

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e Computação
Departamento de Comunicações

Compressão Progressiva de Imagens Usando DCT e Código de Golomb

Autor:

Fabício Corrêa de Araújo Oliveira

Orientador:

Prof. Dr. Max Henrique Machado Costa

Dissertação apresentada à Faculdade de Engenharia Elétrica e Computação da Unicamp como parte dos requisitos exigidos para a obtenção do título de MESTRE EM ENGENHARIA ELÉTRICA.

Campinas, Setembro de 2002

Compressão Progressiva de Imagens Usando DCT e Código de Golomb

Autor:

Fabício Corrêa de Araújo Oliveira

Orientador:

Prof. Dr. Max Henrique Machado Costa

Dissertação apresentada à Faculdade de Engenharia Elétrica e Computação da Unicamp como parte dos requisitos exigidos para a obtenção do título de MESTRE EM ENGENHARIA ELÉTRICA.

Banca Examinadora:

Prof. Dr. Max Henrique Machado Costa	FEEC/Unicamp
Prof. Dr. Rui Seara	UFSC
Prof. Dr. Dalton Soares Arantes	FEEC/Unicamp
Prof. Dr. Amauri Lopes	FEEC/Unicamp

Campinas, Setembro de 2002

Resumo

Nesta dissertação propõe-se um novo método de compressão de imagens estáticas que faz uso da transformada discreta do co-seno (DCT) e do código de Golomb para variáveis aleatórias com distribuição de probabilidade geométrica. Os coeficientes da transformada são codificados de maneira progressiva, plano de bit por plano de bit, de modo que a seqüência comprimida resultante permite que a decodificação seja gradual, isto é, a qualidade da imagem reconstruída depende apenas do ponto em que a decodificação é interrompida. Além disso, o método proposto apresenta como vantagens a simplicidade de implementação e a possibilidade de sua utilização, em conjunto com técnicas de "compensação de movimento", em sistemas de compressão de seqüências de vídeo. Neste trabalho são também apresentados resultados experimentais, em termos da razão de máximo sinal por ruído em função da taxa de bits, obtidos pelo sistema de compressão proposto e por outros sistemas.

Abstract

This dissertation proposes a new method for compression of still image pictures, which is based on the discrete cosine transform (DCT) and on Golomb's entropy code for random variables with geometric probability distributions. The DCT transform coefficients are encoded progressively, in a bit-plane by bit-plane fashion, so that the compressed output sequence can be decoded gradually. Therefore, the quality of the reconstructed images depends only on the point at which the decoding process is interrupted. Besides, the proposed method has such advantages as simplicity and the possibility to be used, in association with motion compensation, in video compression systems. This work also presents experimental results, in terms of peak signal to noise ratio (PSNR) versus bit-rate, obtained by the proposed system and by other systems.

Aos meus pais, Eli e Fátima.

Agradecimentos

Gostaria de agradecer a todas as pessoas que, direta ou indiretamente, contribuíram para a realização deste trabalho.

Agradeço especialmente aos meus pais, Eli Alves de Oliveira e Maria de Fátima Corrêa de Araújo Oliveira, sem o apoio de quem não teria conseguido ultrapassar todos os obstáculos que a vida nos impõe.

Também especialmente agradeço ao meu orientador, Max Henrique Machado Costa, que além do seu apoio profissional, competente e aplicado, também me ofereceu a sua amizade, atenciosa e compreensiva.

Agradeço a Luciana pelo seu carinho, afeto, compreensão e apoio.

Agradeço a André, Carlos Henrique, Daniel, Dimitri, Divanilson, Felipão, Gustavo, Joselan, Márcia Makino, Márcio, Marcus, Sílvia e Vítor, que representaram a minha família durante todo este período, pela amizade e pelo apoio.

E também agradeço à toda a comunidade da FEEC e da Unicamp.

Fabício Corrêa de Araújo Oliveira
Campinas, Setembro de 2002.

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro.

Sumário

Agradecimentos	ii
Lista de Figuras	v
Lista de Algoritmos	vii
1 Introdução	1
2 Conceitos Fundamentais	3
2.1 Imagem Digital	3
2.2 Compressão <i>versus</i> Compactação	5
2.3 Medidas de Compressão	7
2.4 Medidas de Distorção	8
2.5 Redundância e Entropia	8
2.6 Irrelevância da Distorção	11
2.7 Quantização	12
2.8 Quantização e Codificação Progressivas	14
2.9 Codificação de Entropia	15
2.9.1 Compressores Especializados	16
2.10 Sistemas de Compressão de Imagens	17
3 Codificação por Transformada	19
3.1 Modelos Estatísticos	19
3.1.1 Estacionariedade e Separabilidade	20
3.1.2 Densidade Espectral de Potência	21
3.1.3 Ruídos Brancos e Coloridos	22
3.1.4 Modelos Auto-regressivos	23
3.1.5 Matrizes de Autocorrelação, Autovalores e Autovetores	24
3.1.6 Modelo Estatístico de Imagens	26

3.2	Transformadas	28
3.2.1	Transformadas de Processos Aleatórios	31
3.2.2	Transformada de Karhunen-Loève	32
3.2.3	Transformada Discreta do Co-seno	34
3.3	Codificação dos Coeficientes da Transformada	37
3.3.1	Alocação Ótima de Bits	38
3.3.2	Ganho da Codificação por Transformada	40
3.3.3	Transformada Ótima (KLT)	41
4	Método Proposto	44
4.1	Transformada Discreta do Co-seno	45
4.2	Código de Corrida de Zeros de Golomb	46
4.2.1	Método de Codificação	46
4.2.2	Adaptatividade	49
4.3	Codificação dos Coeficientes da Transformada	52
4.3.1	Propagação de Erros	55
4.4	Reorganização dos Coeficientes da Transformada	56
5	Resultados Experimentais	59
5.1	Metodologia	59
5.2	Resultados	60
6	Conclusão	67
6.1	Sugestões para Trabalhos Futuros	69

Lista de Figuras

2.1	Diagrama de uma imagem digital.	4
2.2	Exemplo de imagem natural: Lena (512×512 pixels)	6
2.3	Processo de quantização.	13
2.4	Exemplo de quantização progressiva.	15
2.5	Diagrama de um sistema de compressão/descompressão digital de imagens.	17
3.1	Ruído colorido $\{v[n]\}$ gerado por um sistema linear de resposta ao impulso $\{h[n]\}$	22
3.2	Modelo auto-regressivo.	23
3.3	Densidade espectral de potência normalizada ($\sigma_x^2 = 1$) de processo auto-regressivo de primeira ordem.	27
3.4	Autovalores de processos AR de primeira ordem com diferentes valores de ρ	28
3.5	Funções-base de comprimento 8 da DCT-1D (a) e da KLT-1D de um processo AR com $\rho = 0,9$ (b).	35
3.6	Variâncias dos coeficientes da DCT de processos AR de primeira ordem com diferentes valores de ρ	36
3.7	Ganhos da codificação por transformada para processos AR de primeira ordem em função do coeficiente de correlação ρ	42
4.1	Diagrama do transformador DCT-2D usado pelo método proposto.	45
4.2	Parte (iii) do código de Golomb de comprimento 10.	48
4.3	Comprimentos ℓ e ℓ_R do codificador de Golomb em função do parâmetro p	50
4.4	Diferença entre o comprimento do codificador de Golomb ℓ e o valor aproximado ℓ_R em função de ℓ_R	51
4.5	Numeração dos coeficientes da DCT em um bloco.	57
5.1	Resultados experimentais para a imagem Lena (512×512)	61
5.2	Resultados experimentais para a imagem Goldhill (512×512)	62

5.3	Resultados experimentais para a imagem Bárbara (512 x 512)	62
5.4	Resultados experimentais para a imagem Boat (512 x 512)	63
5.5	Imagens originais utilizadas para obter os resultados experimentais: (a) Lena, (b) Goldhill, (c) Bárbara e (d) Boat.	64
5.6	Resultados visuais obtidos para a imagem Lena pelo método proposto: (a) Imagem original, (b) com 0,25 bpp, (c) com 0,5 bpp e (d) com 1 bpp.	65
5.7	Resultados visuais obtidos para a imagem Goldhill com o método proposto: (a) Imagem original, (b) com 0,25 bpp, (c) com 0,5 bpp e (d) com 1 bpp. . .	66

Lista de Algoritmos

4.1	Codificação de Golomb para variáveis aleatórias geométricas.	49
4.2	Codificação de Golomb adaptativa.	53
4.3	Codificação dos coeficientes da DCT.	55
4.4	Reorganização dos coeficientes da DCT	58

Capítulo 1

Introdução

O objetivo deste trabalho é a apresentação e a avaliação de um novo esquema eficiente e simples de compressão de imagens naturais estáticas. O esquema proposto é baseado na transformada discreta do co-seno (DCT) [2] e utiliza o código de corrida de zeros de Golomb [3] associado a um método especial de codificação dos coeficientes da transformada. O codificador efetua uma compressão progressiva da imagem, processando cada plano de bits dos coeficientes da transformada separadamente a partir do mais significativo. O resultado do esquema de codificação é uma seqüência comprimida “embutida”, isto é, na qual as seqüências de menores taxas estão presentes no início das seqüências de maiores taxas. Ademais, o processo de decodificação é gradual, ou seja, a imagem reconstruída é refinada gradativamente à medida que o decodificador processa a informação comprimida.

A DCT usada pelo esquema proposto é idêntica à utilizada pelos padrões de compressão JPEG [4], para imagens estáticas, e MPEG [5], para seqüências de vídeo. Por ser uma transformada de bloco, isto é, que é aplicada a blocos de amostras do sinal que se deseja comprimir, a DCT é a transformada predominante nos esquemas de compressão de vídeo. Esses esquemas usualmente fazem uso de estimação e compensação de movimento para eliminar a redundância temporal. Como exemplo, citam-se os padrões MPEG [6, 7, 8]. Portanto, o esquema proposto pode ser facilmente adaptado para ser utilizado em compressão de vídeo, em conjunto com estimação e compensação de movimento.

A utilização da DCT também apresenta a vantagem de permitir a utilização de um dos diversos métodos de cálculo rápido da transformada conhecidos atualmente [9]. Além disso, alguns desses métodos já se encontram implementados em *hardware*, em processadores digitais de sinais (DSP's) disponíveis no mercado. Esse fato permite que a implementação de um sistema de compressão de imagens estáticas ou de vídeo baseado no esquema aqui proposto seja realizada em um tempo muito curto e com baixo custo.

A informação de significância dos coeficientes da DCT é codificada utilizando o código de corrida de zeros de Golomb. Os parâmetros do código são adaptados à medida que a informação é codificada e o esquema de adaptação faz uso da estatística da fonte, que é obtida simultaneamente com o processo de codificação. Além disso, o algoritmo de adaptação do codificador se baseia apenas na informação que já foi codificada e transmitida, de modo que é possível ajustar os parâmetros do decodificador sem a necessidade de se transmitir qualquer informação extra.

Esta dissertação também apresenta resultados objetivos da medida da eficiência do esquema proposto e de outros esquemas de compressão de imagens [10, 11, 4], comparados com respeito à razão de máximo sinal por ruído (PSNR – *Peak Signal-to-Noise Ratio*) em função da taxa de compressão utilizada.

Portanto, este trabalho apresenta as seguintes contribuições:

- Propõe-se um novo esquema simples e eficiente de compressão de imagens estáticas que reúne a DCT, a reordenação do sinal em planos de bits e a codificação de Golomb. O método é fundamentalmente novo na forma pela qual os coeficientes da transformada são codificados.
- Propõe-se um método simples de codificação dos coeficientes da transformada, que realiza codificação progressiva com base nos planos de bits, usando o código de corrida de zeros de Golomb.
- Propõe-se um método de adaptação dos parâmetros do código de Golomb que se baseia apenas em informação disponível tanto no codificador quanto no decodificador, sem a necessidade de transmissão de informação extra.
- Propõem-se métodos simples, sem o uso de tabelas, para construção e decodificação das palavras do código de Golomb.

Organização do trabalho

O Capítulo 2 apresenta conceitos fundamentais relativos aos sistemas de compressão digital de imagens. O Capítulo 3 apresenta uma exposição teórica a respeito do uso da transformada na compressão de imagens. O Capítulo 4 introduz o esquema de compressão proposto, e o Capítulo 5 apresenta os resultados do esquema proposto e os compara aos resultados de outros esquemas de compressão. Finalmente, o Capítulo 6 apresenta as conclusões do trabalho e sugestões para trabalhos futuros.

Capítulo 2

Conceitos Fundamentais

Este capítulo trata dos conceitos fundamentais relacionados à *codificação digital de imagens*, tema deste trabalho de dissertação. Dentre os conceitos apresentados estão os de imagem digital (Seção 2.1); de *compressão* e *compactação* (Seção 2.2); de medidas de compressão (Seção 2.3) e distorção (Seção 2.4), usadas para avaliar o desempenho dos sistemas de compactação e compressão; de redundância e entropia (Seção 2.5) e de distorção irrelevante (Seção 2.6). Além disso, são também discutidos os componentes fundamentais de um sistema de compressão, tais como: a quantização (Seções 2.7 e 2.8) e a codificação de entropia (Seção 2.9). Na Seção 2.10 é apresentado um modelo geral de sistemas de compressão de imagens.

2.1 Imagem Digital

Definimos *imagem digital* como sendo uma seqüência bidimensional de amostras,

$$\mathbf{x} = \{x[n_1, n_2]; \quad 0 \leq n_1 < N_1, \quad 0 \leq n_2 < N_2\}, \quad (2.1)$$

com limites finitos, N_1 e N_2 , nas direções vertical e horizontal, respectivamente. O termo **pixel** (*picture element*) será o termo mais utilizado daqui por diante para indicar uma amostra da imagem. A primeira coordenada, n_1 , representa o índice da linha, enquanto a segunda coordenada, n_2 , representa o índice da coluna, conforme ilustrado pela Fig. 2.1.

O valor de um *pixel*, $x[n_1, n_2]$, representa a intensidade luminosa ou o brilho da imagem na posição $[n_1, n_2]$. Imagens coloridas geralmente possuem *pixels* com três valores, que correspondem aos componentes da luz colorida que são percebidas pelo sistema visual humano. Portanto, uma imagem colorida pode também ser vista como sendo formada por três seqüências bidimensionais, $x_R[n_1, n_2]$, $x_G[n_1, n_2]$ e $x_B[n_1, n_2]$, que representam os componen-

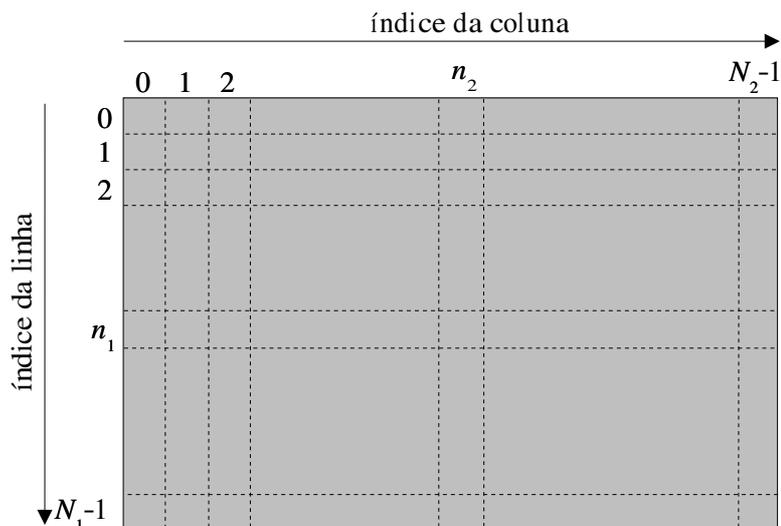


Fig. 2.1: Diagrama de uma imagem digital.

tes de cores primárias do sistema aditivo de cores: vermelha, verde e azul, respectivamente. As imagens formadas por mais de um componente são geralmente denominadas imagens *multi-espectrais*, nas quais cada componente corresponde à intensidade luminosa em uma determinada faixa do espectro, visível ou não. Em geral, uma imagem multi-espectral pode ser formada por um conjunto arbitrário de componentes,

$$x_c[n_1, n_2], \quad c = 1, 2, \dots, C. \quad (2.2)$$

Uma imagem preparada para impressão colorida é um exemplo de imagem composta por mais de três componentes, que correspondem às cores das tintas usadas na impressão. Tipicamente, esse tipo de imagem possui quatro componentes, correspondentes às cores ciana, magenta, amarela e preta, mas algumas imagens podem possuir até seis componentes, incluindo verde e violeta. Imagens *hiperespectrais* [12], geralmente obtidas por satélites, podem conter centenas de componentes.

Neste trabalho, nos dedicamos exclusivamente às imagens com apenas um componente, frisando que é sempre possível aplicar o método de codificação proposto a cada componente separadamente. No entanto, a dependência que existe entre os componentes que formam uma imagem pode ser aproveitada para projetar sistemas de codificação mais eficientes.

Os valores dos *pixels* são geralmente representados por inteiros de B bits, com ou sem sinal¹. No entanto, para efeito de análise dos métodos de compressão, esses valores são

¹Sinal positivo ou negativo.

muitas vezes considerados como sendo números reais. Quando são inteiros com sinal², os *pixels* têm valores entre -2^{B-1} e $2^{B-1} - 1$, inclusive, e quando são inteiros sem sinal, eles têm valores entre 0 e $2^B - 1$. É sempre possível transformar a representação sem sinal em uma representação com sinal, ou vice-versa, subtraindo-se ou somando-se 2^{B-1} a cada *pixel*.

As imagens digitais mais comuns, como as encontradas na internet e em aplicações computacionais voltadas ao consumidor, possuem *pixels* com componentes cujos valores são inteiros de 8 bits sem sinal, que variam entre 0 e 255. Em geral, antes de se efetuar a compressão, esses *pixels* são transformados em valores com sinal de modo a se obter um valor médio próximo a zero. Representações com maior número de bits são freqüentemente encontradas em aplicações médicas, científicas e militares.

As características da imagem que podem ser exploradas pelo sistema de compressão dependem fundamentalmente da natureza do conteúdo da imagem. Podemos distinguir quatro classes de imagens com base em seu conteúdo:

Imagens naturais representam cenas naturais, incluindo imagens fotográficas (*cf.* Fig. 2.2).

Imagens de texto representando textos escaneados ou gerados por computador, como por exemplo, fac-símile.

Gráficos escaneados ou gerados por computador, como desenhos em geral.

Imagens compostas que são tipicamente formadas por composições dos três tipos anteriores.

Nesta dissertação abordamos o problema da codificação de imagens naturais. Embora o sistema proposto também possa ser usado para codificar outros tipos de imagem, o seu desempenho será consideravelmente inferior ao de sistemas especificamente projetados para tais fins.

2.2 Compressão *versus* Compactação

O termo **codificação** utilizado nesta dissertação expressa a ação pela qual se traduz uma imagem, para uma representação formada por uma seqüência de dígitos binários (bits), através de um método determinado. Um exemplo de codificação seria representar a imagem colocando em seqüência os B bits de cada amostra, amostra por amostra e linha por linha. Essa é uma das maneiras mais simples de se representar uma imagem em forma binária.

²Neste caso, o valor do *pixel* representa geralmente a diferença entre a intensidade do *pixel* e o valor médio da mesma.



Fig. 2.2: Exemplo de imagem natural: Lena (512×512 pixels)

Ela é denominada codificação por **varredura**, em vista do sistema de varredura da grande maioria dos monitores de vídeo, e utiliza $N_1 N_2 B$ bits para produzir uma representação exata da imagem digital original. É comum não se referir a esta representação como um tipo de codificação, para não se confundir com os métodos de codificação mais complexos. Ela tem a vantagem de ser de fácil utilização, sem envolver o uso de um complexo sistema de codificação e decodificação.

O tipo de codificação em que estamos interessados é o que tem por objetivo utilizar o menor número de bits possível para representar a imagem, podendo ou não a imagem reconstruída ser, até certo ponto, diferente da imagem original. A esse tipo de codificação denominamos **compressão**. Denominamos **compactação** à compressão sem perdas, ou seja, àquela em que a imagem reconstruída deve ser idêntica à original. O termo compressão geralmente indica a **compressão com perdas**, a menos que seja explicitado o contrário. Na compressão com perdas, alguma distorção na imagem reconstruída pelo decodificador em relação a imagem original é permitida. A compressão com perdas pode ser aceitável, em grande parte dos casos, pois perdas de determinados componentes do sinal podem ser toleradas pelo sistema visual humano sem deterioração notável da qualidade da imagem. Além disso, em muitos casos, a imagem digital, que será tratada pelo sistema de codificação, já é uma representação imperfeita da cena real, visto que as amostras são versões quantizadas (discretas) de valores reais contínuos. O motivo principal para se preferir a compressão no

lugar da compactação é que, em geral, essa última é incapaz de atingir o grau de compressão exigido por muitos sistemas de armazenamento e comunicação.

Por outro lado, em alguns casos não é possível abrir mão de uma reconstrução perfeita, como no caso de imagens de aplicações médicas ou quando os valores dos *pixels* não representam diretamente valores de intensidade luminosa. Como exemplo desse último caso, têm-se as imagens que utilizam *palheta* de cor, nas quais os valores dos *pixels* são os índices das cores com as quais eles devem ser reproduzidos. Para esse tipo de imagem, uma pequena diferença entre a imagem original e a reconstruída pode representar uma diferença inaceitável no resultado visualmente percebido.

O método de codificação proposto nesta dissertação é um método de **compressão com perdas** e, portanto não se aplica aos casos que requerem apenas compactação.

