



UNIVERSIDADE ESTADUAL PAULISTA  
“JÚLIO DE MESQUITA FILHO”  
Campus de Ilha Solteira

**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

“Sistema de Armazenamento de Imagens Comprimidas  
Através da Transformada Wavelet”

***JAQUELINE FERREIRA DA SILVA***

**Orientador:** Prof. Dr. Francisco Villarreal Alvarado

Dissertação apresentada à Faculdade de  
Engenharia - UNESP – Campus de Ilha  
Solteira, para obtenção do título de  
Mestre em Engenharia Elétrica. Área de  
Conhecimento: Automação.

Ilha Solteira – SP

junho/2008

## FICHA CATALOGRÁFICA

Elaborada pela Seção Técnica de Aquisição e Tratamento da Informação  
Serviço Técnico de Biblioteca e Documentação da UNESP - Ilha Solteira.

S586s Silva, Jaqueline Ferreira da.  
Sistema de armazenamento de imagens comprimidas através da transformada wavelet / Jaqueline Ferreira da Silva. -- Ilha Solteira : [s.n.], 2008  
99 f. : il. (algumas color.)

Dissertação (mestrado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Área de conhecimento: Automação, 2008

Orientador: Francisco Villarreal Alvarado  
Bibliografia: p. 91-94

1. Processamento de imagens – Técnicas digitais. 2. Wavelets (Matemática).  
3. Compressão de imagens.



**UNIVERSIDADE ESTADUAL PAULISTA**  
CAMPUS DE ILHA SOLTEIRA  
FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA

### CERTIFICADO DE APROVAÇÃO

**TÍTULO:** Sistema de Armazenamento de Imagens Comprimidas Através da Transformada Wavelet

**AUTORA:** JAQUELINE FERREIRA DA SILVA

**ORIENTADOR:** Prof. Dr. FRANCISCO VILLARREAL ALVARADO

Aprovada como parte das exigências para obtenção do Título de MESTRE em ENGENHARIA ELÉTRICA pela Comissão Examinadora:

Prof. Dr. FRANCISCO VILLARREAL ALVARADO  
Departamento de Matemática / Faculdade de Engenharia de Ilha Solteira

Prof. Dr. JOZUE VIEIRA FILHO  
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira

Prof. Dr. PEDRO HENRIQUE COX  
Departamento de Engenharia Elétrica / Universidade Federal de Mato Grosso do Sul

Data da realização: 25 de junho de 2008.



Presidente da Comissão Examinadora  
Prof. Dr. FRANCISCO VILLARREAL ALVARADO

# Dedicatória

À minha família e amigos.

À Silvia Leonoura Bezerra (In Memoriam).

# Agradecimentos

À Deus por tudo que ele já me proporcionou.

Ao meu orientador, Dr. Francisco Villarreal Alvarado pela oportunidade e confiança.

Ao professor Dr. Marco A. Queiroz Duarte pelo incentivo, confiança, acompanhamento e dedicação no desenvolvimento do trabalho.

A minha mãe que nunca mediu esforços para me ajudar e que sempre apoiou as minhas escolhas.

As minhas amigas que me acompanham há mais tempo: Tatiana e Selma, pela motivação que sempre recebi delas e ao meu primo Alex, que de forma direta ou indireta contribuiu para o término desse trabalho.

Aos amigos que fiz durante o mestrado.

“A alegria de ver e entender é o mais perfeito dom da natureza.”

(Albert Einstein).

## Resumo

Neste trabalho é apresentado um sistema interativo para o processamento e armazenamento de imagens no domínio wavelet e uma interface para processamento de imagens digitais. A proposta apresentada baseia-se na transformada wavelet e em métodos de limiar. O sistema de armazenamento tem como objetivo a otimização do espaço computacional, tanto para armazenamento como para transmissão de imagens. Sendo para isso necessário a aplicação da Transformada Wavelet nas respectivas imagens e dos diversos métodos de limiar a serem escolhidos. Estas aplicações permitem extrair informações relevantes para o armazenamento de uma imagem com um menor custo computacional e com uma margem de erro muito pequena quando se compara as imagens, original e processada, ou seja, não há perda de qualidade ao aplicar o sistema de codificação apresentado. Os resultados obtidos a partir das informações extraídas das imagens são apresentados numa interface gráfica. É através da interface gráfica que o usuário usa os arquivos para visualizar e analisar os resultados dos programas diretamente na tela do computador sem a preocupação de lidar com os códigos fontes. A interface gráfica, os programas de processamento de imagens via Transformada Wavelet e os métodos de limiar foram desenvolvidos no ambiente do *MATLAB*, possibilitando uma troca direta de informações entre eles e o usuário.

**Palavras-chave:** Processamento de imagens, Transformada Wavelet, Compressão de imagens.

## **Abstract**

In this work an interactive system processing and storage of compressed images in the wavelet domain, and an interface for digital image processing are presented. The storage method proposed is based on the wavelet transform and a threshold method. It has the objective of optimizing the computational space, for storage and for transmission of images. It is necessary first a pre-processing routine and after, the characterization through the application of the wavelet transform to the respective images and the several threshold methods to be chosen. These applications allow the extraction of relevant information for an image compression or storage image with a smaller computational cost and with a very small of error when the original and processed images are compared, in other words, there is no quality loss when applying the codification system presented. The results obtained from the extracted information of the images are presented in a graphic interface for image processing. The graphic interface, the programs for image processing using the wavelet transform and the thresholding methods were developed in a MATLAB environment, making possible a direct change of information between them and the user, without the concern of working with source codes. Through this graphic interface the user can use files which permits him visualize and analyze the results of the programs in the computer screen.

**Keywords:** Processing of images, Wavelet Transform, Compression of images.

## Lista de Ilustrações

Figura 2. 1: (a) Imagem biomédica; (b) Imagem de geoprocessamento. ....	24
Figura 2. 2: (a) Imagem digital da mão de um paciente que sofre de encondroma; (b) Ampliação da área de uma das articulações; (c) Valores de intensidade na região em (b); (d) Nível de quantificação. ....	25
Figura 2. 3: (a) Imagem de dimensão 256x256 pixels; (b) Imagem de dimensão 128x128 pixels; (c) Imagem de dimensão 64x64 pixels. ....	26
Figura 2. 4: (a) Imagem adquirida da Orquídea; (b) Dispositivo de captura; (c) Sinal digital. ....	28
Figura 2. 5: Pré-processamento de uma Imagem .....	28
Figura 2. 6: (a) Ampliação de uma região da imagem da orquídea; (b) Valores de intensidade na região selecionada. ....	29
Figura 2. 7: (a) Reconhecimento; (b) Interpretação. ....	30
Figura 2. 8: (a) Imagem original; (b) Imagem contaminada por ruído branco; (c) Imagem após a eliminação do ruído utilizando a Transformada Wavelet.....	30
Figura 3. 1: (a) Função escala de Haar; (b) Função wavelet de Haar.....	34
Figura 3. 2: Função wavelet de Morlet.....	35
Figura 3. 3: (a) Função escala de Daubechies; (b) Função wavelet de Daubechies.....	37
Figura 3. 4: (a) Função escala de Meyer; (b) Função wavelet de Meyer. ....	38
Figura 3. 5: Bandas de frequências entre $V_j$ e $V_{j+1}$ .....	44
Figura 3. 6: Decomposição padrão de uma imagem. ....	45
Figura 3. 7: Decomposição não padrão de uma imagem.....	46
Figura 3. 8: Diagrama de um banco de filtros no processo de decomposição. ....	50
Figura 3. 9: Diagrama de um banco de filtros no processo de reconstrução.....	51
Figura 3. 10: Estrutura em cascata da Transformada Wavelet.....	52

Figura 4. 1: Imagem decomposta e comprimida com 98%.	57
Figura 4. 2: Limiar Rígido.	60
Figura 4. 3: Limiar Suave.	61
Figura 4. 4: Limiar Rígido Suave.	62
Figura 4. 5: Limiar Super Suave.	63
Figura 4. 6: Limiar Super Super Suave.	64
Figura 4. 7: Limiar Rígido Super Suave.	65
Figura 4. 8: Limiar Semi-Suave.	66
Figura 4. 9: Limiar Sigmoidal.	67
Figura 4. 10: Diagrama ilustrativo do sistema de redução de ruído ou compressão por limiar no domínio wavelet.	68
Figura 5. 1: Tela 1	76
Figura 5. 2: Tela 2	77
Figura 5. 3: Tela 3.	78
Figura 5. 4: Tela 4.	79
Figura 5. 5: Tela 5	80
Figura 6. 1: Flor, original e com compressão de 86,34% utilizando o Limiar Rígido.	82
Figura 6. 2: Mão, original e com compressão de 86,85% utilizando o Limiar Rígido.	83
Figura 6. 3: Meninos, original e com compressão de 92,24% utilizando o Limiar Rígido.	83
Figura 6. 4: Ultra-sonografia, original e com compressão de 86,69% utilizando o Limiar Rígido.	83
Figura 6. 5: Usina, original e com compressão de 89,38% utilizando o Limiar Rígido.	84
Figura 6. 6: Flor, original e com compressão de 86,34% utilizando o Limiar Suave	85
Figura 6. 7: Mão, original e com compressão de 86,85% utilizando o Limiar Suave.	86
Figura 6. 8: Meninos, original e com compressão de 92,24% utilizando o Limiar Suave.	86
Figura 6. 9: Ultra-sonografia, original e com compressão de 86,69% utilizando o Limiar Suave.	86

Figura 6. 10: Usina, original e com compressão de 89,38% utilizando o Limiar Suave.....	87
Figura 6. 11: Flor, original e com compressão de 86,34% utilizando o Limiar Semi Suave. .....	88
Figura 6. 12: Mão, original e com compressão de 60,53% utilizando o Limiar Semi Suave. .....	88
Figura 6. 13: Meninos, original e com compressão de 76,73% utilizando o Limiar Semi Suave. ....	88
Figura 6. 14: Ultra-sonografia, original e com compressão de 60,05% utilizando o Limiar Semi Suave. ....	89
Figura 6. 15: Usina, original e com compressão de 68,15% utilizando o Limiar Semi Suave. .....	89

## Lista de Tabelas

Tabela 6. 1: Imagens comprimidas e decodificadas usando o sistema de armazenamento proposto com o limiar rígido .....	82
Tabela 6. 2: Imagens processadas usando o sistema de armazenamento proposto com o limiar suave .....	85
Tabela 6. 3: Imagens processadas usando o sistema de armazenamento proposto com o limiar semi-suave.....	87
Tabela 6. 4: Erro estimado para os métodos utilizados para a compressão usando o sistema de armazenamento proposto .....	90

## Lista de Abreviaturas

AMR	Análise de Multirresolução;
db4	Função Wavelet de Daubechies Nível 4
db6	Função Wavelet de Daubechies Nível 6
GUI	Development Environment;
<i>MATLAB</i>	Matrix Laboratory;
TW	Transformada Wavelet;
TWC	Transformada Wavelet Contínua;
TWCI	Transformada Wavelet Contínua Inversa;
TWD	Transformada Wavelet Discreta;
TWDB	Transformada Wavelet Discreta Bidimensional;
TWDI	Transformada Wavelet Discreta Inversa;

## Lista de Símbolos

$x, y$	Coordenadas espaciais da função bidimensional $f(x,y)$ ;
$M,N$	Linhas e colunas de uma matriz $M \times N$ ;
$n$	Número máximo de níveis e faixa
$d(x)$	Resolução horizontal;
$d(y)$	Resolução vertical;
$\log_2$	Logaritmo de um número na base 2;
$V_j$	Espaço de Escala $2^j$
$\phi(t)$	Função Escala;
$\psi(t)$	Função Wavelet;
$L^2(\mathbb{R})$	Espaço das funções mensuráveis de Lebesgue de quadrado integrável;
$\hat{\psi}(t)$	Transformada de Fourier de $\psi$
$\omega_0$	Frequência inicial
$\langle f, g \rangle$	Produto interno de $f$ e $g$ .
$*$	Conjugado complexo de $\psi$ ;
$h_n$	Coefficientes dos filtros gerados pela função $\phi$ ;
$g_n$	Coefficientes dos filtros gerados pela função $\psi$ ;
$C^k$	Classe das funções com derivadas contínuas até a ordem $k$
$(W_\psi x)$	Transformada wavelet continua de uma função $x$ com relação a $\psi$ ;
$a$	Parâmetro de escala da função wavelet
$b$	Parâmetro de translação da função wavelet;

$\mathfrak{R}$	Conjunto dos números reais;
$\mathbb{Z}$	Conjunto dos números inteiros;
$W_j$	Complemento ortogonal de $V_j$
$\oplus$	Soma direta de conjuntos;
$\alpha_j$	Seqüência de cortes do filtro passa-baixa;
$H(t)$	Filtro passa-baixa;
$G(t)$	Filtro passa-alta;
$c_j$	Coefficientes de aproximação do nível $j$ da TWD;
$d_j$	Coefficientes de detalhe do nível $j$ da TWD;
*	Operação de convolução;
$(\uparrow 2)$	Operador de dizimação de ordem 2;
$(\downarrow 2)$	Operador de inserção de ordem 2;
$\lambda$	Limiar usado para a filtragem dos sinais;
$\hat{y}$	Sinal filtrado pelo limiar $\lambda$ ;
$\gamma$	Parâmetro de inclinação da Sigmóide;
$\sigma$	Estimativa do ruído;
$V_{cod}$	Vetor codificado;
$C$	Taxa de compressão;
$C_{el}$	Número de coeficientes eliminados;
$T$	Número total de coeficientes;
$traço(\bullet)$	Traço da matriz $\bullet$ .

# Sumário

<b>CAPÍTULO 1.....</b>	<b>19</b>
<b>INTRODUÇÃO .....</b>	<b>19</b>
1.1    Considerações Preliminares .....	19
1.2    Objetivos e Contribuições .....	21
1.3    Organização do Texto .....	22
<b>CAPÍTULO 2.....</b>	<b>23</b>
<b>PROCESSAMENTO DE IMAGENS.....</b>	<b>23</b>
2.1    Imagem Digital.....	24
2.2    Armazenamento de Imagem Digital.....	26
2.3    Etapas do Processamento de Imagens .....	27
<b>CAPÍTULO 3.....</b>	<b>32</b>
<b>TEORIA WAVELET.....</b>	<b>32</b>
3.1    Wavelets .....	33
3.2    Transformada Wavelet (TW) .....	38
3.2.1    Transformada Wavelet Contínua (TWC) .....	39
3.2.1.1    Função Wavelet Mãe.....	39
3.2.2    Transformada Wavelet Contínua Inversa (TWCI).....	39
3.2.3    Transformada Wavelet Discreta (TWD) .....	40
3.2.4    Transformada Wavelet Discreta Inversa (TWDI) .....	41
3.3    Análise de Multirresolução (AMR).....	42
3.4    Transformada Wavelet Discreta Bidimensional (TWD Bidimensional).....	44
3.5    Transformada Wavelet: Uma interpretação do ponto de vista de processamento de imagens .....	46
3.5.1    Filtro passa-baixa .....	47

3.5.2	Filtro passa-alta .....	48
3.5.3	Banco de Filtros .....	49
3.5.4	Algoritmo Piramidal de Mallat .....	49
<b>CAPÍTULO 4.....</b>		<b>53</b>
<b>WAVELET EM PROCESSAMENTO DE IMAGENS .....</b>		<b>53</b>
4.1	Compressão de Imagens.....	55
4.2	Métodos de Compressão e Eliminação de Ruído.....	57
4.3	Métodos de Limiar .....	59
4.3.1	Limiar Rígido (Hard- Thresholding).....	59
4.3.2	Limiar Suave (Soft - Thresholding) .....	60
4.3.3	Limiar Rígido Suave (Hard Soft - Thresholding) .....	61
4.3.4	Limiar Super Suave (Super Soft - Thresholding).....	62
4.3.5	Limiar Super Super Suave (Super Super Soft- Thresholding).....	63
4.3.6	Limiar Rígido Super Suave (Hard Super Soft- Thresholding).....	64
4.3.7	Limiar Semi-Suave (Semi- Soft - Thresholding) .....	65
4.3.8	Limiar Sigmoidal (Sigmoid - Thresholding).....	66
4.4	Atuação de um Método de Limiar numa Imagem.....	68
<b>CAPÍTULO 5.....</b>		<b>69</b>
<b>SISTEMA DE ARMAZENAMENTO DE IMAGENS E UMA INTERFACE GRÁFICA PARA O PROCESSAMENTO .....</b>		<b>69</b>
5.1	Sistema de Armazenamento de Imagens através da Transformada Wavelet .....	70
5.1.1	Codificação da Imagem Comprimida.....	71
5.1.2	Decodificação da Imagem Comprimida.....	73
5.2	Interfaces Gráficas para o Sistema de Armazenamento de Imagens Comprimidas Proposto.....	74
5.2.1	Apresentação das Telas .....	76
<b>CAPÍTULO 6.....</b>		<b>81</b>
<b>RESULTADOS E CONSIDERAÇÕES .....</b>		<b>81</b>

6.1	Utilização do Limiar Rígido no Sistema Armazenamento de Imagens Comprimidas..	81
6.2	Utilização dos Limiares Suave e Semi-Suave no Sistema de Armazenamento de Imagens Comprimidas.....	84
6.2.1	Limiar Suave .....	85
6.2.2	Limiar Semi Suave.....	87
6.3	Considerações Sobre os Métodos Utilizados na Compressão .....	89
6.4	Considerações Finais e Sugestões de Trabalhos Futuros .....	91
<b>REFERÊNCIAS .....</b>		<b>92</b>
<b>APÊNDICE A .....</b>		<b>96</b>
<b>INTERFACES GRÁFICAS NO MATLAB.....</b>		<b>96</b>

# CAPÍTULO 1

## INTRODUÇÃO

### 1.1 Considerações Preliminares

Diversos elementos que analisamos em nossa interação cotidiana, com o universo físico, podem ser representados por funções. Na literatura de processamento de sinais, uma função é chamada de sinal (GOMES; VELHO; GOLDSTEIN, 1997). O fato dos sinais, em geral, serem fontes de informações, as imagens essencialmente representadas em forma digital, são tipos de sinais cada vez mais presentes em nosso cotidiano, e a cada dia tem despertado a necessidade de processamentos indispensáveis, desde as imagens para o mais simples entretenimento até as aplicações mais importantes, por exemplo, imagens biomédicas (PORTILLA; SIMONCELLI, 2000).

O termo “sinal” está associado a um fenômeno, que pode ocorrer no tempo contínuo, por exemplo, a fala, ou no tempo discreto, neste caso o tempo é uma variável discreta que, normalmente assume valores periódicos (DINIZ; SILVA; NETTO, 2004).

É possível, através do processamento de imagens, que informações sejam disponibilizadas para uma determinada aplicação, como por exemplo, na medicina que exige diagnósticos precisos, com a análise de bordas e análises estatísticas com qualidade das informações disponíveis (ALBUQUERQUE; CANER; MELO, 2004).

Os diversos tipos de sinais gerados e os diversos métodos de obtê-los, muitas vezes acarretam perda de qualidade, pois a maioria dos sinais é contaminada por algum tipo de ruído. Por exemplo, ao adquirir uma imagem, o meio em que ela se encontra ou o dispositivo que a captura, pode fornecer uma imagem de baixa qualidade, com aspectos que comprometem uma boa análise por parte do observador.

O processamento de sinais visa, de modo geral, melhorar a qualidade de um sinal para possíveis análises.

Uma das aplicações do processamento de sinais é a compressão, com o objetivo de uma transmissão ou armazenamentos eficientes, com baixo custo computacional. A compressão geralmente se baseia na eliminação de redundâncias do sinal através do uso de alguma transformada, geralmente transformadas integrais (STOLLNITS; DEROSE; SALESIN, 1996).

A Transformada Wavelet tem sido aplicada nas mais diversas áreas, disponibilizando algoritmos rápidos, fundamentais para compressão de imagens, ou para a remoção de ruído (OLIVEIRA, 2007). Mesmo sendo uma técnica nova, se comparada com outras técnicas de processamento de sinais, a transformada wavelet tem atraído a atenção de muitos matemáticos, engenheiros e cientistas em geral, pois se trata de uma ferramenta muito versátil e de conteúdo matemático muito rico. A ferramenta mais importante na Análise Wavelet é a Transformada Wavelet (DAUBECHIES, 1992). Quando um sinal é analisado com a Transformada Wavelet, obtêm-se informações tanto no domínio do tempo quanto no domínio da frequência, ou seja, é uma ferramenta que possibilita saber exatamente quando começa e quando termina um determinado evento (DAUBECHIES, 1990) esse é o principal objetivo da transformada wavelet.

Devido ao grande número de funções moduladoras, ou funções bases, chamadas de *wavelets*, é possível o processamento dos mais variados tipos de sinais. Há sempre uma função wavelet que se ajusta ao sinal que se deseja processar. Caso tal wavelet não exista, é possível, no contexto da teoria wavelet, construí-la.

As *wavelets* são ondas pequenas, com determinadas propriedades, que as tornam adequadas para decomposição de uma determinada função em outras funções (MISITI et al., 1996). As funções do domínio wavelet podem ser aplicadas de forma específica ou se ajustarem conforme a necessidade do processamento, o que torna a transformada wavelet uma ferramenta versátil.

Uma propriedade matemática que faz com que a Transformada Wavelet seja uma ferramenta eficiente no processamento de imagens é a análise de multirresolução que, ao ser aplicada em um sinal, evidencia seus detalhes, tornando possível a análise dos coeficientes wavelets (DAUBECHIES, 1992).

Dentre as aplicações envolvendo imagens, está a compressão, pois com o advento da internet e com a necessidade da criação diária de banco de dados, imagens são transmitidas a todo o momento, podendo causar lentidão no sistema o qual elas se encontram (SILVA et al., 2007).

Em geral, a compressão de uma imagem tem como objetivo principal a eliminação de coeficientes dessa imagem que não comprometam a sua reconstrução ou que causem mínima diferença entre a imagem original e a imagem processada (FONSECA, 2004). Esta eliminação não pode ser feita diretamente na imagem, para isso é aplicado algum tipo de transformada para que o sinal seja analisado no domínio transformado, evidenciados seus detalhes assim como suas redundâncias.

Com esse objetivo, vários métodos de compressão ou redução de ruído, tanto para imagens quanto para sinais unidimensionais, são propostos na literatura especializada (SOARES et al., 2007). Após a compressão, são armazenados apenas os coeficientes que não foram eliminados na imagem transformada e as informações que possibilitem a reconstrução da imagem através desses coeficientes.

Outra questão tão importante quanto às técnicas que envolvem o processamento de imagens é a criação de dispositivos que permitam o acesso rápido e fácil a qualquer usuário de computadores, de forma que o mesmo usuário, independentemente de seu nível de conhecimento computacional, possa manipular imagens e processá-las. Uma forma de resolver esse problema é a criação de interfaces que permitam ao usuário o acesso ao processamento de imagens sem se preocupar com a complexidade computacional que está por trás do processo.

## **1.2 Objetivos e Contribuições**

Com base nos dois últimos parágrafos, este trabalho apresenta uma metodologia simples e interativa de armazenamento de imagens que possibilita uma economia considerável no espaço computacional que essas imagens ocupam, podendo ser implementada em qualquer linguagem computacional. Além da metodologia de armazenamento de imagens, uma interface gráfica é criada para o processamento de imagens, possibilitando ao usuário selecionar imagens, fazer escolha do tipo de compressão a ser executado, fazer comparações entre a imagem original e a processada e fazer análises estatísticas do processo.

O sistema de armazenamento consiste em dois processos: um para codificação da imagem, sendo esta comprimida pela transformada wavelet usando um dos diversos métodos de limiar propostos e outro para decodificação da imagem para posterior reconstrução da imagem original. Para isso, um único vetor é criado para armazenar todas as informações da imagem comprimida. Em geral esse vetor ocupa menos de 50% do espaço que a imagem original ocupa, sendo este um dado importante para contribuição na área de processamento digital de imagens.

A interface gráfica tem como principal objetivo o acesso ao processamento de imagens de usuários que não tem domínio de linguagens de programação, mas que necessitam manusear imagens. Caso contrário, também é possível apenas lidar com os códigos fontes para eventuais modificações e melhoramento no processamento.

