

Codificação de Imagens

Marcelo Coelho Ferraz

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE MATEMÁTICA PURA
E APLICADA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO
DO GRAU DE MESTRE EM CIÊNCIAS

Aprovada por:

Jonas de Miranda Gomes

Luiz Velho

Marcelo Dreaux

RIO DE JANEIRO, RJ - BRASIL

28 de Junho de 1998

À minha querida esposa Juliana

Agradecimentos:

Ao Professor Jonas de Miranda Gomes, por seu trabalho de orientação

Ao CNPq, FAPERJ, FINEP e IBM BRASIL, por seu apoio ao Projeto VISGRAF, do qual este trabalho faz parte.

Ao CNPq, pelo apoio financeiro recebido durante os cursos que fiz no IMPA.

Aos participantes do Projeto VISGRAF do IMPA que ajudaram a elaboração do trabalho, especialmente ao Romildo, à Kátia e ao Luiz Velho.

À toda minha família, por seus ininterruptos apoios e incentivos.

Ao Dr. Romualdo, por suas preciosas revisões, especialmente no capítulo “Percepção e Codificação de Imagens”.

Ao meu pai Inhaúma, entre muitas outras razões, por suas grandes contribuições a este trabalho através de discissões técnicas e minuciosas revisões.

À minha esposa Juliana, que com seu auxílio e companheirismo constantes aliados à compreensão e incentivo na horas mais necessárias, contribuiu enormemente para este trabalho.

A todos aqueles que, de alguma forma, contribuíram para a realização deste trabalho.

Resumo da Tese apresentada ao IMPA como parte integrante dos requisitos necessários para a obtenção do grau de mestre em ciências (M.Sc.)

Codificação de Imagens

Marcelo Coelho Ferraz

Junho de 1998

Orientador: Jonas de Miranda Gomes

Diversos métodos de codificação têm sido desenvolvidos com o objetivo de reduzir a quantidade de informação necessária para representar imagens. Apesar destes métodos apresentarem diferentes abordagens, existem bases comuns que podem ser utilizadas para entendê-los e compará-los.

Este trabalho tem como objetivo apresentar as abordagens mais importantes de codificação de imagens, como por exemplo a codificação preditiva, por transformadas, em sub-bandas e fractais, enfocadas através de duas bases gerais: a Teoria da Codificação, e a divisão do processo de codificação nas etapas de “Transformação de Representação”, “Quantização” e “Atribuição de códigos binários aos elementos da fonte de imagens quantizadas”.

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

Image Coding

Marcelo Coelho Ferraz

June, 1998

Advisor: Jonas de Miranda Gomes

Several coding methods have been developed in order to reduce the amount of information required to represent images. Although these methods use different approaches, there are common basis that may be used to understand and compare them.

The objective of this thesis is to present the most important image coding approaches, such as predictive, transform, subband and fractal coding, through a common framework grounded on two general basis: the Coding Theory, and the decomposition of the coding process in the following steps: “Transform of Representation”, “Quantization” and “Code-Word Assignment (Entropy Coding)”.

Sumário

1	Introdução	9
2	Sistemas de Comunicações e Fontes de Imagens	11
2.1	Introdução	11
2.2	Sistemas de Comunicações	11
2.3	Fontes e Alfabetos	13
2.3.1	Fontes e Alfabetos de Imagens	14
2.4	Espaço de Cor	15
2.4.1	Visão Humana	15
2.4.2	Colorimetria	16
2.4.3	Imagens Artificiais	19
2.5	Amostragem	19
2.5.1	Teoria da Amostragem	20
2.5.2	Teoria da Amostragem no Caso de Imagens	23
2.6	Quantização no Espaço de Cores	27
3	Teoria da Codificação	31
3.1	Introdução	31
3.2	Fundamentos	31
3.2.1	Medidas de Informação	31
3.2.2	Entropia	32
3.2.3	Informação Mútua	35
3.2.4	Critério de Fidelidade	37
3.3	Função Taxa-Distorção	38
3.3.1	Aplicações para Imagens	39
4	Modelos de Fontes de Imagens	41
4.1	Introdução	41
4.1.1	Modelos Locais	43
4.2	Modelos Preditivos	44
4.2.1	Modelos Unidimensionais	45
4.2.2	Modelos Bidimensionais	50
4.2.3	Modelos de Predição de Movimentos	53
4.3	Modelos de Covariância	55

4.3.1	Covariância para Imagens Estáticas	56
4.3.2	Covariância para Imagens Dinâmicas	58
4.3.3	Modelos de Covariância e Transformadas	59
4.4	Não Estacionariedade das Imagens	65
4.5	Modelos de Correlação das Cores	66
4.5.1	Classificação das Representações de Cores	66
4.5.2	Comportamento Probabilístico de Imagens Coloridas	68
5	Percepção e Codificação de Imagens	71
5.1	Introdução	71
5.2	Componentes da Visão Humana	72
5.3	Aproximação de Imagens Levando em Consideração a Visão Humana	75
5.4	Características da Visão Humana	77
5.4.1	Caracterização de Cores em Três Dimensões	77
5.4.2	Filtragem Passa-Baixas no Tempo	78
5.4.3	Não-Linearidade na Percepção de Intensidade Luminosa	80
5.4.4	Filtragem Passa-Faixa no Espaço	81
5.4.5	Banda Passante Diferenciada nas Coordenadas de Cores	81
5.4.6	Importância das Bordas no Entendimento das Imagens	82
5.4.7	Conteúdo de Informação das Bordas	84
5.4.8	Comportamento Anisotrópico da Percepção Visual em Regiões Próximas de Arestas	84
5.4.9	Redução de Sensibilidade em Áreas em Movimento	84
6	Concepção Geral da Codificação de Imagens	87
6.1	Introdução	87
6.2	Estrutura Geral para o Estudo de Codificação de Imagens	87
7	Quantização	91
7.1	Exemplo de Quantização: Caso Unidimensional	92
7.2	Caso Multidimensional	93
7.3	Aplicações da Quantização	96
7.3.1	Quantização Vetorial no Espaço	96
7.3.2	Quantização Adaptativa	97
7.3.3	Classificação de Imagens antes da Quantização	98
7.3.4	Máquina de Estado de Quantização Vetorial	98
7.3.5	Codificação por “Truncagem de Blocos” (BTC)	99
7.3.6	Quantização dos Coeficientes Transformados	100
7.3.7	Quantização Escalar dos Coeficientes	101
7.3.8	Quantização Vetorial Inter-Bloco	103

8	Atribuição de Códigos a Fontes Discretas	105
8.1	Introdução	105
8.2	Códigos de Huffman	106
8.3	Codificação Aritmética	111
8.3.1	Base Teórica da Codificação Aritmética	114
8.4	Codificação Universal e Métodos de Lempel-Ziv	117
8.4.1	Introdução	117
8.4.2	Métodos de Ziv e Lempel	119
8.4.3	LZ 1	119
8.4.4	LZ 2	121
8.4.5	Aplicações dos Métodos LZ em Codificação de Imagens	121
9	Transformações de Representação	123
9.1	Introdução	123
9.2	Classificação dos Métodos de Codificação	123
9.3	Base Comum para as Transformações de Representação	124
9.3.1	Características Estatísticas de Imagens	124
9.3.2	Características da Visão Humana	125
10	Codificação Preditiva	127
10.1	Descrição do Método	127
10.2	Base Teórica da Codificação	128
10.2.1	Fundamentos para a Redução de Redundância	128
10.2.2	Características da Visão Humana Exploradas	130
10.2.3	Codificação Preditiva para Imagens	130
10.2.4	Determinação dos Sinais a serem Preditos	131
10.2.5	Determinação das vizinhanças dos elementos a serem preditos	133
10.2.6	Determinação dos Preditores e Estimção de Parâmetros	136
10.2.7	Quantização do Sinal Residual	137
10.3	Possibilidade de Emprego de Adaptabilidade	141
10.4	Possibilidade de Emprego de Metodos Híbridos	142
10.5	Possibilidade de Emprego de Codificação sem Perdas	142
10.6	Complexidade da Implementação	143
10.7	Variantes da Codificação	143
10.7.1	Modulação Delta	143
10.7.2	Modulação Delta Adaptativa	146
10.7.3	Modulação DPCM (“ <i>Differential Pulse Code Modulation</i> ”)	146
10.7.4	Reabastecimento Condicional	146
10.7.5	Predição de Movimentos	146
11	Codificação por Transformada	149
11.1	Introdução	149
11.2	Descrição do Método	150
11.2.1	Decomposição da Imagem em Blocos	150

11.2.2	Transformação de Coordenadas	150
11.2.3	Quantização dos Coeficientes Transformados	152
11.2.4	Atribuição de Códigos Binários aos Símbolos da Quantização	152
11.3	Base teórica para a Codificação	152
11.3.1	Fundamentos para a Redução de Redundância - Desacoplamento Estatístico entre Coeficientes	152
11.3.2	Fundamentos para a Redução de Redundância - Comportamento dos Coeficientes	154
11.3.3	Características da Visão Humana Exploradas	155
11.4	Possibilidade de Emprego de Adaptabilidade	155
11.4.1	Validação Local da Aproximação da Transformada KL por Transformadas Rápidas	156
11.4.2	Quantização dos Coeficientes de Acordo com as Características Locais das Imagens	156
11.4.3	Atribuição Adaptativa de Códigos Binários aos Símbolos Codificados	156
11.5	Possibilidade de Emprego de Métodos Híbridos	157
11.5.1	Codificação Preditiva na Coordenada Tempo	157
11.5.2	Codificação Especial para Regiões com Arestas	157
11.6	Possibilidade de Emprego de Codificação Sem Perdas	158
11.7	Complexidade de Implementação	158
11.8	Faixa de Compressão Atingida	158
12	Codificação em Sub-bandas	161
12.1	Introdução	161
12.2	Descrição do Método	161
12.2.1	Decomposição da Imagem em Sub-bandas	161
12.2.2	Quantização dos Coeficientes das Sub-bandas	163
12.2.3	Atribuição de Códigos Binários aos Símbolos da Quantização	163
12.3	Base Teórica para a Codificação	164
12.3.1	Filtros de Reconstrução Perfeita em Dimensão 1	164
12.3.2	Filtros de Reconstrução Perfeita em Dimensões Maiores	170
12.3.3	Análise Multi-Escala	171
12.3.4	Conexão entre Filtragem de Recuperação Perfeita e Análise Multi-Escala	174
12.3.5	Características da Visão Humana Exploradas	177
12.4	Possibilidade de Emprego de Adaptabilidade	178
12.4.1	Ajuste de Direções de Filtragem Adaptativa	178
12.4.2	Quantização Vetorial Adaptativa	178
12.4.3	Filtragem em Sub-Bandas Adaptativa	178
12.5	Possibilidade de Emprego de Métodos Híbridos	179
12.5.1	Codificação Preditiva na Coordenada Tempo	179
12.5.2	Codificação Preditiva Entre Diferentes Escalas	179
12.5.3	Codificação Preditiva dos Livros-Códigos	179
12.6	Possibilidade de Emprego de Codificação sem Perdas	179
12.7	Complexidade de implementação	180