2.3 Medidas de Compressão

O propósito de um sistema de compressão é representar uma imagem através de uma seqüência de bits, denotada por \mathbf{c} , com o menor comprimento possível. O comprimento da seqüência binária \mathbf{c} é representado por $\|\mathbf{c}\|$. Sem a utilização de qualquer compressão, como, por exemplo, na representação em *varredura*, o número de bits necessários para representar uma imagem de dimensões N_1 e N_2 é N_1N_2B . Definimos, então, uma medida da quantidade de compressão, denominada *razão de compressão*, como

$$\text{razão de compressão} \triangleq \frac{\# \text{ de bits da imagem original}}{\# \text{ de bits da imagem comprimida}} \quad (2.3)$$

$$= \frac{N_1N_2B}{\|\mathbf{c}\|}. \quad (2.4)$$

Também definimos a *taxa* de um compressor como sendo o número de bits utilizados na representação comprimida dividido pelo número total de *pixels* da imagem, isto é,

$$\text{taxa} \triangleq \frac{\|\mathbf{c}\|}{N_1N_2}, \quad (2.5)$$

que é expressa em bits por *pixel* (bpp).

O desempenho dos métodos de compactação são, em geral, medidos pela razão de compressão, pois, como as reproduções são idênticas, o que está em questão é a capacidade de codificar a imagem usando um número de bits menor do que o número de bits da imagem original. Já no caso dos sistemas de compressão, o número de bits da representação original

não é um fator tão relevante, e a medida mais utilizada é a taxa de compressão, que denota o número médio de bits necessários para representar cada *pixel* da imagem codificada, e não depende do número de bits de cada *pixel* da imagem original.

2.4 Medidas de Distorção

Ao permitir que a reconstrução não seja perfeita e que pequenos erros sejam introduzidos pelo processo de compressão e descompressão, podemos obter taxas de compressão menores do que no caso contrário. E quanto mais distorção for permitida, maiores são os níveis de compressão que podem ser obtidos. Portanto, é necessário haver uma maneira de medir a distorção provocada por um sistema, a fim de que se possa estabelecer limites objetivos para esse valor, ou para que se possa medir e comparar o desempenho de diferentes sistemas.

Formalmente, a distorção entre a imagem original $\mathbf{x} = \{x[n_1, n_2]\}$ e a sua versão reconstruída $\hat{\mathbf{x}}$ é representada por $D(\mathbf{x}, \hat{\mathbf{x}})$. A medida de distorção mais comumente utilizada é o **erro quadrático médio** ou MSE (*Mean Squared Error*), definido por

$$D_{\text{MSE}}(\mathbf{x}, \hat{\mathbf{x}}) \triangleq \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (x[n_1, n_2] - \hat{x}[n_1, n_2])^2. \quad (2.6)$$

Na literatura da área, o erro quadrático médio é geralmente representado pela sua medida recíproca equivalente, a **razão de máximo sinal versus ruído** ou PSNR (*Peak Signal-to-Noise Ratio*), definida por

$$\text{PSNR} \triangleq 10 \log_{10} \frac{(2^B - 1)^2}{\text{MSE}}, \quad (2.7)$$

e expressa em dB (decibéis). Boas reconstruções têm valores de 30 dB ou superiores para valores típicos de B , como, por exemplo, $B = 8$.

A popularidade do erro quadrático médio se deve principalmente à simplicidade com que pode ser calculado, além da facilidade de tratar problemas de otimização que o utilizam como métrica.

Métricas que levam em consideração as particularidades do sistema visual humano também são utilizadas, porém em menor escala [13].

2.5 Redundância e Entropia

Em princípio uma imagem de N_1 linhas, N_2 colunas e B bits por *pixel*, necessitaria de um total de $N_1 N_2 B$ bits para ser representada. No entanto, é possível representá-la utilizando

menos bits devido basicamente a dois fatores: a redundância da informação contida nos sinais de imagem e a irrelevância de certos detalhes também presentes nestes sinais.

A redundância se apresenta basicamente de duas formas: no conhecimento *a priori* das probabilidades com que os *pixels* podem assumir diferentes valores, e na dependência entre conjuntos de *pixels* da imagem. Se a distribuição de probabilidade dos valores dos *pixels* é mais concentrada em certos valores, ou se existe dependência entre grupos de *pixels* de uma imagem, é possível representá-la utilizando menos bits do que na representação original. Por sorte, as imagens possuem bastante redundância, principalmente no que diz respeito à dependência entre *pixels* próximos entre si.

Um *pixel* X_1 , por exemplo, que pode assumir 2^B valores distintos, necessitaria em média de B bits para ser representado, caso seja utilizado um código binário de comprimento fixo, isto é, um código que represente todos os possíveis valores do *pixel* com palavras-código de mesmo comprimento. No entanto, é possível representar o mesmo *pixel* utilizando em média entre $H(X_1)$ e $H(X_1) + 1$ bits, caso se utilize um código de comprimento variável cujas palavras sejam escolhidas adequadamente. $H(X_1)$ é função da distribuição de probabilidade de X_1 , isto é, $p_{X_1}(x_1)$, e denomina-se **entropia**. O seu valor é dado por

$$H(X_1) \triangleq E \left[\log \frac{1}{p_{X_1}(X_1)} \right] = \sum_{x_1} p_{X_1}(x_1) \log \frac{1}{p_{X_1}(x_1)}. \quad (2.8)$$

No caso descrito acima a entropia $H(X_1)$ nunca é maior do que B , o que implica que é possível ganhar eficiência na compressão através do conhecimento *a priori* da distribuição de probabilidade de X_1 e da utilização de um código de comprimento variável adequado a essa distribuição. Por outro lado, não é possível comprimir além da entropia sem que haja perdas [14]. Portanto, a partir da entropia é possível estabelecer que

$$\text{razão de compressão} \leq \frac{B}{H(X_1)}, \quad (2.9)$$

considerando que um único *pixel* é codificado por vez, e que nenhuma informação sobre outros *pixels* é utilizada.

É possível, no entanto, obter razões de compressão ainda maiores que o limite determinado por (2.9), se a redundância devido à dependência entre os *pixels* também for levada em conta. Esse é um detalhe fundamental nos sistemas de compressão de imagens, já que este tipo de redundância é o mais preponderante.

O número médio de bits necessário para codificar dois *pixels*, X_1 e X_2 , é limitado pela

entropia conjunta dos *pixels*,

$$H(X_1, X_2) = H(X_1) + H(X_2|X_1),$$

na qual $H(X_2|X_1)$ denota a **entropia condicional** de X_2 dado X_1 , que é a média das entropias calculadas a partir das distribuições de probabilidade condicionais $p_{X_2|X_1}(x_2|x_1)$ para cada valor de x_1 ponderada pela distribuição de probabilidade $p_{X_1}(x_1)$, isto é,

$$H(X_2|X_1) \triangleq \sum_{x_1} p_{X_1}(x_1) H(X_2|X_1 = x_1) \quad (2.10)$$

$$= \sum_{x_1} p_{X_1}(x_1) \sum_{x_2} p_{X_2|X_1}(x_2|x_1) \log \frac{1}{p_{X_2|X_1}(x_2|x_1)} \quad (2.11)$$

$$= \sum_{x_1, x_2} p_{X_2, X_1}(x_2, x_1) \log \frac{1}{p_{X_2|X_1}(x_2|x_1)}. \quad (2.12)$$

Se X_1 e X_2 são independentes, $H(X_2|X_1) = H(X_2)$, caso contrário, $H(X_2|X_1) < H(X_2)$. Isso implica que, se existe dependência entre dois *pixels*, é melhor codificá-los conjuntamente ao invés de separadamente, conclusão intuitiva.

O grau de dependência entre dois *pixels* pode ser medido pela função $I(X_1; X_2)$, denominada **informação mútua** e definida por

$$I(X_1; X_2) \triangleq H(X_2) - H(X_2|X_1) = H(X_1) - H(X_1|X_2). \quad (2.13)$$

A entropia conjunta pode ser expressa por

$$H(X_1, X_2) = H(X_1) + H(X_2) - I(X_1; X_2), \quad (2.14)$$

onde observamos que a informação mútua representa o número de bits que podem ser economizados através da codificação conjunta de dois *pixels* dependentes.

Mesmo se os *pixels* forem independentes, é possível aumentar a compressão através da codificação conjunta. Vamos supor, por exemplo, que desejamos codificar N *pixels* independentes, $\{X_1, X_2, \dots, X_N\}$. Codificando cada um separadamente, utilizaríamos entre $\sum_{n=1}^N H(X_n)$ e $\sum_{n=1}^N [H(X_n) + 1]$ bits, enquanto que, com a codificação conjunta, o limite superior seria igual a $\left[\sum_{n=1}^N H(X_n) \right] + 1$. O “codificador aritmético”, por exemplo, tira proveito desse fato sem aumentar sua complexidade com N . Já o “codificador de Huffman”, embora seja ótimo, tem uma complexidade que aumenta com N ao ser utilizado para realizar a codificação conjunta (*cf.* Seção 2.9).

Para tirar proveito da dependência entre *pixels*, podem ser utilizados diversos métodos. Dentre eles, destacam-se:

1. A **codificação vetorial**, que utiliza códigos para blocos de *pixels*. Por exemplo, considerando o caso dos dois *pixels* X_1 e X_2 , pode-se associar a cada par (x_1, x_2) uma palavra-código.
2. A **codificação preditiva**, na qual o codificador utiliza informação dos *pixels* codificados anteriormente para codificar o *pixel* seguinte. Nesse caso, por exemplo, X_1 é codificado em primeiro lugar usando um código baseado em uma estimativa de p_{X_1} , que utiliza algo próximo a $H(X_1)$ bits, e X_2 é codificado posteriormente usando um código baseado em $p_{X_2|X_1}$, que utiliza em torno de $H(X_2|X_1)$ bits.
3. A **codificação por transformada**, que é utilizada quando, por exemplo, X_1 e X_2 podem ser escritos como combinações lineares de duas variáveis não correlatas, Y_1 e Y_2 , que podem ser codificadas separadamente. Nesse caso,

$$H(X_1, X_2) = H(Y_1, Y_2) \quad (2.15)$$

$$= H(Y_1) + H(Y_2|Y_1) \quad (2.16)$$

$$\approx H(Y_1) + H(Y_2) \quad (2.17)$$

$$< H(X_1) + H(X_2). \quad (2.18)$$

Esse é um método muito utilizado em codificação de imagens, na qual geralmente se faz uso de uma transformada linear, que é aplicada aos conjuntos de *pixels* da imagem, a fim de se separar os seus componentes decorrelatos. Mais detalhes sobre a codificação por transformada aplicada à compressão de imagens são discutidos no Cap. 3.

2.6 Irrelevância da Distorção

Quando a exigência de reconstrução perfeita é descartada, é possível reduzir ainda mais o tamanho da representação binária de uma imagem. Esse ganho de eficiência ocorre em contra-partida à introdução controlada de distorções na imagem codificada. Tais distorções são aceitáveis desde que sejam irrelevantes do ponto de vista do observador ou da aplicação a que se destina a imagem.

As particularidades do sistema visual humano (SVH) determinam, em muitos casos, o tipo e a quantidade da distorção que o sistema de compressão pode introduzir na imagem. É conhecido, por exemplo, o fato de que a percepção pelo SVH dos componentes de

alta frequência de uma imagem é menos acurada do que a percepção dos componentes de baixa frequência. Essa característica é levada em conta, por exemplo, na *codificação por transformada*, quando se prioriza durante a *quantização* a precisão dos coeficientes de baixa frequência em detrimento da precisão dos coeficientes de alta frequência.

Mesmo que sejam perceptíveis ao SVH, alguns tipos de distorção podem ser considerados irrelevantes, desde que não comprometam a qualidade da reprodução além de um certo ponto. O caráter de irrelevância desses tipos de distorção depende fundamentalmente do grau de qualidade exigido pela aplicação, e da relação entre o custo da banda de transmissão e o benefício obtido com o incremento de qualidade da imagem reproduzida. Além disso, limitações nos sistemas de reprodução da imagem podem reduzir as exigências sobre a qualidade da imagem comprimida. Aplicações nas quais o custo da banda de transmissão é muito elevado e a capacidade de reprodução de imagens dos aparelhos é baixa, como a telefonia móvel com vídeo, exigem qualidades de reprodução inferiores às exigidas por aplicações em que a boa qualidade da reprodução é essencial e as restrições quanto à banda de transmissão são menos duras, como na transmissão digital de televisão.

Algumas aplicações determinam ainda outros tipos de irrelevância das distorções que podem ser introduzidas na imagem. São elas, por exemplo, as aplicações que necessitam apenas extrair, na maioria das vezes de maneira automatizada, algumas informações específicas da imagem, para utilização em processamento posterior. Nesses casos, o sistema de codificação deve considerar detalhes específicos da aplicação, a fim de reduzir as taxas de compressão.

2.7 Quantização

A quantização é um passo fundamental no processo de compressão de imagens. É ela que determina quanta distorção será introduzida pelo processo, e que permite o controle da taxa de compressão a ser obtida.

A **quantização** consiste basicamente em uma operação através da qual se reduz o conjunto dos possíveis valores que uma variável aleatória (escalar ou vetorial) pode assumir. Ela pode ser representada através de uma função Q que leva subconjuntos de possíveis valores da variável aleatória a valores representativos desses subconjuntos, geralmente denominados níveis de quantização. A Fig. 2.3 ilustra essa operação. Um exemplo simples de quantização é o arredondamento de um número real para o inteiro mais próximo, para o qual a regra de quantização é

$$Q(x) = n, \quad \text{se } n - 0,5 \leq x < n + 0,5, \quad n \in \mathbb{Z}. \quad (2.19)$$

Observe-se que a quantização é uma função que mapeia muitos pontos em um (*muitos-*

para-um) e, por esse motivo, não possui função inversa. Portanto, não há como recuperar os valores originais da variável após a sua quantização.

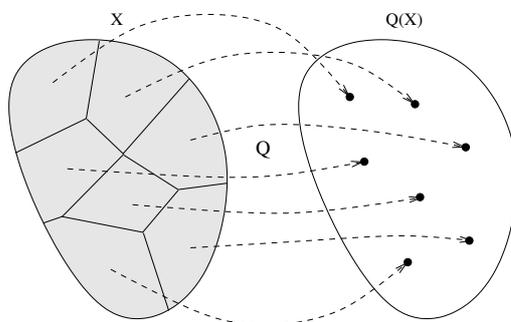


Fig. 2.3: Processo de quantização.

A aplicação de uma quantização a um vetor aleatório \mathbf{X} resulta em um novo vetor aleatório discreto $Q(\mathbf{X})$ cujo número de possíveis valores é inferior ao do vetor original. No caso da codificação digital de vetores contínuos, a quantização é uma etapa essencial, posto que não há como mapear o conjunto incontável dos valores que esse vetor pode assumir em um conjunto contável de palavras-código digitais. Em se tratando da quantização de vetores aleatórios discretos, o vetor quantizado tipicamente possui entropia menor que a entropia do vetor original, e portanto necessita de menos bits para ser representado digitalmente. Dessa forma, o controle da quantização permite que o número de bits necessários para efetuar a codificação seja escolhido arbitrariamente.

Em contra-partida à redução da entropia, a quantização produz distorção irre recuperável no sinal quantizado. Essa distorção é dada pelo vetor aleatório denominado **erro de quantização**, definido por

$$\mathbf{E}_Q = \mathbf{X} - Q(\mathbf{X}). \quad (2.20)$$

A quantidade de distorção introduzida pelo processo de quantização pode ser medida pela esperança do erro quadrático, isto é,

$$E [\|\mathbf{X} - Q(\mathbf{X})\|^2], \quad (2.21)$$

que pode ser estimada pelo erro quadrático médio (MSE) resultante da compressão, discutido na Seção 2.4.

2.8 Quantização e Codificação Progressivas

Uma característica bastante útil encontrada em alguns dos compressores de imagem é a **codificação progressiva**, que permite que uma seqüência binária comprimida c possa ser decodificada até um ponto escolhido arbitrariamente, obtendo-se um nível de distorção que varia gradativamente dependendo da escolha desse ponto. Quanto mais bits da seqüência codificada são processados pelo decodificador, menos distorcida se torna a versão reconstruída do sinal. A seqüência comprimida resultante da codificação progressiva é muitas vezes chamada de **seqüência embutida**, pois as seqüências que correspondem às diferentes taxas de compressão com que o sinal pode ser codificado estão “embutidas” em uma única seqüência.

Uma vantagem da codificação progressiva é que, caso seja necessário reduzir a taxa de compressão da imagem para, por exemplo, transmiti-la através de um canal de menor capacidade, não é necessário decodificá-la e recodificá-la novamente com menor taxa. Basta truncar a seqüência embutida no ponto desejado. Portanto, a utilização de seqüências embutidas pode ser uma importante ferramenta para lidar com os problemas advindos da heterogeneidade das redes de comunicação utilizadas atualmente.

A codificação progressiva requer que os bits que são adicionados ao fim da seqüência comprimida durante o processo de codificação sejam bits de refinamento, isto é, sejam bits que sirvam para melhorar a precisão da representação dos valores que estão sendo codificados. Portanto, em muitos casos, a codificação progressiva requer um processo de quantização progressiva, isto é, um processo em que a representação quantizada dos valores que estão sendo codificados seja refinada iterativamente.

Um exemplo de codificação e quantização progressivas é o caso em que são codificados e transmitidos os bits da representação binária de uma variável aleatória X , um a um, iniciando-se pelo bit que representa o sinal da variável, positivo ou negativo, seguido dos demais bits em ordem decrescente de significância. Como pode ser observado, à medida que os bits são codificados e transmitidos pelo codificador e são recebidos e processados pelo decodificador, a reconstrução de X se torna cada vez mais precisa, com o seu intervalo de incerteza se tornando cada vez menor.

A Fig. 2.4 mostra um exemplo de quantização progressiva. Supondo que X é uniformemente distribuída entre -128 e 128 , o primeiro bit da codificação reduz o intervalo de incerteza para um dos intervalos, $[-128, 0]$ ou $[0, 128]$, e o erro quadrático médio de $128^2/3$ para $128^2/12$. Cada bit posterior reduz os intervalos de incerteza pela metade e os erros quadráticos médios por um fator de 4.

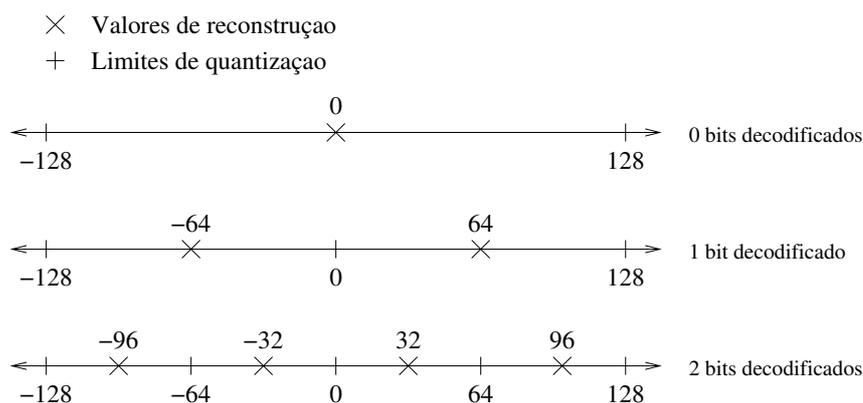


Fig. 2.4: Exemplo de quantização progressiva.

Valor	Prob.	Código	Comprimento
0	0,125	000	3
1	0,125	001	3
2	0,25	01	2
3	0,5	1	1

Tab. 2.1: Exemplo de código de entropia.

2.9 Codificação de Entropia

A **codificação de entropia** constitui-se na utilização de um código de comprimento variável para representar digitalmente de maneira eficiente, isto é, minimizando o comprimento médio da seqüência codificada, os valores de uma variável ou de um vetor aleatório discreto. Geralmente, no processo de compressão, a codificação de entropia é a etapa posterior à quantização. Ela recebe esse nome pois é com ela que se reduz a representação digital de uma variável ou vetor aleatório a um comprimento médio próximo ao valor da sua entropia.

O código de comprimento variável utilizado é baseado na distribuição de probabilidade do objeto que se deseja codificar. Desejam-se códigos que usem palavras curtas para representar os símbolos mais freqüentes e palavras mais longas para representar os símbolos mais raros.

A Tab. 2.1 mostra um exemplo de código binário que pode ser utilizado para codificar uma variável aleatória que assume os valores 0, 1, 2 e 3 com as probabilidades 0,125, 0,125, 0,25 e 0,5, respectivamente. O comprimento médio obtido por esse código é igual a 1,75 bits por símbolo da variável aleatória. Nesse caso, o comprimento médio é igual à entropia da variável aleatória.

Os **códigos de Huffman** [14] são códigos de entropia aplicáveis a seqüências de variáveis

aleatórias de alfabeto finito. Eles são ótimos no sentido de minimizar o comprimento médio da representação dentre todos os códigos instantâneos (isto é, livres de prefixos). No método de codificação, a tabela do código é produzida com base na distribuição de probabilidade da variável que se deseja codificar. A construção da tabela de código é um passo importante do projeto de um sistema de codificação, e é também o de maior complexidade computacional. As operações de codificação e decodificação são implementadas através de buscas na tabela do código e, portanto, apresentam complexidade relativamente reduzida. Os códigos de Huffman são os mais indicados para codificar seqüências longas de variáveis aleatórias independentes e cuja distribuição de probabilidade não varia. O ponto negativo desses códigos é que as variações na distribuição de probabilidade exigem que novas tabelas de codificação sejam geradas, o que pode aumentar demasiadamente a complexidade do processo.