### **1.3 Organização do Texto**

O texto é organizado inicialmente com um capítulo sobre os fundamentos básicos do processamento de imagens, este é o assunto do capítulo 2. No capítulo 3, é apresentada a Teoria Wavelet com seus principais fundamentos matemáticos. No capítulo 4, a aplicação de wavelets ao processamento de imagens e os métodos de limiar para a compressão e redução de ruído em imagens são apresentados. No capítulo 5 é apresentada a proposta de armazenamento de imagens, e a interface gráfica criada para o processamento de imagens, ambos implementados no software *MATLAB*. No capítulo 6 são apresentados os resultados obtidos aplicando a técnica de armazenamento proposta no capítulo 5 e as considerações finais juntamente com as conclusões e sugestões para trabalhos futuros.

## **CAPÍTULO 2**

### **PROCESSAMENTO DE IMAGENS**

O processamento digital de imagens é uma subárea do processamento de sinais que consiste na execução de operações matemáticas, com objetivo de extrair informações que geralmente representam um fenômeno a ser estudado de forma específica. No processamento de uma imagem são realizados milhares de cálculos de forma rápida e segura para que o observador ou o computador possam analisar ou realçar de forma precisa estas informações.

A primeira aplicação da área de processamento de imagens foi na década de 20, na tentativa de aprimorar imagens digitalizadas de um jornal para transmissão, entre Londres e Nova Iorque. O tempo necessário para esta transmissão era de uma semana. O sistema Bartlane de transmissão de imagens por cabo submarino conseguiu reduzir a transmissão para três horas (BOTELHO, 2005).

Avanços expressivos na área vieram apenas com o advento dos computadores digitais três décadas mais tarde, como a evolução da tecnologia computacional, com informações de multimídia e sinais bidimensionais, cada vez mais presentes em nosso cotidiano.

O processamento de imagens surge com o objetivo de suprir as necessidades de análise e compactação de informações, tornando-se assim uma ferramenta essencial no mundo moderno, cuja demanda de processamento é relativamente crescente (GONZÁLES; WOODS, 1992).

Um dos principais interesses na área de processamento digital de imagens é a necessidade de melhorar a qualidade da informação visual para a análise humana, surgindo assim várias técnicas e aperfeiçoamento para o ganho de velocidade, precisão, simplicidade e poder de processamento com um baixo custo computacional. São técnicas hoje aplicadas

em várias áreas do conhecimento humano como, por exemplo, medicina, economia, engenharia, entre outras.

A Figura 2.1 apresenta dois exemplos de imagens muito usadas em estudos, uma imagem biomédica e uma imagem de geoprocessamento, ambas geradas no software do *MATLAB*.

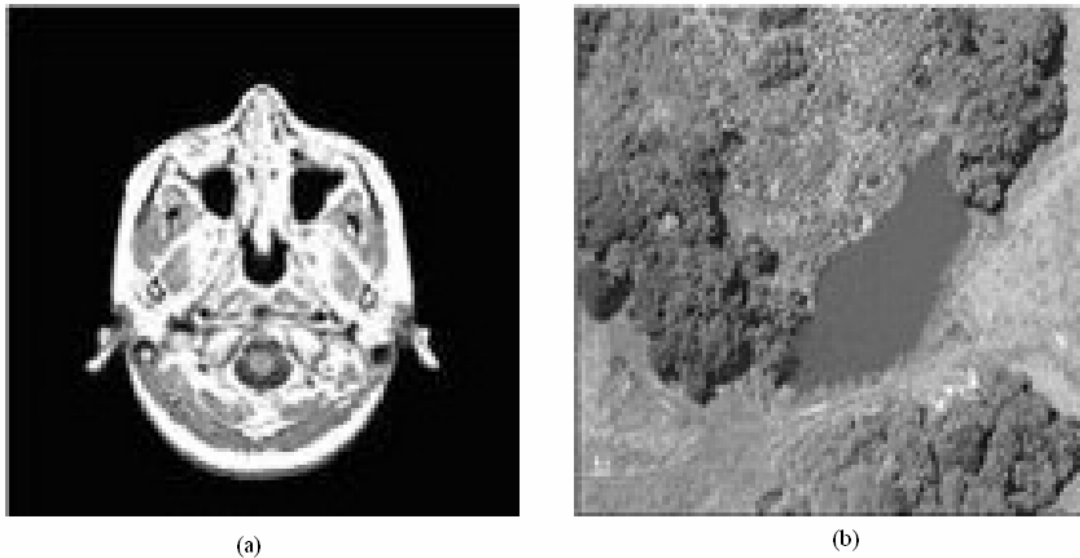


Figura 2. 1: (a) Imagem biomédica; (b) Imagem de geoprocessamento.

O Processamento digital de imagens é sempre expresso através de algoritmos implementados por software. É um processo extremamente dependente do sistema a que está associado e do método de extração de informações, que por sua vez, depende do tipo de imagem, da natureza e das informações contidas nela.

Geralmente, técnicas que funcionam bem em algumas áreas podem não ser adequadas em outras. Assim, não existe até o momento uma solução única e abrangente para todos os problemas, abrindo espaço para varias pesquisas.

Algumas noções básicas serão abordadas a seguir de forma sucinta para um melhor entendimento de como analisar uma imagem digital.

## 2.1 Imagem Digital

As imagens são vistas como exemplos de sinais gerados em nosso cotidiano que apresentam papéis importantes, podendo ser desde os mais simples, usados para

entretenimentos, até as aplicações médicas e tecnológicas mais avançadas (ALBUQUERQUE; CANER; MELO, 2004).

Uma imagem pode ser definida matematicamente como uma função bidimensional do tipo  $f(x, y)$ , onde  $x$  e  $y$  são coordenadas espaciais, ou seja, é um tipo de sinal gerado no domínio do espaço, que representa a variação da intensidade de luz e de cor do espaço.

A imagem denominada digital é uma imagem que pode ser processada de forma direta, pois ela pode ser processada no domínio espacial sem a necessidade de um sistema analógico que a converta em um sinal no domínio do tempo (DINIZ; SILVA; LIMA, 2004), podendo assim ser armazenada no computador. Sua representação é dada por uma matriz  $M \times N$  composta por elementos denominados pixels.

Os pixels são os menores pontos distribuídos em linhas e colunas que uma imagem pode obter. Cada pixel contém um valor inteiro nas direções das coordenadas  $x$  e  $y$  que representa medidas dependentes de variáveis, como por exemplo, o nível de quantificação que normalmente é uma potência de 2.

Cada pixel pode estar associado a um valor da escala de cinza entre 0 e  $2^n - 1$ . Quanto maior o número de pixels numa imagem, melhor será sua resolução, permitindo uma melhor diferenciação entre as estruturas. Na Figura 2.2 tem-se um modelo da digitalização de uma imagem radiográfica.

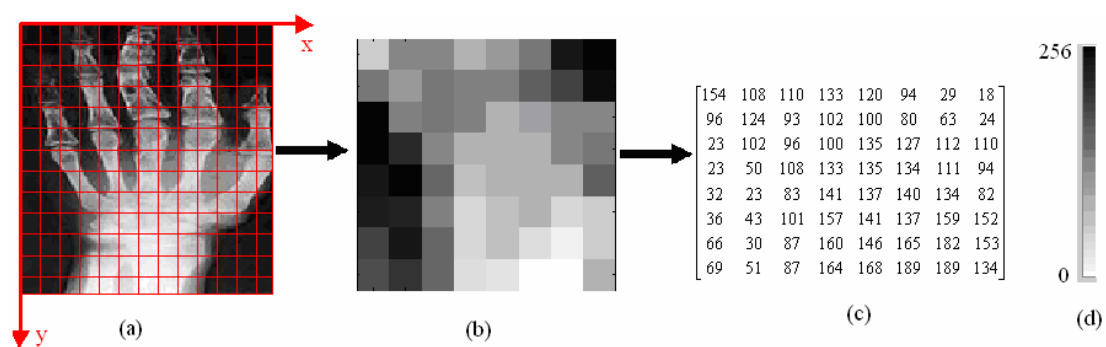


Figura 2. 2: (a) Imagem digital da mão de um paciente que sofre de encondroma; (b) Ampliação da área de uma das articulações; (c) Valores de intensidade na região em (b); (d) Nível de quantificação.

Muitas vezes a digitalização da imagem pode comprometer sua qualidade. Atualmente, existem várias técnicas de análise de imagens, visto que as imagens carregam

em seu interior determinadas informações e também capacidade para a troca dos mesmos, possibilitando, desta forma, a qualidade de resolução.

## 2.2 Armazenamento de Imagem Digital

Um aspecto de grande interesse é o armazenamento da imagem na memória do computador. Um exemplo que apresenta essa necessidade são os milhares de exames de diagnósticos através de imagens geradas em hospitais, que em alguns casos, podem chegar a mais de 45 Gbytes por dia (ALCOCER et. al., 1996).

Atualmente, crescem os processos de transmissão dessas imagens através de redes informatizadas, criando necessidades de se estabelecer e conhecer formatos padronizados e de processos de transmissão de dados em sistemas de rede local ou mesmo pela internet, com rapidez e segurança, envolvendo desta forma a compactação da imagem. Porém, ao passo que melhora a qualidade visual da imagem, o volume de dados a serem armazenados processados ou transmitidos aumenta também, o que proporciona o aumento no número de bits necessários para a codificação binária da imagem.

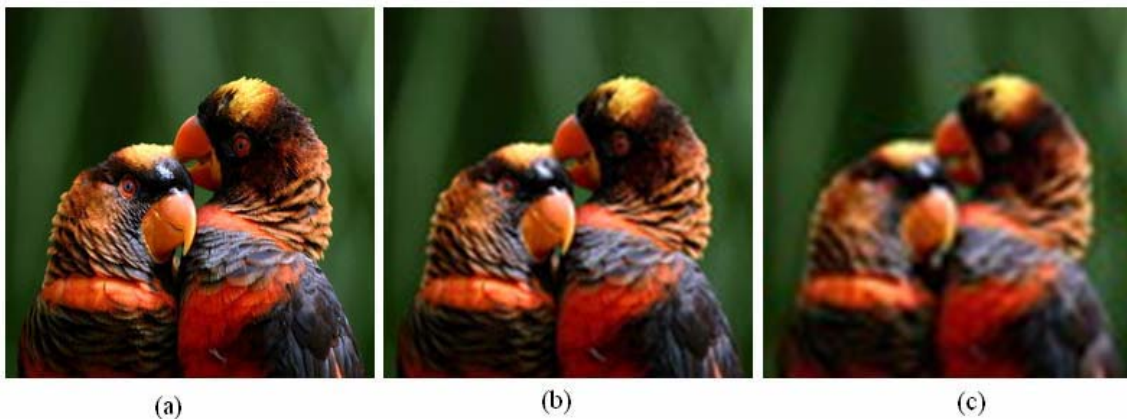


Figura 2. 3: (a) Imagem de dimensão 256x256 pixels; (b) Imagem de dimensão 128x128 pixels; (c) Imagem de dimensão 64x64 pixels.

O número de bits necessários para o armazenamento de uma imagem na memória do computador é dado pela equação (2.1) (GONZÁLES; WOODS, 2000).

$$\text{bits} = M \times N \times n \quad (2.1)$$

onde:

$$n = \log_2(M)$$

M: É o número de linhas da imagem;

N: É o número de colunas da imagem;

Existem dois tipos de compactação, com perda e sem perda de informações. Cada tipo de imagem tem suas exigências quanto ao tipo de compactação, além de ter um *cabeçalho dos arquivos de imagens digitais* que pode conter informações do tipo: número de linhas, número de colunas, número de bits usados na representação da imagem, resolução horizontal ( $dx$ ), resolução vertical ( $dy$ ), número de bandas da imagem, tipo de compactação usado para guardar os dados, data e hora de aquisição, tipo de sensor que captou a imagem, dados paramétricos dos sensores, bem como outras informações relevantes.

## 2.3 Etapas do Processamento de Imagens

Existem diversas formas de representar as informações compactadas da imagem. O fluxo dessas informações com um determinado objetivo é o que descreve as etapas do processamento de imagens (ALBUQUERQUE, 2001).

### 1) Aquisição

É a etapa em que a imagem é capturada através de dispositivos e organizada em forma bidimensional. Para que as informações contidas nas imagens sejam expressas de forma simplificada, é preciso que um sistema interaja com essas informações e as convertam em sinais elétricos.

Os sinais elétricos têm sua representação no tempo contínuo. Porém, a tecnologia computacional não processa sinal no tempo contínuo. Por exemplo, câmeras digitais são projetadas para lidar com computação seqüencial envolvendo números.

Para utilizar a tecnologia computacional é necessário saber quão rápida a informação varia e assim, amostrar a informação no tempo contínuo e convertê-la em informações no tempo discreto.

Quando se trata de imagens digitais não é preciso essa conversão, pois o seu processamento é direto, como dito anteriormente.

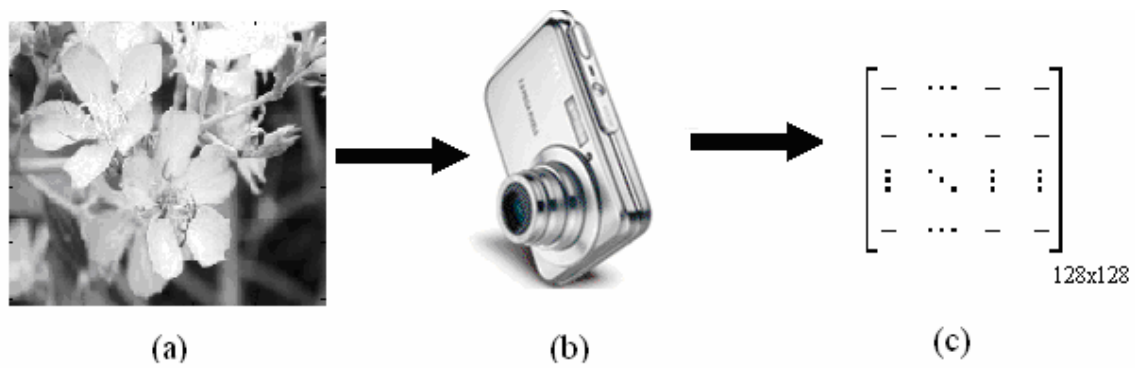


Figura 2. 4: (a) Imagem adquirida da Orquídea; (b) Dispositivo de captura; (c) Sinal digital.

## 2) Pré-processamento

Tem a função de melhorar a qualidade da imagem através de dois métodos: o primeiro baseado em filtros que manipulam o plano da imagem. O segundo que opera baseado em filtros que agem sobre o espectro da imagem.

Ambos os métodos visam o melhoramento de contraste, remoção de ruídos, regiões de interesse, reamostragem dos pixels em uma nova escala, treinamento e extração de características, etc.



Figura 2. 5: Pré-processamento de uma Imagem.

## 3) Segmentação

Significa separar a imagem em regiões disjuntas com o objetivo de extrair informações dos objetos da imagem, como ilustra a Figura 2.6.

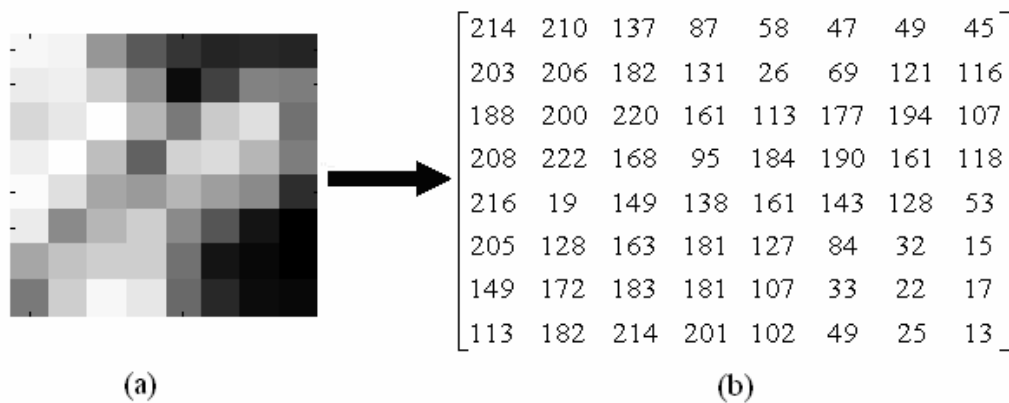


Figura 2. 6: (a) Ampliação de uma região da imagem da orquídea; (b) Valores de intensidade na região selecionada.

A denominação “objeto” da imagem refere-se aos grupos de pixels que fornecem informações desejadas.

Outro termo bastante usado nessa etapa é denominado “fundo” da imagem, que classifica o grupo dos pixels que podem ser desprezados. Esses dois termos mencionados, juntos determinam regiões na imagem sem representar necessariamente um objeto presente na imagem processada.

Essa é uma etapa crítica, pois é preciso ter cuidados para não gerar erros, que posteriormente serão refletidos nas etapas seguintes, produzindo ao final do processo resultados comprometedores.

A segmentação age de forma adaptativa às características particulares de cada tipo de imagem e aos objetivos desejados. Deste modo, pode-se dizer que não existe um modelo específico de segmentação. Porém, as técnicas são bem diversificadas e despertam interesse no melhoramento e desenvolvimento de novas técnicas.

#### 4) **Representação e descrição:**

É uma forma de armazenar as informações através de uma matriz, a qual caracteriza a forma e a topologia dos objetos, seja a imagem representada como sinal unidimensional ou bidimensional. Essa etapa é classificada pela parametrização dos objetos da imagem.

### 5) Reconhecimento e interpretação

Identifica os objetos segmentados na imagem associando um rótulo a cada objeto. Após isso, os objetos são classificados conforme sua forma apresentada, como ilustra a Figura 2.7.

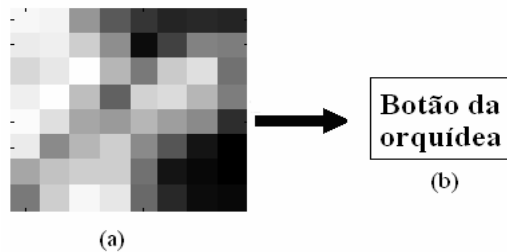


Figura 2. 7: (a) Reconhecimento; (b) Interpretação.

As informações contidas nas imagens podem ser extraídas e analisadas tanto no domínio do tempo quanto no domínio da frequência, ou ainda no domínio tempo-escala como é o caso da transformada wavelet (DAUBECHIES, 1990).

Duas transformadas muito usadas para o processamento de imagem são a Transformada de Fourier (OPPENHEIM; SCHAFER; BUCK, 1998) e a Transformada Wavelet (STOLLNITS; DEROSE; SALESIN, 1996).

Na Figura 2.8 é apresentado um exemplo de redução de ruído em imagens através da transformada wavelet (DUARTE, 2005).

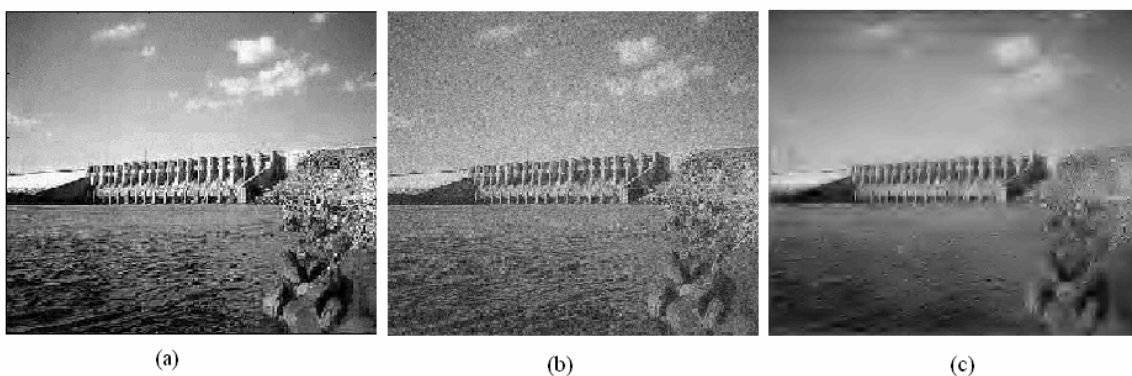


Figura 2. 8: (a) Imagem original; (b) Imagem contaminada por ruído branco; (c) Imagem após a eliminação do ruído utilizando a Transformada Wavelet.

No campo do processamento de imagens, o presente trabalho apresenta duas propostas: uma é a criação de um sistema interativo para o armazenamento de imagens, que otimiza o espaço computacional que essas imagens ocupam, tendo a diminuição do número de bits usados para armazenamento. O processo de codificação será feito usando a Transformada Wavelet no capítulo 5. A outra proposta é a criação de uma interface gráfica para processamento de imagens no que diz respeito à compressão e redução de ruído. Esta interface é criada com o objetivo de facilitar o trabalho de quem precisa manipular imagens, mas não tem conhecimento de programação computacional.

## CAPÍTULO 3

### TEORIA WAVELET

Uma função wavelet é a interpretação de uma onda de curta duração com crescimento e decrescimento rápidos. Sua teoria baseia-se na representação de funções em diferentes escalas e diferentes resoluções (tempo-escala), considerando assim uma das suas principais características (DAUBECHIES, 1992).

O primeiro registro do termo “wavelet” data de 1909, em uma tese de Alfred Haar (HAAR, 1910), apresentando uma função que, décadas depois, viria a ser conhecida como a primeira função wavelet.

O conceito de wavelet, em sua forma teórica atual, foi proposto em meados dos anos oitenta por Jean Morlet (geofísico), Yves Meyer (matemático) e a equipe do Centro de Física Teórica de Marseille, trabalhando sob orientação de Alex Grossman (físico teórico) na França. Os métodos de análise wavelet foram desenvolvidos principalmente por Yves Meyer (MEYER, 1993) e seus colegas, que asseguraram a sua disseminação.

A atenção da comunidade de processamento de sinais foi atraída quando Ingrid Daubechies (DAUBECHIES, 1990; DAUBECHIES, 1992; DAUBECHIES, 1998;) e Stephane Mallat (MALLAT, 1989 (a), MALLAT, 1989 (b)), além de suas contribuições para a teoria de wavelets, estabeleceram a conexão entre os dois assuntos e obtiveram resultados via processamento de sinal discreto.

O algoritmo de Mallat (MALLAT, 1989 (a)) é considerado o elo definitivo entre wavelets e processamento de sinais. Desde então, pesquisas em wavelets tornaram-se difundidas internacionalmente (COIFMAN et al. 1993, COIFMAN, 1990, COIFMAN; WICKERHAUSER, 1993, RIOUL, VETTERLI, 1991).

### 3.1 Wavelets

As funções wavelets, geralmente denotadas por  $\psi(t)$ , são definidas como um conjunto de funções originadas através das operações matemáticas de translação e escalonamento da função escala, com propriedades particulares que as tornam adequadas para servirem de base para a decomposição de outras funções (FARIA, 1997).

A *função escala* é uma função básica definida num espaço  $V_j$ , usualmente denotada por  $\phi$ , tendo como funções básicas associadas:

$$\phi_i^j(t) := \phi(2^j t - i), \quad i = 0, \dots, 2^{j-1} \quad (3.1)$$

sendo:

$\phi$ : função escala;

$i$ : deslocamento;

$j$ : escala;

$t$ : tempo.

Para que uma função seja considerada uma wavelet é preciso satisfazer as seguintes condições básicas e necessárias:

1. que  $\psi(t) \in L^2(\mathbb{R})$ , ou seja, a função pertença ao espaço das funções de quadrado integrável ou, ainda, o espaço das funções de energia finita, no sentido que:

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty. \quad (3.2)$$

2. que sua Transformada de Fourier  $\hat{\psi}(\omega)$  satisfaça a condição de admissibilidade (DAUBECHIES, 1992):

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty. \quad (3.3)$$

Segue da condição de admissibilidade que:

$$\lim_{\omega \rightarrow 0} \hat{\psi}(\omega) = 0. \quad (3.4)$$

Assim, se  $\hat{\psi}(\omega)$  é contínua então,  $\hat{\psi}(0)=0$ , ou seja,

$$\int_{-\infty}^{\infty} \psi(t) dt = 0. \tag{3.5}$$

Geometricamente, a condição (3.3) estabelece que o gráfico de  $\psi(t)$  deve oscilar de modo a cancelar as áreas negativas a fim de anular a integral (3.5). Portanto, o gráfico de  $\psi(t)$  tem a forma de onda, conforme ilustra a Figura 3.1 (b), que é um exemplo de wavelet.