12.8	Faixa de Compressão Atingida	180
13	Codificação Através de Fractais	183
13.1	Introdução	183
13.2	Descrição do Método	183
13.2.1	Contrações no Espaço das Imagens	184
13.2.2	Contrações Compostas por Transformações Afins Locais	184
13.2.3	Restrições nos Formatos e Tamanhos de Células Imagem e Domínio	187
13.2.4	Classificação das Células Imagem e Domínio	192
13.3	Base Teórica para a Codificação	194
13.3.1	Teorema da Colagem	194
13.3.2	Características da Visão Humana Exploradas	194
13.3.3	Importância Perceptual das Arestas	194
13.4	Possibilidade de Emprego de Adaptabilidade	194
13.4.1	Adaptação de Tamanhos e Formatos de Regiões	194
13.5	Possibilidade de Emprego de Métodos Híbridos	195
13.5.1	Reabastecimento Condicional	195
13.6	Possibilidade de Emprego de Codificação sem Perdas	195
13.7	Complexidade de Implementação	195
13.8	Faixa de Compressão Atingida	195
14	Segunda Geração de Métodos de Codificação de Imagens	197
14.1	Introdução	197
14.2	Descrição dos Métodos	198
14.2.1	Classificação quanto ao tipo de técnica de detecção de arestas	198
14.2.2	Classificação quanto ao uso das arestas na codificação	199
14.2.3	Aumento proporcional da importância das arestas na codificação	199
14.2.4	Inferência das imagens a partir de suas informações nos bordos	200
14.3	Base Teórica para a Codificação	201
14.3.1	Características da Visão Humana Exploradas	201
14.4	Possibilidade de Emprego de Adaptabilidade	201
14.5	Possibilidade de Emprego de Métodos Híbridos	201
14.6	Possibilidade de Emprego de Codificação sem Perdas	201
14.7	Complexidade de Implementação	202
14.8	Faixa de Compressão Atingida	202
15	Codificação por Modelos	203
15.1	Introdução	203
15.2	Descrição do Método	204
15.2.1	Restrição do Universo de Objetos Pertencentes às Cenas	204
15.2.2	Modelos de Faces Humanas	205
15.2.3	Extração dos Parâmetros do Modelo	206
15.3	Possibilidade de Emprego de Métodos Híbridos	207
15.4	Possibilidade de Emprego de Codificação sem Perdas	207

15.5 Complexidade de Implementação	208
15.6 Faixa de Compressão Atingida	208
16 Conclusão	209

Capítulo 1

Introdução

A utilidade prática da codificação de imagens é muito grande devido à importância das imagens no dia-a-dia das pessoas e aos custos decorrentes de seu armazenamento e transmissão.

Diversos métodos de codificação de imagens objetivando a redução da quantidade de informação necessária para representar imagens (compressão) têm sido desenvolvidos. Alguns destes métodos derivaram diretamente do processamento de sinais de voz, como a codificação preditiva e a baseada em decomposição em sub-bandas. Outras foram desenvolvidas especificamente para o caso de imagens.

Apesar dos diversos métodos apresentarem diferentes abordagens, existem bases comuns que podem ser utilizadas tanto para descobrir quais os fatores essenciais na codificação quanto para entender e comparar as diferentes técnicas utilizadas. O objetivo deste trabalho é justamente apresentar as diferentes abordagens da codificação enfocadas dentro de tais bases comuns. Mais especificamente, este trabalho apresenta duas formulações gerais para entendimento do problema de codificação de imagens.

Na primeira parte do estudo a teoria da codificação, originalmente proposta por Shannon ([164]), é utilizada para formalizar o problema (caracterizando as fontes de imagens e seus alfabetos) e detectar os fatores essenciais para o sucesso da codificação. Uma vez determinados estes fatores, a saber caracterização probabilística das fontes de imagens e características da visão humana, este trabalho apresenta os principais modelos, conceitos e informações provenientes das áreas de estudos destes fatores para a codificação de imagens.

Em seguida apresenta-se um modelo de entendimento do processo de codificação de imagens. Este modelo divide a codificação em três fases:

- Transformação de representação
- Quantização
- Atribuição de códigos binários aos elementos da fonte de imagens quantizadas

Finalmente as diversas técnicas utilizadas em cada uma das etapas deste modelo são descritas, com destaque para a etapa de transformação de representação que, por apresentar uma variedade maior de abordagens e ser objeto de ativa pesquisa, requer seis capítulos para sua apresentação.

Capítulo 2

Sistemas de Comunicações e Fontes de Imagens

2.1 Introdução

Para entender os conceitos existentes nos métodos de compressão de imagens ou criar novos métodos na área é importante conhecer um modelo matemático que descreva o fenômeno envolvido.

Um modelo matemático bastante adequado para entender a compressão de imagens é o modelo de sistemas de comunicações. Através deste modelo foi elaborada a teoria da codificação [164]. A compressão de imagens é na verdade um caso particular de Teoria de Codificação.

Este capítulo visa introduzir o conceito de sistemas de comunicações e definir o modelo de fontes de imagens digitais a ser utilizado no decorrer do trabalho.

2.2 Sistemas de Comunicações

Um *sistema de comunicações* tem como objetivo reproduzir informações geradas por uma fonte em um destino remoto.

Tais sistemas apresentam cinco elementos:

- **Fonte de Informação**, que gera as informações a serem reproduzidas.
- **Codificador**, que transforma a informação numa forma mais adequada para ser enviada ao receptor.
- **Canal**, pelo qual a informação codificada flui até chegar ao decodificador.
- **Decodificador**, que realiza os passos inversos do codificador. Em outras palavras, o decodificador transforma a informação codificada para sua forma original.
- **Receptor**, que é o destino final da informação.

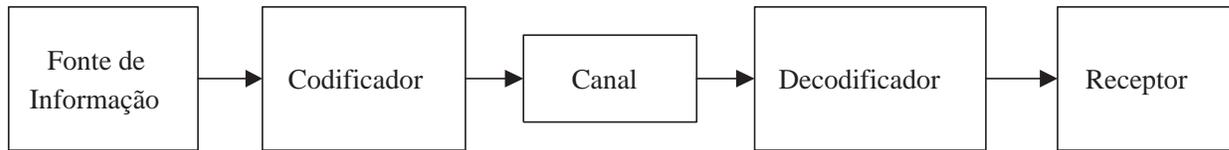


Figura 2.1: Sistema de Comunicações

Este modelo está ilustrado na figura 2.1. Ele é facilmente adequado para o problema de codificação de imagens.

Quando se deseja codificar imagens, na verdade objetiva-se representar imagens da forma mais eficiente possível. A eficiência é medida em termos de quantidade de informação necessária para representar as imagens¹.

As imagens que se deseja codificar fazem parte de um conjunto de imagens possíveis de serem geradas. Este conjunto de imagens pode ser entendido como a saída (“output”) de uma fonte de imagens.

A codificação processa as imagens originais. O resultado deste processamento, isto é, imagens codificadas, são mais adequadas para fluírem pelo canal. A adequação, conforme mencionado anteriormente, é medida em termos de quantidade de informação necessária para representar as imagens provenientes das fontes.

O canal pode ser um meio físico para armazenamento ou transmissão de imagens. A característica mais importante do canal no caso de codificação de imagens é a presença ou não de ruído. Um canal ruidoso produz erros nas mensagens (imagens) codificadas. Durante o decorrer deste trabalho os canais serão considerados não-ruídos.

O decodificador é um elemento inteiramente relacionado com o codificador, realizando a operação inversa deste. Na verdade quando se menciona codificação ou compressão de imagens, implicitamente refere-se à decodificação ou descompressão de imagens respectivamente.

O receptor, destino final da imagem, é o elemento capaz de avaliar a qualidade dos processos de compressão e descompressão de imagens.

O problema de codificação de imagens sob a ótica dos sistemas de comunicações consiste em encontrar uma codificação que reduza a quantidade de informação necessária para caracterizar imagens mantendo um “padrão de qualidade” de imagem para o receptor final.

Alguns dos conceitos citados anteriormente, tais como quantidade de informação para representar imagens e qualidade de representação para o receptor, necessitam de definições mais precisas. O próximo capítulo tem como objetivo esclarecer estes e outros pontos. As próximas seções, por outro lado, visam caracterizar as fontes e alfabetos de imagens.

¹A definição precisa de quantidade de informação para representar uma imagem será fornecida no capítulo 3.1.

2.3 Fontes e Alfabetos

As *fontes* são os elementos geradores de informação. Sob o ponto de vista dos sistemas de comunicações (considerado nesta abordagem), as fontes geram informações de forma aleatória. Com efeito, se as fontes tivessem um padrão determinístico, sua representação seria trivial, seguindo o padrão conhecido.

Para entender a natureza aleatória das fontes nos Sistemas de Comunicações convém analisar dois exemplos:

Exemplo 2.1 Vóz humana em telefonia: *O sistema telefônico tem como objetivo reproduzir os sons gerados por um transmissor no destino final (receptor). O transmissor gera sons, isto é, ondas de pressão, que são convertidas em sinais elétricos no aparelho telefônico. Portanto, para o sistema telefônico o conjunto transmissor/aparelho telefônico constitui uma fonte de sinais elétricos. Estes sinais devem ser reproduzidos no destino. É importante destacar que o transmissor produz sons à sua própria vontade, ou seja, de forma totalmente independente do Sistema Telefônico. Desta forma a geração de sinais elétricos pela fonte é um processo aleatório para o sistema telefônico.*

Exemplo 2.2 Bits em uma rede de comunicação digital: *Uma rede de comunicação digital é responsável por transmitir uma informação na forma digital (seqüência de dígitos binários ou bits) de uma estação transmissora para uma estação receptora. A estação transmissora gera uma seqüência de bits a serem transmitidos de acordo com os processos que estão sendo executados nela (transmissão de arquivos, acesso a banco de dados remoto, transmissão de notas e mensagens, etc ...). Como os processos sendo executados na estação transmissora são independentes da rede de comunicação, as seqüências de bits a serem transmitidas também são independentes da rede. Em outras palavras, as seqüências de bits a serem transmitidos são eventos aleatórios para a rede de comunicação digital.*

A natureza aleatória das fontes de imagens deve ser modelada através de processos estocásticos. Desta forma, em cada instante t as fontes são modeladas por uma variável aleatória $X_t : \Omega \rightarrow \mathcal{A}_t$, que mapeia o espaço dos eventos Ω de um espaço de probabilidade $(\Omega, \mathcal{B}, \mathcal{P})$ ² num conjunto de informações \mathcal{A}_t . Este conjunto de informações é chamado de *alfabeto da fonte no instante t* .

Os conjuntos \mathcal{A}_t podem variar no tempo, isto é, pode-se ter $\mathcal{A}_{t_1} \neq \mathcal{A}_{t_2}$ para $t_1 \neq t_2$. Apesar disto, este trabalho contempla apenas fontes com alfabeto invariante no tempo \mathcal{A} . Em outras palavras, neste trabalho fontes são processos estocásticos $X_t : \Omega \rightarrow \mathcal{A}$, ou $X(t) : \Omega \rightarrow \mathcal{A}$.

