Journal*, 2005.
- [16] R. Mukundan, S. H. Ong, and P. a Lee. Image analysis by tchebichef moments. *IEEE transactions on image processing*, 2001.
- [17] Tony Lindeberg. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 1996.
- [18] David L Donoho. Sparse Components of Images and Optimal Atomic Decompositions. *Constructive Approximation*, 1998.
- [19] N. Ahmed, T. Natarajan, and K.R. Rao. Discrete cosine transform. *Computers, IEEE Transactions on*, 1974.
- [20] Rangel Arthur. *Avaliação Objetiva de Codecs de Vídeo Avaliação Objetiva de Codecs de Vídeo*. Mestrado, Universidade Estadual de Campinas, 2002.
- [21] Fernando Silvestre and Yuzo Iano. *Procedimentos para Medição e Minimização do Efeito de Bloco Decorrente do Processamento Digital de Imagens (PDI)*. Mestrado, Universidade Estadual de Campinas, 2001.
- [22] Gregory K Wallace. The JPEG Still Picture Compression Standard. *Comm. ACM*, 1991.
- [23] Gary J. Sullivan, Pankaj Topiwala, and Ajay Luthra. The h.264/avc advanced video coding standard: Overview and introduction to the fidelity range extensions. *SPIE conference on Applications of Digital Image Processing XXVII*, 2004.
- [24] Arib std-b31 isdb-t standard for digital terrestrial broadcasting, transmission system for digital terrestrial television broadcasting, version 1.6. 2005.
- [25] Altera Corporation. White Paper The Quest for Digital Broadcast Quality : Addressing Quality Hot Spots. 2008.
- [26] Gary J. Sullivan, Jens-Rainer Ohm, Woojin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems Video Technology*, 2012.
- [27] Ingrid Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992.
- [28] Lajos Hanzo, Peter J. Cherriman, and Jürgen Streit. *Wireless video communications - second to third generation systems and beyond*. IEEE, 2001.
- [29] Kenneth R. Castleman. *Digital Image Processing*. Prentice Hall Press, Upper Saddle River, NJ, USA, 1996.

- [30] Ana Lúcia Mendes Cruz Silvestre da Silva and Yuzo Iano. *Procedimentos para método híbrido de compressão de imagens digitais utilizando transformadas wavelet e codificação fractal*. Doutorado, Universidade Estadual de Campinas, 2005.
- [31] M. Wakin, Hyeokho Choi, R. Baraniuk, and J. Romberg. Wedgelets: A multiscale geometric representation for images. *UCLA IPAM Tutorials on Multiscale Geometry and Analysis in High Dimensions*, 2004.
- [32] Vladimir Khryashchev, Ilya Apalkov, and Lev Shmaglit. A Novel Smart Bilateral Filter for Ringing Artifacts Removal in JPEG2000 Images. *Proc. of Int. Conf. on Computer Graphics and Vision*, 2010.
- [33] Hartmut Führ, Laurent Demaret, and Felix Friedrich. Beyond wavelets: New image representation paradigms. *Document and Image Compression*, 2006.
- [34] R. R. Coifman and M. V. Wickerhauser. Entropy based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 1992.
- [35] V. Velisavljevic, B. Beferull-Lozano, and M. Vetterli. Space-frequency quantization for image compression with directionlets. *IEEE Transactions on Image Processing*, 2007.
- [36] V. Velisavljevic, B. Beferull-Lozano, M. Vetterli, and P. L. Dragotti. Directionlets: Anisotropic multi-directional representation with separable filtering. *IEEE Transactions on Image Processing*, 2006.
- [37] Vladan Velisavljevic, M. Mrak, and A. Kondoz. Application of directionlets in video coding. *5th International Conference on Visual Information Engineering (VIE 2008)*, 2008.
- [38] M. S. Joshi, R. R. Manthalkar, and Y. V. Joshi. Image compression using curvelet, ridgelet and wavelet transform, a comparative study. *International Journal on Graphics, Vision and Image Processing (GVIP)*, 2008.
- [39] M. N. Do and M. Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *IEEE Transactions on Image Processing*, 2005.
- [40] A. G. Flesia, H. Hel-Or, A. Averbuch, E. J. Candes, R. R. Coifman, and D. L. Donoho. Digital implementation of ridgelet packets. *Studies in Computational Mathematics*, 2003.