Desde que  $\psi(t)$  esteja bem localizada no tempo, este decaimento será muito rápido, formando uma onda de curta duração.

Atualmente, existem inúmeras funções wavelets que geralmente recebem o nome de seus criadores, dentre as quais serão apresentadas, a seguir, as mais conhecidas.

Começando pelo exemplo mais simples, proposto em 1909 pelo matemático húngaro Alfred Haar (HAAR, 1910). A *wavelet de Haar*, que demonstra as grandezas que envolvem os valores de forma não contínua, tornando-se deste modo um caso particular da transformada wavelet discreta definida por:

$$\psi_i^j(t) = \phi(2t) - \phi(2t - 1). \tag{3.6}$$

É através da equação (3.6) que podemos obter  $\psi(t)$ , apresentada na Figura 3.1 (b).

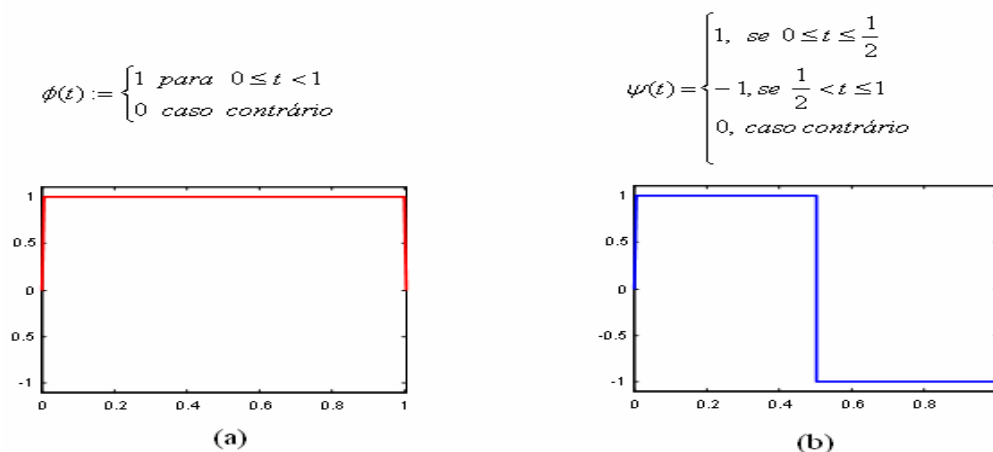


Figura 3. 1: (a) Função escala de Haar; (b) Função wavelet de Haar.

Outra função é a *wavelet de Morlet*, introduzida por Jean Morlet pertence a família das wavelets não-ortogonais. A wavelet de Morlet não possui função escala e é explícita

por uma Gaussiana modulada (*shifted*), levemente ajustada. De forma que  $\psi(0) = 0$ , conforme a equação (3.7), cujo gráfico é apresentado na Figura 3.2:

$$\psi(t) = Ce^{-t^2/2} \cos(5t) \quad (\text{MISITI et al., 1996}). \quad (3.7)$$

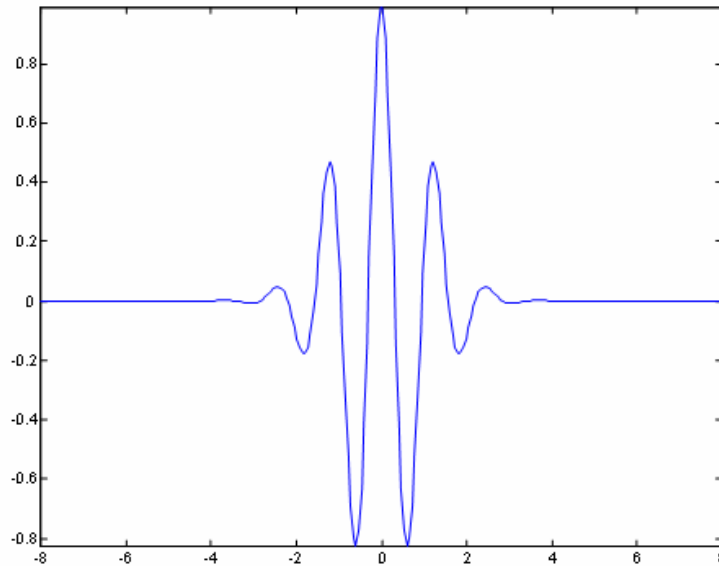


Figura 3. 2: Função wavelet de Morlet.

*Daubechies* propôs um procedimento de partida para a construção das bases ortonormais ao invés de construir a wavelet e a função de escala através de um subespaço  $V_j$ .

O procedimento parte de coeficientes apropriados e então investiga se eles correspondem a uma base de wavelet ortonormal. Esses coeficientes representam um conjunto particular de números gerados por filtros. Em 1987 as bases ortonormais de wavelets foram consideradas como sendo funções de suporte compacto contidas no intervalo  $[0, 2r + 1]$ .

Quanto maior o número de coeficientes, mais suave será a wavelet. A construção de Daubechies resulta em uma coleção de coeficientes  $h_n$ , sendo:

$$N = 2, 3, 4, \dots \quad \text{e} \quad 0 \leq n < 2N - 1$$

A seguir é apresentado um exemplo da wavelet mais simples de Daubechies, a DAUB4, gerada a partir de apenas quatro coeficientes (NIEVERGELT, 1999).



$$B = \begin{pmatrix} c_0 & c_3 & & & & & & & & c_2 & c_1 \\ c_1 & -c_2 & & & & & & & & c_1 & -c_2 \\ c_2 & c_1 & c_0 & c_3 & & & & & & & \\ c_3 & -c_0 & c_1 & -c_2 & & & & & & & \\ & & & & \ddots & & & & & & \\ & & & & & c_2 & c_1 & c_0 & c_3 & & \\ & & & & & c_3 & -c_0 & c_1 & -c_2 & & \\ & & & & & & & & & c_2 & c_1 & c_0 & c_3 \\ & & & & & & & & & c_3 & -c_0 & c_1 & -c_2 \end{pmatrix}$$

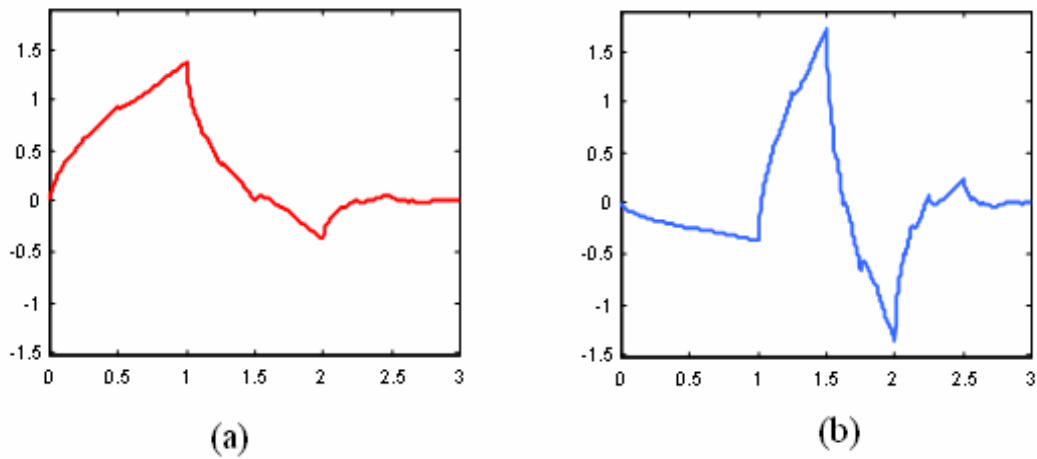


Figura 3. 3: (a) Função escala de Daubechies; (b) Função wavelet de Daubechies.

*Yves Meyer* em 1980 construiu a primeira wavelet trivial diferente da wavelet de Haar, que é continuamente diferenciável, o que limita suas aplicações (SILVA; EYNG, 2000). Desta forma, uma base wavelet suave ortonormal foi criada. Primeiro, definiu-se a Transformada de Fourier  $\hat{\phi}(t)$  de uma função escala  $\phi(t)$  como:

$$\hat{\phi}(t) = \begin{cases} (2\pi)^{-1/2}, & \text{se } |t| \leq \frac{2\pi}{3} \\ (2\pi)^{-1/2} \cos\left[\frac{\pi}{2} \nu\left(\frac{3}{4\pi}|t| - 1\right)\right], & \text{se } \frac{2\pi}{3} \leq |t| \leq \frac{4\pi}{3} \\ 0, & \text{se } |t| > \frac{4\pi}{3} \end{cases} \quad (3.12)$$

onde

$$v(a) = a^4(35 - 84a + 70a^2 - 20a^3), \quad a \in [0,1] \quad (\text{MISITI et al., 1996}). \quad (3.13)$$

Deste modo, a função wavelet  $\psi(t)$  pode ser encontrada facilmente através de  $\phi(t)$ .

$$\hat{\psi}(\omega) = \begin{cases} (2\pi)^{-1/2} e^{i\omega t/2} \sin\left[\frac{\pi}{2} v\left(\frac{3}{2\pi} |t| - 1\right)\right], & \text{se } \frac{2\pi}{3} \leq |\omega| \leq \frac{4\pi}{3} \\ (2\pi)^{-1/2} e^{i\omega t/2} \cos\left[\frac{\pi}{2} v\left(\frac{3}{4\pi} |\omega| - 1\right)\right], & \text{se } \frac{4\pi}{3} \leq |\omega| \leq \frac{8\pi}{3} \\ 0, & \text{caso contrário} \end{cases} \quad (3.14)$$

A Figura 3.4 ilustra as equações (3.12) e (3.14) respectivamente.

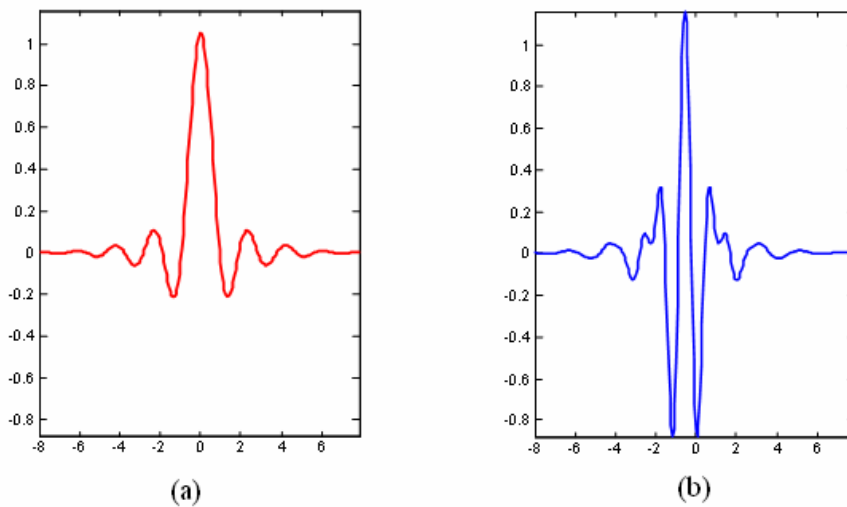


Figura 3. 4: (a) Função escala de Meyer; (b) Função wavelet de Meyer.

### 3.2 Transformada Wavelet (TW)

A Transformada Wavelet é uma ferramenta conhecida pela característica de decompor funções e de reconstruí-las, apresentando uma resolução razoavelmente boa. Por exemplo, a reconstrução de um sinal que apresente coeficientes com valores próximos de zero requer um trabalho árduo além de um alto custo computacional.

São apresentadas as duas formas da transformada, a contínua e a discreta. A transformada wavelet contínua é de grande interesse teórico, principalmente para a derivação e compreensão das propriedades matemáticas das funções wavelets. Porém a sua discretização é necessária para aplicações práticas, por exemplo, ao descrever um sinal unidimensional em uma representação bidimensional ou quando se tem a necessidade de se inverter a operação.

O problema dos parâmetros  $a$  e  $b$  variarem continuamente é resolvido com a operação de discretização, surgindo, desta maneira, a transformada wavelet discreta, com a finalidade de propor mais eficiência ao trabalho.

### 3.2.1 Transformada Wavelet Contínua (TWC)

De acordo com Young (YOUNG, 1995), a TWC pode ser considerada como uma operação de ruptura, ou seja, a transformada wavelet “quebra” uma função em muitos pedaços e estes pedaços são representados por coeficientes, chamados de coeficientes wavelet definidas na equação (3.15).

Na *decomposição* de uma função surge um conjunto de funções especiais chamadas wavelets. As wavelets são funções resultantes da atuação simultânea de duas operações (escalamento e translação) numa única função, denominada wavelet “mãe”.

#### 3.2.1.1 Função Wavelet Mãe

Matematicamente, uma função  $\psi(t)$  para ser considerada uma wavelet mãe, deve pertencer ao espaço  $L^2(\mathfrak{R})$  e satisfazer a condição de admissibilidade.

Sem muito rigor matemático, uma wavelet mãe é uma função que oscila, tem energia finita e tem valor médio nulo. As funções básicas associadas são do tipo:

$$\psi_i^j(t) = \psi(2^j t - i). \quad (3.15)$$

### 3.2.2 Transformada Wavelet Contínua Inversa (TWCI)

A TWCI segue o raciocínio de reconstruir uma função  $x(t)$  decomposta pela transformada wavelet contínua através das wavelets filhas, combinando os coeficientes wavelets. Quanto melhor a união maior será o valor do coeficiente wavelet. O conjunto de todos os coeficientes wavelet constitui a representação da função  $x(t)$  no domínio wavelet.

Assim, o sinal  $x(t)$  decomposto usando uma wavelet  $\psi(t)$  que satisfaz a condição de admissibilidade, será reconstruído através da TWCI conforme a equação (3.18).

$$x(t) = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \{(W_\psi x)(a, b)\} \left\{ \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right) \right\} \frac{dad b}{a^2} \quad (3.18)$$

Observa-se da equação (3.20) que tem o mesmo núcleo:  $\frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right)$  é utilizado na TWC e em sua inversa.

De acordo com Daubechies (DAUBECHIES, 1998), a equação (3.20) pode ser vista de dois modos diferentes:

- Modo de reconstrução de  $x(t)$ , desde que sua transformada wavelet inversa seja conhecida;
- Modo de representação de  $x(t)$ , como uma superposição de wavelets filhas.

### 3.2.3 Transformada Wavelet Discreta (TWD)

Na TWC, mais especificamente na equação (3.18), depara-se com a presença de redundâncias, pois os parâmetros  $a$  e  $b$  variam continuamente. Esse problema é contornado por meio de discretização de  $a$  e  $b$ .

Este processo origina a transformada wavelet discreta (TWD). De acordo com a literatura (GOMES; VELHO; GOLDSTEIN, 1997), uma discretização típica é do tipo:

$$a = a_0^m \quad e \quad b = n a_0^m b_0 \quad (3.19)$$

com  $m$  e  $n \in Z$ ,  $a_0 > 1$  e  $b_0 \neq 0$ .

Deste modo, tem-se a transformada wavelet discreta:

$$(W_\psi x)(m, n) = \frac{1}{\sqrt{a_0^m}} \int_{-\infty}^{+\infty} x(t) \psi\left(\frac{t - n a_0^m b_0}{a_0^m}\right) dt \quad (3.20)$$

Destas equações observa-se:

- A TWD é definida apenas para valores de escalas positivos ( $a_0 > 1$ );
- O passo da translação é proporcional à escala ( $b = n a_0^m b_0$ );

- A TWD produz um conjunto finito de coeficientes wavelet  $(W_\psi x)(m, n)$ ;
- O processamento é realizado sobre tempo contínuo.

### 3.2.4 Transformada Wavelet Discreta Inversa (TWDI)

No caso contínuo, dada uma função wavelet mãe, uma função qualquer  $x(t)$  pode sempre ser recuperada do seu conjunto de coeficientes wavelet contínuos. No caso discreto, entretanto, o processo de reconstrução pode não convergir para a função  $x(t)$ . A reconstrução depende da escolha da wavelet mãe e do processo de discretização realizado.

De acordo com Daubechies (DAUBECHIES, 1992), a reconstrução ideal seria aquela que ocorresse com o máximo de eficiência e com um mínimo de perda de informação. Neste sentido, a função  $x(t)$  pode ser reconstruída a partir dos seus coeficientes wavelet discretos com uma aproximação razoavelmente boa, através da equação (3.21).

$$x(t) \approx c \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} ((W_\psi x)(m, n)) (\psi_{m,n}(t)) \quad (3.21)$$

Sendo  $c$  uma constante que depende do processo de discretização e da wavelet mãe utilizada.

A seguir, tem-se um típico modelo TWD aplicada em um sinal unidimensional através da transformada de Haar, que consiste em calcular a média e a diferença entre os elementos de um vetor dois a dois (STOLLNITS; DEROSE; SALESIN, 1996).

Considere  $S_n$  um vetor com  $n$  elementos representantes do sinal.

$$S_n = (s_1, s_2, \dots, s_n) \quad (3.22)$$

onde:

$$M_{n/2} = (m_1, m_2, \dots, m_{n/2}) = \left( \frac{(s_1 + s_2)}{2}, \frac{(s_3 + s_4)}{2}, \dots, \frac{(s_{n-1} + s_n)}{2} \right) \quad (3.23)$$

$$D_{n/2} = (d_1, d_2, \dots, d_{n/2}) = \left( \frac{(s_1 - s_2)}{2}, \frac{(s_3 - s_4)}{2}, \dots, \frac{(s_{n-1} - s_n)}{2} \right) \quad (3.24)$$

sendo:

$n$ : número de elementos pertencentes ao vetor;

$M$ : vetor que representa as médias  $(m_1, m_2, \dots, m_{n/2})$  calculadas;

$D$ : vetor que representa as diferenças  $(d_1, d_2, \dots, d_{n/2})$  calculadas.

A transformada inversa de Haar recalcula os valores originais a partir da média e das diferenças obtidas da seguinte maneira:

$$S'_n = (m_1 + d_1, m_1 - d_1, m_2 + d_2, m_2 - d_2, \dots, m_{n/2} + d_{n/2}, m_{n/2} - d_{n/2}) \quad (3.25)$$

As equações (3.23), (3.24) e (3.25), respectivamente representam o comportamento da wavelet de Haar na decomposição e reconstrução simultaneamente de um sinal.

Um tópico muito importante, que envolve a transformada wavelet para o processamento de imagens, é a Análise de Multirresolução (AMR).

A análise de Multirresolução permite analisar uma imagem em diferentes resoluções além de ser versátil para aplicações em várias outras áreas tais como sinais de áudio e vídeo, sinais biomédicos, medidas industriais, análise de dados, dentre outros.

É através da análise de multirresolução que se pode comprimir ou remover ruídos, reconstruir dados perdidos, reconhecer padrões, entre outras aplicações.

### 3.3 Análise de Multirresolução (AMR)

Mallat (1986) concebeu uma análise de multirresolução, estabelecendo uma seqüência de subespaços encaixantes  $V_j \subset L^2(\mathfrak{R})$ , sendo:

$j$ : número inteiro ( $j \in \mathbb{Z}$ );

$L^2(\mathfrak{R})$ : espaço das funções de quadrado integrável a Lebesgue;

$V_j$ : seqüência de espaços encaixantes; ou seja:

$$\dots V_{-1} \supset V_0 \supset V_1 \supseteq V_2 \dots \quad (3.26)$$

Tem-se o complemento ortogonal de  $V_j$  como subconjunto de  $V_{j+1}$ , denotado por  $W_j$ .

$$V_{j+1} = V_j \oplus W_j \quad (3.27)$$

$\oplus$ : indica a soma direta dos subespaços ortogonais.

Os subespaços  $W_j$  contêm os detalhes necessários para construir  $V_{j+1}$  a partir de  $V_j$ .

$$V_j = W_j \oplus W_{j-1} \oplus W_{j-2} \oplus \dots \quad (3.28)$$

A cada estágio de decomposição:

$$W_j = V_j - V_{j+1} \quad (3.29)$$

no domínio da frequência, tem-se:

$W_j$ : componentes de alta frequência de  $V_{j+1}$ ;

$V_j$ : componentes de baixa frequência de  $V_{j+1}$ .

Estes espaços têm as seguintes propriedades descritas a seguir (DAUBECHIES, 1990).

(M1)  $V_j = V_{j+1} \oplus W_j$ : existe um subespaço complementar  $W_j$  em que os detalhes são armazenados.

(M2)  $V_{j+1} \subset V_j$ : são subespaços encaixados como citado anteriormente.

(M3)  $\overline{\bigcup_{j \in \mathbb{Z}} V_j} = L^2(\mathfrak{R})$ : esta seqüência contém a seguinte propriedade: Toda função  $f \in L^2(\mathfrak{R})$  pode ser representada por suas projeções.

(M4)  $\overline{\bigcap_{j \in \mathbb{Z}} V_j} = \{0\}$ : a propriedade (M4) equivale dizer, que as projeções têm energia arbitrariamente pequena ao passo quando  $j$  aumenta, fazendo com que os detalhes sejam cada vez mais evidentes, nível por nível, em ordem crescente, conforme  $j \rightarrow \infty$ .

(M5)  $x(t) \in V_{j+1} \Leftrightarrow x(2t) \in V_j$ : a função escalar gera um subespaço de referência e aplicando a propriedade (M5) sucessivamente concluí-se, que todos os subespaços  $V_j$  também são gerados pela função escala, assim como os subespaços  $W_j$  também são gerados pela função wavelet.

A projeção ortogonal de uma função  $f \in L^2(\mathbb{R})$  em  $V_j$  é obtida através da função de escala, que por sua vez define um filtro passa - baixa. A frequência de corte desse filtro é indicada por  $\alpha_j$ .

A Figura 3.5 explica como são representados esses cortes e quais os espaços criados por eles (KOSLOSKI, 2000)

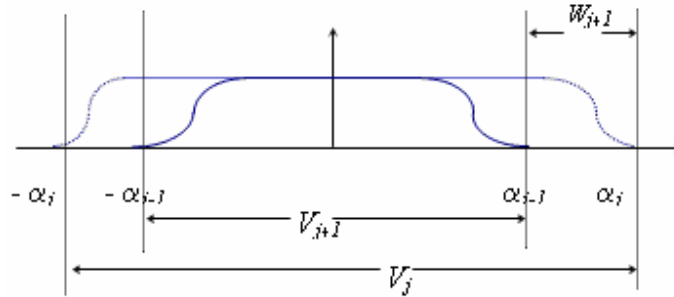


Figura 3. 5: Bandas de frequências entre  $V_j$  e  $V_{j+1}$ .

Pode-se observar que o espaço  $V_j$  contido em  $[-\alpha_j, \alpha_j]$  é obtido a partir do espaço  $V_{j+1}$  que está contido em  $[-\alpha_{j+1}, \alpha_{j+1}]$ , e  $W_{j+1}$  que está contido em  $[\alpha_{j+1}, \alpha_j]$ .

Quando se passa de  $V_j$  para  $V_{j+1}$  a escala aumenta de  $2^j$  para  $2^{j+1}$  e a banda de frequência diminui para um intervalo  $[-\alpha_{j+1}, \alpha_{j+1}]$ . O que significa, do ponto de vista de processamento de imagens, melhorar a visão em relação à imagem, ou seja, dar um zoom na imagem.

A seguir serão abordados os principais tópicos que nos permite compreender como a análise de multiresolução é aplicada usando a transformada wavelet em processamento de imagens.

### 3.4 Transformada Wavelet Discreta Bidimensional (TWD Bidimensional)

Existem duas formas comuns nas quais as wavelets podem ser usadas para transformar os valores dos pixels dentro de uma imagem. Cada uma destas transformações é uma generalização bidimensional da TWD unidimensional.

A primeira transformada é chamada de *decomposição padrão*. Para obter a decomposição padrão de uma imagem, aplica-se primeiro a TWD unidimensional a cada linha de valores de pixels. Esta operação resulta em um valor médio para cada linha.

Feito isto, tratam-se estas linhas transformadas como se elas fossem uma imagem, e aplica-se a TWD unidimensional para cada coluna. Os valores resultantes são todos os coeficientes de detalhes, exceto por um único coeficiente que representa a média geral.

