

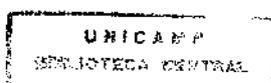
Universidade Estadual de Campinas - UNICAMP
Instituto de Matemática, Estatística e Computação Científica - IMECC

**Novas Medidas de Distorção Estatísticas
para Avaliar a Qualidade de Algoritmos
de Compressão de Imagens**

Lucila Helena E. Allan

Prof. Dr. Arnaldo José de Hoyos Guevara
Orientador

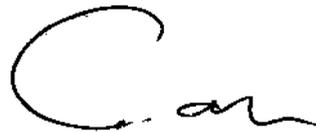
Fev/97



Novas Medidas de Distorção Estatísticas para Avaliar a Qualidade de Algoritmos de Compressão de Imagens

Este exemplar corresponde a redação final da
tese devidamente corrigida e defendida por
Lucila Helena E. Allan e aprovada pela
Comissão Julgadora.

Campinas, 25 de Fevereiro de 1997



Prof. Dr. Arnaldo José de Hoyos Guevara
Orientador

Dissertação apresentada ao Instituto de
Matemática, Estatística e Computação
Científica, UNICAMP, como requisito parcial
para obtenção do Título de MESTRE em
Estatística.

UNICAMP

UNIDADE	BC		
N.º CHAMADA:			
V.			
TOMBO	30557		
PROC.	28A197		
C	<input type="checkbox"/>	D	<input checked="" type="checkbox"/>
PREÇO	R\$ 11,00		
DATA	24/05/97		
N.º GPO			

CM-00 098452-1

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP

Allan, Lucila Helena E.

ALS2n Novas medidas de distorção estatísticas para avaliar a qualidade de algoritmos de compressão de imagens / Lucila Helena E. Allan -- Campinas, [S.P. :s.n.], 1997.

Orientador : Arnolde José de Hoyos Guevara

Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Processamento de imagens - Métodos estatísticos. 2. Compressão de dados (Computação). I. Hoyos Guevara, Arnolde J. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Dissertação de Mestrado defendida e aprovada em 25 de fevereiro de 1997

pela Banca Examinadora composta pelos Profs. Drs.



Prof (a). Dr (a). REGINA CÉLIA DE CARVALHO PINTO MORAN



Prof (a). Dr (a). LEE LUAN LING



Prof (a). Dr (a). ARNOLDO JOSÉ DE HOYOS GUEVARA

AGRADECIMENTOS

Aos alunos de Graduação da Estatística por sua colaboração e paciência durante minha pesquisa de opinião. Em especial ao César por sua ajuda na seleção das pessoas.

Aos alunos do Mestrado em Estatística por sua amizade e apoio durante o trabalho.

Aos funcionários da biblioteca do IMECC, em especial à Júlia, por sua amizade e colaboração nas busca de referências bibliográficas.

A minha colega Magda, do curso de Mestrado em Matemática Aplicada, por sua paciência e ajuda nos estudos sobre wavelets.

Ao meu noivo Yuri por seu apoio e ajuda em todos os momentos da elaboração deste trabalho.

A meu pai, Prof. Nelo, por seu apoio e ajuda na escrita de alguns capítulos deste trabalho.

Ao Prof. Arnoldo por sua orientação e incentivo.

Ao CNPq por seu apoio financeiro durante estes anos de pesquisa.

A Jesus pois esteve comigo em todos os momentos e, sem Ele, nada disto seria possível.

RESUMO

O objetivo principal deste trabalho é procurar novas medidas de distorção estatísticas que permitam avaliar objetivamente algoritmos de compressão de imagens. Procuramos as novas medidas utilizando técnicas estatísticas, relacionadas principalmente as áreas de Análise Multivariada e Teoria da Informação. Para avaliar seu desempenho, aplicamos estas medidas em imagens de qualidade visual distintas produzidas por quatro tipos diferentes de algoritmos de compressão utilizados atualmente. A fim de comparar a eficiência destas medidas foi feita uma pesquisa de opinião visando avaliar a qualidade subjetiva das várias imagens, que serviria de padrão comparativo de qualidade. Como resultado deste trabalho, selecionamos duas nova medidas que apresentaram bons resultados em relação aos critérios clássicos relacionados ao Erro Quadrático Médio.

ABSTRACT

The main purpose of this work is to develop statistical models that could help to measure distortions, so that one could compare objectively the quality of image compression algorithms. The statistical models were developed using methods that come from Multivariate Statistical Analysis and Information Theory. For developing and testing our image quality models we used four different and very popular image compression algorithms. Moreover for efficiency comparison purposes we obtained, by using a poll, a mean subjective measure of image quality. As a result of our work we obtained two new statistical image quality models that in general behave better than the classical and commercial ones based on a global measure of the Mean Square Error.

CONTEÚDO

CAPÍTULO 1 Introdução: a Importância da Compressão de Imagens

1.1	Processamento Digital de Imagens	14
1.2	Texturas e Bordas	16
1.3	Compressão de Imagens e Medidas de Distorção	20
1.4	Descrição do Trabalho	22

CAPÍTULO 2 Técnicas de Compressão Analisadas

2.1	Introdução - Imagens como Representações de Processos Estocásticos	24
2.2	Técnicas Básicas que Utilizam Transformações no Domínio das Frequências ..	25
2.2.1	Representações Ortonormais para $L^2(\mathbb{R})$	25
2.2.2	Compressão Através da Transformada de Fourier	30
2.2.2.1	Algoritmos	31
2.2.2.2	Exemplos	32
2.2.3	Compressão Através da DCT	35
2.2.3.1	Método 1	37
2.2.3.2	Método JPEG	38
2.2.3.3	Algoritmos	39
2.2.3.4	Exemplos	42
2.3	Técnica que Utiliza Wavelets	44
2.3.1	Algoritmo	67
2.3.2	Exemplos	67

2.4	Técnicas que Utilizam Fractais	69
2.4.1	Algoritmos	76
2.4.2	Exemplos	82

CAPÍTULO 3 Avaliando a Qualidade de Compressão

3.1	Introdução	85
3.2	Medidas Clássicas: PNSR	86
3.3	Medidas de Distância e Informação	87
3.3.1	Abordagem por Blocos	88
3.3.2	Abordagem Geral	90
3.3.3	Medidas de Informação	90
3.4	Medidas para Análise Descritiva	92
3.5	Medidas para Avaliar Bordas	94
3.6	Estudos sobre Texturas	99
3.6.1	Modelo Estatístico para a Análise de Textura das Imagens	100
3.6.2	Método de Estudo das Texturas Neste Trabalho	106

CAPÍTULO 4 Análise dos Resultados

4.1	Introdução	107
4.2	Abordagem por Blocos e Geral	107
4.3	Qualidade Subjetiva das Imagens	125
4.4	Estudo da Qualidade de Bordas e Texturas	134
4.5	Comparação dos Nossos Resultados com o PNSR	138

CAPÍTULO 5 Conclusão

5.1 Discussão dos Resultados Obtidos	140
5.2 Considerações Finais	142

REFERÊNCIAS BIBLIOGRÁFICAS

APÊNDICE A Resultados da Pesquisa de Opinião

LISTA DE TABELAS

TABELA 3.3.1.1	Medidas de Distância Utilizadas	89
TABELA 3.3.2.1	Medidas de Informação Utilizadas	92
TABELA 4.2.1	Notação Utilizada para Referenciar as Medidas de Distância	108
TABELA 4.2.2	Valores de R^2 Parciais das Variáveis que Entraram no Modelo a um Nível de Significância de 15% para Blocos de Tamanho $2^4 \times 2^4$	109
TABELA 4.2.3	Valores de R^2 Parciais das Variáveis que Entraram no Modelo a um Nível de Significância de 15% para Blocos de Tamanho $2^5 \times 2^5$	110
TABELA 4.2.4	Coefficiente de Correlação de Pearson entre as Variáveis Analisadas ...	117
TABELA 4.2.5	Valores de T^2 para a Imagem Original em Blocos de Tamanho $2^5 \times 2^5$ e Ordenados Segundo o Valor Calculado	118
TABELA 4.2.6	Valores de T^2 para a Imagem Após o Filtro Laplaciano em Blocos de Tamanho $2^5 \times 2^5$ e Ordenados Segundo o Valor Calculado	120
TABELA 4.2.7	Valores de R^2 Parciais das Variáveis que Entraram no Modelo a um Nível de Significância de 15% para Blocos de Tamanho $2^5 \times 2^5$ para o Novo Conjunto de Dados	125
TABELA 4.3.1	Imagens Ordenadas pela Média das Notas	127
TABELA 4.3.2	Percentuais de Classificação em Cada Qualidade e Escore Estimado ...	129
TABELA 4.3.3	Escore Predito e o Original Ordenados Segundo o Valor Predito Exceto para as Imagens FRC's	131
TABELA 4.3.4	Resultados da Análise Discriminante	133
TABELA 4.4.1	Qualidade Estimada Através do Método de Análise de Bordas	136
TABELA 4.4.2	Escore Estimado para as Texturas	137
TABELA 4.5.1	Valores do EQM, PNSR e as Duas Medidas de Distorção Encontradas ..	139

LISTA DE FIGURAS

FIGURA 1.1	Amostras de Texturas Artificias	17
FIGURA 1.2	Amostras de Texturas Naturais	17
FIGURA 1.3	"Borda em Rampa"	18
FIGURA 1.4	"Borda de Passo"	19
FIGURA 1.5	"Borda em Teto"	19
FIGURA 1.6	Imagem Original da Lena	23
FIGURA 2.1	Imagem Reconstruída com 20% dos Coeficientes após a Aplicação da FFT na Imagem Original.	33
FIGURA 2.2	Imagem Reconstruída com 10% dos Coeficientes após a Aplicação da FFT na Imagem Original	33
FIGURA 2.3	Imagem Reconstruída com 5% dos Coeficientes após a Aplicação da FFT na Imagem Original	33
FIGURA 2.4	Imagem Reconstruída com 50% dos Coeficientes após a Aplicação da FFT na Imagem Original Dividida em Blocos	34
FIGURA 2.5	Imagem Reconstruída com 40% dos Coeficientes após a Aplicação da FFT na Imagem Original Dividida em Blocos	34
FIGURA 2.6	Imagem Reconstruída com 30% dos Coeficientes após a Aplicação da FFT na Imagem Original Dividida em Blocos	35
FIGURA 2.7	Imagem Reconstruída com 10% dos Coeficientes após a Aplicação da FFT na Imagem Original Dividida em Blocos	35
FIGURA 2.8	Suposição Feita Pelos Métodos de Compressão Envolvendo a DCT e a FFT com Relação à Disposição dos Blocos	37

FIGURA 2.9	Imagem Reconstruída com 20% dos Coeficientes após a Aplicação da DCT na Imagem Original	42
FIGURA 2.10	Imagem Reconstruída com 10% dos Coeficientes após a Aplicação da DCT na Imagem Original	42
FIGURA 2.11	Imagem Reconstruída com 5% dos Coeficientes após a Aplicação da DCT na Imagem Original	43
FIGURA 2.12	Imagem Reconstruída após a Compressão da Imagem Original através do Método JPEG com Fator igual a 95	43
FIGURA 2.13	Imagem Reconstruída após a Compressão da Imagem Original através do Método JPEG com Fator igual a 75	43
FIGURA 2.14	Imagem Reconstruída após a Compressão da Imagem Original através do Método JPEG com Fator igual a 50	43
FIGURA 2.15	Imagem Reconstruída após a Compressão da Imagem Original através do Método JPEG com Fator igual a 35	44
FIGURA 2.16	Decomposição do Sinal f	58
FIGURA 2.17	Reconstrução do Sinal f	61
FIGURA 2.18	Processo de Decomposição de uma Imagem f	64
FIGURA 2.19	Decomposição de uma Imagem f	64
FIGURA 2.20	Reconstrução de uma Imagem f	65
FIGURA 2.21	Imagem Reconstruída Utilizando 20% dos Coeficientes após a Aplicação da Transformada Wavelet na Imagem Original	68
FIGURA 2.22	Imagem Reconstruída Utilizando 10% dos Coeficientes após a Aplicação da Transformada Wavelet na Imagem Original	68
FIGURA 2.23	Imagem Reconstruída Utilizando 5% dos Coeficientes após a Aplicação da Transformada Wavelet na Imagem Original	68
FIGURA 2.24	Triângulo de Sierpinski	70

FIGURA 2.25	Fern de Sierpinski	70
FIGURA 2.26	Operação de Média	78
FIGURA 2.27	Método Quad-tree	79
FIGURA 2.28	Três Posições Canônicas Para Ordenar Imagens Quadradas	80
FIGURA 2.29	Imagem Reconstruída após a Compressão da Imagem Original Utilizando o Método 1 (Fractal) com $e_c=2$	83
FIGURA 2.30	Imagem Reconstruída após a Compressão da Imagem Original Utilizando o Método 1 (Fractal) com $e_c=3$	83
FIGURA 2.31	Imagem Reconstruída após a Compressão da Imagem Original Utilizando o Método 1 (Fractal) com $e_c=4$	84
FIGURA 2.32	Imagem Reconstruída após a Compressão da Imagem Original Utilizando o Método 2 (Fractal) depois de 8 Iterações	84
FIGURA 2.33	Imagem Reconstruída após a Compressão da Imagem Original Utilizando o Método 2 (Fractal) depois de 6 Iterações	84
FIGURA 2.34	Imagem Reconstruída após a Compressão da Imagem Original Utilizando o Método 2 (Fractal) depois de 5 Iterações	84
FIGURA 3.1	Imagem após a Aplicação do Algoritmo de Detecção de Bordas Finas	95
FIGURA 3.2	Imagem após a Aplicação do Algoritmo do Chapéu Mexicano para Detecção de Bordas	97
FIGURA 3.3	Imagem após a Aplicação do Método de Roberts para Detecção de Bordas	99
FIGURA 3.4	Amostras de Texturas da Imagem da Lena	103
FIGURA 3.5	Imagem Classificada após a Primeira Análise	104
FIGURA 3.6	Imagem da Lena após a Aplicação de um Filtro de Alta Ênfase	104
FIGURA 3.7	Imagem Classificada após a Segunda Análise	105
FIGURA 3.8	Imagem Classificada após a Remoção de Ruído	105

LISTA DE QUADROS

QUADRO 4.1	Critério para Transformação do Escore em Uma Medida Qualitativa	128
-------------------	---	-----

LISTA DE GRÁFICOS

GRÁFICO 2.1	Exemplo de Função Escala (ϕ)	53
GRÁFICO 2.2	Transformada de Fourier de $\phi(x)$	53
GRÁFICO 2.3	Função Wavelet $\psi(x)$ Associada a Função Escala do Gráfico 2.1	57
GRÁFICO 2.4	Módulo da Transformada de Fourier de $\psi(x)$	57
GRÁFICO 4.1	Diferenças entre a Imagem Original e a Reconstruída - DCT80	111
GRÁFICO 4.2	Diferenças entre a Imagem Original e a Reconstruída - DCT90	111
GRÁFICO 4.3	Diferenças entre a Imagem Original e a Reconstruída - DCT95	111
GRÁFICO 4.4	Diferenças entre a Imagem Original e a Reconstruída - FFT80	111
GRÁFICO 4.5	Diferenças entre a Imagem Original e a Reconstruída - FFT90	112
GRÁFICO 4.6	Diferenças entre a Imagem Original e a Reconstruída - FFT95	112
GRÁFICO 4.7	Diferenças entre a Imagem Original e a Reconstruída - WAV80	112
GRÁFICO 4.8	Diferenças entre a Imagem Original e a Reconstruída - WAV90	112
GRÁFICO 4.9	Diferenças entre a Imagem Original e a Reconstruída - WAV95	112
GRÁFICO 4.10	Diferenças entre a Imagem Original e a Reconstruída - FTQ50	112
GRÁFICO 4.11	Diferenças entre a Imagem Original e a Reconstruída - FTQ60	113
GRÁFICO 4.12	Diferenças entre a Imagem Original e a Reconstruída - FTQ70	113

GRÁFICO 4.13	Diferenças entre a Imagem Original e a Reconstruída - FTQ90	113
GRÁFICO 4.14	Diferenças entre a Imagem Original e a Reconstruída - FRC8	113
GRÁFICO 4.15	Diferenças entre a Imagem Original e a Reconstruída - FRC6	113
GRÁFICO 4.16	Diferenças entre a Imagem Original e a Reconstruída - FRC5	113
GRÁFICO 4.17	Diferenças entre a Imagem Original e a Reconstruída - FRAC2	114
GRÁFICO 4.18	Diferenças entre a Imagem Original e a Reconstruída - FRAC3	114
GRÁFICO 4.19	Diferenças entre a Imagem Original e a Reconstruída - FRAC4	114
GRÁFICO 4.20	Diferenças entre a Imagem Original e a Reconstruída - JPG95	114
GRÁFICO 4.21	Diferenças entre a Imagem Original e a Reconstruída - JPG75	114
GRÁFICO 4.22	Diferenças entre a Imagem Original e a Reconstruída - JPG50	114
GRÁFICO 4.23	Diferenças entre a Imagem Original e a Reconstruída - JPG35	115
GRÁFICO 4.24	Box-Plot da Distância 1	121
GRÁFICO 4.25	Box-Plot do Chi-quadrado	121
GRÁFICO 4.26	Box-Plot da Distância de Minkowski	121
GRÁFICO 4.27	Box-Plot da Divergência	121
GRÁFICO 4.28	Box-Plot da Métrica de Canberra	121
GRÁFICO 4.29	Box-Plot da Distância de Bray e Curtis	121
GRÁFICO 4.30	Box-Plot da Distância de Soergel	122
GRÁFICO 4.31	Box-Plot da Distância de Ware e Hedges	122
GRÁFICO 4.32	Box-Plot da Correlação	122
GRÁFICO 4.33	Box-Plot do F de Snedecor	122
GRÁFICO 4.34	Box-Plot do t de Student	122
GRÁFICO 4.35	Box-Plot da Afinidade	122
GRÁFICO 4.36	Box-Plot da Atração	123
GRÁFICO 4.37	Box-Plot da Distância 1 de Mathai	123
GRÁFICO 4.38	Box-Plot da Distância 2 de Mathai	123

GRÁFICO 4.39	Box-Plot da Distância de Matusita	123
GRÁFICO 4.40	Box-Plot da Distância 3 de Matusita	123
GRÁFICO 4.41	Box-Plot da Discrepância	123
GRÁFICO 4.42	Box-Plot da Imprecisão	124
GRÁFICO 4.43	Box-Plot da Divergência Dirigida	124
GRÁFICO 4.44	Árvore de Classificação Final	128
GRÁFICO 4.45	Gráfico das Duas Primeiras Componentes Principais	131
GRÁFICO 4.46	Árvore de Regressão para Análise de Bordas	135

Capítulo 1

Introdução: A Importância da Compressão de Imagens

1.1 Processamento Digital de Imagens

Podemos observar, em nosso dia-a-dia, com os avanços cada vez maiores nos sistemas de telecomunicações, que grande parte das informações que recebemos são informações visuais. Lemos revistas, assistimos a filmes, à TV, vamos ao teatro etc. Por esse motivo, foi surgindo através dos anos a necessidade de gravarmos tais informações e, com elas, o desejo de adquirir, processar e analisar essas figuras (ou imagens) através de computadores digitais. A essas etapas de aquisição, processamento e análise das imagens damos o nome de Processamento Digital de Imagens.

O Processamento Digital de Imagens (PDI) tem sido utilizado desde então como um auxílio em diversas áreas, principalmente às relacionadas à óptica, informação, teoria dos sinais, reconhecimento de padrões e inteligência artificial. Podemos citar, como exemplos de aplicações, a análise de figuras médicas como raios X, tomografias, medidas ultrassônicas e análise de imagens de células e cromossomos; auxílio à meteorologia na estimação da umidade e pressão do ar e sensoriamento remoto em geral; auxílio a investigações criminais através da análise de impressões

digitais e perfis faciais, restauração de pinturas e desenhos de padrões de texturas têxteis, entre outras.

Em geral, o objetivo do processamento digital de imagem é realçar, melhorar ou extrair informação de determinada imagem. E, para tal, a imagem precisa passar por algumas operações. Rosenfeld (1982)¹ subdividiu essas operações em três grandes áreas:

a) área de digitalização e compressão: compreende as operações de conversão das imagens (ou figuras) à forma discreta (digital), a codificação ou busca de aproximações para estas a fim de diminuir o seu espaço de armazenamento;

b) área de realce, restauração e reconstrução: compreende as operações de melhoria da qualidade de uma imagem na presença de “defeitos” como perda da nitidez, presença de ruído, baixo contraste e qualquer outro tipo de degradação da qualidade em geral. Também fazem parte as operações de reconstrução da imagem através de conjuntos de projeções;

c) área de combinação, descrição e reconhecimento: compreende operações de descrição, registro e segmentação de imagens e a comparação destes resultados com modelos que definem classes de figuras.

O processo de digitalização é o primeiro passo a ser realizado antes de processarmos a imagem. Tal processo consiste em convertermos os sinais da imagem, que são contínuos em espaço e intensidade, em valores discretos. Desta forma, a função contínua espacial $f(x,y)$ (nossa imagem) é convertida no sinal de imagem amostrado $f(m \Delta x, n \Delta y)$ que coincide com os elementos de uma matriz imagem $f(m,n)$ nas posições $(m \Delta x, n \Delta y)$. Este processo é chamado de amostragem e não nos deteremos em sua explicação neste momento. Durante toda a dissertação trabalharemos somente com imagens em 2^8 tons de cinza, ou seja, aquelas cujos valores de $f(m,n)$ estão no intervalo $[0,255]$

¹ Rosenfeld, A. e Kak, A. C. *Digital Picture Processing*. 2nd ed. San Diego: Academic Press, Inc., Vol 1, 1982.

e com $m, n = 1, 2, \dots, 512$.

A partir da imagem digitalizada, antes de iniciarmos o processo de aplicação de operações, devemos modelá-la para aplicarmos os algoritmos correspondentes. Schalkoff (1989)² divide os modelos em 4 tipos:

- a) modelos estocásticos (ou probabilísticos) que não possuem dependência geométrica;
- b) modelos determinísticos baseados em variações dos tons de cinza locais;
- c) modelos estocásticos baseados em dependência dos tons de cinza locais (ex.: utilização de matrizes de co-ocorrência);
- d) modelos determinísticos baseados em propriedades globais da imagem como geometria e intensidade.

É sobre estes modelos que as operações desejadas serão realizadas. Além disso, é importante, para este trabalho, ressaltarmos duas características da imagem: texturas e bordas.

1.2 Texturas e Bordas

Ao observarmos uma imagem além das características gerais (formas, cores, etc.) observamos duas características importantes: as texturas e as bordas.

Texturas: podemos definir uma textura como um agrupamento de arranjos similares que possuem determinada cor, forma, brilho, tamanho e inclinação entre outras características. Cada textura possui um padrão que pode ou não ser composto por subpadrões.

De acordo com Pratt (1991)³, podemos dividir as texturas em dois tipos: as naturais e as

² Schalkoff, R. J. *Digital Image Processing and Computer Vision*. New York: John Wiley, 1989.

³ Pratt, W. K. *Digital Image Processing*. 2nd ed. New York: John Wiley, 1991.

artificiais. As texturas artificiais consistem em arranjos de símbolos tais como segmentos de retas, pontos e formas geométricas dispostas sobre um fundo neutro. Na figura (1.1) podemos ver exemplos de três tipos de texturas artificiais.

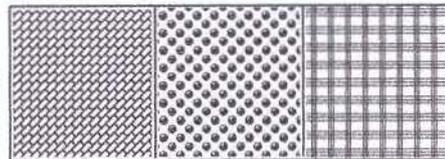


Figura 1.1. Amostras de Texturas Artificiais.

As texturas naturais possuem arranjos por vezes mais complexos e podem ser coloridas ou não. Em nosso caso, como estamos trabalhando com imagens em tons de cinza, apresentaremos somente exemplos desse tipo. Na figura (1.2) temos o exemplo de 6 texturas diferentes retiradas de uma coletânea de texturas. Podemos observar que, muitas vezes, apesar dos tons de cinza presentes em duas texturas serem os mesmos, a forma como estão arranjos define texturas diferentes.

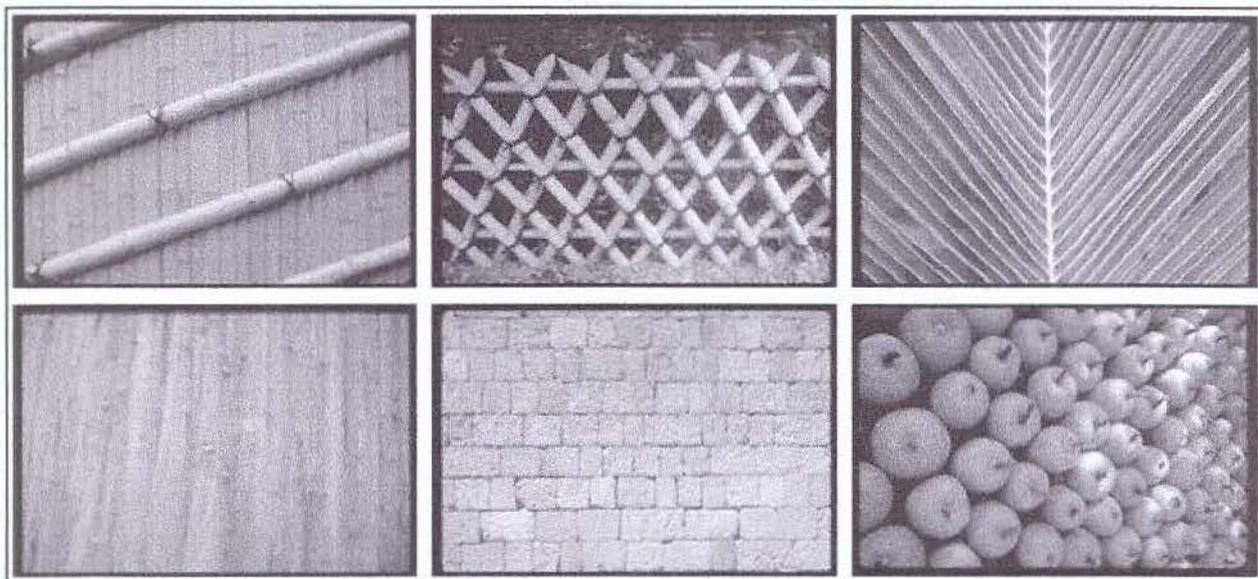


Figura 1.2. Amostras de Texturas Naturais.

Para extrairmos ou caracterizarmos as texturas de uma imagem, várias operações são utilizadas. Numa abordagem estatística, podemos modelar a textura ou caracterizá-la através de medidas como a média, a variância ou a entropia. Mais detalhes podem ser vistos em Pratt (1991)⁴ e Haralick (1986)⁵.

Bordas: podemos definir borda da imagem como os pixels que separam duas regiões de intensidades distintas. Na figura (1.3) podemos ver um exemplo de comportamento da função intensidade numa linha de uma imagem. Nesta, a intensidade da imagem passa do valor a para o valor b .

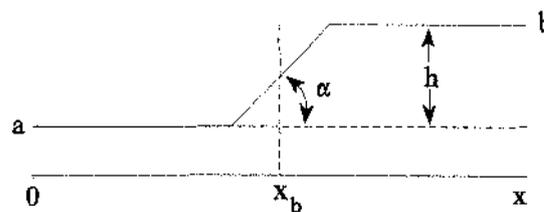


Figura 1.3. "Borda em Rampa".

De acordo com Pratt (1991)⁶, uma borda é caracterizada por sua altura h , ângulo de inclinação α e a coordenada horizontal do ponto médio da inclinação. Uma borda existe se sua altura é maior que um valor específico. Uma borda como a que se apresenta na figura (1.3) em geral é chamada de "borda em rampa". Se o ângulo α é igual a 90° temos uma situação que pode ser vista na figura (1.4) caracterizando uma "borda de passo".

⁴ Pratt, W. K., op. cit. p. 16.

⁵ Haralick, P. M. Statistical Image Texture Analysis. In: Young, T. Y. e Fu, K. *Handbook of Pattern Recognition and Image Processing*. San Diego: Academic Press, 1986. Cap. 11, pp. 247-279.

⁶ Pratt, W. K., op. cit. p. 16.

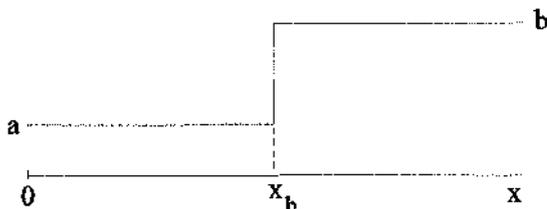


Figura 1.4. "Borda de Passo".

Para detectarmos as bordas de uma imagem, existem vários operadores que trabalham com o cálculo do gradiente da função de intensidade em várias direções (horizontal, vertical, diagonal etc.). Dependendo do operador e da imagem, os resultados podem ou não ser mais eficientes. Isto acontece porque algumas imagens podem possuir as bordas verticais e o operador só detectar bordas horizontais. Para se evitar esse tipo de problema, em geral são aplicados vários operadores que detectem bordas em várias direções.

A situação como vista na figura (1.4) não acontece frequentemente em imagens em tons de cinza. Por esse motivo, o operador detector de bordas ideal é aquele que produz um indicador de borda localizada em x_b , o ponto médio da inclinação. Por último, na figura (1.5) apresentamos outro tipo de borda que pode ser encontrada nas imagens, chamada de "borda em teto".

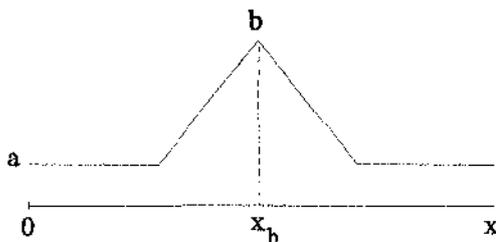


Figura 1.5 "Borda em Teto".

1.3 Compressão de Imagens e Medidas de Distorção

Como já foi visto anteriormente, devido ao aumento no volume das imagens e à necessidade de guardar e transmitir essa informação, uma das áreas de maior interesse do Processamento Digital de Imagem é a compressão de imagens. Podemos definir compressão de imagens como a área que se preocupa com a busca de métodos que codifiquem ou aproximem uma imagem de tal forma que facilite sua transmissão, armazenamento e taxa de acesso. Esta é, sem dúvida, uma área bastante importante, e vem sendo pesquisada desde o início do Processamento Digital de Imagens.

Dependendo do tamanho da imagem, sua transmissão pode ser extremamente demorada, o que inviabiliza projetos como, por exemplo, o de vídeo-conferência ou vídeo-fone. Como o armazenamento de imagens pressupõe um custo financeiro, se pudermos armazenar mais imagens em menos espaço, economizaremos um montante, que poderá ser investido na aquisição de hardware. O maior objetivo da codificação de imagens que leva à compressão é fazê-lo de forma que as distorções não sejam percebidas.

De modo geral, faz-se isto procurando reduzir a redundância estatística ou irrelevância dos dados originais. A principal idéia relacionada à compressão é a de que os tons de cinza de uma imagem possuem muita informação redundante. Assim, se pudermos aplicar transformações que eliminem esta redundância poderemos trabalhar melhor com as imagens reduzidas. Isto pode ser feito da seguinte forma, por exemplo: ao aplicarmos essas transformações pode acontecer de obtermos variáveis não correlacionadas, havendo a possibilidade de umas serem mais importantes que as outras - isto nos possibilitaria, então, resumir boa parte da informação da imagem.

As imagens possuem muitas informações que, por vezes, nem são visíveis ao olho humano e, por isso, dependendo do motivo pelo qual desejamos armazenar nossas imagens, essa informação

pode ser eliminada. Os métodos de compressão que se aproveitam desse fato são chamados de métodos com perda e em geral comprimem mais que os métodos que não admitem perda de informação. Nos métodos que não admitem perda, procuramos apenas uma maneira de representarmos as informações de uma forma mais resumida sem as eliminar (como resumir dados bancários, por exemplo).

Nos métodos onde há uma perda de informação, uma questão comum é: como avaliar essa perda ? Para tratar desse problema foram criadas algumas medidas chamadas de medidas de distorção. Tais medidas visam associar a determinado índice a qualidade visual da imagem reconstruída observada, e sua importância resume-se no fato de podermos comparar vários métodos de compressão através delas.

As medidas de distorção podem ser classificadas em dois tipos: as subjetivas e as convencionais. As subjetivas procuram associar a opinião das pessoas ao observarem as imagens com um índice. Essa associação em geral envolve questionários que procuram detectar a insatisfação ou satisfação ao observar a imagem. Um exemplo de trabalho preocupado com a procura de uma medida subjetiva foi realizado por Karunasekera e Kingsbury (1995)⁷ onde os autores propõem uma medida subjetiva para avaliar imagens com artefatos de bloco.

As medidas convencionais em geral ignoram a opinião das pessoas ao observarem as imagens no momento do cálculo da medida. São baseadas na diferença entre a matriz da imagem reconstruída e a matriz da imagem original. Dentre as medidas convencionais as mais comuns são: razão sinal-ruído (PNSR⁸) e o erro quadrático médio (EQM). Na maioria dos artigos publicados referentes à compressão de imagens, sempre encontramos um destes dois índices. No entanto, o

⁷ Karunasekera, S. A. e Kingsbury, N. G. A Distortion Measure for Blocking Artifacts in Images Based on Human Visual Sensivity. *IEEE Transactions on Image Processing*, Vol. 4, nº 6, pp. 713-723, Jun. 1995.

⁸ $PNSR = 10 \times \log \left(\frac{255^2}{EQM} \right)$. Mais detalhes podem ser vistos no capítulo 3.

EQM não é uma medida perfeita; podemos encontrar imagens de qualidade completamente distintas com o mesmo valor. O mesmo acontece com o PNSR, já que este é simplesmente uma transformação não-linear do EQM.

1.4 Descrição do Trabalho

O objetivo principal deste trabalho é procurarmos outras medidas estatísticas de distorção além do EQM, de preferência que fossem melhores ou complementares a este. Inicialmente, escolhemos quatro métodos de compressão de imagens para, a partir deles, gerarmos imagens com graus de qualidades diferentes. Era necessário avaliarmos vários métodos pois, em geral, cada um produz um tipo de distorção característica à medida que a qualidade da imagem gerada diminui. Os métodos escolhidos foram os que se baseavam na Transformação Cosseno Discreta, na Transformada Rápida de Fourier, na Transformada Wavelets e em Fractais.

A fim de compararmos os resultados obtidos com os dos outros autores da área, escolhemos a imagem clássica da Lena (512×512 pixels) para utilizarmos durante o trabalho. Essa imagem será referenciada como original, e pode ser vista na figura (1.6). Ela possui a vantagem de ter sido utilizada por vários autores para testar seus métodos (foi utilizada na maioria dos artigos referenciados nos próximos capítulos).

Aplicando os métodos de compressão a essa imagem, regulamos a compressão em cada caso de tal forma a sempre obtermos pelo menos uma imagem boa e outra ruim. A partir das imagens reconstruídas, calculamos a distância entre elas e a imagem original, utilizando as medidas selecionadas para esse cálculo. Com esses valores iniciou-se a análise.



Figura 1.6. Imagem Original da Lena.

Com a finalidade de obtermos uma medida que correspondesse à qualidade observada visualmente, fizemos uma pesquisa de opinião com 25 pessoas. Obtivemos, ao final, uma medida subjetiva que seria utilizada como auxiliar à nossa procura de uma medida estatística de distorção de forma a tornar objetiva a qualidade subjetivamente observada.

Capítulo 2

Técnicas de Compressão Analisadas

2.1 Introdução - Imagens como Representações de Processos Estocásticos

Neste capítulo vamos falar um pouco sobre as técnicas de compressão que serão estudadas, descrevendo seus algoritmos e mostrando algumas imagens por estes reconstruídas, que foram utilizadas no trabalho.

Um processo estocástico é definido como uma família de variáveis aleatórias $f(t, \omega)$, onde t é o tempo e os valores de f podem ser discretos ou contínuos, unidimensionais ou multidimensionais. Uma imagem $f(\vec{r}, \omega)$ pode ser considerada como a realização de um processo estocástico ou campo aleatório onde o tempo é representado pelas duas dimensões do espaço euclidiano, $\vec{r} = (x, y) \in \mathbb{R}^2$.

Quando trabalhamos com séries temporais que representam as realizações de processos estocásticos no tempo, normalmente as discretizamos. Da mesma forma, no caso da digitalização de imagens tanto o tempo quanto os valores assumidos (espaços de estados) são discretizados. Normalmente, isto significa que $x, y = 0, \dots, 2^k - 1$ e $f(\vec{r}, \omega)$ pode assumir valores entre 0 e 255. Para facilitarmos a notação, ao invés de $f(\vec{r}, \omega)$ passaremos a escrever somente $f(x, y)$.

Assim como no caso unidimensional, podemos definir uma média e função de autocorrelação para o nosso campo aleatório, onde a média é dada por $\mu(x,y) = E\{f(x,y)\}$ e a função de autocorrelação entre duas variáveis aleatórias $f(x,y)$ e $f(x',y')$ é dada por $R_{ff}(x,y,x',y') = E\{f(x,y) \cdot f(x',y')\}$.

Um campo aleatório é chamado de homogêneo ou estacionário se o valor de $\mu(x,y)$ independe da posição (x,y) , ou seja, é constante com valor igual a μ e sua função de autocorrelação é invariante a translações. Neste caso,

$$R_{ff}(x,y,x',y') = R_{ff}(x-x',y-y') = R_{ff}(\alpha,\beta). \quad (2.1.1)$$

A seguir, descreveremos algumas técnicas utilizadas para a compressão desse tipo de imagens.

2.2 Técnicas Básicas que Utilizam Representações no Domínio das Freqüências

Da mesma forma que vetores podem ser representados por coeficientes relacionados a bases ortonormais, a fim de comprimirmos nossa imagem, vamos expandi-la em termos de um conjunto de funções ortonormais onde os coeficientes a representarão, ao invés dos tons de cinza de cada pixel.

2.2.1 Representações Ortonormais para $L^2(\mathbb{R}^2)$

Seja $f(x,y)$ uma função definida numa região ζ do plano xy . Esta função é dita ser quadrado integrável se

$$\iint_{\zeta} (f(x,y))^2 dx dy < \infty. \quad (2.2.1)$$

Portanto, um conjunto de funções quadrado integráveis $\varphi_{mn}(x,y)$ definidas em ζ é chamado de ortogonal se

$$\iint_{\zeta} \varphi_{mn}(x,y) \varphi_{pq}^*(x,y) \, dx \, dy = 0 \quad m,n \neq p,q \tag{2.2.2}$$

e, se

$$\iint_{\zeta} |\varphi_{mn}(x,y)|^2 \, dx \, dy = 1, \tag{2.2.3}$$

será chamado de ortonormal, sendo que φ_{mn} pode assumir valores reais ou complexos.

Desejamos aproximar a função $f(x,y)$ pelo somatório $\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} a_{mn} \varphi_{mn}(x,y)$ de tal forma que o erro quadrático médio e_{MN}^2 seja minimizado. A expressão de e_{MN}^2 é dada pela equação (2.2.4), a seguir.

$$e_{MN}^2 = \iint_{\zeta} |f(x,y) - \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} a_{mn} \varphi_{mn}(x,y)|^2 \, dx \, dy. \tag{2.2.4}$$

Somente a escolha certa dos a_{mn} poderá minimizar esse erro. Agora, como devemos escolhê-los, quais as condições? A resposta a essa pergunta foi dada por Rosenfeld (1982)⁹ através do seguinte teorema:

Teorema 2.2.1. As constantes a_{mn} que minimizam e_{MN}^2 são dadas por:

$$a_{mn} = \iint_{\zeta} f(x,y) \varphi_{mn}^*(x,y) \, dx \, dy. \tag{2.2.5}$$

⁹ Rosenfeld, A. e Kak, A. C., op. cit. p. 15.

A partir do teorema 2.2.1 podemos ver que, no caso discreto, para reconstruirmos $f(x,y)$ a partir destes coeficientes basta fazermos

$$f(x,y) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} a_{mn} \varphi_{mn}(x,y). \tag{2.2.6}$$

Se $f(x,y)$ for um campo aleatório homogêneo com função de autocorrelação $R_{ff}(\alpha, \beta)$, então

$$f(x,y) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} a_{mn} \varphi_{mn}(x,y). \tag{2.2.7}$$

Como $a_{mn} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \varphi_{mn}(x,y)$, a_{mn} também será uma variável aleatória e a equação (2.2.7) pode ser vista como uma família de equações que podem ser usadas para reconstruirmos nossa figura a partir das amostras.