A **codificação aritmética** [15] não faz uso da tabela do código, como no caso dos códigos de Huffman. No entanto, a sua complexidade computacional é um pouco maior em ambas as etapas de codificação e de decodificação. Assim como o código de Huffman, esse método codifica seqüências de variáveis aleatórias de alfabeto finito, porém a distribuição de probabilidade dessas variáveis pode variar livremente sem que a complexidade do processo de codificação seja significativamente alterada. A codificação aritmética codifica as variáveis de um vetor aleatório de forma seqüencial e produz uma seqüência final que corresponde à codificação conjunta de todos os componentes do vetor original. Por este motivo, a codificação aritmética pode apresentar um ganho de eficiência em relação à codificação de Huffman [14].

O método de **codificação de Golomb** [3] foi concebido para codificar variáveis aleatórias com distribuição de probabilidade geométrica [16], mas é também apropriado para a codificação de seqüências de variáveis aleatórias binárias independentes cuja distribuição de probabilidade “favorece” um dos dois possíveis resultados da variável. Esse método permite que ocorram variações na distribuição de probabilidade utilizada, desde que elas sejam lentas e conservem a propriedade de “favorecer” o mesmo resultado. Uma discussão mais detalhada sobre esse método de codificação é apresentada na Seção 4.2.

2.9.1 Compressores Especializados

Alguns sistemas de compressão utilizam métodos de quantização e codificação de entropia especialmente adaptados à estrutura dos dados que se propõem a codificar. Como exemplo podem ser citados alguns compressores de imagens baseados nas transformadas de wavelets [17], como o SPIHT (*Set Partitioning in Hierarchical Trees*) [10], que está entre os sistemas de melhor desempenho da atualidade, e o EZW (*Embedded Zero-tree Wavelet*) [11]. Esses

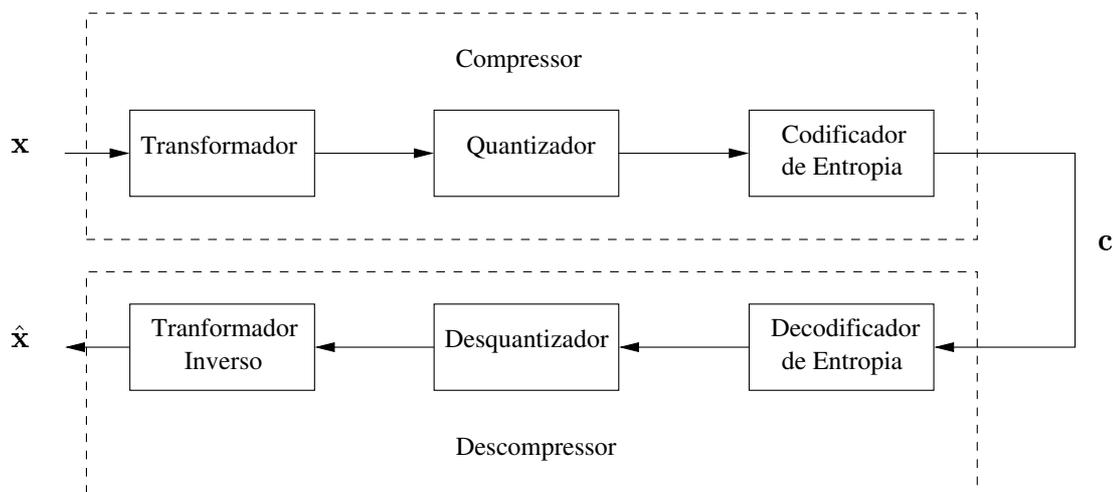


Fig. 2.5: Diagrama de um sistema de compressão/descompressão digital de imagens.

codificadores utilizam quantização progressiva e codificam os coeficientes da transformada à medida que a quantização é realizada, levando em consideração a relação de hierarquia existente entre eles. Alguns sistemas de compressão especializados, como o SPIHT [10], ainda podem utilizar um dos métodos típicos de codificação de entropia a posteriori. No entanto, a maior parte da compressão obtida por esses sistemas se deve principalmente ao método de codificação específico por eles utilizado.

2.10 Sistemas de Compressão de Imagens

A Fig. 2.5 mostra um esquema típico de um sistema de compressão e descompressão de imagens.

Na compressão, a imagem x é primeiramente submetida a uma transformada, que produz um conjunto de coeficientes que a representa de maneira completa. Essa transformada equivale a uma rotação no espaço de sinais, de modo que a correlação entre os coeficientes resultantes é geralmente menor do que a correlação entre as amostras do sinal original.

O papel do quantizador é reduzir a quantidade de possíveis valores que os coeficientes da transformada podem assumir. Essa operação constitui-se tipicamente na divisão dos valores dos coeficientes por fatores de quantização arbitrários e pelo arredondamento dos resultados.

Os coeficientes quantizados são em seguida processados pelo codificador de entropia que gera a seqüência comprimida resultante c . Quando a codificação é progressiva, quantização e codificação de entropia podem realizar-se em várias iterações, de modo que a cada iteração ocorre o refinamento das regiões de quantização dos coeficientes.

Na descompressão, o papel do “desquantizador” é tipicamente multiplicar os coeficientes quantizados pelos seus fatores de quantização correspondentes, de modo que sejam obtidos os valores aproximados dos coeficientes originais. Caso a quantização seja progressiva, o “desquantizador” deve refinar os níveis de quantização dos coeficientes a cada iteração. A transformada inversa é então aplicada aos coeficientes recuperados de modo a se obter a imagem reconstruída. Essa imagem, no entanto, será apenas uma versão aproximada da imagem original, devido aos erros decorrentes do processo de quantização.

Naturalmente, o termo “desquantizador”, também chamado de “quantizador inverso”, constitui-se em um abuso de linguagem, conquanto a operação de quantização não admite inversa. O erro de quantização não pode ser recuperado pelo sistema. Assim, o “desquantizador” representa a operação de mapear os símbolos produzidos pelo decodificador de entropia nos níveis (discretos) de quantização correspondentes.

Os sistemas de compressão geralmente variam quanto à transformada que utilizam. As mais comuns são as transformadas discretas do co-seno (DCT) [4] e as transformadas de wavelets (DWT) [11, 10]. Quanto aos métodos de quantização, os sistemas podem ser classificados em: métodos que realizam ou não quantização progressiva; e métodos que utilizam quantização vetorial [18, 19] ou quantização escalar [4] dos coeficientes. Quanto ao codificador de entropia, geralmente são utilizados os códigos de Huffman, como em [4, 11], a codificação aritmética, como em [10], ou ainda algum tipo de codificação mais especializada, como em [10, 11].

Capítulo 3

Codificação por Transformada

A classe mais vasta dos métodos de compressão de imagens é a que se baseia na utilização de uma transformada linear para tirar proveito da redundância existente entre as amostras desses sinais. A função da transformada nesses métodos é produzir uma seqüência de amostras descorrelacionadas, a partir da seqüência formada pelos *pixels* da imagem, que são em geral muito correlacionados. A seqüência de variáveis descorrelacionadas pode então ser submetida a um quantizador escalar sem grandes perdas de desempenho. Dessa forma, evita-se a alta complexidade de sistemas de quantização vetorial.

Este capítulo analisa o funcionamento das transformadas, em particular da transformada discreta do co-seno (DCT) utilizada no esquema de compressão proposto nesta dissertação.

O capítulo está dividido em 3 seções. A Seção 3.1 discute detalhes relativos a modelos estatísticos de sinais, e em particular discute um modelo estatístico para os sinais de imagem. A Seção 3.2 apresenta a definição de transformada em geral, e em particular da DCT, e examina a teoria que analisa a aplicação das transformadas a sinais modelados por processos aleatórios, como as imagens, por exemplo. A Seção 3.3 trata da quantização e codificação dos coeficientes da transformada, abordando a questão da quantização ótima e da transformada ótima, segundo o critério de minimização de erro quadrático médio.

3.1 Modelos Estatísticos

Antes de analisarmos o princípio de funcionamento dos métodos de compressão baseados em transformadas, necessitamos adotar um modelo estatístico geral que estabeleça matematicamente as características típicas dos sinais que desejamos comprimir.

Assim, vamos considerar que imagens são resultados de um processo aleatório bidimensional [16], representado por $\{x[n_1, n_2]\}$. Dessa forma, $\mathbf{x} = \{x[n_1, n_2]\}$ pode representar tanto

uma imagem propriamente dita, quanto o processo aleatório do qual ela resulta. Do mesmo modo, as amostras $x[n_1, n_2]$ podem representar tanto os *pixels* da imagem como as variáveis aleatórias que os produziram, dependendo do contexto. Ao conjunto \mathcal{X} de todos os sinais descritos acima, isto é, o espaço amostral do processo aleatório em questão, denominaremos **espaço de sinais**.

A determinação de um modelo estatístico para um tipo qualquer de sinal constitui-se em estabelecer uma função de densidade de probabilidade (f.d.p.) em \mathcal{X} . Como pode ser observado, no caso de imagens, essa não é uma tarefa simples. Uma imagem típica, com 256×256 (65536) *pixels*, por exemplo, necessitaria de uma f.d.p. com igual número de variáveis, cuja determinação pode ser de uma complexidade inviável. Além disso, não seria também fácil deduzir a f.d.p. dos coeficientes da transformada com base na f.d.p. de um sinal com tal número de variáveis. Felizmente, é possível analisar os efeitos da aplicação da transformada de maneira aproximada e com resultados interessantes apenas estabelecendo algumas hipóteses sobre o modelo estatístico utilizado.

3.1.1 Estacionariedade e Separabilidade

Um processo aleatório estacionário no sentido amplo (ESA) é um processo aleatório no qual o valor esperado das amostras é constante em relação ao tempo, isto é,

$$E[x[n_1, n_2]] = \mu_{\mathbf{x}}. \quad (3.1)$$

Além disso, a correlação entre duas amostras do processo deve depender apenas do deslocamento entre elas, isto é,

$$E[x[m_1, m_2]x[m_1 - n_1, m_2 - n_2]] = R_{xx}(n_1, n_2). \quad (3.2)$$

A função R_{xx} é denominada **função de autocorrelação** do processo, e como pode ser observado, no caso de um processo ESA ela depende apenas do deslocamento (n_1, n_2) entre as amostras. Observe-se que em (3.1) e (3.2), os $x[n_1, n_2]$ representam variáveis aleatórias.

Um processo ESA bidimensional $\{x[n_1, n_2]\}$ é separável caso

$$R_{xx}(n_1, n_2) = R_1(n_1)R_2(n_2). \quad (3.3)$$

Além disso, se ambas as direções possuem estatisticamente as mesmas características, temos que $R_1(n) = R_2(n) \triangleq R_{xx}(n)$. Portanto, podemos analisar os efeitos da aplicação de uma transformada a um processo bidimensional separável considerando apenas uma direção, isto

é, analisando apenas o caso unidimensional.

Para simplificar ainda mais a nossa análise, podemos considerar que o valor esperado do processo é nulo, de modo que o valor da função de autocorrelação para o deslocamento nulo equivale a própria variância do processo, isto é,

$$R_{xx}(0, 0) = \sigma_x^2. \quad (3.4)$$

Na prática, isto pode ser obtido subtraindo-se dos valores dos *pixels* da imagem os seus valores médios. Dessa forma, a covariância entre dois *pixels* deslocados entre si de (n_1, n_2) é dada pela própria função de autocorrelação avaliada no ponto (n_1, n_2) . Analogamente, para o sinal unidimensional, a autocorrelação entre dois *pixels* separados de n posições é dada pela função de autocorrelação $R_{xx}(n)$.

3.1.2 Densidade Espectral de Potência

A função de autocorrelação descreve parcialmente o comportamento do processo aleatório no domínio do tempo ou do espaço. Para analisar o seu comportamento no domínio da frequência utilizamos a transformada de Fourier da função de autocorrelação, a qual se denomina **densidade espectral de potência** (DEP). Ela é dada por

$$S_{xx}(e^{j\omega}) \equiv \mathcal{F}\{R_{xx}(n)\} = \sum_{n=-\infty}^{\infty} R_{xx}(n)e^{-jn\omega}. \quad (3.5)$$

A variável da frequência é ω , que é medida em radianos por amostra e varia entre $-\pi$ e π . Fora desse intervalo, o valor de $S_{xx}(e^{j\omega})$ se repete com período 2π .

Observa-se que $R_{xx}(n)$ é uma função par, isto é,

$$R_{xx}(n) = R_{xx}(-n). \quad (3.6)$$

Com isto, verifica-se que $S_{xx}(e^{j\omega})$ é real e par para qualquer valor de ω . Além disso, é possível mostrar também que $S_{xx}(e^{j\omega}) \geq 0, \forall \omega \in \mathbb{R}$.

A densidade espectral de potência está relacionada à transformada de Fourier do próprio processo aleatório. Seja $\{x[n]\}$ uma seqüência amostra de um processo aleatório estacionário. Chamando-se de $X_M(e^{j\omega})$ a transformada de Fourier de M amostras de $\{x[n]\}$, isto é,

$$X_M(e^{j\omega}) = \sum_{n=-M/2}^{M/2} x[n]e^{-jn\omega}, \quad (3.7)$$

é possível mostrar que [16]

$$S_{xx}(e^{j\omega}) = \lim_{M \rightarrow \infty} \frac{1}{M} |X_M(e^{j\omega})|^2. \quad (3.8)$$

Se considerarmos que $|X_M(e^{j\omega})|^2$ corresponde à densidade de energia de M amostras do sinal em função da frequência, esse valor dividido por M deve corresponder à densidade de energia média por amostra ou densidade de potência. Esse valor converge quando M tende a infinito e no limite corresponde à densidade espectral de potência de $\{x[n]\}$. Como esse limite é o mesmo para qualquer seqüência resultante do processo aleatório em questão, podemos dizer que ele é a própria densidade espectral de potência do processo aleatório. Isso mostra porque este nome é atribuído à transformada de Fourier da função de autocorrelação.

3.1.3 Ruídos Brancos e Coloridos

Se a função de autocorrelação é nula para $n \neq 0$ e $R_{xx}(0) = \sigma_x^2$, então $S_{xx}(e^{j\omega})$ é constante e igual a σ_x^2 . Nesse caso, chamamos o processo aleatório de **ruído branco**, posto que a densidade de potência é a mesma para todos os valores de frequência. Se amostras distintas do processo aleatório são correlacionadas, isto é, se $R_{xx}(n) \neq 0$ para algum valor de $n \neq 0$, a densidade espectral de potência não é mais plana e o processo é dito **colorido**.

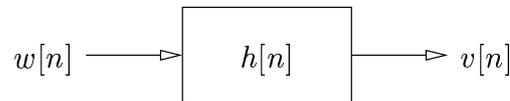


Fig. 3.1: Ruído colorido $\{v[n]\}$ gerado por um sistema linear de resposta ao impulso $\{h[n]\}$.

Um ruído colorido $\{v[n]\}$ pode ser gerado aplicando-se um ruído branco $\{w[n]\}$ à entrada de um sistema linear, como ilustrado na Fig. 3.1. Nesse caso, tem-se

$$v[n] \equiv w[n] * h[n] = \sum_{m=-\infty}^{\infty} h[m]w[n-m], \quad (3.9)$$

onde $*$ denota a operação de convolução entre seqüências e $\{h[n]\}$ corresponde à resposta ao impulso do sistema linear [2]. A função de autocorrelação de $\{v[n]\}$ é, portanto, dada por

$$R_{vv}(n) = \sigma_w^2 h[n] * h[-n], \quad (3.10)$$

e a sua densidade espectral de potência é

$$S_{vv}(e^{j\omega}) = \sigma_w^2 |H(e^{j\omega})|^2, \quad (3.11)$$

onde $H(e^{j\omega})$ é a transformada de Fourier da resposta ao impulso $\{h[n]\}$.

3.1.4 Modelos Auto-regressivos

Diversos tipos de sinais podem ser modelados por um tipo especial de ruído colorido, no qual o valor de uma amostra é gerado a partir dos valores das amostras anteriores de acordo com a seguinte equação:

$$v[n] = a_1 v[n-1] + a_2 v[n-2] + \dots + a_N v[n-N] + w[n], \quad (3.12)$$

em que $w[n]$ é um ruído branco. Nesse caso, como o valor presente do sinal é função dos valores passados do sinal e do valor presente do ruído branco, esse modelo é denominado **auto-regressivo** (AR). Um sinal auto-regressivo é produzido por um sistema linear que possui apenas pólos [2] e cuja entrada é um ruído branco, como ilustrado pela Fig. 3.2, onde $H(z)$ é a transformada Z da resposta ao impulso do sistema em questão, dada por

$$H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n}. \quad (3.13)$$

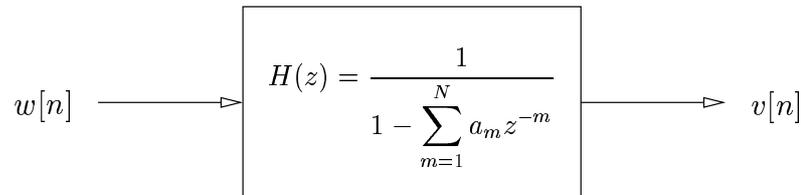


Fig. 3.2: Modelo auto-regressivo.

Um processo AR de primeira ordem, isto é, com $N = 1$, é definido por

$$x[n] = \rho x[n-1] + w[n], \quad (3.14)$$

em que ρ possui um significado especial: ele é o coeficiente de correlação de duas amostras adjacentes, isto é,

$$\rho = \frac{E[x[n]x[n-1]]}{\sigma_x^2}. \quad (3.15)$$

Para este processo tem-se que

$$R_{xx}(0) = \sigma_x^2 = \frac{\sigma_w^2}{1 - \rho^2}, \quad (3.16)$$

$$R_{xx}(n) = \sigma_x^2 \rho^{|n|}, \quad (3.17)$$

$$S_{xx}(e^{j\omega}) = \frac{1 - \rho^2}{1 + \rho^2 - 2\rho \cos \omega} \sigma_x^2. \quad (3.18)$$

3.1.5 Matrizes de Autocorrelação, Autovalores e Autovetores

Em certas situações é vantajoso utilizar uma notação matricial para representar os elementos que constituem o nosso modelo de sinais aleatórios. Por exemplo, na discussão sobre as transformadas de sinais (Seção 3.2) essa notação será de grande utilidade. Assim, apresentamos aqui alguns conceitos que utilizam essa notação.

Representaremos N amostras de um processo aleatório $\{x[n]\}$ na forma de um vetor-coluna de N elementos dado por $\mathbf{x} = [x[0] \ x[1] \ \dots \ x[N-1]]^t$, onde $()^t$ indica a operação de transposição matricial. Definimos, então, a matriz de autocorrelação desse vetor aleatório como $\mathbf{R}_{\mathbf{xx}} \triangleq E[\mathbf{xx}^t]$. Considerando que $\{x[n]\}$ é um processo estacionário com função de autocorrelação $R_{xx}(n)$, tem-se que

$$\mathbf{R}_{\mathbf{xx}} = \begin{bmatrix} R_{xx}(0) & R_{xx}(1) & \dots & R_{xx}(N-1) \\ R_{xx}(1) & R_{xx}(0) & \dots & R_{xx}(N-2) \\ \vdots & \vdots & & \vdots \\ R_{xx}(N-1) & R_{xx}(N-2) & \dots & R_{xx}(0) \end{bmatrix}. \quad (3.19)$$

Observa-se que $\mathbf{R}_{\mathbf{xx}}$ é uma matriz **não-negativa definida**, o que significa que o valor escalar $\mathbf{v}^t \mathbf{R}_{\mathbf{xx}} \mathbf{v}$ é não-negativo, para qualquer vetor-coluna \mathbf{v} arbitrário de dimensão N . A demonstração parte da própria definição de $\mathbf{R}_{\mathbf{xx}}$. Assim,

$$\mathbf{v}^t \mathbf{R}_{\mathbf{xx}} \mathbf{v} = E[\mathbf{v}^t \mathbf{xx}^t \mathbf{v}] = E[(\mathbf{v}^t \mathbf{x})^2] \geq 0. \quad (3.20)$$

onde a desigualdade é inferida devido ao fato de $E[(\mathbf{v}^t \mathbf{x})^2]$ ser o valor esperado do quadrado de uma nova variável aleatória $y \triangleq \mathbf{v}^t \mathbf{x}$. Pode-se observar também que, se os elementos de \mathbf{x} forem linearmente independentes, $E[y^2]$ só é nulo se \mathbf{v} for o vetor nulo. Se essa independência linear for verificada, afirma-se que $\mathbf{R}_{\mathbf{xx}}$ é **positiva definida**. Isso implica que a matriz $\mathbf{R}_{\mathbf{xx}}$ é não singular, isto é, possui inversa, e também que todos os seus *autovalores* são positivos.

Associado a toda matriz de autocorrelação $\mathbf{R}_{\mathbf{xx}}$, existe um conjunto de **autovalores** λ_i 's

e autovetores \mathbf{e}_i 's definidos por

$$\mathbf{R}_{\mathbf{xx}}\mathbf{e}_i = \lambda_i\mathbf{e}_i, \quad i = 0, 1, \dots, N-1. \quad (3.21)$$

Estes autovetores são ortogonais, isto é,

$$\mathbf{e}_i^t\mathbf{e}_j = 0, \quad \text{se } i \neq j, \quad (3.22)$$

e além disso, podem ser escolhidos de modo a serem ortonormais, isto é, tais que $\mathbf{e}_i^t\mathbf{e}_i = 1$, $0 \leq i < N$.

A matriz de autocorrelação pode ser reconstruída a partir dos seus autovalores e autovetores através da equação:

$$\mathbf{R}_{\mathbf{xx}} = \sum_{i=0}^{N-1} \lambda_i \mathbf{e}_i \mathbf{e}_i^t, \quad (3.23)$$

supondo-se que os autovetores estão normalizados.