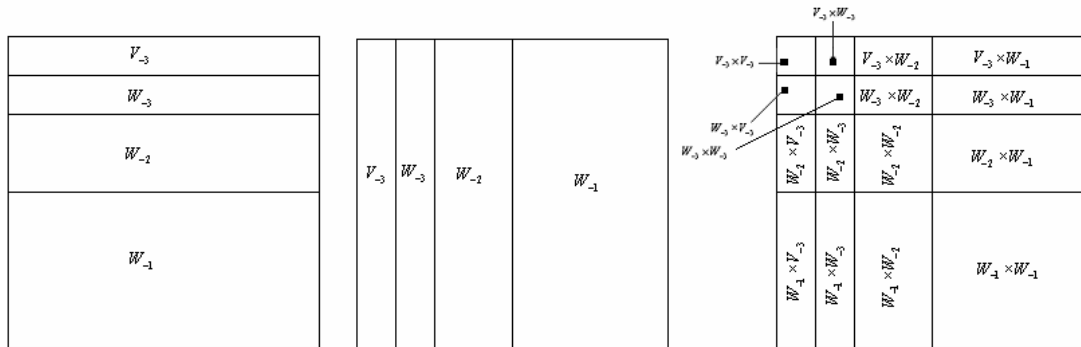


Figura 3. 6: Decomposição padrão de uma imagem.

O segundo tipo de TWD bidimensional, chamado de *decomposição não-padrão*. São realizadas operações de decomposição alternadas entre linhas e colunas. Primeiro aplica-se o cálculo da média nos pares horizontais e faz-se a diferença dos valores dos pixels em cada linha da matriz que representa a imagem. Depois, aplica-se o cálculo da média nos pares verticais e encontra-se a diferença para a coluna do resultado.

A transformação é completada, repetindo o processo recursivamente apenas no quadrante contendo as médias em ambas as direções.

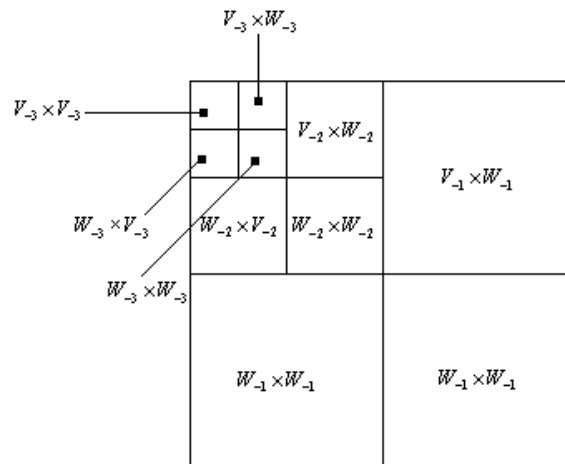


Figura 3. 7: Decomposição não padrão de uma imagem.

### 3.5 Transformada Wavelet: Uma interpretação do ponto de vista de processamento de imagens

A análise de multiressolução também conhecida como Algoritmo Piramidal de Mallat, é um método proposto para implementar a transformada wavelet discreta. Este método refere-se ao procedimento de se obter "aproximações" e "detalhes" de uma dada imagem.

Uma aproximação é uma representação de baixa frequência da imagem original, enquanto que um detalhe é a diferença entre duas aproximações sucessivas da imagem original. Uma aproximação mantém a tendência geral da imagem original enquanto que um detalhe mostra os componentes de alta frequência da mesma (SANTIAGO; PEDERIVA, 2006).

O método proposto por Mallat aborda técnicas de filtragem. A filtragem é feita utilizando matrizes denominadas máscaras, as quais são aplicadas sobre a imagem, com centro na posição  $(i,j)$ , sendo  $i$  o número de uma dada linha e  $j$  o número de uma dada coluna sobre a imagem, o valor do pixel na posição  $(i,j)$  é substituído por um novo valor que depende dos valores dos pixels vizinhos e dos pesos da máscara, gerando uma nova imagem com a eliminação das linhas e colunas iniciais e finais da imagem original, assim algumas componentes de frequência são eliminadas e outras não.



Outros tipos de filtro **passa-baixa**, conhecidos como **filtros de média ponderada**, são usados quando os pesos são definidos em função de sua distância do peso central. A seguir são apresentados filtros desse tipo com dimensão 3x3:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} * \frac{1}{10} \qquad \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} * \frac{1}{16}$$

### 3.5.2 Filtro passa-alta

A filtragem passa-alta  $G(t)$  tende a realçar os detalhes, produzindo uma "agudização" ("sharpering") da imagem, isto é, as transições entre regiões diferentes tornam-se mais nítidas.

Por exemplo, certas características presentes na imagem são realçadas, tais como bordas, linhas curvas ou manchas. O efeito indesejado provocado por esse tipo de filtro é o de enfatizar um ruído, porventura existente na imagem.

A representação matricial do filtro passa-alta é definida por:

$$\frac{1}{n} \times \begin{bmatrix} -d_1^1 & -d_2^1 & \dots & -d_i^1 \\ -d_1^2 & -d_2^2 & \dots & -d_i^2 \\ \vdots & \vdots & \ddots & \vdots \\ -d_1^j & -d_2^j & \dots & -d_j^j \end{bmatrix}_{i \times j}$$

onde, o pixel central será do tipo  $-\sum_{i=1}^{\infty} d_i^j$ .

No caso de filtragem passa-alta, alguns exemplos são dados a seguir:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Os filtros passa-baixa  $H(t)$  e passa-alta  $G(t)$  dão origem ao banco de filtros.

### 3.5.3 Banco de Filtros

A função de um banco de filtros é proporcionar o processo de decomposição e reconstrução de um sinal. Para que um banco de filtros tenha uma construção perfeita,  $H(t)$  e  $G(t)$  devem satisfazer certas condições.

Antes é preciso encontrar um filtro passa-baixa  $P(t)$  que satisfaça as duas condições seguintes:

$$P(t) \geq 0 \quad (3.30)$$

e

$$P(t) + P(t + \pi) = 1 \quad (3.31)$$

Em seguida, determina-se  $H(t)$ , de modo que:

$$|H(t)|^2 = P(t) \quad (3.32)$$

Portanto para todo  $t$ :

$$|H(t)|^2 + |H(t + \pi)|^2 = 1 \quad (3.33)$$

O par de filtros  $\{H(t) \text{ e } G(t)\}$  é denominado: *Quadrature Mirror Filters*.

### 3.5.4 Algoritmo Piramidal de Mallat

O algoritmo Piramidal apresenta dois estágios: um para o cálculo da Transformada Wavelet Discreta (TWD) e outro para o cálculo da Transformada Wavelet Discreta Inversa (TWDI) (REIS; SILVA, 2004).

No processo de decomposição de um sinal, neste caso, uma imagem original representada por  $s[t]$ , a função escala é correlacionada com o filtro passa-baixa  $H(t)$  produzindo do ponto de vista matemático, coeficientes de aproximação  $c_1[t]$ . A função wavelet é correlacionada com o filtro passa-alta  $G(t)$  o qual fornece os chamados coeficientes de detalhes  $d_1[t]$ , ambos os coeficientes são obtidos em diferentes níveis de resolução.

As equações a seguir expressam as relações entre  $s[t]$ ,  $c_1[t]$  e  $d_1[t]$  com a transformada wavelet:

$$c_j[s(t)] = s(t) * \phi_i^j(t) \tag{3.34}$$

$$d_j[s(t)] = s(t) * \psi_i^j(t) \tag{3.35}$$

onde  $\phi_i^j(t)$  e  $\psi_i^j(t)$  representam as funções de escala e as funções wavelets e \* denota a operação de convolução.

Como  $\phi_i^j(t)$  e  $\psi_i^j(t)$  estão correlacionadas através de um par de filtros

$\{H(t)$  e  $G(t)\}$ , elas podem ser definidas por:

$$\phi_i^j(t) = H(t) * \phi_i^{j-1}(t) \tag{3.36}$$

$$\psi_i^j(t) = G(t) * \psi_i^{j-1}(t) \tag{3.37}$$

Resolvendo  $H(t)$  e  $G(t)$  das expressões (3.36) e (3.37) e combinando com as expressões (3.34) e (3.35) a Transformada Wavelet Discreta é obtida usando as seguintes fórmulas:

$$c_j[s(t)] = H(t) * c_{j-1}[s(t)] \tag{3.38}$$

$$d_j[s(t)] = G(t) * d_{j-1}[s(t)] \tag{3.39}$$

A Figura 3.8 apresenta um exemplo da aplicação da Transformada Wavelet Discreta (TWD) no processo de decomposição da imagem original  $s[t]$ .

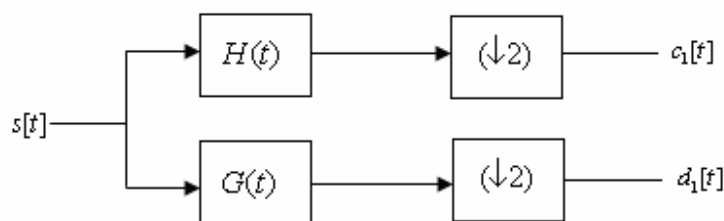


Figura 3. 8: Diagrama de um banco de filtros no processo de decomposição.

Após a filtragem, a cada duas saídas dos filtros, descarta-se uma delas. Esse processo é feito através dos operadores *downsampling* ( $\downarrow 2$ ). No final do processo são geradas as saídas  $c_1[t]$ , que representam os componentes de baixa frequência, e  $d_1[t]$ , que representam os componentes de alta frequência. Esse processo pode ser repetido quantas vezes forem necessárias.

A TWDI reconstrói a imagem original  $s[t]$  a partir de  $c_1[t]$  e  $d_1[t]$ , inserindo zeros entres os coeficientes wavelets obtidos no processo de decomposição. Também são utilizados operadores *upsampling* ( $\uparrow 2$ ) nos dois sinais de saída, juntamente com dois filtros  $H_1(t)$  e  $G_1(t)$ , para processar as imagens resultantes e as combinarem, reconstruindo assim, a imagem original ou uma aproximação dela ( $\hat{s}[t]$ ), conforme o diagrama apresentado na Figura 3.9.

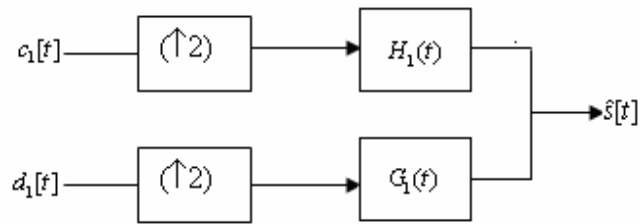


Figura 3. 9: Diagrama de um banco de filtros no processo de reconstrução.

As relações entre  $s[t]$ ,  $c_1[t]$  e  $d_1[t]$  estão dadas através das seguintes equações:

$$c_1[t] = \sum_n h[n-2t].s[n]. \tag{3.40}$$

$$d_1[t] = \sum_n g[n-2t].s[n]. \tag{3.41}$$

A partir do sinal  $c_1[t]$  podem ser obtidas mais duas faixas, e assim sucessivamente, com o processo sendo finalizado quando o comprimento do último sinal  $c_L[t]$  é um.

O número de níveis e de faixas que pode ser obtido é proporcional ao comprimento do sinal a ser processado. Definindo  $N$  como sendo o tamanho da seqüência de entrada, o número máximo de níveis e faixas é dado por (RIOUL; VETTERLI, 1991):

$$L = \log_2(N) \tag{3.42}$$

o que exige um sinal de entrada com comprimento em potência de 2.

Na Figura 3.10 é apresentada uma estrutura completa de filtragem através da Transformada Wavelet.



## CAPÍTULO 4

# WAVELET EM PROCESSAMENTO DE IMAGENS

O uso da transformada wavelet (TW) na área de processamento de sinais tem ganhado cada vez mais adeptos, pois ela, além de evidenciar muitos detalhes de um sinal, é uma ferramenta de fácil implementação.

Uma área importante em processamento de sinais, na qual a TW é muito usada, é o processamento de imagens. Além das imagens apresentarem características que encaixam nas técnicas propostas pela TW, elas também têm sido o foco principal de pesquisas devido a grande quantidade de informações a serem armazenadas, transmitidas e manuseadas por parte de usuários de computadores (OSBORNE et al., 2002).

Como as imagens são representadas como sinais bidimensionais, o seu processamento por meio da transformada wavelet não é tarefa difícil, para isto basta a aplicação da transformada wavelet bidimensional na matriz que representa a imagem (STOLLNITS; DEROSE; SALESIN, 1996).

A ferramenta matemática que faz com que a TW seja um instrumento potente na área de processamento de imagens é a *análise de multirresolução* (DAUBECHIES, 1990), que permite que uma imagem seja decomposta em vários níveis de resolução (FALCÃO; LEITE, 1998), evidenciando seus detalhes em cada um desses níveis (COIFMAN, 1990).

É como se aplicasse um “zoom” na imagem fazendo com que em uma análise de grande escala os pequenos detalhes sejam caracterizados em uma escala menor. Ao final do processo a caracterização das formas é um conjunto de dados extraídos em diferentes escalas.

Em outras palavras, as wavelets com o auxílio da análise de multiresolução decompõem a imagem em uma combinação linear de versões transladadas da wavelet mãe

em diferentes escalas detectando, por exemplo, bordas que são classificadas com uma das características mais importante em uma imagem, tornando-a muito útil, por exemplo, no processamento de imagens biomédicas (ALBUQUERQUE, 2001).

Outra aplicação muito útil da transformada wavelet no processamento de imagens é a *compressão* que se baseia na eliminação de coeficientes redundantes da imagem no domínio wavelet (STOLLNITS; DEROSE; SALESIN,1996), visando reduzir a quantidade de dados desnecessários para representar uma imagem digital.

A escolha da wavelet para esse tipo de problema depende muito do que se deseja resolver. No caso de compressão, ao escolher uma wavelet que melhor se adapta aos dados, os coeficientes menores do que um valor previamente estabelecido devem ser truncados, sendo representados por uma codificação esparsa, fazendo, desta forma, uma excelente ferramenta no campo de compressão de imagens.

Recentemente, técnicas baseadas na transformada wavelet, como a compressão de imagens no sistema JPEG 2000, tornou-se uma tecnologia extremamente necessária (OSBORNE et al., 2002). O uso da TW, em compressão de imagens, visa representar uma imagem com o menor número de bits possível. Para isso é preciso reduzir a correlação existente entre os pixels, concentrando a maior quantidade de informação possível em um número menor de coeficientes.

Uma imagem comprimida ou compactada por meio da TW ocupa menos espaço em seu armazenamento, otimizando, desta forma, o espaço a qual se encontra, podendo posteriormente ser descompactada obtendo a imagem original ou uma aproximação dela com uma menor degradação a ser percebida pelo observador (STOLLNITS; DEROSE; SALESIN, 1996).

A escolha no uso da transformada wavelet no sistema de compressão é devido à sua característica de apresentar a capacidade de concentrar informações relevantes em um número pequeno de coeficientes e também, por ser flexível com relação a duração e posição das funções bases, permitindo uma boa representação por um número reduzido de funções.

A partir daí pode-se, então, utilizar métodos com ou sem perda, dependendo da aplicação, embora o uso de transformadas em geral esteja associada a métodos com perda (STOLLNITS; DEROSE; SALESIN,1996).

## 4.1 Compressão de Imagens

Nas últimas décadas, a necessidade de compressão de dados com uma alta taxa de compressão se tornou muito necessária. O requerimento básico de qualquer método de compressão é que se possa mudar rapidamente as informações originais e vice-versa.

Há dois tipos de compressão: compressão sem perda e compressão com perda.

*Compressão sem perda:* é normalmente aplicada quando a qualidade e a fidelidade da imagem são importantes e devem mantidas, consistindo na recuperação das informações sem perdas da qualidade, ou seja, os coeficientes eliminados não comprometem o processo de reconstrução (FALCÃO; LEITE, 1998).

*Compressão com perda:* é utilizada quando a portabilidade e a redução do tamanho da imagem são mais importantes que a qualidade visual, sem no entanto menosprezá-la, neste caso pode ocorrer perda de informações. Este problema pode ser contornado desde que a decomposição seja aceitável. Esse tipo de compressão permite que os fatores de resolução sejam atingidos em um grau maior, porém, só poderá ser usada quando os dados originais forem substituídos por uma aproximação fácil de comprimir (FALCÃO; LEITE, 1998). Normalmente o uso de transformada está associado a métodos com perda.

A idéia geral de representar os dados é utilizar uma base matemática. Tal base deve satisfazer certas condições, como boa localização no tempo e na frequência, tornando os coeficientes muito pequenos (FALCÃO; LEITE, 1998).

O uso da TW no processo de compressão faz com que coeficientes que são menores de um determinado valor possam ser eliminados sem comprometer a reconstrução da imagem. Desta forma, o número de coeficientes significativos para a reconstrução do sinal se torna cada vez menor e as informações mais importantes estão concentradas nesse conjunto. Porém, há certos requisitos para satisfazer as condições de um algoritmo baseado em compressão (CENÁRIO, 2004):

- 1) ser independente dos dados;
- 2) ter um baixo custo computacional;
- 3) ser capaz de remover a correlação para um conjunto grande de dados.

As bases de funções wavelets se encaixam nessas condições, tornando-se uma alternativa de trabalho. A compressão de imagens usando a TW segue três passos descritos a seguir.

1) calcular os coeficientes através de um algoritmo rápido tanto na decomposição quanto na reconstrução, a partir de seus coeficientes do filtro.

2) produzir uma sequência em ordem decrescente do valor dos coeficientes, podendo usar funções bases para normalizar e usar qualquer técnica padrão;

3) estipular um erro, onde qualquer busca padrão poderá ser usada. Porém, para imagens, grandes escolhas tornam o processo lento.

Em geral, a compressão de uma imagem consiste na eliminação dos coeficientes redundantes no domínio transformado, que são trocados por zero. A imagem comprimida é armazenada no domínio transformado e, para tanto, a sua recuperação deve ser feita usando a transformada inversa.

A Figura 4.1 apresenta a imagem de ultra-som decomposta e reconstruída pela transformada wavelet de Haar, usando quatro níveis de resolução, no software *MATLAB*. A taxa de compressão obtida nesse processamento é de 98%.

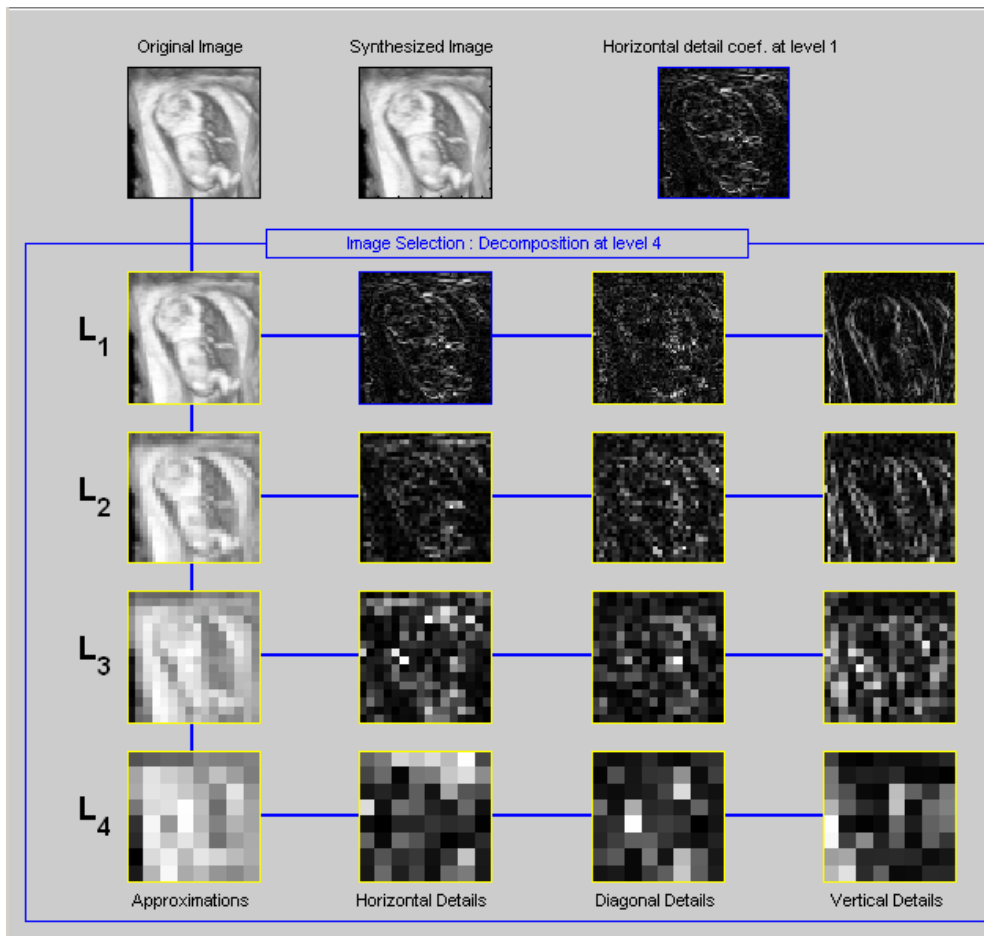


Figura 4. 1: Imagem decomposta e comprimida com 98%.

## 4.2 Métodos de Compressão e Eliminação de Ruído

Os métodos de compressão são utilizados quando se deseja processar dados de forma correta com taxas de compressão cada vez mais altas, preservando o aspecto visual e levando em conta diversos aspectos computacionais e matemáticos.

Dentre vários métodos de compressão propostos, a TWD tem se destacado em várias pesquisas, com aplicações em um número cada vez maior de problemas ligados diretamente ou indiretamente à área de processamento de sinais (SILVA et al., 2007).

Das várias aplicações podemos citar a eliminação de ruído (denoising) em sinais, bem como compressão, detecção e reconhecimento de padrões, ressaltando que na medida em que a teoria vem sendo difundida e compreendida, novas aplicações vêm surgindo nas mais diversas áreas.

O problema mais comum com as imagens é a presença de ruído. Podemos classificar os ruídos como pequenas características que atrapalham uma boa visualização da imagem, ou seja, são informações indesejadas presentes na maioria das imagens, devido principalmente as condições ambientais em que elas são obtidas.

Os pixels da imagem com ruído aparecem como pontos com níveis de cinza bem diferentes da sua vizinhança de forma escura (pretos) ou saturados (brancos). Esses pontos ruidosos podem aparecer distribuídos aleatoriamente ou de forma sistemática (listas verticais e horizontais) e dentre as técnicas mais utilizada para reduzir ruídos no domínio wavelet, os métodos de redução por limiar são os mais usados (SOARES et al., 2007).

A idéia básica usada na eliminação de ruído, utilizando a TW, é escolher quais os coeficientes que serão mantidos para preservar a informação que a imagem carrega, removendo, assim os coeficientes associados ao ruído. Ou seja, obtém-se um sinal com menos ruído pelo cancelamento dos coeficientes abaixo de um valor que é estabelecido baseado na potência do ruído.

A TW possui duas propriedades que facilitam a eliminação de ruído em sinais. A primeira é que apenas alguns coeficientes de decomposição serão não-nulos se as funções base forem selecionadas adequadamente de acordo com as características do sinal. Assim se consegue uma alta concentração de energia nesses coeficientes.

A outra propriedade é que o sinal apresenta a mesma distribuição de ruído em todos os seus coeficientes. Por exemplo, se o sinal apresenta distribuição gaussiana, os coeficientes *wavelet* também apresentarão a mesma distribuição, supondo sempre que a contribuição do ruído se distribua de maneira uniforme por todos os coeficientes.

Uma abordagem genérica para resolver o problema da eliminação de ruído foi proposta originalmente por Donoho e Johnstone (DONOHO; JOHNSTONE, 1994). Este método consiste em eliminar os coeficientes que têm valores absolutos menores que certo limiar (*thresholding*) estabelecido de acordo com alguns critérios, ou seja, coeficientes do sinal transformado que estão abaixo de certo valor são eliminados ou reduzidos.

O algoritmo genérico proposto por Donoho e Johnstone para realizar este processo pode ser resumido nos seguintes passos:

- 1) aplicar algum algoritmo *wavelet* (TW) rápido aos dados de entrada;

- 2) aplicar uma função de limiar (*thresholding*) aos coeficientes de detalhe da transformada, usando um limiar especialmente estimado;
- 3) aplicar a Transformada wavelet inversa para recompor o sinal no domínio do tempo.