Vamos retirar para cada campo $M \times N$ amostras. Agora, nosso erro quadrático médio será chamado de erro amostral sobre o campo aleatório $f(x,y)$, cuja notação é ϵ_{MN}^2 e pode ser escrito como:

$$\epsilon_{MN}^2 = E \left\{ \iint_{\zeta} \left| f(x,y) - \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} a_{mn} \varphi_{mn}(x,y) \right|^2 dx dy \right\}. \tag{2.2.8}$$

Reescrevendo em termos de $R_{ff}(\alpha, \beta)$, temos:

$$\epsilon_{MN}^2 = S R_{ff}(0,0) - \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \iiint_{\zeta} \iiint_{\zeta} R_{ff}(x-x', y-y') \varphi_{mn}^*(x,y) \varphi_{mn}(x',y') dx dy dx' dy', \tag{2.2.9}$$

onde $R_{ff}(0,0) = E \{ [f(x,y)]^2 \}$ e S é a área da região ζ no plano xy .

Sob a suposição de homogeneidade do campo aleatório, em geral $R_{ff}(\alpha, \beta)$ pode ser descrita da seguinte forma:

$$R_{ff}(\alpha, \beta) = [R_{ff}(0, 0) - \mu^2] e^{-c_1|\alpha| - c_2|\beta|} + \mu^2, \tag{2.2.10}$$

onde c_1 e c_2 são constantes positivas. Essa função de autocorrelação pode ser usada para modelar figuras cuja correlação nas direções horizontais e verticais são diferentes, ajustando-se os valores das constantes.

Originalmente, os valores de uma imagem são altamente correlacionados, e por este motivo, se as transmitíssemos assim, estaríamos enviando informações redundantes. Ao decompor esta numa base ortonormal, se os coeficientes não fossem correlacionados, teríamos a vantagem de que eventualmente alguns seriam mais importantes enquanto outros seriam irrelevantes e poderiam ser eliminados, levando-nos à compressão das informações originais. Segue daí que devemos procurar funções ortonormais que nos forneçam amostras não correlacionadas para um campo aleatório $f(x, y)$. A solução desta questão é respondida pelo Teorema da Transformação de Karhunen-Loève.

Antes de apresentarmos o Teorema de Karhunen-Loève, precisamos comentar suas suposições. Para o caso de figuras contínuas esse teorema é válido para $-\frac{A}{2} \leq x \leq \frac{A}{2}$ e $-\frac{B}{2} \leq y \leq \frac{B}{2}$, onde tais intervalos definem a região ζ no plano xy . Como estamos trabalhando com imagens digitalizadas, o teorema para o caso discreto assume que nossa amostra terá dimensões $N \times N$. Agora, seja $[f]$ a matriz com os elementos $f(x, y)$ e $[\varphi_{mn}]$ a matriz com os elementos $\varphi_{mn}(x, y)$. O Teorema da Transformação de Karhunen-Loève para figuras discretas é dado por:

Teorema 2.2.2. (Teorema de Karhunen-Loève para Figuras Discretas).

Seja $R(x, y, x', y')$ a função de autocorrelação de $[f]$ que corresponde a

$$R(x, y, x', y') = E \{ f(x, y) f(x', y') \}. \tag{2.2.11}$$

Para campos aleatórios com média zero, as matrizes ortogonais $[\varphi_{mn}]$ que resultam em

amostras a_{mn} não correlacionadas satisfazem à equação

$$\sum_{x'=0}^{N-1} \sum_{y'=0}^{N-1} R(x,y,x',y') \varphi_{mn}(x',y') = \delta_{mn} \varphi_{mn}(x,y), \tag{2.2.12}$$

onde $\varphi_{mn}(x',y')$ e $\varphi_{mn}(x,y)$ são os (x',y') e (x,y) -ésimos elementos respectivamente da matriz $[\varphi_{mn}]$ e

$$\delta_{mn} = E\{|a_{mn}|^2\}. \tag{2.2.13}$$

As matrizes $[\varphi_{mn}]$ são chamadas de automatrizes ou matrizes base de $R(x,y,x',y')$.

Se mudarmos um pouco a notação tal que

$$\begin{bmatrix} \varphi_{mn}(0,0) & \dots & \varphi_{mn}(0,N-1) \\ \vdots & \vdots & \vdots \\ \varphi_{mn}(N-1,0) & \dots & \varphi_{mn}(N-1,N-1) \end{bmatrix} \rightarrow \begin{bmatrix} \varphi_0^s \\ \vdots \\ \varphi_{N-1}^s \\ \varphi_N^s \\ \vdots \\ \varphi_{N^2-1}^s \end{bmatrix}, \tag{2.2.14}$$

$\delta_{mn} = \delta_s$, e substituindo $R(x,y,x',y')$ por $K(i,j) = E\{f_i f_j\}$, com $i,j = 0,1,\dots,N^2-1$, podemos reescrever a equação (2.2.12) como

$$\begin{bmatrix} K(0,0) & K(0,1) & \dots & K(0,N^2-1) \\ K(1,0) & K(1,1) & \dots & K(1,N^2-1) \\ \vdots & \vdots & \vdots & \vdots \\ K(N^2-1,0) & K(N^2-1,1) & \dots & K(N^2-1,N^2-1) \end{bmatrix} \cdot \begin{bmatrix} \varphi_0^s \\ \varphi_1^s \\ \vdots \\ \varphi_{N^2-1}^s \end{bmatrix} = \delta_s \cdot \begin{bmatrix} \varphi_0^s \\ \varphi_1^s \\ \vdots \\ \varphi_{N^2-1}^s \end{bmatrix} \tag{2.2.15}$$

Como consequência da equação (2.2.15), para cada s , δ^s é o autovalor da matriz $[K]$ cujo autovetor correspondente é dado pela matriz $[\varphi_0^s, \dots, \varphi_{N^2-1}^s]$.

Tanto o cálculo da função de autocorrelação para a obtenção da matriz $[K]$ como o cálculo de seus autovetores requerem um grande tempo de processamento. Na prática, utilizamos matrizes $[\varphi_{mn}]$ ortonormais cujas propriedades já foram estudadas. Rosenfeld (1982)¹⁰ mostra que se subdividirmos uma imagem em blocos e determinarmos a matriz $[\varphi_{mn}]$ para cada bloco, as matrizes cujos autovalores tiverem maior magnitude vão ter variações mais rápidas em seus elementos e representarão um efeito de maior resolução espacial da imagem. Como muitas vezes as capacidades de resolução do receptor de nossas imagens são limitadas, se eliminarmos os coeficientes correspondentes a estes autovalores, estaremos eliminando uma informação que talvez não fosse necessária e alcançando uma compressão de nossos dados. Em figuras cujo valor esperado da média é zero, a variância de a_{mn} será o autovalor δ_{mn} e, assim, se δ_{mn} for de grande magnitude, seu a_{mn} correspondente também terá grande magnitude. Portanto, se ordenarmos os coeficientes pelo valor de $|a_{mn}|^2$, os a_{mn} correspondentes aos menores valores poderão ser eliminados, como no caso da Análise de Componentes Principais.

2.2.2 Compressão Através da Transformada de Fourier

Ao aplicarmos a transformada de Fourier numa imagem de dimensões $N \times N$, a base ortonormal será dada por:

$$\varphi_{mn}(x,y) = \frac{1}{N} e^{\frac{j2\pi}{N}(mx-ny)}. \tag{2.2.16}$$

Para obtermos os coeficientes basta fazermos $a_{mn} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \varphi_{mn}(x,y)$, cuja inversa é

¹⁰ Rosenfeld, A. e Kak, A. C. op. cit. p. 15.

dada por $f(x,y) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} a_{mn} \Phi_{mn}(x,y)$.

No caso da transformada de Fourier, a correlação dos coeficientes se aproxima de zero quando N vai para infinito, aproximando-se assim da situação ideal onde estes são não correlacionados.

O método que em geral é utilizado para fazermos a compressão usando a FFT (Transformada Rápida de Fourier) é o seguinte: a partir de uma imagem digitalizada com dimensões $2^M \times 2^N$, a dividimos em blocos de 4×4 ou 8×8 pixels (em geral por conveniência para implementação numérica) e a cada bloco aplicamos a FFT obtendo 16 ou 64 coeficientes respectivamente. Em seguida, ordenamos esses coeficientes segundo sua variância, que é calculada tomando-se a média dos valores de $|a_{mn}|^2$ de todos os blocos. A partir dessa ordenação, eliminamos uma porcentagem destes coeficientes correspondentes aos de menor variância. A compressão então é alcançada à medida que armazenamos apenas os coeficientes mantidos.

A vantagem desse método é que ele aproxima-se do ideal, ou seja, os coeficientes obtidos são praticamente não correlacionados. A desvantagem é que dificilmente conseguimos um alto grau de compressão onde a qualidade final da imagem seja satisfatória.

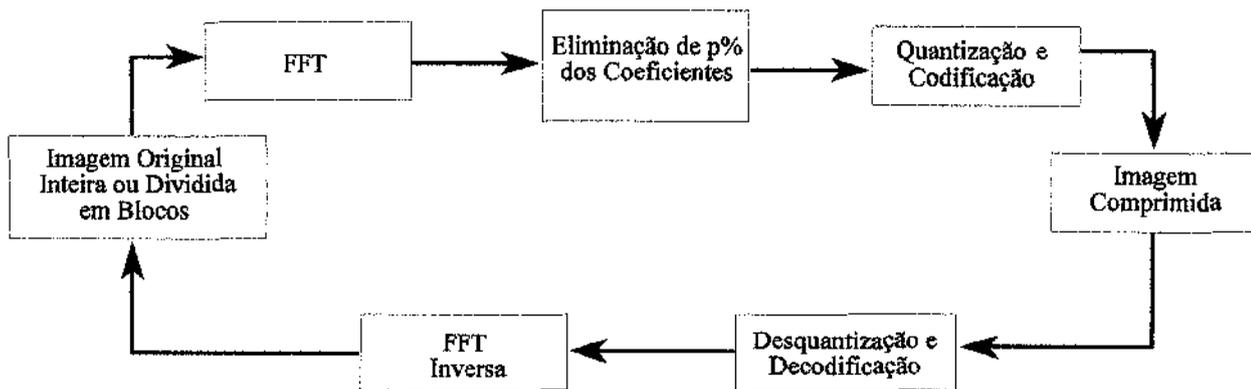
2.2.2.1 Algoritmos

Foram utilizados dois métodos de compressão através da FFT: a compressão através da aplicação da FFT na imagem original e a compressão por blocos (feita em blocos de $2^3 \times 2^3$). Podemos reescrever o algoritmo do 1º método como:

- pegamos a imagem original e a reescrevemos em formato de uma matriz de 512×512 elementos;
- aplicamos a FFT bidimensional na imagem;

- ordenando-se os coeficientes obtidos, estabelecemos o percentual dos coeficientes que serão mantidos e eliminamos os de menor valor.

Para reconstruirmos a imagem basta calcularmos a inversa da FFT bidimensional utilizando os coeficientes guardados. Para fazermos a compressão por blocos o algoritmo é semelhante só que ao invés de o aplicarmos à imagem inteira, o aplicamos a cada um dos subblocos utilizados. Esse algoritmo pode ser resumido como:



2.2.2.2 Exemplos

O primeiro método de compressão através da FFT utilizado foi o correspondente a aplicação da transformada na imagem inteira. A partir dos coeficientes obtidos mantivemos 20%, 10% e 5% obtendo assim três imagens com diferentes graus de compressão, que para serem armazenadas bastava que codificássemos e armazenássemos esses coeficientes. A partir destes reconstruímos cada imagem, mostradas nas figuras (2.1), (2.2) e (2.3). A figura (2.1) que corresponde a imagem reconstruída com 20% dos coeficientes será referenciada por FFT80 no resto do trabalho. A figura (2.2) por FFT90 e a figura (2.3) por FFT95.



Figura 2.1. Imagem Reconstruída com 20% dos Coeficientes após a Aplicação da FFT na Imagem Original.



Figura 2.2. Imagem Reconstruída com 10% dos Coeficientes após a Aplicação da FFT na Imagem Original.



Figura 2.3. Imagem Reconstruída com 5% dos Coeficientes após a Aplicação da FFT na Imagem Original.

O segundo método de compressão através da FFT utilizado foi o correspondente a aplicação da transformada na imagem dividida em blocos. Para cada bloco mantivemos o mesmo número de coeficientes. A partir dos coeficientes obtidos pela aplicação da FFT em cada bloco, mantivemos 50%, 40%, 30% e 10% destes em cada bloco gerando assim 4 imagens com diferentes graus de compressão. As imagens reconstruídas podem ser vistas nas figuras (2.4), (2.5), (2.6) e (2.7). A figura (2.4) será referenciada no resto do trabalho por FTQ50, a figura (2.5) por FTQ60, a figura (2.6) por FTQ70 e a figura (2.7) por FTQ90.



Figura 2.4. Imagem Reconstruída com 50% dos Coeficientes após a Aplicação da FFT na Imagem Original Dividida em Blocos.



Figura 2.5. Imagem Reconstruída com 40% dos Coeficientes após a Aplicação da FFT na Imagem Original Dividida em Blocos.



Figura 2.6. Imagem Reconstruída com 30% dos Coeficientes após a Aplicação da FFT na Imagem Original Dividida em Blocos.



Figura 2.7. Imagem Reconstruída com 10% dos Coeficientes após a Aplicação da FFT na Imagem Original Dividida em Blocos.

2.2.3 Compressão Através da DCT

Para a aplicação da Transformada Cosseno Discreta (DCT) em uma figura, utilizamos

$$\varphi_{mn}(x,y) = \frac{4}{N^2} C(x,y) \cos \frac{(2m+1)x\pi}{2N} \cos \frac{(2n+1)y\pi}{2N}, \quad (2.2.17)$$

onde $x,y = 0,1,\dots,N-1$ e

$$C(x,y) = \begin{cases} \frac{1}{2}, & x = y = 0 \\ 1, & x,y = 1,2,\dots,N-1 \end{cases} \quad (2.2.18)$$

para obtermos os coeficientes e

$$\varphi_{mn}(x,y) = C(x,y) \cos \frac{(2m+1)x\pi}{2N} \cos \frac{(2n+1)y\pi}{2N} \quad (2.2.19)$$

para reconstruirmos a imagem através da equação (2.2.6).

Olhando as equações de transformação vemos que, por definição, as compressões que utilizam a DCT são compressões com perda, uma vez que as equações contêm funções transcendentes de forma que nenhuma implementação será suficientemente precisa. Além disso, a DCT é subótima, já que os coeficientes mantidos são levemente correlacionados. No entanto, em Rosenfeld (1982)¹¹ o autor mostra que a DCT possui algumas características que a leva a ser melhor que a maioria das transformadas. Dentre essas características temos a seguinte que compara a DCT com a FFT.

Vamos supor que a nossa imagem inicial esteja dividida em blocos, pronta para a aplicação da FFT ou DCT. Ao aplicarmos a FFT em um bloco, existe uma suposição, induzida pela teoria associada a esse método, que esse bloco se repetirá sobre todo o plano contendo a imagem. Seja agora f_1 o valor do pixel da borda direita de uma determinada linha desse bloco e f_2 o valor do pixel da borda esquerda do bloco adjacente. Quando esses valores forem muito diferentes, no processo de reconstrução do bloco a partir de apenas um percentual dos coeficientes, os valores de f_1 e f_2 reconstruídos serão distintos dos originais. Quando isso acontece, temos o chamado artefato de bloco, que corresponde ao problema de termos contornos do bloco artificiais, visíveis na imagem reconstruída.

A aplicação da DCT em um bloco não produz esse tipo de artefato. Isto porque na aplicação da DCT nos blocos de uma imagem, existe a suposição que cada bloco faz parte de um conjunto de três blocos numa simetria semelhante à ocasionada por um espelho. Na figura (2.8) podemos ver a comparação dessas suposições feitas pela FFT e DCT. Por esse motivo, quando utilizamos um método de compressão baseado na DCT, ao reconstruirmos a imagem os valores finais obtidos são bem próximos dos originais. Eliminamos assim uma possível descontinuidade que produziria artefatos de bloco na imagem reconstruída.

¹¹ Rosenfeld, A. e Kak, A. C., op. cit. p. 15.

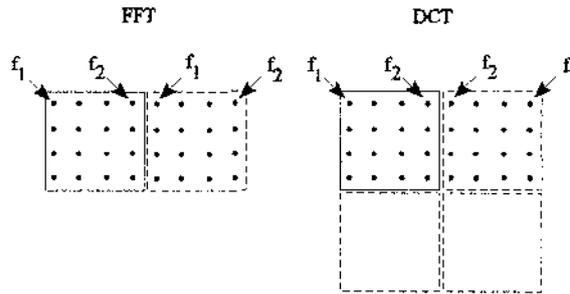


Figura 2.8. Suposição Feita pelos Métodos de Compressão Envolvendo a DCT e a FFT com Relação à Disposição dos Blocos.

Dentre os métodos que utilizam a DCT, neste trabalho utilizamos somente dois. O primeiro é um método cuja implementação é análoga à compressão através da FFT e o segundo é chamado de JPEG, considerado padrão nos sistemas de processamento de imagens comerciais atuais. A descrição de ambos pode ser vista a seguir.

2.2.3.1 Método 1

O método 1 utilizado pode ser descrito como: dividimos uma figura de dimensão $2^M \times 2^N$, em blocos de 4×4 ou 8×8 pixels, e a cada bloco aplicamos a DCT, obtendo-se 16 ou 64 coeficientes. Em seguida, ordenamos esses coeficientes segundo sua variância que é novamente calculada tomando-se a média dos valores de $|a_{mn}|^2$ de todos os blocos. A partir dessa ordenação, eliminamos uma porcentagem desses coeficientes correspondentes aos de menor variância. Por fim, normalizamos os coeficientes mantidos pela sua variância. A compressão é alcançada à medida que armazenamos apenas os coeficientes mantidos e sua variância.

A principal vantagem desse método é a sua facilidade de implementação. A desvantagem é que seu desempenho em termos de compressão não chega a ser tão eficiente como o método JPEG, descrito em mais detalhes a seguir.

2.2.3.2 Método JPEG

O nome deste método de compressão, JPEG, é uma abreviatura de União dos Grupos de Especialistas em Fotografia (*Joint Photographics Experts Group*), e este método foi resultado de uma iniciativa feita no sentido de se criar um único padrão para compressão de imagens. O algoritmo inicial que pode ser visto em Wallace (1991)¹², na época produzia resultados muito superiores aos outros métodos existentes e, antes de se tornar padrão, sofreu algumas modificações que melhoraram seu desempenho. Atualmente vem sendo feitas pesquisas para melhorar sua performance, que já é bastante satisfatória.

A descrição apresentada sobre o método baseia-se na versão descrita em Murray e vanRyper (1994)¹³, que corresponde ao algoritmo do formato JPEG atual, baseado na DCT. No entanto, a boa compressão alcançada pelo método não vem da eliminação de coeficientes, e sim da diminuição da precisão com que estes são armazenados. À medida que diminuimos a precisão, aumentamos o grau de compressão, perdendo na qualidade da imagem armazenada.

Outro fator que possibilita ao método armazenar os coeficientes em tão pouco espaço é o fato dele fazer uma alocação ótima de bits para a codificação dos coeficientes, através do método de Huffman¹⁴. Este método baseia-se na criação de códigos ótimos, onde cada código é unicamente decifrável e instantâneo, isto é, cada código corresponde a um único coeficiente e nenhuma palavra código é prefixo de outra.

¹² Wallace, G. K. The JPEG Still Picture Compression Standard. *Communications of the ACM*. Vol. 34, nº 4, pp. 31-44, Abr. 1991.

¹³ Murray, J. D. e vanRyper, W. *Encyclopedia of Graphics File Formats*. Sebastopol: O'Reilly and Associates, Inc., 1994.

¹⁴ Wallace, G. K., op. cit p. 38.

Para imagens coloridas o método comprime cada componente independentemente, mas o espaço-cor utilizado não é nenhum dos mais conhecidos: RGB (*Red-Green-Blue*), HSI (*Hue-Saturation-Intensity*) ou CMY (*Cyan-Magenta-Yellow*). Isto porque nestes modelos de representação de cor, a informação visual importante em suas três componentes é espalhada, dificultando o processo de seleção para descartar informações.

As melhores taxas de compressão são alcançadas em espaços de cor como o YUV ou o YCbCr que trabalham com luminância/crominância (escala de cinza/informação de cor). A maior parte das informações às quais o olho humano é mais sensível são encontradas na componente de luminância de alta-freqüência em tons de cinza (Y) do espaço-cor YCbCr. As outras componentes de crominância (Cb e Cr) contém informações de alta freqüência às quais o olho humano é menos sensível. Por esta razão, a maior parte dessa informação pode ser descartada. Isto é feito num processo chamado de sub-amostragem, que equivale a, em cada bloco de 2×2 pixels, armazenarmos apenas um pixel correspondente às médias desses pixels em cada canal de crominância.

A taxa de compressão máxima alcançada pelo método, sem visível degradação da imagem reconstruída, é de 25:1. A principal vantagem deste método é a boa taxa de compressão alcançada, e sua principal desvantagem é a de que imagens com grandes áreas com mesma cor não são bem comprimidas, isto é, não alcançam alta taxa de compressão com boa qualidade da imagem. Além disso, esse método não é de fácil implementação.

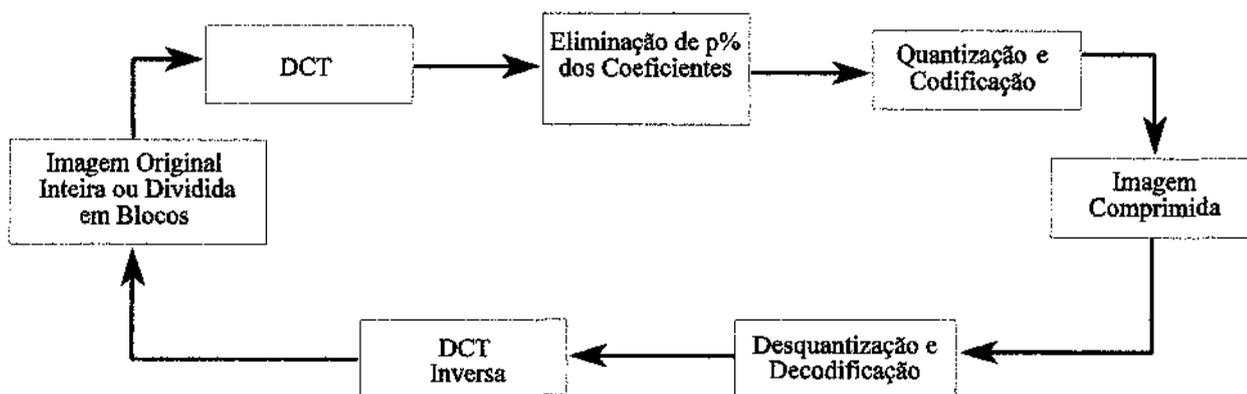
2.2.3.3 Algoritmos

Algoritmo do Método 1

Devido à facilidade de implementação de um algoritmo que fizesse a compressão através de DCT, não foi utilizada a compressão por blocos. O algoritmo utilizado foi:

- pegamos a imagem original e a reescrevemos em formato de uma matriz de 512×512 elementos;
- aplicamos a DCT bidimensional na imagem;
- ordenando-se os coeficientes obtidos, estabelecemos o percentual dos coeficientes que serão mantidos e eliminamos os de menor valor.

Para reconstruirmos a imagem basta calcularmos a inversa da DCT bidimensional nos coeficientes obtidos. Esse algoritmo pode ser resumido como:



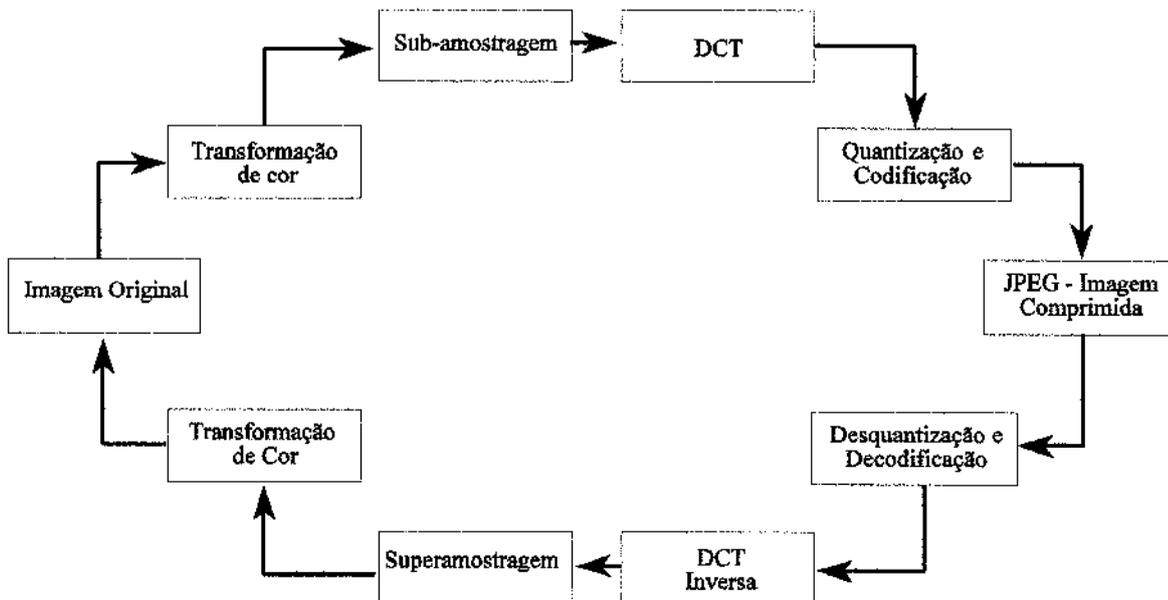
Algoritmo do Método JPEG

O algoritmo do método JPEG pode ser descrito como:

- inicialmente aplicamos uma transformação de cor na imagem (caso esta seja colorida), passando para um espaço de cor como o YUV ou o YcbCr;
- fazemos o processo de sub-amostragem nas componentes de cromaticidade;
- dividimos cada componente da imagem em blocos de 8×8 pixels (a partir dessa etapa processaremos cada componente independentemente);

- aplicamos a Transformação Cosseno Discreta (DCT) a cada bloco de pixels, eliminando assim a informação redundante da imagem;
- quantizamos cada bloco de coeficientes a partir de funções peso pré estabelecidas para a resolução desejada e otimizadas para o olho humano. Este procedimento é parecido ao da normalização dos coeficientes pela sua variância;
- codificamos os coeficientes resultantes usando o método de Huffman para remover as redundâncias dos coeficientes;
- além dos coeficientes são armazenadas informações a respeito do quantizador e da codificação dos coeficientes.

O processo de descompactação é feito executando-se as operações inversas na ordem contrária. O algoritmo JPEG pode ser resumido como:



2.2.3.4 Exemplos

Inicialmente geramos imagens comprimidas através do método 1. Mantivemos 20%, 10% e 5% dos coeficientes resultantes da aplicação da DCT na imagem original, obtendo assim, 3 imagens. Em seguida, reconstruímos cada uma das imagens; o resultado pode ser visto nas figuras (2.9), (2.10) e (2.11). A figura (2.9) será referenciada como DCT80, a figura (2.10) como DCT90 e a figura (2.11) como DCT95.

Em seguida, para o método JPEG, ajustamos um fator chamado ajuste de qualidade (ou fator Q) para os valores 35, 50, 75 e 95, obtendo assim 4 imagens que podem ser vistas nas figuras (2.12) a (2.15) após a reconstituição. A figura (2.12) será referenciada como JPG95, a figura (2.13) como JPG75, a figura (2.14) como JPG50 e a figura (2.15) como JPG35.



Figura 2.9. Imagem Reconstruída com 20% dos Coeficientes após a Aplicação da DCT na Imagem Original.

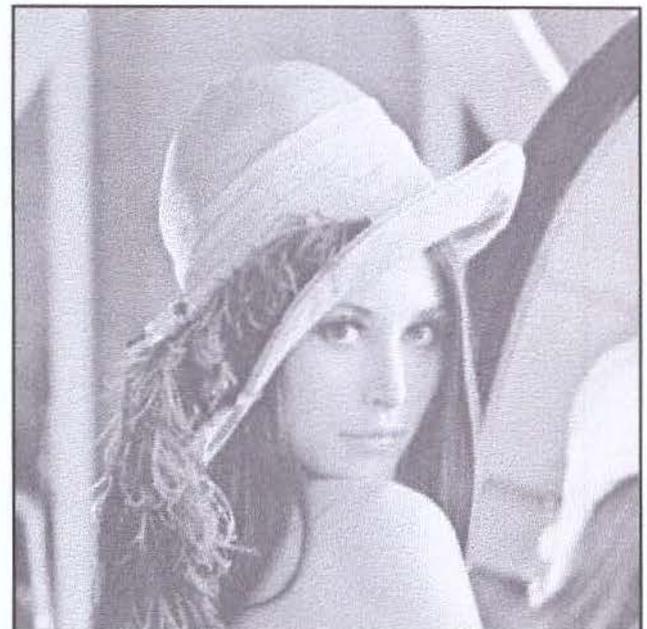


Figura 2.10. Imagem Reconstruída com 10% dos Coeficientes após a Aplicação da DCT na Imagem Original.



Figura 2.11. Imagem Reconstruída com 5% dos Coeficientes após a Aplicação da DCT na Imagem Original.



Figura 2.12. Imagem Reconstruída após a Compressão da Imagem Original através do Método JPEG com Fator Q igual a 95.



Figura 2.13. Imagem Reconstruída após a Compressão da Imagem Original através do Método JPEG com Fator Q igual a 75.



Figura 2.14. Imagem Reconstruída após a Compressão da Imagem Original através do Método JPEG com Fator Q igual a 50.



Figura 2.15. Imagem Reconstruída após a Compressão da Imagem Original através do Método JPEG com Fator Q igual a 35.

2.3 Técnica que Utiliza Wavelets

A princípio, podemos definir wavelets como um pequeno conjunto de ondas que, ao invés de oscilarem sempre regularmente como é o caso das funções trigonométricas, decaem rápido para o valor zero. Da mesma forma que temos a transformada de Fourier, temos também a transformada wavelet, a qual pode ser descrita como uma "ferramenta" que divide dados, funções ou operadores em diferentes componentes de frequência, e então estuda cada componente numa resolução compatível com sua escala.

Este tipo de abordagem nos permite interpretar um sinal independente de sua escala original, o que a torna útil para a aplicação em processamento de imagens. Neste, os detalhes de uma imagem em diferentes resoluções podem caracterizar estruturas diferentes a serem analisadas. Além de processamento de imagens, wavelets vem sendo utilizado em diversas áreas, como por exemplo

processamento de sinais, astronomia, acústica, engenharia nuclear, neurofisiologia, música, fractais, previsão de terremotos, discriminação de falantes e estudo de turbulências, entre outras.

A transformada wavelet de um sinal variável no tempo dependerá de duas variáveis, a escala (frequência) e o próprio tempo, permitindo-nos uma localização por tempo e frequência. O procedimento de análise em geral é adotar uma função ψ de tal forma que a análise temporal será feita através de uma versão de alta frequência, contraída de ψ , enquanto a análise da frequência será feita utilizando uma versão de baixa frequência dilatada.

Assim como na transformada de Fourier, uma função poderá ser representada em termos de coeficientes de uma expansão wavelet. A função ψ não é única, por isso, se escolhermos a função ψ que melhor se adapte aos nossos dados, estes serão esparsamente representados. Esse esparsamento aumentará se, além disso, eliminarmos os coeficientes que estiverem abaixo de um limiar. Por esse motivo wavelets vêm sendo aplicado à compressão de imagens, alcançando resultados superiores aos obtidos por outras técnicas.

No artigo de Davison e Rock (1995)¹⁵, os autores resumem as vantagens da transformada wavelet sobre a de Fourier em duas principais características: ao contrário das funções base da FFT formada por conjuntos de senos e cossenos, a transformada wavelet usa formas mais complicadas, nas quais existem muito mais conjuntos possíveis. Ao contrário de senos e cossenos, as funções wavelets individuais são bem localizadas no espaço, o que nos fornece uma vantagem na representação de dados de alta precisão e dados esparsos.

Como podemos fazer algumas analogias entre a transformada de Fourier e a de wavelets, iniciaremos falando da transformada de Fourier.

Muitas vezes, quando analisamos um sinal $f(t)$ estamos interessados no conteúdo de sua frequência num determinado tempo. Se chamarmos a transformada de Fourier desse sinal de

¹⁵ Davidson, C. e Rock, D. Wavelets and HONN: Pix-Perfect Marriage. *AI Expert*, The Magazine of Artificial Intelligence in Practice. Vol. 10, n° 1, pp. 31-35, Jan. 1995.

$\hat{f}(\omega)$, dado por:

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int f(t) e^{-i\omega t} dt, \quad (2.3.1)$$

ao aplicarmos essa transformação no sinal de interesse, teremos a representação do conteúdo de frequência nesse tempo. No caso de termos picos no sinal, essa informação não poderá ser facilmente obtida e, desse modo, se trabalhássemos com "janelas" no sinal f , poderíamos contornar esse problema. Denotando a transformada de Fourier em janelas por f^{jAN} , temos:

$$f^{jAN}(\omega, t) = \int f(s) g(s-t) e^{-i\omega s} ds. \quad (2.3.2)$$

Existem várias funções de janela g que já foram propostas por alguns autores, e que possuem uma razoável suavização.

A transformada wavelet possui uma descrição de tempo-frequência similar à Fourier em janelas, com algumas diferenças. No caso de wavelets a expressão análoga à da equação (2.3.2) é dada por:

$$f^{WAV}(\alpha, b) = |\alpha|^{-\frac{1}{2}} \int f(t) \psi\left(\frac{t-b}{\alpha}\right) dt. \quad (2.3.3)$$

Para um sinal discreto, restringimos os valores de α para $\alpha = a_0^m$ e o de b para $b = nb_0 a_0^m$, onde $m, n \in Z$ e $a_0 > 1$, $b_0 > 0$ são fixos, podendo reescrever a equação acima como:

$$f_{m,n}^{WAV} = a_0^{-\frac{m}{2}} \int f(t) \psi(a_0^{-m} t - nb_0) dt. \quad (2.3.4)$$

Em ambos os casos, assumimos que ψ satisfaz à seguinte condição:

$$\int \psi(t) dt = 0. \quad (2.3.5)$$

Uma semelhança entre as transformadas wavelets e as de Fourier é que ambas fazem o produto interno de f com uma família de funções indexadas por duas expressões, $g^{\omega t}(s) = e^{i\omega s} g(s-t)$ e $\Psi_{a,b}(s) = |a|^{-\frac{1}{2}} \psi\left(\frac{s-b}{a}\right)$ nas equações (2.3.3) e (2.3.4), respectivamente. As funções $\Psi_{a,b}$ são chamadas de wavelets. Note que assumimos que ψ e g são funções reais. Se isso não for verdade, os complexos conjugados delas devem ser introduzidos nas equações mencionadas.

Existem vários tipos diferentes de transformações wavelets. Daubechies (1991)¹⁶ divide-as em três grupos:

- transformação wavelet contínua;
- transformação wavelet discreta;
 - sistemas discretos redundantes (frames);
 - bases ortonormais de wavelets.

Em processamento de imagens, estamos interessados na área de bases ortonormais de wavelets, que corresponde à análise em multiresoluções.

Para determinados valores de ψ, a_0 e b_0 , a função $\Psi_{m,n}$ constitui uma base ortonormal para $L^2(\mathbb{R})$. Se escolhermos $a_0 = 2$ e $b_0 = 1$, então existe ψ com boas propriedades de localização no tempo e frequência tais que

$$\Psi_{m,n}(x) = 2^{-\frac{m}{2}} \psi(2^m x - n) \quad (2.3.7)$$

constitui uma base ortonormal para $L^2(\mathbb{R})$.

¹⁶ Daubechies, I. *Ten Lectures on Wavelets*. SIAM, 1992.

Utilizando a análise de multiresolução, o cálculo da imagem ou sinal numa resolução 2^j é feito através de um processo de filtragem que na verdade corresponde à decomposição do sinal ou imagem original através de wavelets.

A seguir, mostraremos como a decomposição de um sinal em várias resoluções ou multiresoluções pode ser utilizado para comprimir uma imagem. Apresentaremos alguns teoremas e definições que tornam tal processo possível, além de caracterizarmos as decomposições em multiresoluções.

Antes de começarmos, vamos apresentar a notação utilizada nesta seção:

Se $f, g \in L^2(\mathbb{R})$, o produto interno de f e g será dado por:

$$\langle g(u), f(u) \rangle = \int_{-\infty}^{\infty} g(u)f(u) du, \tag{2.3.8}$$

a norma de f por

$$\|f\| = \sqrt{\int_{-\infty}^{\infty} |f(u)|^2 du} \tag{2.3.9}$$

e $\ell_2(Z)$ será o espaço vetorial de seqüências quadráticamente somáveis

$$\ell_2(Z) = \left\{ (a_i)_{i \in Z} : \sum_{i=-\infty}^{\infty} |a_i|^2 < \infty \right\}. \tag{2.3.10}$$

Para o caso bidimensional das funções $f \in L^2(\mathbb{R}^2)$ a notação é análoga. Começaremos definindo uma aproximação com multiresolução dada pela definição 2.3.1. A maior parte das definições, gráficos e propriedades mostradas a seguir foram obtidas no artigo de Mallat (1989)¹⁷.

¹⁷ Mallat, S. G. Multiresolution Approximations and Wavelet Orthonormal Bases of $L^2(\mathbb{R})$. *Transactions of the American Mathematical Society*. Vol. 315, nº 1, pp. 69-87, Set. 1989.

Definição 2.3.1. Uma aproximação de multiresolução de $L^2(\mathbb{R})$ é uma sequência $(V_j)_{j \in \mathbb{Z}}$ de espaços fechados de $L^2(\mathbb{R})$ tais que as seguintes propriedades valem:

- (1) $V_j \subset V_{j+1}, \forall j \in \mathbb{Z}$.
- (2) $\bigcup_{j \in \mathbb{Z}} V_j$ é denso em $L^2(\mathbb{R})$ e $\bigcap_{j \in \mathbb{Z}} V_j = \{0\}$.
- (3) $f(\cdot) \in V_j \leftrightarrow f(2\cdot) \in V_{j-1}, \forall j \in \mathbb{Z}$.
- (4) $f \in V_j \leftrightarrow f(\cdot - 2^j n) \in V_j, \forall n \in \mathbb{Z}$.
- (5) existe um isomorfismo I de V_0 em $\ell_2(\mathbb{Z})$ que comuta com a ação de \mathbb{Z} . A ação de \mathbb{Z} sobre V_0 é a translação de funções por inteiros enquanto que a ação de \mathbb{Z} sobre $\ell_2(\mathbb{Z})$ é a translação usual.

Vamos tomar uma função em V_j . Se $P_j f$ for a sua projeção ortogonal em V_j , então a propriedade (2) nos garante que:

$$\lim_{j \rightarrow \infty} P_j f = f \quad \forall f \in L^2(\mathbb{R}), \tag{2.3.11}$$

onde esse limite é definido em $L^2(\mathbb{R})$.

Sendo assim, V_j pode ser visto como o conjunto de todas as aproximações possíveis na resolução 2^j das funções em $L^2(\mathbb{R})$ e $P_j f$ como a aproximação das funções f na resolução 2^j .

Para calcularmos as projeções ortogonais temos que mostrar que existe uma função única $\phi \in L^2(\mathbb{R})$ tal que para $j, n \in \mathbb{Z}, \phi_{j,n}(x) = 2^{-\frac{j}{2}} \phi(2^j x - n)$ é uma base ortonormal para V_j .

Vamos mostrar para $j = 0$, ou seja, que $\phi_{0,n}$ é uma base ortonormal de V_0 . A extensão desse resultado para $j \in \mathbb{Z}$ é consequência da propriedade (3), já que tal propriedade implica que:

$$f \in V_j \leftrightarrow f(2^j \cdot) \in V_0 \quad \forall j \in \mathbb{Z}. \tag{2.3.12}$$

Iniciaremos falando sobre a propriedade (5). Uma vez que o operador I é um isomorfismo

de V_0 em $\ell_2(\mathbb{Z})$, existe uma função g tal que

$$g(x) \in V_0 \text{ e } I(g(x)) = \epsilon(n), \text{ onde } \epsilon(n) = \begin{cases} 1, & \text{se } n = 0, \\ 0, & \text{se } n \neq 0. \end{cases} \quad (2.3.13)$$

Como I comuta com translação de inteiros, temos que

$$I(g(x-n)) = \epsilon(k-n). \quad (2.3.14)$$

A seqüência $(\epsilon(k-n))_{n \in \mathbb{Z}}$ é uma base de $\ell_2(\mathbb{Z})$ e, portanto, $(g(x-n))_{n \in \mathbb{Z}}$ é base de V_0 .