- [41] Michael Wakin, Justin Romberg, Hyeokho Choi, and Richard Baraniuk. Geometric methods for wavelet-based image compression. *SPIE Conference*, 2003.
- [42] Michael B Wakin, Justin Romberg, Hyeokho Choi, and Richard G Baraniuk. Image compression using an efficient edge cartoon+ texture model. *Data Compression*, 2002.
- [43] Justin Romberg, Michael B Wakin, and Richard G Baraniuk. Multiscale Wedglet Image Analysis: Fast Decompositions and Modeling. *IEEE International Conference on Image Processing – ICIP*, 2002.

- [44] Laurent Demaret and H Führ. Wedgelet segmentations and approximation classes. *2nd Dolomites Workshop On Constructive Approximation And Applications*, 2009.
- [45] Agnieszka Lisowska. Second Order Wedgelets in Image Coding. *EUROCON - The International Conference on Computer as a Tool*, 2007.
- [46] A. Lisowska. Effective coding of images with the use of geometrical wavelets. *Proceedings of Decision Support Systems Conference*, 2003.
- [47] Michael Wakin, Justin Romberg, Hyeokho Choi, and Richard Baraniuk. Rate-distortion optimized image compression using wedgelets. *IEEE International Conference on Image Processing (ICIP)*, 2002.
- [48] Michael B Wakin, Justin K Romberg, Hyeokho Choi, and Richard G Baraniuk. Wavelet-domain approximation and compression of piecewise smooth images. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 2006.
- [49] Venkat Chandrasekaran, Michael B Wakin, and RD Baron. Representation and compression of multi-dimensional piecewise functions using surflets. *IEEE Transactions on Information Theory*, 2006.
- [50] Michael B. Wakin, R David L. Donoho, S Hyeokho Choi, and Richard G. Baraniuk R. The multiscale structure of non-differentiable image manifolds. in *Proc. Wavelets XI at SPIE Optics and Photonics*, 2005.
- [51] Albert Cohen, Wolfgang Dahmen, Ingrid Daubechies, and Ronald DeVore. Tree Approximation and Optimal Encoding. *Applied and Computational Harmonic Analysis*, 2001.
- [52] Agnieszka Lisowska. Smoothlets–multiscale functions for adaptive representation of images. *IEEE Transactions on Image Processing*, 2011.
- [53] Erwan Le Pennec and Stéphane Mallat. Bandelet Image Approximation and Compression. *Multiscale Modeling & Simulation*, 2005.
- [54] Gabriel Peyré and Stéphane Mallat. Orthogonal bandelet bases for geometric images approximation. *Communications on Pure and Applied Mathematics*, 2008.
- [55] Erwan Le Pennec and Stéphane Mallat. Sparse geometric image representations with bandelets. *IEEE transactions on image processing*, 2005.
- [56] Erwan Le Pennec. Geometrical image compression with bandelets. *Proceedings of SPIE*, 2003.
- [57] Gabriel Peyré and Stéphane Mallat. Surface compression with geometric bandelets. *ACM Transactions on Graphics (TOG)*, 2005.
- [58] B. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin. Wavelet-like bases for the fast solutions of second-kind integral equations. *SIAM J. Sci. Comput.*, 1993.



- [59] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004.
- [60] I Popovici and W.D. Withers. Custom-built moments for edge location. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.
- [61] Felix Friedrich, L. Demaret, Hartmut Fuhr, and Katrin Wicker. Efficient Moment Computation over Polygonal Domains with an Application to Rapid Wedgelet Approximation. *SIAM Journal on Scientific Computing*, 2007.
- [62] Zhou Wang and Alan Conrad Bovik. Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures. *IEEE Signal Processing Magazine*, 2009.
- [63] J.M.S. Prewitt. Object enhancement and extraction. *Picture processing and Psychopictorics*, 1970.
- [64] I. Sobel. An isotropic  $3 \times 3$  image gradient operator. *Sanford A.I. Project*, 1968.
- [65] Irina Popovici and William Douglas Withers. Introduction to Curve and Edge Parametrization by Moments. *Innovations in Intelligent Image Analysis*, 2011.