O procedimento genérico de eliminação de ruído depende principalmente da definição do limiar. Alguns métodos de aplicação de limiar bastante utilizados na literatura especializada são descritos na próxima seção. Estes métodos serão dados como opções para o processamento redução de ruído ou compressão das imagens que poderão ser trabalhadas pela interface gráfica proposta.

### 4.3 Métodos de Limiar

Em (DONOHO; JOHNSTNE, 1994) são propostos dois métodos de compressão por limiar. Esses métodos são os mais antigos e os mais utilizados na literatura quando o objetivo é compressão de imagens. São eles o limiar rígido (*Hard-Thresholding*) e o limiar Suave (*Soft-Thresholding*).

Em todos os métodos, o limiar  $\lambda$  usado é o limiar universal proposto por Donoho e Johnstone (DONOHO; JOHNSTNE, 1994), definido na equação (4.1).

$$\lambda = \sigma \sqrt{2 \log_{10}(N)} \quad (4.1)$$

sendo  $\sigma$  a estimativa ou perfil do ruído presente no sinal, conforme a equação (4.2).

$$\sigma = \text{mediana}(|\hat{y}|) / 0.6745 \quad (4.2)$$

#### 4.3.1 Limiar Rígido (Hard- Thresholding)

O Limiar Rígido é aplicado nos coeficientes do sinal ou imagem transformada conforme a equação (4.3), obtendo a saída  $y_{thr}$ , ou seja, um limiar  $\lambda > 0$  é calculado e os coeficientes que têm valores absolutos menores que  $\lambda$  são eliminados.

$$y_{thr} = \begin{cases} \hat{y} & \text{se } |\hat{y}| > \lambda \\ 0 & \text{se } |\hat{y}| \leq \lambda \end{cases} \quad (4.3)$$

sendo

$\hat{y}$  : sinal transformado;

$\lambda$  : limiar.

As características de entrada e saída do limiar rígido são apresentadas na Figura 4.2.

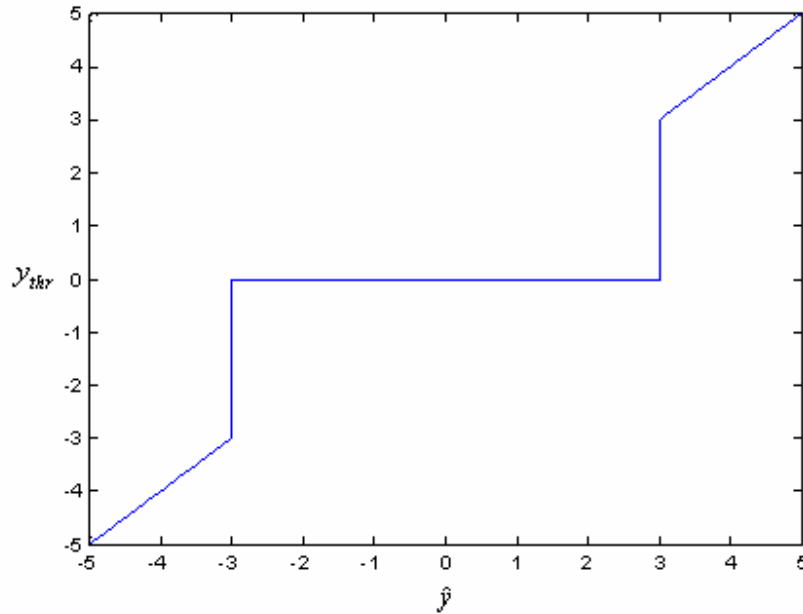


Figura 4. 2: Limiar Rígido.

### 4.3.2 Limiar Suave (Soft - Thresholding)

No Limiar Suave, os coeficientes que têm valores absolutos menores que um dado  $\lambda > 0$  são eliminados da mesma forma que na equação (4.3). Coeficientes que estão acima de  $\lambda$  são diminuídos na magnitude de  $\lambda$ , procurando evitar a descontinuidade apresentada pelo limiar rígido.

A equação (4.4) apresenta a função de transferência do limiar suave, cujo gráfico é apresentado na Figura 4.3.

$$y_{thr} = \begin{cases} \text{sgn}(\hat{y})(|\hat{y} - \lambda|) & \text{se } |\hat{y}| > \lambda \\ 0 & \text{se } |\hat{y}| \leq \lambda \end{cases} \quad (4.4)$$

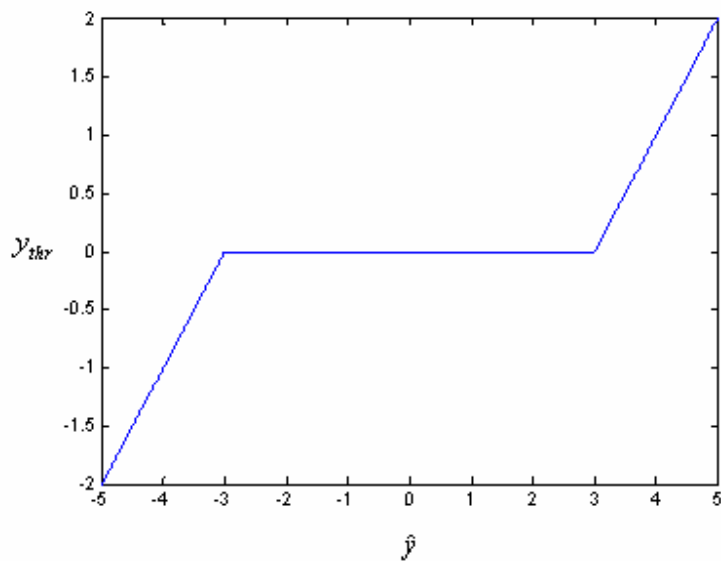


Figura 4. 3: Limiar Suave.

Além dos dois limiares básicos, rígido e suave, existem outros limiares com algumas modificações. O primeiro deles chamado Limiar Rígido Suave (*Hard soft thresholding*) (STORM, 1998), cuja função, funções de transferência, é apresentada na equação (4.5).

### 4.3.3 Limiar Rígido Suave (Hard Soft - Thresholding)

Dados  $\lambda > 0$  e  $\alpha \in [0,1]$  considere a função  $y_{thr} = y_{thr}(\hat{y})$  definida pela equação (4.5).

$$y_{thr} = \begin{cases} \hat{y} & \text{se } |\hat{y}| > \lambda \\ \alpha \hat{y} & \text{se } |\hat{y}| \leq \lambda \end{cases} \quad (4.5)$$

No Limiar Rígido Suave, os coeficientes que têm valor absoluto menor que  $\lambda$  são atenuados por  $\alpha \hat{y}$  enquanto que os que estão acima de  $\lambda$  são mantidos. Considera-se que este limiar é uma variação do limiar rígido, pois, se  $\alpha = 0$  a equação (4.5) é semelhante à equação (4.3). Assim, o que diferencia os limiares rígido e rígido suave é que no segundo os coeficientes que estão abaixo do limiar não são eliminados de forma abrupta, mas sim atenuados.

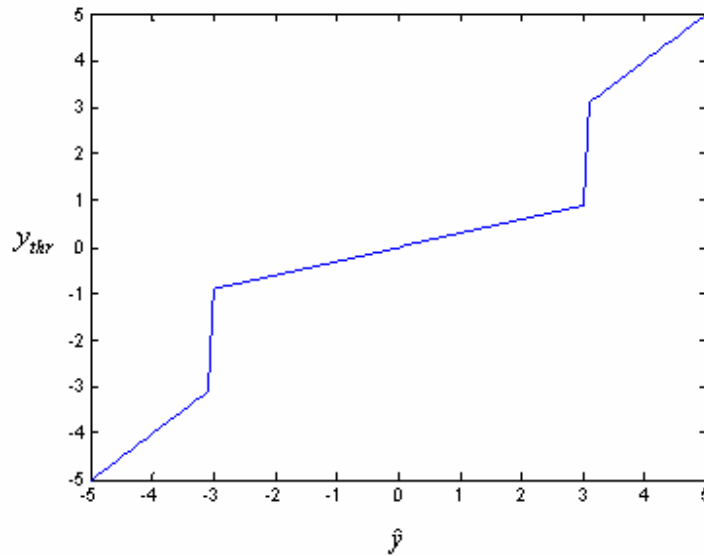


Figura 4. 4: Limiar Rígido Suave.

Outro método proposto é Limiar Super Suave (*Super soft thresholding*) (STORM, 1998), definido na equação (4.6) com gráfico apresentado na Figura 3.5.

#### 4.3.4 Limiar Super Suave (Super Soft - Thresholding)

Dados  $\lambda > 0$  e  $\alpha \in [0,1]$ , considere a função  $y_{thr} = y_{thr}(\hat{y})$  definida pela equação (4.6).

$$y_{thr} = \begin{cases} \hat{y} - \text{sgn}(\hat{y})(1 - \alpha)\lambda & \text{se } |\hat{y}| > \lambda \\ \alpha\hat{y} & \text{se } |\hat{y}| \leq \lambda \end{cases} \quad (4.6)$$

No Limiar Super Suave, os coeficientes de modo geral são atenuados, com a única diferença que, os coeficientes que têm valor absoluto menor que  $\lambda$  são atenuados por  $\alpha\hat{y}$ , enquanto que os coeficientes que têm valor absoluto maior que  $\lambda$  são atenuados por  $\hat{y} - \text{sign}(\hat{y})(1 - \alpha)\lambda$ .

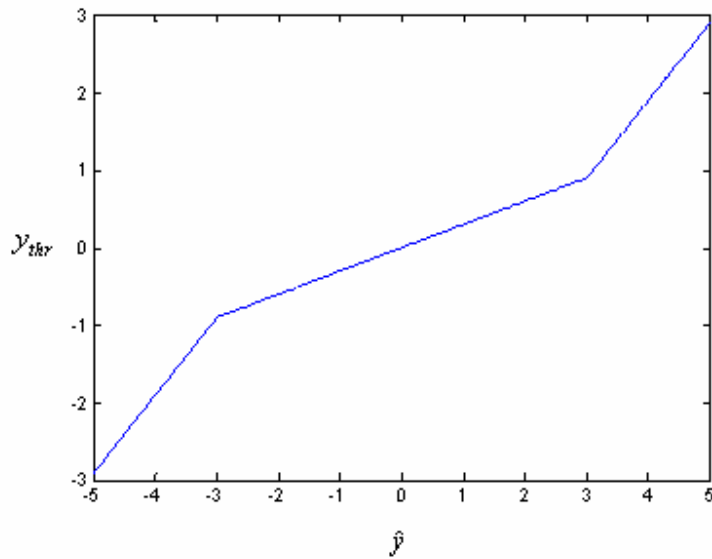


Figura 4. 5: Limiar Super Suave.

Observando a equação (4.6) pode-se verificar que este método é uma variação do limiar suave, com  $\alpha \in [0,1]$ , e quando  $\alpha = 0$ , a equação (4.6) se torna semelhante à equação (4.4).

Os métodos apresentados a seguir são métodos baseados em curvas de atenuação para o sinal transformado  $\hat{y}$  (STORM, 1998).

### 4.3.5 Limiar Super Super Suave (Super Super Soft-Thresholding)

Dados  $\lambda > 0$  e  $\alpha \in [0,1]$ , no Limiar Super Super Suave, ocorre atenuação de todos os coeficientes, de acordo com a função de transferência associada  $y_{thr} = y_{thr}(\hat{y})$  definida por:

$$y_{thr} = \begin{cases} \hat{y} - \text{sgn}(\hat{y})(1-\alpha)\lambda & \text{se } |\hat{y}| > 1.5\lambda \\ \text{sgn}(\hat{y})(c|\hat{y}|^2 + b|\hat{y}| + a) & \text{se } 0.5\lambda < |\hat{y}| \leq 1.5\lambda \\ \alpha\hat{y} & \text{se } |\hat{y}| \leq 0.5\lambda \end{cases} \quad (4.7)$$

sendo  $a = (1-\alpha)\lambda/8$ ,  $b = 1.5\alpha - 0.5$  e  $c = (1-\alpha)/2\lambda$ . Observe que, o gráfico da função polinomial em (4.7) une os pontos  $(0.5\lambda, \alpha\hat{y})$ , com  $\hat{y} = 0.5\lambda$  e  $(1.5\lambda, \hat{y} - \text{sign}(\hat{y})(1-\alpha)\lambda)$ ,

com  $\hat{y} = 1.5\lambda$ , notando que no segundo caso, o valor das funções  $y_{thr} = \hat{y} - \text{sgn}(\hat{y})(1 - \alpha)\lambda$  e  $y_{thr} = \text{sgn}(\hat{y})(c|\hat{y}|^2 + b|\hat{y}| + a)$  em  $\hat{y} = 1.5\lambda$  é  $0.5\lambda + \alpha\lambda$ . Observar também que, a inclinação das retas tangentes ao gráfico da função polinomial em (4.7), nos pontos extremos considerados, são iguais a  $\alpha$  e 1, respectivamente. Assim, a inclinação dessas retas tangentes nesses pontos coincidem com a inclinação das retas definidas na terceira e na primeira equação em (4.7), respectivamente. Na Figura 4.6 é apresentado o gráfico da função  $y_{thr} = y_{thr}(\hat{y})$  definida em (4.7).

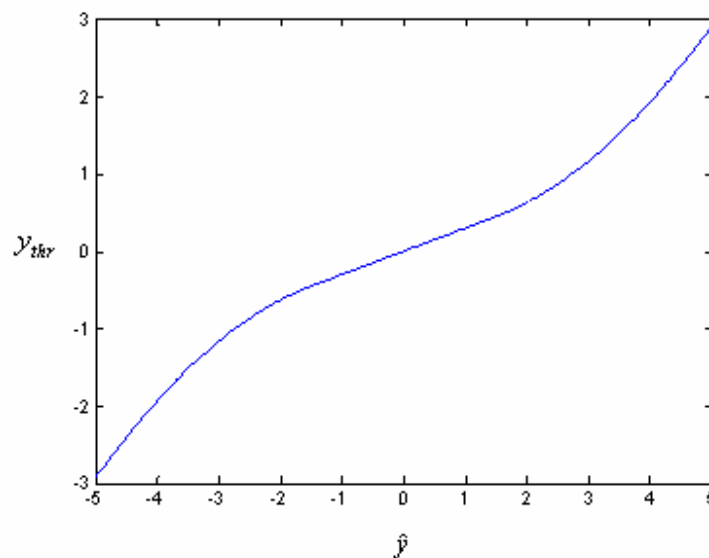


Figura 4. 6: Limiar Super Super Suave.

### 4.3.6 Limiar Rígido Super Suave (Hard Super Soft-Thresholding)

Dados  $\lambda > 0$  e  $\alpha \in [0,1]$ , no Limiar Rígido Super Suave, a atenuação é feita em todos os coeficientes, exceto quando  $|\hat{y}| > 1.5\lambda$ , de acordo com a função de transferência associada  $y_{thr} = y_{thr}(\hat{y})$  definida por:

$$y_{thr} = \begin{cases} \hat{y} & \text{se } |\hat{y}| > 1.5\lambda \\ \text{sgn}(\hat{y})(e|\hat{y}|^4 + d|\hat{y}|^3 + c|\hat{y}|^2 + b|\hat{y}| + a) & \text{se } 0.5\lambda < |\hat{y}| \leq 1.5\lambda \\ \alpha\hat{y} & \text{se } |\hat{y}| \leq 0.5\lambda \end{cases} \quad (4.8)$$

sendo  $a = 0$ ,  $b = 1$ ,  $c = 4.5(\alpha - 1)/\lambda$ ,  $d = 6(1 - \alpha)/\lambda^2$  e  $e = 2(\alpha - 1)\lambda^3$ . Observar que, o gráfico da função polinomial considerada em (4.8) une os pontos  $(0.5\lambda, \alpha\hat{y})$ ,  $(\lambda, 0.5\hat{y}(1 + \alpha))$  e  $(1.5\lambda, \hat{y})$ . Observar também que, a inclinação das retas tangentes ao gráfico da função polinomial em (4.8), nos pontos extremos  $(0.5\lambda, \alpha\hat{y})$  e  $(1.5\lambda, \hat{y})$  são iguais a  $\alpha$  e 1, respectivamente. Assim, a inclinação dessas retas tangentes nesses pontos coincidem com a inclinação das retas definidas na terceira e na primeira equação em (4.8), respectivamente. Na figura 4.7 é apresentada o gráfico da função  $y_{thr} = y_{thr}(\hat{y})$  definida em (4.8).

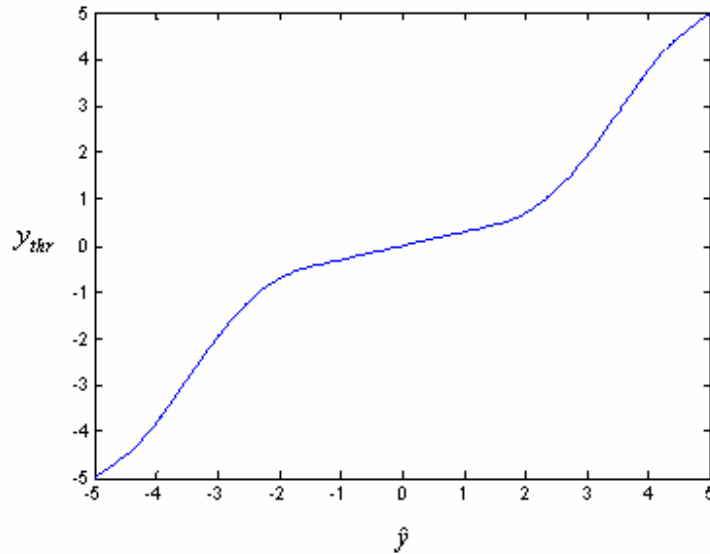


Figura 4. 7: Limiar Rígido Super Suave.

### 4.3.7 Limiar Semi-Suave (Semi- Soft - Thresholding)

No Limiar Semi-Suave (*Semi-soft Thresholding*) (BRUCE; GAO, 1996), conforme a função definida pela equação (4.9), dois limiares são utilizados  $\lambda_1$  e  $\lambda_2$ , com  $0 < \lambda_1 < \lambda_2$ , sendo que  $\lambda_2$  é dependente de  $\lambda_1$ , geralmente  $\lambda_2 = \sqrt{2}\lambda_1$ .

Quando o valor absoluto do coeficiente está abaixo de  $\lambda_1$ , esse coeficiente é substituído por zero, caso esteja acima de  $\lambda_2$  ele é mantido, ou dependendo, ele é atenuado como apresentado na equação (4.9).

Este método também é conhecido por *Firm Thresholding*. A Figura 4.8 apresenta o gráfico da função  $y_{thr} = y_{thr}(\hat{y})$  definida pela equação (4.9).

$$y_{thr} = \begin{cases} \hat{y} & \text{se } |\hat{y}| > \lambda_2 \\ \text{sgn}(\hat{y}) \frac{\lambda_2(|\hat{y}| - \lambda_1)}{\lambda_2 - \lambda_1} & \text{se } \lambda_1 \leq |\hat{y}| \leq \lambda_2 \\ 0 & \text{se } |\hat{y}| < \lambda_1 \end{cases} \quad (4.9)$$

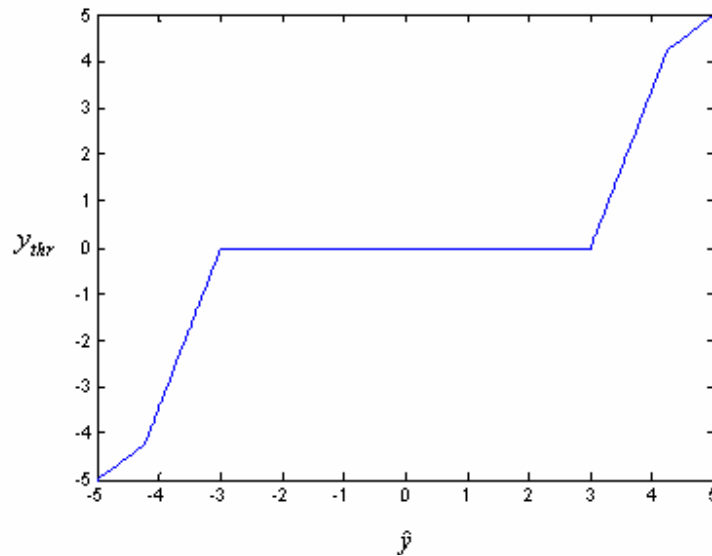


Figura 4. 8: Limiar Semi-Suave.

### 4.3.8 Limiar Sigmoidal (Sigmoid - Thresholding)

Dado  $\lambda > 0$ , considere a função  $y_{thr} = y_{thr}(\hat{y})$  definida pela equação (4.10), o limiar Sigmoidal atenua os coeficientes que estão abaixo do limiar usando a função sigmóide apresentada na equação (4.11). O parâmetro  $\gamma$  controla a inclinação da sigmóide (DUARTE, 2005):

$$y_{thr} = \begin{cases} \hat{y} & \text{se } |\hat{y}| > \lambda \\ \hat{y}|\text{sigmoide}(\hat{y})| & \text{se } |\hat{y}| \leq \lambda \end{cases} \quad (4.10)$$

sendo,

$$\text{sigmoide}(x) = \frac{1 - e(-\lambda x)}{1 + e(-\lambda x)} \quad (4.11)$$

A Figura 4.9 apresenta o gráfico da função definida pela equação (4.10).

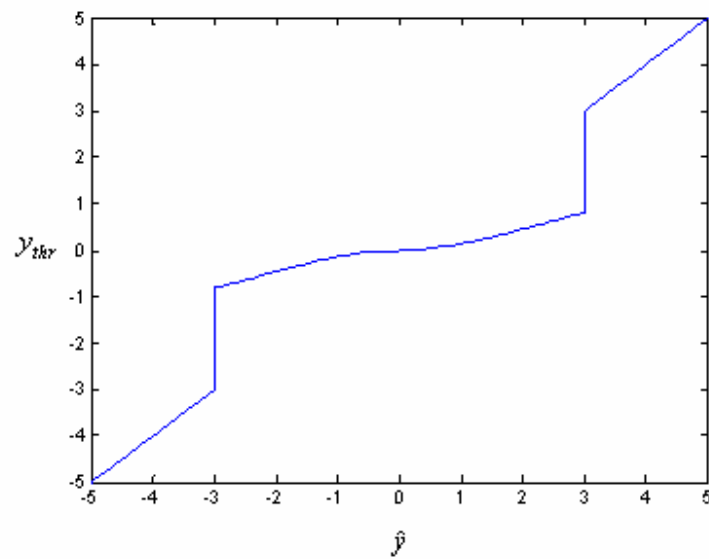


Figura 4. 9: Limiar Sigmoidal.

Dentre os diversos métodos de limiar propostos, o limiar rígido é o mais indicado para casos que têm a compressão como principal objetivo, justamente pela característica de manter os coeficientes no domínio transformado ou trocá-los por zero, visto que os coeficientes eliminados são insignificantes para a representação do sinal. Já os demais limiares são mais indicados para casos de redução de ruído pela característica de atenuar os coeficientes. Porém deve-se levar em conta que não existe uma aplicação específica para a escolha de qual limiar deve ser usado, essa escolha depende do tipo de sinal a ser analisado e dos objetivos a serem alcançados, sendo estes analisados através de inúmeros testes. Em aplicações em imagens devem ser analisados a diminuição de dados e a qualidade visual da imagem.

#### 4.4 Atuação de um Método de Limiar numa Imagem

Após ser decomposta pela transformada wavelet bidimensional, qualquer operação que se faça numa imagem deve considerar suas linhas e suas colunas, ou seja, sendo a imagem representada por uma matriz do tipo  $M \times N$ , existirão  $M$  e  $N$  sinais unidimensionais (STOLLNITS; DEROSE; SALESIN, 1996).

Assim, um método de limiar deve ser aplicado nas linhas e nas colunas da matriz que representa a imagem, de forma separada. Para cada linha, independente do método aplicado, um valor de limiar  $\lambda$  deve ser calculado.

Os métodos que zeram os coeficientes com valores absolutos menores que  $\lambda$  são os mais indicados para compressão de imagens, ou de sinais em geral, enquanto que os métodos que apenas atenuam esses coeficientes são mais indicados para redução de ruído (DUARTE, 2005).