Seja $f \in V_0$ e $I(f) = (a_n)_{n \in \mathbb{Z}}$. Como I é um isomorfismo, $\|f\|$ e $\left(\sum_{k=-\infty}^{\infty} |a_n|^2\right)^{\frac{1}{2}}$ são duas normas equivalentes em V_0 . Expressando essa equivalência em termos de g , f pode ser decomposto da seguinte forma:

$$f(x) = \sum_{n=-\infty}^{\infty} a_n g(x-n). \quad (2.3.15)$$

A partir da Transformada de Fourier da equação acima, temos

$$\hat{f}(\xi) = M(\xi) \hat{g}(\xi), \text{ onde } M(\xi) = \sum_{n=-\infty}^{\infty} a_n e^{-in\xi}, \quad (2.3.16)$$

obtendo a norma de \hat{f} ,

$$\|\hat{f}\|^2 = \int_0^{2\pi} |M(\xi)|^2 \sum_{n=-\infty}^{\infty} |\hat{g}(\xi+2n\pi)|^2 d\xi < \infty. \quad (2.3.17)$$

Segue que existem $c_1, c_2 > 0$ tais que para $\xi \in \mathbb{R}$,

$$c_1 \leq \left(\sum_{n=-\infty}^{\infty} |\hat{g}(\xi+2n\pi)|^2 \right)^{\frac{1}{2}} \leq c_2. \quad (2.3.18)$$

Estamos procurando ϕ tal que $(\phi(x-n))_{n \in \mathbb{Z}}$ seja uma base ortonormal de V_0 . Para

calcularmos ϕ , vamos ortogonalizar a base $(g(x-n))_{n \in \mathbb{Z}}$.

Pode ser mostrado¹⁸ que $(\phi(x-k))_{k \in \mathbb{Z}}$ são ortogonais se e somente se satisfazem à seguinte propriedade:

$$\sum_{n=-\infty}^{\infty} |\hat{\phi}(\xi + 2n\pi)|^2 = 1. \tag{2.3.19}$$

Como $\phi \in V_0$, pela equação (2.3.16) vemos que existe $M_\phi(\xi)$ com periodicidade 2π tal que

$$\hat{\phi}(\xi) = M_\phi(\xi)\hat{g}(\xi). \tag{2.3.20}$$

Substituindo a equação (2.3.20) em (2.3.19), temos

$$M_\phi(\xi) = \left(\sum_{n=-\infty}^{\infty} |\hat{g}(\xi + 2n\pi)|^2 \right)^{\frac{1}{2}}. \tag{2.3.21}$$

A equação (2.3.18) prova que (2.3.21) define uma função $M_\phi(\xi) \in L^2([0, 2\pi])$. Se ϕ é dada por (2.3.20), através da equação (2.3.18) podemos ver que g pode ser decomposta na família ortogonal $(\phi(x-n))_{n \in \mathbb{Z}}$. Isto implica que $(\phi(x-n))_{n \in \mathbb{Z}}$ gera V_0 .

Antes de falarmos sobre as propriedades de ϕ , vamos definir a condição de regularidade de uma aproximação de multiresolução de $L^2(\mathbb{R})$.

Definição 2.3.2. Uma função $f \in L^2(\mathbb{R})$ é regular se e somente se ela é continuamente diferenciável e, para $C > 0$ e $\forall x \in \mathbb{R}$, as seguintes condições são satisfeitas:

$$|f(x)| \leq C(1+x^2)^{-1} \text{ e } |f'(x)| \leq C(1+x^2)^{-1}. \tag{2.3.22}$$

¹⁸ Walter, G. *Wavelets and Other Orthogonal Systems with Applications*. Boca Raton: CRC Press, 1994. P. 37.

Como consequência, uma aproximação de multiresolução $(V_j)_{j \in \mathbb{Z}}$ é chamada regular se e somente se ϕ é regular.

Agora, através da propriedade (2) da definição 2.3.1, vemos que $\frac{1}{\sqrt{2}} \phi\left(\frac{1}{2} \cdot\right) \in V_{-1} \subset V_0$. Decompondo $\frac{1}{\sqrt{2}} \phi\left(\frac{1}{2} \cdot\right)$ na base ortonormal de V_0 , $(\phi(x-n))_{n \in \mathbb{Z}}$, temos

$$\frac{1}{\sqrt{2}} \phi\left(\frac{x}{2}\right) = \sum_n \left\langle \frac{1}{\sqrt{2}} \phi\left(\frac{u}{2}\right), \phi(u-n) \right\rangle \phi(x-n). \tag{2.3.23}$$

Se $h_n = \left\langle \frac{1}{2} \phi\left(\frac{u}{2}\right), \phi(u-n) \right\rangle$,

$$\phi\left(\frac{x}{2}\right) = 2 \sum_n h_n \phi(x-n). \tag{2.3.24}$$

Note que $h_n = \int \frac{1}{2} \phi\left(\frac{u}{2}\right) \cdot \bar{\phi}(u-n) du$ e, fazendo $x = \frac{u}{2}$,

$$h_n = \int \phi(x) \bar{\phi}(2x-n) dx = \langle \phi(x), \phi(2x-n) \rangle. \tag{2.3.25}$$

Como a aproximação de multiresolução é regular, o decaimento de h_n satisfaz $|h_n| = O(1+n^2)^{-1}$. Calculando a transformada de Fourier,

$$\hat{\phi}(2\xi) = m_0(\xi) \cdot \hat{\phi}(\xi), \text{ onde } m_0(\xi) = \sum_n h_n e^{-in\xi}. \tag{2.3.26}$$

As condições de $m_0(\xi)$ são dadas pelo teorema a seguir.

Teorema 2.3.1. A função $m_0(\xi)$ definida acima satisfaz

$$|m_0(\xi)|^2 + |m_0(\xi + \pi)|^2 = 1, \quad m_0(0) = 1. \tag{2.3.27}$$

Como consequência, temos :

Teorema 2.3.2. Seja $m_0(\xi) = \sum_n h_n e^{-in\xi}$ tal que $|h_n| = O(1+n^2)^{-1}$, $m_0(\xi) = 1$,

$\hat{\phi}(\xi) = \prod_{n=1}^{\infty} m_0(2^{-n}\xi)$, $|m_0(\xi)|^2 + |m_0(\xi+\pi)|^2 = 1$, $m_0(\xi) \neq 0$ com $\xi \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, e $\hat{\phi}$ é a transformada de Fourier de ϕ tal que $(\phi(x-n))_{n \in \mathbb{Z}}$ é uma base ortonormal de um subespaço fechado V_0 de $L^2(\mathbb{R})$. Se ϕ é regular, então a seqüência de espaços vetoriais $(V_j)_{j \in \mathbb{Z}}$ definida a partir de V_0 pela propriedade (3) da definição 2.3.1, é uma aproximação de multiresolução regular de $L^2(\mathbb{R})$.

Agora que já definimos ϕ e suas propriedades, podemos ver que a projeção de f num espaço V_j é dada por:

$$P_j f(x) = 2^j \sum_{n=-\infty}^{\infty} \langle f(u), \phi(2^j u - n) \rangle \phi(2^j x - n). \tag{2.3.28}$$

A aproximação do sinal f na resolução 2^j , $P_j f$ é dessa forma caracterizada pelo produto interno denotado por:

$$P_j^d f = \langle \langle f(u), \phi(2^j u - n) \rangle \rangle_{n \in \mathbb{Z}}, \tag{2.3.29}$$

e $P_j^d f$ é chamado de aproximação discreta de f na resolução j .

Observemos agora os gráficos de uma ϕ e sua $\hat{\phi}$ correspondente (gráficos 2.1 e 2.2).

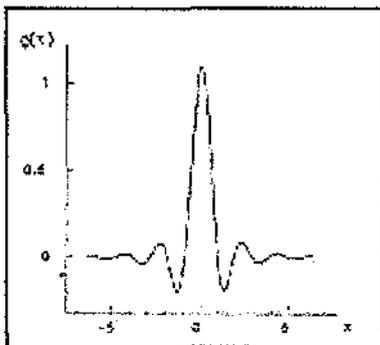


Gráfico 2.1. Exemplo de Função Escala (ϕ).

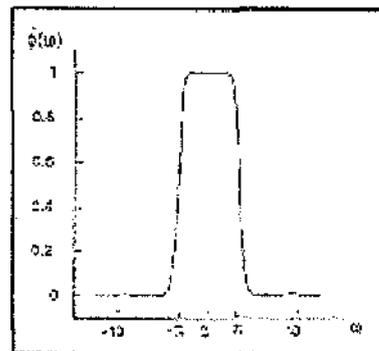


Gráfico 2.2. Transformada de Fourier de $\phi(x)$.

Podemos ver pelo gráfico de $\hat{\phi}$ que a função escala ϕ é um filtro passa-baixa e, sendo assim, $P_j^d f$ pode ser interpretado como uma filtragem passa-baixa de f seguida de uma amostragem uniforme de razão 2^j .

Vamos mostrar em seguida como podemos encontrar esse filtro.

Sabemos que $V_j \subset V_{j+1}$. Portanto, podemos expandir $\phi_{j,n} \in V_j$ na base ortonormal de V_{j+1} da seguinte forma:

$$2^{\frac{j}{2}} \phi(2^j x - n) = \sum_k \left\langle 2^{\frac{j}{2}} \cdot \phi(2^j u - n), 2^{\frac{j+1}{2}} \phi(2^{j+1} u - k) \right\rangle 2^{\frac{j+1}{2}} \phi(2^{j+1} x - k) \tag{2.3.30}$$

$$\Rightarrow \phi(2^j x - n) = \sum_k 2^{j+1} \int \phi(2^j u - n) \bar{\phi}(2^{j+1} u - k) du \phi(2^{j+1} x - k). \tag{2.3.31}$$

Vamos trabalhar agora com $2^{j+1} \int \phi(2^j u - n) \bar{\phi}(2^{j+1} u - k) du$.

Se fizermos $x = 2^{j+1} u - k$, obteremos $\int \phi\left(\frac{x+k}{2} - n\right) \bar{\phi}(x) dx$.

Chamando $\frac{x+k-2n}{2} = y$, a expressão anterior pode ser reescrita como:

$$2 \cdot \int \phi(y) \bar{\phi}(2y+2n-k) dy \tag{2.3.32}$$

Portanto

$$\phi(2^j x - n) = \sum_k 2 \cdot \langle \phi(u), \phi(2u - (k-2n)) \rangle \phi(2^{j+1} x - k). \tag{2.3.33}$$

Calculando o produto interno com f de ambos os lados, temos:

$$\langle f(u), \phi(2^j u - n) \rangle = \sum_k 2 \cdot \langle \phi(u), \phi(2u - (k-2n)) \rangle \cdot \langle f(u), \phi(2^{j+1} x - k) \rangle. \tag{2.3.34}$$

Seja h o filtro discreto com resposta $h_n = \langle \phi(u), \phi(2u - n) \rangle$, $\forall n \in \mathbb{Z}$ e \tilde{h} o filtro "espelho" tal que $\tilde{h}_n = h_{-n}$ então $\langle f(u), \phi(2^j u - n) \rangle = 2 \sum_k \tilde{h}_{2n-k} \cdot \langle f(u), \phi(2^{j+1} x - k) \rangle$.

Reescrevendo em termos de projeções, temos:

$$P_j^d f(x) = \sum_k \tilde{h}_{2^n-k} P_{j+1}^d f(x). \quad (2.3.35)$$

Isto quer dizer que para $j < 0$, iniciando com o sinal na resolução $j = 0$, ou seja, com o sinal original, através da convolução deste com o filtro \tilde{h} poderemos obter o sinal nas resoluções subsequentes.

Vimos até agora que a aproximação de uma função na resolução 2^j é igual à sua projeção em V_j . A precisão adicional, quando passamos da resolução 2^j a 2^{j+1} é dada pela projeção ortogonal do complemento de V_j em V_{j+1} .

Seja O_j o espaço vetorial tal que $V_{j+1} = V_j \oplus O_j$. Queremos mostrar agora que existe uma função $\psi(x)$, chamada de wavelet ortogonal, tal que $(\sqrt{2^j} \psi(2^j x - n))_{n \in \mathbb{Z}}$ é uma base ortonormal de O_j .

Vamos começar mostrando para $j = -1$:

$(\phi(x-n))_{n \in \mathbb{Z}}$ é base ortonormal de V_0 e sabemos que $V_0 = V_{-1} \oplus O_{-1}$. Estamos procurando ψ tal que $\frac{1}{\sqrt{2}} \psi\left(\frac{1}{2} \cdot\right)$ seja uma base ortonormal de O_{-1} . Como $\frac{1}{\sqrt{2}} \psi\left(\frac{1}{2} \cdot\right) \in V_0$, podemos decompô-lo na base ortonormal de V_0 :

$$\frac{1}{\sqrt{2}} \psi\left(\frac{x}{2}\right) = \sum_n \left\langle \frac{1}{\sqrt{2}} \psi\left(\frac{u}{2}\right), \phi(u-n) \right\rangle \cdot \phi(u-n) \quad (2.3.36)$$

se

$$\left\langle \frac{1}{\sqrt{2}} \psi\left(\frac{u}{2}\right), \phi(u-n) \right\rangle = \langle \psi(u), \phi(2u-n) \rangle = g_n, \quad (2.3.37)$$

$$\frac{1}{\sqrt{2}} \psi\left(\frac{x}{2}\right) = \sum_n g_n \phi(u-n). \quad (2.3.38)$$

Calculando a Transformada de Fourier, temos:

$$\hat{\psi}(2\xi) = m_f(\xi) \cdot \hat{\phi}(\xi). \quad (2.3.39)$$

Como $V_0 = V_{-1} \oplus O_{-1}$, a transformada de qualquer $f \in V_0$ pode ser decomposta da seguinte forma:

$$\hat{f}(\xi) = a(\xi)\hat{\phi}(\xi) = b(\xi)\hat{\phi}(2\xi) + c(\xi)\hat{\psi}(2\xi), \tag{2.3.40}$$

onde a é uma função 2π periódica e membro de $L^2([0, \pi])$ e b, c são funções π periódicas e membros de $L^2([0, \pi])$. Podemos reescrever a equação acima como:

$$a(\xi) = b(\xi) \cdot m_0(\xi) + c(\xi) \cdot m_f(\xi), \tag{2.3.41}$$

A ortogonalidade da decomposição é equivalente a

$$\int_0^{2\pi} |a(\xi)|^2 d\xi = \int_0^{\pi} |b(\xi)|^2 d\xi + \int_0^{\pi} |c(\xi)|^2 d\xi \tag{2.3.42}$$

e é satisfeita para qualquer $a(\xi)$ se e somente se

$$|m_0(\xi)|^2 + |m_f(\xi)|^2 = 1 \quad e \quad m_0(\xi)m_f(\xi) + m_0(\xi + \pi) \cdot m_f(\xi + \pi) = 0. \tag{2.3.43}$$

Essas equações são condições necessárias e suficientes em $m_f(\xi)$ para construir ψ . Uma escolha possível de $m_f(\xi)$ é $m_f(\xi) = e^{-i\xi} m_0(\xi + \pi)$.

Agora, sabemos que qualquer espaço vetorial V_j pode ser decomposto como $V_j = \bigoplus_{j=-\infty}^{j-1} O_j$. Se $\bigcup_j V_j$ é denso em $L^2(\mathbb{R})$, a soma $\bigoplus_j O_j$ é também densa em $L^2(\mathbb{R})$. Assim, a família de funções $\left\{ \sqrt{2^j} \psi(2^j x - n) \right\}_{(n,j) \in \mathbb{Z}}$ é base ortonormal de $L^2(\mathbb{R})$.

Se $m_f(\xi) = e^{-i\xi} m_0(\xi + \pi)$, aplicando-se a transformada de Fourier inversa,

$$g_n = (-1)^{1-n} h_{1-n}. \tag{2.3.44}$$

Uma vez definido ψ , vamos chamar de $D_j f$ a projeção de f no espaço O_j :

$$D_j f(x) = 2^j \sum_n \langle f(u), \psi(2^j u - n) \rangle \cdot \psi(2^j x - n). \tag{2.3.45}$$

Essa decomposição é caracterizada pelo produto interno

$$D_j^d f(x) = \langle f(u), \psi(2^j u - n) \rangle, \tag{2.3.46}$$

onde $D_j^d f$ será o detalhe discreto do sinal $f(x)$ na resolução 2^j e conterà a diferença de informação entre $P_j f$ e $P_{j+1} f$. Se observarmos a seguir os gráficos de uma ψ e sua respectiva $\hat{\psi}$, veremos que a função wavelet corresponde a um filtro de passa-alta e $D_j^d f$ pode ser visto como a filtragem passa-alta de f seguida de uma amostragem na razão 2^j .

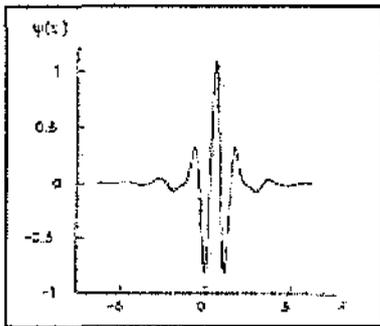


Gráfico 2.3. Função Wavelet $\psi(x)$ Associada a Função Escala do Gráfico 2.1.

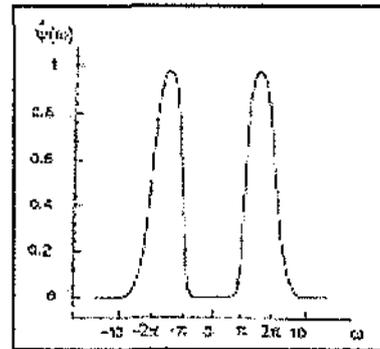


Gráfico 2.4. Módulo da Transformada de Fourier de $\psi(x)$.

Agora, como $2^{\frac{j}{2}} \psi(2^j \cdot - n) \in O_j \subset V_{j+1}$, podemos decompor $2^{\frac{j}{2}} \psi(2^j \cdot - n)$ na base de V_{j+1} :

$$2^{\frac{j}{2}} \psi(2^j x - n) = \sum_k \left\langle 2^{\frac{j}{2}} \cdot \psi(2^j u - n), 2^{\frac{j+1}{2}} \phi(2^{j+1} u - k) \right\rangle 2^{\frac{j+1}{2}} \phi(2^{j+1} x - k) \tag{2.3.47}$$

$$\Rightarrow \psi(2^j x - n) = \sum_k 2^{j+1} \int \psi(2^j u - n) \bar{\phi}(2^{j+1} u - k) du \phi(2^{j+1} x - k). \tag{2.3.48}$$

Portanto,

$$\psi(2^j x - n) = \sum_k 2 \langle \psi(u), \phi(2u - (k - 2n)) \rangle \psi(2^{j+1} x - k). \tag{2.3.49}$$

Calculando o produto interno com f ,

$$\langle f(u), \psi(2^j u - n) \rangle = \sum_k 2 \langle \psi(u), \phi(2u - (k - 2n)) \rangle \langle f(u), \phi(2^{j+1} x - k) \rangle. \tag{2.3.50}$$

Como $g_n = \langle \psi(u), \phi(2u - n) \rangle$, e tomando-se $\tilde{g}_n = g_{-n}$, temos:

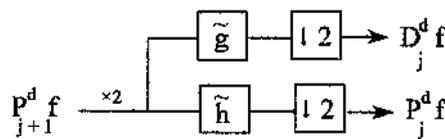
$$\langle f(u), \psi(2^j u - n) \rangle = 2 \sum_k \tilde{g}_{2n-k} \langle f(u), \phi(2^{j+1} x - k) \rangle, \tag{2.3.51}$$

que reescrevendo em termos de projeções fica:

$$D_j^d f(x) = 2 \sum_k \tilde{g}_{2n-k} P_{j+1}^d f(x). \tag{2.3.52}$$

Isto quer dizer que a partir do sinal original $P_0^d f$, podemos obter $D_j^d f(x)$ para $j > 0$, convoluindo o sinal com o filtro \tilde{g} .

Esse algoritmo pode ser descrito como:



onde \boxed{x} : convoluir com o filtro x
 $\boxed{1/2}$: processo de sub-amostragem

Figura 2.16. Decomposição do Sinal f .

Repetindo esse algoritmo para $1 \geq j \geq -J$, obteremos a representação do sinal $P_0^d f$ em J níveis de resolução.

Através da expressão (2.3.44) vemos que g é o reflexo de h . Em processamento de sinais g e h são chamados filtros reflexo de quadratura (QMF). Estes tipos de filtros possuem características relacionadas ao processo de sub-amostragem mencionados na figura (2.16) e possuem várias propriedades dentre as quais podemos destacar a seguinte¹⁹:

Seja x_n nosso sinal de entrada amostrado tal que $n \in \mathbb{Z}$. Um sinal com energia finita será aquele que $\sum_{i=-\infty}^{\infty} |x_i|^2 < \infty$. Vamos chamar de $D: \ell_2(\mathbb{Z}) \rightarrow \ell_2(2\mathbb{Z})$ o operador de decimação denotado por D que corresponde a ficarmos somente com os termos pares da seqüência $(x_i)_{i \in \mathbb{Z}}$, obtendo ao final uma nova seqüência $(x_{2i})_{i \in \mathbb{Z}}$. Para facilitar a notação, vamos chamar $X = (x_i)_{i \in \mathbb{Z}}$. Sejam agora filtros G e H que seguem a propriedade dada pela equação (2.3.44). Se tomarmos dois sinais com energia finita, X_0 e X_1 , tais que $X_0 = G(X)$ e $X_1 = H(X)$, a aplicação do processo de sub-amostragem que corresponde à aplicação do operador D a esses sinais nos fornecerá $Y_0 = D(X_0) = (x_{2i}^0)_{i \in \mathbb{Z}}$ e $Y_1 = D(X_1) = (x_{2i}^1)_{i \in \mathbb{Z}}$. Os dois filtros G e H são chamados de filtros de quadratura se, para todo os sinais X de energia finita,

$$\|Y_0\|^2 + \|Y_1\|^2 = \|X\|^2. \tag{2.3.53}$$

Agora, sabemos que $2^{-\frac{j+1}{2}} \phi(2^{j+1} \cdot -n) \in V_{j+1}$ e, como $V_{j+1} = V_j \oplus O_j$, podemos reescrever $2^{-\frac{j+1}{2}} \phi(2^{j+1} \cdot -n)$ da seguinte forma:

$$\begin{aligned} 2^{-\frac{j+1}{2}} \phi(2^{j+1} x - n) &= \sum_k \left\langle 2^{\frac{j}{2}} \phi(2^j u - k), 2^{-\frac{j+1}{2}} \phi(2^{j+1} u - n) \right\rangle \cdot 2^{\frac{j}{2}} \phi(2^j x - k) \\ &+ \sum_k \left\langle 2^{\frac{j}{2}} \psi(2^j u - k), 2^{-\frac{j+1}{2}} \phi(2^{j+1} u - n) \right\rangle \cdot 2^{\frac{j}{2}} \psi(2^j x - k). \end{aligned} \tag{2.3.54}$$

$$\begin{aligned} \Rightarrow \phi(2^{j+1} x - n) &= \sum_k 2^j \int \phi(2^j u - k) \cdot \bar{\phi}(2^{j+1} u - n) du \phi(2^j x - k) \\ &+ \sum_k 2^j \int \psi(2^j u - k) \phi(2^{j+1} u - n) du \psi(2^j x - k). \end{aligned} \tag{2.3.55}$$

¹⁹ Meyer, Y. e Ryan, R. P. *Wavelets; algorithms and applications*. Philadelphia: SIAM, 1993.

Já vimos que:

$$2^{j+1} \int \phi(2^j u - n) \cdot \bar{\phi}(2^{j+1} u - k) du = \int \phi(y) \bar{\phi}(2y + 2n - k) dy \quad (2.3.56)$$

e

$$2^{j+1} \int \psi(2^j u - n) \bar{\phi}(2^{j+1} u - k) du = \int \psi(y) \bar{\phi}(2y + n - k) dy \quad (2.3.57)$$

$$\begin{aligned} \therefore \phi(2^{j+1} x - n) &= \sum_k 2^j 2^{(j+1)} \int \phi(y) \bar{\phi}(2y + 2k - n) dy \phi(2^j x - k) \\ &+ \sum_k 2^j 2^{(j+1)} \int \psi(y) \bar{\phi}(2y + 2k - n) dy \psi(2^j x - k) \end{aligned} \quad (2.3.58)$$

$$\begin{aligned} \therefore \phi(2^{j+1} x - n) &= \sum_k 2^{-1} \langle \phi(u), \phi(2u - (n - 2k)) \rangle \phi(2^j x - k) \\ &+ \sum_k 2^{-1} \langle \psi(u), \phi(2u - (n - 2k)) \rangle \psi(2^j x - k) \\ &= \sum_k 2^{-1} h_{n-2k} \phi(2^j x - k) + \sum_k 2^{-1} g_{n-2k} \psi(2^j x - k). \end{aligned} \quad (2.3.59)$$

Calculando o produto interno com $f(x)$, temos:

$$\langle f(u) \phi(2^{j+1} u - n) \rangle = \sum_k 2^{-1} h_{n-2k} \langle \phi(2^j u - k), f(u) \rangle + \sum_k 2^{-1} g_{n-2k} \langle \psi(2^j x - k), f(u) \rangle, \quad (2.3.60)$$

que corresponde a

$$P_{j+1}^d f = \sum_k \frac{1}{2} h_{n-2k} P_j^d f + \sum_k \frac{1}{2} g_{n-2k} \cdot D_j^d f. \quad (2.3.61)$$

Podemos ver que, a partir do valor do sinal na resolução j , podemos reconstruir o sinal na resolução $j+1$. Repetindo esse processo até $j = 0$, obtemos novamente a imagem original. Esse processo de reconstrução pode ser descrito como:

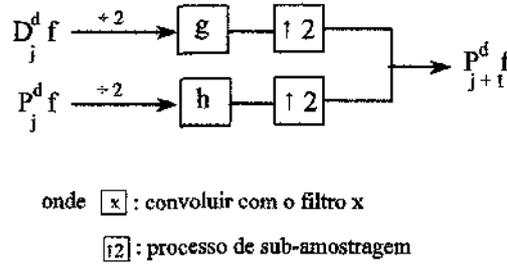


Figura 2.17. Reconstrução do Sinal f .

A operação indicada por $\uparrow 2$ corresponde a acrescentar uma observação com valor zero entre cada amostra. Essa operação na verdade corresponde ao operador adjunto $E = D^*: \ell_2(2Z) \rightarrow \ell_2(Z)$ e equivale a um filtro de interpolação. Mais informações sobre este operador e o operador de decimação citado anteriormente podem ser encontradas em Strang e Nguyen (1995)²⁰.

Vemos assim como através de processos de filtragem podemos decompor um sinal em várias resoluções. Tal processo de decomposição corresponde à decomposição do sinal através de wavelets, e ao final de algumas filtragens teremos os valores de cada $D_j^d f$ e um valor de $P_j^d f$. Os valores de $D_j^d f$ muitas vezes são chamados de coeficientes wavelets ou de ruído, já que eles correspondem à diferença do sinal em duas resoluções consecutivas. Através de um processo de filtragem podemos, através dessas informações, reconstruir o sinal original.

Até agora trabalhamos somente no caso unidimensional. O modelo wavelet pode ser facilmente generalizado para qualquer dimensão $n > 0$. No caso de imagens, estaremos trabalhando com funções bidimensionais $f \in L^2(\mathbb{R}^2)$. A extensão para o caso bidimensional é feita aproveitando-se muitas propriedades do caso unidimensional, e mostraremos somente os principais

²⁰ Strang, G. e Nguyen, F. *Wavelets and Filters Banks*. Workshop Notes: San Jose State University, Jan. 13-16. 1995.

resultados (mais informações sobre o caso bidimensional pode ser vistas em Mallat (1991)²¹).

Seja $(V_j)_{j \in \mathbb{Z}}$ a aproximação de multiresolução de $L^2(\mathbb{R}^2)$. Pode ser mostrado que existe uma função única, chamada função escala Φ , tal que a família de funções $2^j \Phi(2^j \cdot - n, 2^j \cdot - m)$ forma uma base ortonormal de V_j .

Para aproximações de multiresolução, cada espaço vetorial V_j pode ser decomposto como o produto tensorial de dois subespaços idênticos de $L^2(\mathbb{R})$, $V_j = V_j^1 \otimes V_j^2$. Além disso, a seqüência $(V_j)_{j \in \mathbb{Z}}$ forma uma aproximação de multiresolução de $L^2(\mathbb{R}^2)$ se e somente se (V_j^1) é uma aproximação de multiresolução de $L^2(\mathbb{R})$. Pode-se mostrar que

$$\Phi(x,y) = \phi(x) \cdot \phi(y), \tag{2.3.62}$$

onde $\phi(x)$ é a função escala unidimensional da aproximação de multiresolução V_j^1 . Por ser uma aproximação de multiresolução separável, uma importância maior é dada às direções verticais e horizontais da imagem. Assim, para muitos tipos de imagens, como por exemplo imagens de ambientes feitos pelo homem, essa ênfase é apropriada.

A base ortogonal de V_j fica da seguinte forma:

$$2^j \Phi(2^j x - n, 2^j y - m) = 2^j \phi(2^j x - n) \phi(2^j y - m), \tag{2.3.63}$$

e a aproximação do sinal f na resolução 2^j é caracterizada pelo produto interno

$$P_j^d f = \left\langle f(x,y), \phi(2^j x - n) \phi(2^j y - m) \right\rangle_{n,m \in \mathbb{Z}^2}. \tag{2.3.64}$$

Assim como no caso univariado, o detalhe discreto do sinal na resolução 2^j é igual à projeção ortogonal do sinal no complemento de V_j em V_{j+1} .

²¹ Mallat, S. G. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 11, nº 7, pp. 674-693, Jul. 1989.

O seguinte teorema estabelece que podemos construir uma base ortonormal de O_j através de três funções wavelets: $\Psi^1(x,y)$, $\Psi^2(x,y)$ e $\Psi^3(x,y)$.

Teorema 2.3.3 Seja V_j uma aproximação de multiresolução de $L^2(\mathbb{R}^2)$ e $\Phi(x,y) = \phi(x)\phi(y)$ a função escala bidimensional associada. Seja $\psi(x)$ a função unidimensional wavelet associada à função escala $\phi(x)$. Então os três wavelets

$$\Psi^1(x,y) = \phi(x)\psi(y), \quad \Psi^2(x,y) = \psi(x)\phi(y) \text{ e } \Psi^3(x,y) = \psi(x)\psi(y) \tag{2.3.65}$$

são tais que

$$\left(2^j \Psi^1(2^j x - n, 2^j y - m), 2^j \Psi^2(2^j x - n, 2^j y - m), 2^j \Psi^3(2^j x - n, 2^j y - m) \right)_{(n,m) \in \mathbb{Z}^2} \tag{2.3.66}$$

é uma base ortonormal de O_j e

$$\left(2^j \Psi^1(2^j x - n, 2^j y - m), 2^j \Psi^2(2^j x - n, 2^j y - m), 2^j \Psi^3(2^j x - n, 2^j y - m) \right)_{(j,n,m) \in \mathbb{Z}^3} \tag{2.3.67}$$

é uma base ortonormal de $L^2(\mathbb{R}^2)$.

Isto quer dizer que no caso bidimensional o detalhe discreto do sinal no espaço 2^j pode ser escrito como três imagens de detalhe:

$$D_j^{d_1} f = \left\langle f(x,y), \Psi^1(2^j x - n, 2^j y - m) \right\rangle_{(n,m) \in \mathbb{Z}^2} = \left(f(x,y), \phi(2^j x - n)\phi(2^j y - m) \right)_{(n,m) \in \mathbb{Z}^2} \tag{2.3.68}$$

$$D_j^{d_2} f = \left\langle f(x,y), \Psi^2(2^j x - n, 2^j y - m) \right\rangle_{(n,m) \in \mathbb{Z}^2} = \left(f(x,y), \phi(2^j x - n)\psi(2^j y - m) \right)_{(n,m) \in \mathbb{Z}^2} \tag{2.3.69}$$

$$D_j^{d_3} f = \left\langle f(x,y), \Psi^3(2^j x - n, 2^j y - m) \right\rangle_{(n,m) \in \mathbb{Z}^2} = \left(f(x,y), \psi(2^j x - n)\psi(2^j y - m) \right)_{(n,m) \in \mathbb{Z}^2} \tag{2.3.70}$$

Observando as expressões de $P_j^d f$, $D_j^{d1} f$, $D_j^{d2} f$ e $D_j^{d3} f$, podemos ver que a decomposição através de wavelets pode ser calculada através de filtragens separadas ao longo dos eixos x e y . Numa imagem, $P_j^d f$ corresponde às baixas frequências, $D_j^{d1} f$ às altas frequências verticais (bandas horizontais), $D_j^{d2} f$ às altas frequências horizontais (bandas verticais), e $D_j^{d3} f$ às altas frequências em ambas direções (cantos).

Assim, podemos descrever o processo de decomposição como:

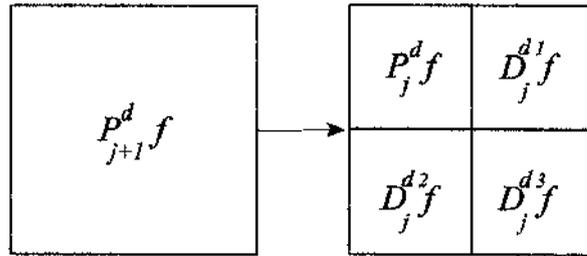


Figura 2.18. Processo de Decomposição de uma Imagem.

tal que

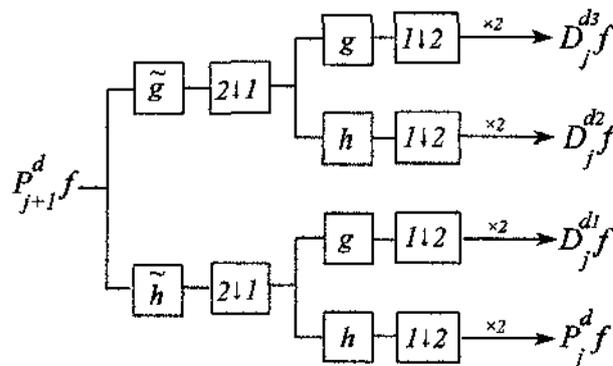


Figura 2.19. Decomposição de uma Imagem f .

onde $2 \downarrow 1$ corresponde a ficarmos somente com as colunas pares e $1 \downarrow 2$ corresponde a ficarmos

somente com as linhas cujos índices são pares.

Se observarmos com cuidado as figuras (2.18) e (2.19), veremos que durante o processo de decomposição de nossa imagem P_j^d na resolução $j+1$, obtemos ao final outra imagem numa resolução menor. Repetindo esse processo para um determinado número de vezes, ao final obteremos os valores de $D_j^{d_1} f, D_j^{d_2} f$ e $D_j^{d_3} f$ em cada resolução e um único P_j^d na resolução final. Este valor de P_j^d na verdade corresponde à imagem original reduzida. Tal fato foi explorado por Rosiek e McLeod (1995)²² em seu artigo onde propõem a utilização da decomposição Wavelet para a construção de "thumbnails" (representação da imagem original em tamanho reduzido) das imagens de um determinado banco de dados.

O processo de reconstrução pode ser descrito como:

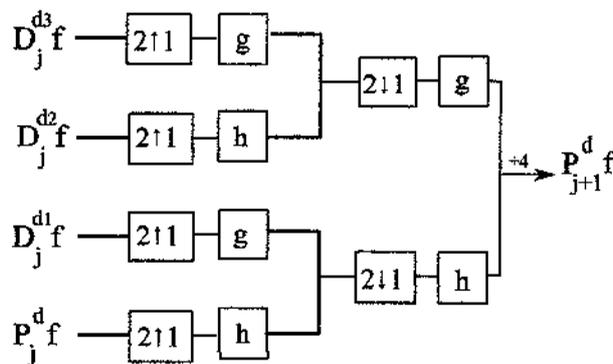


Figura 2.20. Reconstrução de uma Imagem f .

A esta altura, já sabemos como podemos decompor uma imagem utilizando wavelets e como, a partir dos coeficientes obtidos, reconstruí-la novamente. Vamos passar então aos detalhes do método utilizado.

Neste trabalho utilizamos os filtros de quadratura correspondentes à base ortonormal de coiflets. A seguir falaremos um pouco mais de suas propriedades.

²² Rosiek, M. K. e MacLeod, M. L. Daubechies - Based Visual Table of Contents. *Wavelets and Applications I*. Orlando: H. H. Szu (ed), Proc. SPIE 2491, Abr 1995, pp 72 a 77.

Em geral, para uma base ortonormal, quanto maior for o número de momentos de ψ que tenderem a zero maior será a probabilidade de comprimirmos o sinal, já que uma parte dos coeficientes será zero. Nesse caso, temos a seguinte propriedade:

$$\int x^l \psi(x) dx = 0 \quad l = 0, \dots, L-1. \quad (2.3.71)$$

A principal idéia relacionada à base de coiflets é que nesta não somente os momentos de ψ tendem a zero, mas o mesmo acontece com os momentos de ϕ . Assim,

$$\int \phi(x) dx = 1 \quad e \quad \int x^l \phi(x) dx = 0 \quad l = 1, \dots, L-1, \quad (2.3.72)$$

onde L é chamado de ordem do coiflet. No nosso caso, utilizamos $L = 4$. Os valores correspondentes aos coeficientes do filtro, assim como informações adicionais sobre a base de coiflets podem vistos em Daubechies (1991)²³.

Utilizando os conceitos mostrados, vários autores já propuseram métodos de compressão de imagens utilizando wavelets. No entanto, a base utilizada varia de autor para autor. Dentre eles podemos citar os artigos escritos por: Sriram e Marcellin (1995)²⁴, Taswell (1995)²⁵, Zhuang e Baras (1995)²⁶ e Domingues, Gomes e Cortina (1995)²⁷. Agora que já definimos coiflets como a base a ser utilizada, passamos à descrição do algoritmo utilizado neste trabalho.

²³ Daubechies, I., op. cit. p. 45.

²⁴ Sriram, P. e Marcellin, M. Image Coding Using Wavelet Transforms and Entropy - Constrained Trellis Coded Quantization. *IEEE Transactions on Image Processing*. Vol. 4, nº 6, Jun. 1995.

²⁵ Taswell, C. Image Compression by Parameterized Model Coding of Wavelet Packet Near-Best Bases. *Wavelets and Applications I*. Orlando: H. H. Szu (ed), Proc. SPIE 2491, Abr. 1995, pp. 153-172.

²⁶ Zhuang, Y. e Baras, J. S. Image Compression Using Optimal Wavelet Basis. *Wavelets and Applications I*. Orlando: H. H. Szu (ed), Proc. SPIE 2491, Abr. 1995, pp 131-140.

²⁷ Domingues, M. O., Gomes, S. M. e Cortina, E. Biorthogonal Wavelets Applied to Meteosat Image Compressing. *Wavelets and Applications II*. Orlando: H. H. Szu (ed), Proc. SPIE 2491, Abr. 1995, pp 72 6-733.

2.3.1 Algoritmo

- Inicialmente nossa imagem será $P_0 f$;
- Para $0 > j > J$, calculamos as decomposições e vamos armazenando, para cada j , os valores correspondentes a $D_j^{d_1} f$, $D_j^{d_2} f$ e $D_j^{d_3} f$;
- No nosso caso, $J = 3$. Assim, quando $j = J$, teremos vários valores de $D_j^{d_1} f$ para cada resolução e o $P_3^d f$ corresponde à última iteração;
- Ordenamos todos os valores dos D 's e eliminamos um percentual dos coeficientes menores;
- Armazenamos somente os coeficientes restantes e sua resolução correspondente.

Para reconstruirmos a imagem, basta substituímos por zero os coeficientes que não foram armazenados e aplicar o procedimento de reconstrução.