Diversas formas de informações podem ser geradas. As fontes podem gerar sinais elétricos (comunicação telefônica através de pares trançados), sinais luminosos (comunicação por meio de fibras óticas), ondas de pressão (comunicação através de som), símbolos (telegrafia), e mais uma infinidade de formas diferentes de informação.

Estas diversas formas de informações possíveis são refletidas no modelo matemático como diferentes possibilidades para o conjunto \mathcal{A} . No exemplo 1 apresentado anteriormente, o

²Um espaço de probabilidade é uma tripla que consiste de um espaço amostral (conjunto de eventos) Ω , uma σ -Álgebra \mathcal{B} de subconjuntos de Ω e de uma função de probabilidade \mathcal{P} .

alfabeto da fonte é composto pelos níveis possíveis para o sinal elétrico de entrada de um sistema telefônico, ou seja, um intervalo $(a, b) \subset R$. Já no exemplo 2 o conjunto \mathcal{A} é composto unicamente por dois elementos: $\{0, 1\}$. A próxima subseção tem como objetivo modelar os alfabetos de fontes de imagens.

2.3.1 Fontes e Alfabetos de Imagens

Modelo Conceitual

Uma imagem é o resultado de estímulos luminosos produzidos numa superfície bidimensional. A luz proveniente do mundo tridimensional incide sobre os receptores foto-sensíveis da superfície bidimensional gerando os impulsos luminosos.

No caso da visão humana, a superfície bidimensional é a retina e os receptores as células foto-sensíveis do olho. Os dispositivos de captação de imagens, como as câmeras, também dispõem de superfícies contendo foto-receptores, como os CCD's.

A percepção de uma luz incidente nos receptores é dada pelo conjunto de estímulos por ela gerados. Tal percepção corresponde à cor. Em outras palavras, a cor é a manifestação da luz nos receptores pertencentes à superfície bidimensional onde a imagem é formada. Para manter uma seqüência na apresentação do alfabeto de imagens, não entraremos em detalhes sobre a caracterização das cores agora. Basta considerar abstratamente um conjunto \mathcal{C} contendo as cores possíveis. Este conjunto será denominado espaço de cor.

Seja \mathcal{S} a superfície onde a imagem é formada. A caracterização da imagem no instante t é dada por uma função $f : \mathcal{S} \rightarrow \mathcal{C}$. Quando se deseja caracterizar as imagens variando no tempo, a função f requer mais uma dimensão em seu domínio: o tempo. Desta forma, as seqüências de imagens são caracterizadas por $f : \mathcal{S} \times R^+ \rightarrow \mathcal{C}$.

Quando imagens em instantes específicos são analisadas individualmente, isto é, sem considerar dimensão tempo, as imagens são ditas *estáticas*. Quando a dimensão tempo é considerada, as imagens são ditas *dinâmicas*. Esta classificação reflete-se na forma de abordar o problema de compressão de imagens. Isto porque os métodos de compressão em geral procuram reduzir a redundância nas fontes. Os métodos de compressão de imagens estáticas, ao contrário dos de imagens dinâmicas, não levam em conta a redundância temporal que pode existir nas fontes de imagens.

Sob o ponto de vista de sistemas de comunicações as fontes de imagens geram distribuições espaciais de cores. Em outras palavras, o alfabeto das fontes de imagens é formado por distribuições de cores na superfície receptora $f : \mathcal{S} \rightarrow \mathcal{C}$.

Esta determinação de alfabeto ainda é muito abstrata para que tenha aplicação prática. Algumas aproximações e simplificações precisam ser feitas para que tal modelo seja aplicável. Suas principais limitações dizem respeito à impossibilidade de caracterização das imagens em computadores digitais (caracterização através de um conjunto finito de parâmetros). As simplificações necessárias são classificadas da seguinte forma:

- Modelagem do espaço de cor através de um espaço vetorial tri-dimensional
- Discretização do domínio (tanto no espaço, isto é, na superfície receptora \mathcal{S} , quanto no tempo)

- Quantização das coordenadas dos vetores de cor no espaço tri-dimensional

Através destas aproximações a caracterização das imagens pode ser feita por um número finito de informações binárias (bits). Inicialmente porque através da primeira aproximação as cores que compõem a imagem são representadas através de um número finito de parâmetros, e não por distribuições espectrais de potência. Além disto, a terceira aproximação permite que a caracterização de cada um destes parâmetros necessite de um número finito de bits. Estas duas aproximações implicam na discretização do conjunto imagem da função $f : \mathcal{S} \times R^+ \rightarrow \mathcal{C}$.

A segunda aproximação conclui a caracterização das imagens por meio de informação finita ao discretizar também o domínio de f . Ainda dentro do contexto de simplificações e aproximações, vale dizer que em termos práticos a superfície \mathcal{S} é um retângulo $[0, L_x] \times [0, L_y]$, no qual a discretização do domínio é realizada (conhecida por *amostragem espacial*, enquanto que a discretização na dimensão tempo é feita através de *amostragem temporal*).

2.4 Espaço de Cor

Nesta seção o espaço de Cores é estudado para que se entenda como as cores (distribuições espectrais de luz) podem ser representadas através de um número finito de coeficientes.

2.4.1 Visão Humana

Conforme mencionado anteriormente, o espaço de cores \mathcal{C} é o conjunto das distribuições de intensidade luminosa no espectro de comprimentos de onda. Matematicamente o espaço de cor é o conjunto das funções limitadas de R^+ em R^+ .

O menor espaço vetorial que contém o conjunto \mathcal{C} é o conjunto das funções limitadas de R^+ em R . Este espaço vetorial, denotado neste trabalho por \mathcal{C}' possui dimensão infinita.

A aproximação de \mathcal{C}' por um espaço de dimensão finita é fortemente baseada na visão humana.

Esta seção discorre brevemente sobre a percepção humana de cores. Maiores detalhes podem ser encontrados em [68], [200] e [165].

A retina no olho humano corresponde à superfície receptora no modelo de imagens. Ela é composta por dois tipos de células fotosensoras: *bastonetes* e *cones*.

Os bastonetes existem em maior número. Eles apresentam maior sensibilidade que os cones em condições de baixa intensidade luminosa (como durante a noite, por exemplo). Entretanto os bastonetes não são capazes de detectar cores.

Os cones, por outro lado, existem em menor quantidade no olho. Eles são distribuídos de forma bastante não-uniforme na retina, concentrando-se na região “central” do olho: a *fovea centralis*. Os cones necessitam de mais intensidade luminosa que os bastonetes para apresentar resposta. No entanto, eles são responsáveis pela detecção de cor da visão.

Devido às características complementares de cones e bastonetes, pode-se considerar que os cones são responsáveis pela visão diurna, com altos níveis de iluminação, também chamada de visão *fotópica*. Já os bastonetes podem ser considerados responsáveis pela visão noturna, com baixos níveis de iluminação, também chamada de visão *escotópica*.

A grande maioria das aplicações de imagens encontra-se na faixa de visão fotópica. Este trabalho segue a linha da literatura na área, considerando como modelo de visão do usuário final a visão fotópica.

O comportamento das células fotosensoras é modelado através de funcionais \mathcal{S} que associam a cada cor $c \in \mathcal{C}$ a resposta da célula sensora $\mathcal{S}[c]$.

Existem três tipos de cones. Cada tipo apresenta uma fotosensibilidade diferente \mathcal{S}_i . Um modelo bastante aceito dos estímulos \mathcal{S}_i na visão fotópica é o de respostas espectrais. Este modelo estabelece que o estímulo de uma célula receptora $\mathcal{S}_i[c]$ é dado pela equação :

$$\mathcal{S}_i[c] = \int_0^{\infty} h_i(\lambda) f(\lambda) d\lambda$$

onde λ é o comprimento de onda e h_i é uma função contínua chamada de resposta espectral da célula. A cada tipo de cone corresponde uma resposta espectral h_i . As respostas espectrais dos diferentes cones da visão humana estão ilustradas na figura 2.2.

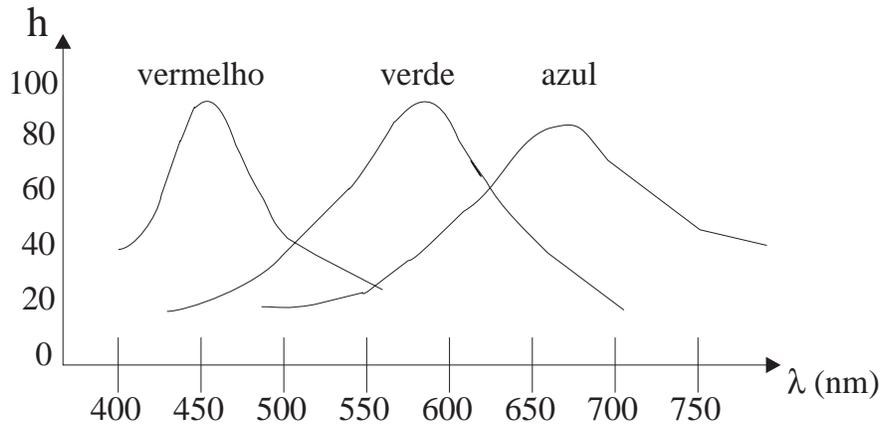


Figura 2.2: Respostas espectrais do olho humano.

É interessante observar que as respostas espectrais dos cones humanos têm suporte compacto contido no intervalo $[\lambda_1, \lambda_2]$, denominado espectro de luz visível (onde $\lambda_1 = 360$ nm e $\lambda_2 = 830$ nm [165]). Outro aspecto importante a observar é que as respostas espectrais não são mutuamente excludentes. Em outras palavras, uma cor nunca estimula apenas um tipo de cone. A percepção humana de cores é dada pela composição das respostas dos três tipos de cones.

Os funcionais de resposta da visão fotópica são lineares segundo este modelo.

2.4.2 Colorimetria

Tomando como base a percepção humana de cores, a colorimetria modela a representação de cores em dimensão finita. A maior parte desta teoria destina-se a representações de cores em três dimensões.