Autovalores possuem ainda as seguintes propriedades:

$$\det(\mathbf{R}_{\mathbf{xx}}) = \prod_{i=0}^{N-1} \lambda_i; \quad \text{traço}(\mathbf{R}_{\mathbf{xx}}) = \sum_{i=0}^{N-1} \lambda_i. \quad (3.24)$$

Observa-se também que o traço de $\mathbf{R}_{\mathbf{xx}}$ é igual a $N\sigma_x^2$, para um processo de média zero, de onde se conclui que a média dos autovalores é igual a variância do processo aleatório, isto é,

$$\sigma_x^2 = \frac{1}{N} \sum_{i=0}^{N-1} \lambda_i. \quad (3.25)$$

Além disso, se $S_{xx}(e^{j\omega})$ é a densidade espectral de potência do processo cuja matriz de autocorrelação é $\mathbf{R}_{\mathbf{xx}}$, então existe um conjunto de frequências $\{\Omega_0, \Omega_1, \dots, \Omega_{N-1}\}$ tais que

$$\lambda_i = S_{xx}(e^{j\Omega_i}), \quad (3.26)$$

isto é, os autovalores são amostras da densidade espectral de potência de $\{x[n]\}$ neste conjunto de frequências, geralmente denominadas **autofrequências**. Ressalva-se que as autofrequências não são necessariamente equidistantes.

Um importante resultado deriva do *teorema da distribuição de Toeplitz (Teorema de*

Szego [20]), que afirma que, para qualquer função $f(\cdot)$,

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=0}^{N-1} f(\lambda_i) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f[S_{xx}(e^{j\omega})] d\omega. \quad (3.27)$$

Se tomarmos $f(\cdot) = \log_e(\cdot)$ teremos que

$$\lim_{N \rightarrow \infty} \log_e \left[\prod_{i=0}^{N-1} \lambda_i \right]^{1/N} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log_e S_{xx}(e^{j\omega}) d\omega, \quad (3.28)$$

de onde se conclui que

$$\lim_{N \rightarrow \infty} |\mathbf{R}_{\mathbf{xx}}|^{1/N} = \exp \left[\frac{1}{2\pi} \int_{-\pi}^{\pi} \log_e S_{xx}(e^{j\omega}) d\omega \right]. \quad (3.29)$$

3.1.6 Modelo Estatístico de Imagens

Em primeiro lugar, vamos supor que o conjunto dos possíveis valores dos *pixels* de uma imagem será contínuo, embora, na prática, estejamos lidando com imagens digitais, cujos *pixels* assumem apenas valores discretos. Essa suposição é válida desde que a discretização realizada pelo sistema de aquisição de imagens seja suficientemente “fina”, isto é, desde que a diferença entre o valor “real” da intensidade luminosa do *pixel* captada pelo sistema de aquisição e o seu valor registrado (quantizado) possa ser considerada desprezível [21]. Essa diferença é em geral muito menor do que a diferença introduzida no sinal pelo próprio processo de compressão, de modo que esta suposição é normalmente considerada aceitável. Assim, deste ponto em diante, consideraremos que uma **imagem** é um **sinal** representado por uma seqüência bidimensional finita descrita por:

$$\mathbf{X} = \{x[n_1, n_2] \in \mathbb{R}, (n_1, n_2) \in Z_{N_1 \times N_2}\}, \quad (3.30)$$

cujo domínio é definido pelo conjunto

$$Z_{N_1 \times N_2} \triangleq \{(i, j) \mid 0 \leq i < N_1, 0 \leq j < N_2\}. \quad (3.31)$$

Algumas regiões de uma imagem, principalmente aquelas em que as variações dos *pixels* ocorrem de maneira mais bem “comportada”, isto é, mais suavemente, podem ser bem modeladas por processos auto-regressivos de 1ª ordem, com ρ próximo a 1 [22, 20]. A densidade espectral de potência de um processo dessa natureza é dada por (3.18), e apresenta a característica da maior parte da potência do sinal estar concentrada nas baixas freqüências, isto é,

o valor da densidade decai à medida que a frequência aumenta, conforme pode ser observado na Fig. 3.3, que mostra as DEP's de processos AR de 1ª ordem com diferentes valores de ρ .

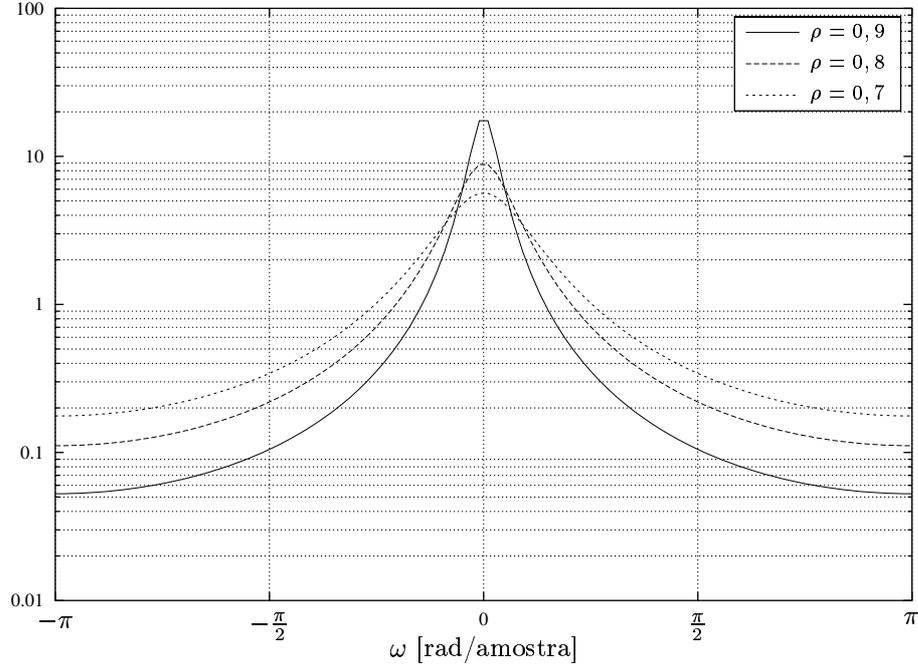


Fig. 3.3: Densidade espectral de potência normalizada ($\sigma_x^2 = 1$) de processo auto-regressivo de primeira ordem.

A matriz de autocorrelação de processos AR de 1ª ordem é dada por

$$\mathbf{R}_{\mathbf{xx}} = \sigma_x^2 \cdot \begin{bmatrix} 1 & \rho & \dots & \rho^{N-1} \\ \rho & 1 & \dots & \rho^{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho^{N-1} & \rho^{N-2} & \dots & 1 \end{bmatrix}, \quad (3.32)$$

e os seus autovalores, considerando blocos de 8 amostras do processo, são mostrados na Fig. 3.4.

Outras regiões da imagem possuem “anomalias”, isto é, variações abruptas ou quase periódicas nos valores dos *pixels*. Essas anomalias, como, por exemplo, bordas de objetos e regiões de textura, não são modeladas adequadamente por processos AR de 1ª ordem, como ocorre com a maioria das regiões típicas de uma imagem. Tais regiões nem mesmo são bem modeladas por processos aleatórios estacionários, de modo que os resultados previstos para o modelo auto-regressivo não são válidos. É devido à existência dessas anomalias, que escapam ao modelo adotado, que as transformadas, ou as funções de base das transformadas, usadas

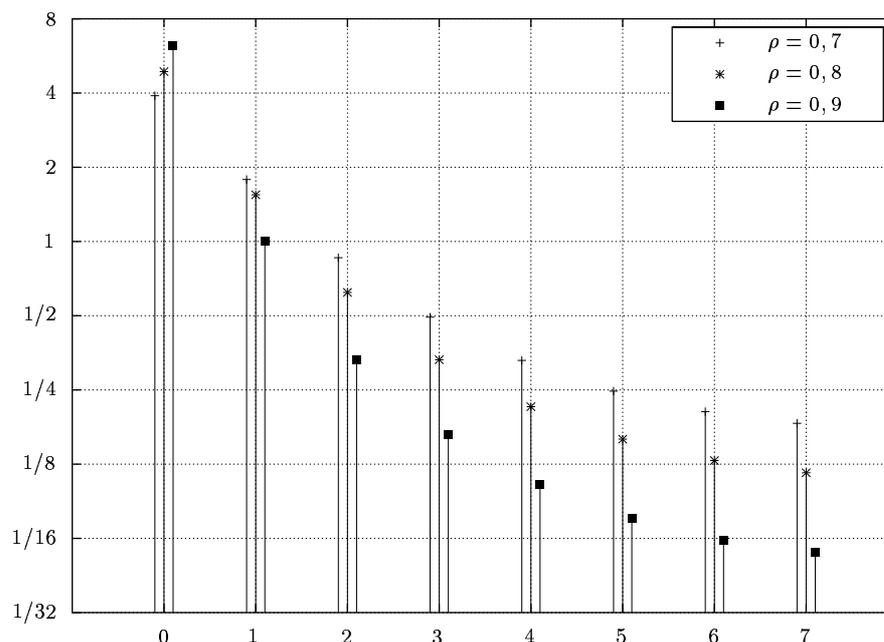


Fig. 3.4: Autovalores de processos AR de primeira ordem com diferentes valores de ρ .

na codificação devem possuir comprimentos relativamente curtos ou, como também é comum afirmar-se, devem ser bem localizadas espacialmente.

Por outro lado, as regiões com anomalias são em geral minoria em relação à imagem como um todo, de modo que é comum considerar-se nas análises apenas o modelo estacionário auto-regressivo. Além de tudo, tais regiões exigem modelos exageradamente complexos e sofisticados, e que, caso fossem levados em consideração, provavelmente não trariam grandes ganhos de eficiência à compressão.

No sistema de compressão proposto por esta dissertação, a existência dessas anomalias é considerada apenas de maneira heurística e superficial no tratamento dado aos coeficientes da DCT durante o processo de compressão (*cf.* Seção 4.4).

3.2 Transformadas

De modo geral, uma **transformada** é um mapeamento linear e inversível entre dois espaços vetoriais quaisquer. No nosso caso, os espaços considerados são formados por vetores cujos elementos são amostras das seqüências que se deseja codificar.

O uso da transformada no processo de compressão tem por objetivo produzir, a partir de uma seqüência de amostras altamente correlacionadas e dependentes, uma nova seqüência de

amostras com pouca correlação e praticamente independentes, de modo que posteriormente possa ser realizada a quantização escalar de cada amostra individual sem que ocorram grandes perdas de eficiência.

Uma transformada unidimensional de comprimento N é representada pela operação matricial

$$\hat{\mathbf{x}} = \mathbf{A}^t \mathbf{x}, \quad (3.33)$$

onde \mathbf{x} é o vetor $[x[0] \ x[1] \ \dots \ x[N-1]]^t$ formado por N amostras do sinal ao qual se deseja codificar. Da aplicação da transformada, representada pela matriz \mathbf{A} , resulta o vetor $\hat{\mathbf{x}} = [\hat{x}[0] \ \hat{x}[1] \ \dots \ \hat{x}[N-1]]^t$, cujos elementos são denominados **coeficientes da transformada**. A matriz \mathbf{A} é denominada **matriz da transformada**.

A transformada é dita **ortogonal** caso a matriz da operação inversa, $(\mathbf{A}^t)^{-1}$, seja igual à matriz \mathbf{A} da transformada, ou seja, se

$$\mathbf{A}^{-1} = \mathbf{A}^t. \quad (3.34)$$

Nesse caso, a transformada pode ser vista como uma decomposição do sinal em uma base de sinais ortonormais $\{\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_{N-1}\}$, que correspondem às colunas de \mathbf{A} . A seqüência original \mathbf{x} pode ser escrita como uma combinação linear dos sinais da base, isto é,

$$\mathbf{x} = \sum_{k=0}^{N-1} \hat{x}[k] \mathbf{w}_k, \quad (3.35)$$

onde os coeficientes correspondem aos produtos internos, denotados por $\mathbf{x}^t \mathbf{w}_k$, entre o sinal que se deseja decompor e os vetores da base, isto é,

$$\hat{x}[k] = \mathbf{x}^t \mathbf{w}_k \quad (3.36)$$

$$= \sum_{n=0}^{N-1} x[n] w_k[n]. \quad (3.37)$$

A equação (3.35) pode ser escrita equivalentemente de forma matricial como

$$\mathbf{x} = \mathbf{A} \hat{\mathbf{x}}. \quad (3.38)$$

Transformadas ortogonais têm a propriedade de preservar a energia do sinal transformado.

Por exemplo, a energia média por amostra de N amostras de um sinal $\{x[n]\}$ é igual a

$$E_{\mathbf{x}} = \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \mathbf{x}^t \mathbf{x}. \quad (3.39)$$

Observando agora os coeficientes da transformada, temos

$$E_{\hat{\mathbf{x}}} = \frac{1}{N} \hat{\mathbf{x}}^t \hat{\mathbf{x}} \quad (3.40)$$

$$= \frac{1}{N} \mathbf{x}^t \mathbf{A} \mathbf{A}^t \mathbf{x} \quad (3.41)$$

$$= \frac{1}{N} \mathbf{x}^t \mathbf{x} \quad (3.42)$$

$$= E_{\mathbf{x}}. \quad (3.43)$$

No domínio da frequência, observamos que, se as funções de base \mathbf{w}_k 's apresentam espectros concentrados em regiões de frequência bem definidas, (3.36) também representa uma decomposição espectral do sinal original \mathbf{x} , com o termo $|\hat{x}[k]|^2$ indicando o quanto da energia de \mathbf{x} está presente na região de frequência em que \mathbf{w}_k está concentrada.

Transformadas Bidimensionais

Como uma generalização do caso unidimensional, a transformada de um sinal bidimensional pode ser vista como uma seqüência bidimensional de amostras que resultam do produto interno do sinal com a família de funções de base da transformada bidimensional. Considerando-se o caso particular das transformadas de bloco de tamanho $M \times N$, a transformada de um bloco \mathbf{X} do sinal $\{x[m, n]\}$ é o bloco $\hat{\mathbf{X}}$ de amostras da seqüência $\{\hat{x}[k, \ell]\}$ que é determinado pela equação

$$\hat{x}[k, \ell] = \langle \mathbf{X}, \mathbf{W}_{k, \ell} \rangle \quad (3.44)$$

$$= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x[m, n] w_{k, \ell}[m, n], \quad (3.45)$$

onde $\langle \mathbf{X}, \mathbf{W}_{k, \ell} \rangle$ representa o produto interno entre o bloco de $M \times N$ amostras da seqüência bidimensional $\{x[m, n]\}$ e uma das funções de base $\mathbf{W}_{k, \ell} = \{w_{k, \ell}[m, n]\}$ de tamanho $M \times N$ da transformada.

Analogamente ao caso unidimensional, se a base de vetores da transformada bidimensional é ortonormal, a transformada é ortogonal (também denominada unitária) e a equação de síntese é nada mais nada menos que uma combinação linear dos vetores da base, tendo

como coeficientes os próprios coeficientes da transformada, isto é,

$$x[m, n] = \sum_{k=0}^{M-1} \sum_{\ell=0}^{N-1} \hat{x}[k, \ell] w_{k, \ell}[m, n]. \quad (3.46)$$

Uma transformada bidimensional é dita separável caso suas funções de base possam ser escritas como produtos tensoriais das funções de base de duas transformadas unidimensionais, isto é,

$$\mathbf{W}_{k, \ell} = \mathbf{u}_k \otimes \mathbf{v}_\ell \triangleq w_{k, \ell}[m, n] = u_k[m] v_\ell[n], \quad (3.47)$$

onde \mathbf{u}_k , $0 \leq k < M$, e \mathbf{v}_ℓ , $0 \leq \ell < N$ são os vetores de base de duas transformadas unidimensionais, com comprimentos M e N , respectivamente. Nesse caso, a transformada bidimensional pode ser calculada através da aplicação das transformadas unidimensionais consecutivamente às linhas e às colunas do sinal, não importando a ordem. Assim, dado que um bloco do sinal $\{x(m, n)\}$ é representado por uma matriz \mathbf{X} de M linhas por N colunas, é possível descrever a transformada bidimensional separável através da operação de multiplicação de matrizes

$$\hat{\mathbf{X}} = \mathbf{U}^t \mathbf{X} \mathbf{V}, \quad (3.48)$$

onde \mathbf{U} é uma matriz $M \times M$ que representa a transformada aplicada às colunas de \mathbf{X} e \mathbf{V} é uma matriz $N \times N$ que representa a transformada aplicada às linhas de \mathbf{X} .

Usualmente a mesma transformada é utilizada nas linhas e nas colunas do sinal bidimensional. Nesse caso, se \mathbf{X} é uma matriz que representa um bloco de $M \times M$ amostras de um sinal bidimensional $\{x(m, n)\}$, a transformada bidimensional separável é escrita da forma:

$$\hat{\mathbf{X}} = \mathbf{U}^t \mathbf{X} \mathbf{U}. \quad (3.49)$$

Nesta dissertação a transformada bidimensional a ser utilizada é separável. Portanto, as análises realizadas consideram apenas o caso unidimensional, já que a mesma transformada é aplicada às linhas e às colunas de \mathbf{X} . Como pode ser observado, esta abordagem é adequada ao modelo estatístico de imagem adotado, que também é suposto separável.

3.2.1 Transformadas de Processos Aleatórios

Vamos supor agora que estejamos aplicando uma transformada a um processo aleatório, ou seja, \mathbf{x} agora é um vetor aleatório cujos elementos são N amostras do processo aleatório $\{x[n]\}$. Também vamos supor que $\{x[n]\}$ seja estacionário. Os valores médios dos coeficientes

da transformada são dados por

$$E[\hat{\mathbf{x}}] = \mathbf{A}^t E[\mathbf{x}] = \mathbf{A}^t [\mu_{\mathbf{x}} \cdots \mu_{\mathbf{x}}]^t, \quad (3.50)$$

de onde concluímos que

$$E[\hat{x}[k]] = \mu_{\mathbf{x}} \sum_{n=0}^{N-1} w_k[n], \quad (3.51)$$

onde $\{w_k[n]\}$ são as funções de base da transformada. Note-se que se $\{x[n]\}$ possuir média nula, a média dos coeficientes também será nula.

Vamos mostrar agora que, caso a transformada seja ortogonal, a média das variâncias dos elementos de \mathbf{x} será preservada. Vamos chamar de σ_k^2 a variância de $\hat{x}[k]$. Supondo que a média de $\{x[n]\}$ seja nula, $\sigma_k^2 = E[\hat{x}[k]^2]$. Observamos que

$$\frac{1}{N} \sum_{k=0}^{N-1} \sigma_k^2 = \frac{1}{N} E[\hat{\mathbf{x}}^t \hat{\mathbf{x}}] \quad (3.52)$$

$$= \frac{1}{N} E[\mathbf{x}^t \mathbf{A} \mathbf{A}^t \mathbf{x}] \quad (3.53)$$

$$= \frac{1}{N} E[\mathbf{x}^t \mathbf{x}] \quad (3.54)$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} \sigma_{x[n]}^2 \quad (3.55)$$

$$= \sigma_{\mathbf{x}}^2. \quad (3.56)$$

onde a última igualdade é válida porque $\{x[n]\}$ é um processo aleatório estacionário.

3.2.2 Transformada de Karhunen-Loève

A **transformada de Karhunen-Loève** (KLT) é uma transformada definida com base em um processo aleatório estacionário, isto é, ela não possui uma definição fixa, mas depende do processo em que se baseia. Uma KLT unidimensional de tamanho N é dada pela matriz

$$\mathbf{E} \triangleq \begin{bmatrix} \mathbf{e}_0 & \mathbf{e}_1 & \cdots & \mathbf{e}_{N-1} \end{bmatrix}, \quad (3.57)$$

onde \mathbf{e}_i , $i = 0, 1, \dots, N-1$, são os autovetores normalizados da matriz de autocorrelação $\mathbf{R}_{\mathbf{xx}}$ de dimensão $N \times N$ do processo aleatório estacionário. Portanto, os vetores de base da KLT são os próprios autovetores \mathbf{e}_i do processo aleatório. Observe-se que, devido ao fato dos autovetores serem ortonormais, a matriz da transformada é ortogonal e, portanto, a matriz

da transformada inversa é igual à transposta da matriz da transformada direta.

A principal propriedade da KLT é descrita a seguir: a matriz de autocorrelação dos coeficientes da KLT é dada por

$$\mathbf{R}_{\hat{\mathbf{x}}\hat{\mathbf{x}}} = E[\hat{\mathbf{x}}\hat{\mathbf{x}}^t] \quad (3.58)$$

$$= E[\mathbf{E}^t \mathbf{x} \mathbf{x}^t \mathbf{E}] \quad (3.59)$$

$$= \mathbf{E}^t E[\mathbf{x} \mathbf{x}^t] \mathbf{E} \quad (3.60)$$

$$= \mathbf{E}^t \mathbf{R}_{\mathbf{x}\mathbf{x}} \mathbf{E}, \quad (3.61)$$

e pela definição de autovalores e autovetores em (3.21), temos que

$$\mathbf{R}_{\mathbf{x}\mathbf{x}} \mathbf{E} = \begin{bmatrix} \lambda_0 \mathbf{e}_0 & \lambda_1 \mathbf{e}_1 & \cdots & \lambda_{N-1} \mathbf{e}_{N-1} \end{bmatrix}. \quad (3.62)$$

Portanto, a matriz de autocorrelação de $\hat{\mathbf{x}}$ é uma matriz diagonal cujos elementos são os próprios autovalores de $\mathbf{R}_{\mathbf{x}\mathbf{x}}$, ou seja,

$$\mathbf{R}_{\hat{\mathbf{x}}\hat{\mathbf{x}}} = \begin{bmatrix} \lambda_0 & 0 & \cdots & 0 \\ 0 & \lambda_1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \lambda_{N-1} \end{bmatrix}. \quad (3.63)$$

Assim, observa-se que os coeficientes da KLT são variáveis aleatórias descorrelacionadas e suas variâncias são os autovalores da matriz de autocorrelação de \mathbf{x} .