2.3.2 Exemplos

Através da biblioteca de rotinas Wavelab (Buckheit, Chen, Donoho, Johnston e Scargle (1995)²⁸) para o software Matlab, implementamos o algoritmo anterior aproveitando muitas funções pré-definidas pela biblioteca. Sendo assim, utilizando a base de coiflets, aplicamos a decomposição wavelet à imagem original. Em seguida, mantivemos 20%, 10% e 5% dos coeficientes, obtendo três imagens. As imagens reconstruídas utilizando estes coeficientes encontram-se nas figuras (2.21), (2.22) e (2.23). A imagem representada na figura (2.21) será chamada de WAV80, a da figura (2.22) de WAV90 e a da figura (2.23) de WAV95.

²⁸ Buckheit, J. B., Chen, S., Donoho, D., Johnston, I. e Scargle, J. *About Wavelab*. <http://playfer.stanford.edu/pub/wavelab>. 1995.

de WAV90 e a da figura (2.23) de WAV95.



Figura 2.21. Imagem Reconstruída Utilizando 20% dos Coeficientes após a Aplicação da Transformada Wavelet na Imagem Original.



Figura 2.22. Imagem Reconstruída Utilizando 10% dos Coeficientes após a Aplicação da Transformada Wavelet na Imagem Original.



Figura 2.23. Imagem Reconstruída Utilizando 5% dos Coeficientes após a Aplicação da Transformada Wavelet na Imagem Original.

2.4 Técnicas que Utilizam Fractais

Observando a geometria de muitos elementos da natureza, como nuvens e montanhas, vemos que estas não seguem as formas geométricas usuais, tais como cones e esferas. Diante de tal fato, Mandelbrot (1983)²⁹, a fim de descrever esses padrões fragmentados e irregularidades, identificou uma família de formas às quais chamou de fractais ou figuras de dimensões fracionárias. Essas formas tendem a ser escaláveis implicando que o grau de sua irregularidade ou fragmentação é idêntico em todas as escalas (autossimilaridade). Devido a essa característica, os fractais vêm sendo utilizados em diversas áreas de aplicação, principalmente na área de compressão onde já existem comercialmente programas que comprimem imagens (Baran 1993³⁰) e banco de dados (Lewinson 1994³¹).

No nosso trabalho, estamos interessados em fractais autossimilares. Eles são chamados de autossimilares porque, se tomarmos uma imagem fractal desse tipo, cada parte desta será formada por cópias de si mesma levemente alteradas por determinadas transformações e em uma escala menor. A estas transformações damos o nome de transformações afins e, em geral, consistem de uma combinação de operações de rotação, escala, assimetria e translação.

Cada imagem fractal é formada por um conjunto único de transformações afins, podendo ser definido com uma combinação delas. O fato desse número de parâmetros ser pequeno é que torna os métodos de compressão através de fractais métodos competitivos. Um exemplo de fractais é dado pelas figuras a seguir:

²⁹ Mandelbrot, B. B. *The Fractal Geometry of Nature*. New York: Freeman, 1983.

³⁰ Baran, N. Fractal Compression Goes On-Line. *Byte*. Vol. 18, nº 10, p. 40, Set. 1993.

³¹ Lewinson, L. Fractal Databases; New Horizons in Database Management. *PCAI*. Pp. 30-33, Abr. 1994.

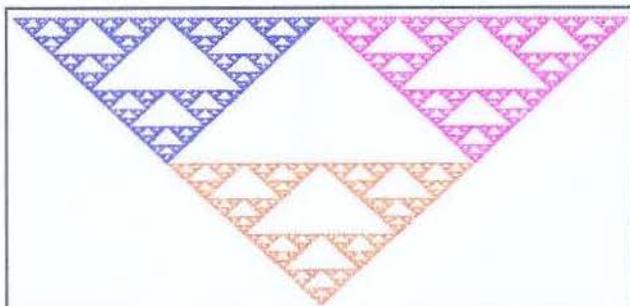


Figura 2.24. Triângulo de Sierpinski.



Figura 2.25. Fern de Sierpinski.

Para vermos como a teoria dos fractais pode ser aplicada ao processamento de imagens, começaremos com algumas definições e teoremas cujas demonstrações serão omitidas por não serem relevantes ao trabalho, mas podem ser encontradas em Fisher (1991)³² e Barnsley (1988)³³.

Uma transformação afim w em \mathbb{R}^2 pode ser definida por:

$$w(x,y) = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} l \\ r \end{pmatrix} \quad (2.4.1)$$

e depende de seis parâmetros a, b, c, d, l e r . No caso de fractais, estamos interessados no estudo de aplicações contrativas. A definição de uma aplicação contrativa é dada pela definição 2.4.1.

Definição 2.4.1. Seja X um espaço métrico com métrica d . Uma aplicação $w: X \rightarrow X$ é contrativa se existe um valor s positivo e real tal que

³² Fisher, Y. *Fractal Image Compression: Theory and Application*. New York: Springer Verlag, 1995.

³³ Barnsley, M. *Fractals Everywhere*. Boston: Academic Press Inc., 1988.

$$d(w(x), w(y)) \leq s \cdot d(x, y) \tag{2.4.2}$$

para todo $x, y \in X$. s é chamado de fator de contratividade.

A partir desta definição, vamos agora introduzir o conceito básico que está por trás da teoria dos fractais, os Sistemas de Funções Iteradas (IFS).

Definição 2.4.2. Seja X um espaço métrico completo. Uma IFS é uma coleção de aplicações contrativas $w_i: X \rightarrow X$, para $i = 1, \dots, n$.

Vamos definir agora o espaço em que vamos trabalhar.

Seja $\mathcal{H}(X) = \{S \subset X \mid S \text{ é compacto}\}$. Se $A \in \mathcal{H}(X)$ então podemos escrever $A_d(\epsilon) = \{x \mid d(x, y) \leq \epsilon \text{ para algum } y \in A\}$, onde $A_d(\epsilon)$ é o conjunto dos pontos que estão a uma distância d -máxima, ϵ de A . Podemos definir a distância de Hausdorff entre dois elementos $A, B \in \mathcal{H}(X)$ como:

$$h_d(A, B) = \max \{ \inf \{ \epsilon \mid B \subset A_d(\epsilon) \}, \inf \{ \epsilon \mid A \subset B_d(\epsilon) \} \}. \tag{2.4.3}$$

Note que h_d depende de uma métrica d .

Voltando ao caso das transformações afins, $w_i: \mathbb{R}^2 \rightarrow \mathbb{R}^2$, $i = 1, \dots, n$ uma coleção de transformações contrativas. Se definirmos um mapa $W: \mathcal{H}(\mathbb{R}^2) \rightarrow \mathcal{H}(\mathbb{R}^2)$ por

$$W(S) = \bigcup_{i=1}^n w_i(S), \tag{2.4.4}$$

desejamos saber se W também é contrativo, já que estamos interessados nessa união. A resposta é dada pelos seguintes teoremas:

Teorema 2.4.1. Seja X um espaço métrico completo com métrica d ; então $\mathcal{H}(X)$, com a métrica de Hausdorff h_d é um espaço métrico completo.

Teorema 2.4.2. Se $w_i: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ é contrativo, com contratividade s_i , para $i = 1, \dots, n$ então $W = \bigcup_{i=1}^n w_i: \mathcal{H}(\mathbb{R}^2) \rightarrow \mathcal{H}(\mathbb{R}^2)$ é contrativo na métrica de Hausdorff com contratividade $s = \max_{i=1, \dots, n} \{s_i\}$.

A partir destes, podemos definir os seguintes teoremas:

Teorema 2.4.3. (Teorema da Aplicação Contrativa para um Ponto Fixo) Seja X um espaço métrico completo e $W: X \rightarrow X$ uma aplicação contrativa. Então existe um único ponto $x_W \in X$ tal que para qualquer ponto $x \in X$

$$x_W = W(x_W) = \lim_{n \rightarrow \infty} W^{on}(x), \tag{2.4.5}$$

onde $W^{on}(x) = W(W(W(\dots W(x)\dots)))$ é resultante da aplicação de W n vezes. Tal ponto é chamado de um ponto fixo ou atrator do mapeamento f .

Teorema 2.4.4. (Teorema Collage) Sejam dados X um espaço métrico completo, $L \in \mathcal{H}(X)$ e $\epsilon \geq 0$. Para cada IFS, $\{X : w_1, \dots, w_n\}$ com fator de contratividade $0 \leq s \leq 1$, tal que $d(L, \bigcup_{i=1}^n w_i(L)) \leq \epsilon$ temos $d(L, L_A) \leq \frac{\epsilon}{1-s}$ onde L_A é o atrator da IFS.

Portanto,

$$d(L, L_A) \leq \frac{d(L, \bigcup_{i=1}^n w_i(L))}{1-s}. \tag{2.4.6}$$

O que esses teoremas nos dizem é que, dado um ponto x_w , se encontrarmos w_i/s tais que $x_w = W(x_w) = \bigcup_{i=1}^n w_i(x_w)$ poderemos, a partir de um x_0 arbitrário, chegar a x_w , fazendo $x_w = \lim_{n \rightarrow \infty} W^{(n)}(x_0)$. Além disso, pelo Teorema Collage, $d(x_0, x_w) \leq \frac{1}{1-s} d(x_0, W(x_0))$.

A unicidade de x_w para cada w é importante, pois tal fato é que vai nos permitir garantir a reconstrução deste ponto somente com a informação dos w_i/s .

Uma função $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ pode ser vista como uma imagem com resolução infinita. Como imagens reais são limitadas, o domínio de f será considerado o quadrado unitário $I^2 = \{(x, y) / 0 \leq x, y \leq 1\}$ e $I = \{x / 0 \leq x \leq 1\}$, com o valor de f representando o nível de cinza. Através da IFS, se encontrássemos w_i/s tais que f fosse o atrator, a partir de uma imagem qualquer poderíamos obter a imagem original. Armazenando somente os w_i/s obteríamos um alto grau de compressão. No entanto, tal procedimento é inviável computacionalmente. No artigo de Barnsley (1988)³⁴, este propõe um método de compressão equivalente ao descrito, onde fica bem claro que o tempo de compressão era extremamente alto.

A partir dessa época, iniciaram-se estudos visando a procura de algoritmos mais eficientes. Observando uma imagem qualquer podemos ver que, em geral, elas apresentam um certo grau de autossimilaridade, e portanto de redundância. Assim, se ao invés de tentarmos reescrever a imagem como cópia de si mesma, a reescreseamos como cópias de suas partes, teríamos um algoritmo mais viável computacionalmente, que ainda assim poderia trazer bons resultados.

Surge, assim a teoria PIFS (Funções de Sistemas Iterados Particionados), em que uma porção da imagem original D_i é utilizada para determinar a aplicação w , onde o atrator de w será R_i , que corresponde a outra porção da imagem.

Neste caso, os w_i/s são da forma

³⁴ Barnsley, M e Sloan, A. D. A Better Way to Compress Images. *Byte*. Vol 13, nº 1, pp.215-223, Jan. 1988.

$$w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} l_i \\ r_i \\ o_i \end{bmatrix}, \tag{2.4.7}$$

onde os parâmetros s_i e o_i que foram acrescentados controlarão o contraste e o brilho da transformação, e $z = f(x,y)$, corresponderá ao nível de cinza nas coordenadas x,y . A aplicação de w_i será dada por $w_i(f) = w_i(x,y,f(x,y))$. Se definirmos

$$v_i(x,y) = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} l_i \\ r_i \end{bmatrix}, \tag{2.4.8}$$

cada v_i determinará como os domínios (D_i) da imagem original serão mapeados para os R_i 's, enquanto s_i e o_i determinarão os ajustes de brilho e contraste necessários para uma boa reconstrução. Cada w_i está restrito a um $D_i \times I$, ou seja, $v_i(D_i) = R_i$. Já que desejamos que $w(f)$ seja uma imagem, devemos ter uma partição $\bigcup_{i=1}^n R_i = I^2$ e $R_i \cap R_j = \emptyset$, para $i \neq j$.

Definição 2.4.3. Seja X um espaço métrico completo, e seja $D_i \subset X$ para $i = 1, \dots, n$. Uma função de sistemas iterados particionados (PIFS) é uma coleção de aplicações contrativas $w_i : D_i \rightarrow X$, $i=1, \dots, n$.

Trabalharemos agora com o espaço das imagens $F = \{f : I^2 \rightarrow \mathbb{R}\}$, definidas como funções mensuráveis sobre I^2 . Utilizando a métrica do supremo dada por

$$d_{sup}(f,g) = \sup_{(x,y) \in I^2} |f(x,y) - g(x,y)|, \tag{2.4.9}$$

temos o seguinte teorema:

Teorema 2.4.5. O espaço F com a métrica d_{sup} é completo.

Sejam agora D_1, \dots, D_n e R_1, \dots, R_n subconjuntos de I^2 aos quais chamaremos de *domínio* e *intervalo*. D_i e R_i serão o domínio e o intervalo de w_i que, na realidade, correspondem a $D_i \times I$ e $R_i \times I$.

Definição 2.4.4. As aplicações w_1, \dots, w_n cobrem I^2 se, para todo $f \in F$, $\bigcup_{i=1}^n w_i(f) \in F$.

Isto quer dizer que quando aplicamos w_i a uma parte da imagem, $f \cap (D_i \cap I)$, o resultado deve ser uma função sobre R_i .

Definição 2.4.5. Se $w : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ é um mapa com $(x', y', z'_1) = W(x, y, z_1)$ e $(x', y', z'_2) = W(x, y, z_2)$, então w é chamado de Z-contrativo se existe um número positivo real $s < 1$ tal que $|z'_1 - z'_2| \leq s \cdot |z_1 - z_2|$ e se x' e y' são independentes de z_1 ou z_2 para todo x, y, z_1, z_2 .

Lema 2.4.1. Se w_1, \dots, w_n são Z-contrativos, então $W = \bigcup_{i=1}^n w_i$ é contrativo em F com a métrica suprema.

Com esses resultados podemos aplicar PIFS para codificar uma imagem calculando os w_i 's a partir da autossimilaridade entre as sub-imagens D_i e R_i . Se os w_i 's forem Z-contrativos, o lema 2.4.1 nos garantirá que $W = \bigcup_{i=1}^n w_i$ define um atrator único $f_N \in F$. Portanto, a partir de uma imagem f_0 arbitrária,

$$f = \lim_{n \rightarrow \infty} W^{on}(f_0) = W(f) = \bigcup_{i=1}^n w_i(f). \tag{2.4.10}$$

Basta armazenarmos os w_i 's que, assim fazendo, já estaremos comprimindo a imagem.

Uma das maiores vantagens de fazermos isso é que os w_i 's não vão depender da resolução da imagem original.

Durante o trabalho, foram utilizados dois métodos diferentes. Como o algoritmo deles é muito parecido, daremos os detalhes dos métodos na seção de algoritmos.

2.4.1 Algoritmos

Os métodos de compressão através de fractais seguem basicamente o mesmo algoritmo. As diferenças que existem entre eles é o método de implementação de cada passo, que pode levar a algoritmos mais rápidos e eficientes.

O algoritmo básico para a compressão de uma imagem através de fractais pode ser descrito como:

- Particionamos a imagem em blocos (R_i 's) tais que $\bigcup_{i=1}^n R_i = I^2$;
- Seleccionamos os blocos D_i 's que serão utilizados;
- Para cada R_i , escolhemos os valores de D_i e w_i que melhor "cobrem" R_i , minimizando

$$d_{EQM}(f \cap (R_i \times I), w_i(f)). \tag{2.4.11}$$

- Ao final, armazenamos os valores dos w_i 's escolhidos.

Note que $d_{EQM}(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$ corresponde ao erro quadrático médio. Falaremos mais dele no próximo capítulo.

Em geral, o cálculo de w_i é feito procurando-se os valores s_i e o_i e a operação de simetria (rotações) de tal forma que se aplicarmos essas transformações em D_i , teremos a melhor

aproximação para R_i .

Para a reconstrução, basta a partir de uma imagem qualquer aplicarmos iterativamente cada w_i na posição correspondente ao D_i utilizado para seu cálculo.

O problema do estudo da convergência dos w_i 's para a reconstrução da imagem foi estudado por Hürtgen e Simon (1994)³⁵, Davis (1995)³⁶ e Oien (1994)³⁷, preocupados com esse mesmo problema, utilizaram no momento da codificação uma constante dependente do tamanho do bloco (chamada de componente DC), que consistia em subtrairmos de cada pixel do bloco D_i o valor correspondente à sua média. Fazendo isto, a principal vantagem é que ao reconstruirmos uma imagem o número de iterações não dependeria da imagem, e sim do tamanho dos blocos D_i e R_i e variaria entre 1 e 4. No entanto, os métodos utilizados por estes autores não faziam estudos sobre a contratividade dos w_i 's e o tamanho dos R_i 's e D_i 's era fixo.

Os dois métodos utilizados no trabalho não removem a componente DC. No entanto, suas performances foram satisfatórias. A descrição detalhada de cada um está a seguir e, apesar de trabalharmos apenas com esses dois, existem outros parecidos, dentre os quais podemos citar o apresentado por Barthel, Schüttemeyer, Voyé e Noll (1994)³⁸ em seu artigo. Tal método também apresenta uma boa performance. Antes de começar a descrição dos métodos, vamos definir a operação de média, utilizada por ambos, como a média de cada bloco de 4 pixels, como mostrado a seguir:

³⁵ Hürtgen, B. e Simon, S. F. On the Problem of Convergence in Fractal Coding Schemes. *IEEE International Conference on Image Processing ICIP'94*. Austin, Texas, Nov. 1994, pp. 103-106.

³⁶ Davis, G. Self-Quantized Wavelet Subtrees: A Wavelet-Based Theory of Fractal Image Compression. *Wavelets and Applications I*. Orlando: H. H. Szu (ed), Proc. SPIE 2491, Abr. 1995, pp. 141-152.

³⁷ Oien, G. E. Parameter Quantization in Fractal Image Coding. *IEEE International Conference on Image Processing ICIP'94*. Austin, Texas, Nov. 1994, pp. 142-146.

³⁸ Barthel, K. U., Schüttemeyer, J., Voyé, T. e Noll, P. A New Image Coding Technique Unifying Fractal and Transform Coding. *IEEE International Conference on Image Processing ICIP'94*. Austin, Texas, Nov. 1994, pp. 112-116.

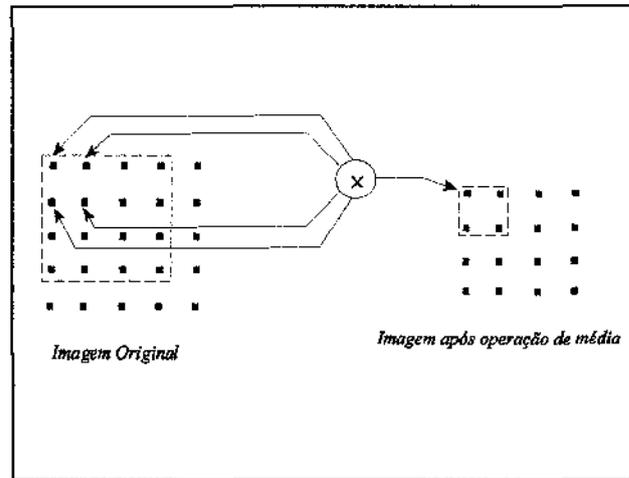


Figura 2.26. Operação de Média.

1º Método

A principal característica deste método é que os R_i 's são calculados através da decomposição da imagem utilizando quadrees.

A decomposição quadtree é uma técnica que divide a imagem em regiões "homogêneas" (clusters) no plano, produzindo a segmentação. A técnica constrói uma árvore na qual cada folha corresponde a uma região homogênea.

Sua aplicação à compressão de imagens pode ser vista no artigo de Shusterman e Feder (1994)³⁹. No nosso caso, ela será apenas uma ferramenta para iniciarmos a compressão.

Em geral a decomposição é feita da seguinte forma:

A imagem original corresponde ao nível 0 de profundidade da árvore. Inicialmente, subdividimos a imagem em 4 regiões iguais. Em seguida, realizamos um teste de homogeneidade que na maioria das aplicações corresponde ao teste da diferença absoluta e pode ser descrito como:

³⁹ Shusterman, E. e Feder, M. Image Compression Via Improved Quadtree Decomposition Algorithms. *IEEE Transactions on Image Processing*. Vol. 4, nº 6, pp. 207-215, Mar. 1994.

Seja $\bar{f}_i(k,l)$ o nível médio de cinza da imagem no nível i e coordenadas k e l . O teste é dado por:

$$\bigcap_{j,m=0}^1 \{|\bar{f}_i(k,l) - \bar{f}_{i-1}(2k+j,2l+m)| \leq T\}, \tag{2.4.12}$$

onde T é um limiar predeterminado, $i = 1, \dots, n$ para uma imagem $2^n \times 2^n$, $k, l = 0, \dots, 2^{n-1} - 1$ e a expressão acima é uma expressão booleana. Se o resultado dessa expressão for "verdadeiro" a região é homogênea e mantemos a subdivisão do nível anterior. Caso contrário, cada uma das 4 subregiões é subdividida em outras 4 regiões iguais, aumentando assim mais um nível, e refazemos o teste. Um exemplo desse algoritmo pode ser visto na figura (2.27).

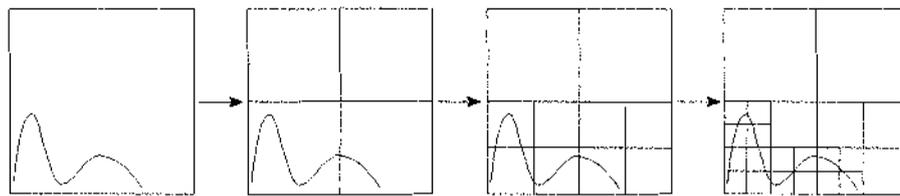


Figura 2.27. Método Quad-tree.

Neste método de compressão com fractais, inicialmente subdividimos a imagem no mínimo m níveis, ao invés de realizarmos o teste de homogeneidade, onde m é especificado sendo em geral $m = 4$. Cada subdivisão da imagem corresponde a um R_i . Os quadrados nos nós (R_i 's) são então comparados com os domínios na biblioteca de domínios D , os quais possuem o dobro do tamanho.

Antes de iniciarmos a comparação, submetemos os D_j 's à operação de média descrita; assim, eles vão estar normalizados com o valor de R_i . Em seguida, procuramos o R_i tal que minimize-se $d_{EQM}(f \cap (R_i \times I), w_i(f))$.

Se $d_{EQM}(f \cap (R_i \times I), w_i(f)) < e_c$, onde e_c é a precisão desejada, particionamos o bloco R_i em mais um nível e repetimos o procedimento com cada nova subregião. O valor de e_c é

fornecido pelo próprio usuário.

Ao procurarmos o valor de D_j que minimiza o erro acima, não precisamos testar todos D_j 's, basta testarmos os potencialmente prováveis. Para isso, os classificamos da seguinte forma:

Podemos orientar o bloco D_i de tal forma que seu brilho ou valores médios dos quadrantes caiam em três posições canônicas:

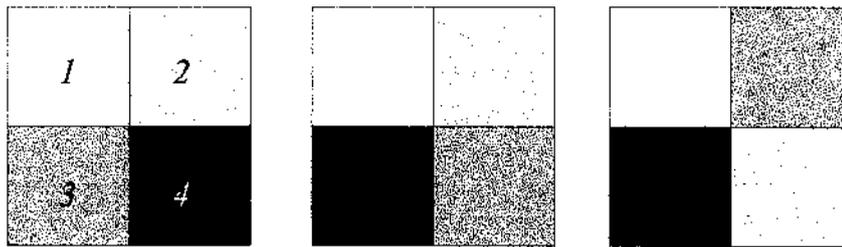


Figura 2.28. Três Posições Canônicas Para Ordenar Imagens Quadradas.

Se os valores dos pixels no quadrante i ($i = 1, 2, 3, 4$) forem r_1^i, \dots, r_n^i , calculamos $A_i = \sum_{j=1}^n r_j^i$ e $V_i = \sum_{j=1}^n (r_j^i)^2 - A_i^2$.

Assim, as classes possuem as seguintes características:

classe 1: $A_1 \geq A_2 \geq A_3 \geq A_4$

classe 2: $A_1 \geq A_2 \geq A_4 \geq A_3$

classe 3: $A_1 \geq A_4 \geq A_2 \geq A_3$

e dentro delas existem 24 ordenações possíveis para V_i .

Com cada D_i classificado, tomamos o R_i , o classificamos e o comparamos somente com aqueles que possuem classificação mais próxima.

Com os D_i 's potencialmente prováveis calculados, ficamos com os valores de o_i e s_i que minimizem $d_{EQM}(f \cap (R_i \times I), w_f(f))$.

Como repetiremos esse processo para todos os R_i 's da imagem, se ao final $|s_i| > 1$ para

algum w_i , existirá uma chance de que a aplicação final w_i não seja contrativa. Por esse motivo, os $|s_i|$ maiores que algum valor máximo $s_{máx}$ são truncados.

Como uma última característica deste método, a escolha dos D'_i 's na imagem é feita tomando-se o D_i cujo espaçamento horizontal entre eles é o tamanho de D_i dividido por um parâmetro l .

Após aplicarmos todos esses procedimentos na imagem, a imagem codificada consistirá de:

- partição quadtree final (que pode ser guardada como uma árvore binária);
- os valores s_i e o_i de cada R_i ;
- para cada R_i , o domínio mapeado;
- a operação de simetria para mapear os D_i 's nos R_i 's.

Para decodificarmos a imagem, basta iterarmos W a partir de uma imagem arbitrária original.

Este método já se encontrava implementado em linguagem C e as fontes estavam disponíveis na *HomePage* de Youval Fisher⁴⁰, em cujo livro⁴¹ se encontram os algoritmos. Para gerarmos nossas imagens comprimidas, utilizamos $l = 1$, e variamos o valor de $e_i = 2, 3$ e 4 , obtendo-as assim com qualidade distintas.

Este método mostrou-se rápido e eficiente. Para decodificarmos as imagens comprimidas, utilizamos como inicial uma imagem onde todos os elementos eram 0 e aplicamos 10 iterações.

2º Método

Este segundo método é relativamente simples, comparado ao anterior. Foi implementado

⁴⁰ <http://nls.ucsd.edu/y/Fractals/>

⁴¹ Fisher, Y., op. cit. p. 70.

por Anson (1993)⁴², e sua performance em termos de tempo de compressão é bem inferior ao anterior. Em nosso trabalho, o método demorou 70 horas para compactar a imagem original.

Neste método a imagem é subdividida em blocos de tamanho $2^2 \times 2^2$, e cada um destes corresponde aos R_i 's. Os D_j 's são obtidos da seguinte forma: aplicamos a operação de média à imagem original e multiplicamos a intensidade final por $\frac{3}{4}$. Tomamos essa imagem final e subdividimos em blocos de tamanho $2^2 \times 2^2$, obtendo assim os D_j 's.

Com os R_i 's e os D_j 's para cada R_i , somamos à intensidade dos pixels de D_j o valor correspondente à média de R_i menos a média desse D_j e calculamos a rotação que minimiza o erro quadrático médio entre o D_j rotacionado e R_i .

Repetimos esse procedimento com todos os D_j disponíveis, e ao encontrarmos o D_j que minimizou o erro quadrático médio, guardamos:

- o valor da média de R_i menos a média do D_j ;
- as coordenadas de D_j ;
- a rotação utilizada.

A compressão é então alcançada, guardando-se somente essas informações após repetirmos esse procedimento para todos os R_i 's.

2.4.2 Exemplos

Utilizando o primeiro método de compressão através de fractais que utiliza partições calculadas com a aplicação do método quadtree, geramos três imagens comprimidas. Para variarmos a qualidade da imagem final, utilizamos três valores distintos de e_c : 2, 3 e 4. A partir das imagens comprimidas, desejávamos descomprimi-las para observar sua qualidade. Utilizando como imagem inicial para as iterações uma imagem completamente "preta", contendo apenas valores iguais a zero,

⁴² Anson, L. F. Fractal Image Compression. *Byte*. Vol. 18, nº 11, pp. 195-202, Nov. 1993.

e fixando o número de iterações igual a 10, obtivemos as imagens mostradas nas figuras (2.29), (2.30) e (2.31).

Em seguida, comprimimos a imagem utilizando o método 2. A partir da imagem comprimida, para reconstruirmos a imagem com qualidade distintas, variamos o número de iterações, gerando assim três imagens correspondentes a 5, 6 e 8 iterações. A imagem inicial utilizada foi a mesma do método anterior. O resultado pode ser visto nas figuras (2.32), (2.33) e (2.34).

A imagem da figura (2.29) será referenciada por FRAC2, a da figura (2.30) por FRAC3, a da figura (2.31) por FRAC4, a da figura (2.32) por FRC8, a da figura (2.33) por FRC6 e a da figura (2.34) por FRC5 nos capítulos seguintes.



Figura 2.29. Imagem Reconstruída após a Compressão da Imagem Original Utilizando o Método 1 (Fractal) com $e_c = 2$.



Figura 2.30. Imagem Reconstruída após a Compressão da Imagem Original Utilizando o Método 1 (Fractal) com $e_c = 3$.



Figura 2.31. Imagem Reconstruída após a Compressão da Imagem Original Utilizando o Método 1 (Fractal) com $\epsilon_c = 4$.



Figura 2.32. Imagem Reconstruída após a Compressão da Imagem Original Utilizando o Método 2 (Fractal) depois de 8 Iterações.



Figura 2.33. Imagem Reconstruída após a Compressão da Imagem Original Utilizando o Método 2 (Fractal) depois de 6 Iterações.



Figura 2.34. Imagem Reconstruída após a Compressão da Imagem Original Utilizando o Método 2 (Fractal) depois de 5 Iterações.

Capítulo 3

Avaliando a Qualidade da Compressão

3.1 Introdução

Neste capítulo vamos descrever as medidas estatísticas estudadas juntamente com a explicação de como foram utilizadas na análise. Como não se tinha conhecimento de estudos feitos anteriormente, relacionados à busca de outras medidas estatísticas para a avaliação da qualidade das imagens além da utilização do Erro Quadrático Médio (EQM), iniciamos a procura de tais medidas a partir de quatro grupos de medidas, medidas de distância, medidas para análise descritiva, medidas de informação e medidas de avaliação de texturas e bordas. Antes, no entanto, falaremos um pouco do PNSR, medida clássica utilizada na maioria dos artigos relacionados à compressão de imagens.

3.2 Medidas Clássicas: PNSR

O PNSR é uma medida de erro obtida a partir do tradicional Erro Quadrático Médio (EQM), porém calculado na escala logarítmica:

$$PSNR = 10 \log \left(\frac{255^2}{EQM} \right). \tag{3.2.1}$$

Por esse motivo, vamos inicialmente falar um pouco mais detalhadamente do EQM. No capítulo 2 vimos que nossa imagem pode ser representada por uma matriz $M \times N$. Vamos representá-la agora através de um vetor $n = N \times M$ dimensional. Em geral a natureza discreta dos pixels é ignorada assim, estamos trabalhando agora no espaço \mathbb{R}^n . As normas mais comuns em \mathbb{R}^n são as p-normas definidas por:

$$\|x_p\| = (|x_1|^p + \dots + |x_n|^p)^{\frac{1}{p}} \tag{3.2.2}$$

que definem uma métrica através de $d_p(x,y) = \|x-y\|_p$. A norma 2 é também chamada de norma l_2 ou raiz do erro quadrático (REQ) e é dada por

$$d_{REQ}(x,y) = \|X-Y\|_2 = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}. \tag{3.2.3}$$

No nosso caso, estamos interessados no erro quadrático médio e este pode ser definido como:

$$EQM = \frac{d_{REQ}^2}{n} = \frac{\sum_{i=1}^n (X_i - Y_i)^2}{n}. \tag{3.2.4}$$

Foi utilizando a equação (3.2.4) que calculamos durante o trabalho o erro quadrático médio entre a imagem original e a reconstruída.

Num outro tipo de abordagem, seja f_1 a imagem a ser comparada com f , a imagem de referência. Sob a suposição que f_1 e f representam amostras de um processo estocástico, o EQM

entre duas imagens pode ser definido como $E_{EQM} = E\{|f_1 - f|^2\}$, onde $E\{.\}$ é o operador esperança. Mesmo assim, ao final o cálculo do EQM é feito através da equação (3.2.4).

Através do cálculo do EQM, podemos obter o PNSR, que, na verdade, é a nossa medida de erro em termos de razão sinal-ruído (em decibéis). Esta medida vem sendo utilizada atualmente para avaliar os métodos de compressão de imagens existentes.

3.3 Medidas de Distância e Informação

Desde o início, observando as imagens produzidas pelos métodos, já sabíamos que umas eram bem piores que outras. Diante de tal fato, surgiu a questão: que tipo de medida poderia descrever a diferença que, por vezes, visualmente percebíamos ser bem grande? A primeira idéia foi que, se duas imagens (a original e a reconstruída) diferem ou não em qualidade, poderíamos associar essa diferença a distância. Isto quer dizer que se duas imagens são muito diferentes, elas são bem distantes em algum sentido objetivo. Seguindo este tipo de abordagem, começamos a procura de algumas medidas de distância.

Medidas de distância ou de dissimilaridade são encontradas em problemas de análise multivariada onde, em geral, são utilizadas para o cálculo de distância entre observações multivariadas de duas populações. De acordo com Mardia (1979)⁴³, distância pode ser definida como:

Definição: Sejam P e Q dois pontos onde estes podem representar medidas x e y em dois objetos. Uma função com valores reais $d(P, Q)$ é uma função de distância se possui as seguintes propriedades:

- (i) simetria $d(P, Q) = d(Q, P)$.

⁴³ Mardia, K. V., Kent, J. T. e Bibby J. M. *Multivariate Analysis*. London: Academic Press, 1979.

- (ii) não negatividade $d(P, Q) \geq 0$.
- (iii) $d(P, P) = 0$.

Para algumas funções de distância, as seguintes propriedades também valem:

- (iv) $d(P, Q) = 0$ se e somente se $P=Q$.
- (v) desigualdade triangular: $d(P, Q) \leq d(P, R) + d(R, Q)$ onde R é um outro ponto.

Se as propriedades (i)-(v) valem a distância é chamada de métrica.

Em geral, as medidas de distância satisfazem essas condições e em alguns casos, podem satisfazer também a desigualdade ultra-métrica:

$$d(P, Q) \leq \max(d(P, R), d(R, Q)) \text{ para todo } i, j, k. \quad (3.3.1)$$

Neste estudo, utilizamos uma distância que não era métrica para dados positivos, a distância de Bray e Curtis. Além disso, algumas das distâncias utilizadas perdem sua propriedade métrica para dados negativos.

3.3.1 Abordagem por Blocos

A fim de observarmos com mais precisão a qualidade das imagens estudadas, aplicaremos medidas de distância a cada bloco de tamanho $2^4 \times 2^4$ da imagem de tal forma que esses blocos cubram a imagem inteira sem se sobreporem. Seria interessante, em nosso estudo, medir a distância entre a imagem original e as imagens reconstruídas. Para fazer isso, aplicaremos as medidas para avaliar a distância entre cada bloco na imagem original e na imagem final, obtendo uma matriz de medidas de distâncias. As medidas utilizadas se encontram na tabela (3.3.1.1). Note que b corresponde ao número de elementos no bloco e i, j aos blocos nas imagens.

Essas medidas foram obtidas em Gower (1985)⁴⁴ e mais informações sobre estas podem ser vistas em Gower e Legendre (1986)⁴⁵.

Tabela 3.3.1.1. Medidas de Distância Utilizadas.

Medidas de Distância	
Nome	Fórmula
Distância 1	$D = \frac{\sum_{k=1}^b (x_{ik} - x_{jk})^2}{b}$
Minkowski (t=1,5)	$D = \frac{1}{b} \frac{\sum_{k=1}^b x_{ik} - x_{jk} ^{1,5}}{r_k^{1,5}}$
Canberra Metric	$D = \sum_{k=1}^b \frac{ x_{ik} - x_{jk} }{ x_{ik} + x_{jk} }$
Divergência	$D = \frac{1}{b} \sum_{k=1}^b \frac{(x_{ik} - x_{jk})^2}{(x_{ik} + x_{jk})^2}$
Bray e Curtis	$D = \frac{\sum_{k=1}^b x_{ik} - x_{jk} }{\sum_{k=1}^b (x_{ik} + x_{jk})}$
Soergel	$D = \frac{\sum_{k=1}^b x_{ik} - x_{jk} }{\sum_{k=1}^b \max(x_{ik}, x_{jk})}$
Ware e Hedges	$D = \frac{1}{b} \sum_{k=1}^b \left(1 - \frac{\min(x_{ik}, x_{jk})}{\max(x_{ik}, x_{jk})} \right)$

⁴⁴ Gower, J. C. Measures of Similarity, Dissimilarity and Distance. *Encyclopaedia of Statistical Sciences*. Vol. 5, Eds. S. Katz, N. L. Johnson and C. B. Read. New York: John Wiley, pp. 397-405, 1985.

⁴⁵ Gower, J. C. e Legendre, P. Metric and Euclidian Properties of Dissimilarity Coefficients. *Journal of Classification*. Vol. 3, pp. 397-405, 1986.

3.3.2 Abordagem Geral

Vamos agora observar a distância entre as imagens inteiras, não mais entre cada bloco das imagens. Para cada imagem, cada bloco será considerado uma variável, assim, estaremos medindo a distância entre duas populações multivariadas. Para tal, utilizaremos a medida de Mahalanobis. A distância de Mahalanobis pode ser definida como:

$$D = (\mu_1 - \mu_2)' \Sigma^{-1} (\mu_1 - \mu_2), \tag{3.3.2.1}$$

que, em termos dos valores amostrais pode ser reescrita como:

$$T^2 = (\bar{x}_1 - \bar{x}_2)' S_u^{-1} (\bar{x}_1 - \bar{x}_2) \tag{3.3.2.2}$$

onde $S_u = \frac{n_1 S_1 + n_2 S_2}{n_1 + n_2 - 2}$ é a estimativa de Σ .

Sabemos que a hipótese de normalidade dos dados não é válida para nosso caso, mas como não estamos interessados na distribuição da nossa estatística nem queremos fazer nenhum teste com ela, tal fato não constitui problema. Nos limitaremos a apenas observar o valor da estatística.

3.3.3 Medidas de Informação

A partir de dois conjuntos de dados, muitas vezes desejamos saber se suas distribuições temporais ou espaciais são parecidas. Se a relação que existe nesses conjuntos é linear, o coeficiente de correlação linear pode nos dizer se as distribuições são próximas ou não. Já quando essa relação não é linear, este método não é adequado. Nesse caso, procuram-se outras estatísticas que possam nos fornecer tal informação e, para tal, existem algumas medidas de proximidade pertencentes à área de Teoria da Informação que medem a proximidade da forma de duas distribuições.

Em nosso caso, se observarmos como os tons de cinza se distribuem na imagem original e na imagem reconstruída, poderemos medir a proximidade entre essas imagens e assim obter uma medida desta proximidade. Vamos trabalhar com nossa imagem dividida em blocos. Sabemos que a cada ponto observamos a intensidade $f(x,y)$, assim, se dividirmos esse valor pelo total observado do bloco, vamos ter a probabilidade de ocorrência dessa intensidade.

Repetindo o processo para todos os pontos de cada bloco, vamos ter a distribuição da intensidade deste e, desta forma, basta medirmos a proximidade entre o mesmo bloco na imagem original e na reconstruída. As medidas de proximidade utilizadas encontram-se na tabela a seguir.

Como podemos ver na tabela (3.3.2.1), algumas dessas medidas são medidas de distância e podem ser interpretadas como tal. A medida de afinidade pode ser vista como o $\cos \theta$, onde θ é o ângulo entre os vetores quando representamos a distribuição de duas populações como pontos no círculo unitário. A medida de imprecisão é, em geral, uma medida de informação do experimento e a divergência dirigida pode ser vista como uma medida da informação média ganha no experimento. No nosso caso, esse ganho será representado pela utilização de determinado método de compressão.

Todas estas medidas de informação se encontram em Mathai e Rathie (1973)⁴⁶ e Bonvino (1993)⁴⁷.

⁴⁶ Mathai, A. M. e Rathie, P. N. *Basic Concepts in Information Theory and Statistics*. New Delhi: Wiley Eastern Limited, 1975.

⁴⁷ Bonvino, H. *Notas de Aula fornecidas durante o curso de Teoria de Informação*. Campinas: Unicamp, Mar. 1993.

Tabela 3.3.2.1. Medidas de Informação Utilizadas.