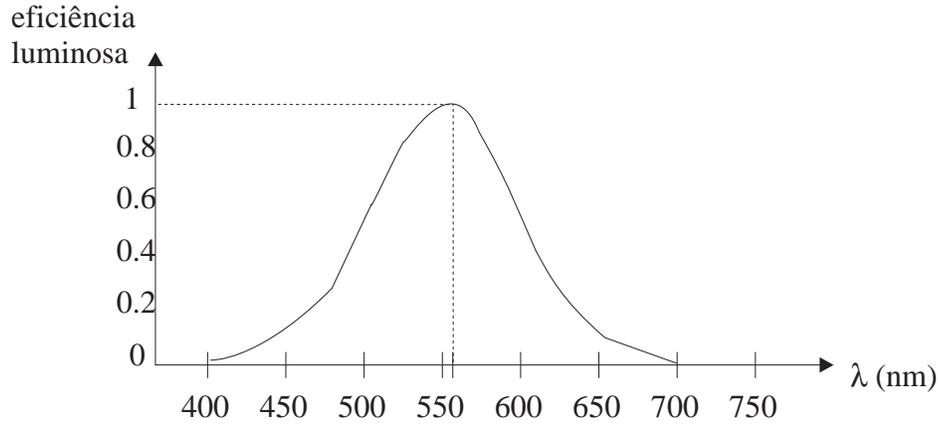


Figura 2.3: Função de eficiência luminosa relativa

O modelo tridimensional de cores está baseado no fato da visão humana ter três diferentes células cones para recepção de estímulos luminosos. O princípio básico é que qualquer cor pode ser “casada”, isto é, aproximada sem perda perceptual por uma combinação linear de três cores “independentes”. A independência significa que nenhuma das três cores pode ser obtida como combinação linear das demais. Mais precisamente, duas cores são ditas *metaméricas*, caso a visão humana não seja capaz de identificar diferenças entre estas cores (duas cores podem ser aproximadas sem perda perceptual).

As leis de Grassman ([71]) regem este modelo tridimensional. Antes de apresentar as Leis de Grassman é necessário definir *luminância*.

A luminância de uma cor é uma medida que procura avaliar a intensidade luminosa associada à cor. Ela é obtida através da seguinte equação:

$$\text{Lum} = K \int_{\lambda_1}^{\lambda_2} f(\lambda)v(\lambda)d\lambda \quad (2.1)$$

onde $[\lambda_1, \lambda_2]$ é o espectro de luz visível, $f(\lambda)$ é a distribuição de intensidade luminosa referente à cor, $v(\lambda)$ é uma função de ponderação chamada *eficiência luminosa relativa do sistema visual* e K é uma constante que depende do sistema métrico utilizado (no sistema MKS K vale aproximadamente 680 lumens/ watt). O gráfico desta função está ilustrado na figura 2.3.

Definida a luminância, podemos apresentar as Leis de Grassman:

1. Qualquer cor pode ser casada por uma combinação de no máximo três cores. As cores utilizadas para gerar a cor inicial são chamadas de componentes.
2. A luminância de uma combinação de cores é igual à soma das luminâncias das cores componentes.
3. A visão humana é incapaz de determinar as cores componentes a partir de sua combinação.

4. Um casamento de cores em um nível de luminância vale por uma larga faixa de luminâncias (esta lei especifica a linearidade do modelo de visão de cores).
5. Se C_1 e C_2 casam e C_2 e C_3 casam então C_3 e C_1 casam (Transitividade).
6. Adição de Cores:
 - as operações de soma e subtração de cores e multiplicação de uma cor por um escalar equivalem respectivamente à soma e subtração das distribuições de intensidades luminosas das cores, e à multiplicação da distribuição de intensidade luminosa da cor pelo escalar.
 - duas cores são consideradas “iguais” quando houver casamento entre elas. Na verdade a relação de casamento estabelece classes de equivalência no conjunto de cores. Simplificadamente representa-se a igualdade de cores quando, na verdade, a igualdade ocorre entre as classes de equivalência induzidas pela operação de casamento de cores.

Com esta notação:

$$\begin{aligned} \text{Se } C_1 &= C_2 \text{ e} \\ C'_1 &= C'_2 \\ \text{Então } \alpha_1 C_1 + \alpha_2 C_2 &= \alpha_1 C'_1 + \alpha_2 C'_2 \end{aligned}$$

7. Subtração de Cores Se uma combinação das cores C_1 e C_2 casa com a mesma combinação de C'_1 e C'_2 , e ainda se as cores C_1 e C'_1 casam, então as cores C_2 e C'_2 casam.

$$\begin{aligned} \text{Se } \alpha_1 C_1 + \alpha_2 C_2 &= \alpha_1 C'_1 + \alpha_2 C'_2 \\ \text{e } C_1 &= C'_1 \\ \text{então } C_2 &= C'_2 \end{aligned}$$

8. Possibilidades de Casamentos de Cores:

- Casamento Direto:

$$\alpha C = \alpha_1 C_1 + \alpha_2 C_2 + \alpha_3 C_3 \quad (2.2)$$

- Casamento Indireto 1:

$$\alpha C + \alpha_1 C_1 = \alpha_2 C_2 + \alpha_3 C_3 \quad (2.3)$$

- Casamento Indireto 2:

$$\alpha C + \alpha_1 C_1 + \alpha_2 C_2 = \alpha_3 C_3 \quad (2.4)$$

Estas leis têm boa aplicação na prática, exceto quando os níveis de luminância são muito altos (saturação) ou muito baixos (luz no limite da visão fotópica).

2.4.3 Imagens Artificiais

As imagens artificiais são compostas por câmeras (dispositivos de recepção de imagens). As câmeras transformam a informação luminosa recebida em uma representação padrão.

O primeiro passo de uma representação padrão é a determinação de um conjunto de cores primárias. Através da composição destas cores primárias obtém-se o casamento das cores observáveis.

A representação padronizada de uma cor consiste nos coeficientes α_1 , α_2 , e α_3 utilizados para gerar a combinação das cores primárias que casa com a cor. Assim, dada uma cor C , tal que

$$C = \alpha_1 C_1 + \alpha_2 C_2 + \alpha_3 C_3 \quad (2.5)$$

então C é representada por intermédio da tripla ordenada $(\alpha_1, \alpha_2, \alpha_3)$.

Uma vez capturadas por câmeras e representadas por $(\alpha_1, \alpha_2, \alpha_3)$ as cores podem ser apresentadas. Para isto basta recompor as cores primárias usando os coeficientes $(\alpha_1, \alpha_2, \alpha_3)$.

Existem diversas representações padronizadas de cores. Estas são chamadas de *Sistemas de Representação de Cores* (Para maiores informações ler [200], [68], [101]).

Embora o número de cores primárias usado normalmente seja 3, pode-se utilizar 4 ou mais cores primárias. Por isto, este trabalho considera n_c o número de cores primárias, e consequentemente, o número de coordenadas necessárias para representar uma cor.

Após esta aproximação do espaço de cores, as fontes de imagens ficam caracterizadas através de um processo estocástico $\mathcal{X}(t)$, tendo como imagem o conjunto das funções limitadas de R (superfície receptora) em R^{n_c} . Pode-se representar as fontes através da notação $\mathcal{X}(t, x, y)$. $\mathcal{X}(t, x, y)$ denota a cor observada no ponto (x, y) do retângulo receptor no instante t .

Em verdade a aproximação do espaço de cor por R^{n_c} faz com que uma fonte $\mathcal{X}(t, x, y)$ possa ser entendida como um conjunto de fontes $\mathcal{X}_k(t, x, y)$, $k \in \{1, \dots, n_c\}$. Em outras palavras, uma fonte de imagens gera um sinal n_c -dimensional tendo como entrada três parâmetros (tempo e duas coordenadas de posição numa superfície).

2.5 Amostragem

Esta seção descreve dois processos de discretização de fontes de imagens: amostragem temporal e amostragem espacial.

Antes de entrar nos detalhes da amostragem é interessante entender sua motivação. Para que uma imagem seja determinada é preciso que se conheça distribuição de cores em toda a superfície receptora \mathcal{S} e em todo tempo. A representação completa deste conhecimento requer uma quantidade infinita de informação. Em outras palavras, não se consegue representar esta informação de forma precisa e completa através de um conjunto finito de parâmetros. Isto porque a região onde se distribuem as cores é contínua e, portanto, não enumerável.

Pode-se contornar esta característica indesejada através de representações finitas aproximadas das distribuições de cores. A amostragem é um método de aproximação de distribuições contínuas que permite a reconstrução perfeita das distribuições originais desde que

estas atendam a algumas condições. A próxima sub-seção estabelece estas condições, enquanto que as demais sub-seções desta seção concentram-se na aplicação da amostragem no domínio do tempo e do espaço respectivamente.

2.5.1 Teoria da Amostragem

A aproximação de um sinal $v(t)$ através de amostragem consiste na representação do sinal por uma sequência de coeficientes $\{c_m\}_{m \in L}$, $L \subseteq \mathcal{Z}$. Os coeficientes $\{c_m\}$ são chamados de amostras do sinal.

Esta sequência é obtida através da aproximação de $v(t)$ por meio de uma série de funções. Em outras palavras, a amostragem baseia-se inicialmente na escolha de uma sequência de funções $\{\varphi_m\}_{m \in L}$. Esta sequência é normalmente uma base do espaço de funções considerado (normalmente o L^2). Em seguida o sinal é aproximado pela série

$$\sum_{m \in L} c_m \varphi_m(t)$$

As amostras c_m são obtidas através da minimização da distância entre o sinal original e o sinal amostrado. Para o caso de L^2 como espaço de funções, a obtenção das amostras é realizada minimizando a integral:

$$\int_R (v(t) - \sum_{m=-\infty}^{\infty} c_m \mathcal{X}_{I_m}(t))^2 dt$$

Dado que a sequência de funções $\{\varphi_m\}_{m \in L}$ é previamente conhecida, a aproximação de um sinal por amostragem é representada pela sequência de amostras $\{c_m\}_{m \in L}$.

Através da amostragem uma fonte de sinais $v(t)$, que é originalmente um processo estocástico contínuo, passa a ser representada por uma sequência de variáveis aleatórias, também conhecida por processo estocástico discreto.

A amostragem mais comum é chamada de amostragem uniforme. A base de funções utilizada é parametrizada por Δt , conforme descrito abaixo:

$$\varphi_m(t) = \frac{\sin(\pi/(\Delta t)(t - m\Delta t))}{\pi/(\Delta t)(t - m\Delta t)}$$

O parâmetro Δt é conhecido por período de amostragem.

A grande vantagem de amostragem uniforme é que a obtenção de amostras é trivial. Segundo o teorema da amostragem ([105]) as amostras são os próprios valores do sinal em intervalos de tempo uniformemente espalhados, isto é $c_m = v(m\Delta t)$. Não é por mera coincidência que o parâmetro Δt está presente na determinação do valor das amostras.

Para que se possa entender as condições impostas para que os sinais $v(t)$ possam ser recuperados através de suas amostras é preciso estabelecer uma notação para servir como base.

Em cada instante t_1 o sinal $v(t)$ contém uma variável aleatória. A esperança matemática da variável aleatória $v(t_1)$ é chamada de *valor médio* do processo estocástico no instante t_1 . Sendo assim, define-se uma função $\overline{v(t)}$:

$$\overline{v(t)} = E[v(t)]$$

onde E denota a esperança matemática.

Outra medida importante das variáveis aleatórias de um sinal é a *autocorrelação*. Para cada par de instantes t_1 e t_2 define-se a autocorrelação $R_v(t_1, t_2)$ entre as variáveis aleatórias $v(t_1)$ e $v(t_2)$ através da expressão :

$$R_v(t_1, t_2) = E[v(t_1)v(t_2)]$$

À medida que t_1 e t_2 variam a expressão percorre a dimensão tempo e, portanto, pode-se também definir uma *função de autocorrelação* $R_v(t_1, t_2)$ para o sinal $v(t)$.