Devido a este fato, verifica-se que a KLT é a melhor transformada que pode ser utilizada no processo de compressão de sinais produzidos por processos aleatórios estacionários, já que ela descorrelaciona completamente as amostras do sinal original. A “optimalidade” da KLT é demonstrada na Seção 3.3.3.

Por outro lado, como a KLT é uma transformada que depende do sinal a ser comprimido, ela raramente é utilizada na prática em sistemas de compressão. A sua utilização geralmente implica em uma sobre-taxa de transmissão, devido à necessidade de se enviar ao decodificador os elementos da matriz de transformação. Além disso, o cálculo da KLT geralmente apresenta maior custo computacional, devido à inexistência de algoritmos rápidos para este fim. Outras transformadas, como a DCT (transformada discreta do co-seno) e a DFT (transformada discreta de fourier), possuem algoritmos rápidos de cálculo. Esse é um importante fator a ser levado conta na escolha de uma transformada para aplicações práticas.

A DCT, discutida a seguir, é uma transformada que não depende do sinal ao qual ela é

aplicada e que possui diversos algoritmos rápidos de cálculo. Ela é muito semelhante à KLT de processos AR de 1^a ordem em que ρ é próximo de 1. A Fig. 3.5 mostra as funções de base da KLT de comprimento 8 de um processo AR de 1^a ordem com $\rho = 0,9$ e da DCT de mesmo comprimento.

3.2.3 Transformada Discreta do Co-seno

As funções de base da DCT unidimensional são obtidas a partir da amostragem de funções co-seno com diferentes frequências. Dependendo dos instantes de amostragem escolhidos a DCT pode ser definida de diferentes maneiras. As mais usuais são as denominadas DCT-I e DCT-II [2]. A DCT-II é a transformada mais comumente usada em aplicações de compressão de imagens.

O método de compressão de imagens proposto por esta dissertação utiliza uma DCT bidimensional separável baseada na DCT-II, como ela está definida em [2], com comprimento igual a 8 e normalizada, de modo que ela também seja uma transformada unitária (ortogonal).

A **equação de análise** ou **transformada direta** da DCT-II normalizada de comprimento N é dada por

$$\hat{x}[k] = \sqrt{\frac{\beta[k]}{N}} \sum_{n=0}^{N-1} x[n] \cos \left[\frac{\pi k(2n+1)}{2N} \right], \quad 0 \leq k \leq N-1, \quad (3.64)$$

onde $\beta[k]$ é uma função dada por

$$\beta[k] = \begin{cases} 1, & \text{se } k = 0, \\ 2, & \text{se } 1 \leq k \leq N-1. \end{cases} \quad (3.65)$$

A DCT unidimensional (DCT-1D) decompõe um sinal discreto de comprimento N em um conjunto de componentes $\hat{x}[k]\mathbf{w}_k$, para $k \in \{0, 1, \dots, N-1\}$, onde \mathbf{w}_k são as funções de base da transformada de comprimento N , dadas por

$$\mathbf{w}_k \equiv w_k[n] = \sqrt{\frac{\beta[k]}{N}} \cos \left[\frac{\pi k(2n+1)}{2N} \right], \quad 0 \leq n \leq N-1. \quad (3.66)$$

Estas funções compõem a base vetorial da transformada discreta do co-seno. Na Fig. 3.5 podem ser vistas as funções de base da DCT-1D de comprimento 8. Pode ser observada a semelhança entre as funções de base da DCT e da KLT de um processo AR de 1^a ordem, também mostradas na figura.

O sinal \mathbf{x} pode ser reconstruído através da soma de seus componentes $\hat{x}[k]\mathbf{w}_k$, $k \in$

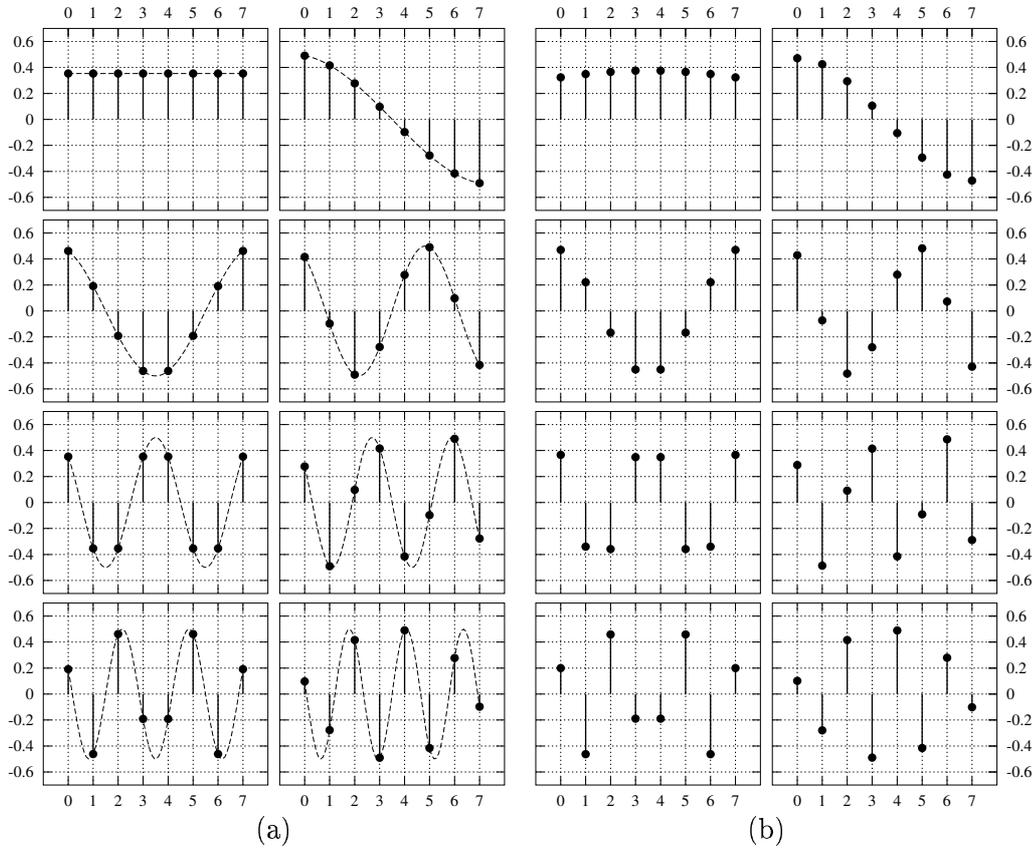


Fig. 3.5: Funções-base de comprimento 8 da DCT-1D (a) e da KLT-1D de um processo AR com $\rho = 0,9$ (b).

$\{0, 1, \dots, N - 1\}$, o que é denominado equação de síntese, isto é,

$$\mathbf{x} = \sum_{k=0}^{N-1} \hat{x}[k] \mathbf{w}_k, \quad 0 \leq k \leq N - 1, \quad (3.67)$$

também denominada **transformada inversa** da DCT-1D, em oposição a transformada direta ou equação de análise (3.64).

A transformada descrita por (3.64) e (3.67) é unitária, no sentido de que ela preserva a energia do sinal, ou seja,

$$\sum_{k=0}^{N-1} |\hat{x}[k]|^2 = \sum_{n=0}^{N-1} |x[n]|^2, \quad (3.68)$$

para qualquer seqüência real $\{x[n]\}$ de comprimento N .

Uma vantagem da DCT em relação a outras transformadas é a possibilidade de calculá-la de maneira rápida. Loeffler *et al.* [9], por exemplo, mostram uma maneira de calcular a DCT-1D de comprimento 8 com apenas 11 multiplicações e 29 somas.

Processos AR de Primeira Ordem

A Fig. 3.6 mostra as variâncias dos coeficientes da DCT de comprimento 8 aplicada a amostras de processos AR de 1^a ordem. Observa-se que esses valores são muito próximos dos autovalores do processo, mostrados na Fig. 3.4, o que sugere que a DCT obtém praticamente os mesmos resultados que a KLT para esses processos. O índice k , que corresponde ao índice das diferentes funções de base que compõem a DCT, está relacionado à região de frequência correspondente a cada coeficiente. À medida que esse índice aumenta, maior é a frequência da sua função de base correspondente. A variância de um coeficiente corresponde à energia média do sinal na região de frequência da função da base correspondente. Como se observa, essa energia diminui à medida que a frequência aumenta.

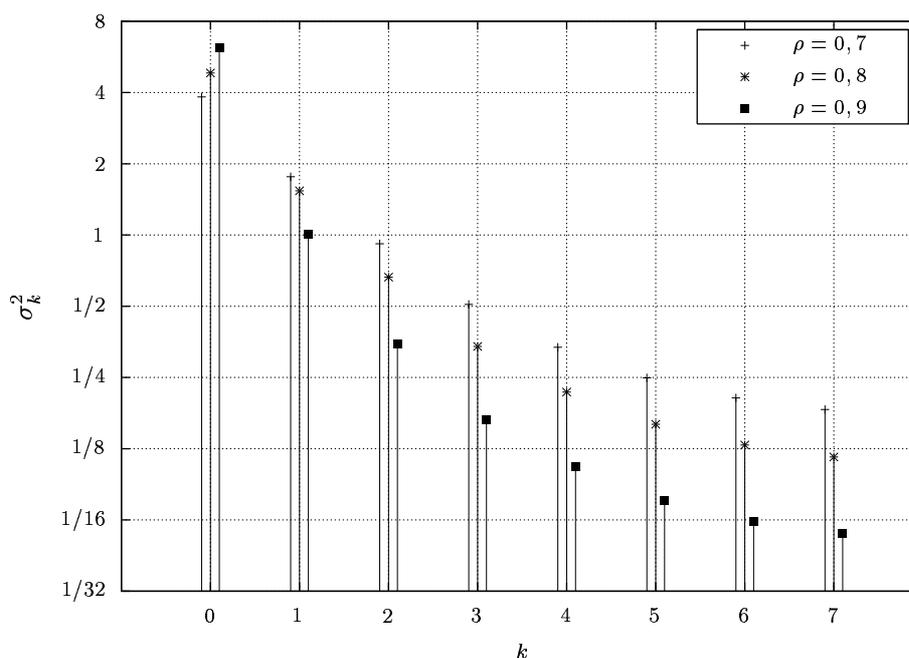


Fig. 3.6: Variâncias dos coeficientes da DCT de processos AR de primeira ordem com diferentes valores de ρ .

DCT Bidimensional

A DCT bidimensional (DCT-2D), que é utilizada pelo método de compressão proposto nesta dissertação, é uma transformada separável que se deriva diretamente da transformada unidimensional. As funções de base da DCT-2D são geradas pelo produto tensorial entre as funções de base da DCT unidimensional (DCT-1D). Assim, dado que \mathbf{w}_k , $0 \leq k < N$, são as funções de base da DCT-1D, as funções de base da DCT-2D são dadas por

$$\mathbf{W}_{k,\ell} = \mathbf{w}_k \otimes \mathbf{w}_\ell, \quad 0 \leq k < N, \quad 0 \leq \ell < N. \quad (3.69)$$

Desta forma, se \mathbf{A} é a matriz da transformada unidimensional e \mathbf{X} é uma matriz formada por um bloco de $N \times N$ *pixels* de uma imagem, a DCT-2D pode ser escrita como

$$\hat{\mathbf{X}} = \mathbf{A}^t \mathbf{X} \mathbf{A}, \quad (3.70)$$

de modo que o elemento $\hat{x}_{k,\ell}$ da matriz $\hat{\mathbf{X}}$ corresponde ao coeficiente da função de base $\mathbf{W}_{k,\ell}$ da transformada bidimensional. As regiões de frequência dos coeficientes da DCT-2D pertencem a um plano bidimensional de frequências, que correspondem as variações no eixo vertical e no eixo horizontal.

A transformada inversa é escrita de forma matricial como

$$\mathbf{X} = \mathbf{A} \hat{\mathbf{X}} \mathbf{A}^t, \quad (3.71)$$

e corresponde a síntese do sinal a partir dos coeficientes da transformada.

3.3 Codificação dos Coeficientes da Transformada

Nos sistemas de codificação por transformada que estamos considerando, isto é, os que utilizam transformadas de bloco, o sinal que se deseja codificar é inicialmente dividido em blocos de N amostras sucessivas, ou $N \times N$ amostras, no caso de sinais bidimensionais. Posteriormente a transformada é aplicada a cada bloco de amostras e os coeficientes resultantes são quantizados. Após a quantização, os coeficientes são codificados digitalmente para posterior transmissão. Na decodificação, os valores quantizados são recuperados e a transformada inversa é aplicada a eles, de modo a se obter o sinal reconstruído.

Vamos denominar de $\hat{\mathbf{y}} = Q(\hat{\mathbf{x}})$ a um bloco de coeficientes quantizados, enquanto \mathbf{y} é o resultado da aplicação da transformada inversa a $\hat{\mathbf{y}}$, ou seja, é um bloco do sinal reconstruído. O erro de quantização, nesse caso, é o vetor $\mathbf{u} \triangleq \mathbf{x} - \mathbf{y} = [u[0] \ u[1] \ \dots \ u[N-1]]^t$. Os

erros quadráticos médios das amostras reconstruídas são iguais às variâncias dos elementos de \mathbf{u} , isto é, $\sigma_{un}^2 = E[u[n]^2]$, para $n = 0, 1, \dots, N - 1$, e a média desses erros acerca das N amostras do bloco é

$$\sigma_u^2 = \frac{1}{N} \sum_{n=0}^{N-1} \sigma_{un}^2 \quad (3.72)$$

$$= \frac{1}{N} E[\mathbf{u}^t \mathbf{u}]. \quad (3.73)$$

Se definirmos $\hat{\mathbf{u}} \triangleq \hat{\mathbf{X}} - \hat{\mathbf{Y}} = [\hat{u}[0] \hat{u}[1] \dots \hat{u}[N-1]]^t$ como sendo o vetor dos erros de quantização dos coeficientes da transformada, observamos que os erros quadráticos médios são dados por $\sigma_{\hat{u}k}^2 = E[|\hat{u}[k]|^2]$, para $k = 0, 1, \dots, N-1$. Observando que $\hat{\mathbf{u}} = \mathbf{A}^t \mathbf{U}$, pode-se verificar que a média acerca das N amostras dos erros quadráticos médios dos coeficientes, representada por

$$\sigma_{\hat{u}}^2 = \frac{1}{N} \sum_{k=0}^{N-1} \sigma_{\hat{u}k}^2, \quad (3.74)$$

é igual a σ_u^2 , o que pode ser demonstrado por:

$$\begin{aligned} \sigma_{\hat{u}}^2 &= \frac{1}{N} \sum_{k=0}^{N-1} \sigma_{\hat{u}k}^2 \\ &= \frac{1}{N} E[\hat{\mathbf{u}}^t \hat{\mathbf{u}}] \\ &= \frac{1}{N} E[\mathbf{u}^t \mathbf{A} \mathbf{A}^t \mathbf{u}] \\ &= \frac{1}{N} E[\mathbf{u}^t \mathbf{u}] \\ &= \frac{1}{N} \sum_{n=0}^{N-1} \sigma_{un}^2 \\ &= \sigma_u^2. \end{aligned} \quad (3.75)$$

Portanto, o uso de uma transformada ortogonal garante que o MSE dos coeficientes quantizados da transformada é igual ao MSE do sinal reconstruído.

3.3.1 Alocação Ótima de Bits

Agora vamos considerar o problema da alocação de bits para a codificação dos coeficientes quantizados. O objetivo do nosso sistema de codificação é minimizar o MSE do sinal reconstruído, isto é, σ_u^2 . Levando em conta (3.75), que mostra que $\sigma_u^2 = \sigma_{\hat{u}}^2$, a busca por se

minimizar a média dos erros quadráticos médios dos *pixels*, σ_u^2 , equivale a procurar minimizar a média aritmética dos erros quadráticos médios dos coeficientes da transformada, $\sigma_{\hat{u}}^2$.

A variância do erro de quantização de cada coeficiente da transformada está relacionada à variância do próprio coeficiente através da expressão

$$\sigma_{\hat{u}k}^2 = \epsilon_{\hat{u}k}^2 \sigma_k^2, \quad (3.76)$$

onde $\sigma_k^2 = E[\hat{x}[k]^2]$ representa a variância do k -ésimo coeficiente da transformada, e $\epsilon_{\hat{u}k}^2$ representa o **fator de desempenho** do quantizador que é definido como a razão entre a variância do erro de quantização e a variância da variável aleatória que está sendo quantizada [20]. À medida que a quantização se torna mais “fina”, isto é, que os níveis de quantização ficam mais próximos entre si, esse fator de desempenho se torna menor. Se a quantização já é suficientemente fina, ao dobrar-se o número de níveis de quantização, $\epsilon_{\hat{u}k}^2$ é reduzido por um fator de 1/4, já que os intervalos de quantização são reduzidos aproximadamente pela metade. Além disso, o dobro do número de níveis de quantização geralmente necessita de um bit a mais para ser codificado. Portanto, a variância do erro de quantização pode ser escrita, em função do número de bits utilizados para codificar o coeficiente, de maneira aproximada como

$$\sigma_{\hat{u}k}^2 = \epsilon_{\hat{u}k}^2 \sigma_k^2 \approx \epsilon_*^2 2^{-2R_k} \sigma_k^2, \quad (3.77)$$

onde ϵ_*^2 é uma constante de proporcionalidade que leva em consideração o desempenho do quantizador utilizado na prática e o método de codificação de entropia usado na saída do quantizador. R_k representa o número de bits utilizados para codificar o k -ésimo coeficiente quantizado, enquanto σ_k^2 representa a variância desse coeficiente. É necessário salientar que a aproximação expressa em (3.77) é tanto melhor quanto mais “fina” for a quantização. Para valores baixos de R_k ela é geralmente pouco precisa.

Podemos assim expressar a taxa de bits R_k do k -ésimo coeficiente por

$$R_k = \frac{1}{2} \log_2 \epsilon_*^2 + \frac{1}{2} \log_2 \frac{\sigma_k^2}{\sigma_{\hat{u}k}^2}, \quad (3.78)$$

e a taxa de bits média por coeficiente considerando os N coeficientes da transformada por

$$R = \frac{1}{N} \sum_{k=0}^{N-1} R_k. \quad (3.79)$$

De acordo com o critério de mínimo MSE e levando em conta (3.75), o método de quantização e codificação deve procurar minimizar a média das variâncias dos erros de quantização dos

coeficientes da transformada, isto é,

$$\sigma_{\hat{u}}^2 = \frac{1}{N} \sum_{k=0}^{N-1} \sigma_{\hat{u}k}^2 \quad (3.80)$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} \epsilon_*^2 2^{-2R_k} \sigma_k^2, \quad (3.81)$$

para qualquer valor de taxa de bits R . Isso equivale a determinar a melhor maneira de alocar os NR bits disponíveis para codificar os N coeficientes da transformada, de modo que o MSE seja minimizado, ou seja, é preciso determinar uma N -upla $(R_0, R_1, \dots, R_{N-1})$ que satisfaça (3.79) e que minimize (3.81). A solução deste problema de otimização é dada por [20]

$$R_k = R + \frac{1}{2} \log_2 \frac{\sigma_k^2}{\left[\prod_{j=0}^{N-1} \sigma_j^2 \right]^{1/N}}, \quad 0 \leq k < N. \quad (3.82)$$

Pode-se verificar ainda [20] que a alocação de bits expressa por (3.82) também implica que $\sigma_{\hat{u}k}^2$ possui o mesmo valor para todos os coeficientes, ou seja,

$$\sigma_{\hat{u}k}^2 = \epsilon_*^2 2^{-2R} \left[\prod_{j=0}^{N-1} \sigma_j^2 \right]^{1/N}, \quad 0 \leq k < N, \quad (3.83)$$

de modo que, considerando-se que a quantização é uniforme, a minimização do MSE requer que o fator de quantização seja o mesmo para todos os coeficientes da transformada.

Na prática, muitos algoritmos de compressão, como o JPEG [4], por exemplo, utilizam fatores de quantização que variam dependendo da região de frequência à qual o coeficiente corresponde. Nesses sistemas, coeficientes de alta frequência geralmente sofrem uma quantização mais “grosseira” do que coeficientes de baixa frequência, devido à diferença de percepção do SVH a diferentes regiões do espectro espacial. Essa estratégia pode melhorar a qualidade subjetiva da imagem reconstruída obtida pelo sistema de compressão e descompressão, mas não é ótima no sentido de minimizar o erro quadrático médio.

3.3.2 Ganho da Codificação por Transformada

O ganho de codificação obtido pelo uso da transformada é avaliado em comparação à modulação codificada por pulsos (PCM), para a qual a quantização e a codificação são realizadas diretamente sobre as amostras do sinal que está sendo processado. Analogamente

a (3.77), no caso de PCM, supondo-se que são utilizados R bits para codificar cada amostra do sinal, o erro quadrático médio de quantização é dado por

$$\sigma_{u,\text{PCM}}^2 \approx \epsilon_*^2 2^{-2R} \sigma_x^2, \quad (3.84)$$

onde σ_u^2 é a variância da diferença entre as amostras originais e as quantizadas, e σ_x^2 é a variância das amostras.