A Figura 4.10 apresenta um diagrama ilustrativo do sistema de redução de ruído ou compressão por limiar no domínio wavelet.

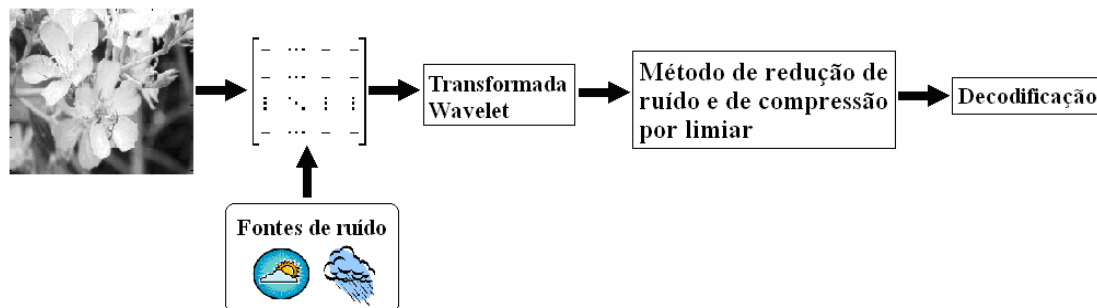


Figura 4. 10: Diagrama ilustrativo do sistema de redução de ruído ou compressão por limiar no domínio wavelet.

No capítulo seguinte será apresentado um novo sistema de codificação de imagens comprimidas usando a transformada wavelet e uma interface gráfica para compressão de imagens, com os respectivos objetivos de diminuir o espaço de armazenamento de imagens comprimidas e de facilitar o acesso a um software de compressão de imagens.

## CAPÍTULO 5

# SISTEMA DE ARMAZENAMENTO DE IMAGENS E UMA INTERFACE GRÁFICA PARA O PROCESSAMENTO

Uma das principais questões que envolvem a compressão de imagens é o armazenamento da imagem comprimida, para que realmente haja economia de espaço computacional, pois se uma imagem possui uma representação matricial de dimensão  $M \times N$ , sua versão comprimida também terá uma representação da mesma dimensão. Como a intensidade luminosa dos pixels da imagem original e dos pixels da imagem comprimida é praticamente a mesma, o espaço computacional ocupado pelas duas imagens será praticamente o mesmo. Assim, não faz sentido fazer a compressão de uma imagem sendo que sua versão comprimida ocupa o mesmo espaço.

A compressão é feita através do uso de transformada, a compressão de imagens usando a transformada wavelet não é uma tarefa difícil, pois o algoritmo da transformada wavelet bidimensional é de fácil implementação computacional (NIEVERGELT, 1999), e os métodos de limiar mais recomendados para compressão também são computacionalmente simples (DONOHO; JOHNSTNE, 1994).

Quando um algoritmo de compressão baseado na transformada wavelet atua numa imagem, ele atua na versão transformada dessa imagem, reduzindo a correlação existente entre os pixels (SILVA et al., 2008). Desta forma, para futura reconstrução, não há necessidade de armazenar todos os coeficientes da imagem, mas apenas os coeficientes significativos.

Por esse motivo, o primeiro objetivo desse trabalho é a implementação de um algoritmo que faça a compressão da imagem decomposta pela Transformada Wavelet, que identifique os coeficientes que não foram eliminados pelo método de limiar escolhido, e os armazene, com informações suficientes para que um sistema de descompressão seja capaz

de reconstruir a imagem usando apenas estes coeficientes. Assim, se torna possível à compressão de uma imagem, por exemplo, seu envio para outros computadores, usando o mínimo espaço computacional.

Outra questão importante no processamento de imagens é a criação de pacotes que permitam fazer o tratamento da imagem sem a necessidade de se trabalhar com códigos fontes, evitando que apenas usuários que têm domínio de linguagem de programação tenham acesso. Ou seja, a criação de pacotes que permitam o acesso ao processamento de imagens a qualquer tipo de usuário.

Por esta razão, neste trabalho, é apresentada uma interface de processamento de imagens, com o objetivo de compressão. Nesta interface, é possível ao usuário escolher uma imagem, o algoritmo de limiar que usará para fazer sua compressão ou eliminação de ruído, fazer comparações entre a imagem original e a imagem comprimida e fazer comparações entre os diversos métodos de limiar.

Nas duas seções que seguem serão apresentados, respectivamente, o sistema de armazenamento proposto e a interface de processamento de imagens.

## **5.1 Sistema de Armazenamento de Imagens através da Transformada Wavelet**

Quando a Transformada Wavelet é aplicada a uma imagem, ela a decompõe em vários níveis de resolução, devido à Análise de Multirresolução (SILVA et al., 2007). Após a decomposição completa da imagem, sua energia está concentrada em poucos coeficientes wavelet, ou seja, a imagem pode ser reconstruída usando apenas estes coeficientes, praticamente sem perda de informações.

Os métodos de limiar atuam eliminando os coeficientes que são redundantes, ou seja, aqueles cuja energia representa uma porção muito pequena da energia da imagem (SOARES et al., 2007).

O algoritmo proposto neste trabalho cria um vetor apenas para o armazenamento dos coeficientes significativos para a reconstrução da imagem. Através desse vetor, é possível reconstruir a imagem original. Além do valor de cada coeficiente não eliminado pelo método de limiar escolhido, é necessário que este vetor contenha também informações como a dimensão da imagem e sua posição na imagem original.

### 5.1.1 Codificação da Imagem Comprimida

A Transformada Wavelet, usada nesse trabalho, é a Transformada Wavelet Discreta Bidimensional e, a escala de resolução usada nas imagens é a escala diádica, isto é, cada espaço de resolução  $V_j$  é amostrado na escala  $2^j$  (GOMES; VELHO; GOLDSTEIN, 1997).

As imagens processadas por este método são imagens de dimensão quadrada de potência de 2, ou seja, imagens do tipo  $N \times N$ , sendo  $N=2^j$ , sendo  $j$  o número máximo de níveis de resolução no qual a Transformada Wavelet decomporá a imagem. Logo, a primeira componente desse vetor deverá ser a dimensão da matriz que representa a imagem. A partir da segunda componente, os coeficientes já podem ser armazenados, sendo que cada coeficiente deverá trazer consigo a informação em qual linha e qual coluna ele está localizado. A seguir, é feita uma ilustração do funcionamento desse sistema de armazenamento para uma matriz quadrada de ordem 4.

Seja  $A$ , a matriz original,  $\hat{A}$  sua versão decomposta pela Transformada Wavelet Discreta (TWD) e  $A_L$  a matriz após aplicação do limiar escolhido.

$$A = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \rightarrow TWD \rightarrow \hat{A} = \begin{bmatrix} \bullet & \circ & \bullet & \circ \\ \bullet & \circ & \bullet & \circ \\ \bullet & \circ & \circ & \circ \\ \bullet & \circ & \circ & \circ \end{bmatrix}$$

Em  $\hat{A}$ , os coeficientes representados por  $\bullet$  são coeficientes significantes, enquanto que os que estão representados por  $\circ$  podem ser eliminados.

Supondo que o método de limiar aplicado para a compressão seja o Limiar Rígido, então a matriz  $A_L$  resultante é a seguinte:

$$A_L = \begin{bmatrix} \bullet & 0 & \bullet & 0 \\ \bullet & 0 & \bullet & 0 \\ \bullet & 0 & 0 & 0 \\ \bullet & 0 & 0 & 0 \end{bmatrix}$$

A matriz  $A_L$  é uma matriz esparsa com poucos coeficientes, por isso só há a necessidade de armazenar os coeficientes não-nulos. Assim, o vetor que representaria a matriz comprimida, observando as linhas da matriz  $A_L$  seria:

$$V\_cod = [4 \bullet 1 1 \bullet 1 3 \bullet 2 1 \bullet 2 3 \bullet 3 1 \bullet 4 1]$$

sendo que o número 4 na primeira posição indica a dimensão da matriz. Após o primeiro elemento  $\bullet$ , os números 1 e 1 indicam que este coeficiente se encontra na linha 1 e na coluna 1 e, assim por diante. Assim, para uma matriz transformada e comprimida de ordem  $N$ , o vetor  $V\_cod$  seria escrito da seguinte forma:

$$V\_cod = [N \bullet i \ j \bullet \dots]$$

sendo  $i$  a linha e  $j$  a coluna em que se encontra o elemento  $\bullet$ .

Portanto, não será necessária uma matriz quadrada para representar a matriz comprimida, apenas uma matriz linha.

O vetor  $V\_cod$  pode ser gerado na mesma rotina que faz a eliminação dos coeficientes que estão abaixo do limiar, pois nesta rotina há a identificação dos coeficientes que estão acima do limiar. Logo, o custo computacional para gerar este vetor será desprezível.

O vetor  $V\_cod$  é a versão compactada da matriz no domínio wavelet. O pseudocódigo da rotina para sua construção é apresentado a seguir:

**Procedimento** compactação de matriz( $m$ : matriz –  $N \times N$ ,  $\lambda$ ,  $V\_cod$ )

*DWT(matriz)*

$V\_cod \leftarrow []$

$V\_cod(1) \leftarrow N$

Para  $i$  de 1 até  $N$

Para  $j$  de 1 até  $N$

Se  $|m(i,j)| > \lambda$  então

$V\_cod \leftarrow [V\_cod \ m(i,j) \ i \ j]$

Senão

*Limiar( $m(i,j)$ )*

Fim se

Fim para

Fim para

**Fim do procedimento**

### 5.1.2 Decodificação da Imagem Comprimida

O procedimento de reconstrução da matriz, a partir do vetor  $V\_cod$ , também é muito simples. A ordem da matriz é o valor do primeiro elemento do vetor. Portanto, o primeiro passo é a criação de uma matriz quadrada nula cuja ordem seja igual ao primeiro elemento do vetor  $V\_cod$ .

A partir do segundo elemento, cada três elementos representam, respectivamente, o valor do coeficiente preservado na matriz transformada, sua linha e sua coluna de localização, então bastam usar esses três valores para localizar e posicionar o coeficiente no seu lugar na nova matriz nula criada inicialmente. Após a localização de todos os elementos da matriz, a transformada wavelet discreta inversa (TWDI) é aplicada nesta matriz.

O pseudocódigo da rotina de reconstrução da matriz transformada a partir do vetor  $V\_cod$  é apresentado a seguir:

**Procedimento** decodificação ( $V\_cod$  - vetor  $[1 \times K]$ )

$N = V\_cod(1)$

$M = \text{matriz\_nula}(N \times N)$

$X = V\_cod[2 \dots K]$

$m = \text{comprimento}(X)$

Para  $i$  de 1, passo 3, até  $m$

$M(X(i+1), X(i+2)) = X(i)$

Fim para

$mr = \text{idwt}(M)$

**fim do procedimento**

A forma de armazenamento usada no algoritmo proposto pode ser simplificada se a linguagem de programação ou pacote usado tiver a função que permite criar uma variável que contenha apenas os elementos não-nulos de uma matriz esparsa.

No *MATLAB*, por exemplo, a função *sparse* realiza essa operação e a função *full* faz a operação inversa (MATSUMOTO, 2002). Porém, a maioria das linguagens de programação não tem uma função que realiza esta tarefa.

## 5.2 Interfaces Gráficas para o Sistema de Armazenamento de Imagens Comprimidas Proposto

O objetivo de apresentar um modelo de interface gráfica para o processamento de imagens é devido a facilidade para a criação e implementações de demonstrações interativas do trabalho, disponibilizando um aplicativo para o uso de outras pessoas com as demonstrações do método desenvolvido e as análises realizadas.

Esta interface permite ao usuário:

- 1- selecionar uma imagem para processamento;

- 2- escolher a escala de cores, no *MATLAB*, desejada;
- 3- escolher o método de limiar que será usado para comprimi-la;
- 4- comparar as imagens, original e processada;
- 5- verificar o número de coeficientes eliminados na compressão;
- 6- verificar a taxa de compressão, isto é, a relação entre os coeficientes eliminados/total de coeficientes;
- 7- verificar o erro entre as duas imagens, original e processada;
- 8- fazer análises estatísticas entre as duas imagens.

A taxa de compressão  $C$ , em porcentagem, é a razão entre o número de coeficientes eliminados e o número total de coeficientes, definida por:

$$C = \frac{C_{el}}{T} \times 100 \quad (5.1)$$

sendo  $C_{el}$  o número de coeficientes eliminados e  $T$ , o número total de coeficientes.

O erro, em porcentagem, é calculado da seguinte forma:

$$Erro = \frac{|\text{traço}(A_p) - \text{traço}(A)|}{|\text{traço}(A)|} \quad (5.2)$$

sendo  $A$ , a matriz que representa a imagem original,  $A_p$ , a matriz que representa a imagem processada, e a função  $\text{traço}(\bullet)$  faz a soma dos elementos da diagonal principal da matriz  $\bullet$ .

Calculando o erro conforme a equação (5.2), pelo menos um valor em cada linha de cada uma das matrizes consideradas será levado em conta, aumentando a confiabilidade no erro calculado.

Outra forma de se calcular o erro seria usando a equação (5.3). Porém, neste caso, apenas um valor de cada matriz é tomado para compará-las.

$$Erro = \frac{|\max(A_p) - \max(A)|}{|\max(A)|} \quad (5.3)$$

sendo  $A$ , a matriz que representa a imagem original,  $A_p$ , a matriz que representa a imagem processada e  $\max(\bullet)$  é a função que seleciona o valor máximo de uma matriz.

### 5.2.1 Apresentação das Telas

O Sistema de Armazenamento de Imagens Comprimidas Através da Transformada Wavelet e um Método de Limiar é acessado ao acionar o comando *sistema* no prompt do *MATLAB* que por sua vez apresenta a Tela 1.



Figura 5. 1: Tela 1

Ao clicar no botão “Processar”, será disponibilizado a Tela 2, onde é apresentado o conjunto de imagens e as opções de cores a serem selecionadas pelo usuário para análise, conforme apresenta a Figura 5.2.

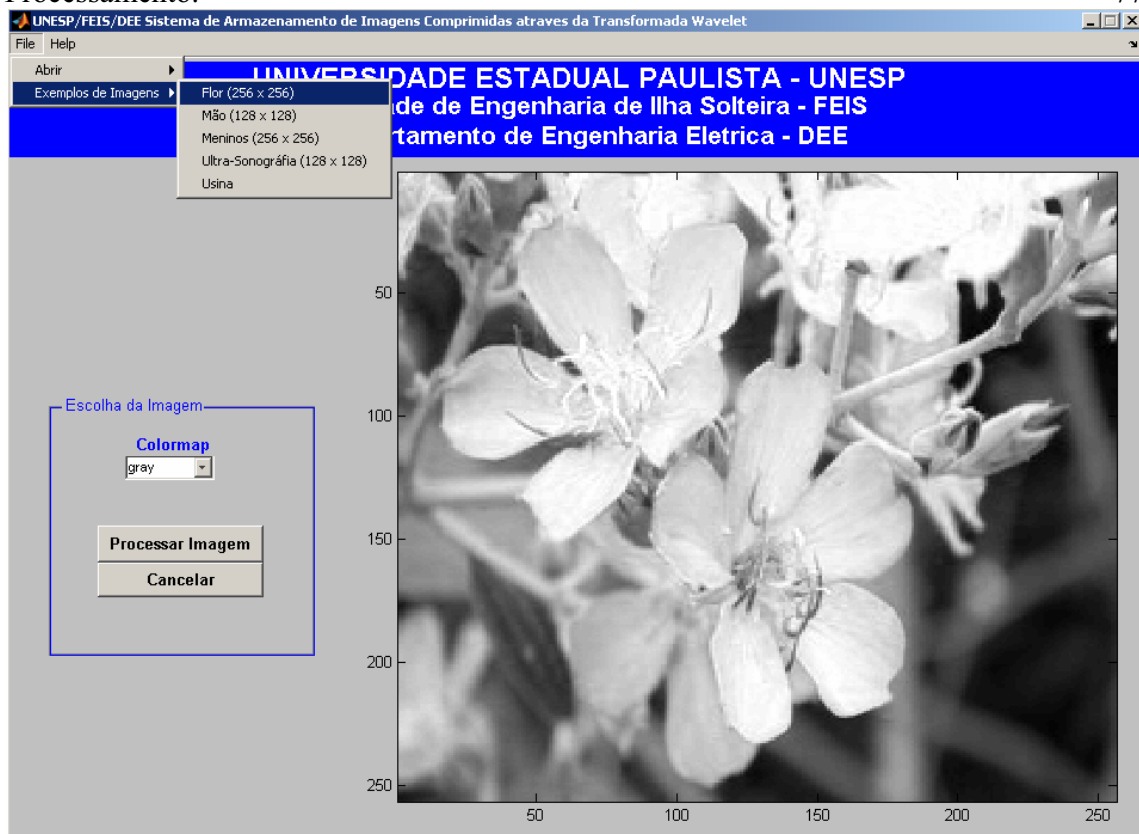


Figura 5. 2: Tela 2

O usuário poderá entrar com qualquer imagem, além das que são fornecidas pelo sistema. Na Tela 2, o botão “Processar Imagem” disponibiliza a Tela 3 como ilustra a Figura 5.3, e o botão “Cancelar”, anula o processamento.

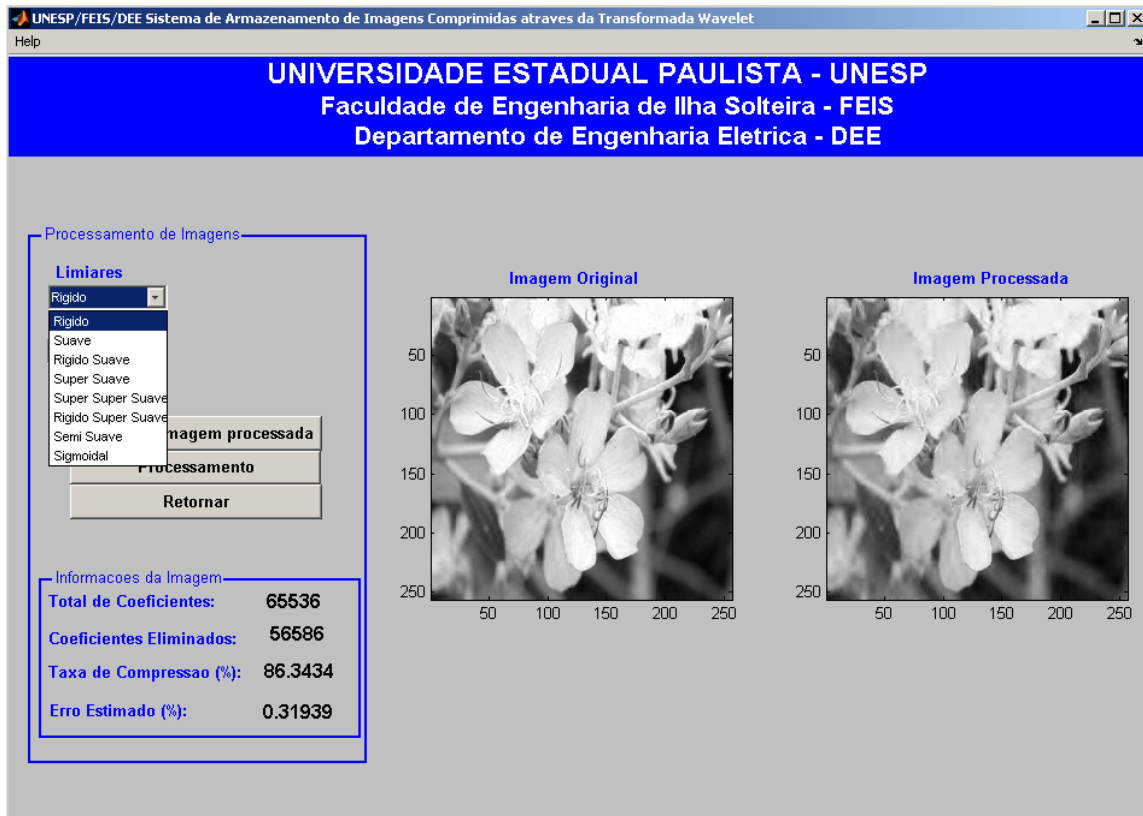


Figura 5. 3: Tela 3.

Na Tela 3 é possível escolher que método de limiar será usado para compressão da imagem ou para redução de ruído quando for o caso, além disso é disponibilizado as principais informações do processamento. O botão “Visualizar Imagem Processada”, fornece a Tela 4 como ilustra a Figura 5.4 e o botão “Analisar Dados” fornece a tela Tela 5 como ilustra a Figura 5.5.

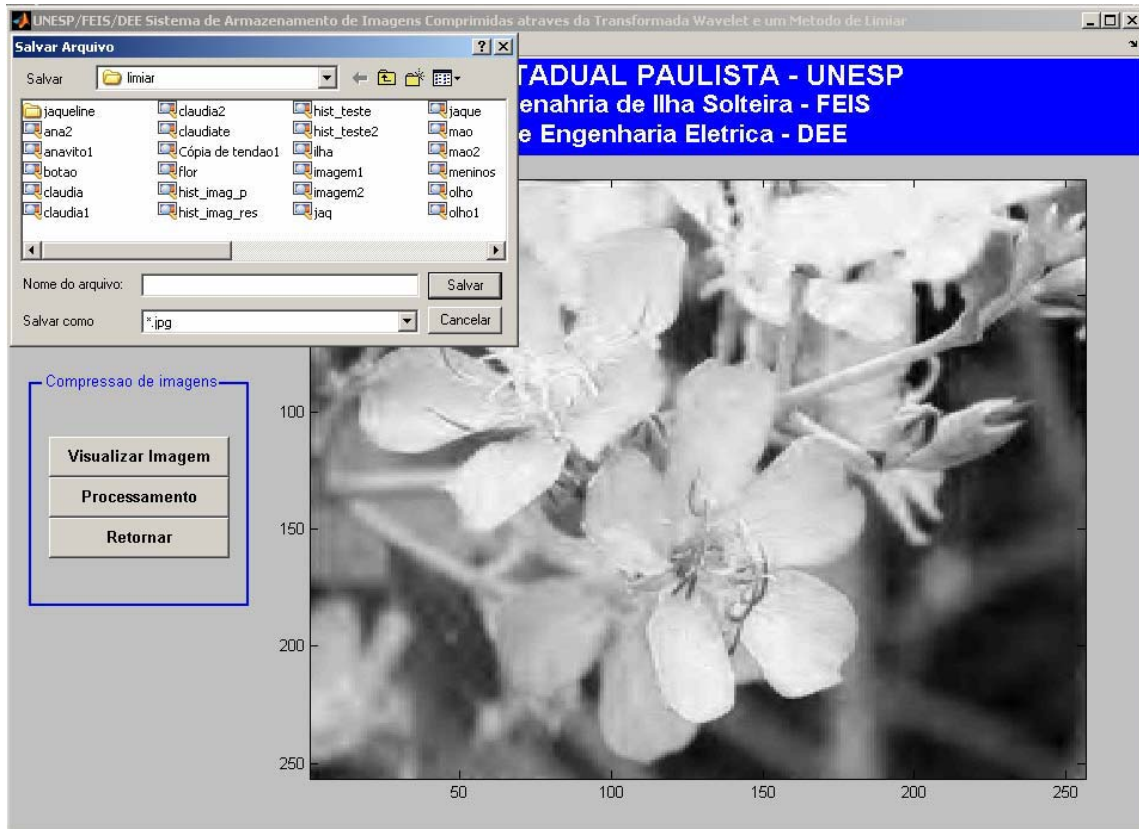


Figura 5. 4: Tela 4.

Ao clicar no botão “Visualizar Imagem”, a imagem processada na Tela 3 será apresentada de modo mais amplo, além disso, é disponibilizada a opção de salvar a imagem processado através do menu *File/Salva/Imagem...* A imagem será salva com extensão JPEG e posteriormente poderá ser utilizada. Assim como na Tela 3 o Botão “Analisar dados” fornece a Tela 5 como ilustra a Figura 5.5.