Um processo estocástico (em particular um sinal) é dito estacionário no sentido “lato” se suas funções valor médio e autocorrelação não variam com o tempo.

Um processo estocástico (em particular um sinal) é dito estacionário no sentido estrito caso a distribuição de probabilidade de suas variáveis aleatórias $v(t)$ seja constante. Como o valor médio e a autocorrelação dependem diretamente da distribuição de probabilidade das variáveis aleatórias envolvidas, estacionariedade no sentido estrito implica em estacionariedade no sentido “lato”. Estacionariedade no sentido “lato”, por outro lado, não implica em estacionariedade em sentido estrito. Daqui por diante um sinal será dito estacionário caso seja estacionário pelo menos em sentido “lato”.

É importante notar que a função de autocorrelação de um sinal estacionário $R_v(t_1, t_2)$ depende apenas da diferença $\tau = t_1 \leftrightarrow t_2$ entre os instantes de tempo t_1 e t_2 . Com efeito, R_v pode ser escrita na forma:

$$R_v(t_1, t_2) = R_v(t_1, t_1 \leftrightarrow \tau)$$

que evidencia a não dependência de t_2 . Neste caso a função de autocorrelação é descrita simplesmente como $R_v(\tau)$.

Um conceito importante na teoria de sinais é o de espectro de potências. Para entender o espectro de um sinal é preciso imaginar inicialmente que o sinal é decomposto em ondas senoidais de diferentes frequências e fases. A cada frequência corresponde uma potência. O espectro de um sinal estabelece como a potência do sinal distribui-se entre as frequências. De acordo com o *teorema de Wiener-Kinchine* (página 112 de [21]) o espectro de potência $G_v(f)$ de um sinal estacionário pode ser calculado através da função de autocorrelação do sinal segundo a equação 2.6:

$$G_v(f) = \mathcal{F}_\tau[R_v(\tau)] = \int_{-\infty}^{\infty} R_v(\tau)e^{-j2\pi f\tau} d\tau \quad (2.6)$$

onde \mathcal{F}_τ é a transformada de Fourier com respeito a variável τ .

Um sinal é dito de *banda limitada* se seu espectro de potência $G_v(f)$ tiver suporte compacto. Neste caso, a banda do sinal W é definida como o supremo do conjunto

$$\{|f|, \text{ tal que } G_v(f) \neq 0\}$$

O teorema da amostragem estabelece ³ :

Teorema da Amostragem : Um sinal estacionário $v(t)$ com banda limitada W é determinado linearmente a partir de suas amostras $v(m\Delta t)$ se $2W\Delta t \leq 1$.

³ver referência [105]

Esta é uma condição suficiente para que a amostragem seja um processo sem perdas (outros tipos de amostragem podem ser válidos para funções que não tenham banda limitada).

O processo de recuperação de $v(t)$ através de suas amostras é bastante simples para sinais de banda limitada. Em verdade, o espectro de potências das amostras do sinal $G_{amostras}(f)$ corresponde ao espectro de $v(t)$ mais uma série de cópias do espectro centradas nas frequências $k/(\Delta t)$:

$$G_{amostras}(f) = \sum_{k \in \mathbb{Z}} G_v(f + k/(\Delta t))$$

A figura 2.4 ilustra o espectro do sinal original e o espectro do sinal amostrado.

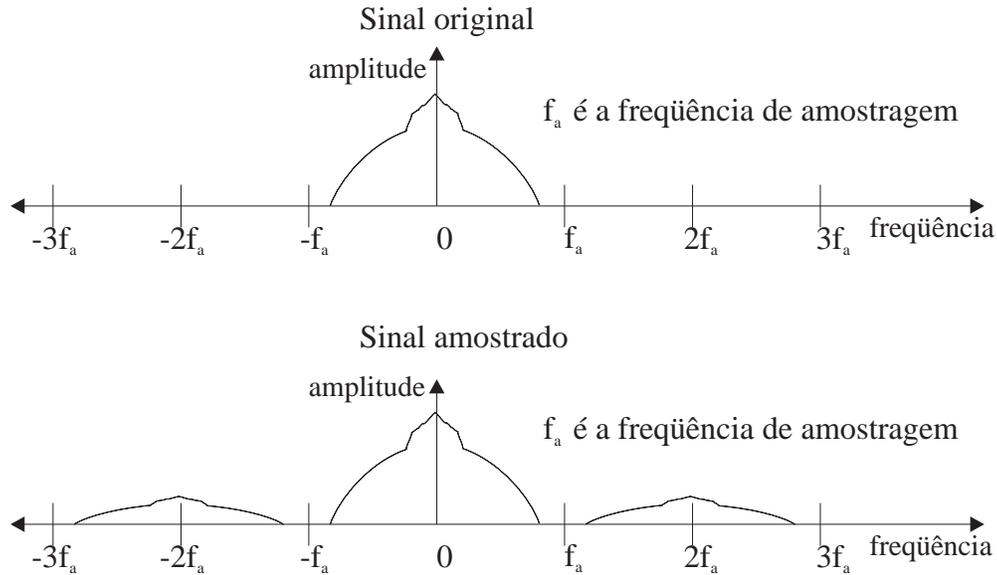


Figura 2.4: Espectro dos sinais original (a) e amostrado (b).

Para recuperar o sinal original basta remover as cópias do espectro do sinal original centradas nas frequências $k/(\Delta t)$ com $k \neq 0$ do espectro do sinal amostrado. A operação de remoção de partes do espectro de potência de um sinal é bastante conhecida tanto sob o ponto de vista teórico quanto prático. Esta operação é conhecida como filtragem. Em particular esta filtragem é conhecida como filtragem passa baixa por que preserva apenas o espectro do sinal original, que ocupa a porção de frequências baixas do sinal amostrado.

Quando este método de recuperação do sinal original é usado a condição do teorema da amostragem é também necessária. Considere-se por exemplo um sinal com componentes de frequência superior à $1/(2\Delta t)$. O espectro do sinal é ilustrado na figura 2.5.

O espectro do sinal amostrado, conforme descrito anteriormente, é composto de uma série de cópias do sinal original centradas nas frequências $k/(\Delta t)$. Tal espectro está ilustrado na figura 2.6

Pode-se verificar que as componentes de altas frequências (frequência maiores que $1/(2\Delta t)$) sobrepõem-se ao espectro original do sinal. Desta forma uma filtragem passa baixa não

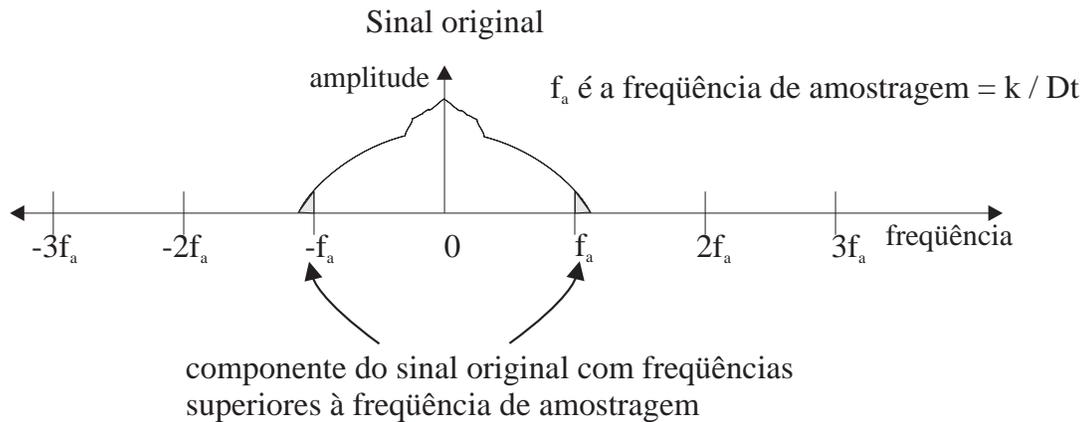


Figura 2.5: Espectro de um sinal que não satisfaz às condições do teorema da amostragem

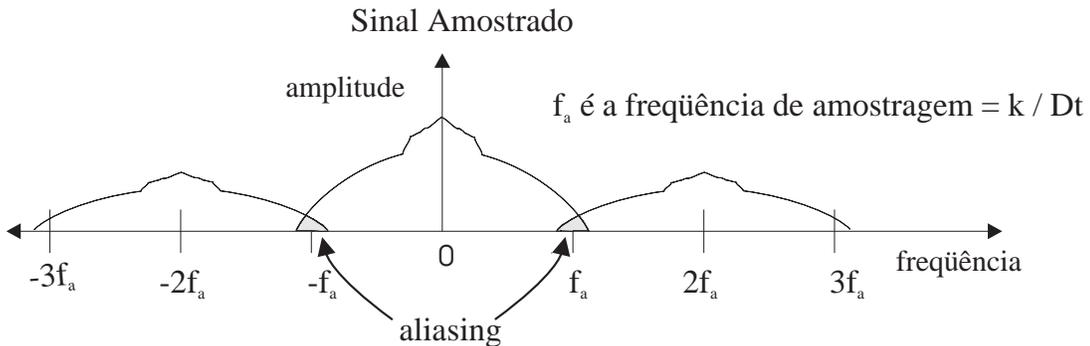


Figura 2.6: Espectro de um sinal que não satisfaz às condições do teorema da amostragem após a realização de amostragem

consegue eliminar tais componentes de altas frequências que se infiltram no espectro original. Tais componentes causam uma distorção no sinal após sua recuperação. Este ruído é chamado de *aliasing*.

2.5.2 Teoria da Amostragem no Caso de Imagens

O teorema da amostragem é bastante utilizado para a realização de processamento digital de sinais.

Considere-se por exemplo um sinal de áudio convertido em sinal elétrico através de um microfone. Este sinal de áudio tem um espectro de potência que não é de banda limitada. Apesar do sinal de uma música ou de uma voz terem bandas limitadas, todo sinal apresenta também ruído que se distribui através de todo o espectro de frequências. Portanto, mesmo em altas frequências o sinal apresenta alguma potência.