O ganho de codificação da transformada é definido como a razão entre o erro quadrático médio obtido pela PCM e o erro quadrático médio obtido pela codificação usando a transformada (TC). Supondo-se que a alocação de bits utilizada no segundo caso é ótima, o erro quadrático médio obtido é dado por (3.83), isto é,

$$\sigma_{\hat{u},\text{TC}}^2 \approx \epsilon_*^2 2^{-2R} \left[\prod_{j=0}^{N-1} \sigma_j^2 \right]^{1/N}, \quad (3.85)$$

onde os σ_j^2 's são as variâncias dos coeficientes da transformada. Tomando-se (3.84) e (3.85), pode-se expressar o ganho da codificação por transformada como

$$G_{\text{TC}} \triangleq \frac{\sigma_x^2}{\left[\prod_{j=0}^{N-1} \sigma_j^2 \right]^{1/N}}. \quad (3.86)$$

3.3.3 Transformada Ótima (KLT)

Da teoria de matrizes sabe-se que dada uma matriz qualquer \mathbf{R} de dimensões $N \times N$ e cujos elementos da diagonal são $R(k, k) = \sigma_k^2$, tem-se que

$$\det[\mathbf{R}] \leq \prod_{k=0}^{N-1} \sigma_k^2. \quad (3.87)$$

Com base em (3.87) é possível determinar um limitante superior para o ganho da codificação por transformada, definido em (3.86), ou seja,

$$G_{\text{TC}} \leq \frac{\sigma_x^2}{\{\det[\mathbf{R}]\}^{1/N}}, \quad (3.88)$$

onde \mathbf{R} é a matriz de autocorrelação do processo aleatório que está sendo codificado usando uma transformada de comprimento N .

Além disso, sabe-se também que, dada uma matriz ortogonal \mathbf{A} ,

$$\mathbf{R}' = \mathbf{A}^t \mathbf{R} \mathbf{A} \quad (3.89)$$

$$\det[\mathbf{R}'] = \det[\mathbf{A}^t \mathbf{R} \mathbf{A}] \quad (3.90)$$

$$= \det[\mathbf{R}] \det[\mathbf{A}^t \mathbf{A}] \quad (3.91)$$

$$= \det[\mathbf{R}]. \quad (3.92)$$

Se \mathbf{A} é a matriz da KLT do processo aleatório cuja matriz de autocorrelação é \mathbf{R} , então \mathbf{R}' é a matriz diagonal dos autovalores λ_k de \mathbf{R} , e neste caso

$$\det[\mathbf{R}'] = \prod_{k=0}^{N-1} \lambda_k. \quad (3.93)$$

Portanto em (3.88) a igualdade ocorre quando a transformada utilizada na codificação é a KLT do processo aleatório que se está codificando.

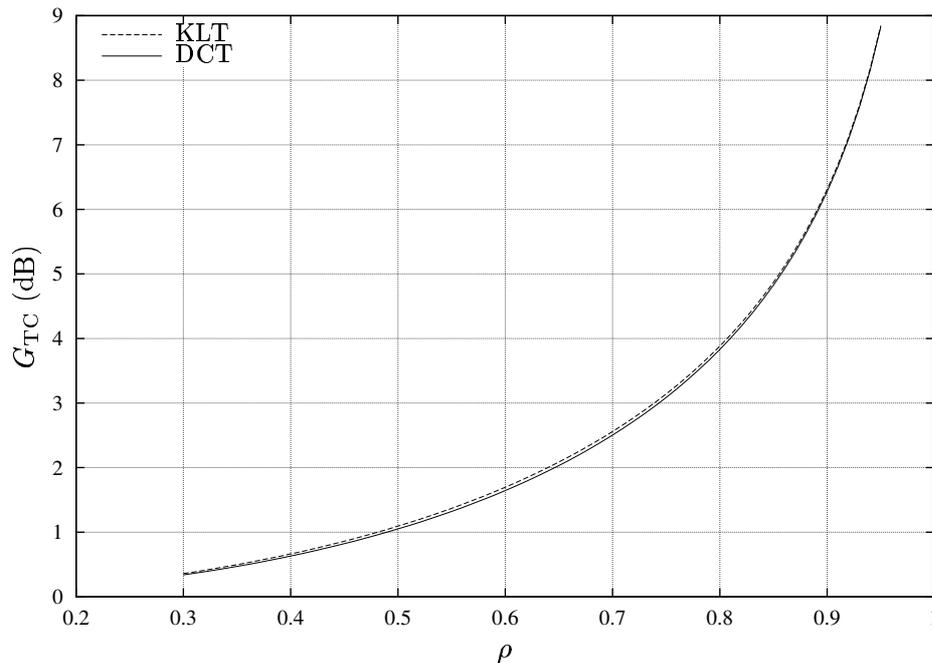


Fig. 3.7: Ganhos da codificação por transformada para processos AR de primeira ordem em função do coeficiente de correlação ρ .

A Fig. 3.7 mostra os ganhos de codificação obtidos através uso da DCT e da KLT, ambas de comprimento 8, na codificação de um processo aleatório AR de 1ª ordem com diferentes valores de autocorrelação ρ . Observe-se que o uso da DCT ao invés da KLT não representa

grandes perdas de eficiência [20].

Capítulo 4

Método Proposto

Esta dissertação propõe um novo método de compressão de imagens, baseado na transformada discreta do co-seno (DCT – *Discrete Cosine Transform*). De acordo com o método, a imagem é subdividida em blocos de 8×8 *pixels* aos quais é aplicada a DCT. Em seguida, os coeficientes da transformada de todos dos blocos da imagem são codificados progressivamente através da codificação individual de cada um dos seus planos de bits. A informação de significância dos coeficientes, presente em cada plano de bits, é codificada utilizando uma versão adaptativa do codificador de corrida de zeros de Golomb [3].

Como a codificação é aplicada a blocos de *pixels*, o método proposto pode ser usado para codificar tanto uma imagem completa como apenas algumas de suas regiões. Além disso, a quantização é realizada de forma progressiva, isto é, os bits dos coeficientes da DCT são codificados um a um, em ordem decrescente de significância (os bits mais significativos antes dos menos significativos), de modo que a qualidade da imagem reconstruída pelo decodificador é melhorada progressivamente, à medida que a decodificação é realizada. Desse modo, a qualidade da reconstrução final depende apenas do ponto até o qual a seqüência comprimida é decodificada. O controle da taxa de compressão é realizado, portanto, através da escolha do momento em que o processo de codificação deve ser interrompido.

O presente capítulo apresenta os detalhes do codificador proposto. Ele está organizado da seguinte forma: a Seção 4.1 especifica a transformada utilizada pelo método e descreve como ela é aplicada à imagem a ser comprimida; a Seção 4.2 descreve o codificador de corrida de zeros de Golomb e seu mecanismo de adaptação; a Seção 4.3 introduz o método de codificação dos coeficientes da DCT; e a Seção 4.4 descreve o algoritmo de reorganização dos coeficientes.

4.1 Transformada Discreta do Co-seno

O método de codificação proposto utiliza uma DCT bidimensional (DCT-2D) idêntica à transformada utilizada pelo padrão de compressão de imagens estáticas JPEG [4] e pelos padrões de compressão de seqüências de vídeo MPEG 1,2 e 4 [6, 7, 8]. A transformada pode ser realizada através das aplicações consecutivas, às direções horizontal e vertical, ou vice-versa, da DCT unidimensional (DCT-1D), descrita na Seção 3.2.3.

A imagem a ser comprimida é subdividida em blocos de *pixels* de tamanho 8×8 e a DCT-1D de comprimento 8 é aplicada às linhas e às colunas de cada bloco, consecutivamente, conforme ilustrado na Fig. 4.1. A transformada produz blocos de 8×8 coeficientes, que são posteriormente codificados como será mostrado a seguir.

É necessário salientar que estamos supondo que os *pixels* da imagem possuem média nula. Caso eles sejam inteiros de B bits, utilizamos a sua representação com valores entre -2^{B-1} e $2^{B-1} - 1$ (*cf.* Seção 2.1).

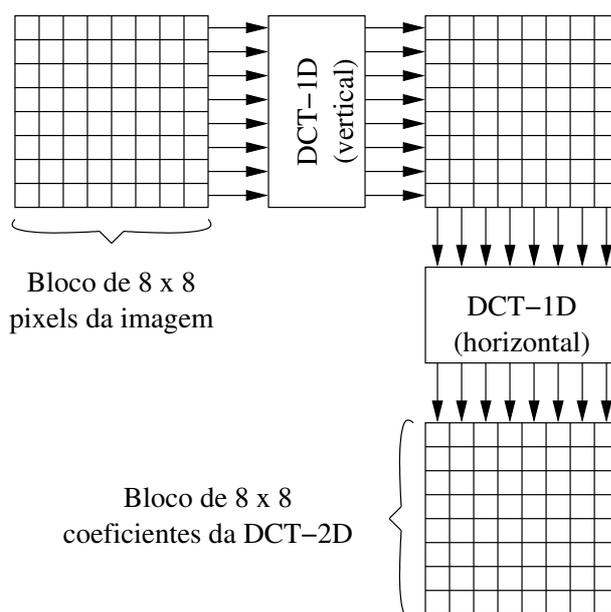


Fig. 4.1: Diagrama do transformador DCT-2D usado pelo método proposto.

4.2 Código de Corrida de Zeros de Golomb

O código de corrida zeros de Golomb é um código de entropia que pode ser usado para codificar tanto variáveis aleatórias com distribuição de probabilidade geométrica como para codificar seqüências de variáveis aleatórias de Bernoulli independentes e idênticamente distribuídas (i.i.d.).

Uma variável aleatória (v.a.) X com distribuição de probabilidade **geométrica** é uma v.a. que assume apenas valores inteiros não-negativos com probabilidade

$$\text{Prob}\{X = k\} = (1 - p)p^k, \quad k \geq 0. \quad (4.1)$$

Uma v.a. de **Bernoulli** possui apenas dois possíveis resultados: “fracasso” (‘0’) e “sucesso” (‘1’), com probabilidades p e $1 - p$, respectivamente. Variáveis aleatórias com distribuição geométrica podem ser geradas através da contagem do número de “fracassos” consecutivos ocorridos antes da ocorrência de um “sucesso” em uma seqüência de variáveis de Bernoulli independentes e idênticamente distribuídas (i.i.d.). Observa-se então que existe um mapeamento biunívoco entre seqüências de v.a.’s de Bernoulli e seqüências de variáveis com distribuição geométrica, de modo que o código de Golomb pode ser utilizado para codificar seqüências de qualquer um dos dois tipos de variáveis.

Seqüências de resultados de v.a.’s Bernoulli são representadas por seqüências de zeros e uns, de modo que a codificação de seqüências de Bernoulli usando o codificador de Golomb corresponde a codificar as corridas de zeros que ocorrem entre dois uns consecutivos dentro dessas seqüências.

4.2.1 Método de Codificação

O principal parâmetro do codificador de Golomb é denominado **comprimento do codificador**, que é representado por número inteiro não negativo ℓ . Esse parâmetro é ajustado de acordo com a distribuição de probabilidade da variável que está sendo codificada.

De acordo com o método, o resultado k de uma v.a. geométrica X é representado por uma palavra-código formada por três partes:

Parte (i) – uma seqüência de $\lfloor k/\ell \rfloor$ zeros (dígitos binários) consecutivos¹;

Parte (ii) – um bit ‘1’, para indicar o fim da primeira parte; e

Parte (iii) – uma palavra-código que representa o resto da divisão k/ℓ .

¹ $\lfloor x \rfloor$ representa a parte inteira de x .

A parte (iii) é determinada utilizando-se um código de Huffman correspondente à v.a. Y definida por

$$Y \triangleq X \bmod \ell, \quad (4.2)$$

cuja distribuição de probabilidade é representada por

$$\text{Prob}\{Y = r\} = \begin{cases} \frac{(1-p)p^r}{1-p^\ell}, & \text{se } 0 \leq r < \ell \\ 0, & \text{se } r < 0 \text{ ou } r \geq \ell. \end{cases} \quad (4.3)$$

Em [3], Gallager e van Voorhis demonstram que este codificador é ótimo se o seu comprimento ℓ satisfaz

$$p^\ell + p^{\ell+1} \leq 1 < p^\ell + p^{\ell-1}. \quad (4.4)$$

Além disso, é mostrado também que, para ℓ satisfazendo (4.4), o código de Huffman da v.a. Y possui apenas palavras-código de dois comprimentos distintos: b e $b-1$, em que 2^b é a menor potência de 2 maior ou igual a ℓ , ou seja, $b = \lceil \log_2 \ell \rceil$ ².

Para determinar o número de palavras-código com $b-1$ bits a serem utilizadas, vamos supor inicialmente que utilizaremos um código apenas com palavras de b bits. Observamos que, nesse caso, existe um total de 2^b palavras-código, das quais apenas ℓ serão necessárias, restando $2^b - \ell$. Se as palavras forem escolhidas de tal modo que as $2^b - \ell$ palavras restantes (não utilizadas) sejam diferentes apenas no valor do último bit de um conjunto Γ de palavras utilizadas pelo código (pode-se, por exemplo, selecionar primeiro as palavras terminadas em '0' e depois as palavras terminadas em '1'), observa-se então que não há necessidade de se enviar o último bit das palavras em Γ , pois ele apenas serve para distingüi-las de palavras que não pertencem ao código. Portanto, conclui-se que devem ser utilizadas $2^b - \ell$ palavras-código de comprimento $b-1$ e $2\ell - 2^b$ palavras-código de comprimento b . Observando-se que os menores valores de Y são os mais prováveis, concluimos que, se $Y < 2^b - \ell$, Y será codificada por uma palavra de $b-1$ bits, caso contrário, uma palavra de b bits será utilizada. Portanto, um método simples para se codificar Y é o seguinte:

Se $Y < 2^b - \ell$, a sua palavra-código corresponde à representação binária de $b-1$ bits de Y ; caso contrário, o resultado de Y é codificado pela representação binária de b bits do inteiro $Y + 2^b - \ell$.

Por exemplo, supondo-se que $\ell = 10$, as palavras do código de Y devem possuir 4 ou 3 bits de comprimento. Escolhendo-se para o código as palavras não hachuradas da Fig. 4.2, observa-se que as 6 primeiras palavras diferem das palavras não escolhidas (hachuradas)

² $\lceil x \rceil$ representa o menor inteiro maior ou igual a x .

apenas no último bit, de modo que este não necessita ser transmitido. Portanto, se $Y < 6$, seu resultado será codificado usando apenas 3 bits, caso contrário, serão necessários 4 bits. Além disso, se $Y < 6$, a sua palavra-código corresponde a sua representação binária com 3 dígitos, e se $Y \geq 6$, a sua palavra-código corresponde a representação binária de $Y + 6$ com 4 dígitos.

Y	→	Código
0	→	0 0 0 0
		0 0 0 1
1	→	0 0 1 0
		0 0 1 1
2	→	0 1 0 0
		0 1 0 1
3	→	0 1 1 0
		0 1 1 1
4	→	1 0 0 0
		1 0 0 1
5	→	1 0 1 0
		1 0 1 1
6	→	1 1 0 0
7	→	1 1 0 1
8	→	1 1 1 0
9	→	1 1 1 1

Fig. 4.2: Parte (iii) do código de Golomb de comprimento 10.

Observe-se que o código proposto não possui prefixos, isto é, não possui palavras-código que são prefixos de outras palavras-código, pois as palavras-código de b bits utilizadas são inteiros maiores ou iguais a $2 \cdot (2^b - \ell)$, cujos prefixos de $b - 1$ bits são inteiros maiores ou iguais a $2^b - \ell$. Por outro lado, o maior inteiro de $b - 1$ bits utilizado como palavra-código é $2^b - \ell - 1$.

Se, por exemplo, deseja-se codificar uma corrida de 23 zeros usando um código de comprimento 10, a palavra-código utilizada é da forma 001011, onde os primeiros 2 zeros correspondem a $\lfloor 23/10 \rfloor = 2$, o 1 seguinte indica o fim da seqüência de zeros inicial e a palavra-código 011 indica o resto da divisão, que é igual a 3.

A codificação de Golomb pode ser realizada através do Algoritmo 4.1, no qual x é o resultado da variável aleatória geométrica que se deseja codificar, e ℓ é o comprimento do codificador.

Algoritmo 4.1 Codificação de Golomb para variáveis aleatórias geométricas.

$x \leftarrow$ resultado da v.a. geométrica a ser codificado;
 $\ell \leftarrow$ comprimento do codificador.
 calcule quociente q e resto r da divisão x/ℓ
 envie q zeros para a saída $\{parte(i)\}$
 envie um '1' para a saída $\{parte(ii)\}$
 calcule $b \leftarrow \lceil \log_2 \ell \rceil$
 calcule $t \leftarrow 2^b - \ell$
se $r < t$ **então**
 envie os $b - 1$ bits menos significativos de r , iniciando a partir do mais significativo $\{parte(iii)\}$
senão
 calcule $s \leftarrow r + t$
 envie os b bits menos significativos de s , iniciando a partir do mais significativo $\{parte(iii)\}$
fim-se

4.2.2 Adaptatividade

No método de codificação proposto o codificador de Golomb é adaptativo, ou seja, ele ajusta o seu comprimento ℓ de acordo com uma estimativa da distribuição de probabilidade da fonte. Essa estimativa é baseada no valor esperado da seqüência de v.a.'s geométricas que está sendo codificada. A relação entre o valor esperado e o comprimento do codificador é descrita abaixo.

Relação entre o comprimento do codificador e o valor esperado

O comprimento do codificador está relacionado ao parâmetro p da distribuição de probabilidade através das desigualdades (4.4), dadas por

$$p^\ell + p^{\ell+1} \leq 1 < p^\ell + p^{\ell-1}. \quad (4.5)$$

Observe-se que apenas um número inteiro satisfaz (4.5). Visto que o número real ℓ_R , definido por

$$\ell_R \triangleq \frac{1}{\log_2 \frac{1}{p}}, \quad (4.6)$$

também satisfaz a desigualdade acima, vamos expressar ℓ por

$$\ell = \ell_R + \gamma(\ell_R), \quad (4.7)$$

onde $|\gamma(\ell_R)|$ não é maior do que 1. A Fig. 4.3 mostra ℓ e ℓ_R em função de p variando entre 0,5 e 0,95. A Fig. 4.4 mostra os valores de $\gamma(\ell_R)$ em função de ℓ_R .

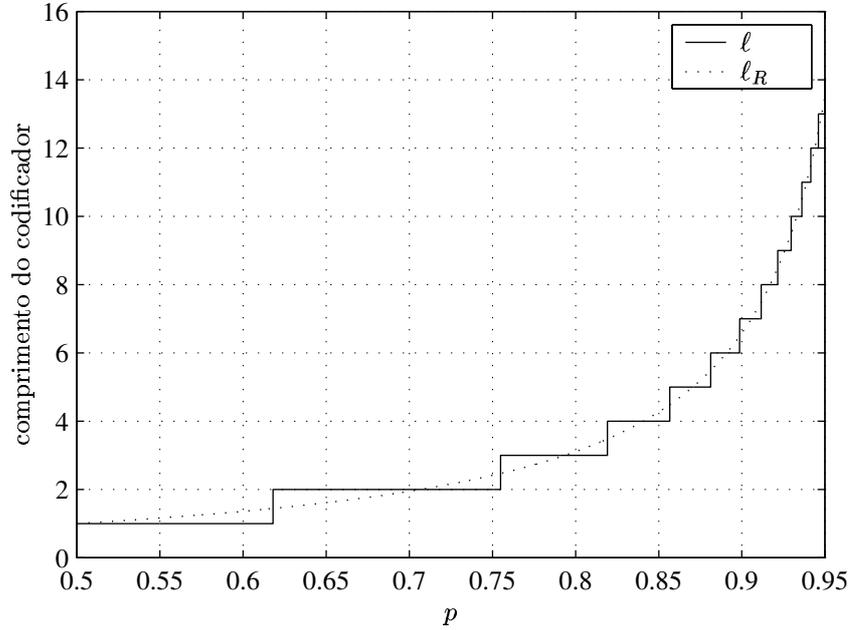


Fig. 4.3: Comprimentos ℓ e ℓ_R do codificador de Golomb em função do parâmetro p .