Medidas de Proximidade	
Nome	Fórmula
Afinidade	$D = \sum_{i=1}^b \sqrt{p_i q_i}$
Média Harmônica (ou Atração)	$D = \sum_{i=1}^b \frac{2p_i q_i}{p_i + q_i}$
Distância 1 (Mathai)	$D = \frac{1}{b} \sum_{i=1}^b p_i - q_i $
Distância 2 (Mathai)	$D = \max_i p_i - q_i $
Distância 3 (Matusita)	$D = \frac{1}{b} \sqrt{2(1 - \sum_{i=1}^b \sqrt{p_i q_i})}$
Discrepância	$D = \sum_{i=1}^b \frac{q_i^2}{p_i} - 1$
Imprecisão	$D = \sum_{i=1}^b p_i \log_2 q_i$
Divergência Dirigida	$D = \sum_{i=1}^b p_i \log_2 \frac{p_i}{q_i}$
Distância de Matusita	$D = \sqrt{\sum_{i=1}^b (\sqrt{p_i} - \sqrt{q_i})^2}$

3.4 Medidas para Análise Descritiva

Além das medidas descritas anteriormente, resolvemos calcular também algumas estatísticas clássicas muito utilizadas para compararmos conjuntos de dados, tanto com relação à média quanto com relação à variância. As estatísticas foram aplicadas a cada bloco e a descrição do

modo como foram utilizadas encontra-se a seguir.

- Chi-Quadrado

O teste Chi-Quadrado é muito utilizado para verificarmos se determinado conjunto de dados pertence a certa distribuição. Consideremos a nossa imagem original como sendo a distribuição de referência. Cada imagem reconstruída irá conter os valores observados. Portanto, calculamos para cada bloco:

$$C = \sum_{i=1}^b \frac{(O_i - R_i)^2}{O_i}, \quad (3.4.1)$$

onde O_i é o valor do i -ésimo pixel do bloco da imagem original e R_i é o correspondente na imagem reconstruída.

- Correlação

Como já foi citado no item anterior, mede a proximidade entre duas populações, mas só é adequada caso a relação entre os dados seja linear.

$$C = \frac{\sum_{i=1}^b (O_i - \bar{O})(R_i - \bar{R})}{\sqrt{\sum_{i=1}^b (O_i - \bar{O})^2 \sum_{i=1}^b (R_i - \bar{R})^2}}. \quad (3.4.2)$$

- F para variâncias

Caso a imagem reconstruída seja visualmente bem parecida com a imagem original, devemos imaginar que a variância nos blocos também o seja. Calculando o valor da estatística F para comparação de variâncias esperamos que à medida que as imagens reconstruídas "piorem", o seu valor se altere significativamente. Para calcular-se o F basta fazermos:

$$F = \frac{S_R^2}{S_O^2}, \quad (3.4.3)$$

onde S_O^2 é a variância do bloco na imagem original e S_R^2 na imagem reconstruída.

- Teste t para dados pareados

Como na verdade cada bloco na imagem original e na imagem reconstruída são um conjunto de dados pareados, ao calcularmos o teste t para verificar se a diferença entre esses dados é quase nula ou não, podemos encontrar uma medida útil para avaliar a qualidade final das imagens. O cálculo do t é feito da seguinte forma:

$$t = \frac{\frac{\sum_{i=1}^b (O_i - R_i)}{b}}{\sqrt{\frac{(S_O^2 + S_R^2)}{b-1}} \sqrt{\frac{2}{b}}}. \quad (3.4.4)$$

3.5 Medidas para Avaliar Bordas

Como já pudemos ver no capítulo 2, ao aplicarmos os métodos de compressão na imagem original, procuramos produzir imagens com níveis de qualidade distintos. À medida que a qualidade piorava, muitos detalhes das imagens iam sendo perdidos e, dentre eles, as bordas. Na maioria dos métodos, quanto menor a qualidade das imagens, pior a definição de suas bordas. Por este motivo, vimos que, se observássemos as mudanças na qualidade destas, talvez fosse possível encontrarmos uma medida de distorção.

A fim de avaliarmos as bordas, era necessário antes de mais nada que utilizássemos algum procedimento capaz de detectá-las. Em processamento de imagens existem vários algoritmos que,

uma vez aplicados, produzem uma nova imagem contendo as informações das bordas. No entanto, a maior parte destes algoritmos introduz ruído juntamente com a informação desejada. Neste trabalho, utilizamos três métodos: o de detector de bordas finas, chapéu mexicano e método de Roberts.

O método detector de bordas finas é composto por duas fases. Na primeira, para cada pixel procuramos saber se seu nível de cinza em relação aos pixels vizinhos variou mais que um determinado limiar que, no nosso caso, foi igual a 8. Se isto se deu na direção horizontal ou vertical este pixel pertencerá à borda e receberá valor igual a 1. Caso essa variação não alcance o limiar, o pixel receberá valor zero. Após esse procedimento, nossa imagem será formada por bordas "grossas". Na segunda fase, passamos na direção horizontal e em seguida na vertical, marcando como pontos de borda apenas os pontos correspondentes à média aritmética do ponto inicial e do final das áreas cuja variação dos pixels consecutivos ultrapassou o limiar. A aplicação desse algoritmo na imagem inicial pode ser visto na figura (3.1).



Figura 3.1. Imagem após a Aplicação do Algoritmo de Detecção de Bordas Finas.

O método do Chapéu Mexicano consiste em aplicarmos na imagem uma máscara de convolução dada pela seguinte matriz:

$$\begin{bmatrix}
 0 & 0 & 0 & -1 & -1 & -2 & -1 & -1 & 0 & 0 & 0 \\
 0 & 0 & -2 & -4 & -8 & -9 & -8 & -4 & -2 & 0 & 0 \\
 0 & -2 & -7 & -15 & -22 & -23 & -22 & -15 & -7 & -2 & 0 \\
 -1 & -4 & -15 & -24 & -14 & -1 & -14 & -24 & -15 & -4 & -1 \\
 -1 & -8 & -22 & -14 & 52 & 103 & 52 & -14 & -22 & -8 & -1 \\
 -2 & -9 & -23 & -1 & 103 & 178 & 103 & -1 & -23 & -9 & -2 \\
 -1 & -8 & -22 & -14 & 52 & 103 & 52 & -14 & -22 & -8 & -1 \\
 -1 & -4 & -15 & -24 & -14 & -1 & -14 & -24 & -15 & -4 & -1 \\
 0 & -2 & -7 & -15 & -22 & -23 & -22 & -15 & -7 & -2 & 0 \\
 0 & 0 & -2 & -4 & -8 & -9 & -8 & -4 & -2 & 0 & 0 \\
 0 & 0 & 0 & -1 & -1 & -2 & -1 & -1 & 0 & 0 & 0
 \end{bmatrix}$$

A aplicação de um filtro de convolução em uma imagem é feita da seguinte maneira: seja A uma matriz de convolução de dimensões $(p+1) \times (p+1)$, onde p é par e, uma imagem de dimensão $N \times M$. Para todo (x,y) tais que $x = p, \dots, N-p-1$ e $y = p, \dots, M-p-1$, fazemos:

$$f'(x,y) = \sum_{k=1}^{p+1} \sum_{l=1}^{p+1} a_{kl} f(x+k-p, y+l-p). \tag{3.4.1}$$

A função conhecida como Chapéu Mexicano (cujo nome está associado à sua forma) na verdade corresponde ao operador Laplaciano-Gaussiano, que pode ser definido por:

$$\nabla^2 G(x,y) = \frac{1}{2\pi\sigma^4} \left(2 - \frac{x^2+y^2}{\sigma^2} \right) \exp \left(-\frac{x^2+y^2}{2\sigma^2} \right). \tag{3.4.2}$$

A parte Gaussiana desse operador equivale a uma função de suavização e o Laplaciano nos fornece uma derivação não-direcional. A utilização desse operador, sugerida por Marr e Hildreth

(1980)⁴⁸ e estudada por Nevatia⁴⁹ e Pratt (1991)⁵⁰ está relacionada à idéia da procura de um operador que fosse derivativo de segunda ordem e pudesse localizar bordas em perfis de intensidade inclinados. A visão discreta desse operador que corresponde à matriz de convolução pode ser obtida amostrando a função dada pela equação (3.4.2) sobre uma "janela" de dimensão $(p+1) \times (p+1)$. No nosso caso, usamos $p = 10$, como utilizada por De Hoyos e Corbera (1991)⁵¹ com resultados satisfatórios. A aplicação desse algoritmo na imagem original pode ser vista na figura (3.2).



Figura 3.2. Imagem após a Aplicação do Algoritmo do Chapéu Mexicano para Detecção de Bordas.

⁴⁸ Marr, D. e Hildreth, E. Theory of Edge Detection. *Proc. of Royal Society London*. Vol. B207, pp. 187-217, 1980.

⁴⁹ Nevatia, R. Image Segmentation. Young, T. Y. e Fu, K. *Handbook of Pattern Recognition and Image Processing*. San Diego: Academic Press, 1986, Cap. 9, pp. 215-231.

⁵⁰ Pratt, W. K., op. cit. p. 16.

⁵¹ De Hoyos, A. J. e Corbera, R. P. Estudio Comparativo de Operadores de Borda Utilizando el LATIM. *Proceedings of the V Latin American Remote Sensing Symposium*. Cuzco: Selper, 1991, pp. 452-458.

Através da observação da figura (3.2) podemos ver que a aplicação desse algoritmo à nossa imagem produz uma outra com uma certa quantia de ruído. Por último, vamos falar do método de Roberts para detecção de borda.

Esse método trabalha com o cálculo da derivada da função intensidade em duas direções equivalentes à 45° e 135°, detectando, assim, bordas principalmente nessas duas direções. O cálculo dessas derivadas é feita pela aplicação das seguintes máscaras de convolução à imagem:

$$A_{45} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad e \quad A_{135} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

Em seguida, a partir dos valores obtidos, fazemos o cálculo do gradiente correspondente ao pixel central da aplicação das máscaras da seguinte forma:

$$G(x,y) = \sqrt{(A_{45})^2 + (A_{135})^2}. \quad (3.4.3)$$

Se o valor desse gradiente ultrapassar um limiar, que em nosso caso foi estabelecido como 8, esse pixel será considerado um pixel pertencente a uma borda e receberá valor igual a 1. Caso contrário, receberá valor igual a 0. Mais informações sobre esse método podem ser vistas em Kashyap (1986)⁵². O resultado da aplicação do método de Roberts à nossa imagem podem ser vistas na figura (3.3).

⁵² Kashyap, R. L. Image Models. Young, T. Y. e Fu, K. *Handbook of Pattern Recognition and Image Processing*. San Diego: 1986, Cap. 12, pp. 281-310.



Figura 3.3. Imagem após a Aplicação do Método de Roberts para Detecção de Bordas.

3.6 Estudos Sobre Texturas

Uma imagem é formada por diversos tipos de texturas. Por esse motivo, quando há uma degradação na qualidade desta, é de se esperar que esta ocorra também nas texturas. Um estudo desta degradação poderia nos revelar aspectos importantes dos métodos de compressão e talvez nos ajudar na procura de uma nova medida de distorção estatística.

Antes de falarmos do tipo de abordagem a ser utilizada para o estudo das texturas, apresentaremos os principais resultados de um trabalho que fizemos sobre a separação das texturas de uma imagem. Os resultados deste trabalho foram apresentados no VI CLAPEM (novembro de 1995).

3.6.1 Modelo Estatístico para a Análise de Texturas das Imagens

Ao observarmos uma imagem, em geral associamos regiões homogêneas a regiões nas quais o nível médio de cinza é o mesmo. No entanto, isto nem sempre é verdade, pois podemos ter regiões que sejam visualmente diferentes mas que contenham o mesmo nível médio de cinza. O que diferencia as regiões neste caso é a forma nas quais os níveis de cinza estão arranjados formando as diferentes texturas. Por esse motivo, Nevatia (1986)⁵³, em seu artigo, levantou as seguintes questões:

- Dada uma região textural, determinar a que número finito de classes essa região pertence.
- Dada uma região textural, determinar uma descrição ou modelo para ela.
- Dada uma imagem com muitas áreas texturais, determinar os limites entre as diferentes regiões texturais.

No momento, estamos preocupados com problemas de, dada uma imagem com diferentes texturas, através de amostras dessas regiões e algumas informações da imagem, encontrar uma forma de classificar a figura inteira nessas classes de textura.

Consideremos agora que somente tenhamos uma imagem e esta seja composta de texturas diferentes. Através do uso da técnica estatística análise discriminante queremos classificar a imagem original utilizando algumas transformações que possam conter informações importantes das texturas. Sendo assim, selecionamos algumas transformações para serem aplicadas na imagem. Todas as transformações foram feitas utilizando-se o software Khoros e estão descritas a seguir:

Filtros de Gradiente

Vamos considerar a textura de uma imagem como a quantidade de "bordas" por unidade de área da imagem. Dessa forma, uma "borda" pode ser detectada por vários operadores matemáticos

⁵³ Nevatia, R., op. cit. p. 97.

relacionados ao gradiente da função de intensidade da imagem.

Nas amostras retiradas das imagens aplicaremos duas técnicas de realce de bordas relacionadas com o gradiente da imagem. A primeira é aplicar os operadores de Sobel, que consiste em calcularmos $G(x,y) = \sqrt{G_x^2 + G_y^2}$, onde G_x corresponde ao valor do pixel central de uma máscara 3×3 após aplicarmos um filtro de Sobel N-S, e G_y corresponde ao valor do pixel central após aplicarmos um filtro de Sobel E-N. A descrição deste filtros e mais detalhes sobre este operador podem ser vistos em Pratt (1991)⁵⁴. A outra é aplicarmos os operadores de Roberts que já foram citados na seção 3.4.

Co-ocorrência de Tons de Cinza

A matriz de co-ocorrência ou dependência espacial dos tons de cinza é uma medida característica de segunda ordem para a variação da imagem. Portanto, feições texturais podem ser calculadas através dessa matriz. A co-ocorrência (P_{ij}) dos tons de cinza i e j de uma imagem é definida como o número de pares de pixels tendo os níveis de cinza i e j respectivamente. Algumas estatísticas utilizadas a partir das probabilidades de co-ocorrência são a energia, entropia, contraste, correlação e homogeneidade.

Neste trabalho, utilizaremos a medida de entropia calculada a partir de janelas 5×5 e 15×15. O processo de cálculo dessas entropias será o de, a cada janela 5×5 ou 15×15 na imagem, substituir o valor do pixel central pela entropia calculada.

Filtro da Mediana

Ao aplicar-se um filtro da mediana em uma imagem, obtém-se uma imagem suavizada na qual eliminam-se pontos de intensidade distinta isolados. Uma das vantagens desse filtro é que ele preserva os contornos da imagem. O filtro consiste em substituímos o valor do pixel central de uma

⁵⁴ Pratt, W. K., op. cit. p. 16.

máscara 3×3 pela mediana dos seus valores.

Filtro Laplaciano

Laplaciano é um operador derivativo da forma:

$$\nabla^2 f = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2}$$

e seu principal efeito numa figura é aumentar o contraste das bordas, ou seja, aumentar a nitidez. Assim, neste trabalho, aplicamos o seguinte filtro laplaciano, correspondente à aplicação de uma máscara de convolução 3×3 do tipo:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Filtro de Alta Ênfase

Este tipo de filtro consiste em fortalecermos as freqüências altas da imagem. Isto foi feito aplicando-se a seguinte máscara de convolução:

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

Esta máscara corresponde ao filtro laplaciano+original.

Varição dos Tons de Cinza

Como cada textura é caracterizada pela forma com que os pixels estão arranjados, uma medida proposta é o cálculo da variância dos valores ao redor dos pixels. Dessa forma, para cada janela 5×5 e 15×15 substituiremos o valor do pixel central pela variância da janela.

Em seguida, a partir da imagem da Lena retiramos amostras de texturas diferentes através do programa Xview, como vemos na figura (3.4).

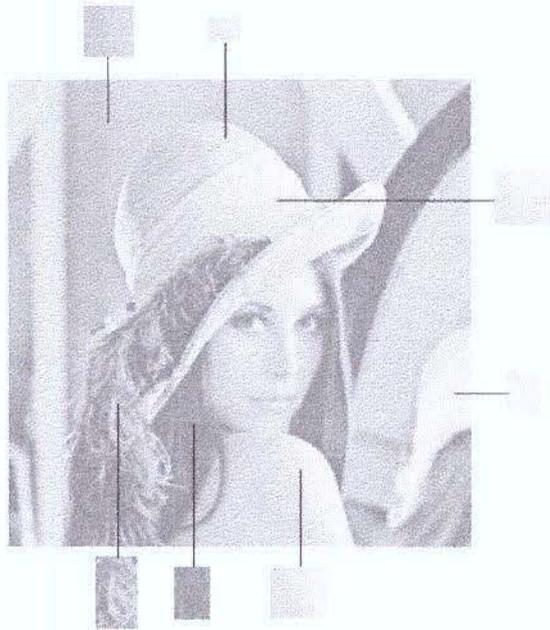


Figura 3.4. Amostras de Texturas da Imagem da Lena.

Em seguida, a cada amostra aplicamos as transformações descritas anteriormente obtendo 11 variáveis. Com todas essas variáveis, juntamente com uma variável textura - que indica no conjunto de dados a que amostra os dados pertenciam - aplicamos uma análise discriminante *stepwise* utilizando-se o software estatístico SAS. Isto foi feito para verificarmos a importância da contribuição de cada uma das variáveis no análise, e, possivelmente, retirar alguma que se mostrasse desnecessária. A partir do resultado observado, eliminamos a variável correspondente ao filtro laplaciano e realizamos uma análise discriminante canônica, obtendo a função linear discriminante a partir da qual classificamos a imagem inteira por textura. O resultado da classificação da imagem original através dessa função pode ser vista na figura (3.5).



Figura 3.5. Imagem Classificada após a Primeira Análise.

A partir desse resultado vimos que, se aplicássemos um filtro de alta ênfase à imagem original, talvez obtivéssemos um resultado melhor, uma vez que este realça as bordas da imagem e, conseqüentemente, os detalhes das texturas.

Assim, aplicamos o filtro de alta ênfase à imagem original (Figura 3.6).



Figura 3.6. Imagem da Lena após a Aplicação de um Filtro de Alta Ênfase.

Repetindo a análise feita anteriormente, utilizamos o procedimento de análise *stepwise* discriminante com todas as transformações (exceto a correspondente ao filtro de alta ênfase), sendo que desta vez as amostras foram retiradas desta última imagem.

Em seguida aplicamos a análise discriminante canônica, a partir da qual obtivemos uma função linear discriminante que, sobre os dados originais, apresentou 18,85% de erros de má classificação. O resultados da classificação da imagem original utilizando esta última função linear discriminante pode ser vista na figura (3.7).

Vemos que apesar do ruído que existe agora nas imagens, a figura está melhor classificada. Aplicando duas vezes um filtro para remoção de ruído, obtemos a figura (3.8). Podemos observar que agora existem poucos pixels mal classificados na imagem resultante .



Figura 3.7. Imagem Classificada após a Segunda Análise.



Figura 3.8. Imagem Classificada após a Remoção de Ruído.

3.6.2 Método de Estudo das Texturas Neste Trabalho

Após observarmos o estudo anterior, vemos que a aplicação de um filtro de alta ênfase na imagem realça a qualidade das texturas. Por esse motivo, retiramos as mesmas amostras apresentadas na figura (3.4) após a aplicação do filtro, referenciando-as por:

1- Faixa do chapéu

4- Cabelo

6- Detalhe branco

2- Pele do ombro

5- Fundo

7- Chapéu

3- Detalhe do chapéu

Utilizando essas amostras de cada imagem reconstruída a partir dos métodos de compressão, fizemos o estudo dos efeitos dos métodos sobre as texturas como será mostrado no capítulo seguinte.

Capítulo 4

Análise dos Resultados

4.1 Introdução

No capítulo 2, vimos quais métodos de compressão seriam utilizados na análise e no capítulo 3 apresentamos as medidas de distâncias entre a imagem original e as comprimidas a serem estudadas. Neste capítulo mostraremos como, a partir das imagens resultantes dos métodos de compressão e das medidas de distâncias, conseguimos obter as medidas de distorção.

A principal parte da metodologia utilizada bem como os principais resultados serão apresentados na seqüência em que foram realizados. A aplicação das medidas de distância às imagens e obtenção destas em forma matricial foram realizadas por alguns programas feitos em linguagem C para UNIX. Além disso, a parte estatística foi feita principalmente utilizando-se os softwares estatísticos SAS e S-PLUS.

4.2 Abordagem por Blocos e Geral

Para facilitarmos o trabalho com as medidas de distância nas análises foi necessário

atribuirmos nomes diferentes dos originais. Assim, a recodificação deu-se como está descrito na tabela (4.2.1). Durante a exposição, tentaremos, na medida do possível, nos referenciar às variáveis através de seu nome original para facilitar a leitura e compreensão dos resultados.

Tabela 4.2.1. Notação utilizada para referenciar as medidas de distância.

Medida Utilizada	Notação	Medida Utilizada	Notação
Distância 1	V1	t de Student	V11
Chi-quadrado	V2	Afinidade	V12
Minkowski	V3	Atração	V13
Divergência	V4	Dist1 (Mathai)	V14
Canberra Metric	V5	Dist2 (Mathai)	V15
Bray e Curtis	V6	Dist. de Matusita	V16
Soergel	V7	Dist3 (Matusita)	V17
Ware e Hedges	V8	Discrepância	V18
Correlação	V9	Imprecisão	V19
F de Snedecor	V10	Divergência	V20

Nossa principal idéia relacionada à procura de medidas de distorção estatísticas era trabalharmos com a imagem subdividida em blocos. Isto porque, se observarmos as imagens resultantes dos diferentes métodos de compressão, veremos que, dependendo do método, certas regiões da imagem apresentam maior degradação que outras. Uma medida local que trabalhasse com a imagem particionada em blocos seria, assim, mais sensível a determinadas alterações provocadas por diferentes métodos que uma medida global.

Passamos então à seguinte questão: Qual seria o tamanho do bloco ideal ? Qual o critério que usaríamos para defini-lo ? Tendo em vista o objetivo do trabalho, o tamanho de bloco ideal seria o maior dentre aqueles que fossem sensíveis a todas as mudanças que ocorressem à medida que as imagens fossem reconstruídas com qualidades distintas.

Optamos então por ver quantas medidas de distância discriminavam as imagens produzidas por um mesmo método à medida que variávamos o tamanho do bloco. Isto porque, quanto maior

fosse o número de variáveis que discriminassem todas as imagens, mais adequado seria esse tamanho de bloco para o trabalho, já que desejávamos uma medida de distorção baseada em nossas variáveis que fosse boa para classificar todos os métodos.

Para tal, fizemos uma análise *stepwise* discriminante utilizando o *proc stepdisc*, observando-se quais variáveis ao final eram significantes no modelo. Os resultados referentes aos blocos de tamanho $2^4 \times 2^4$ e $2^5 \times 2^5$ são apresentados nas tabelas (4.2.2) e (4.2.3).

Tabela 4.2.2. Valores de R^2 parciais das variáveis que entraram no modelo a um nível de significância de 15%, para blocos de tamanho $2^4 \times 2^4$.

Variável	Método de Compressão Utilizado						
	DCT	FFT	WAV	JPG	FTQ	FRC	FRAC
V1	0,0028	0,0140	0,0079	0,0094	0,0434	0,8658	-
V2	0,0475	0,0140	0,0621	-	0,0032	0,0060	-
V3	0,0354	0,0052	0,0131	0,0107	0,0097	0,9764	-
V4	-	0,0353	0,0058	0,0031	0,0069	0,0554	-
V5	0,0085	-	0,0056	-	0,0094	0,1653	-
V6	0,0194	0,0107	0,0069	-	-	0,1009	0,0221
V7	-	0,0039	0,0029	-	-	0,1152	-
V8	0,4456	0,0047	0,3470	0,0676	0,0093	0,0090	0,0750
V9	0,0014	0,3181	0,0899	0,0581	0,0272	0,0145	0,0404
V10	0,0488	0,1156	-	0,0047	0,1977	0,0216	-
V11	0,0355	0,0285	0,0103	0,0077	0,1788	0,1138	0,0090
V12	-	0,0268	-	-	-	-	-
V13	0,0108	-	0,0082	0,0015	-	0,0522	-
V14	0,0035	0,0167	0,0075	-	-	0,2969	-
V15	-	0,0014	0,0019	0,0075	0,0054	0,0115	-
V16	-	-	-	-	-	0,0013	-
V17	0,0046	-	0,0006	0,0215	0,0033	0,2980	-
V18	0,0093	0,0033	0,0210	-	0,0036	0,0109	-
V19	0,0296	0,0281	0,0042	0,0092	0,0488	0,1347	-
V20	0,0031	-	0,0073	0,0087	0,0087	0,0151	-

Tabela 4.2.3. Valores de R^2 parciais das variáveis que entraram no modelo a um nível de significância de 15%, para blocos de tamanho $2^5 \times 2^5$.

Variável	Método de Compressão Utilizado						
	DCT	FFT	WAV	JPG	FTQ	FRC	FRAC
V1	0,0204	0,0177	-	-	0,0640	0,9427	-
V2	0,0134	0,0075	0,0284	-	0,0170	0,1593	-
V3	0,0438	-	0,0550	0,0090	-	0,9893	-
V4	0,0075	0,0095	-	0,0079	0,0242	0,0811	-
V5	0,0510	0,0154	-	-	-	0,0436	-
V6	0,0102	0,0106	0,0349	-	-	0,0263	-
V7	-	0,0342	0,0881	-	0,0104	0,0960	-
V8	0,5731	0,4220	0,4390	0,0882	0,1597	0,0383	0,1024
V9	0,0266	0,0292	0,0585	0,0337	0,0108	0,0106	-
V10	0,0941	0,0919	0,0052	-	0,1639	0,0762	-
V11	0,0245	0,0225	-	0,0200	0,0116	0,1657	0,0603
V12	0,0159	-	-	-	-	0,0722	-
V13	0,0187	-	-	-	-	0,1015	-
V14	0,0409	-	-	-	-	0,0471	-
V15	-	-	-	0,0123	-	0,1173	-
V16	-	-	0,0226	0,0288	0,0377	0,0657	-
V17	-	-	0,0053	0,0193	-	-	0,0994
V18	0,0144	0,0136	0,0185	-	0,0081	0,3893	-
V19	0,1820	0,1596	-	0,0276	0,0532	0,1776	-
V20	0,0110	-	-	-	0,0113	-	-

Através da observação da tabela (4.2.2) podemos ver que, para blocos de tamanho $2^4 \times 2^4$, as variáveis V8, V9 e V11 foram significativas em todos os métodos, e outras 3 variáveis, V1, V3 e V19 só não foram significativas no método fractal referenciado como FRAC. Observando-se os resultados para blocos $2^5 \times 2^5$ na tabela (4.2.3), vemos que, neste caso, somente uma variável, V8, foi significativa em todos os métodos e a variável V9 só não classificou as imagens do método fractal. Para os blocos de tamanho $2^3 \times 2^3$, os resultados foram apenas um pouco melhores do que aqueles obtidos com o tamanho de bloco anterior. Como desejávamos o maior tamanho de bloco que possuísse mais variáveis que discriminassem todos os métodos (ou grande parte deles), vimos que o tamanho do bloco ideal para trabalharmos com as medidas de

distância era de $2^4 \times 2^4$. Este será o tamanho em todas as análises feitas por bloco daqui em diante.

Uma vez definido o tamanho do bloco, antes de analisarmos as medidas de distância dos mesmos, desejamos ver como as imagens reconstruídas estão em relação à imagem original.

A melhor maneira de vermos isto é fazermos gráficos semelhantes aos de controle de qualidade, da média da diferença entre as imagens por bloco. Os gráficos correspondentes são os numerados de (4.1) a (4.23), mostrados a seguir:

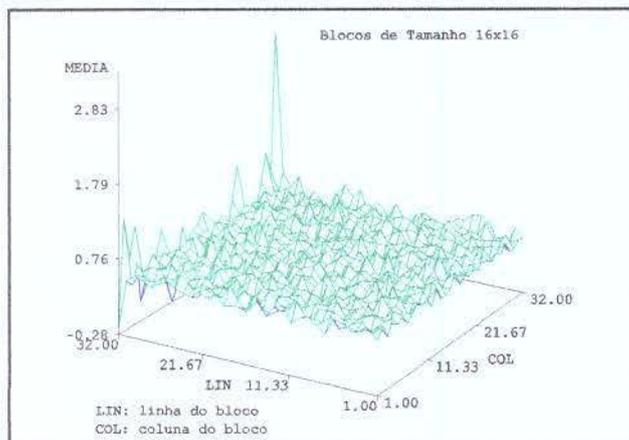


Gráfico 4.1. Diferenças entre a Imagem Original e a Reconstruída - DCT80.

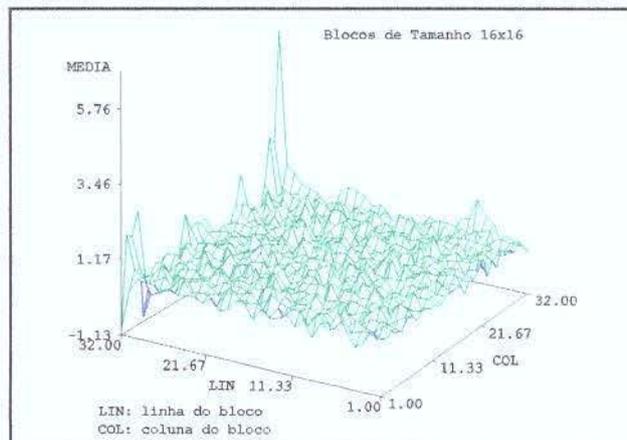


Gráfico 4.2. Diferenças entre a Imagem Original e a Reconstruída - DCT90.

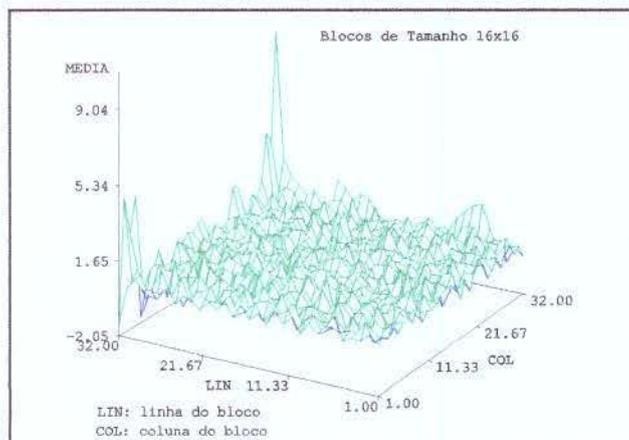


Gráfico 4.3. Diferenças entre a Imagem Original e a Reconstruída - DCT95.

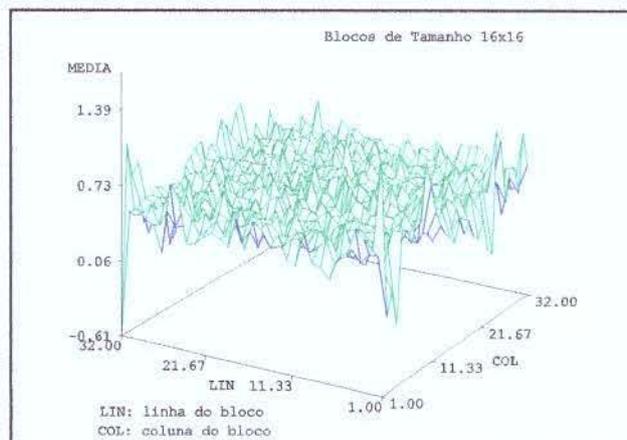


Gráfico 4.4. Diferenças entre a Imagem Original e a Reconstruída - FFT80.

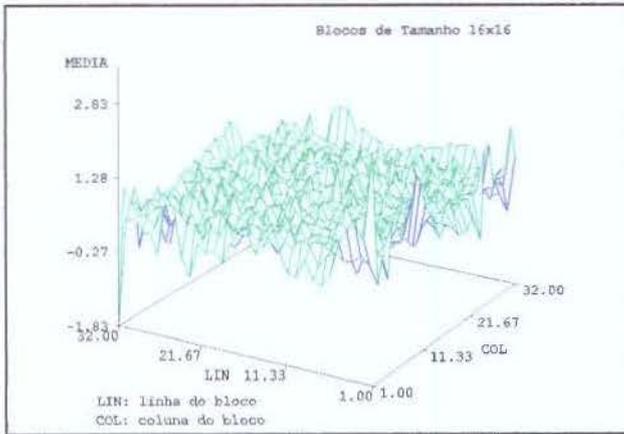


Gráfico 4.5. Diferenças entre a Imagem Original e a Reconstruída - FFT90.

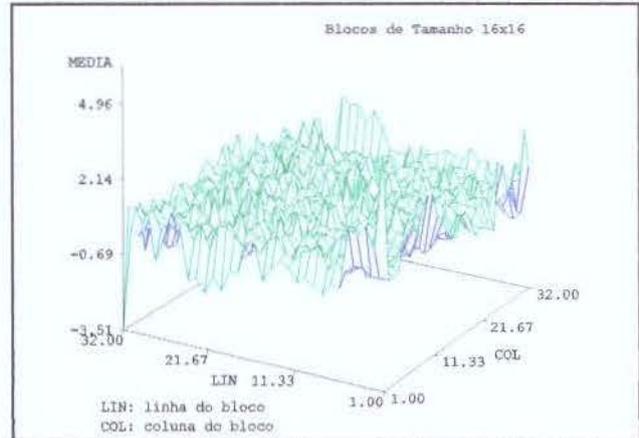


Gráfico 4.6. Diferenças entre a Imagem Original e a Reconstruída - FFT95.

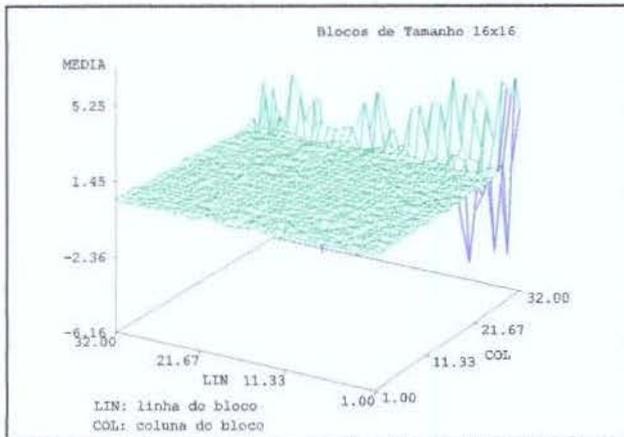


Gráfico 4.7. Diferenças entre a Imagem Original e a Reconstruída - WAV80.

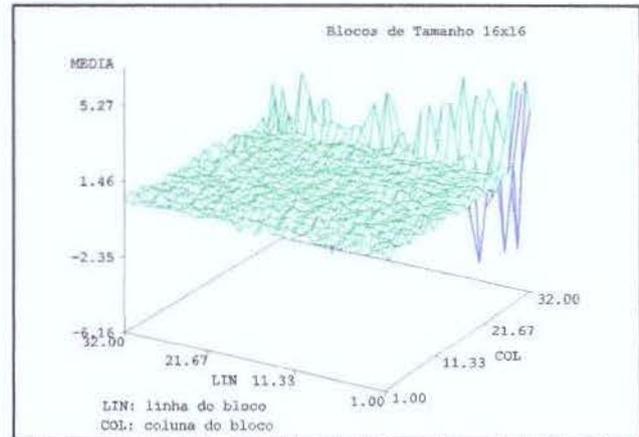


Gráfico 4.8. Diferenças entre a Imagem Original e a Reconstruída - WAV90.

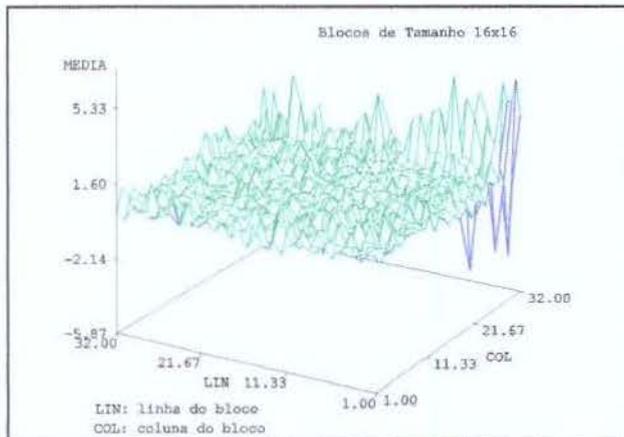


Gráfico 4.9. Diferenças entre a Imagem Original e a Reconstruída - WAV95.

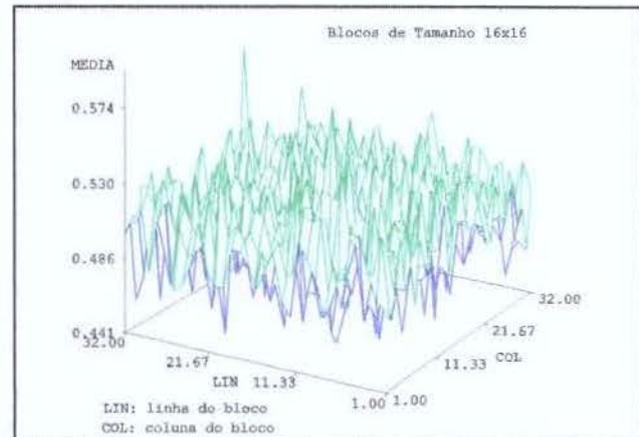


Gráfico 4.10. Diferenças entre a Imagem Original e a Reconstruída - FTQ50.

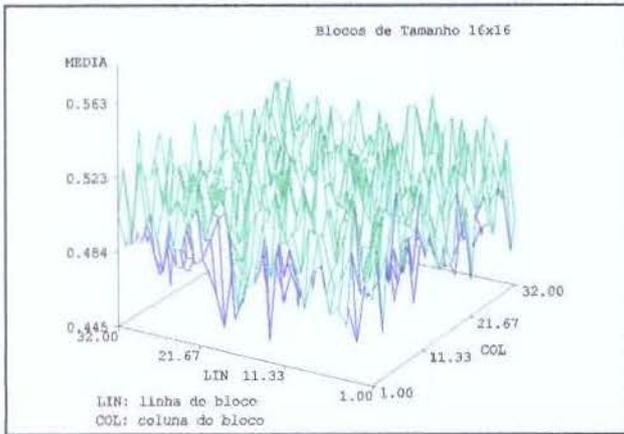


Gráfico 4.11. Diferenças entre a Imagem Original e a Reconstruída - FTQ60.

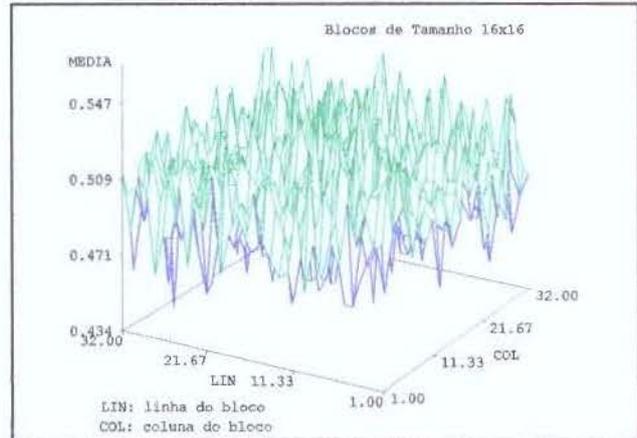


Gráfico 4.12. Diferenças entre a Imagem Original e a Reconstruída - FTQ70.

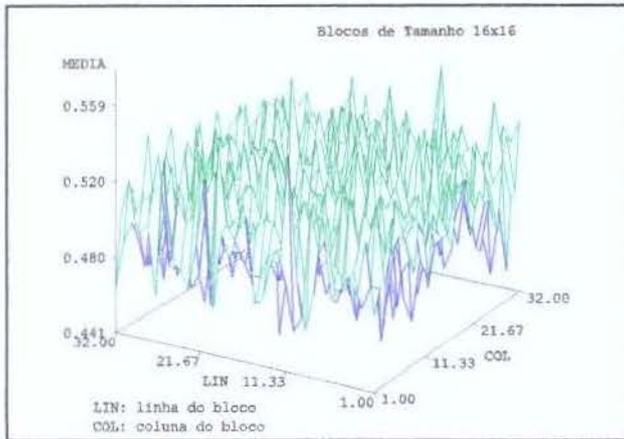


Gráfico 4.13. Diferenças entre a Imagem Original e a Reconstruída - FTQ90.