Desta forma o sinal original não satisfaz às condições do teorema de amostragem e, por-

tanto não pode ser recuperado à partir de suas amostras através de filtragem passa baixas. Entretanto, o sinal pode ser pré-filtrado antes da amostragem. Assim, não apenas o sinal fica mais limpo, isto é, livre do ruído de altas frequências, como também passa a ser de banda limitada. Após a pré-filtragem o sinal é amostrado. Neste instante está assegurado que as amostras do sinal realmente o representam. Logo, qualquer processamento digital linear pode ser realizado nas amostras. Após o processamento se obtém uma nova sequência de amostras. Esta sequência tem que ser filtrada (passa baixa) para que o sinal correspondente a ela seja obtido. Os passos do processamento digital de um sinal estão ilustrados na figura 2.7

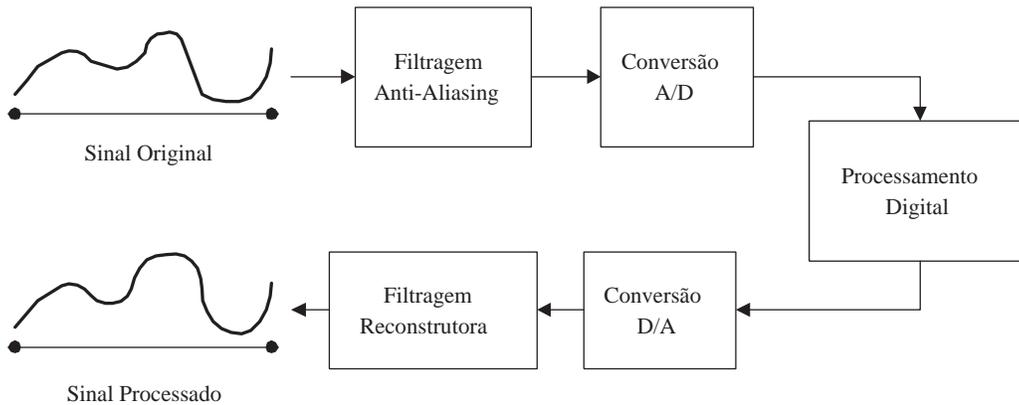
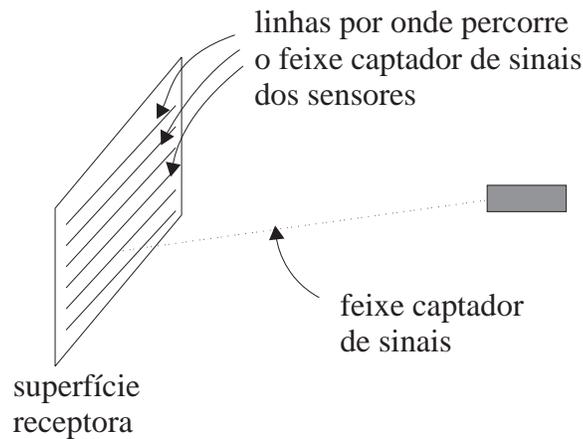


Figura 2.7: Esquema de processamento digital de sinais

No caso de imagens os processos não são exatamente como descritos nos parágrafos anteriores. Existe uma diferença fundamental entre como uma fonte de imagens e uma fonte de áudio são analisadas inicialmente. O sinal de áudio está disponível em sua forma original para ser pré-filtrado. O sinal de imagem, por outro lado, já é obtido amostrado na maioria, senão em todas, as aplicações. Este é o caso da televisão e do cinema por exemplo. Isto acontece porque nos processos de captação de imagens os dados dos sensores não são obtidos continuamente no espaço e no tempo.

No caso de câmeras de TV analógicas, embora o sensoriamento seja contínuo, a coleta dos sinais para composição das imagens é discreta. Mais especificamente, os sensores estão distribuídos continuamente na superfície receptora (sensoriamento contínuo no espaço) e transformam informação luminosa em informação elétrica continuamente (não apenas em instantes de tempo determinados). A captura dos dados dos sensores, entretanto, é realizada através de um feixe que percorre a superfície receptora, tal como os feixes de um tubo de raios catódicos. A figura 2.8 ilustra este processo de captura de imagens. Esta varredura implica que a única dimensão na qual a coleta de dados é contínua é a espacial horizontal. Logo a própria captação implica na amostragem espacial em uma das dimensões da superfície receptora e na amostragem temporal.

No caso de câmeras de TV que utilizam CCD's a amostragem ocorre na própria distribuição discreta dos sensores (malha retangular de sensores distribuídos na superfície receptora



Amostragem espacial na direção vertical porque somente pontos pertencentes às linhas percorridas têm seus sinais captados.

Amostragem temporal porque os sinais de cada pontos são captados de tempos em tempos.

Figura 2.8: Captação de dados de sensores em câmeras analógicas

que transforma informação luminosa em elétrica em instantes determinados de tempo).

Portanto, como a amostragem ocorre na captação das imagens, não há a possibilidade de pré-filtragem dos sinais de fontes de imagens antes de torná-las discretas. Portanto caso o sinal original não satisfaça as condições de limitação de banda o processo de amostragem implicará em impossibilidade de reconstrução (perda irrecuperável do sinal), conhecida como *aliasing*.

Para evitar os efeitos indesejados do *aliasing* deve-se escolher taxas de amostragem tais que as imagens visíveis a olho nú satisfaçam as condições de limitação de banda. Admite-se os efeitos de *aliasing* em situações em que as imagens não são visíveis a olho nú, como nos casos de imagens com movimentos muito rápidos (observadas borradas a olho nú) e imagens com detalhes muito finos. As primeiras apresentam componentes de frequência temporal muito elevadas, enquanto que as últimas apresentam frequências espaciais além do limite de amostragem com recuperação das câmeras.

A determinação das taxas de amostragem das imagens depende, portanto, das faixas de frequência perceptíveis pela visão humana.

Taxa da Amostragem Temporal

Em termos de frequências temporais a visão humana é limitada pela resposta temporal das células foto-receptoras. Para que um sinal temporal discreto seja visto da mesma forma que um contínuo é preciso que dois efeitos perceptuais sejam garantidos:

- continuidade de movimento
- ausência de cintilação

Considerando continuidade de movimentos, as amostras têm de ser tomadas de forma que a sequência de imagens amostradas tenha a mesma noção perceptual de movimento que a imagem contínua original. Estudos perceptuais indicam que são necessárias pelo menos 24 amostras por segundo para garantir a continuidade de movimento de uma cena. O cinema, por exemplo, utiliza exatamente 24 amostras por segundo na representação de suas cenas.

Sob o ponto de vista da cintilação deve-se considerar que cada ponto observado de uma cena é uma fonte de luz. A apresentação de diversas amostras de imagens faz com que estas fontes de luz emitam luz de forma alternada. Enquanto uma amostra está sendo apresentada o ponto gera luz. No intervalo de tempo em que o ponto não está apresentando uma amostra ele está apagado. Esta alternância de emissão de luz causa o fenômeno de cintilação. O olho humano percebe uma flutuação na luz da sequência de amostras a não ser que a frequência de apresentação seja suficientemente alta. A frequência de apresentação depende do iluminamento total recebido pelo olho. Estudos perceptuais levaram a uma lei, conhecida como lei de Ferry Porter, que estabelece que a frequência necessária para apresentação é proporcional ao logaritmo da iluminação ambiente média ([184]).

Em termos práticos considera-se que uma frequência de apresentação de amostras de 60 Hz é suficiente para evitar a cintilação em qualquer nível de iluminação.

As aplicações de imagens usam em geral um importante artifício para apresentarem cenas: as amostras são colhidas a uma frequência f e apresentadas com uma frequência $2f$. Uma mesma amostra é apresentada duas vezes. Assim a imagem resultante apresenta continuidade tanto de movimentos quanto de iluminação.

Desta forma observa-se que a amostragem temporal aproxima razoavelmente um processo estocástico contínuo por uma sequência de variáveis aleatórias, ou processo estocástico discreto.

Taxa da Amostragem Espacial

Para o caso espacial a taxa de amostragem da visão humana é de aproximadamente 120 ciclos por grau pois existem cerca de 120 células foto-receptoras por grau ([162]). Segundo estimativa apresentada no artigo [162], esta amostragem implica na utilização de pelo menos 512×512 amostras uniformemente espalhadas pela superfície foto-receptora para imagens observadas inteiramente na fóvea.

Conclusão

Desde que a amostragem seja suficientemente detalhada para que imagens visíveis a olho nú sejam recuperadas, isto é, as condições citadas nas sub-seções anteriores sejam atendidas,

sua aproximação do sinal original contínua é aceitável (cerca de 30 amostras por segundo contendo 512×512 pontos distribuídos numa malha retangular da superfície receptora).

Esta aproximação implica na discretização da região aonde as cores devem ser determinadas para a caracterização das imagens, o que simplifica bastante suas representações.

Aproveitando esta simplificação, na seqüência deste trabalho consideraremos as imagens amostradas, tanto no espaço quanto no tempo. As amostras temporais são separadas entre si por intervalos δt , enquanto que as amostras espaciais são tomadas em em reticulado $N \times M$ da superfície receptora

Do ponto de vista de modelagem de fontes de imagens os valores de δt , N e M não são importantes. O importante é que cada coordenada de cor de uma fonte de imagens será modelada através de $N \times M$ seqüências de variáveis aleatórias. Cada uma destas variáveis aleatórias corresponde à seqüência de valores da intensidade luminosa em um dado ponto da superfície receptora nos intervalos múltiplos de δt . Desta forma cada coordenada de cor corresponde a uma fonte que gera a cada instante uma matriz $N \times M$ cores.

2.6 Quantização no Espaço de Cores

O modelo de fontes de imagens descrito na seção anterior já é bem mais simples que o modelo apresentado originalmente na seção 2.3.1. Fontes dentro deste modelo podem ser caracterizadas por uma seqüência de coeficientes ou por um sinal unidimensional. Apesar disto a saída de uma fonte ainda não pode ser caracterizada por um número finito de dígitos binários (bits). Isto porque cada imagem estática é composta de matrizes de números reais.

As componentes de cores são caracterizadas por números reais. Para que as cores possam ser representadas por bits é preciso que haja uma quantização. Seja, o volume V em R^n onde as cores estão definidas. Este volume é dado por:

$$V = \{(\mathcal{S}_1[c], \dots, \mathcal{S}_n[c]) \text{ tal que } c \in \mathcal{C}\}$$

onde \mathcal{C} é o espaço de cores.

Uma quantização consiste em dividir este volume V em um número finito de células duas a duas disjuntas C_i . Todas as cores pertencentes a uma dada célula são representadas por uma cor característica da célula. Como há um número finito de células, as cores (ou melhor a aproximação das cores) podem ser representadas por um número finito de bits.

A quantização é uma transformação que leva muitos elementos de um conjunto (em verdade uma célula C_i) em apenas um elemento. Esta transformação implica em perda de informação do sinal original. Esta perda de informação corresponde a um *erro de quantização*. Para uma cada cor $q \in V$ contida na célula C_q o erro de quantização é dado por:

$$e_Q = d(q, \text{cor característica de } C_q)$$

onde d é a uma métrica no espaço de cores. Seja c_q a cor característica da célula C_q .

Uma quantização ótima minimiza o erro de quantização. A determinação da quantização ótima depende da divisão do volume V (correspondente ao espaço de cores) em células. Seja p a função que associa a cada cor em V a sua probabilidade de ocorrência. A escolha ótima

das células C_i consiste na minimização da esperança matemática do erro de quantização no conjunto de cores. Em linguagem matemática, busca-se minimizar o valor D :

$$D = E[d(q, c_q)] = \int_V d(q, c_q) p(q) dq$$

Apesar de introduzir erro, a quantização implica em redução da quantidade de informação necessária para caracterizar uma imagem. A quantização ainda permite que o modelo de fontes de imagens, e em particular as imagens (amostras da fonte), possam ser processadas através de algoritmos discretos.