Por outro lado, o valor esperado de uma variável aleatória geométrica é dado por

$$\bar{X} = \frac{p}{1-p}, \quad (4.8)$$

de modo que p pode ser escrito como

$$p = \frac{\bar{X}}{\bar{X} + 1}. \quad (4.9)$$

Dessa forma, o comprimento do codificador está relacionado ao valor esperado da fonte através de

$$\ell = \frac{1}{\log_2 \left[1 + \frac{1}{\bar{X}} \right]} + \gamma(p). \quad (4.10)$$

O logaritmo acima pode ser escrito em forma de série como

$$\log_2 \left[1 + \frac{1}{\bar{X}} \right] = \frac{1}{\ln 2} \left(\frac{1}{\bar{X}} - \frac{1}{2\bar{X}^2} + \frac{1}{3\bar{X}^3} - \dots \right), \quad (4.11)$$

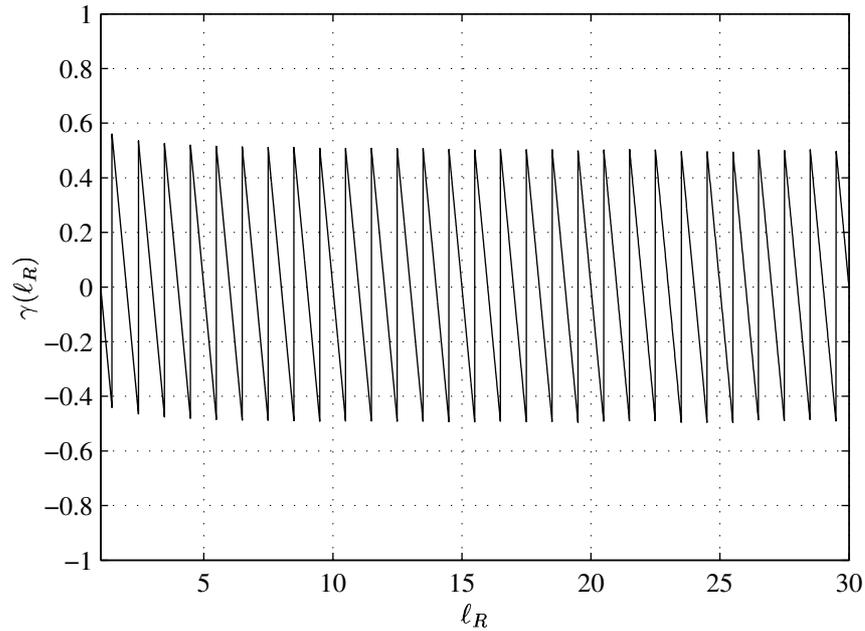


Fig. 4.4: Diferença entre o comprimento do codificador de Golomb ℓ e o valor aproximado ℓ_R em função de ℓ_R .

e, portanto, pode-se aproximar ℓ_R em função de \bar{X} por

$$\ell_R \approx \bar{X} \ln 2, \quad (4.12)$$

se p tiver um valor próximo de 1, de modo que \bar{X} seja suficientemente grande. O comprimento do codificador pode, portanto, ser ajustado através de um arredondamento para o inteiro mais próximo, isto é,

$$\ell = \lfloor \bar{X} \ln 2 + 0,5 \rfloor. \quad (4.13)$$

Algoritmo de adaptação do comprimento do codificador/decodificador

O método de compressão proposto ajusta o comprimento do codificador de Golomb em duas ocasiões:

1. quando o codificador (decodificador) envia (recebe) um '0' pertencente à parte (i) (*cf.* Seção 4.2.1) de uma palavra do código de Golomb;
2. e quando o codificador (decodificador) envia (recebe) o final da palavra-código, ou seja, quando a transmissão de um símbolo da fonte é completada.

Na primeira ocasião, o ajuste do comprimento do codificador (decodificador) $\ell(n)$ é realizado acrescentando-se $\lfloor (\ell(n) + 1)/2 \rfloor$ ao valor atual do comprimento, de modo que o novo comprimento seja

$$\ell(n + 1) = \ell(n) + \left\lfloor \frac{\ell(n) + 1}{2} \right\rfloor. \quad (4.14)$$

Na segunda ocasião, uma estimativa \bar{X} do valor médio dos símbolos codificados ou decodificados é calculada e o comprimento do codificador e do decodificador são reajustados em função dessa estimativa. Levando-se em conta os objetivos de reduzir a complexidade e, por conseguinte, aumentar a velocidade do codificador e do decodificador, consideramos a aproximação $\ln 2 \approx 0,5$. Assim, tomando-se como ponto de partida a equação (4.13), o cálculo do novo comprimento do codificador (decodificador) é realizado utilizando apenas operações com números inteiros. Dessa forma, o novo comprimento é dado por

$$\ell(n + 1) = \left\lfloor \frac{\bar{X}(n + 1) + 1}{2} \right\rfloor. \quad (4.15)$$

A nova estimativa $\bar{X}(n + 1)$ do valor esperado \bar{X} é calculada com base em uma média ponderada entre a estimativa anterior $\bar{X}(n)$ e o último símbolo codificado (decodificado) $X(n)$, ou seja,

$$\bar{X}(n + 1) = \alpha \bar{X}(n) + (1 - \alpha)X(n). \quad (4.16)$$

α é um fator de esquecimento escolhido arbitrariamente e que geralmente possui valor próximo a 1.

O Algoritmo 4.2 é a versão adaptativa do algoritmo de codificação de corridas de zeros descrito anteriormente.

4.3 Codificação dos Coeficientes da Transformada

Após o cálculo da transformada, os coeficientes da DCT de todos os blocos da imagem são reorganizados de modo a formarem uma única seqüência unidimensional \mathbf{v} . Esta reorganização é descrita na Seção 4.4. Nessa seqüência cada coeficiente será representado na forma de um sinal³ mais um valor absoluto em binário. No processo de codificação cada *plano de bits* dos valores absolutos dos coeficientes será codificado a cada iteração do algoritmo, iniciando-se pelo *plano de bits* mais significativo.

Definimos a função $B_n(x)$, que representa o n -ésimo bit do valor absoluto de um coefi-

³Positivo ou negativo.

Algoritmo 4.2 Codificação de Golomb adaptativa.

$x \leftarrow$ resultado da v.a. geométrica a ser codificado;
 $\ell \leftarrow$ comprimento do codificador;
 $\bar{x} \leftarrow$ estimativa do valor médio dos símbolos anteriores.
 faça $r \leftarrow x$
enquanto $r > \ell$ **faça**
 envie um '0' para a saída
 calcule $r \leftarrow r - \ell$
 calcule $\ell \leftarrow \ell + \lfloor \ell + 1/2 \rfloor$ {*Primeira parte adaptativa.*}
fim-enquanto
 envie um '1' para a saída
 calcule $b \leftarrow \lceil \log_2 \ell \rceil$
 calcule $t \leftarrow 2^b - \ell$
se $r < t$ **então**
 envie os $b - 1$ bits menos significativos de r para a saída, iniciando a partir do mais significativo
senão
 calcule $r \leftarrow r + t$
 envie os b bits menos significativos de r para a saída, iniciando a partir do mais significativo
fim-se
 calcule $\bar{x} \leftarrow \alpha \bar{x} + (1 - \alpha)x$ {*Nova estimativa do valor médio*}
 calcule $\ell \leftarrow \bar{x} + 1/2$ {*Segunda parte adaptativa.*}

ente x , como

$$B_n(x) = \begin{cases} 1, & \text{se } (2k + 1) \cdot 2^n \leq |x| < (2k + 2) \cdot 2^n, \quad k \in \mathbb{Z}_+, \\ 0, & \text{caso contrário.} \end{cases} \quad (4.17)$$

Por extensão, podemos definir $B_n(\mathbf{v})$ como sendo a seqüência formada pelos n -ésimos bits dos coeficientes de \mathbf{v} , ou seja, o n -ésimo **plano de bits** de \mathbf{v} . Considerando que $B_n(\mathbf{v})$ é uma seqüência de variáveis de Bernoulli independentes, ela pode ser representada de maneira completa e inequívoca por uma seqüência de variáveis aleatórias geométricas, em que cada uma dessas variáveis representa uma corrida de zeros em $B_n(\mathbf{v})$.

A seqüência \mathbf{v} será denominada de seqüência de coeficientes **não significativos**. Ela possui inicialmente $64 \cdot N_B$ coeficientes, sendo N_B o número total de blocos da imagem que estão sendo codificados.

Os planos de bits de \mathbf{v} são codificados usando o codificador de corrida de zeros de Golomb, descrito na Seção 4.2, com a seguinte ressalva: quando um bit não nulo de um coeficiente em \mathbf{v} é transmitido, o sinal do coeficiente é transmitido em seguida, aproveitando-se que

este é exatamente o momento em que a transmissão de uma palavra-código de Golomb foi completada.

É importante também que o codificador e o decodificador de corridas de zeros levem em conta o fato de que as seqüências binárias $B_n(\mathbf{v})$ são finitas, ou seja, que as corridas de zeros não podem ser mais longas que um determinado valor. Esse valor máximo corresponde exatamente ao número de bits de $B_n(\mathbf{v})$ que ainda não foram processados. Se o valor máximo for levado em conta durante a codificação, é possível evitar o desperdício decorrente da transmissão desnecessária dos bits que representam o resto da divisão no código de Golomb. Por exemplo, supondo-se que a última corrida de zeros a ser codificada apresenta um resto igual a $k < \ell$, se o codificador não levar em conta o fato de que o maior resto possível é k , ele utilizará entre b e $b + 1$ bits para codificá-lo, onde $b = \lceil \log_2 \ell \rceil$. No entanto, essa informação está disponível tanto para o codificador quanto para o decodificador, de modo que o codificador não precisa codificar o resto da maneira usual.

À medida que cada plano de bits de \mathbf{v} é codificado, os coeficientes cujos bits processados não são nulos, eles são transferidos para uma seqüência \mathbf{s} , dos coeficientes **significativos**, que está inicialmente vazia. Os planos de bits de \mathbf{s} não contêm o primeiro bit significativo não nulo de nenhum coeficiente, mas apenas os bits de menor significância. Como pode ser verificado, esses bits podem ser 0 ou 1 com aproximadamente a mesma probabilidade, isto é, 0,5. Por isso eles são transmitidos sem qualquer codificação.

A disposição dos coeficientes em \mathbf{s} é a mesma que inicialmente foi utilizada na geração da seqüência \mathbf{v} , e não depende da ordem em que os coeficientes são transferidos para essa outra seqüência. Essa disposição é mantida a fim de refinar os coeficientes na mesma ordem em que os primeiros bits significativos foram inicialmente codificados, isto é, dando prioridade aos coeficientes que estão no início da seqüência (*cf.* Seção 4.4).

Portanto, a codificação dos coeficientes ocorre da seguinte maneira: depois dos coeficientes da DCT de todos os blocos da imagem terem sido reorganizados em uma única seqüência unidimensional \mathbf{v} , é determinado o bit mais significativo não nulo dentre todos os coeficientes, isto é,

$$n_{\max} = \left\lceil \log_2 \left(\max_{0 \leq k < N_{\mathbf{v}}} |v[k]| \right) \right\rceil, \quad (4.18)$$

onde $N_{\mathbf{v}}$ é o número inicial de coeficientes em \mathbf{v} ($64 \cdot N_B$, como visto anteriormente). Iniciando-se por $n = n_{\max}$, os n -ésimos planos de bits de \mathbf{v} e \mathbf{s} são codificados consecutivamente, e ao fim desse processo, os coeficientes de \mathbf{v} cujos bits processados não são nulos, eles são transferidos para \mathbf{s} . Então n é decrementado e a codificação dos planos de bits de \mathbf{v} e \mathbf{s} é novamente realizada. Observe-se que, no método proposto, antes do início da codificação de cada plano de bits, o comprimento do codificador de Golomb e a estimativa

do valor médio da fonte são inicializados com o valor 1. A codificação dos planos de bits é realizada para cada valor de n até que a taxa de bits desejada seja atingida. O Algoritmo 4.3 apresenta o processo de codificação.

Algoritmo 4.3 Codificação dos coeficientes da DCT.

$\mathbf{v} \leftarrow$ vetor com os coeficientes da DCT
 $\mathbf{s} \leftarrow$ vetor vazio {*coeficientes significativos.*}
 calcule n_{\max} conforme (4.18)
 envie n_{\max} para a saída
 $n \leftarrow n_{\max}$
enquanto não for atingida a taxa de bits desejada **faça**
 inicialize o codificador de Golomb com $\ell \leftarrow 1$ e $\bar{X} \leftarrow 1$
 codifique $B_n(\mathbf{v})$ usando o codificador de Golomb adaptativo
 envie $B_n(\mathbf{s})$ para a saída sem codificação
 transfira de \mathbf{v} para \mathbf{s} os coeficientes para os quais $B_n(v[k]) = 1$.
 faça $n \leftarrow n - 1$ {*próximo plano de bits a ser processado.*}
fim-enquanto

4.3.1 Propagação de Erros

Pode-se observar que o método de codificação proposto ainda apresenta um benefício colateral contra os erros no canal de transmissão. Tal benefício resulta do modo como os coeficientes da transformada são codificados.

Como os bits de refinamento, isto é, os bits do vetor \mathbf{s} , são transmitidos sem codificação, os erros que por ventura venham a ocorrer durante a sua transmissão não se propagam para o resto da seqüência binária. Como conseqüência, os sintomas desses erros na imagem reconstruída representam apenas alterações de magnitude nos pixels que dependem do coeficiente afetado. E essas alterações são da ordem da magnitude correspondente ao bit errôneo.

Já os erros que afetam os bits produzidos pelo codificador de Golomb, podem gerar perda de sincronismo entre codificador e decodificador e, assim, produzir defeitos mais graves na imagem reconstruída.

Portanto, um esquema de proteção contra erros para a seqüência comprimida poderia levar em conta estas características e proporcionar níveis de proteção distintos, de acordo com a sensibilidade a erros de cada conjunto de bits.

4.4 Reorganização dos Coeficientes da Transformada

Como visto na Seção 4.3, antes de ser realizada a codificação dos coeficientes da DCT, é necessário reorganizá-los de modo que eles formem uma única seqüência unidimensional \mathbf{v} . Um dos objetivos dessa reorganização deve ser o de colocar os coeficientes da transformada em uma ordem tal que a probabilidade de que os valores absolutos dos coeficientes sejam maiores do que um dado limiar decresça à medida que essa seqüência de coeficientes é percorrida. Desse modo, o comprimento do codificador de Golomb adaptativo utilizado deve aumentar à medida que os planos de bits dos coeficientes são codificados.

Por outro lado, essa reorganização deve procurar manter próximos (uns dos outros) os coeficientes que eventualmente possuam alta probabilidade de possuir grandes valores absolutos. Essa característica é importante para que os comprimentos das corridas de zeros não varie demasiadamente rápido.

A variância de um coeficiente da DCT (considerando que o seu valor esperado é nulo) corresponde à esperança da sua energia, isto é, do quadrado do seu valor absoluto. Portanto, quanto maior for a sua variância, maior será a probabilidade de que o coeficiente possua valor absoluto maior do que um limiar estipulado. Como foi discutido na Seção 3.2, a energia de um coeficiente da DCT corresponde à energia que o sinal transformado possui na região de freqüência ocupada pela função de base que corresponde a esse coeficiente. Nos sinais de imagem, considerando-se o modelo AR de primeira ordem visto na Seção 3.1.6, a densidade espectral de potência decresce à medida que a freqüência aumenta. Portanto, a variância dos coeficientes da DCT geralmente decresce à medida que as freqüências correspondentes às suas funções de base aumentam. Os coeficientes que correspondem às freqüências mais altas, geralmente possuem valores absolutos menores.

Dessa forma, conclui-se que o algoritmo de reorganização deve procurar colocar os coeficientes de baixa freqüência antes dos coeficientes de alta freqüência.

Além desse fato, outro fenômeno importante ocorre com os sinais de imagem. Existem diversos tipos de “anomalias”, como, por exemplo, bordas de objetos e regiões de textura, que não são adequadamente modeladas por processos aleatórios AR de primeira ordem. Essas “anomalias” geralmente apresentam componentes de alta freqüência com energia razoável. A distribuição de energia entre os coeficientes dos blocos da imagem que contêm essas “anomalias” é geralmente mais plana ou próxima de uniforme, isto é, todos os coeficientes possuem mais ou menos a mesma energia. Desse modo, é de se esperar que quando forem encontrados coeficientes de alta freqüência com valores absolutos elevados, eles geralmente estarão próximos de outros coeficientes de valor absoluto também elevado. Assim sendo, raramente um coeficiente de alta freqüência que possui um valor absoluto alto será encontrado isolado,

sem que haja outros coeficientes também com valores absolutos elevados nas suas vizinhanças. Por esse motivo, o algoritmo de reorganização deve procurar manter coeficientes de alta frequência do mesmo bloco, ou de blocos vizinhos, próximos entre si.

1.1	2.1	5.1	5.2	8.1	8.2	8.5	8.6
3.1	4.1	5.3	5.4	8.3	8.4	8.7	8.8
6.1	6.2	7.1	7.2	8.9	8.10	8.13	8.14
6.3	6.4	7.3	7.4	8.11	8.12	8.15	8.16
9.1	9.2	9.5	9.6	10.1	10.2	10.5	10.6
9.3	9.4	9.7	9.8	10.3	10.4	10.7	10.8
9.9	9.10	9.13	9.14	10.9	10.10	10.13	10.14
9.11	9.12	9.15	9.16	10.11	10.12	10.15	10.16

Fig. 4.5: Numeração dos coeficientes da DCT em um bloco.

Com base nos princípios descritos acima, um algoritmo de ordenação dos coeficientes foi proposto. Em primeiro lugar, os blocos de coeficientes são divididos em sub-blocos e numerados de acordo com a Fig. 4.5. Esses sub-blocos possuem tamanhos diferentes e são constituídos de 1 até 16 coeficientes cada um. Essa divisão é inspirada na forma como os coeficientes gerados pela transformada de wavelets [17] são geralmente organizados.

Como pode ser observado (*cf.* Fig. 4.5), cada bloco de coeficientes é dividido em 10 sub-blocos, numerados de 1 a 10 em ordem crescente de frequência. A numeração de cada coeficiente é representada por $m.n$, onde m indica o número do sub-bloco ao qual pertence o coeficiente, e n indica o número do coeficiente dentro desse bloco. Os coeficientes da DCT dos N_B blocos de coeficientes da imagem, ou da região da imagem, que serão codificados são reorganizados de modo a formarem uma única seqüência unidimensional \mathbf{v} de acordo com o Algoritmo 4.4, mostrado a seguir. Como resultado desse algoritmo, à medida que a frequência aumenta, mais coeficientes vizinhos ficam próximos entre si na seqüência final.

Como a codificação é realizada através da varredura da seqüência de coeficientes, outra característica do sistema de compressão é que os coeficientes de baixa frequência são codificados e refinados a cada iteração do codificador antes dos coeficientes de alta frequência. Portanto, como o algoritmo de compressão pode parar a qualquer momento, aqueles recebem

Algoritmo 4.4 Reorganização dos coeficientes da DCT

crie uma seqüência de coeficientes vazia

para m de 1 até 10 **faça**

para k de 1 até N_B **faça**

para n de 1 até o número de coeficientes do m -ésimo sub-bloco **faça**

 adicione ao fim de \mathbf{v} o coeficiente $m.n$ do bloco k

fim-para

fim-para

fim-para

prioridade em relação a estes.

Capítulo 5

Resultados Experimentais

Neste capítulo descrevemos os resultados experimentais obtidos pelo método de compressão de imagens proposto no Capítulo 4. Os resultados são comparados aos resultados de outros métodos, dentre eles o padrão JPEG [4], que também faz uso da DCT. Em alguns casos, é feita a comparação com métodos baseados em transformadas de wavelets, como o EZW (*embedded zero-tree wavelet coder* [11]) e o SPIHT (*set partitioning in hierarchical trees*) [10].

5.1 Metodologia

Os desempenhos dos métodos de compressão são avaliados quanto à razão de sinal máximo por erro quadrático médio (PSNR), apresentada na Seção 2.4. Os valores de PSNR são apresentados em decibéis (dB) e calculados através da expressão:

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{MSE}}, \quad (5.1)$$

onde o erro quadrático médio (MSE) é calculado a partir da média empírica (*i.e.*, a média observada experimentalmente) do quadrado da diferença entre os *pixels* da imagem original e os *pixels* da imagem descomprimida, isto é,

$$\text{MSE} = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (x[n_1, n_2] - y[n_1, n_2])^2, \quad (5.2)$$

onde $\{x[n_1, n_2]\}$ é a imagem original e $\{y[n_1, n_2]\}$ é a imagem gerada pelo processo de decodificação.

Os resultados para o método proposto foram obtidos a partir da implementação dos

algoritmos de compressão e descompressão descritos no Capítulo 4 para imagens em escala de cinzas (com apenas um componente referente à luminância) com *pixels* representados por inteiros de 8 bits. A implementação foi realizada em linguagem C/C++ padrão e foi compilada e utilizada em sistemas Sun/Solaris, PC/Linux e PC/Cignus Windows¹.

Os resultados para o método de compressão de JPEG foram obtidos a partir da implementação desenvolvida pelo *Independent JPEG Group* (IJG), que disponibiliza gratuitamente através da internet uma biblioteca de compressão de imagens baseada nesse padrão. Essa biblioteca pode ser obtida através do endereço da internet mostrado na referência [23].

Nos processos de compressão e descompressão JPEG foram utilizadas tabelas de quantização uniformes de valores arbitrários, escolhidos de modo a se obter as taxas de bits desejada. As tabelas de quantização uniforme apresentam o mesmo fator de quantização para todos os coeficientes da DCT. Essa opção foi adotada devido ao fato dos resultados estarem sendo comparados usando-se o critério de PSNR, para o qual o desempenho ótimo é obtido com a quantização uniforme, como foi discutido na Seção 3.3.1. Além disso, foi utilizado um código de entropia otimizado para a estatística particular de cada imagem, e que também é mais adequado à estatística dos coeficientes quantizados uniformemente.

Os resultados dos codificadores SPIHT e EZW foram extraídos diretamente de [10].

5.2 Resultados

A Fig. 5.5 mostra as imagens que foram utilizadas para obter os resultados experimentais. Todas as imagens utilizadas são imagens em níveis de cinza com resolução de 512×512 *pixels* e com 8 bits por *pixel*. Os resultados para Lena, Goldhill, Barbara e Boat são mostrados nas Figuras 5.1, 5.2, 5.3 e 5.4, respectivamente.

As Figs. 5.6 e 5.7 mostram os resultados visuais obtidos pelo sistema proposto para a imagem Lena e Goldhill. São mostradas as imagens reconstruídas utilizando 0,25 bpp (bits por *pixel*), 0,5 bpp e 1 bpp. Vale salientar que a qualidade dos dispositivos de impressão disponíveis não permite uma melhor reprodução desses resultados, o que dificulta demonstrar a sua qualidade subjetiva.