- [66] Gabriel Peyré and Stéphane Mallat. Discrete bandelets with geometric orthogonal filters. *International Conference on Image Processing (ICIP)*, 2005.
- [67] Fundação Instituto Brasileiro de Geografia e Estatística IBGE. *Censo Demográfico 2010*. 2010.
- [68] Mariana Durão e Agência Estado. Tv digital deve chegar a 65% da população neste ano. *Estadão Economia Negócios*, 2014.
- [69] Ahtsham Ali, Nadeem A. Khan, Shahid Masud, and Syed Farooq Ali. Survey of dirac: A wavelet based video codec for multiparty video conferencing and broadcasting. *Studies in Computational Intelligence*. 2011.
- [70] P. Cheeseman, B. Kanefsky, R. Kraft, and J. Stutz. Super-resolved surface reconstruction from multiple images. *Technical report*, 1994.
- [71] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar. Fast and robust multi-frame super-resolution. *IEEE Transactions on Image Processing*, 2004.
- [72] S. Schuon, C. Theobalt, J. Davis, and S. Thrun. Lidarboost: Depth superresolution for tof 3d shape scanning. *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- [73] Stephane Mallat. Super-resolution bandlet upconversion for hdtv. 2006.
- [74] Miguel Granados, Kwang In Kim, James Tompkin, Jan Kautz, and Christian Theobalt. Background inpainting for videos with dynamic objects and a free-moving camera. *Proceedings of European Conference in Computer Vision (ECCV)*, 2012.

- [75] Aldo Maalouf, Philippe Carré, Bertrand Augereau, and Christine Fernandez Maloigne. Banded-Based Video Inpainting. *European Conference on Signal Processing*, 2008.
- [76] Ali Mosleh, Nizar Bouguila, and A. Ben Hamza. Banded-based sparsity regularization in video inpainting. *Journal of Visual Communication and Image Representation*, 2014.
- [77] Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 2006.
- [78] D.L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 2006.
- [79] Aamir Saeed Malik, Tae Sun Choi, and Humaira Nisar. *Depth Map and 3D Imaging Applications: Algorithms and Technologies*. IGI Global, 2011.
- [80] 3d world glossary. *Computer Arts*, 2011.
- [81] Christoph Fehn. A 3d-tv approach using depth-image-based rendering. *Visualization, Imaging, and Image Processing*, 2003.
- [82] Om Prakash Gangwal and Robert Paul Berretty. Depth map post-processing for 3d-tv. *IEEE International Conference on Consumer Electronics*, 2009.
- [83] R. Krishnamurthy, B.B. Chai, H. Tao, and S. Sethuraman. Compression and transmission of depth maps for image-based rendering. *IEEE International Conference on Image Processing*, 2001.
- [84] D. Tzovaras, N. Grammalidis, and M. Strintzis. Disparity field and depth map coding for multiview image sequence. *IEEE International Conference on Image Processing*, 1996.
- [85] B.B. Chai, S. Sethuraman, and H. S. Sawhney. A depth map representation for real-time transmission and view-based rendering of a dynamic 3d scene. *First International Symposium on 3D Data Processing Visualization and Transmission*, 2002.
- [86] Yannick Morvan, Dirk Farin, and et al. Platelet-based coding of depth maps for the transmission of multiview images. *Proceedings of SPIE, Stereoscopic Displays and Applications*, 2006.
- [87] G.C.V. Perera, V. De Silva, AM. Kondo, and S. Dogan. An improved model of binocular energy calculation for full-reference stereoscopic image quality assessment. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.
- [88] Seungchul Ryu and Kwanghoon Sohn. No-reference quality assessment for stereoscopic images based on binocular quality perception. *Circuits and Systems for Video Technology, IEEE Transactions on*, 2014.
- [89] Y. M. Lu and M. N. Do. Multidimensional directional filter banks and surfacelets. *IEEE Transactions on Image Processing*, 2007.

- [90] V. Chandrasekaran, M. B. Wakin, D. Baron, and R. Baraniuk. Compressing piecewise smooth multidimensional functions using surflats: Rate-distortion analysis. Technical report, Rice University, Department of Electrical and Computer Engineering, 2004.