A implementação de algoritmos discretos é realizada através de processadores digitais. A utilização de processadores digitais aumenta bastante a flexibilidade e a capacidade computacional disponíveis para os algoritmos. Isto, por sua vez permite que algoritmos mais sofisticados de compressão de imagens possam ser realizados no “mundo digital”.

As câmeras e dispositivos de apresentação de imagens não trabalham normalmente com imagens quantizadas. Em outras palavras, a quantização não é uma aproximação decorrente da tecnologia de captação e apresentação de imagens. Ao invés disto, a quantização é uma aproximação utilizada para empregar os algoritmos e a tecnologia digital nas imagens. Para que tal emprego seja realizado as imagens não quantizadas precisam passar pelo processo de quantização.

Uma imagem representada com as aproximações descritas até este ponto do trabalho, isto é, com suas cores representadas por componentes, com amostragem temporal e espacial e, finalmente, com os valores das cores quantizados, é dita uma *imagem digital*.

As fontes de imagens digitais são processos estocásticos discretos \mathcal{X}_m com alfabeto único \mathcal{A} contendo elementos formados por matrizes de números inteiros. Os números inteiros são usados para representar os volumes V_i .

A forma mais simples de realizar a quantização é dividir o volume V em paralelepípedos retangulares. Esta quantização é o resultado da quantização independente de cada uma das componentes de cores. Cada componente de cor tem seus valores definidos num intervalo real I_k , $k = \{1, \dots, n\}$. Os intervalos I_k são divididos em subintervalos I_{kj} com $j \in S_k$ (S_k subconjunto limitado de N). Os subintervalos são dois a dois disjuntos e cobrem todo o intervalo I_k correspondente. Em linguagem matemática escreve-se:

$$\forall k \in \{1, \dots, n\}$$

$$I_k = \bigcup_{j \in S_k} I_{kj}$$

$$I_{kj} \cap I_{ki} = \emptyset, \forall i, j \in S_k, i \neq j$$

As cores na célula $I_{1,j_1} \times \dots \times I_{n,j_n}$ são representadas por apenas uma cor dentro deste volume (normalmente o centro do volume). Desta forma a quantização no espaço de cores herda as características da quantização realizada em cada coordenada de cor. Este tipo de quantização é chamado de quantização escalar.

Na quantização escalar cada cor é representada por uma n-upla ordenada de inteiros. Desta forma uma imagem é composta de n matrizes, cada matriz corresponde a uma coordenada de cor.

Quando a quantização corresponde a uma divisão do espaço de cores em células V_i que não são paralelepípedos a quantização é dita vetorial.

A quantização vetorial apresenta uma grande vantagem em relação à quantização escalar. A maior flexibilidade na escolha do formato das células permite a obtenção de menores esperanças matemáticas de erros de quantização no conjunto de cores. Este ganho não vem sem esforço. A quantização vetorial exige mais processamento tanto na determinação das células a serem utilizadas quanto na caracterização de qual a célula de uma dada cor. A vantagem comparativa da quantização vetorial é tão maior quanto maior for a dependência estatística entre as coordenadas de cor.

A forma mais comum de representar imagens digitais consiste em considerar três componentes de cores (quantização escalar). Uma das componentes é chamada componente de luminância, enquanto que as outras duas componentes são conhecidas por componentes de crominância. A componente de luminância corresponde a imagem observada sem cores, como no caso de televisores ou fotografias em preto-e-branco. As outras componentes trazem a informação adicional necessária para a percepção humana de cores.

Esta divisão em três componentes vem desde o tempo da introdução de televisão a cores. Ela tem características muito interessantes para a representação de imagens. Inicialmente a informação de cor fica separada da informação de luminância. Desta forma, as aplicações que não necessitam de cores podem se restringir apenas a uma componente de cor isoladamente.

Além disto, estudos de percepção visual humana (vide seção 5.4.5) mostram que a visão humana é mais sensível a variações de luminância do que a variações de crominância. Em outras palavras, a percepção humana de cores tem características passa-baixas mais acentuadas que a percepção de luminância. Assim, a amostragem espacial das componentes de cores pode ser menos detalhada que a amostragem espacial de luminância.

Outra característica importante da representação de cores em uma coordenada de luminância e duas de crominância é o baixo correlacionamento estatístico existente entre estas três coordenadas (vide seção 4.5.1).

Para o estudo subsequente de codificação, as fontes de imagens são modeladas por processos estocásticos discretos com alfabeto único \mathcal{A} , que é o conjunto de matrizes $i \times j \times n_c$ de números inteiros. i e j são respectivamente o número de linhas e colunas de cada uma das n_c componentes de cor das imagens. Na verdade o conjunto \mathcal{A} é ainda mais restrito. Como o conjunto de cores é dividido num número finito de células, cada componente de cor é também dividida num número finito de intervalos de quantização. Portanto a representação de uma cor requer apenas um número finito de inteiros. Desta forma, as imagens geradas por uma fonte de imagens digitais pertencem a um conjunto finito de elementos. Seja k o número de inteiros usados para representar cada componente de cor. O número de imagens possíveis será $kij n_c$. Na prática, os valores i, j, k costumam ser potências de dois. Isto facilita o processamento digital em máquinas binárias.

Capítulo 3

Teoria da Codificação

3.1 Introdução

De posse de um modelo para fontes de imagens digitais este capítulo tem como objetivo estabelecer uma base em teoria da codificação sobre a qual se estrutura o restante deste trabalho.

Os resultados encontrados na teoria da codificação não fornecem métodos de codificação de sinais (em particular de imagens). Ao invés disto, tais resultados mostram os desempenhos ótimos que os sistemas de comunicações podem apresentar, isto é, os limites superiores que as codificações de sinais podem atingir nos sistemas de comunicações.

Apesar de não apresentar diretamente métodos eficientes de codificação de sinais, a teoria da codificação é bastante útil. Isto porque ela mostra os compromissos existentes na codificação (mais especificamente fidelidade \times compressão de informação) e, com isto, indica os caminhos que podem levar a codificações eficientes.

Antes da apresentação dos resultados, alguns conceitos relativos a medidas de informação e critérios de fidelidade são definidos.

3.2 Fundamentos

3.2.1 Medidas de Informação

O objetivo da compressão de imagens é reduzir a quantidade de informação necessária para representar imagens. O próprio objetivo fica sem sentido caso a quantidade de informação de uma fonte não possa ser medida.

A unidade de informação normalmente utilizada é o dígito binário ou bit.

A quantidade de bits necessária para caracterizar a saída de uma fonte de imagens sem nenhuma codificação pode ser facilmente medida. Cada imagem isolada (cada amostra temporal) é representada por meio de n_c matrizes $i \times j$ de elementos definidos no conjunto $\{0, \dots, k \Leftrightarrow 1\}$. Para representar cada elemento das matrizes necessita-se de um número de bits maior ou igual a $\log_2 k$. Normalmente k é uma potência de 2. Portanto, para representar a imagem toda são necessários $ijn_c \log_2 k$ bits. Supondo as imagens amostradas a uma taxa

de P imagens por segundo, a representação de uma fonte requer uma vazão de $ijn_cP \log_2 k$ bits por segundo.

Quando os literais i , j , k , n_c e P são substituídos por valores práticos constata-se que a representação de imagens digitais sem codificação necessita de uma quantidade grande de informação.

As duas maiores aplicações de compressão de imagens são o armazenamento e a transmissão de imagens. Num caso prático de armazenamento em computador, por exemplo, cada imagem pode ser representada por uma matriz 512×512 com 24 bits usados para cores. Isto leva à necessidade de mais de 750 Kbytes para armazenar cada imagem. Mesmo considerando que os dispositivos para armazenamento de informação tornam-se mais acessíveis e baratos dia a dia, esta quantidade de informação por imagem é muito grande.

No caso de transmissão de imagens, a necessidade de informação também é grande. Valores práticos para transmissão de televisão a cores são: 512 linhas, 512 colunas, 8 bits para representar cores e 30 imagens por segundo. A vazão necessária para um canal de comunicações transmitir tal fonte de imagens sem codificação é $512 \times 512 \times 8 \times 30$, ou 60 Mbits/s (1Mbit equivale a 1024×1024 bits).

Estes números são muito elevados. Só a título de ilustração vale ressaltar que uma sequência de 10 segundos de imagens armazenadas em computador com estes números resultaria em 200 MBytes. Outra comparação interessante é com a televisão analógica. A TV a cores analógica ocupa um canal de 4.2 MHz. Considerando uma modulação que permita transportar 2 bits por cada Hertz, a transmissão de uma fonte de imagens digitais sem codificação necessitaria de 30 MHz. Desta forma o uso de imagens digitais torna-se pouco atraente sob o ponto de vista econômico (espaço para armazenamento de informação e vazão de um canal de comunicações são ambos recursos caros).

Será que toda esta informação é realmente necessária para representar uma fonte de imagens? Para poder responder a esta pergunta é preciso ter uma ferramenta para medir a informação proveniente de uma fonte.

Shannon em seu trabalho “*The Mathematical Theory of Communication*” ([164]) apresentou tal ferramenta, conhecida como entropia. Outro conceito importante na medição de quantidade de informação de uma fonte é a informação mútua. Esta é a informação que um processo estocástico carrega a respeito de outro. Este conceito é interessante uma vez que uma fonte $\{\mathcal{X}_n\}$ e sua codificação, isto é, a sequência $\{\mathcal{Y}_n\}$ de códigos associados às imagens da fonte podem ser considerados dois processos estocásticos discretos tais que um deles ($\{\mathcal{Y}_n\}$) carrega informação sobre o outro ($\{\mathcal{X}_n\}$).

As definições de tais medidas, bem como o fundamento teórico que fornece a base para que elas realmente sejam consideradas medidas de informação estão apresentados nas próximas seções.

3.2.2 Entropia

Seja uma fonte de imagens $\{\mathcal{X}_n\}$, $n \in Z^+$ com alfabeto finito \mathcal{A} , conforme o modelo visto no capítulo 2. Seja $P(\mathcal{X}_n = a)$ a probabilidade da imagem $a \in \mathcal{A}$ ser escolhida no instante n (na verdade no instante $n \Delta t$).

A *entropia* da variável aleatória \mathcal{X}_n é dada por

$$H(\mathcal{X}_n) = \Leftrightarrow \sum_{a \in \mathcal{A}} P(\mathcal{X}_n = a) \log_2 [P(\mathcal{X}_n = a)] \quad (3.1)$$

onde por definição $0 \cdot \log_2 0 = 0$.