Observa-se que, para todas as imagens, o esquema de compressão proposto apresentou resultados similares ou superiores aos resultados do JPEG. Ganhos de eficiência foram verificados principalmente a taxas de bits maiores, como acima de 0,8 bpp para a imagem Lena, e 0,7 para as imagens Goldhill e Boat. Observa-se também que para as imagens Lena, Goldhill e Boat, o ganho de eficiência do esquema proposto tende a aumentar em relação ao

¹Conjunto de ferramentas Linux para sistemas MS Windows.

JPEG à medida que as taxas aumentam. Portanto, o esquema proposto representa uma real alternativa ao método de compressão JPEG. Principalmente no que diz respeito a sua utilização em sistemas de compressão de vídeo, onde esquemas essencialmente iguais ao JPEG² podem ser usados para comprimir os quadros *intra*³ e os resíduos dos quadros *inter*⁴. Como os resultados do método proposto são superiores aos do JPEG para taxas de compressão maiores, pode-se afirmar que a sua aplicação na compressão de vídeo é recomendada nos casos em que uma melhor qualidade da imagem é exigida, como por exemplo, na compressão de vídeo para TV digital de alta definição.

Além da comparação objetiva entre os resultados obtidos pelo sistema proposto e pelo método do JPEG, também vale ressaltar que o sistema proposto apresenta complexidade razoavelmente inferior a este, já que não envolve buscas em tabelas de código de Huffman.

Na comparação com os sistemas baseados em transformadas de wavelets, verifica-se que a eficiência do sistema proposto ainda está bem aquém dos resultados obtidos por esses algoritmos, principalmente para baixas taxas, como em torno de 0,25 bpp. Isso decorre, talvez, da maior eficiência das transformadas de wavelets ao tratar a imagem como um todo, sem incorrer nas discontinuidades causadas pela blocagem.

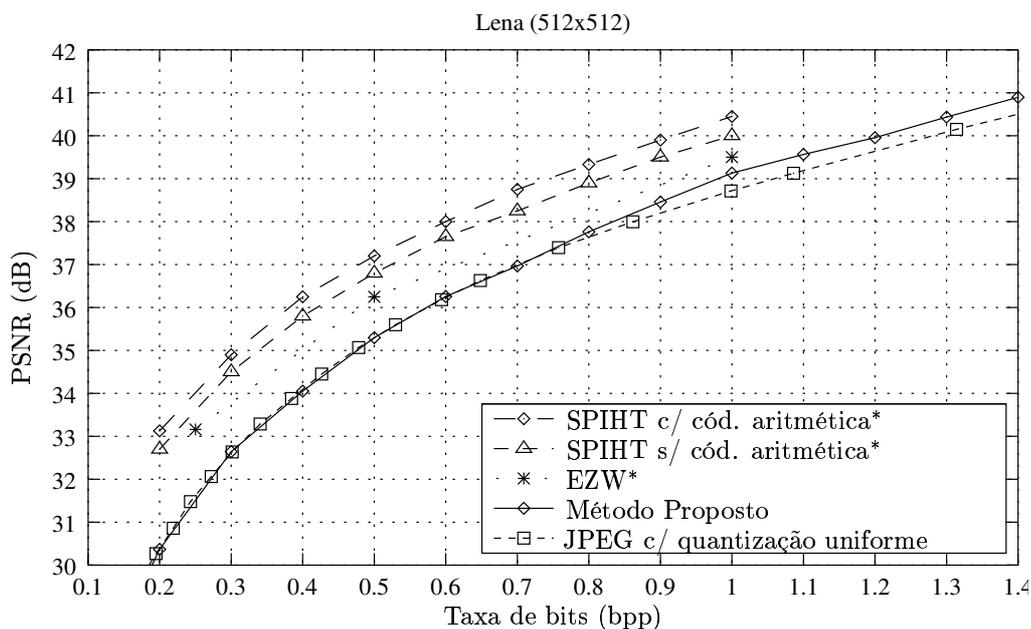


Fig. 5.1: Resultados experimentais para a imagem Lena (512 x 512)

² cf. MPEG ou Motion Picture Experts Group [24]

³ Codificados isoladamente, sem informação dos quadros anteriores ou posteriores.

⁴ Codificados através da estimação obtida a partir dos quadros vizinhos.

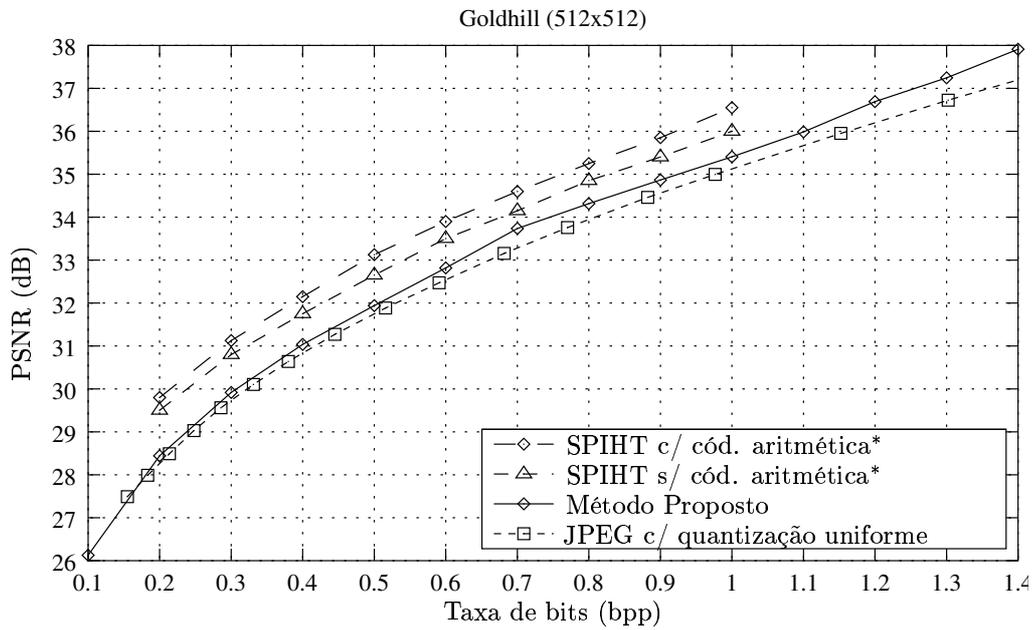


Fig. 5.2: Resultados experimentais para a imagem Goldhill (512 x 512)

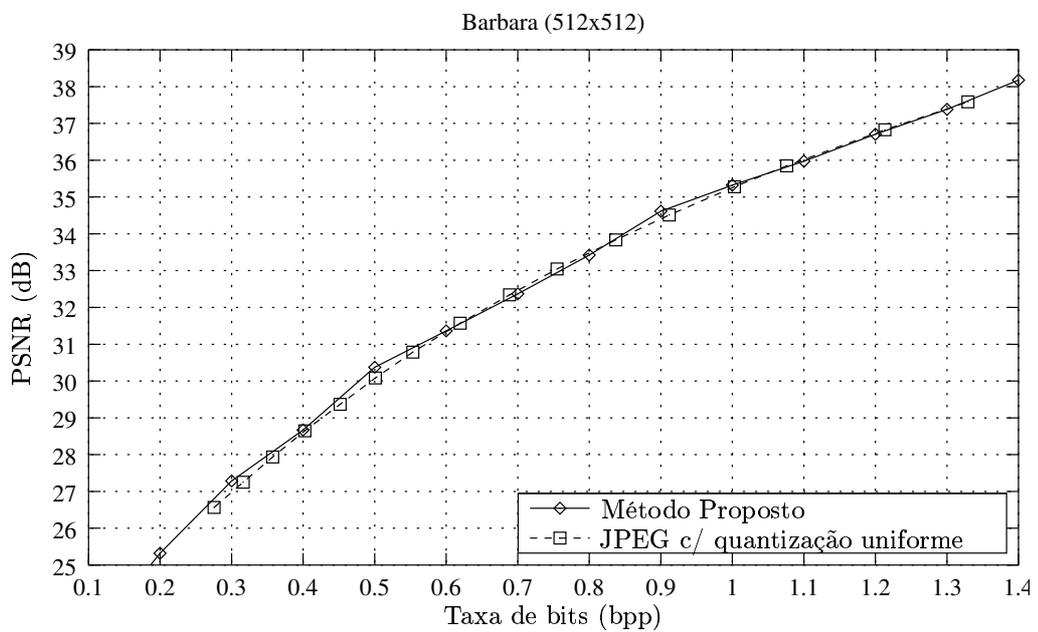


Fig. 5.3: Resultados experimentais para a imagem Bárbara (512 x 512)

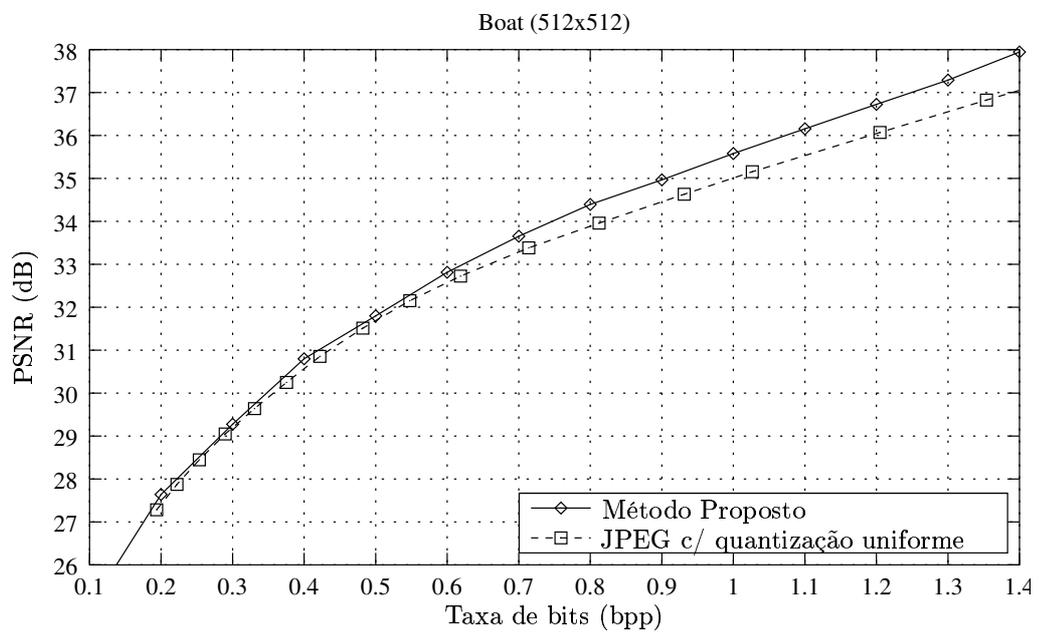


Fig. 5.4: Resultados experimentais para a imagem Boat (512 x 512)



Fig. 5.5: Imagens originais utilizadas para obter os resultados experimentais: (a) Lena, (b) Goldhill, (c) Bárbara e (d) Boat.



Fig. 5.6: Resultados visuais obtidos para a imagem Lena pelo método proposto: (a) Imagem original, (b) com 0,25 bpp, (c) com 0,5 bpp e (d) com 1 bpp.



Fig. 5.7: Resultados visuais obtidos para a imagem Goldhill com o método proposto: (a) Imagem original, (b) com 0,25 bpp, (c) com 0,5 bpp e (d) com 1 bpp.

Capítulo 6

Conclusão

Neste trabalho propõe-se um novo método de compressão digital de imagens naturais estáticas que se baseia na transformada discreta do co-seno (DCT) e na codificação progressiva dos coeficientes da transformada, utilizando o código de corrida de zeros de Golomb.

Por realizar codificação progressiva, o esquema proposto produz uma seqüência comprimida “embutida”, isto é, em que as seqüências comprimidas com menores taxas estão presentes no início das seqüências de maiores taxas. Essa característica permite que a decodificação seja gradual, isto é, pode-se escolher a qualidade da imagem reconstruída através da escolha do ponto em que a decodificação deve ser interrompida.

Por utilizar uma transformada de bloco, o esquema de codificação proposto é mais adequado à adaptação para uso em sistemas de compressão de vídeo. Isso se deve ao fato de que a divisão da imagem em blocos permite que sejam utilizadas técnicas de codificação de vídeo baseadas em estimação e compensação de movimento. Por exemplo, os padrões de compressão MPEG fazem uso desses métodos para codificação dos quadros preditos (tipo P) e interpolados (tipo B).

Como parte de um esquema de compressão de vídeo, o método proposto apresenta certas vantagens em relação ao método do JPEG, que é semelhante aos métodos utilizados pelos padrões MPEG 1 e 2 na codificação dos quadros *intra* das seqüências de vídeo. Entre essas vantagens podem-se destacar o ganho de eficiência, representado pelo maior PSNR para taxas altas (*cf.* Seção 5.2), e uma redução na complexidade, devido à não-utilização dos códigos de Huffman. O ganho de eficiência é ainda mais preponderante para altas taxas, onde o PSNR é um bom indicador da qualidade visual percebida. De modo que a utilização do esquema proposto pode ser indicada principalmente em aplicações que exigem alta qualidade da reprodução.

O codificador de corrida de zeros de Golomb, visto aqui como um codificador de seqüên-

cias binárias, é utilizado pelo método de compressão proposto na codificação das seqüências formadas pelos bits mais significativos dos coeficientes da transformada. A escolha desse codificador de entropia deve-se ao fato de que ele apresenta um bom desempenho na codificação de seqüências binárias cujas características se assemelham às das seqüências formadas pelos planos de bits dos coeficientes. Além disso, ele é bastante simples de ser implementado, se comparado a outros métodos de codificação capazes de realizar a mesma tarefa. Outra característica importante é o fato de que seu funcionamento pode ser ajustado às características da seqüência processada sem grandes esforços, o que permite o seu funcionamento adaptativo.

Portanto, o codificador e o decodificador de corrida de zeros do método proposto trabalham de maneira adaptativa, associados a um sistema de ajuste de parâmetro que leva em conta as estatísticas da seqüência binária à medida que ela é processada. Além disso, esse esquema de adaptação, que também é uma contribuição original deste trabalho, não exige que qualquer informação extra seja transmitida ao decodificador.

Quanto aos resultados experimentais, foram apresentados resultados visuais, como as imagens Lena e Goldhill (*cf.* Figs. 5.6 e 5.7) comprimidas pelo esquema proposto e reconstruídas com taxas de bits de 0,25 bpp, 0,5 bpp e 1 bpp. Também foram apresentados resultados objetivos, como as curvas de PSNR em função da taxa de bits para as imagens Lena, Goldhill, Barbara e Boat, codificadas pelos esquemas SPIHT [10] (Lena e Goldhill), EZW [11] (Lena), JPEG [4] e pelo método proposto. Nos resultados objetivos, observou-se que o esquema proposto apresenta eficiência igual ou superior ao JPEG utilizando quantização uniforme. Em alguns casos (Lena, Goldhill e Boat), o esquema proposto apresentou eficiência progressivamente maior que a do JPEG à medida que taxas de bits maiores eram utilizadas (*cf.* Figs. 5.1, 5.2 e 5.4). Em comparação aos esquemas que utilizam transformadas de wavelets, o esquema proposto apresentou eficiência inferior em todos os casos verificados, o que sugere a superioridade das transformadas de wavelets, em termos de eficiência na compressão, em relação as transformadas de bloco, como a DCT. Em compensação, os métodos que utilizam transformadas de wavelets não são tão facilmente adaptáveis à utilização em sistemas de compressão de vídeo.

Os resultados experimentais apresentados foram obtidos através da implementação do método proposto em linguagem C/C++.

6.1 Sugestões para Trabalhos Futuros

Aplicações em imagens coloridas

O método proposto poderia ser adaptado para utilização na compressão de imagens coloridas. A maneira mais direta seria a codificação em conjunto dos coeficientes resultantes da aplicação da transformada aos componentes de luminância (Y) e cromaticidade (Cr e Cb), devidamente dizimados, conforme desejado.

Ajuste preliminar às características do sistema visual humano (SVH)

O processo de quantização que está associado ao método de compressão proposto, quantiza todos os coeficientes da transformada de maneira uniforme, isto é, usando os mesmos intervalos de quantização, exceto no caso dos coeficientes que não são processados antes da interrupção durante a última iteração do algoritmo. Do ponto de vista da otimização com respeito ao MSE, este é o método ótimo de quantização. No entanto, se for levada em conta a variação de sensibilidade do SVH aos componentes de diferentes frequências, a quantização ótima do ponto de vista subjetivo não é a quantização uniforme, como a realizada pelo esquema proposto. Neste caso, um ajuste preliminar e simplificado desse esquema às características do SVH pode ser realizada através da multiplicação dos coeficientes da DCT por fatores que dependam das frequências associadas a cada um deles. Desse modo, embora a codificação realizada ainda seja progressiva, os intervalos de quantização e refinamento de cada coeficiente são variáveis, podendo ser escolhidos arbitrariamente através do fator multiplicativo.

Uma adaptação mais completa do método proposto ao SVH requer um estudo mais rigoroso sobre as suas particularidades, o que é um problema bem mais complexo.

Adaptação à codificação de seqüências de vídeo

Como discutido anteriormente, o esquema proposto pode ser adaptado para a utilização em sistemas de compressão de vídeo que utilizam compensação de movimento para tratar da redundância temporal.

Utilização na compressão de outros tipos de sinais

O método de compressão dos coeficientes da transformada também pode ser adaptado para a codificação de outros tipos de sinais, que também são usualmente codificados através de transformadas. Como exemplo, citam-se os sinais de áudio e voz.

Utilização de outras transformadas

Outras transformadas, como as transformadas de wavelets [17] e as transformadas com superposição (LT's – Lapped Transforms) [22], poderiam ser utilizadas no lugar da DCT.

Referências Bibliográficas

- [1] F. C. de A. Oliveira e M. H. M. Costa, “Embedded DCT image encoding,” in *Proc. of the IEEE International Telecommunications Symposium*, (Natal, RN, Brazil), Set. 2002.
- [2] A. V. Oppenheim e R. W. Schaffer, *Discrete-time Signal Processing*. Prentice Hall Signal Processing Series, Upper Saddle River, NJ, USA: Prentice Hall, 1999.
- [3] R. G. Gallager e D. C. van Voorhis, “Optimal source codes for geometrically distributed integer alphabets,” *IEEE Transactions on Information Theory*, vol. IT-21, pp. 228–230, Mar. 1975.
- [4] G. K. Wallace, “The JPEG still picture compression standard,” *Communications of the ACM*, vol. 34, pp. 30–44, Abr. 1991.
- [5] J. L. Mitchell, W. B. Pennebaker, C. E. Frogg, e D. J. Legall, *MPEG Video Compression Standard*. New York: Chapman and Hall, 1996.
- [6] “Coding of moving pictures and associated audio for digital storage media up to 1.5 Mbits/s.” ISO/IEC JTC1 CD 11172, 1992.
- [7] “Generic coding of moving pictures and associated audio.” ISO/IEC JTC1 CD 13818, 1994.
- [8] “Information technology – coding of audio-visual objects.” ISO/IEC 14496, 1999.
- [9] C. Loeffler, A. Ligtenberg, e G. Moschytz, “Practical fast 1D DCT algorithms with 11 multiplications,” *Proceedings ICASSP*, vol. 2, pp. 988–991, 1989.
- [10] A. Said e W. A. Pearlman, “A new fast and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, 1996.

- [11] J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 41, no. 12, pp. 3445–3462, 1993.
- [12] G. Shaw e D. Manolakis, “Signal processing for hyperspectral image exploitation,” *IEEE Signal Processing Magazine*, vol. 19, pp. 12–16, Jan. 2002.
- [13] A. K. Jain, *Fundamentals of Digital Image Processing*. Prentice-hall Information and System Sciences Series, Englewood Cliffs, NJ, USA: Prentice-hall Inc., 1989.
- [14] T. M. Cover e J. Thomas, *Elements of Information Theory*. New York, NY, USA: John Wiley & Sons, 1991.
- [15] J. Rissanen e G. G. Langdon, Jr, “Arithmetic coding,” *IBM Journal of Research and Development*, vol. 23, pp. 149–162, Mar. 1979.
- [16] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York, NY, USA: McGraw-Hill, 2nd ed., 1984.
- [17] M. Vetterli e J. Kovačević, *Wavelets and Subband Coding*. Prentice Hall Signal Processing Series, Upper Saddle River, NJ, USA: Prentice Hall PTR, 1995.
- [18] R. L. Joshi, V. J. Crump, e T. R. Fisher, “Image subband coding using arithmetic and trellis coded quantization,” *IEEE Trans. Circ. & Syst. Video Tech.*, vol. 5, pp. 515–523, Dez. 1995.
- [19] R. L. Joshi, T. R. Fischer, e R. H. Bamberger, “Optimum classification in subband coding of images,” in *Proc. 1994 IEEE Int. Conf. on Image Processing*, vol. II, (Austin, TX, USA), pp. 883–887, Nov. 1994.
- [20] N. S. Jayant e P. Noll, *Digital Coding of Waveforms*. Prentice-hall Signal Processing Series, Englewood Cliffs, NJ, USA: Prentice-hall Inc., 1984.
- [21] D. S. Taubman e M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Norwell, MA, USA: Kluwer Academic Publishers, 2002.
- [22] H. S. Malvar, *Signal Processing with Lapped Transforms*. Norwood, MA, USA: Artech House Inc., 1992.
- [23] “Independent JPEG group.” <http://www.ijg.org/>, Jul. 2002.
- [24] “The MPEG home page.” <http://mpeg.telecomitalialab.com/>.