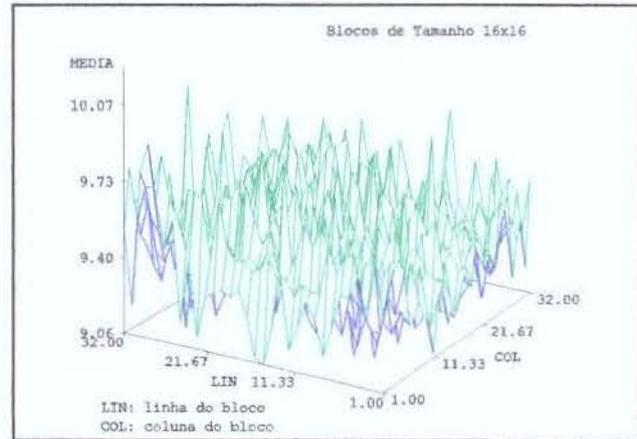


Gráfico 4.14. Diferenças entre a Imagem Original e a Reconstruída - FRC8.

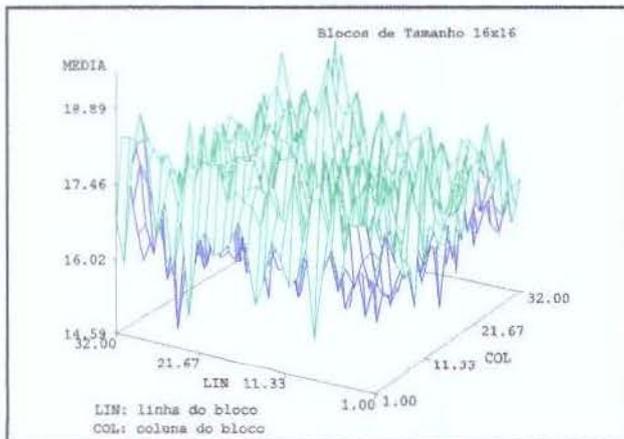


Gráfico 4.15. Diferenças entre a Imagem Original e a Reconstruída - FRC6.

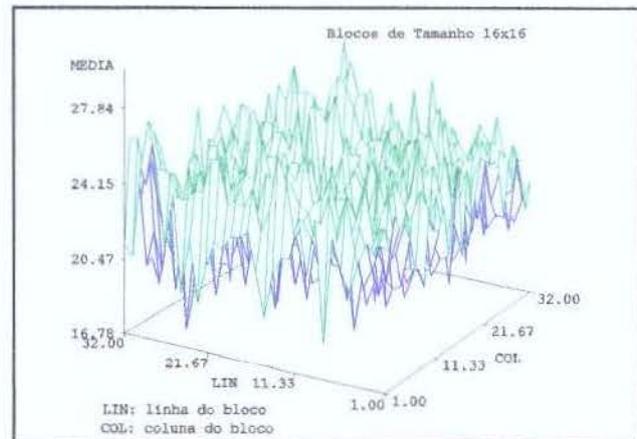


Gráfico 4.16. Diferenças entre a Imagem Original e a Reconstruída - FRC5.

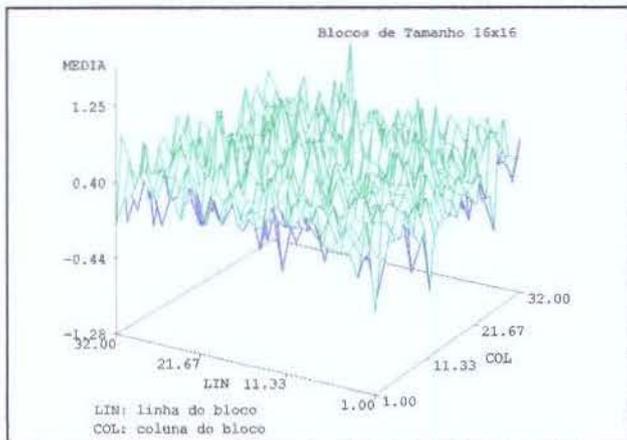


Gráfico 4.17. Diferenças entre a Imagem Original e a Reconstruída - FRAC2.

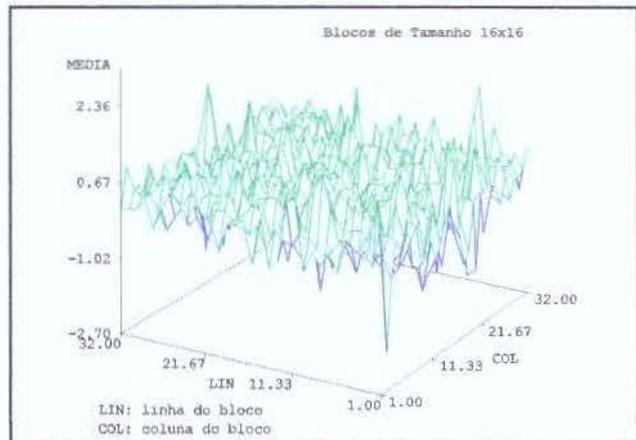


Gráfico 4.18. Diferenças entre a Imagem Original e a Reconstruída - FRAC3.

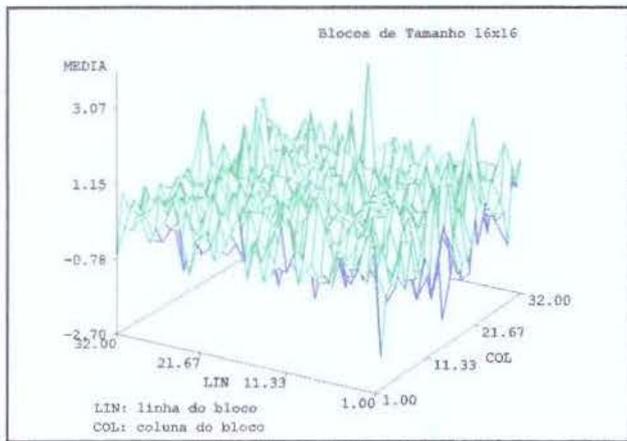


Gráfico 4.19. Diferenças entre a Imagem Original e a Reconstruída - FRAC4.

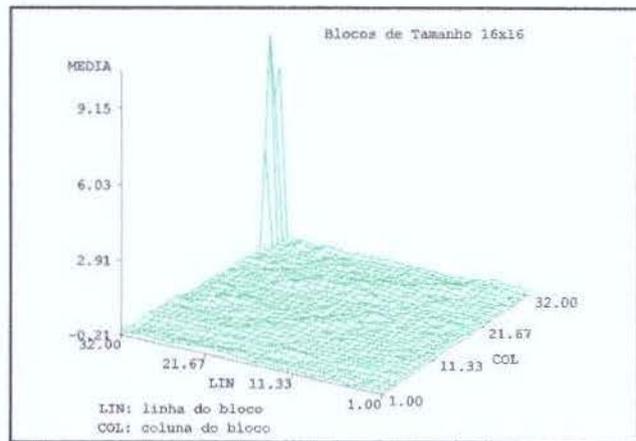


Gráfico 4.20. Diferenças entre a Imagem Original e a Reconstruída - JPG95.

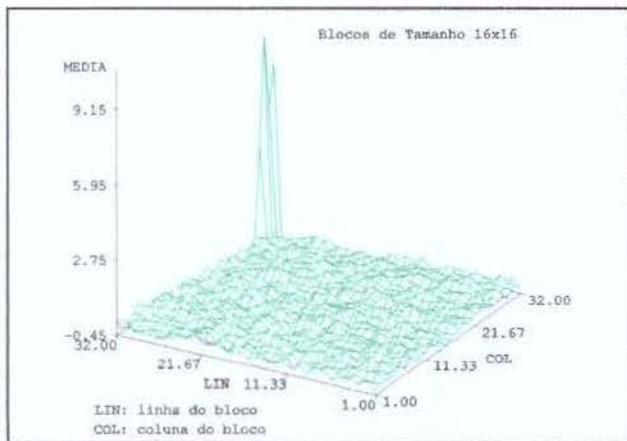


Gráfico 4.21. Diferenças entre a Imagem Original e a Reconstruída - JPG75.

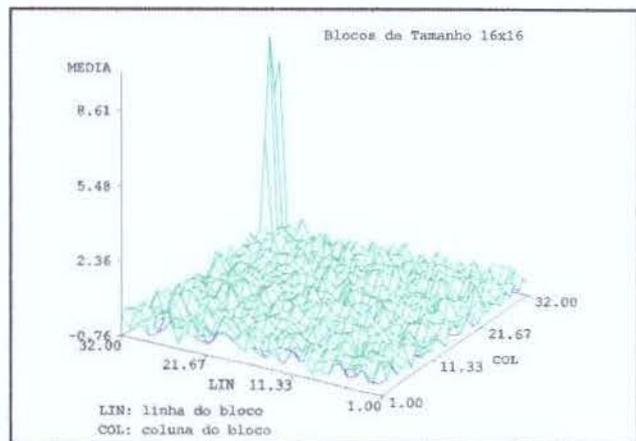


Gráfico 4.22. Diferenças entre a Imagem Original e a Reconstruída - JPG50.

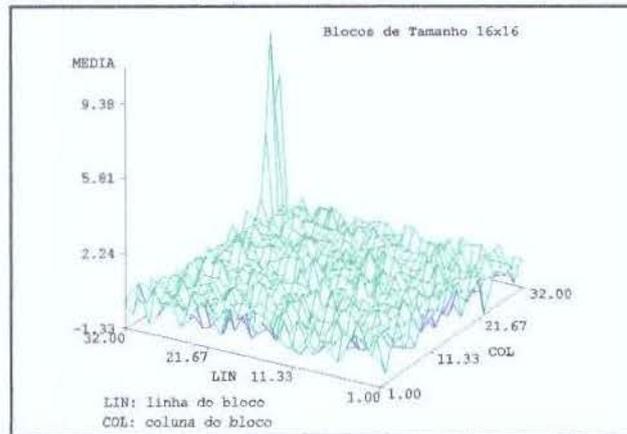


Gráfico 4.23. Diferenças entre a Imagem Original e a Reconstruída - JPG35.

Através da observação dos gráficos (4.1) a (4.23) podemos ver que a imagem cuja média dos pixels mais se aproxima da imagem original é a JPG95 (gráfico 4.20), já que as médias se distribuem mais uniformemente próximas a zero. No entanto, podemos ver que as imagens produzidas pelo método JPEG apresentam uma pequena falha na reconstrução: os pixels correspondentes aos blocos da parte inferior direita da imagem se afastam muito das intensidades originais. O mesmo acontece com as imagens referenciadas por DCT (gráficos 4.1 a 4.3), mostrando-nos que esse problema de distorção nas fronteiras está provavelmente relacionado à utilização do próprio algoritmo da transformada cosseno discreta.

Outra imagem cuja média se aproxima bem da imagem original é a WAV80 (gráfico 4.7). Podemos observar que neste método também há uma falha na reconstrução onde alguns blocos (os da direita) apresentam a média dos valores de intensidade levemente distintos do original. De uma forma geral, esses gráficos revelam detalhes interessantes do comportamento dos métodos de compressão, detalhes estes que, em geral, não são perceptíveis a olho. Além disso, pode-se ver que, dentro de um mesmo método de compressão, à medida que a qualidade das imagens vai piorando, os gráficos vão perdendo a homogeneidade, apresentando mais variação entre as médias do blocos.

Se observarmos os gráficos das imagens obtidas pelo método de fractal denotadas por FRC

(gráficos 4.14 a 4.16), veremos que a média das diferenças por bloco é muito grande, assim como sua variância. Mesmo para a melhor imagem deste método, FRC8, esta média é muito alta. Este tipo de comportamento nos indica que, para muitos blocos destas imagens, as medidas de distância serão muito distoantes das medidas calculadas sobre as imagens resultantes de outros métodos.

Uma vez tendo observado como a média da intensidade dos pixels das imagens reconstruídas se relacionavam com a mesma na imagem original, antes de iniciar a procura de um índice de distorção, queríamos ver como as nossas medidas de distância estavam relacionadas entre si, de forma a selecionar variáveis mais representativas e evitar redundâncias. Por esse motivo calculamos as distâncias entre a imagem original e as reconstruídas e ao final observamos a correlação entre essas variáveis independente da imagem. O resultado encontra-se na tabela (4.2.4).

Após observarmos a tabela (4.2.4), podemos verificar a existência de agrupamentos entre certas variáveis, com correlação razoavelmente alta entre si. Por esse motivo, a ordem de disposição das variáveis na tabela segue os agrupamentos observados. Vemos que as variáveis V5, V6 e V7 correspondentes às medidas Soergel, Camberra Metric e Bray e Curtis são altamente correlacionadas. Analisando suas fórmulas na tabela (3.3.1.1) do capítulo 3, veremos que de fato essas medidas são bem parecidas entre si. Uma vez que essa correlação foi bem alta, deveríamos eliminar pelo menos duas dessas variáveis. No entanto, voltando a observar as tabelas (4.2.2) e (4.2.3), veremos que essas variáveis possuem um comportamento distinto entre si quando trabalhamos com elas nos preocupando com seus valores relacionados ao método de compressão utilizado. Por esse motivo, podemos concluir que não podemos eliminar nenhuma dessas variáveis já que estamos interessados neste último caso.

Na verdade, a observação da tabela (4.2.4) nos dá apenas uma idéia dos resultados que devemos esperar, não nos permitindo eliminar variáveis. Voltando à observação dos agrupamentos, vemos que estes ocorrem em geral com medidas do mesmo tipo. Um exemplo é o grupo formado pelas variáveis V15, V16 e V17, que correspondem a medidas de informação.

Tabela 4.2.4. Coeficiente de correlação de Pearson entre as variáveis analisadas.

	V1	V2	V3	V5	V6	V7	V4	V12	V13	V18	V14	V15	V16	V17	V8	V9	V10	V11	V19	V20
V1	1,0000	0,9603	0,9898	0,8967	0,8961	0,8959	0,7904	-0,5579	-0,6561	0,6979	0,7640	0,6701	0,7527	0,7433	0,5045	-0,3293	-0,0402	0,4485	-0,0028	0,6394
V2	0,9603	1,0000	0,9523	0,9598	0,9544	0,9486	0,9133	-0,6607	-0,7804	0,8359	0,8543	0,7508	0,8419	0,8316	0,4986	-0,2848	-0,0380	0,4173	-0,0390	0,7578
V3	0,9898	0,9523	1,0000	0,9233	0,9256	0,9278	0,7639	-0,5005	-0,6112	0,6690	0,7747	0,6614	0,7524	0,7339	0,5732	-0,3180	-0,0429	0,5160	-0,0127	0,6011
V5	0,8967	0,9598	0,9233	1,0000	0,9986	0,9974	0,8726	-0,5902	-0,7259	0,8004	0,8904	0,7643	0,8656	0,8428	0,6289	-0,2271	-0,0421	0,5043	-0,1104	0,7128
V6	0,8961	0,9544	0,9256	0,9986	1,0000	0,9994	0,8550	-0,5669	-0,7063	0,7856	0,8882	0,7582	0,8587	0,8325	0,6358	-0,2369	-0,0422	0,5133	-0,0883	0,6979
V7	0,8959	0,9486	0,9278	0,9974	0,9994	1,0000	0,8421	-0,5598	-0,6969	0,7745	0,8877	0,7596	0,8591	0,8331	0,6512	-0,2353	-0,0429	0,5221	-0,0977	0,6882
V4	0,7904	0,9133	0,7639	0,8726	0,8550	0,8421	1,0000	-0,8429	-0,9187	0,9277	0,8086	0,7459	0,8245	0,8478	0,3837	-0,2100	-0,0292	0,3068	-0,0964	0,8510
V12	-0,5579	-0,6607	-0,5005	-0,5902	-0,5669	-0,5598	-0,8429	1,0000	0,9601	-0,8622	-0,6580	-0,7078	-0,7343	-0,8156	-0,2702	0,1423	0,0209	-0,0990	0,2331	-0,8232
V13	-0,6561	-0,7804	-0,6112	-0,7259	-0,7063	-0,6969	-0,9187	0,9601	1,0000	-0,9677	-0,8076	-0,8073	-0,8573	-0,9010	-0,3416	0,1609	0,0250	-0,1278	0,2017	-0,9157
V18	0,6979	0,8359	0,6690	0,8004	0,7856	0,7745	0,9277	-0,8622	-0,9677	1,0000	0,8882	0,8432	0,9070	0,9170	0,3819	-0,1667	-0,0269	0,1445	-0,1662	0,9212
V14	0,7640	0,8543	0,7747	0,8904	0,8882	0,8877	0,8086	-0,6580	-0,8076	0,8882	1,0000	0,8917	0,9844	0,9579	0,6101	-0,1752	-0,0393	0,1709	-0,2301	0,7989
V15	0,6701	0,7508	0,6614	0,7643	0,7582	0,7596	0,7459	-0,7078	-0,8073	0,8432	0,8917	1,0000	0,9352	0,9325	0,5317	-0,1263	-0,0368	0,1094	-0,3037	0,7796
V16	0,7527	0,8419	0,7524	0,8656	0,8587	0,8591	0,8245	-0,7343	-0,8573	0,9070	0,9844	0,9352	1,0000	0,9877	0,5973	-0,1556	-0,0396	0,1511	-0,2854	0,8436
V17	0,7433	0,8316	0,7339	0,8428	0,8325	0,8331	0,8478	-0,8156	-0,9010	0,9170	0,9579	0,9325	0,9877	1,0000	0,5767	-0,1568	-0,0388	0,1450	-0,3143	0,8402
V8	0,5045	0,4986	0,5732	0,6289	0,6358	0,6512	0,3837	-0,2702	-0,3416	0,3819	0,6101	0,5317	0,5973	0,5767	1,0000	-0,1982	-0,0234	0,4278	-0,1970	0,3383
V9	-0,3293	-0,2848	-0,3180	-0,2271	-0,2369	-0,2353	-0,2100	0,1423	0,1609	-0,1667	-0,1752	-0,1263	-0,1556	-0,1568	-0,1982	1,0000	-0,1402	-0,1029	-0,3730	-0,1548
V10	-0,0402	-0,0380	-0,0429	-0,0421	-0,0422	-0,0429	-0,0292	0,0209	0,0250	-0,0269	-0,0393	-0,0368	-0,0396	-0,0388	-0,0234	-0,1402	1,0000	-0,0380	0,0269	-0,0242
V11	0,4485	0,4173	0,5160	0,5043	0,5133	0,5221	0,3068	-0,0990	-0,1278	0,1445	0,1709	0,1094	0,1511	0,1450	0,4278	-0,1029	-0,0380	1,0000	0,1038	0,1278
V19	-0,0028	-0,0390	-0,0127	-0,1104	-0,0883	-0,0977	-0,0964	0,2331	0,2017	-0,1662	-0,2301	-0,3037	-0,2854	-0,3143	-0,1970	-0,3730	0,0269	0,1038	1,0000	-0,1093
V20	0,6394	0,7578	0,6011	0,7128	0,6979	0,6882	0,8510	-0,8232	-0,9157	0,9212	0,7989	0,7796	0,8436	0,8402	0,3383	-0,1548	-0,0242	0,1278	-0,1093	1,0000

A seguir, calculamos as medidas de distância das imagens e iniciamos a procura de uma medida de distorção que poderia ser formada de uma ou mais medidas que classificassem bem as imagens em termos de qualidade (ou equivalentemente em termos de qualidade e compressão). Apesar de muitos esforços, não encontrávamos bons resultados. Em vista disso, resolvemos abandonar um pouco a abordagem por blocos e calcular a distância da Mahalanobis das imagens. Agora cada bloco seria considerado uma população. O tamanho do bloco escolhido anteriormente era o que levava as distâncias aplicadas a ele serem mais sensíveis a variações de qualidade na imagem.

Como o novo critério era o tamanho de bloco que minimizasse o T^2 para as melhores imagens, seria possível que o tamanho $2^4 \times 2^4$ não fosse mais o ideal. Calculando T^2 com este tamanho de bloco e para blocos de tamanho $2^5 \times 2^5$ comparamos os resultados e observamos que, para este último os resultados eram melhores. Para blocos de tamanho $2^6 \times 2^6$ não fizemos os cálculos, uma vez que este tamanho de bloco era muito grande, e certamente os resultados não seriam precisos, pois misturaria diversas texturas distintas. Concluimos, assim, que o bloco de tamanho $2^5 \times 2^5$ seria o ideal. Calculando o valor de T^2 , observamos os seguintes resultados:

Tabela 4.2.5. Valores de T^2 para a imagem original em blocos de tamanho $2^5 \times 2^5$ e ordenados segundo o valor calculado.

Imagem	T^2	Imagem	T^2
JPG95	29,12	JPG50	260,04
FTQ50	52,39	WAV80	424,31
FTQ90	54,16	FRAC2	477,78
FTQ60	55,27	WAV90	546,95
FTQ70	55,65	WAV95	755,54
FFT80	58,99	JPG35	780,50
DCT80	60,75	FRAC3	1764,68
FFT90	69,79	FRAC4	1923,36
JPG75	92,93	FRC5	2343,99
DCT90	107,34	FRC6	9798,95
DCT95	174,42	FRC8	17375,17
FFT95	214,52	-	-

Vimos que mesmo neste tipo de abordagem os resultados não foram satisfatórios em termos de classificação por qualidade. Visualmente sabíamos que uma imagem como a WAV80 aparentava ser tão boa quanto a JPG95 ou JPG75. Como seu T^2 correspondente poderia ser maior que uma imagem como FFT95 ?

Diante desses resultados, era o momento de pararmos e reavaliarmos todas as análises, pois algo estava faltando. Observando novamente tanto os gráficos (4.1) a (4.23) quanto as imagens originais (figuras 2.1 a 2.7, 2.9 a 2.1, 2.21 a 2.23 e 2.29 a 2.34), começamos a reparar que alguns blocos eram muito mais afetados pelos métodos de compressão do que outros. Além disso, em alguns métodos, como por exemplo o JPEG, as intensidades do nível de cinza das imagens reconstruídas, mesmo das melhores, não eram exatamente as mesmas do original.

Para solucionarmos esse problema, resolvemos filtrar as imagens com um filtro laplaciano, uma vez que este tipo de filtro, para cada conjunto de 9 pixels realiza a seguinte operação:

$$\begin{array}{ccc} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{array} \rightarrow a_5 = 4a_5 - a_2 - a_6 - a_4 - a_8$$

A imagem resultante seria formada pela variação dos pixels da imagem, correspondendo a perfis de variação dos tons de cinza, independentemente dos tons alcançados ao final da descompressão. Problemas como imagens de boa qualidade visual com uma leve variação no tom de cinza em relação à imagem original seriam solucionados.

Desta vez, começamos calculando o T^2 de Mahalanobis para algumas imagens, cujo resultado foi:

Tabela 4.2.6. Valores de T^2 para a imagem após o filtro laplaciano em blocos de tamanho $2^5 \times 2^5$ e ordenados segundo o valor calculado.

Imagem	T^2	Imagem	T^2
DCT80	0	JPG95	20,68
FTQ50	0,26	WAV90	21,94
FRC8	1,46	JPG75	25,54
FRC6	2,35	WAV95	25,92
FRC5	4,07	DCT90	28,20
FTQ90	7,51	JPG35	30,67
WAV80	18,51	DCT95	55,02

Vimos que alguns problemas que já haviam surgido desde o início persistiram, como, por exemplo, a imagem FTQ90 ser melhor do que quase todas as anteriores. Além disso, as imagens FRC5 e FRC6 apareciam sendo melhores que as restantes. Observamos que as imagens que apresentaram problemas neste método foram justamente as que tinham degradação só de texturas (FRC's) ou só de bordas (FTQ's), indicando que, se fôssemos utilizar esse T^2 como medida de distorção haveria muitos erros dependendo do método. Outro problema surgido é que a imagem WAV95 estava classificada igual em qualidade à JPG75, o que não correspondia ao observado.

Partimos, então, à procura de variáveis que servissem para classificar as imagens com resultados mais satisfatórios. Como já havíamos visto, tínhamos o problema de termos imagens nas quais muitos blocos eram perfeitamente reconstruídos e, certamente, tal fato estava prejudicando as análises. Por que, então, não tomar somente os blocos mais afetados pelos métodos ? Certamente estariam localizados acima do quartil superior ou abaixo do inferior. Para a seleção destes, deveríamos observar como as variáveis se comportavam com imagens de qualidades distintas. Resolvemos fazer os box-plots por método e por variável, mostrados nos gráficos (4.24) a (4.43). Por serem muitos, mostraremos somente os referentes ao método FFT80.

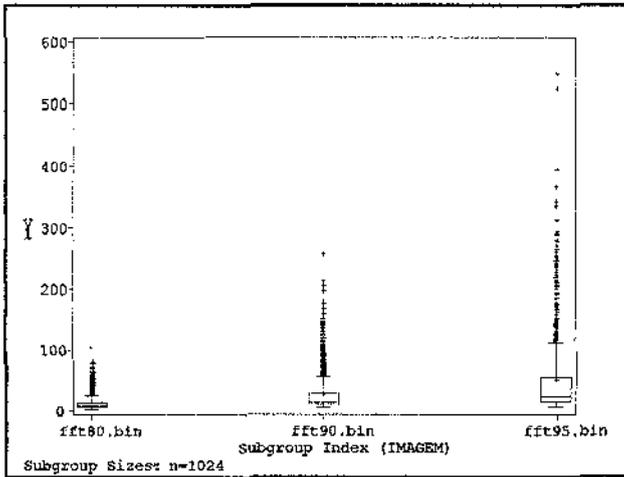


Gráfico 4.24. Box-Plot da Distância 1.

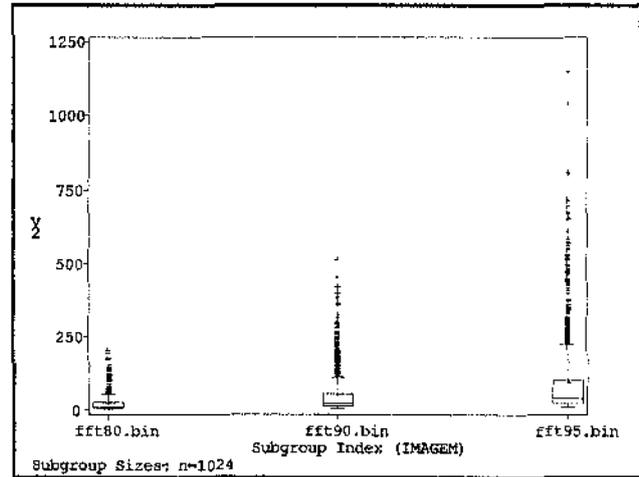


Gráfico 4.25. Box-Plot do Chi-quadrado.

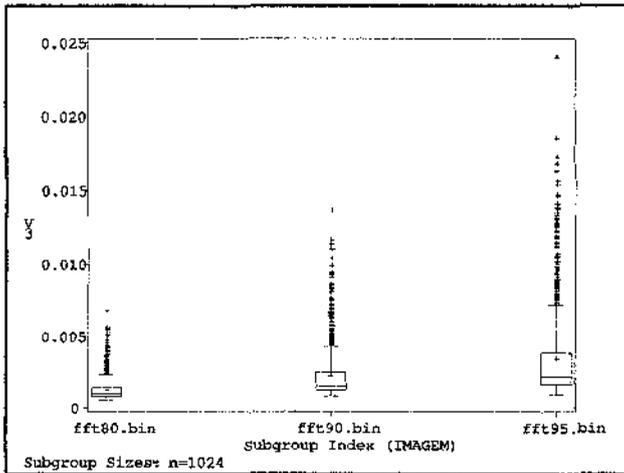


Gráfico 4.26. Box-Plot da Distância de Minkowski.

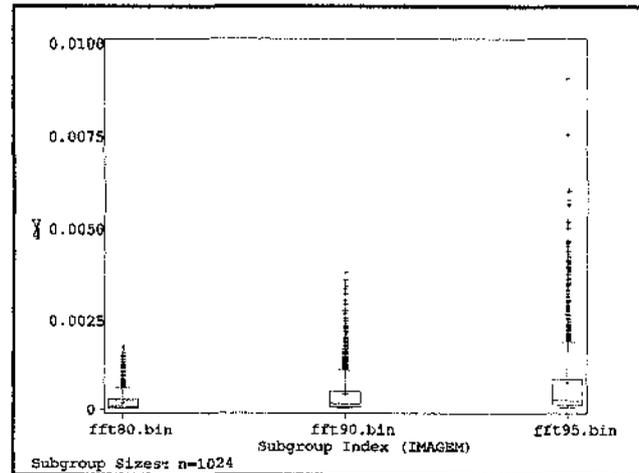


Gráfico 4.27. Box-Plot da Divergência.

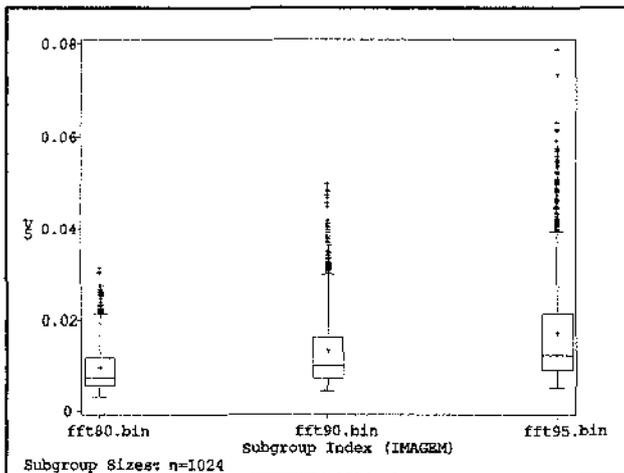


Gráfico 4.28. Box-Plot da Métrica de Canberra.

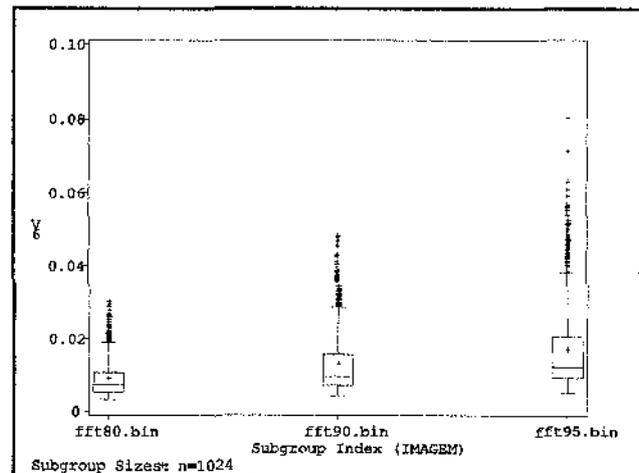


Gráfico 4.29. Box-Plot da Distância de Bray e Curtis.

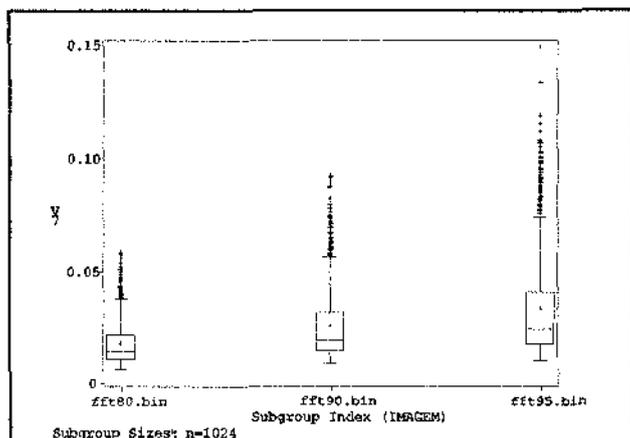


Gráfico 4.30. Box-Plot da Distância de Soergel.

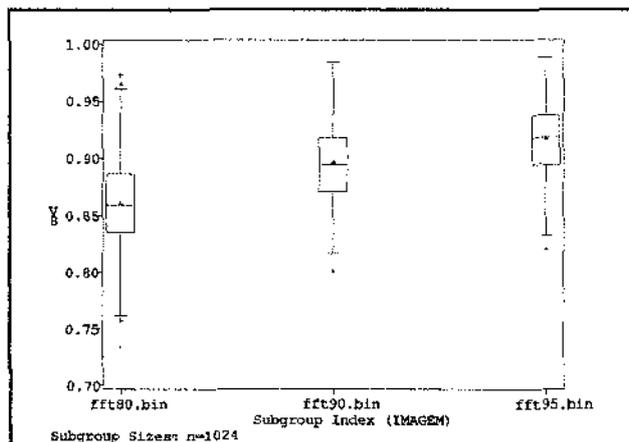


Gráfico 4.31. Box-Plot da Distância de Ware e Hedges.

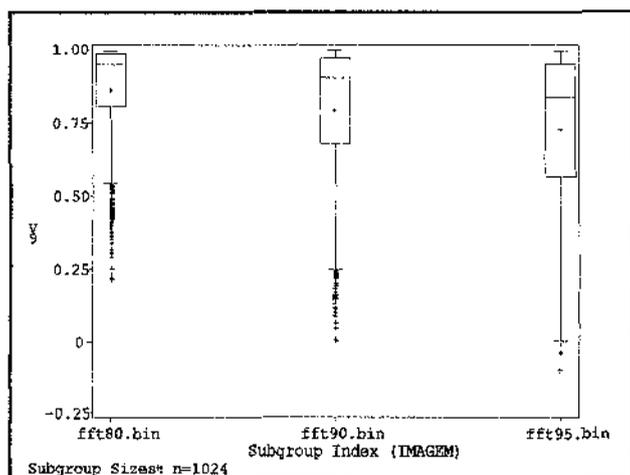


Gráfico 4.32. Box-Plot da Correlação.

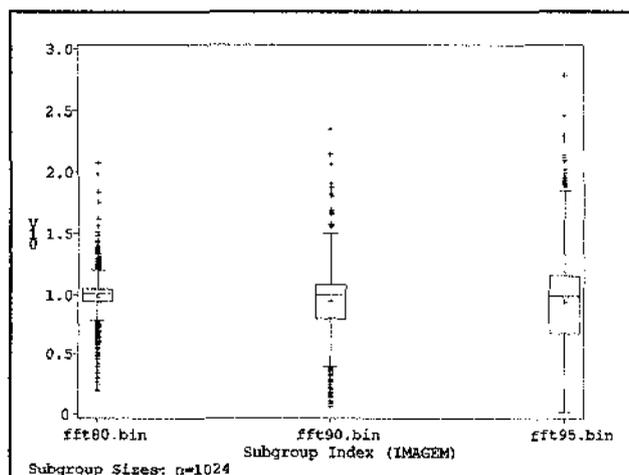


Gráfico 4.33. Box-Plot do F de Snedecor.

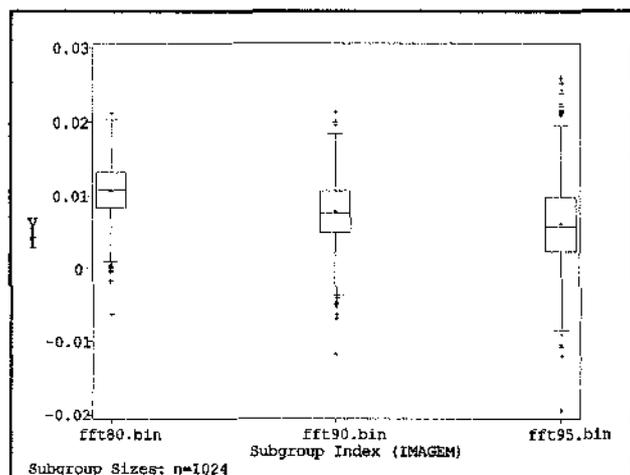


Gráfico 4.34. Box-Plot do t de Student.

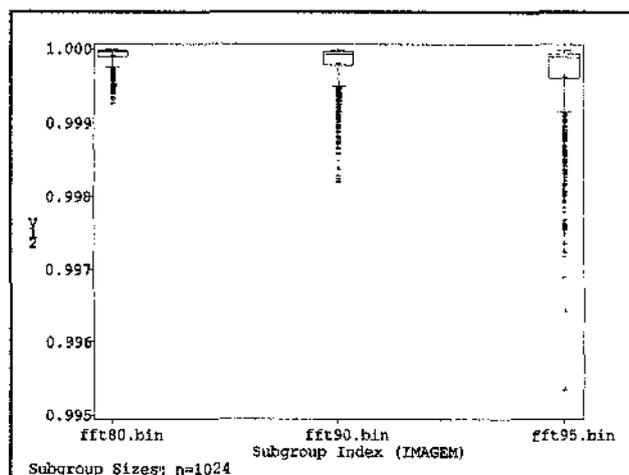


Gráfico 4.35. Box-Plot da Afinidade.

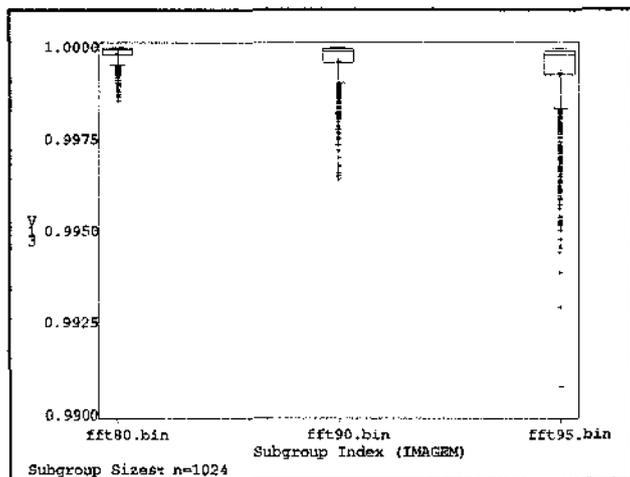


Gráfico 4.36. Box-Plot da Atração.

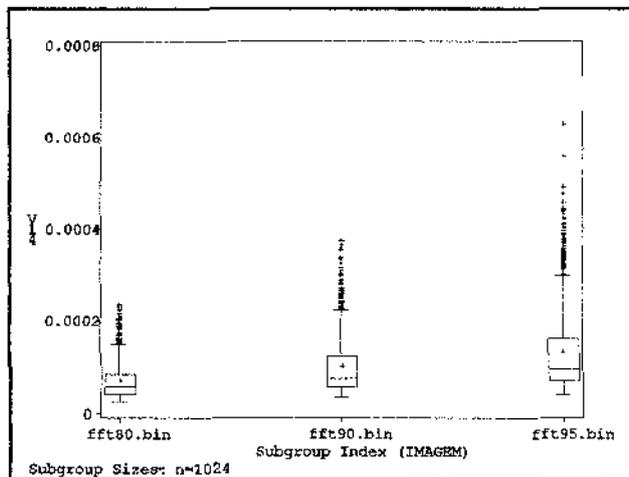


Gráfico 4.37. Box-Plot da Distância 1 de Mathai.

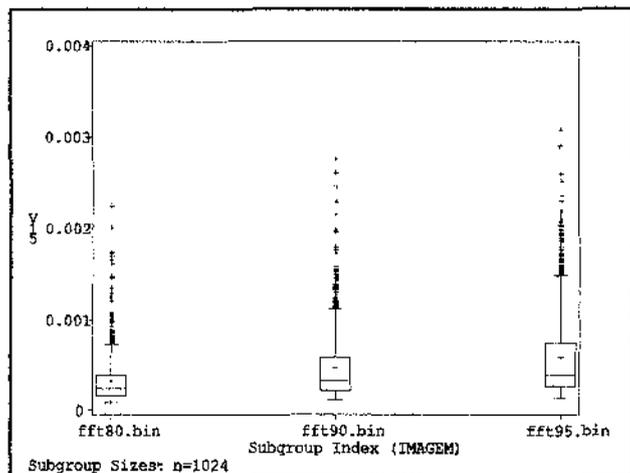


Gráfico 4.38. Box-Plot da Distância 2 de Mathai.

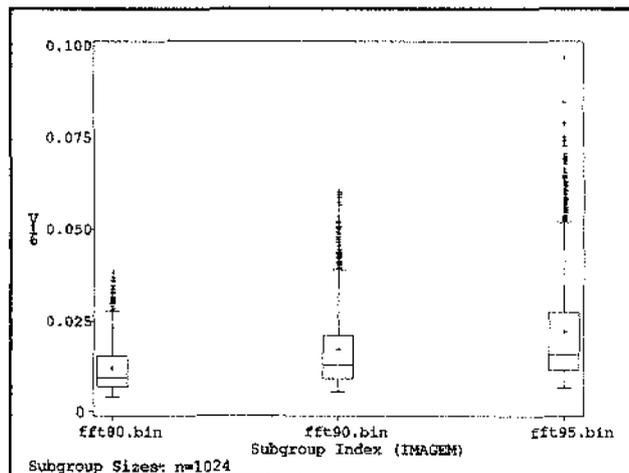


Gráfico 4.39. Box-Plot da Distância de Matusita.