- [91] J. Smith, G. Petrova, and S. Schaefer. Progressive encoding and compression of surfaces generated from point cloud data. *Computer Graphics*, 2012.
- [92] Christian Bähnišch, Peer Stelldinger, and Ullrich Köthe. Fast and accurate 3d edge detection for surface reconstruction. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009.
- [93] M. Kolomenkin, I. Shimshoni, and A. Tal. On edge detection on surfaces. *IEEE Conference on Computer Vision and Pattern Recognition CVPR 2009*, 2009.
- [94] Hong Han, Xing Wu, Licheng Jiao, and Lichuan Yue. Research of method in human detection based on bandelet transform. *Proceedings SPIE*, 2009.
- [95] R. Sudhakar, N. Vignesh, and S. Jayaraman. Application of wavelet footprints for fingerprint compression. *ICGST International Journal on Graphics, Vision and Image Processing GVIP*, 2005.
- [96] Mariusz Rachubiński. Iris identification using geometrical wavelets. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009.
- [97] Jun Zhang, Lin Lian, and Guohui Li. Two-layered image compression using 2G bandelets and wavelets. *Journal of Electronics (China)*, 2008.
- [98] Chengfu Huo, Rong Zhang, and Dong Yin. An adaptive sparse representation for remote sensing image based on combination of wavelet and adaptive directional filter. *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010.
- [99] A. Maalouf, P. Carre, B. Augereau, and C. Fernandez-Maloigne. A bandelet-based inpainting technique for clouds removal from remotely sensed images. *IEEE Transactions on Geoscience and Remote Sensing*, 2009.
- [100] S. Yang, W. Qi, Z. Wang, and L. Jiao. Sar image compression based on multibandelets and geometric flow optimization. *Proceedings of SPIE*, 2009.
- [101] S. Yang, Z. Wang, W. Qi, and L. Jiao. Low-complexity bandelet for sar image compression. *Proceedings of SPIE*, 2009.
- [102] Hanqiang Liu, Biao Hou, Shuang Wang, and Licheng Jiao. Sar image compression using bandelets and spilt. *International Conference on Radar (CIE)*, 2006.
- [103] Ruchan Dong, Biao Hou, Shuang Wang, and Licheng Jiao. Sar image compression based on wedgelet-wavelet. Multimedia Systems and Applications Series. Springer US, 2008.

- [104] I Shanthi and ML Valarmathi. Comparative study of transform domain filters with modified particle swarm optimization for speckle noise suppression of sar images. *International Journal of Tomography & Simulation*, 2014.
- [105] Sujitha Juliet, Elijah Blessing Rajsingh, and Kirubakaran Ezra. A novel image compression method for medical images using geometrical regularity of image structure. *Signal, Image and Video Processing*, 2014.
- [106] M. Mansour, H. Mouhadjer, A. Alipacha, and K. Draoui. Comparative analysis on image compression techniques for chromosome images. *2nd International Conference on Advances in Biomedical Engineering*, 2013.
- [107] R. Rajeswari and R. Rajesh. Efficient compression of 4d fmri images using bandelet transform and fuzzy thresholding. *World Congress on Nature & Biologically Inspired Computing NaBIC*, 2009.
- [108] NguyenThanh Binh, VoThiHong Tuyet, and PhanCong Vinh. Ultrasound images denoising based context awareness in bandelet domain. volume 128 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Springer International Publishing, 2014.
- [109] Beladgham Mohammed, Habchi Yassine, Moulay Lakhdar Abdelmouneim, Abdesselam Basou, and Taleb Ahmed Abdelmalik. The Performance of Discret Bandelet Transform Coupled by SPIHT Coder to Improve the Visual Quality of Biomedical Color Image Compression. *International Journal of Image, Graphics and Signal Processing*, 2014.
- [110] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi. The jpeg 2000 still image compression standard. *IEEE Signal processing Magazine*, 2001.
- [111] C. Coelho, N. Roma, and L. Sousa. Pipeline architectures for computing 2-d image moments. *Proceedings of DCIS'99*, 1999.
- [112] Jaime-Andres Arteaga and Jaime Velasco-Medina. Design of image codec based on bandelet transform using a nios ii processor. *Ingeniare. Revista chilena de ingeniería*, 2012.