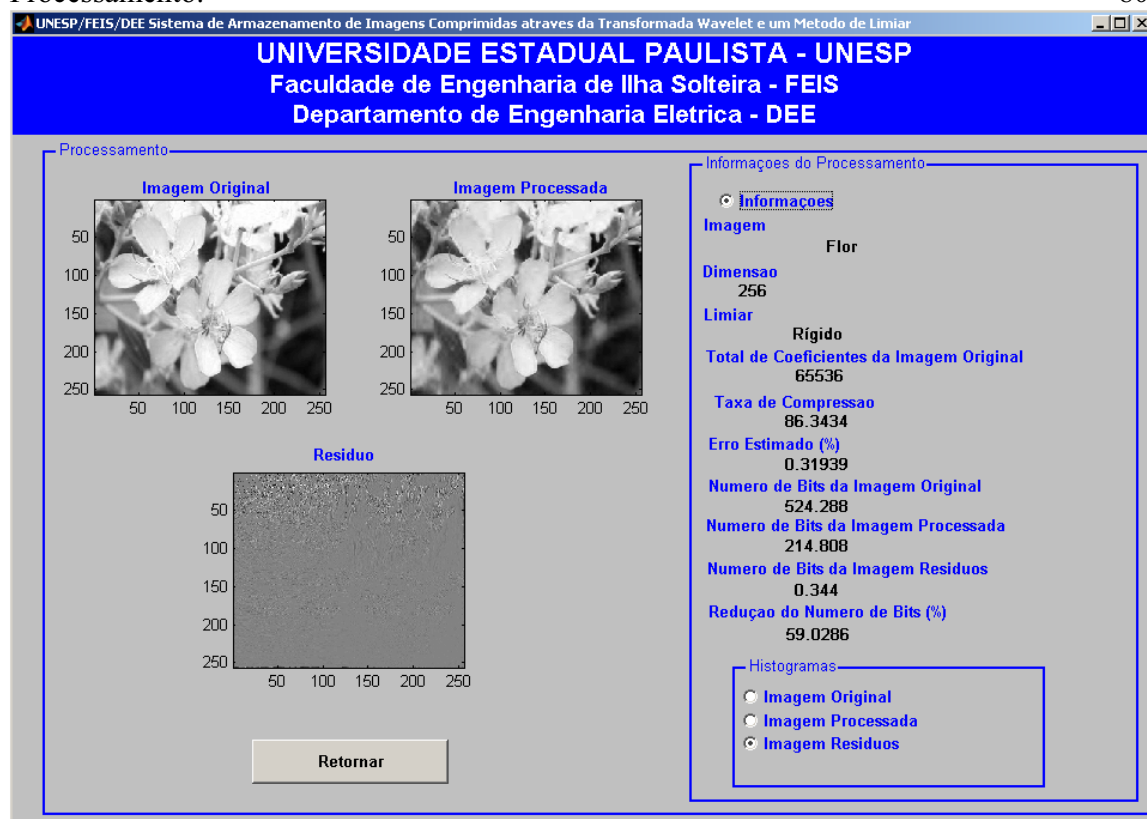


Figura 5. 5: Tela 5.

Na Tela 5 é possível fazer uma análise estatística do processamento da imagem selecionada, como por exemplo, o número de bits utilizado em seu armazenamento computacional ao selecionar a opção “Informações”. Também é disponibilizado os histogramas das imagens original, processada ou dos resíduos, ao selecionar uma das opções disponíveis. O botão “Retornar”, faz com que o sistema retorne a Tela 4.

# CAPÍTULO 6

## RESULTADOS E CONSIDERAÇÕES

Para testar o sistema de armazenamento de imagens comprimidas proposto na Seção 5.1 foi usada a Transformada Wavelet Bidimensional, implementada usando a função wavelet de Daubechies 6 (db6), processada através do *MATLAB*. Além disso, foi utilizado o Limiar Rígido para melhor eficiência do sistema.

A função wavelet de Daubechies 6 (db6), foi usada porque apresenta um número de momentos nulos suficiente para evidenciar detalhes da imagem e para propiciar uma boa reconstrução da mesma (DAUBECHIES, 1992).

Para os testes, foi usado um conjunto de cinco imagens, sendo três (Flor, Meninos e Usina), de 256x256 pixels e duas, (Mão e Ultra-sonografia), de 128x128 pixels. Conforme se pode observar nas Tabelas 6.1 a 6.9, para o *MATLAB*, imagens monocromáticas (imagens em escala de cinza) de mesma dimensão ocupam o mesmo espaço na memória do computador, com relação ao seu tamanho original.

Para verificar a eficiência do método de armazenamento, os únicos valores que devem ser observados são: o tamanho original da imagem, em kBytes, o tamanho, em kBytes, do vetor de codificação da mesma e a redução, em porcentagem, quando se passa da matriz original para o seu vetor de codificação.

### **6.1 Utilização do Limiar Rígido no Sistema Armazenamento de Imagens Comprimidas**

A escolha em utilizar o Limiar Rígido, foi feita devido ao comportamento de tal limiar, que quando aplicado em uma imagem o valor de seus coeficientes não sofre variações irreversíveis, ou seja, os valores dos coeficientes são zerados ou mantidos.

As imagens usadas para o teste utilizando o *Limiar Rígido* e suas versões processadas são apresentadas nas Figuras 6.1 a 6.5, respectivamente, conforme a Tabela 6.1.

Tabela 6. 1: Imagens comprimidas e decodificadas usando o sistema de armazenamento proposto com o limiar rígido

<b>Imagens</b>	<b>Taxa de Compressão (%)</b>	<b>Erro (%)</b>	<b>Tamanho original (kBytes)</b>	<b>Tamanho codificado (kBytes)</b>	<b>Redução (%)</b>
Flor	86,34	0,32	524,29	214,81	59,03
Mão	86,85	1,78	131,07	51,73	60,53
Meninos	92,24	0,004	524,29	122,00	76,73
Ultra-sonografia	86,69	0,16	131,07	52,35	60,05
Usina	89,38	0,25	524,29	166,95	68,15

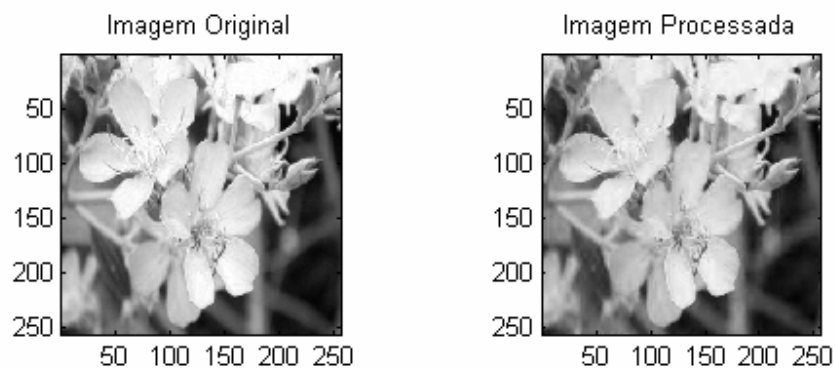


Figura 6. 1: Flor, original e com compressão de 86,34% utilizando o Limiar Rígido.

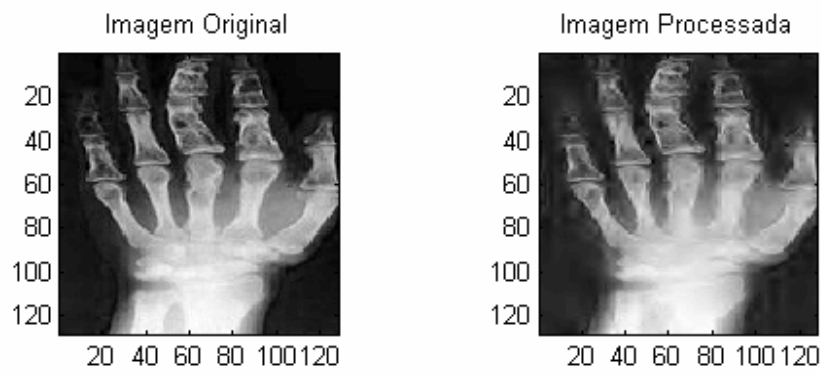


Figura 6. 2: Mão, original e com compressão de 86,85% utilizando o Limiar Rígido.

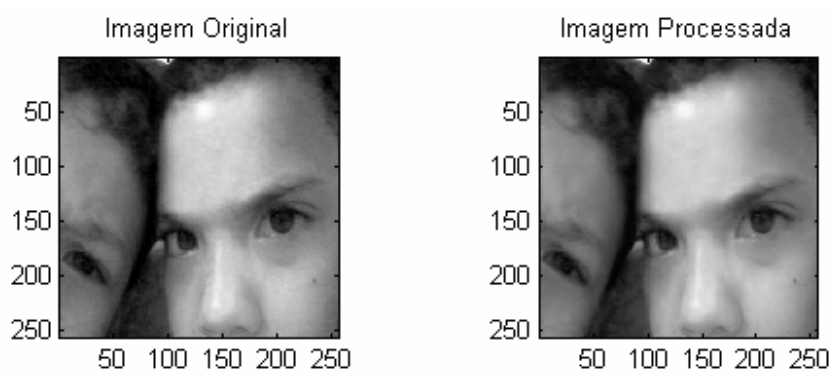


Figura 6. 3: Meninos, original e com compressão de 92,24% utilizando o Limiar Rígido.

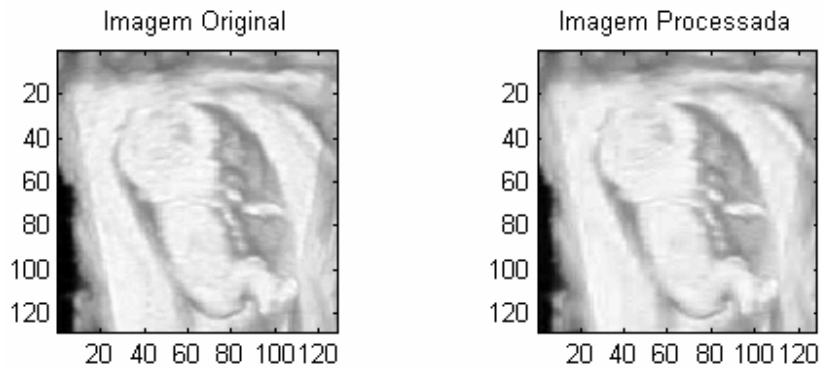


Figura 6. 4: Ultra-sonografia, original e com compressão de 86,69% utilizando o Limiar Rígido.

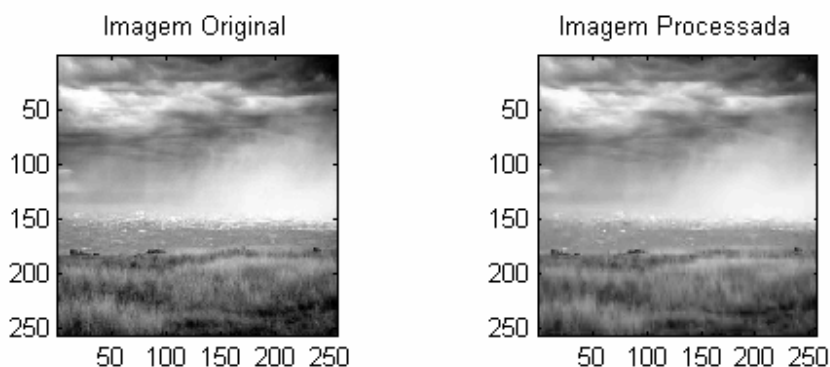


Figura 6. 5: Usina, original e com compressão de 89,38% utilizando o Limiar Rígido.

Conforme se pode observar na Tabela 6.1, o conjunto de imagens originais tem um total de 1835,01 kBytes. Após o processamento ele passa a ter um total de 607,84 kBytes.

Representando uma redução total de 64,89% do espaço necessário para o seu armazenamento. Além disso, é possível observar que a taxa de compressão de todas as imagens é superior a 86% e o erro apresentado é inferior a 2%.

O sistema de armazenamento não se limita a trabalhar apenas com o limiar rígido, porém ele é o mais adequado para a compressão de imagens. Foram testados também outros métodos apresentados na seção 3.3, porém esses métodos atenuam os valores dos coeficientes da imagem, sendo mais indicados a redução de ruído. Ressaltando, que os demais limiares testados a seguir, não se limitam apenas a redução de ruído, mas também a compressão.

## 6.2 Utilização dos Limiares Suave e Semi-Suave no Sistema de Armazenamento de Imagens Comprimidas

As imagens usadas para os testes utilizando o demais limiar (Suave e Semi-Suave), e suas versões processadas são apresentadas nas Figuras 6.6 a 6.15, respectivamente, conforme as Tabelas 6.2 e 6.3. Os dados obtidos no processamento desses dois métodos são praticamente os mesmos obtidos no processamento utilizando o método rígido, apresentado na Tabela 6.1, exceto o erro apresentado, é o que se pode observar nas tabelas que seguem.

### 6.2.1 Limiar Suave

As imagens usadas para o teste utilizando o método suave e suas versões processadas são apresentadas nas Figuras 6.6 a 6.10, respectivamente, conforme a Tabela 6.2.

Tabela 6. 2: Imagens processadas usando o sistema de armazenamento proposto com o limiar suave.

<b>Imagens</b>	<b>Taxa de Compressão (%)</b>	<b>Erro (%)</b>	<b>Tamanho original (kBytes)</b>	<b>Tamanho codificado (kBytes)</b>	<b>Redução (%)</b>
Flor	86,34	0,5674	524,29	214,81	59,03
Mão	86,85	1,4601	131,07	51,73	60,53
Meninos	92,24	0,046	524,29	122,00	76,73
Ultra-sonografia	86,69	0,1976	131,07	52,35	60,05
Usina	89,38	0,0320	524,29	166,95	68,15

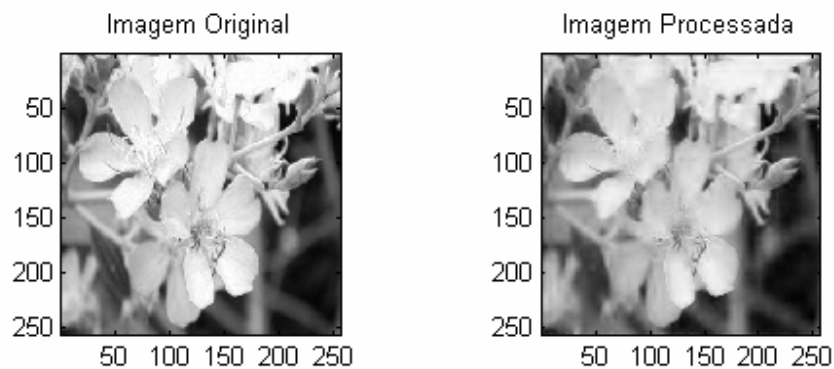


Figura 6. 6: Flor, original e com compressão de 86,34% utilizando o Limiar Suave

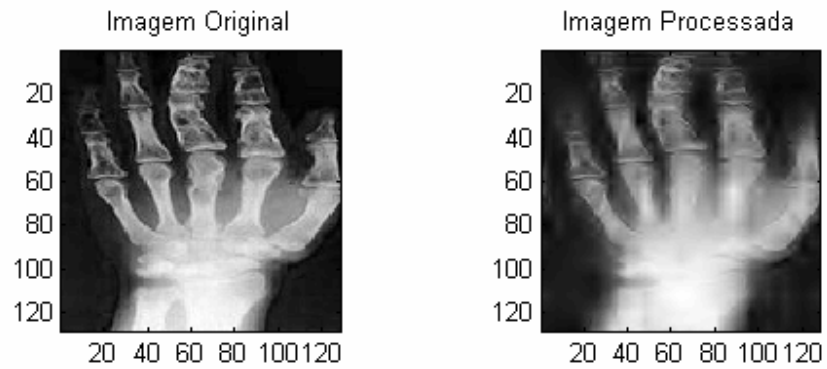


Figura 6. 7: Mão, original e com compressão de 86,85% utilizando o Limiar Suave.

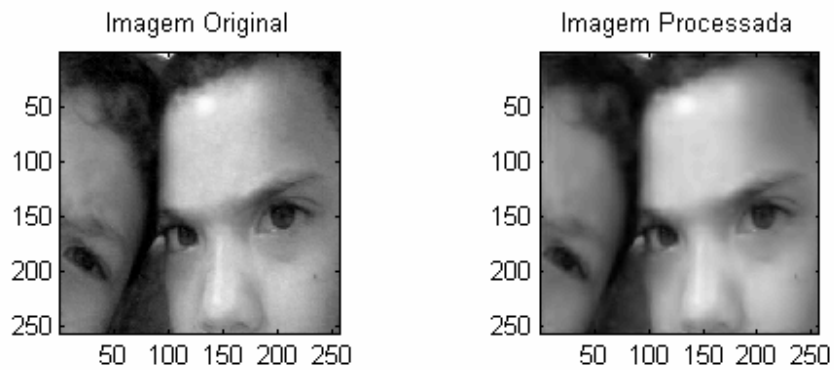


Figura 6. 8: Meninos, original e com compressão de 92,24% utilizando o Limiar Suave.

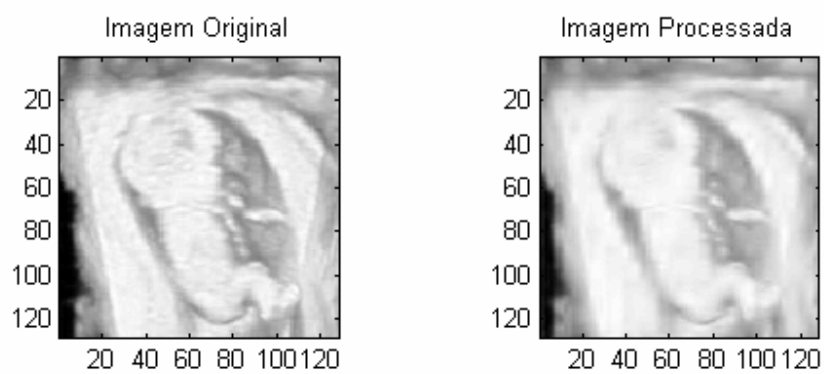


Figura 6. 9: Ultra-sonografia, original e com compressão de 86,69% utilizando o Limiar Suave.

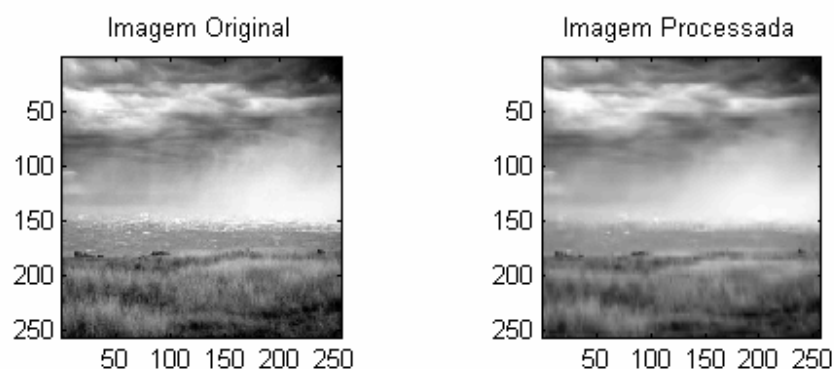


Figura 6. 10: Usina, original e com compressão de 89,38% utilizando o Limiar Suave.

No processamento do conjunto de imagens utilizando o limiar suave, pode-se observar na Tabela 6.2, que o erro apresentado, neste caso, nas imagens Flor, Meninos e Ultra-sonografia foram maiores, enquanto que as imagens, Mão e Usina foram menores que os apresentados no limiar rígido.

### 6.2.2 Limiar Semi Suave

As imagens usadas para o teste utilizando o método de *Limiar Semi Suave* e suas versões processadas são apresentadas nas Figuras 6.11 a 6.15, respectivamente, conforme a Tabela 6.3.

Tabela 6. 3: Imagens processadas usando o sistema de armazenamento proposto com o limiar semi-suave.

<b>Imagens</b>	<b>Taxa de Compressão (%)</b>	<b>Erro (%)</b>	<b>Tamanho original (kBytes)</b>	<b>Tamanho codificado (kBytes)</b>	<b>Redução (%)</b>
Flor	86,34	0,2303	524,29	214,81	59,03
Mão	86,85	0,939	131,07	51,73	60,53
Meninos	92,24	0,0263	524,29	122,00	76,73
Ultra-sonografia	86,69	0,2006	131,07	52,35	60,05
Usina	89,38	0,1244	524,29	166,95	68,15

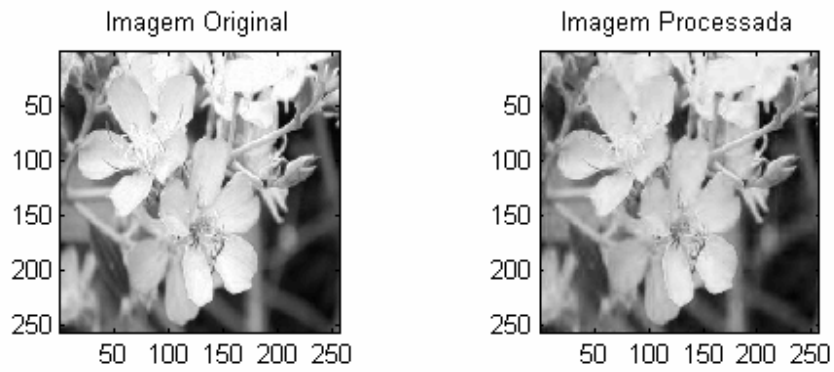


Figura 6. 11: Flor, original e com compressão de 86,34% utilizando o Limiar Semi Suave.

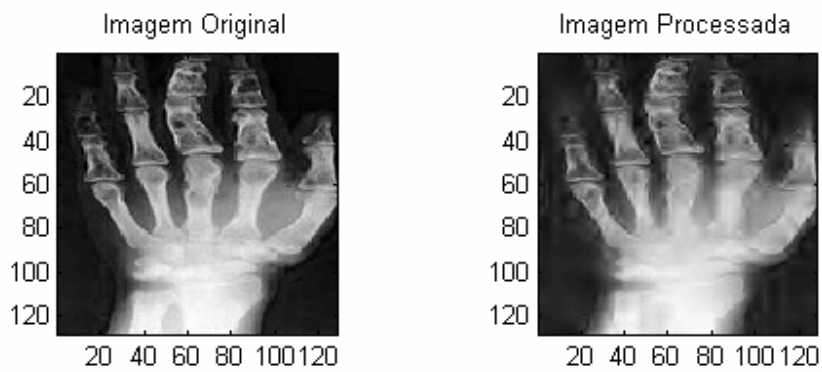


Figura 6. 12: Mão, original e com compressão de 60,53% utilizando o Limiar Semi Suave.

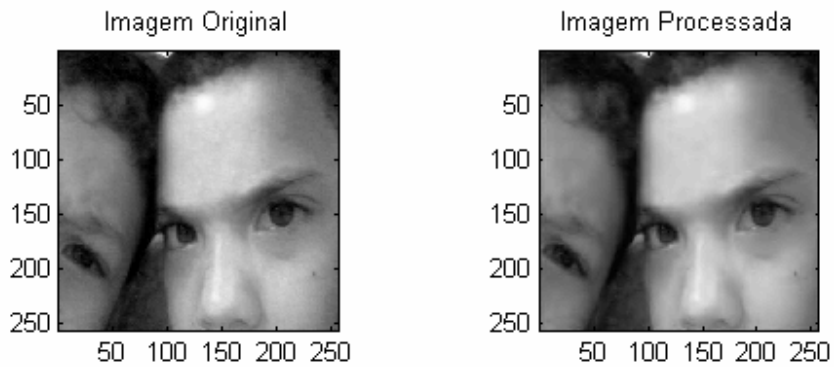


Figura 6. 13: Meninos, original e com compressão de 76,73% utilizando o Limiar Semi Suave.