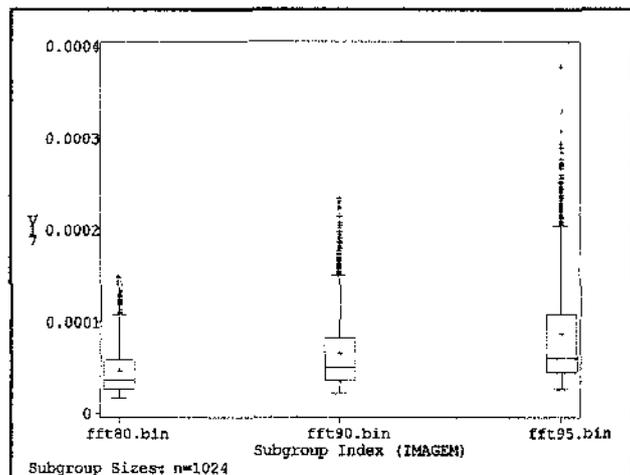


Gráfico 4.40. Box-Plot da Distância 3 de Matusita.

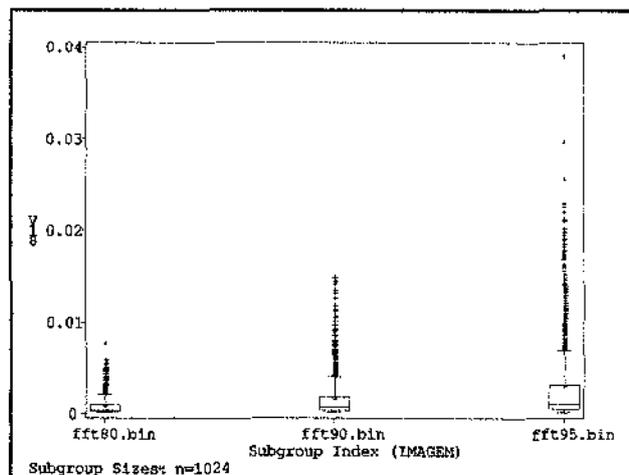


Gráfico 4.41. Box-Plot da Discrepância.

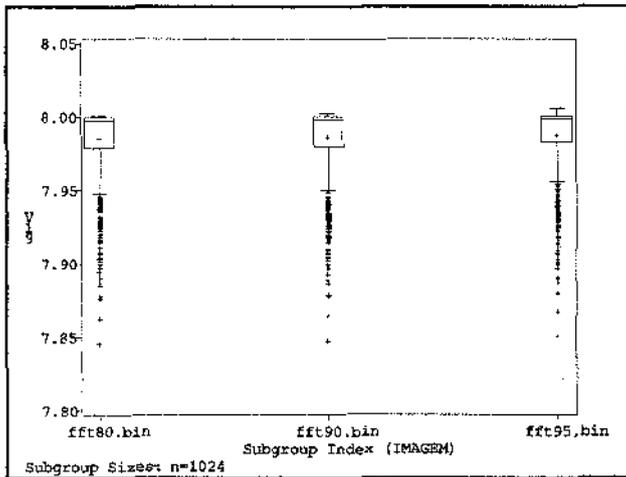


Gráfico 4.42. Box-Plot da Imprecisão.

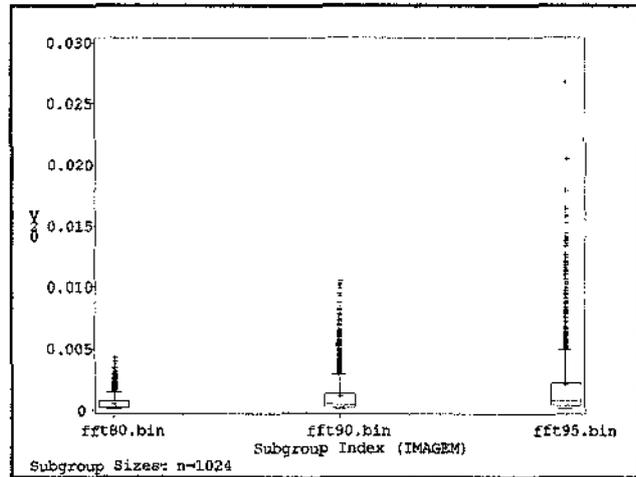


Gráfico 4.43. Box-Plot da Divergência Dirigida.

Através dos box-plots pudemos ver que algumas variáveis diferenciavam melhor as imagens para os valores do 1º quartil: V8, V9, V12, V13 e V19, enquanto as restantes diferenciavam melhor para valores acima do 3º quartil.

A partir dessa informação, eliminamos 75% das observações, mantendo as mais importantes por variável (isto é, maiores distorções). Para observarmos se houvera uma melhora do conjunto de dados, refizemos a análise *stepdisc* discriminante cujos resultados encontram-se na tabela (4.2.7).

Pudemos ver que 9 variáveis discriminaram todos os métodos, resultado bem superior ao obtido quando tomávamos todas as observações juntas. A partir desse resultado, voltamos à procura de índices de distorção. Quando observávamos o comportamento das nossas "novas" variáveis, víamos que elas agora separavam os métodos, ordenando-os de maneira levemente distinta.

Tabela 4.2.7. Valores de R^2 parciais das variáveis que entraram no modelo a um nível de significância de 15%, para blocos de tamanho $2^4 \times 2^4$ para o novo conjunto de dados.

Variável	Método de Compressão Utilizado						
	DCT	FFT	WAV	JPG	FTQ	FRC	FRAC
V1	0,3037	0,0298	0,0238	-	0,0511	0,1267	-
V2	0,0213	0,0428	0,1047	0,0856	0,0099	0,0126	0,0199
V3	0,0573	0,0133	0,0414	0,1265	0,5568	0,9943	0,0507
V4	-	-	0,0488	0,0269	0,0131	0,0610	0,0321
V5	0,0314	0,0799	0,0089	0,0376	-	-	0,0348
V6	-	0,0699	0,1285	0,1424	0,0591	0,4983	-
V7	-	0,0112	0,3572	0,4996	0,0172	0,4108	-
V8	0,8445	0,7459	0,7653	0,8229	0,8441	0,0387	0,2843
V9	0,4811	0,2280	0,0887	0,6318	0,8973	0,4181	0,0825
V10	-	0,1413	0,3508	0,3302	0,7062	0,9595	0,0379
V11	0,2992	0,4901	0,8194	0,3233	0,6482	0,9717	0,1409
V12	0,2467	-	0,0635	0,0176	0,0063	0,0692	0,9242
V13	0,1947	0,5202	0,0910	0,0294	0,0691	0,2140	-
V14	0,1240	-	-	0,0327	-	0,1386	0,0061
V15	0,2929	0,2309	0,0491	0,1040	0,7447	0,1675	0,0355
V16	-	0,6254	0,3620	0,0651	0,0222	0,0572	0,0350
V17	0,4545	-	-	0,0179	-	0,0063	-
V18	0,2830	0,2944	0,0238	0,0378	0,0165	0,1209	0,0256
V19	0,9713	0,9715	0,0674	0,0490	0,1116	0,5139	0,0150
V20	0,1435	0,0160	0,2328	0,0073	0,0159	0,1663	0,3156

4.3 Qualidade Subjetiva das Imagens

Uma boa medida de distorção deveria ser aquela cujo resultado se aproximasse do que é visto pelo olho humano. Qual seria a melhor imagem sob o aspecto visual ? Sabíamos quais eram boas e quais eram ruins, mas decidir em qual ordem deveriam estar era um pouco mais difícil, já que a medida de distorção ideal deveria seguir o que era visto pela maior parte das pessoas (inter-subjetividade). Resolvemos, portanto, fazer uma pesquisa de opinião para avaliar preferências, a qual foi realizada da seguinte forma:

Visto que uma imagem perde muito em sua resolução quando imprimida, decidimos fazer a pesquisa na tela do próprio computador. Para tal, cada imagem deveria ter um nome diferente, que não indicasse nem qual método foi utilizado nem qual sua qualidade. Por este motivo, foram sorteadas 4 letras do alfabeto para cada imagem, utilizando-se um gerador de números aleatórios. Na Tabela (A.1) do Apêndice, podemos ver qual nome foi associado à cada imagem. Em seguida, sorteamos 8 seqüências nas quais as imagens seriam mostradas. Isto foi feito para verificar se a ordem em que as imagens eram apresentadas afetavam o resultado. As seqüências encontram-se na Tabela (A.2) do Apêndice.

A entrevista durava cerca de 40 minutos por pessoa, e era feita da seguinte forma: sorteávamos uma seqüência a ser utilizada e ordenávamos as imagens segundo este sorteio. A cada imagem mostrada a pessoa falava se era melhor ou pior que alguma das imagens anteriores e seqüencialmente as anteriores iam sendo mostradas até que a pessoa a colocasse na posição correspondente à sua qualidade (por exemplo, as imagens mais à direita eram melhores). Ao final, da pior para a melhor imagem, as pessoas passavam dando notas de 0 a 10, que podiam ser iguais e modificar a ordem já estabelecida - para o caso de se desejar efetuar eventuais correções. Após entrevistarmos 24 pessoas, padronizamos suas notas. As respostas dos questionários encontram-se no Apêndice A.

A partir das notas padronizadas, procuramos ver se havia alguma separação das respostas por pessoa, já que aproximadamente 50% das pessoas entrevistadas alegavam nunca terem trabalhado com imagens em computador. Não houve separação que se mostrasse relevante. Por este motivo, calculamos a média e a mediana das notas por imagem e as ordenamos tanto pela média como pela mediana. Estes resultados podem ser observados na tabela (4.3.1):

Tabela 4.3.1. Imagens ordenadas pela média das notas.

Imagem	Média	Mediana	Imagem	Média	Mediana
WAV80	0,93	1,00	JPG35	0,62	0,67
WAV90	0,88	0,94	WAV95	0,56	0,56
FTQ50	0,85	0,83	FRAC4	0,51	0,50
JPG95	0,84	0,85	FTQ70	0,47	0,50
JPG75	0,82	0,85	DCT90	0,45	0,43
FRC8	0,75	0,80	FFT90	0,4	0,37
JPG50	0,73	0,73	FRC6	0,3	0,26
FRAC2	0,71	0,75	FTQ90	0,23	0,24
FFT80	0,70	0,72	DCT95	0,20	0,17
DCT80	0,69	0,75	FFT95	0,15	0,13
FTQ60	0,67	0,70	FRC5	0,06	0
FRAC3	0,66	0,70	-	-	-

Podemos ver que, por ambas as medidas, os resultados dessas ordenações não variam muito entre si. Assim, tomaremos as médias das notas como seu escore obtido na ordenação. Quanto mais alto o escore, melhor sua qualidade visual. Adicionando esses valores correspondentes a cada imagem como uma variável adicional, desta vez obtivemos resultados positivos.

Utilizando o software estatístico S-PLUS, inicialmente construímos uma árvore de regressão⁵⁵ utilizando todas as variáveis, sendo que a variável dependente era a que continha os escores da tabela (4.3.1). De todas as variáveis, somente as variáveis V8, V9, V10, V11, V13 e V15 foram significativas para a construção da árvore. Como o erro (igual a 0,002763) era bem pequeno, fomos eliminando variáveis e eventuais ramos da árvore até ficarmos com uma que dependia apenas das variáveis V8, V11, V13 e V15, que correspondem à distância Ware e Hedges, t de Student, atração e distância 2 de Mathai. O erro correspondente a essa árvore final, vista a seguir (gráfico

⁵⁵ Uma árvore de regressão pode ser vista como uma coleção de várias regras dispostas sob a forma de uma árvore binária. Tais regras são determinadas por um procedimento conhecido como particionamento recursivo. Mais informações podem ser obtidas em: Statistical Sciences, *S-PLUS Guide to Statistical and Mathematical Analysis*, version 3.2, Seattle: StatSci, a dimension of Math Soft, Inc., 1993.

4.44), foi correspondente à 0,0061:

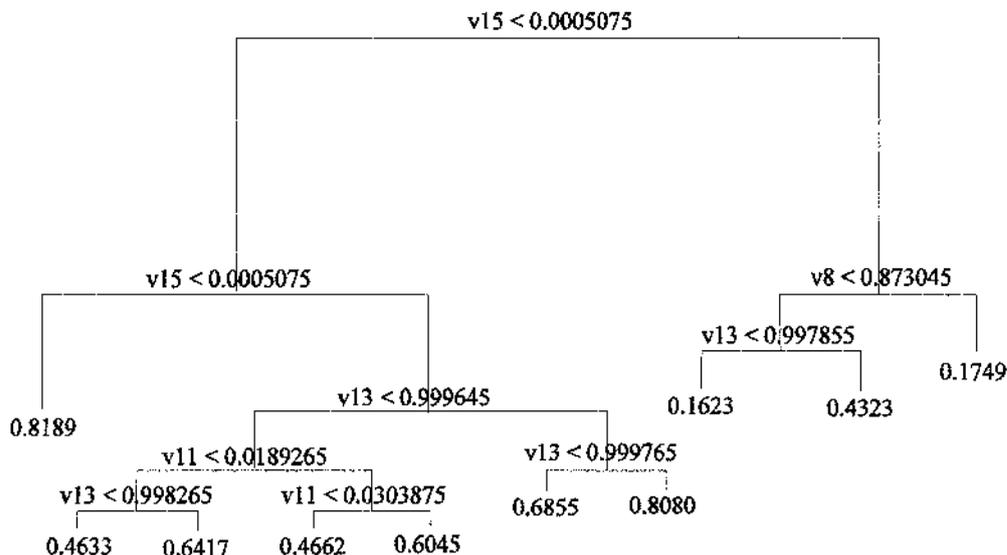


Gráfico 4.44. Árvore de Classificação Final.

Observando novamente a tabela (4.3.1), utilizamos o seguinte critério:

<p> score \leq 0,30: qualidade ruim 0,30 < score \leq 0,60: qualidade média 0,60 < score \leq 0,70: qualidade boa score > 0,70: qualidade ótima. </p>
--

Quadro 4.1. Critério para Transformação do Escore em uma Medida Qualitativa.

A partir deste, testamos nossas observações na árvore. Utilizando o ponto médio de cada intervalo acima, através de sua multiplicação com o percentual observado na classe, estimamos o escore e a qualidade para cada imagem. O resultado dessas operações pode ser visto na tabela (4.3.2):

Tabela 4.3.2. Percentuais de classificação em cada qualidade e escore estimado.

Imagem	Percentual de Classificação em cada Qualidade				Score Original	Score Estimado
	Ótima	Boa	Média	Ruim		
WAV80	0,9961	0,0039	0	0	0,9347	0,8492
WAV90	0,9102	0,0898	0	0	0,8766	0,8320
FTQ50	0,9961	0,0039	0	0	0,8504	0,8492
JPG95	0,9961	0	0,0039	0	0,8406	0,8484
JPG75	0,6250	0,3711	0,0039	0	0,8242	0,7742
FRC8	0	0	0	1	0,7545	0,1500
JPG50	0,1875	0,8086	0,0039	0	0,7306	0,6867
FRAC2	0,1133	0,8867	0	0	0,7067	0,6726
FFT80	0,3633	0,6328	0,0039	0	0,6994	0,7218
DCT80	0,7383	0,2578	0,0039	0	0,6943	0,7968
FTQ60	0,4609	0,5391	0	0	0,6695	0,7422
FRAC3	0,0430	0,9297	0,0273	0	0,6624	0,6531
JPG35	0	0,8672	0,1328	0	0,6185	0,6234
WAV95	0	0,4609	0,5391	0	0,5593	0,5422
FRAC4	0	0,750	0,2500	0	0,5097	0,6000
FTQ70	0	0,2773	0,7227	0	0,4684	0,5055
DCT90	0	0	1	0	0,4548	0,4500
FFT90	0	0	1	0	0,3988	0,4500
FRC6	0	0	0	1	0,2983	0,1500
FTQ90	0	0,0625	0,9375	0	0,2258	0,4625
DCT95	0	0	0,0078	0,9922	0,1956	0,1523
FFT95	0	0	0,0156	0,9844	0,1486	0,1547
FRC5	0	0	0	1	0,0600	0,1500

Observando os percentuais de observações de cada imagem classificadas em cada qualidade, podemos ver que estes se aproximam muito da situação esperada. Esses percentuais podem ser considerados uma medida de distorção e só não foram eficientes para classificar as imagens obtidas pelo método de fractal, denotadas por FRC. Nestas imagens, como já pudemos ver, apesar das melhores parecerem visualmente muito boas, temos o problema dos valores das distâncias calculadas sobre os blocos terem sido muito altos. Sendo assim, a árvore não conseguiu conciliar

essas diferenças com a qualidade das imagens. Para tentar criar um escore a partir desse percentual, multiplicamos o percentual obtido em cada qualidade pelo ponto médio desta (ver quadro 4.1). O escore obtido pode ser visto na última coluna da tabela (4.3.2) e, como se pode observar, ele não se aproxima muito do escore original obtido da tabela (4.3.1) apesar de poder ser considerado uma medida de distorção. Observe que na tabela (4.3.2) a ordem de apresentação das imagens corresponde à ordem do escore original. A esta medida de distorção encontrada chamaremos MAR por ser baseada no método da árvore de regressão.

Além da abordagem através da árvore de regressão, outro tipo de análise baseada em componentes principais também produziu resultados satisfatórios. Inicialmente fizemos o cálculo das componentes principais utilizando todas as variáveis. Fazendo o gráfico das duas primeiras componentes principais que correspondiam a 73,53% do total da variação (gráfico 4.45), vimos que existia uma certa separação e ordenação principalmente em relação à média das componentes por imagem.

Por esse motivo, vimos que, se tomássemos a média ou a mediana das variáveis por imagem e fizéssemos uma regressão *stepwise* onde a variável preditora fosse o escore obtido através da pesquisa de opinião, poderíamos obter um modelo que fosse um bom preditor do escore. Note-se que as imagens correspondentes a FRC não foram incluídas pois a sua inclusão prejudicava a análise dos resultados. Seus dados serão incluídos novamente no momento de teste do método. Como ao fazermos o modelo com a mediana e a média os resultados eram muito parecidos, optamos por trabalhar com a média. Acrescentando os quadrados das médias no modelo, o melhor modelo (polinomial) obtido foi:

$$SCORE = 0,8229 + 505,01578 \bar{V}_3 + 24778 \bar{V}_4^2 - 188460628 \bar{V}_{17}^2 - 0,1603 \bar{V}_9^2$$

onde \bar{V} corresponde à média do valor de cada variável.

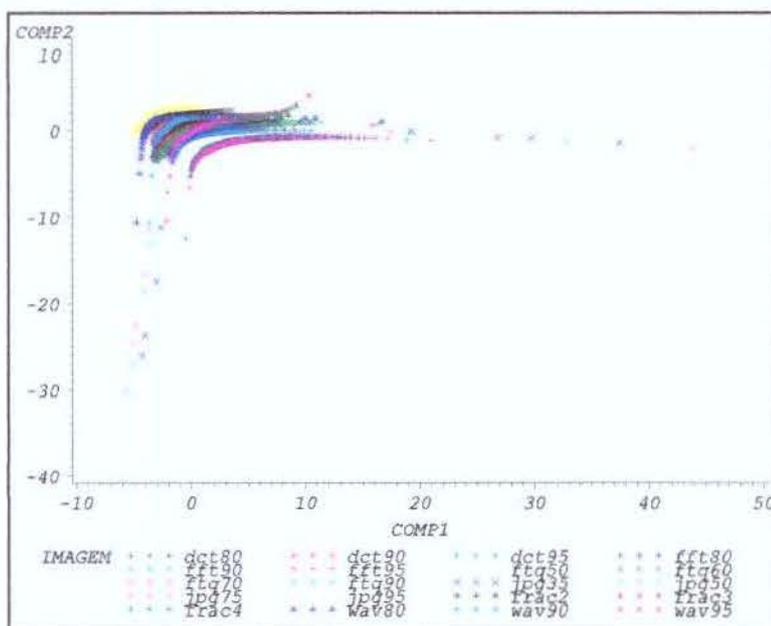


Gráfico 4.45. Gráfico das Duas Primeiras Componentes Principais.

O R^2 ajustado foi de 0,9406. Observando os valores preditos ao lado dos originais na tabela (4.3.3), vemos que esse modelo produz uma medida de distorção que, apesar de apresentar alguns erros com relação às imagem produzidas pelo método de fractais, possui uma performance razoável. A esta medida chamaremos MR, já que é uma medida de distorção baseada no método de regressão.

Tabela 4.3.3. Escore predito e o original ordenados segundo o valor predito exceto para as imagens FRC's.

Imagem	Score Predito	Score Original	Imagem	Score Predito	Score Original
WAV80	0,9445	0,9347	FRAC4	0,6082	0,5097
JPG95	0,9033	0,8401	FTQ70	0,5815	0,4684
JPG75	0,7896	0,8242	WAV95	0,5302	0,5593
WAV90	0,7887	0,866	DCT90	0,4616	0,4548
FTQ50	0,7637	0,8504	FFT90	0,4004	0,3988
DCT80	0,7121	0,6943	FTQ90	0,3449	0,2258
JPG50	0,7084	0,7306	DCT95	0,1812	0,1950
FTQ60	0,7054	0,6695	FFT95	0,1618	0,1486
FRAC3	0,6736	0,6624	FRC5	141,313	0,0600
FFT80	0,6720	0,6994	FRC6	31,6874	0,2983
FRAC2	0,6560	0,7067	FRC8	5,5325	0,7545
JPG35	0,6541	0,6185	-	-	-

Após essa última análise, estávamos já com duas medidas de distorção. Antes de continuarmos à procura de outras medidas de distorção através de outros tipos de abordagens, mostraremos um resultado interessante:

Enquanto procurávamos uma medida de distorção através da abordagem geral e por blocos, observamos que, se puséssemos esse dados numa análise discriminante canônica para discriminarmos as imagens, os erros de má classificação seriam muito pequenos. Diante de tal fato, através de uma análise *stepdisc*, ordenamos as variáveis segundo sua ordem de importância e as fomos eliminando até alcançarmos um resultado onde os erros de má classificação fossem aceitáveis. Isto aconteceria quando tivéssemos imagens de qualidade praticamente iguais sendo confundidas. O resultado foi um modelo linear discriminante contendo as variáveis V7, V8, V9 e V11 e cuja taxa de má classificação foi de 17,29 %. Os resultados dessa análise se encontram na tabela (4.3.4), cuja primeira coluna traz as imagens às quais as observações pertenciam e a primeira linha traz as imagens em que elas foram classificadas.

Analisando a tabela (4.3.4), vimos que algumas imagens tiveram uma taxa de acerto muito grande, ou seja, boa parte das observações foram bem classificadas. Olhando agora as imagens que tiveram muitos erros de má classificação, notamos que tais erros correspondem, na grande maioria das vezes, à classificação das observações em imagens que possuíam mesma qualidade visual. Isto mostra que esse modelo discriminante para separar essas imagens foi bom, já que a maioria dos erros apresentados já eram esperados devido às observações visuais já feitas sob essas imagens.

Apesar desse bom resultado, tal modelo não distingue entre os grupos de imagens qual grupo é melhor ou pior, inviabilizando, pelo menos por enquanto, a obtenção de uma medida de distorção a partir deste. Talvez em um trabalho futuro possamos obter algum resultado que seja útil na procura de medidas de distorção para avaliarmos os métodos de compressão de imagem.

Tabela 4.3.4. Resultados da análise discriminante.

-	DCT80	DCT90	DCT95	FTQ50	FTQ60	FTQ70	FTQ90	FFT80	FFT90	FFT95	JPG35	JPG50	JPG75	JPG95	FRAC2	FRAC3	FRAC4	FRC5	FRC6	FRC8	WAV80	WAV90	WAV95	
DCT80	84,31	0	0	0	0	0	0	13,73	0	0	0	0	0	0	0	1,96	0	0	0	0	0	0	0	0
DCT90	0	86,27	0	0	0	0	0	0	13,33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DCT95	0	0	65,88	0	0	0	0	0	0	34,12	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FTQ50	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FTQ60	0	0	0	13,33	74,12	12,55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FTQ70	0	0	0	0,39	16,47	72,16	10,98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FTQ90	0	0	0	0	0,39	14,9	84,71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FFT80	8,24	0	0	0	0	0	0	84,31	0	0	1,18	0	0	0	0	0,39	4,71	0	0	0	0	0	0	1,18
FFT90	0	38,43	0	0	0	0	0	0	61,18	0,39	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FFT95	0	0	56,86	0	0	0	0	0	0	43,14	0	0	0	0	0	0	0	0	0	0	0	0	0	0
JPG35	0	0	0	0	0	0	0,78	0	0	0	91,37	0	0	0	0,78	5,88	0	0	0	0	0	0	0	1,18
JPG50	0	0	0	0	0	0	0	0	0	0	0,78	95,29	0	0	3,53	0,39	0	0	0	0	0	0	0	0
JPG75	0	0	0	0	0	0	6,27	0	0	0	0	1,18	92,55	0	0	0	0	0	0	0	0	0	0	0
JPG95	0	0	0	0	0	0,78	1,18	0	0	0	0	0	0	98,04	0	0	0	0	0	0	0	0	0	0
FRAC2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
FRAC3	0	0	0	0	0	0	0	10,59	0	0	0	0	0	0	0	40,00	35,69	0	0	0	0	13,73	0	0
FRAC4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16,86	66,27	0	0	0	0	0	16,86	0
FRC5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
FRC6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
FRC8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
WAV80	0	0	0	0	0	0	0,39	0	0	0	0	0	0	0	0	0,39	0	0	0	0	97,25	1,96	0	0
WAV90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8,63	0	0	0	0	0	91,37	0	0
WAV95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40,39	0	0	0	0	0	56,61	0

4.4 Estudo da Qualidade de Bordas e Texturas

Como vimos no capítulo 3, neste trabalho vamos extrair as bordas das imagens utilizando três métodos, o de bordas finas, o do chapéu mexicano e o do filtro de Roberts. Ao aplicarmos esses algoritmos nas imagens vimos que as imagens resultantes eram formadas apenas por duas cores, preto e branco. Considerando tudo que se apresentava na cor preta de borda, estávamos interessados na análise da seguinte tabela de erros para cada imagem:

-	Imagem Reconstruída		
	-	Bordas	Não Bordas
Imagem Original	Bordas	a	b
	Não Bordas	c	d

Os principais erros de interesse eram o percentual de pixels que eram bordas na imagem original e se tornaram não bordas na imagem reconstruída (p_1), o percentual de pixels que eram não bordas na imagem original e se tornaram bordas na imagem reconstruída (p_2) e a taxa total de erro (p_3). O cálculo dessas três proporções é dado por:

$$p_1 = \frac{b}{a+b} \quad p_2 = \frac{c}{c+d} \quad p_3 = \frac{c+d}{a+b+c+d}$$

Para cada imagem, aplicamos os três algoritmos para detecção de bordas obtendo para cada uma três tipos de bordas. Em seguida, para cada, calculamos essas três proporções. A partir desses dados, da mesma forma que anteriormente, construímos uma árvore de regressão onde a variável

preditora era o escore original da tabela (4.3.2). O resultado foi uma árvore formada somente por dois percentuais, pr2 correspondente à proporção p_2 nas imagens resultantes dos algoritmo de Roberts para extração de bordas e pb1 que corresponde à proporção p_1 nas imagens resultantes do algoritmo de bordas finas. O erro foi de 0,0305 e a árvore pode ser vista no gráfico (4.46).

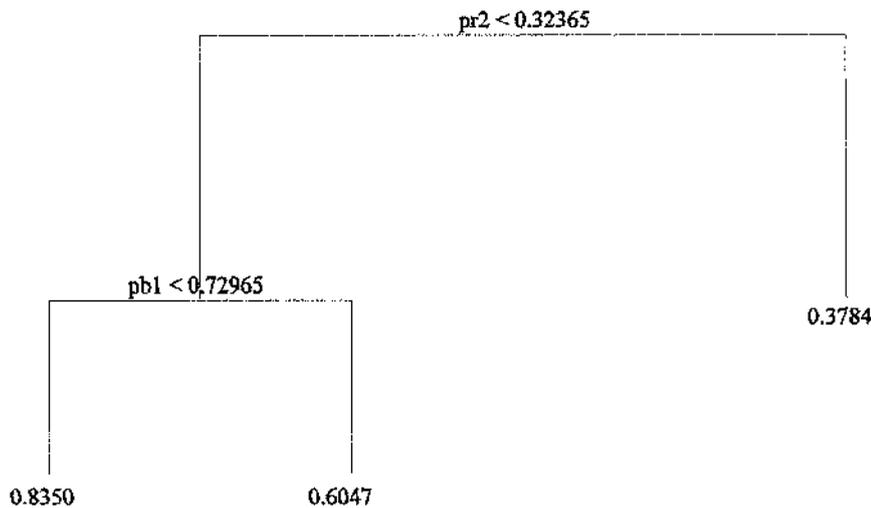


Gráfico 4.46. Árvore de Regressão para Análise de Bordas.

Como nessa árvore temos três classes diferentes, vamos considerar as imagens cujos resultados estiverem na folha de valor 0,3784, imagens de qualidade ruim, imagens que estiverem na folha de valor 0,6047 de qualidade boa e as restantes serão de ótima qualidade. Seguindo essa regra classificamos nossas imagens e o resultado se encontra na tabela (4.4.1). Como se pode observar, como medida de distorção, essa medida baseada em bordas não é muito eficiente. Apesar de classificar bem algumas imagens, ela deixa algumas como FTQ90, FTQ50, FRAC4 e FRAC2 com a mesma classificação o que seria inaceitável se pensarmos que desejamos obter uma medida de distorção que seja razoavelmente eficiente.

Como a abordagem da árvore de regressão não se mostrou eficiente, procuramos então encontrar um modelo de regressão que pudesse estimar o escore. No entanto, apesar dos esforços,

não encontramos nenhum modelo que desse uma boa aproximação para o escore, possivelmente porque os algoritmos para bordas não são suficientemente eficientes.

Tabela 4.4.1. Qualidade estimada através do método de análise de bordas.

Imagem	Qualidade Estimada	Imagem	Qualidade Estimada
DCT80	ruim	FTQ60	boa
DCT90	ruim	FTQ70	boa
DCT95	ruim	FTQ90	boa
FFT80	ótima	FRAC2	boa
FFT90	ruim	FRAC3	boa
FFT95	ruim	FRAC4	boa
WAV80	ótima	JPG35	boa
WAV90	ótima	JPG50	boa
WAV95	boa	JPG75	ótima
FTQ50	boa	JPG95	ótima

Em vista desses resultados, partimos ao estudo das texturas. Como vimos no capítulo 1, uma imagem é formada por várias texturas distintas. A nossa idéia de procurar uma medida de distorção pesquisando somente as texturas da imagens estava relacionada ao fato de que cada método de compressão afeta as imagens de maneira distinta. Sendo assim, as texturas de cada imagem também deveriam sofrer a mesma alteração. Selecionamos as texturas da imagem como mostrado na figura (3.4) do capítulo 3 e, para cada imagem reconstruída, calculamos as distâncias entre a textura original e a reconstruída. No entanto, a principal dificuldade encontrada foi que as amostras de texturas eram muito pequenas, fornecendo-nos poucos dados. Por esse motivo, ao invés de procurarmos uma medida de distorção entre as texturas, resolvemos aplicar as duas últimas medidas encontradas para estudar essas texturas. Aplicando a medida MAR (correspondente à árvore de regressão) aos dados das distâncias calculadas em blocos de tamanho $2^4 \times 2^4$ obtivemos, como anteriormente, a proporção de blocos classificados em cada qualidade e, a partir desses dados, estimamos um escore como mostrado na tabela (4.4.2).

Tabela 4.4.2. Escore Estimado para as Texturas.

Imagem	Textura						
	1	2	3	4	5	6	7
DCT80	0,8000	0,8100	0,7333	0,8056	0,8000	0,7500	0,7500
DCT90	0,8000	0,8100	0,5000	0,6278	0,8000	0,7500	0,7500
DCT95	0,8000	0,8100	0,5000	0,4500	0,8000	0,7500	0,7500
FFT80	0,8000	0,8100	0,6333	0,8056	0,8000	0,7500	0,7500
FFT90	0,8000	0,8100	0,5000	0,5389	0,8000	0,7500	0,7500
FFT95	0,8000	0,8100	0,5000	0,4500	0,8000	0,7500	0,7500
WAV80	0,8000	0,8100	0,8000	0,8056	0,8000	0,7500	0,7500
WAV90	0,8000	0,8100	0,8000	0,7167	0,8000	0,7500	0,7500
WAV95	0,8000	0,8100	0,5000	0,4944	0,8000	0,7500	0,7500
FRC5	0,6500	0,6500	0,6500	0,6500	0,6500	0,6500	0,6500
FRC6	0,7667	0,7300	0,6500	0,6500	0,6500	0,6833	0,7500
FRC8	0,8000	0,8100	0,6667	0,8056	0,8000	0,7500	0,7500
FRAC2	0,8000	0,8100	0,5333	0,5389	0,8000	0,7500	0,7500
FRAC3	0,8000	0,8100	0,5333	0,5389	0,8000	0,7500	0,7500
FRAC4	0,8000	0,8100	0,5333	0,5833	0,8000	0,7500	0,7500
JPG35	0,8000	0,8100	0,5000	0,5389	0,8000	0,7500	0,7500
JPG50	0,8000	0,8100	0,5667	0,6722	0,8000	0,7500	0,7500
JPG75	0,8000	0,8100	0,7000	0,7611	0,8000	0,7500	0,7500
JPG95	0,8000	0,8100	0,8000	0,8056	0,8000	0,7500	0,7500
FTQ50	0,8000	0,8100	0,7000	0,8056	0,8000	0,7500	0,7500
FTQ60	0,8000	0,8100	0,7000	0,8056	0,8000	0,7500	0,7500
FTQ70	0,8000	0,8100	0,5667	0,6722	0,8000	0,7500	0,7500
FTQ90	0,8000	0,8100	0,5000	0,5833	0,8000	0,7500	0,7500

Observando esse resultado por textura, vimos que as texturas 1, 2, 5, 6 e 7 foram sempre bem reconstruídas, com exceção das correspondentes imagens FRC5 e FRC6, lembrando que estas texturas correspondem à faixa do chapéu, pele do ombro, fundo da imagem, detalhe branco e chapéu. Observando estas texturas nas imagens originais, vimos que, de fato, elas foram pouco afetadas. As texturas 3 e 4, correspondentes ao detalhe do chapéu e ao cabelo, foram afetadas de formas distintas pelos métodos de compressão. No entanto, em geral, à medida que a qualidade da imagem melhorava, estas também eram melhor reconstruídas.

Voltando ao caso das imagens FRC5 e FRC6, vimos que as texturas nestas imagens realmente não se apresentavam boas, e portanto o resultado era esperado.

Em seguida, aplicamos a medida MR às texturas. No entanto, não conseguimos obter resultados satisfatórios pois agora o escore obtido não diferenciava a qualidade entre as texturas. Possivelmente esse resultado está associado ao fato das amostras terem sido muito pequenas.

4.5 Comparação dos nossos Resultados com o PNSR

Uma vez tendo concluído a procura de medidas de distorção, passamos à comparação de nossos resultados como PNSR. Os resultados do cálculo do EQM, e o PNSR entre a imagem original e a reconstruída juntamente com os escores obtidos pelas nossas medidas de distorção encontram-se na tabela (4.5.1).

Inicialmente, ordenando o PNSR com relação à qualidade das imagens, vimos que através desse índice a melhor imagem era a FTQ50. Comparando essa ordem com a obtida na tabela (4.3.1), observamos que o principal problema surgido fora com relação às imagens correspondentes ao método fractal, referenciadas como FRC. Além disso, as imagens WAV80 e WAV90 pelo PNSR não são mais consideradas as melhores. É interessante observar que, para as piores imagens, FFT95, DCT95, FTQ90, FFT90 e DCT90, essa ordem foi muito próxima em ambos os métodos.

Comparando então o PNSR com as nossas medidas de distorção obtidas, podemos ver que, em relação à ordem de classificação das imagens em termos de qualidade, a medida obtida através da árvore de regressão apresentou resultados semelhantes aos do PNSR.

Com relação à outra medida, baseada na regressão, vemos que esses resultados já não foram tão próximos. Podemos ver que, assim como o PNSR, nossas medidas não souberam classificar bem as imagens FRC8, FRC6 e FRC5. No entanto, os escores estimados por nossas medidas de uma maneira geral se aproximaram mais dos escores obtidos através da medida subjetiva. Sendo

assim, podemos concluir que o PNSR não é uma medida tão boa, pois seu desempenho com essas imagens foi apenas razoável.

Tabela 4.5.1. Valores do EQM, PNSR e as duas medidas de distorção encontradas.

Imagem	EQM	PNSR	MAR	MR
FTQ50	5,75	40,534	0,849	0,764
JPG95	5,83	40,474	0,848	0,903
WAV80	8,46	38,867	0,849	0,945
FTQ60	9,22	38,483	0,742	0,705
DCT80	9,89	38,178	0,797	0,712
FFT80	12,39	37,200	0,722	0,672
JPG75	12,43	37,186	0,774	0,789
WAV90	12,49	37,165	0,832	0,789
FTQ70	14,82	36,422	0,505	0,582
FRAC2	15,34	36,272	0,673	0,656
FRAC3	16,56	35,940	0,653	0,674
JPG50	16,94	35,842	0,687	0,708
FRAC4	18,08	35,559	0,600	0,608
WAV95	21,92	34,722	0,542	0,530
JPG35	22,04	34,699	0,623	0,654
DCT90	23,08	34,498	0,450	0,462
FFT90	26,85	33,841	0,450	0,400
FTQ90	28,09	33,645	0,463	0,345
DCT95	46,25	31,479	0,152	0,181
FFT95	49,37	31,196	0,154	0,162
FRC8	99,42	28,156	0,150	5,533
FRC6	340,08	22,815	0,150	31,687
FRC5	726,89	19,516	0,150	141,313

Capítulo 5

Conclusão

5.1 Discussão dos Resultados Obtidos e Conclusão

O objetivo principal deste trabalho foi encontrar novas medidas de distorção estatísticas, que pudessem avaliar a qualidade de algoritmos de compressão. Inicialmente, escolhemos quatro tipos de algoritmos: os que se baseavam na Transformação Cosseno Discreta, na Transformada Rápida de Fourier, na Transformada Wavelets e em Fractais. Em seguida, a partir desses algoritmos, geramos 23 imagens que se dividiam em três grupos de qualidade visual: as muito boas, as médias e as ruins. Produzindo imagens com tipos de degradação levemente distintos, os algoritmos nos permitiram testar as várias medidas sob diferentes aspectos.

Em seguida, começamos a nossa procura em quatro grupos diferentes de medidas: medidas de distância, medidas de informação, medidas para análise descritiva e medidas de avaliação de texturas e bordas. Selecionando várias medidas de cada grupo, especificamos o tipo de abordagem a ser utilizada para a aplicação de cada uma. Dos três primeiros grupos, escolhemos 20 medidas para serem aplicadas nas imagens na “abordagem por blocos”, que consista em trabalharmos com a imagem dividida em blocos, aplicando as medidas sobre estes. Numa abordagem referenciada como

“abordagem geral”, aplicamos uma medida de distância (distância de Mahalanobis) às imagens em sua forma original. Este mesmo tipo de abordagem foi utilizada na aplicação das medidas pertencentes ao quarto grupo (medidas para avaliar texturas e bordas).

Iniciamos as análises trabalhando com as medidas dos três primeiros grupos. Apesar das várias tentativas de encontrar uma medida dentre as selecionadas, não obtivemos nenhum resultado que considerássemos satisfatório. Nenhuma medida conseguia explicar a variabilidade existente entre as imagens com diferentes graus de qualidade visual. Observando os gráficos gerados, notamos que algumas imagens, ao invés de apresentarem uma degradação uniforme em sua qualidade, apresentavam apenas alguns detalhes muito mal reconstruídos. Para contornar esse problema, passamos a trabalhar apenas com as observações mais afetadas pelos métodos, criando assim um novo conjunto de dados a partir do original.

As análises feitas sobre esse novo conjunto apresentaram resultados significativamente melhores. Por este motivo continuamos a procura de uma nova medida, utilizando técnicas como análise de componente principais, análise de regressão, análise discriminante e árvore de regressão. Todas as possíveis medidas foram aplicadas às imagens selecionadas.

Após algumas análises terem sido feitas, notamos que não havia consistência nos resultados: uma mesma imagem era classificada como boa por uma medida e como ruim por outra. Uma pesquisa de opinião mostrou-se a melhor opção para a solução deste problema. Através dela, foi possível obter uma medida subjetiva de qualidade que foi utilizada como referência para as outras. Com a ajuda desta, encontramos duas novas medidas de distorção estatísticas.

Ao aplicarmos essas medidas sobre nossas imagens, obtivemos resultados muito semelhantes aos produzidos por nossa medida de referência. Essas medidas receberam os nomes de MAR e MR por terem sido obtidas a partir do método da árvore de regressão e do método de regressão, respectivamente. Estes modelos utilizavam variáveis associadas às 20 medidas estatísticas propostas, sendo cada uma que cada um utilizou diferentes variáveis. Além disso, pelo menos uma

medida de cada um dos três grupos (medidas de distância, medidas de informação e medidas para análise descritiva) foi considerada significativa para a obtenção desse resultado final, confirmando a adequação da escolha desses grupos para o desenvolvimento do trabalho.

Analisando novamente os resultados obtidos por essas medidas quando aplicadas às nossas imagens, e comparando os resultados com os produzidos pela medida subjetiva, pudemos ver que em geral essas medidas apresentaram uma boa performance, principalmente quando comparamos os resultados aos obtidos utilizando as medidas clássicas - PNSR e EQM. Isto nos permitiu chegar à conclusão de que nosso objetivo principal foi alcançado no sentido de termos conseguido desenvolver novas medidas de distorção estatísticas que permitissem avaliar objetivamente os algoritmos de compressão de imagens.

5.2 Considerações Finais

Já havíamos mencionado no capítulo 1 a importância das bordas e texturas de uma imagem. Observando novamente as figuras selecionadas, vimos que algumas apresentavam sinais de degradação nessas duas características. Por esse motivo, se analisássemos a degradação destas duas características, possivelmente obteríamos outras medidas de distorção. Inicialmente, procuramos uma medida que, a partir da observação da degradação ocorrida nas bordas, avaliasse a degradação que ocorria nas imagens. As medidas utilizadas para essa procura pertenciam ao quarto grupo de medidas: medidas para avaliar texturas e bordas. Após várias tentativas, a melhor medida encontrada não apresentou resultados satisfatórios, ao contrário do que esperávamos. Este resultado certamente está associado à falta de precisão dos métodos de avaliação de bordas utilizados, que não conseguiram detectar as pequenas variações produzidas pelos diferentes métodos.

Procurando agora uma medida de distorção a partir da observação da degradação apresentada nas texturas, também não obtivemos nenhum resultado satisfatório. Possivelmente esse

este fato também está associado à falta de precisão das técnicas empregadas. Por esse motivo, resolvemos aproveitar as amostras de texturas para testar nossas novas medidas, MAR e MR. Aplicando-as às texturas, após analisarmos os resultados, pudemos ver que a medida MR não conseguiu distinguir os diferentes graus de qualidade das amostras, ao contrário do que aconteceu quando aplicamos a medida MAR, que distinguiu-as muito bem. Os resultados da aplicação desta última medida aproximaram-se muito do observado visualmente. Deste resultado, podemos concluir que as medidas estatísticas empregadas na construção da medida de distorção MAR (Distância de Ware e Hedges, t de Student (para a diferença entre os blocos das imagens), Atração e Distância 2 de Mathai) são mais sensíveis às degradações de texturas produzidas pelos métodos de compressão.

Através da observação de nossos resultados, vemos que a procura de novas medidas de distorção estatísticas é um tema que pode ser ainda muito explorado. A falta de tempo não nos permitiu fazer mais pesquisas e testar os resultados em outras imagens.

REFERÊNCIAS BIBLIOGRÁFICAS

- Anson, L. F. Fractal Image Compression. *Byte*. Vol. 18, nº 11, pp. 195-202, Nov. 1993.
- Baran, N. Fractal Compression Goes On-Line. *Byte*. Vol. 18, nº 10, p. 40, Set. 1993.
- Barnsley, M e Sloan, A. D. A Better Way to Compress Images. *Byte*. Vol 13, nº 1, pp.215-223, Jan. 1988.
- Barnsley, M. *Fractals Everywhere*. Boston: Academic Press Inc., 1988.
- Barthel, K. U., Schüttemeyer, J., Voyé, T. e Noll, P. A New Image Coding Technique Unifying Fractal and Transform Coding. *IEEE International Conference on Image Processing ICIP'94*. Austin, Texas, Nov. 1994, pp. 112-116.
- Bonvino, H. *Notas de Aula fornecidas durante o curso de Teoria de Informação*. Campinas: Unicamp, Mar. 1993.
- Buckheit, J. B., Chen, S., Donoho, D., Johnston, I. e Scargle, J. *About Wavelab*.
<ftp://playfer.stanford.edu/pub/wavelab>. 1995.
- Daubechies, I. *Ten Lectures on Wavelets*. SIAM, 1992.
- Davidson, C. e Rock, D. Wavelets and HONN: Pix-Perfect Marriage. *AI Expert*, The Magazine of Artificial Intelligence in Practice. Vol. 10, nº 1, pp. 31-35, Jan. 1995.
- Davis, G. Self-Quantized Wavelet Subtrees: A Wavelet-Based Theory of Fractal Image Compression. *Wavelets and Applications I*. Orlando: H. H. Szu (ed), Proc. SPIE 2491, Abr. 1995, pp. 141-152.

- De Hoyos, A. J. e Corbera, R. P. Estudio Comparativo de Operadores de Borda Utilizando el LATIM. *Proceedings of the V Latin American Remote Sensing Symposium*. Cuzco: Selper, 1991, pp. 452-458.
- Domingues, M. O., Gomes, S. M. e Cortina, E. Biorthogonal Wavelets Applied to Meteosat Image Compressing. *Wavelets and Applications II*. Orlando: H. H. Szu (ed), Proc. SPIE 2491, Abr. 1995, pp 726-733.
- Fisher, Y. *Fractal Image Compression*, Theory and Application. New York: Springer Verlag, 1995.
- Gower, J. C. e Legendre, P. Metric and Euclidian Properties of Dissimilarity Coefficients. *Journal of Classification*. Vol. 3, pp. 397-405, 1986.
- Gower, J. C. Measures of Similarity, Dissimilarity and Distance. *Encyclopaedia of Statistical Sciences*. Vol. 5, Eds. S. Katz, N. L. Johnson and C. B. Read. New York: John Wiley , pp. 397-405, 1985.
- Haralick, P. M. Statistical Image Texture Analysis. In: Young, T. Y. e Fu, K. *Handbook of Pattern Recognition and Image Processing*. San Diego: Academic Press, 1986. Cap. 11, pp. 247-279.
- Hürtgen, B. e Simon, S. F. On the Problem of Convergence in Fractal Coding Schemes. *IEEE International Conference on Image Processing ICIP'94*. Austin, Texas, Nov. 1994, pp. 103-106.
- Karunasekera, S. A. e Kingsbury, N. G. A Distortion Measure for Blocking Artifacts in Images Based on Human Visual Sensivity. *IEEE Transactions on Image Processing*, Vol. 4, n° 6, pp. 713-723, Jun. 1995.

- Kashyap, R. L. Image Models. Young, T. Y. e Fu, K. *Handbook of Pattern Recognition and Image Processing*. San Diego: 1986, Cap. 12, pp. 281-310.
- Lewinson, L. Fractal Databases; New Horizons in Database Management. *PC AI*. Pp. 30-33, Abr. 1994.
- Mallat, S. G. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 11, n° 7, pp. 674-693, Jul. 1989.
- Mallat, S. G. Multiresolution Approximations and Wavelet Orthonormal Bases of $L^2(\mathbb{R})$. *Transactions of the American Mathematical Society*. Vol. 315, n° 1, pp. 69-87, Set. 1989.
- Mandelbrot, B. B. *The Fractal Geometry of Nature*. New York: Freeman, 1983.
- Mardia, K. V., Kent, J. T. e Bibby J. M. *Multivariate Analysis*. London: Academic Press, 1979.
- Marr, D. e Hildreth, E. Theory of Edge Detection. *Proc. of Royal Society London*. Vol. B207, pp. 187-217, 1980.
- Mathai, A. M. e Rathie, P. N. *Basic Concepts in Information Theory and Statistics*. New Delhi: Wiley Eastern Limited, 1975.
- Meyer, Y. e Ryan, R. P. *Wavelets*; algorithms and applications. Philadelphia: SIAM, 1993.
- Murray, J. D. e vanRyper, W. *Encyclopedia of Graphics File Formats*. Sebastopol: O'Reilly and Associates, Inc., 1994.
- Nevatia, R. Image Segmentation. Young, T. Y. e Fu, K. *Handbook of Pattern Recognition and Image Processing*. San Diego: Academic Press, 1986, Cap. 9, pp. 215-231.

- Oien, G. E. Parameter Quantization in Fractal Image Coding. *IEEE International Conference on Image Processing ICIP'94*. Austin, Texas, Nov. 1994, pp. 142-146.
- Pratt, W. K. *Digital Image Processing*. 2nd ed. New York: John Wiley, 1991.
- Rosenfeld, A. e Kak, A. C. *Digital Picture Processing*. 2nd ed. San Diego: Academic Press, Inc., Vol 1, 1982.
- Rosiek, M. K. e MacLeod, M. L. Daubechies - Based Visual Table of Contents. *Wavelets and Applications I*. Orlando: H. H. Szu (ed), Proc. SPIE 2491, Abr 1995, pp. 72 a 77.
- Schalkoff, R. J. *Digital Image Processing and Computer Vision*. New York: John Wiley , 1989.
- Shusterman, E. e Feder, M. Image Compression Via Improved Quadtree Decomposition Algorithms. *IEEE Transactions on Image Processing*. Vol. 4, n° 6, pp. 207-215, Mar. 1994.
- Sriram, P. e Marcellin, M. Image Coding Using Wavelet Transforms and Entropy - Constrained Trellis Coded Quantization. *IEEE Transactions on Image Processing*. Vol. 4, n° 6, Jun. 1995.
- Statistical Sciences, *S-PLUS Guide to Statistical and Mathematical Analysis*; version 3.2, Seattle: StatSci, a dimension of Math Soft, Inc., 1993.
- Strang, G. e Nguyen, F. *Wavelets and Filters Banks*. Workshop Notes: San Jose State University, Jan. 13-16. 1995.
- Taswell, C. Image Compression by Parameterized Model Coding of Wavelet Packet Near-Best Bases. *Wavelets and Applications I*. Orlando: H. H. Szu (ed), Proc. SPIE 2491, Abr. 1995, pp. 153-172.

Wallace, G. K. The JPEG Still Picture Compression Standard. *Communications of the ACM*.
Vol. 34, n° 4, pp. 31-44, Abr. 1991.

Walter, G. *Wavelets and Other Orthogonal Systems with Applications*. Boca Raton: CRC Press,
1994. P. 37.

Zhuang, Y. e Baras, J. S. Image Compression Using Optimal Wavelet Basis. *Wavelets and
Applications I*. Orlando: H. H. Szu (ed), Proc. SPIE 2491, Abr. 1995, pp 131-140.

APÊNDICE A

Resultados da Pesquisa de Opinião

A seguir apresentaremos os resultados da pesquisa de opinião realizada de 16/05/96 a 25/06/96 no Laboratório de Estatística - IMECC (UNICAMP). Nas tabelas (A.1) e (A.2) vemos os códigos e seqüências de apresentação das imagens. Nas páginas seguintes as tabelas, apresentamos a respostas obtidas no momento da pesquisa.

Tabela A.1 Nome Código Associado a Cada Imagem.

Nome das Imagens			
Original	Codificado	Original	Codificado
dct80	ntuo	ftq90	qydj
dct90	dcrp	frac2	veax
dct95	rnxr	frac3	hxtk
fft80	fewq	frac4	cglb
fft90	rrpb	frc5	tnuw
fft95	irvn	frc6	bvma
wav80	vwuj	frc8	hngj
wav90	cjrv	jpg35	mtiy
wav95	ohts	jpg50	rwej
ftq50	cbnb	jpg75	luqv
ftq60	bfql	jpg95	rvgk
ftq70	lhpj	feath2	dmii

Tabela A.2 Sequências Sorteadas para a Amostra.

Sequências							
1	2	3	4	5	6	7	8
hxtk	rxnr	luqv	ohts	vwuj	rxnr	rvgk	hxtk
dcrp	ohts	cjrv	cjrv	cjrv	rwej	dmii	fewq
ntuo	dcrp	vwuj	lhpj	fewq	rvgk	irvn	ntuo
cjrv	vwuj	tnuw	vwuj	irvn	veax	mtiy	rvgk
veax	cbnb	mtiy	bfql	luqv	dcrp	hxtk	mtiy
fewq	lhpj	bvma	rvgk	hxtk	bfql	bvma	bvma
cglb	luqv	hngj	hxtk	rwej	ohts	cjrv	cglb
vwuj	dmii	rvgk	dcrp	lhpj	tnuw	hngj	luqv
ohts	rvgk	ntuo	mtiy	dcrp	qydj	bfql	dcrp
rwej	cglb	rwej	hngj	veax	irvn	tnuw	bfql
qydj	rrpb	hxtk	veax	bfql	ntuo	veax	veax
mtiy	fewq	dmii	dmii	rxnr	cjrv	ntuo	rxnr
rxnr	bfql	dcrp	rwej	qydj	cbnb	vwuj	vwuj
cbnb	bvma	lhpj	luqv	cbnb	bvma	rwej	rwej
hngj	hngj	cbnb	cglb	rrpb	lhpj	cbnb	rrpb
dmii	hxtk	fewq	rxnr	ohts	mtiy	dcrp	irvn
luqv	qydj	bfql	fewq	rvgk	vwuj	lhpj	hngj
lhpj	tnuw	irvn	qydj	tnuw	dmii	fewq	qydj
irvn	cjrv	rxnr	ntuo	hngj	cglb	rrpb	cjrv
rrpb	veax	cglb	cbnb	dmii	hngj	rxnr	ohts
rvgk	mtiy	ohts	rrpb	cglb	luqv	ohts	lhpj
bfql	irvn	rrpb	tnuw	mtiy	hxtk	cglb	dmii
bvma	rwej	qydj	bvma	ntuo	rrpb	luqv	cbnb
tnuw	ntuo	veax	irvn	bvma	fewq	qydj	tnuw

quest nº	<u>1</u>		seq nº	<u>1</u>								
RNXR	<u>23</u>	<u>3</u>	RRPB	<u>19</u>	<u>5</u>	OHTS	<u>12</u>	<u>5</u>	BVMA	<u>20</u>	<u>4</u>	
BFQL	<u>11</u>	<u>6</u>	NTUO	<u>14</u>	<u>7</u>	RVGK	<u>10</u>	<u>7</u>	CBNB	<u>2</u>	<u>9</u>	
DCRP	<u>18</u>	<u>5</u>	VEAX	<u>7</u>	<u>8</u>	FEWQ	<u>6</u>	<u>9</u>	HNGJ	<u>3</u>	<u>9</u>	
MITY	<u>17</u>	<u>6</u>	IRVN	<u>21</u>	<u>2</u>	TNUW	<u>24</u>	<u>0</u>	LUQV	<u>4</u>	<u>8</u>	
HXTK	<u>13</u>	<u>8</u>	RWEJ	<u>15</u>	<u>7</u>	LHPJ	<u>9</u>	<u>8</u>	VWUJ	<u>8</u>	<u>8</u>	
QYDJ	<u>22</u>	<u>3</u>	CJRV	<u>5</u>	<u>8</u>	CGLB	<u>16</u>	<u>6</u>	DMII	<u>1</u>	<u>10</u>	
quest nº	<u>2</u>		seq nº	<u>5</u>								
RNXR	<u>20</u>	<u>4</u>	RRPB	<u>18</u>	<u>5</u>	OHTS	<u>16</u>	<u>6</u>	BVMA	<u>19</u>	<u>4</u>	
BFQL	<u>13</u>	<u>7</u>	NTUO	<u>6</u>	<u>8</u>	RVGK	<u>7</u>	<u>8,5</u>	CBNB	<u>4</u>	<u>9</u>	
DCRP	<u>17</u>	<u>7</u>	VEAX	<u>14</u>	<u>7</u>	FEWQ	<u>3</u>	<u>9</u>	HNGJ	<u>9</u>	<u>8</u>	
MITY	<u>10</u>	<u>7,5</u>	IRVN	<u>21</u>	<u>4</u>	TNUW	<u>23</u>	<u>3</u>	LUQV	<u>5</u>	<u>8,5</u>	
HXTK	<u>12</u>	<u>7</u>	RWEJ	<u>15</u>	<u>7</u>	LHPJ	<u>22</u>	<u>3</u>	VWUJ	<u>2</u>	<u>10</u>	
QYDJ	<u>24</u>	<u>0</u>	CJRV	<u>1</u>	<u>10</u>	CGLB	<u>11</u>	<u>7</u>	DMII	<u>8</u>	<u>8</u>	
quest nº	<u>3</u>		seq nº	<u>3</u>								
RNXR	<u>22</u>	<u>1,5</u>	RRPB	<u>20</u>	<u>3</u>	OHTS	<u>13</u>	<u>3,5</u>	BVMA	<u>21</u>	<u>2</u>	
BFQL	<u>14</u>	<u>3,5</u>	NTUO	<u>11</u>	<u>5,5</u>	RVGK	<u>6</u>	<u>6,5</u>	CBNB	<u>7</u>	<u>6</u>	
DCRP	<u>19</u>	<u>3</u>	VEAX	<u>15</u>	<u>3,5</u>	FEWQ	<u>17</u>	<u>3,5</u>	HNGJ	<u>6</u>	<u>6,5</u>	
MITY	<u>8</u>	<u>6</u>	IRVN	<u>23</u>	<u>1</u>	TNUW	<u>24</u>	<u>0</u>	LUQV	<u>2</u>	<u>6,5</u>	
HXTK	<u>9</u>	<u>6</u>	RWEJ	<u>10</u>	<u>5,5</u>	LHPJ	<u>12</u>	<u>4</u>	VWUJ	<u>4</u>	<u>7</u>	
QYDJ	<u>18</u>	<u>3</u>	CJRV	<u>1</u>	<u>7,5</u>	CGLB	<u>16</u>	<u>3</u>	DMII	<u>3</u>	<u>7</u>	
quest nº	<u>4</u>		seq nº	<u>6</u>								
RNXR	<u>23</u>	<u>3</u>	RRPB	<u>18</u>	<u>7</u>	OHTS	<u>20</u>	<u>8</u>	BVMA	<u>21</u>	<u>8</u>	
BFQL	<u>1</u>	<u>9</u>	NTUO	<u>13</u>	<u>7</u>	RVGK	<u>4</u>	<u>10</u>	CBNB	<u>7</u>	<u>10</u>	
DCRP	<u>2</u>	<u>9</u>	VEAX	<u>3</u>	<u>9</u>	FEWQ	<u>12</u>	<u>7</u>	HNGJ	<u>11</u>	<u>9</u>	
MITY	<u>15</u>	<u>6</u>	IRVN	<u>22</u>	<u>2</u>	TNUW	<u>24</u>	<u>7</u>	LUQV	<u>10</u>	<u>9</u>	
HXTK	<u>14</u>	<u>6</u>	RWEJ	<u>6</u>	<u>10</u>	LHPJ	<u>16</u>	<u>4</u>	VWUJ	<u>9</u>	<u>10</u>	
QYDJ	<u>19</u>	<u>1</u>	CJRV	<u>5</u>	<u>10</u>	CGLB	<u>17</u>	<u>1</u>	DMII	<u>8</u>	<u>10</u>	

quest nº 5		seq nº 6									
RNXR	20	3	RRPB	16	4	OHTS	15	3,75	BVMA	19	3,75
BFQL	5	6	NTUO	10	4,25	RVGK	6	5,5	CBNB	3	7
DCRP	17	4	VEAX	11	4,25	FEWQ	12	4,25	HNGJ	8	4,75
MITY	13	4	IRVN	22	3,5	TNUW	24	3	LUQV	7	5,5
HXTK	14	4	RWEJ	9	4,5	LHPJ	21	3	VWUJ	2	7
QYDJ	23	2,5	CJRV	4	6	CGLB	18	4	DIII	1	8
quest nº 6		seq nº 5									
RNXR	22	3,5	RRPB	20	4	OHTS	14	5,25	BVMA	21	3,75
BFQL	9	6	NTUO	16	5	RVGK	12	5,5	CBNB	10	5,75
DCRP	19	4	VEAX	13	5,5	FEWQ	2	7	HNGJ	11	5,75
MITY	15	5	IRVN	23	3	TNUW	24	3	LUQV	1	7
HXTK	7	6	RWEJ	6	6,5	LHPJ	8	6	VWUJ	4	7
QYDJ	18	4	CJRV	3	7	CGLB	17	4,5	DIII	5	7
quest nº 7		seq nº 3									
RNXR	22	2,5	RRPB	19	3	OHTS	14	4,5	BVMA	20	3
BFQL	10	4,75	NTUO	7	5,25	RVGK	6	6	CBNB	11	5
DCRP	17	3,5	VEAX	13	4,5	FEWQ	12	4,25	HNGJ	4	5,25
MITY	5	5,25	IRVN	24	0,5	TNUW	23	1	LUQV	2	6
HXTK	15	4	RWEJ	8	5,25	LHPJ	16	3,5	VWUJ	3	6,5
QYDJ	21	2,5	CJRV	1	6,5	CGLB	18	3	DIII	9	5,5
quest nº 8		seq nº 1									
RNXR	22	3	RRPB	17	4,1	OHTS	15	5	BVMA	19	3,8
BFQL	10	6,3	NTUO	11	6	RVGK	6	7,4	CBNB	7	7,2
DCRP	16	4,5	VEAX	2	8,7	FEWQ	12	5,7	HNGJ	9	6,6
MITY	14	5,2	IRVN	23	2,5	TNUW	24	2	LUQV	8	6,7
HXTK	2	8,3	RWEJ	13	5,5	LHPJ	18	4,2	VWUJ	5	8
QYDJ	21	3,5	CJRV	3	9	CGLB	20	4	DIII	4	8,5

quest nº	<u>9</u>		seq nº	<u>3</u>							
RNXR	<u>21</u>	<u>5</u>	RRPB	<u>18</u>	<u>7</u>	OHTS	<u>16</u>	<u>7</u>	BVMA	<u>19</u>	<u>6</u>
BFQL	<u>7</u>	<u>8</u>	NTUO	<u>11</u>	<u>8</u>	RVGK	<u>2</u>	<u>9</u>	CBNB	<u>4</u>	<u>9</u>
DCRP	<u>17</u>	<u>7</u>	VEAX	<u>13</u>	<u>8</u>	FEWQ	<u>9</u>	<u>8</u>	HNGJ	<u>12</u>	<u>9</u>
MITY	<u>15</u>	<u>8</u>	IRVN	<u>22</u>	<u>5</u>	TNUW	<u>23</u>	<u>4</u>	LUQV	<u>8</u>	<u>9</u>
HXTK	<u>5</u>	<u>8</u>	RWEJ	<u>6</u>	<u>8</u>	LHPJ	<u>20</u>	<u>5</u>	VWUJ	<u>3</u>	<u>10</u>
QYDJ	<u>24</u>	<u>4</u>	CJRV	<u>10</u>	<u>8</u>	CGLB	<u>14</u>	<u>8</u>	DMII	<u>1</u>	<u>9</u>
quest nº	<u>10</u>		seq nº	<u>4</u>							
RNXR	<u>22</u>	<u>6</u>	RRPB	<u>18</u>	<u>6,5</u>	OHTS	<u>1</u>	<u>8</u>	BVMA	<u>20</u>	<u>6</u>
BFQL	<u>11</u>	<u>7</u>	NTUO	<u>16</u>	<u>7</u>	RVGK	<u>6</u>	<u>8</u>	CBNB	<u>5</u>	<u>7</u>
DCRP ¹⁷	<u>17</u>	<u>6,5</u>	VEAX	<u>7</u>	<u>7,5</u>	FEWQ	<u>15</u>	<u>7,5</u>	HNGJ	<u>8</u>	<u>7</u>
MITY	<u>13</u>	<u>7</u>	IRVN	<u>23</u>	<u>5</u>	TNUW	<u>24</u>	<u>3</u>	LUQV	<u>10</u>	<u>8</u>
HXTK	<u>12</u>	<u>6</u>	RWEJ	<u>9</u>	<u>7</u>	LHPJ	<u>14</u>	<u>6</u>	VWUJ	<u>4</u>	<u>8</u>
QYDJ	<u>21</u>	<u>5,5</u>	CJRV	<u>2</u>	<u>8,5</u>	CGLB	<u>19</u>	<u>6,5</u>	DMII	<u>3</u>	<u>8,5</u>
quest nº	<u>11</u>		seq nº	<u>2</u>							
RNXR	<u>22</u>	<u>2,5</u>	RRPB	<u>20</u>	<u>4</u>	OHTS	<u>15</u>	<u>6</u>	BVMA	<u>21</u>	<u>3,5</u>
BFQL	<u>17</u>	<u>5</u>	NTUO	<u>7</u>	<u>7,5</u>	RVGK	<u>11</u>	<u>7</u>	CBNB	<u>8</u>	<u>7,5</u>
DCRP	<u>18</u>	<u>5</u>	VEAX	<u>3</u>	<u>8</u>	FEWQ	<u>16</u>	<u>6</u>	HNGJ	<u>9</u>	<u>7</u>
MITY	<u>14</u>	<u>6</u>	IRVN	<u>23</u>	<u>2,25</u>	TNUW	<u>24</u>	<u>2</u>	LUQV	<u>4</u>	<u>8</u>
HXTK	<u>10</u>	<u>7</u>	RWEJ	<u>6</u>	<u>7,5</u>	LHPJ	<u>12</u>	<u>6</u>	VWUJ	<u>2</u>	<u>8,5</u>
QYDJ	<u>19</u>	<u>4</u>	CJRV	<u>1</u>	<u>9</u>	CGLB	<u>13</u>	<u>6,5</u>	DMII	<u>5</u>	<u>8</u>
quest nº	<u>12</u>		seq nº	<u>4</u>							
RNXR	<u>22</u>	<u>2</u>	RRPB	<u>19</u>	<u>4</u>	OHTS	<u>15</u>	<u>7</u>	BVMA	<u>20</u>	<u>3</u>
BFQL	<u>12</u>	<u>7,5</u>	NTUO	<u>11</u>	<u>8,5</u>	RVGK	<u>4</u>	<u>10</u>	CBNB	<u>1</u>	<u>10</u>
DCRP	<u>18</u>	<u>5</u>	VEAX	<u>10</u>	<u>8</u>	FEWQ	<u>13</u>	<u>7,5</u>	HNGJ	<u>6</u>	<u>9</u>
MITY	<u>14</u>	<u>7</u>	IRVN	<u>23</u>	<u>1,5</u>	TNUW	<u>24</u>	<u>1</u>	LUQV	<u>2</u>	<u>10</u>
HXTK	<u>3</u>	<u>7</u>	RWEJ	<u>9</u>	<u>8,5</u>	LHPJ	<u>16</u>	<u>5,5</u>	VWUJ	<u>5</u>	<u>9,5</u>
QYDJ	<u>21</u>	<u>3</u>	CJRV	<u>7</u>	<u>9,5</u>	CGLB	<u>17</u>	<u>5,5</u>	DMII	<u>8</u>	<u>9</u>

quest nº	<u>13</u>		seq nº	<u>1</u>							
RNXR	<u>22</u>	<u>3</u>	RRPB	<u>21</u>	<u>4</u>	OHTS	<u>15</u>	<u>5</u>	BVMA	<u>20</u>	<u>3,5</u>
BFQL	<u>4</u>	<u>6</u>	NTUO	<u>10</u>	<u>5</u>	RVGK	<u>1</u>	<u>8</u>	CBNB	<u>3</u>	<u>7</u>
DCRP	<u>18</u>	<u>5</u>	VEAX	<u>12</u>	<u>5</u>	FEWQ	<u>9</u>	<u>5,5</u>	HNGJ	<u>8</u>	<u>6</u>
MITY	<u>13</u>	<u>5</u>	IRVN	<u>23</u>	<u>3</u>	TNUW	<u>24</u>	<u>2</u>	LUQV	<u>5</u>	<u>7</u>
HXTK	<u>16</u>	<u>5</u>	RWEJ	<u>7</u>	<u>6</u>	LHPJ	<u>14</u>	<u>4,5</u>	VWUJ	<u>6</u>	<u>6,5</u>
QYDJ	<u>19</u>	<u>4,5</u>	CJRV	<u>11</u>	<u>5</u>	CGLB	<u>17</u>	<u>5</u>	DMII	<u>2</u>	<u>7,5</u>
quest nº	<u>14</u>		seq nº	<u>4</u>							
RNXR	<u>22</u>	<u>3,5</u>	RRPB	<u>19</u>	<u>4,5</u>	OHTS	<u>8</u>	<u>6,5</u>	BVMA	<u>20</u>	<u>3,8</u>
BFQL	<u>1</u>	<u>6,5</u>	NTUO	<u>17</u>	<u>6</u>	RVGK	<u>3</u>	<u>6,5</u>	CBNB	<u>9</u>	<u>6,5</u>
DCRP	<u>18</u>	<u>4,5</u>	VEAX	<u>7</u>	<u>6,5</u>	FEWQ	<u>16</u>	<u>6,5</u>	HNGJ	<u>11</u>	<u>6,5</u>
MITY	<u>12</u>	<u>6</u>	IRVN	<u>23</u>	<u>3</u>	TNUW	<u>24</u>	<u>3</u>	LUQV	<u>13</u>	<u>6,5</u>
HXTK	<u>2</u>	<u>6</u>	RWEJ	<u>14</u>	<u>6,5</u>	LHPJ	<u>10</u>	<u>5,5</u>	VWUJ	<u>5</u>	<u>6,5</u>
QYDJ	<u>21</u>	<u>4</u>	CJRV	<u>4</u>	<u>6,5</u>	CGLB	<u>15</u>	<u>6</u>	DMII	<u>6</u>	<u>7</u>
quest nº	<u>15</u>		seq nº	<u>1</u>							
RNXR	<u>22</u>	<u>4</u>	RRPB	<u>17</u>	<u>4,5</u>	OHTS	<u>6</u>	<u>6,5</u>	BVMA	<u>20</u>	<u>4,5</u>
BFQL	<u>1</u>	<u>7</u>	NTUO	<u>12</u>	<u>6</u>	RVGK	<u>2</u>	<u>6,5</u>	CBNB	<u>11</u>	<u>6,5</u>
DCRP	<u>18</u>	<u>5</u>	VEAX	<u>14</u>	<u>5,5</u>	FEWQ	<u>15</u>	<u>6</u>	HNGJ	<u>10</u>	<u>6,5</u>
MITY	<u>13</u>	<u>6</u>	IRVN	<u>21</u>	<u>4</u>	TNUW	<u>24</u>	<u>3</u>	LUQV	<u>3</u>	<u>6,5</u>
HXTK	<u>9</u>	<u>6,5</u>	RWEJ	<u>5</u>	<u>7</u>	LHPJ	<u>19</u>	<u>4</u>	VWUJ	<u>7</u>	<u>6,5</u>
QYDJ	<u>23</u>	<u>3</u>	CJRV	<u>16</u>	<u>6</u>	CGLB	<u>8</u>	<u>6</u>	DMII	<u>4</u>	<u>7</u>
quest nº	<u>16</u>		seq nº	<u>6</u>							
RNXR	<u>22</u>	<u>4,5</u>	RRPB	<u>16</u>	<u>5</u>	OHTS	<u>18</u>	<u>4,5</u>	BVMA	<u>21</u>	<u>4,5</u>
BFQL	<u>3</u>	<u>7,5</u>	NTUO	<u>6</u>	<u>7,5</u>	RVGK	<u>1</u>	<u>8</u>	CBNB	<u>5</u>	<u>7</u>
DCRP	<u>17</u>	<u>5</u>	VEAX	<u>4</u>	<u>7</u>	FEWQ	<u>8</u>	<u>7</u>	HNGJ	<u>13</u>	<u>6</u>
MITY	<u>15</u>	<u>5,5</u>	IRVN	<u>23</u>	<u>4</u>	TNUW	<u>24</u>	<u>3</u>	LUQV	<u>10</u>	<u>6,5</u>
HXTK	<u>11</u>	<u>6,5</u>	RWEJ	<u>2</u>	<u>7,5</u>	LHPJ	<u>19</u>	<u>4,5</u>	VWUJ	<u>9</u>	<u>7</u>
QYDJ	<u>20</u>	<u>4</u>	CJRV	<u>7</u>	<u>7</u>	CGLB	<u>14</u>	<u>6</u>	DMII	<u>12</u>	<u>6,5</u>

quest nº		17		seq nº		8									
RNXR	24	7	RRPB	18	9	OHTS	13	7	BVMA	20	7				
BFQL	7	7	NTUO	3	10	RVGK	1	9	CBNB	17	10				
DCRP	9	9	VEAX	6	8	FEWQ	5	8	HNGJ	19	9				
MITY	4	9	IRVN	23	8	TNUW	21	6	LUQV	10	8				
HXTK	8	10	RWEJ	12	7	LHPJ	16	9	VWUJ	2	10				
QYDJ	22	5	CJRV	14	8	CGLB	11	8	DMII	15	8				

quest nº		18		seq nº		1									
RNXR	22	3	RRPB	19	4	OHTS	16	6	BVMA	20	4				
BFQL	6	6	NTUO	2	6	RVGK	5	6	CBNB	10	7				
DCRP	18	5	VEAX	13	7	FEWQ	14	7	HNGJ	4	6				
MITY	17	6	IRVN	23	3	TNUW	24	2	LUQV	3	6				
HXTK	8	6	RWEJ	11	6	LHPJ	7	5	VWUJ	12	7				
QYDJ	21	5	CJRV	9	7	CGLB	15	6	DMII	1	6				

quest nº		19		seq nº		3									
RNXR	23	2	RRPB	20	4	OHTS	15	4	BVMA	21	3				
BFQL	13	5	NTUO	7	7	RVGK	5	8	CBNB	12	7				
DCRP	19	4	VEAX	17	5	FEWQ	14	5	HNGJ	6	8				
MITY	4	7	IRVN	22	2	TNUW	24	2	LUQV	2	8				
HXTK	10	5	RWEJ	8	7	LHPJ	11	5	VWUJ	3	8				
QYDJ	18	4	CJRV	1	8	CGLB	16	5	DMII	9	7				

quest nº		20		seq nº		4									
RNXR	22	4	RRPB	19	4,8	OHTS	14	5,2	BVMA	21	4,3				
BFQL	10	5,5	NTUO	9	6	RVGK	3	7,7	CBNB	1	8,5				
DCRP	20	5	VEAX	12	6	FEWQ	8	6,5	HNGJ	11	6,2				
MITY	16	5,2	IRVN	23	4,1	TNUW	24	4	LUQV	5	7				
HXTK	15	5,3	RWEJ	7	6,8	LHPJ	13	5,2	VWUJ	2	9				
QYDJ	18	5,2	CJRV	4	8	CGLB	17	5,3	DMII	6	7				

quest nº	<u>21</u>		seq nº	<u>5</u>							
RNXR	<u>22</u>	<u>1</u>	RRPB	<u>21</u>	<u>2,5</u>	OHTS	<u>12</u>	<u>5,5</u>	BVMA	<u>19</u>	<u>2</u>
BFQL	<u>4</u>	<u>7,5</u>	NTUO	<u>14</u>	<u>4</u>	RVGK	<u>11</u>	<u>6</u>	CBNB	<u>7</u>	<u>7,5</u>
DCRP	<u>20</u>	<u>2</u>	VEAX	<u>3</u>	<u>8</u>	FEWQ	<u>5</u>	<u>7</u>	HNGJ	<u>9</u>	<u>6</u>
MITY	<u>16</u>	<u>4</u>	IRVN	<u>23</u>	<u>1</u>	TNUW	<u>24</u>	<u>0,5</u>	LUQV	<u>6</u>	<u>7</u>
HXTK	<u>13</u>	<u>5</u>	RWEJ	<u>10</u>	<u>6</u>	LHPJ	<u>15</u>	<u>4,5</u>	VWUJ	<u>1</u>	<u>9</u>
QYDJ	<u>18</u>	<u>2,5</u>	CJRV	<u>2</u>	<u>8,5</u>	CGLB	<u>17</u>	<u>3,5</u>	DMII	<u>8</u>	<u>7</u>
quest nº	<u>22</u>		seq nº	<u>5</u>							
RNXR	<u>22</u>	<u>5</u>	RRPB	<u>16</u>	<u>6</u>	OHTS	<u>13</u>	<u>6,5</u>	BVMA	<u>20</u>	<u>5</u>
BFQL	<u>12</u>	<u>6,5</u>	NTUO	<u>19</u>	<u>8</u>	RVGK	<u>9</u>	<u>9</u>	CBNB	<u>7</u>	<u>8</u>
DCRP	<u>15</u>	<u>6</u>	VEAX	<u>6</u>	<u>8</u>	FEWQ	<u>4</u>	<u>8</u>	HNGJ	<u>10</u>	<u>7,5</u>
MITY	<u>18</u>	<u>8</u>	IRVN	<u>23</u>	<u>5</u>	TNUW	<u>24</u>	<u>5</u>	LUQV	<u>3</u>	<u>8</u>
HXTK	<u>11</u>	<u>7</u>	RWEJ	<u>5</u>	<u>8</u>	LHPJ	<u>14</u>	<u>6,5</u>	VWUJ	<u>2</u>	<u>9</u>
QYDJ	<u>21</u>	<u>5,5</u>	CJRV	<u>1</u>	<u>9</u>	CGLB	<u>17</u>	<u>7</u>	DMII	<u>8</u>	<u>9</u>
quest nº	<u>23</u>		seq nº	<u>2</u>							
RNXR	<u>22</u>	<u>3,5</u>	RRPB	<u>19</u>	<u>4,5</u>	OHTS	<u>15</u>	<u>6</u>	BVMA	<u>21</u>	<u>3,5</u>
BFQL	<u>11</u>	<u>7</u>	NTUO	<u>8</u>	<u>6,5</u>	RVGK	<u>10</u>	<u>7</u>	CBNB	<u>4</u>	<u>7,5</u>
DCRP	<u>18</u>	<u>4,5</u>	VEAX	<u>12</u>	<u>6,5</u>	FEWQ	<u>13</u>	<u>6,5</u>	HNGJ	<u>5</u>	<u>7</u>
MITY	<u>17</u>	<u>5</u>	IRVN	<u>23</u>	<u>3</u>	TNUW	<u>24</u>	<u>2</u>	LUQV	<u>2</u>	<u>8</u>
HXTK	<u>6</u>	<u>7</u>	RWEJ	<u>14</u>	<u>6</u>	LHPJ	<u>7</u>	<u>6,5</u>	VWUJ	<u>1</u>	<u>8</u>
QYDJ	<u>20</u>	<u>4</u>	CJRV	<u>9</u>	<u>7</u>	CGLB	<u>16</u>	<u>5,5</u>	DMII	<u>3</u>	<u>8</u>
quest nº	<u>24</u>		seq nº	<u>8</u>							
RNXR	<u>23</u>	<u>3,5</u>	RRPB	<u>18</u>	<u>4,25</u>	OHTS	<u>4</u>	<u>5,6</u>	BVMA	<u>21</u>	<u>3,75</u>
BFQL	<u>12</u>	<u>4,75</u>	NTUO	<u>8</u>	<u>5</u>	RVGK	<u>6</u>	<u>5,25</u>	CBNB	<u>1</u>	<u>5,7</u>
DCRP	<u>19</u>	<u>4,25</u>	VEAX	<u>10</u>	<u>4,65</u>	FEWQ	<u>9</u>	<u>4,85</u>	HNGJ	<u>7</u>	<u>5,5</u>
MITY	<u>5</u>	<u>5,2</u>	IRVN	<u>22</u>	<u>3,55</u>	TNUW	<u>24</u>	<u>2</u>	LUQV	<u>14</u>	<u>4,33</u>
HXTK	<u>15</u>	<u>4,29</u>	RWEJ	<u>13</u>	<u>4,5</u>	LHPJ	<u>17</u>	<u>4,26</u>	VWUJ	<u>11</u>	<u>4,75</u>
QYDJ	<u>20</u>	<u>4</u>	CJRV	<u>3</u>	<u>5,65</u>	CGLB	<u>16</u>	<u>4,5</u>	DMII	<u>2</u>	<u>5,63</u>