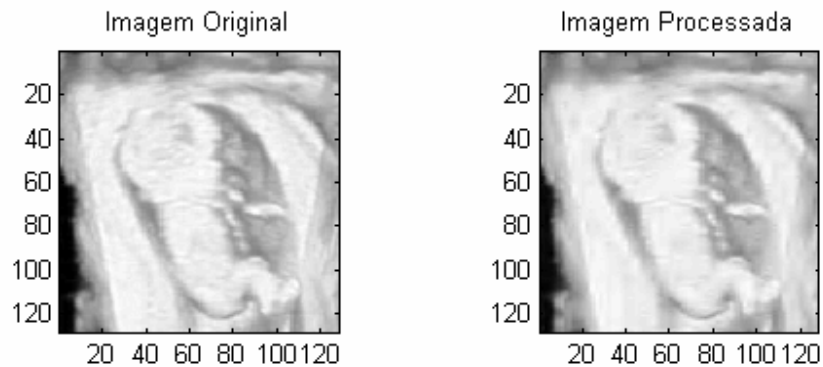


Figura 6. 14: Ultra-sonografia, original e com compressão de 60,05% utilizando o Limiar Semi Suave.

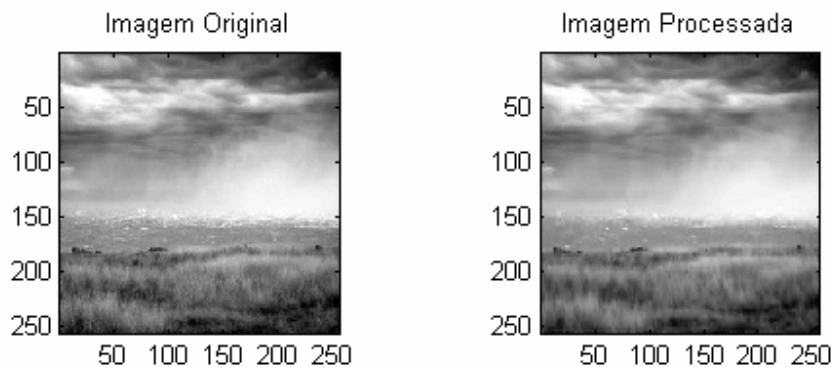


Figura 6. 15: Usina, original e com compressão de 68,15% utilizando o Limiar Semi Suave.

Após o processamento do conjunto de imagens utilizando o limiar semi-suave, pode-se observar na comparação entre as Tabela 6.1, 6.2 e 6.3, que os dados são praticamente os mesmos exceto com relação aos erros apresentados.

Comparando os dados obtidos na Tabela 6.3 com os dados obtidos na Tabela 6.1, os erros apresentados nas imagens Flor, Mão e Usina são menores, enquanto que nas imagens Menino e Ultra-sonografia passam a serem maiores, porém na comparação dos dados obtidos na Tabela 6.3 com os dados obtidos na Tabela 6.2, os erros apresentados nas imagens Flor, Ultra-sonografia e Usina são menores, enquanto que nas imagens Mão e Menino são maiores.

### 6.3 Considerações Sobre os Métodos Utilizados na Compressão

A Tabela 6.4 apresenta de modo geral, a relação de imagens processadas e o erro obtido de cada limiar.

Tabela 6. 4: Erro estimado para os métodos utilizados para a compressão usando o sistema de armazenamento proposto.

<b>Limiares</b>	<b>Flor</b>	<b>Mão</b>	<b>Meninos</b>	<b>Ultra-sonografia</b>	<b>Usina</b>
Rígido	0,32	1,78	0,004	0,16	0,25
Suave	0,5674	1,4601	0,0460	0,1973	0,0320
Semi Suave	0,2303	0,939	0,0263	0,2006	0,1244

Pode-se observar na Tabela 6.4, considerando a média entre os erros obtidos no conjunto de imagens, que o melhor desempenho foi do Limiar Semi Suave.

Os métodos que zeram os coeficientes que estão abaixo do limiar são úteis para o caso de compressão, mesmo aqueles que atenuam os coeficientes que estão acima do limiar, que é o caso dos métodos suave e semi-suave. Já os métodos que apenas atenuam os coeficientes que estão abaixo do limiar são úteis para redução de ruído, neste caso são os métodos rígido suave, super suave, super super suave, rígido super suave e sigmoidal.

Os métodos apropriados apenas para a redução de ruído também foram colocados na interface, pois a interface pode ser utilizada para a compressão e redução de ruído. A escolha de qual método de limiar deve ser utilizado, como dito anteriormente, dependerá tanto dos dados estatísticos quanto das análises visuais das imagens a serem processadas.

O sistema de armazenamento proposto foi implementado, usando a linguagem de programação do *MATLAB*, num conjunto de cinco imagens e apresentou uma redução superior a 50% do espaço de armazenamento necessário para qualquer uma das imagens. Sendo que a reconstrução dessas imagens não foi comprometida, conforme se pode observar nas Figuras 6.1 a 6.15 e nas Tabelas 6.1 a 6.4.

É importante ressaltar que as funções implementadas para o sistema de armazenamento proposto podem ser utilizadas sem a necessidade da interface gráfica, permitindo maior agilidade caso seja necessário utilizar especificamente uma das funções, ou caso deseje efetuar alguma alteração ou melhoria no código (SILVA et al, 2008).

Portanto, pode-se concluir que o sistema de armazenamento de imagens comprimidas proposto é eficiente e possui como pontos fortes a facilidade de

implementação e resultados expressivos na área de processamento de imagens (SILVA et al, 2008).

## 6.4 Considerações Finais e Sugestões de Trabalhos Futuros

Neste trabalho foi proposto um sistema de armazenamento de imagens, que por sua vez são comprimidas usando a transformada wavelet com o objetivo de reduzir o espaço de memória computacional utilizado para armazenamento de imagens. Este sistema armazena apenas os coeficientes que não são eliminados na compressão da imagem, suas posições e a dimensão desta imagem. Tornando assim bem menor o espaço utilizado para armazenamento da imagem. De acordo com o conjunto de imagens usado para testes, este sistema reduz em mais de 50% o espaço de armazenamento da imagem.

Embora o sistema armazenamento tenha sido implementado no *MATLAB*, o mesmo pode ser feito em qualquer linguagem de programação e para compressão de imagens usando qualquer tipo de técnicas, não apenas as técnicas baseadas na transformada wavelet.

Também foi proposta uma interface gráfica para o processamento de imagens com o objetivo de proporcionar o acesso ao processamento de imagens a todo tipo de usuário de computadores, principalmente aqueles que não têm domínio de nenhuma linguagem de programação.

A interface proposta permite ao usuário selecionar uma imagem para processamento, escolher a escala de cores usadas para representá-la, escolher um método de limiar para compressão ou redução de ruído, comparar visualmente as imagens, original e processada, e analisar os dados referentes à operação realizada e salvar a imagem processada como se pode observar nas Figuras 5.1 a 5.5 no capítulo 5.

Como sugestão de futuros trabalhos está à questão do processamento de imagens coloridas, pois a interface proposta processa apenas imagens monocromáticas. E, a possibilidade de uma interface onde se possam fazer recortes em imagens para fazer processamento em regiões específicas da imagem.

# REFERÊNCIAS

ALBUQUERQUE, M. P. **Processamento de imagens: métodos e análises**. Rio de Janeiro: FACET, 2001.

ALBUQUERQUE, M. P.; CANER, E. S.; MELO, A. G. **Análise de imagem e visão computacional**. Rio de Janeiro: CBPF, 2004.

ALCOCER, P. R. C.; FURUIR, S. S.; ARIE, S.; MELO, C. P. A DICOM graphic user interface for PC's: towards a hierarchical system for dynamic digital angiographic image storage and visualization. **RBE – Revista Brasileira de Engenharia. Caderno de Engenharia Biomédica**, Rio de Janeiro, v.12, p.191-201, 1996.

ASHI, R. Y.; AMERI, A. **Introduction to graphical user interface (GUI) MATLAB 6.5**. (s.l.): UAE University – College of Engineering Electrical Engineering Department IEE.(1999?). 35 p. Disponível em: <<http://ewh.ieee.org/r8/uae/GUI.pdf>>. Acessado em: 05 set. 2007.

BOTELHO, S. S. C. **Processamento digital de imagens: conceitos básicos de imagens**. (s.l.:s.n.),2005.Disponível em: <[http://www.ee.furg.br/~silviacb/DIP/Conceitos\\_Basicos.html](http://www.ee.furg.br/~silviacb/DIP/Conceitos_Basicos.html)>. Acessado em: 27 set. 2007.

BRUCE, A.; GAO, H. Y. Understanding waveshrink: variance and bias estimation **Biometrika**, London, v.83, n. 4, p.727-745, 1996.

CENÁRIO atual. Rio de Janeiro: Pontifica Universidade Católica. Centro Digital de Referência. 2004. p. 1-15. Disponível em: < [http://www.maxwell.lambda.ele.puc-rio.br/cgi-bin/PRG\\_0599.EXE/7626\\_3.PDF?NrOcoSis=22053&CdLinPrg=pt](http://www.maxwell.lambda.ele.puc-rio.br/cgi-bin/PRG_0599.EXE/7626_3.PDF?NrOcoSis=22053&CdLinPrg=pt) >. Acessado em: 27 out. 2007.

COIFMAN, R. Adapted multiresolution analysis, computation, signal processing and operator theory. In: CONGRESS OF MATHEMATICIANS, 1990, Kioto. **Proceedings...** Kioto:(s.n.), 1990. p. 879-887.

COIFMAN, R.; MEYER Y.; QUAKE, S.; WICKERHAUSER, M. V. Signal processing and compression with wavelet packets. In: MEYER, Y.; ROQUESED, S. **Progress in wavelet analysis and applications**. (s.l.:s.n), 1993. p. 77-93.

COIFMAN, R.; WICKERHAUSER M. V. Wavelets and adapted waveform analysis: a toolkit for signal processing and numerical analysis. In: SYMPOSIUM IN APPLIED MATHEMATICS, 47, 1993, (s.l.). **Proceedings...** (s.l.:s.n.) 1993. p. 119-153.

DAUBECHIES, I. Orthonormal bases of compactly supported wavelets. **Communications on Pure and Applied Mathematics**, New York, v. 41, p. 909-996, 1998.

DAUBECHIES, I. The wavelet transform, time frequency localization and signal analysis. **IEEE Transactions on Information Theory**, New York, v.36, p.961-1005, 1990.

DAUBECHIES, I. **Ten lectures on wavelets**. Philadelphia: SIAM Books, 1992.

DINIZ, P. S. R.; SILVA, E. A. B.; LIMA, S. N. **Processamento digital de sinais: projeto e análise de sistemas**. Porto Alegre: Bookman, 2004.

DONOHO, D. L.; JOHNSTONE, I. M. Ideal spatial adaptation via wavelet shrinkage **Biométrica**, London, v.81, n.3, 425-455, 1994.

DUARTE, M. A. **Redução de ruído em sinais de voz no domínio Wavelet**. 2005. 105 f. Tese (Doutorado)-Faculdade de Engenharia, Universidade Estadual Paulista, Ilha Solteira, 2005.

FALCÃO, A. X.; LEITE, N. J. **Fundamentos de processamento de imagem digital**. Campinas: Unicamp, 1998.

FARIA, R. R. A. **Wavelets e as artes multirresolucionárias**. (s.l.:s.n.,1997?). Disponível em: <<http://www.lsi.usp.br/~regis/wlets.html>>. Acessado em: 13 set. 2007.

FONSECA, M. S. **Um estudo sobre a influência das famílias wavelets na compressão de imagens**. 2004. 152 f. Dissertação (Mestrado) – Universidade Federal de Fluminense, Niterói, 2004

GOMES, J.; VELHO, L. ; GOLDSTEIN, S. **Wavelets: teoria, software e aplicações**. São Paulo: IMPA, 1997.

GONZÁLES, R. C.; WOODS, R. E. **Processamento de imagens digitais**. São Paulo: Edgard Blucher, 2000.

GONZÁLES, R. C.; WOODS, R. E. **Digital image processing**. (s.l.): Publishing Company, 1992.

HAAR, A. Zur theorie der orthogonalen functionen-systeme. **Mathematische Annalen**, Berlin, v.69, p. 331-371, 1910.

KOSLOSKI, R. P. **Wavelet: um estudo e aplicação à detecção e caracterização de distúrbios em sistema de energia elétrica**. 2000. 84 f. Dissertação (Mestrado) – Faculdade de Engenharia, Universidade Estadual Paulista, Ilha Solteira, 2000.

MALLAT, S. A theory for multiresolution representation signal decomposition: the Wavelet representation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, New York, v.11, n.7, p. 674-693, 1989a.

MALLAT, S. Multiresolution approximation and wavelets. **Transactions American Mathematical Society**, Providence, v.315, p. 69-88, 1989b.

MATSUMOTO, E. Y. **MATLAB 6.5**: fundamentos de programação. São Paulo: Érica, 2002.

MEYER, Y. **Wavelets**: algorithms and applications. Philadelphia: SIAM Books, 1993.

MISITI, M.; MISITI, Y.; OPPENHEIM, G.; POGGI, J. **Matlab**: wavelet toolbox user's guide. Natick: Math Works, 1996.

NIEVERGELT, Y. **Wavelets made easy**. Boston: Birkhäuser, 1999.

OLIVEIRA, H. M. **Análise de sinais para engenheiros**: uma abordagem via wavelets. Rio de Janeiro: BRASPORT Livros e Multimídia, 2007.

OPPENHEIM, A. V.; SCHAFER, R. W.; BUCK, J. R. **Discrete-time signal processing**. New Jersey: Prentice Hall, 1998.

OSBORNE, D.; ROGERS, D.; MAZUMDAR, J.; COUTTS, R. ;ABBOT, D. An overview of wavelets for image processing for wireless applications. **Proceedings of SPIE**, Aurlia v. 4935, p. 427-435, 2002.

PORTILA J. ;SOMONCELLI, E. P. Image denoising via adjustment of wavelet coefficient magnitude correlation. In: PROCEEDINGS OF THE IEEE INT'L CONF ON IMAGE PROCESSING, 3, 2000, Vancouver. **Proceedings...** Vancouver: IEEE,2000.p.277-280,2000.

REIS, A. J. R.; SILVA, A. P. A. Aplicação da transformada wavelet discreta na previsão de carga a curto prazo via redes neurais. **Sba- Controle & Automação Sociedade Brasileira de Automática**, Campinas, v.15, n.1, p. 101-108, 2004

RIOUL, O.; VETTERLI, M. Wavelets and signal processing. **IEEE Signal Processing Magazine**, New York, v.8, n.4, p. 14-38, 1991.

SANTIAGO, D. F. A. ; PEDERIVA, R. Monitoramento de falhas em máquinas rotativas usando a transformada wavelet de packet. In: CONGRESSO DE PESQUISA E INOVAÇÃO DA REDE NORTE NORDESTE DE EDUCAÇÃO TECNOLÓGICA, 1, 2006, Natal. **Processing...** Natal: (s.n.), 2006. p.1-14. Disponível em: <[http://www.redenet.edu.br/publicacoes/arquivos/20070502\\_154055\\_IND%20053.pdf](http://www.redenet.edu.br/publicacoes/arquivos/20070502_154055_IND%20053.pdf)> Acessado em: 15 abr 2008.

SANTOS, E. M. S.; ALBUQUERQUE, M. P.; ALBUQUERQUE, M. P.; MELLO, A. R. G.; CANER, E. S. **Introdução a interface gráfica no MATLAB**. Rio de Janeiro: LPDSI-CBPF, 2004.

SILVA, A.V.; EYNG, J. Wavelets e wavelet de packets. In: SEMINÁRIO VISÃO COMPUTACIONAL, 2, 2000, Florianópolis. **Visão computacional...** Florianópolis: PPGCC - INE - UFSC, 2000. Disponível em: <<http://www.inf.ufsc.br/~visao/2000/Wavelets/index.html>>. Acessado em: 18 out. 2007.

SILVA, J. F.; DUARTE, M. A.; VILLARREAL, F.; SOARES, W.C. Processamento digital de imagens e a transformada Wavelet. In: BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND THEIR APPLICATIONS, 6, 2007, São José do Rio Preto. **Proceedings...** São José do Rio Preto: IBLCE/UNESP, 2007, p.1213-1219.

SILVA, J. F.; DUARTE, M. A.; VILLARREAL, F. Sistema de armazenamento de imagem comprimidas através da transformada Wavelet e um método de limiar. In: BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND THEIR APPLICATIONS, 7, 2008, Presidente Prudente. **Proceeding...** Presidente Prudente: FCT/UNESP, 2008. p.922-927.

SOARES, W.C.; VIEIRA FILHO, J.; DUARTE, M. A. Q.; VILLARREAL, F. Análise de métodos de redução de ruído em sinais de voz no domínio Wavelet. In: BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND THEIR APPLICATIONS, 6, 2007, São José do Rio Preto. **Proceedings...** São José do Rio Preto: IBLCE/UNESP, 2007. p.918-923.

STOLLNITS, E. J.; DEROSE, T. D.; SALESIN, D. H. **Wavelets for computer graphics theory and applications**. San Francisco: Morgan Kaufmann Publishers, 1996.

STORM, H. **Noise reduction of speech signals with Wavelets**. Gothenburg: Department of Mathematics, Chalmers University of Technology and Gothenburg University, 1998. (Tech. rep. N.1998-02/ISSN 0347-2809).

YOUNG, R. K. **Wavelet theory and its applications**. Boston: Kluwer Academic Publisher, 1995.

# APÊNDICE A

## INTERFACES GRÁFICAS NO MATLAB

A partir dos anos 90, as maiorias dos softwares começaram a utilizar interfaces gráficas. O uso de interface gráfica com o usuário é uma forma de interação entre uma pessoa e um computador para troca de informações (MATSUMOTO, 2002).

Os programas com interfaces gráficas usam uma programação por evento, ou seja, passam a maioria do tempo sem fazer nada a espera que o usuário se comunique com o computador utilizando um dispositivo de entrada, como o teclado ou o mouse, exibindo nas interfaces textos ou imagens.

Uma interface gráfica pode ser usada para criar ferramentas muito eficientes no *MATLAB*, pois o *MATLAB* inclui facilidades para a criação de demonstrações iterativas e implementações de trabalhos através de uma ferramenta chamada *guide* (*GUI Development Environment*). A Figura 5.1 ilustra a janela que o *MATLAB* exibe quando a função *guide* é chamado (SANTOS, et al. 2004, ASHI e AMERI, 1999).

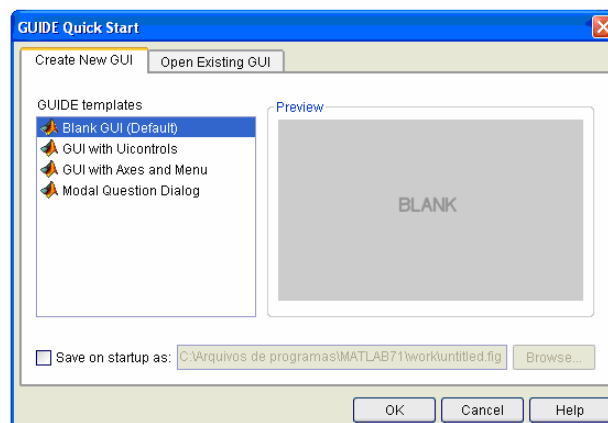


Figura A.1: Toolbox GUI.

A função *guide* armazena a interface criada em dois arquivos, que são gerados quando salva e interpreta a interface gráfica. O arquivo-FIG (Interface) contém a descrição completa sobre as figuras presentes na interface, é no arquivo.fig que se realiza as modificações desejadas na aparência da interface. O arquivo-M (arquivo fonte) contém o código que controla a interface, cabe ao programador inserir os comandos necessários para seu funcionamento. A criação de interface gráfica no *MATLAB* divide-se em três passos principais:

1) criar objetos necessários e colocá-los na janela. Por exemplo, botões, textos e imagens. Cada objeto tem uma série de propriedades, como, tamanho, cor de fundo, tipo de texto e posição. A manipulação dessas propriedades é feita através dos comandos *get* e *set*. O *MATLAB* associa a cada objeto um número que o identifica, chamado de *handle*.

2) manipular as propriedades dos objetos. Para isso é preciso manipular uma estrutura de dados, onde todas as informações são mantidas, acedendo aos campos do *handle*.

3) escrever as funções que devem ser acionadas em resposta aos eventos. Neste caso, usa-se a função *callback*, que é associada ao evento para realizar determinada tarefa.

## A.1 Principais Estilos de Controle

O *MATLAB* aceita dez tipos de controle chamados de *uicontrol* (permite o controle de interface com o usuário) (SANTOS, et al. 2004; ASHI e AMERI, 1999)

1. Botões simples (*pushbuttons*): são objetos de tela que normalmente contêm um rótulo de texto. Os botões simples costumam ser usados para executar uma ação definida pela *callback* string do botão.

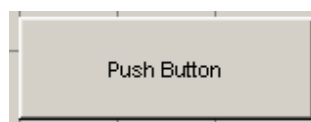


Figura A.2: Botão Simples.

2. Botões de chave (*toggle buttons*): A diferença desse tipo de botão é o fato de alternarem entre dois estados (ON/OFF), para cima e para baixo quando pressionados. A propriedade “value” de um botão de chave é fixada como igual ao valor especificado pela

propriedade “Max” quando o botão está para baixo, e fixado como igual ao valor da propriedade “Min” quando o botão está para cima.

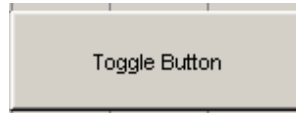


Figura A.3: Botão de Chave.

3. Botões de rádio (*radio button*): São botões que contêm um rótulo em pequenos círculos, usados geralmente quando se deseja selecionar uma opção de um grupo de opções. Quando selecionado, o círculo é preenchido e a propriedade “value” é fixada como valor especificado pela propriedade “Max”, que por padrão é 1. Caso o botão não seja selecionado a propriedade “value” é fixada como valor especificado pela propriedade “Min”, que por padrão é 0. Neste tipo de botão, apenas uma opção deve ser ativada de cada vez, o que garante a exclusividade da callback do botão, é a propriedade “value”, por exemplo:

`set(handles.radiobutton1,'value',1)` a callback acionada.

`set(handles.radiobutton1,'value',0)` a callback não será acionada.



Figura A.4: Botão de radio.

4. Caixas de controle (*check box*): Consiste em botões similares aos botões-chave, com um rótulo e uma pequena caixa quadrada. Costumam ser usadas para indicar o estado de uma opção ou atributo. Normalmente, são objetos independentes de forma que e a propriedade “value” é fixada da mesma forma que nos demais botões.

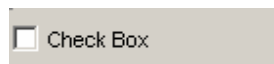


Figura A.5: Caixa de controle.

5. Caixas de edição (*edit text*): As caixas de texto editáveis exibem texto em caixas, de maneira que o usuário possa modificar ou substituir a string do texto dinamicamente, podendo conter uma ou mais linhas de texto. As modificações se tornam disponíveis na

propriedade “string” do unicontrol. A cada linha na caixa de texto são fixados valores das propriedades “Max” e “Min” em valores tais que  $Max - Min > 1$ . As strings de múltiplas linhas podem ser especificadas como vetores celulares de texto ou vetor de caracteres.



Figura A.6: Caixa de edição.

6. Caixas de texto estáticas (*static text*): Diferente das caixas de texto editáveis, as caixas de texto estáticas não permitem mudanças dinâmicas na string do texto. As mudanças só poderão ser feitas na propriedade “string” do unicontrol. As caixas de texto estáticas são utilizadas para mostrar texto nas interfaces e elas não possuem *callback*.



Figura A.7: Caixa de texto estática.

7. Sliders: São controles deslizantes que permitem mudar um valor numérico. Consistem em três partes distintas: o canal (área retangular, representando o intervalo de valores de objetos válidos); o indicador dentro do canal representando o valor atual do slider e setas em cada extremidade do canal.

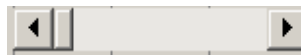


Figura A.8: Slide.

8. Frame: São regiões retangulares opacas, sombreadas com uma margem, usadas para organizar as informações. As frames devem ser definidas antes que outros objetos sejam colocados dentro delas.



Figura A.9: Frame.

9. Caixa de listagem (*listbox*): As caixas de listagem são semelhantes as caixas de texto de múltiplas linhas, porém elas permitem que o usuário selecione um item individual, cada item contém um índice. A propriedade “value” é atualizada conforme o índice do item selecionado.



Figura A.10: Caixa de listagem.

10. Menu popup: Este menu apresenta uma lista de escolhas mutuamente exclusivas ao usuário. Cada escolha contém um índice, a propriedade “value” é atualizada conforme o índice do item selecionado.

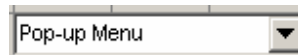


Figura A.11: Menu popup.