Esta medida corresponde a apenas uma das variáveis aleatórias que compõem uma fonte. Uma medida global da fonte é a taxa de entropia da fonte. Antes de defini-la convém definir a entropia de um número finito de variáveis aleatórias $\{\mathcal{X}_n, \dots, \mathcal{X}_q\}$. Este conjunto de variáveis aleatórias pode ser entendido como um vetor aleatório. A definição de entropia facilmente estende-se para este vetor:

$$H^{n,q}(\{\mathcal{X}_n, \dots, \mathcal{X}_q\}) = \Leftrightarrow \frac{1}{(q \Leftrightarrow n)} \sum_{\vec{a} \in \mathcal{A}^{q-n}} P(\vec{a}) \log_2 [P(\vec{a})] \quad (3.2)$$

onde

$$P(\vec{a}) = P[(\mathcal{X}_n, \dots, \mathcal{X}_q) = \vec{a}] \quad (3.3)$$

A *taxa de entropia*, ou entropia média de uma fonte de $\{\mathcal{X}_n\}$ é definida por:

$$\bar{H}(\mathcal{X}) = \lim_{n \rightarrow \infty} \sup H^{0,n}(\mathcal{X}) \quad (3.4)$$

Para provar que o limite existe para toda fonte com alfabeto finito convém provar inicialmente alguns lemas:

Lema 3.1 *Seja f uma variável aleatória com alfabeto finito $\mathcal{A} = \{a_0, \dots, a_{q-1}\}$ definida no espaço de probabilidade $(\Omega, \mathcal{B}, \mathcal{P})$. A entropia de f satisfaz à inequação:*

$$H(f) \leq \Leftrightarrow \log_2 \left(\frac{1}{q} \right) = \log_2(q)$$

Além disto a inequação é verdadeira se e somente se cada elemento a_i de \mathcal{A} for tal que $\mathcal{P}[f = a_i] = \frac{1}{q}$.

Demonstração:

Seja \mathcal{V} o conjunto das variáveis aleatórias com alfabeto \mathcal{A} . A entropia de uma variável aleatória f nada mais é do que uma função $H : \mathcal{V} \rightarrow \mathbb{R}$ definida da seguinte forma:

$$H(f) = \Leftrightarrow \sum_{i=0}^{q-1} p_i \log_2(p_i)$$

onde $p_i = P(f = a_i)$.

Seja a função $g : \mathcal{V} \rightarrow \mathbb{R}^q$ que associa a cada variável aleatória f com alfabeto \mathcal{A} um ponto $s_f \in \mathbb{R}^q \mid s_f = (p_0, \dots, p_{q-1})$.

A entropia da variável aleatória f pode ser obtida conhecendo-se o ponto s_f através da função $H_g : R^q \rightarrow R$ definida a seguir:

$$H_g(s_f) = \Leftrightarrow \sum_{i=0}^{q-1} p_i \log_2(p_i)$$

Portanto a entropia H equivale à composição de funções $H_g \circ g$.

Como a soma das probabilidades dos elementos de \mathcal{A} tem de ser 1 a imagem de \mathcal{V} segundo g em R^q é a hiperfície $S^{-1}(1)$, onde $S : (p_0, \dots, p_{q-1}) \mapsto \sum_{i=0}^{q-1} p_i$. Portanto para maximizar a entropia das fontes em \mathcal{V} devemos maximizar H_g em $S^{-1}(1)$.

Segundo o teorema do multiplicador de Lagrange H_g terá um ponto crítico em $s \in S^{-1}(1) \Leftrightarrow \exists k \in R \mid \nabla H_g(s) = k \nabla S(s)$, isto é, os gradientes de H_g e S forem paralelos no ponto s .

Os gradientes das funções H_g e S valem respectivamente:

$$\nabla H_g(s_f) = \Leftrightarrow (\log_2 p_0 + \frac{1}{\ln 2}, \dots, \log_2 p_{q-1} + \frac{1}{\ln 2}) \quad (3.5)$$

$$\nabla S(s_f) = (1, \dots, 1) \quad (3.6)$$

É fácil de verificar que $\nabla H_g(s_f)$ e $\nabla S(s_f)$ são paralelos se e só se $p_0 = p_1 = \dots = p_{q-1} = \frac{1}{q}$. Portanto o ponto $s_0 = (\frac{1}{q}, \dots, \frac{1}{q})$ é o único ponto crítico da função H_g em $S^{-1}(1)$. Para ver se este ponto crítico é de máximo ou mínimo podemos calcular H_g em algum outro ponto $S^{-1}(1)$. Escolhemos o ponto $(1, 0, \dots, 0)$. Neste ponto H_g vale $0 \leq \log_2(q) = H_g(s_0) \Rightarrow s_0$ é ponto de máximo da função H_g em $S^{-1}(1) \Rightarrow \log_2(q) = H_g(s_0)$ maximiza a entropia em \mathcal{V} . Como s_0 é o único ponto de máximo de H_g em $S^{-1}(1)$ a entropia só atinge o valor máximo somente quando $p_0 = p_1 = \dots = p_{q-1} = \frac{1}{q}$ ♠

- **Lema 2.2** : Seja $\{\mathcal{X}_n\}$ uma fonte com alfabeto finito $\mathcal{A} = \{a_0, \dots, a_{q-1}\}$. A taxa de entropia da fonte satisfaz à inequação:

$$0 \leq \bar{H}(\mathcal{X}) \leq \log_2(q)$$

Demonstração:

A demonstração está baseada no fato da entropia de vetores aleatórios de qualquer dimensão de elementos da fonte \mathcal{A} estar sempre limitada a $\log_2(q)$, isto é, $H^{0,n}(\mathcal{X}) \leq \log_2(q) \forall n > 0$. Este fato decorre do lema anterior.

De fato, dado um vetor aleatório $\{\mathcal{X}_1, \dots, \mathcal{X}_n\}$ de dimensão n cujos componentes tomam valores no alfabeto \mathcal{A} , define-se uma variável aleatória $\mathcal{X}^{(n)}$ tomando valores no alfabeto \mathcal{A}^n da seguinte forma:

$$\mathcal{X}^{(n)} = (\mathcal{X}_1, \dots, \mathcal{X}_n)$$

A entropia da variável $\mathcal{X}^{(n)}$ é dada por:

$$H(\mathcal{X}^{(n)}) = \quad (3.7)$$

$$\Leftrightarrow \sum_{\vec{a} \in \mathcal{A}^n} P(\mathcal{X}^{(n)} = \vec{a}) \log_2[P(\mathcal{X}^{(n)} = \vec{a})] \quad (3.8)$$

Esta entropia está relacionada com a entropia do vetor aleatório, definida em 3.2, da seguinte forma:

$$H^{1,n} = \frac{1}{n}H(\mathcal{X}^{(n)}) \quad (3.9)$$

Segundo o lema anterior, $H(\mathcal{X}^{(n)})$, sendo uma variável aleatória tomando valores numa fonte de q^n elementos, têm sua entropia limitada por $\log_2(q^n) = n\log_2(q)$. Portanto, $H^{1,n}$ está limitada a $\log_2(q)$, $\forall n > 0$.

Voltando à definição da taxa de entropia \bar{H} na equação 3.4, fica claro que se cada $H^{1,n}$ estiver limitada a $\log_2(q)$ o limite supremo \bar{H} também estará. ♠

Se o processo for estacionário no sentido estrito então a taxa de entropia é igual a entropia de cada variável aleatória \mathcal{X}_n que compõe o processo estocástico de fonte.

A importância destas medidas reside no teorema de codificação de fontes sem a presença de ruído, devido a Shannon. O teorema, que está provado em [164] afirma o seguinte:

Teorema 3.1 Teorema de Codificação de Fontes sem Ruído: *Seja uma fonte com entropia média H bits por símbolo e um canal de comunicações com capacidade (vazão) de C bits por segundo. Então, $\forall \epsilon > 0$ é possível codificar a saída da fonte a uma taxa média de $C/H \pm \epsilon$ símbolos por segundo através do canal. Por outro lado, não é possível codificar a fonte de forma a transmitir informação a uma taxa superior a C/H símbolos por segundo.*

Portanto a taxa de entropia de uma fonte assume o papel de número mínimo de bits necessários para transmitir cada símbolo (no caso imagem) de uma fonte sem que haja perda de informação.

3.2.3 Informação Mútua

A entropia mede a quantidade de informação intrínseca de uma fonte. Outra medida importante na teoria da codificação é a *informação mútua*. Esta mede o quanto de informação uma variável aleatória \mathcal{X} carrega a respeito de outra \mathcal{Y} .

Sejam duas variáveis aleatórias \mathcal{X} e \mathcal{Y} com alfabetos respectivamente $\mathcal{A}_X = \{x_0, x_1, \dots, x_{m_x-1}\}$ e $\mathcal{A}_Y = \{y_0, y_1, \dots, y_{m_y-1}\}$, ambos finitos. A partir destes processos define-se o processo composto $\mathcal{X}\mathcal{Y}$, com alfabeto $\mathcal{A}_X \times \mathcal{A}_Y$.

Inicialmente deseja-se dimensionar a informação que a ocorrência do evento $[\mathcal{Y} = y_j]$ traz sobre o evento $[\mathcal{X} = x_k]$. Define-se

$$I_{\mathcal{X};\mathcal{Y}}(x_k, y_j) = \log_2 \frac{P_{\mathcal{X}|\mathcal{Y}}(x_k | y_j)}{P_{\mathcal{X}}(x_k)} \quad (3.10)$$

como a medida desta informação.

Caso os eventos $[\mathcal{Y} = y_j]$ e $[\mathcal{X} = x_k]$ sejam independentes $P_{\mathcal{X}|\mathcal{Y}}(x_k | y_j) = P_{\mathcal{X}}(x_k)$. Desta forma o valor da medida fica 0, pois é igual ao $\log_2 1$. Por outro lado, se o evento $[\mathcal{Y} = y_j]$ implicar no evento $[\mathcal{X} = x_k]$ a medida vale $\log_2 \frac{1}{P_{\mathcal{X}}(x_k)}$, que é seu valor máximo. Portanto a medida de informação definida varia entre o valor mínimo 0 (para o caso de eventos independentes) e o valor máximo (para o caso de eventos totalmente correlacionados).

A informação mútua entre as duas variáveis \mathcal{X} e \mathcal{Y} é definida como:

$$I_{\mathcal{X};\mathcal{Y}} = \sum_{k=0}^{m_x-1} \sum_{j=0}^{m_y-1} P_{\mathcal{X}\mathcal{Y}}(x_k, y_j) I_{\mathcal{X};\mathcal{Y}}(x_k, y_j) \quad (3.11)$$

$$= \sum_{k=0}^{m_x-1} \sum_{j=0}^{m_y-1} P_{\mathcal{X}\mathcal{Y}}(x_k, y_j) \log_2 \frac{P_{\mathcal{X}|\mathcal{Y}}(x_k | y_j)}{P_{\mathcal{X}}(x_k)} \quad (3.12)$$

No caso das variáveis serem independentes, todos os eventos $[\mathcal{Y} = y_j]$ e $[\mathcal{X} = x_k]$ são independentes entre si. Portanto todos os valores $I_{\mathcal{X};\mathcal{Y}}(x_k, y_j)$ são nulos e a informação mútua é nula.

No outro extremo, se as variáveis forem totalmente correlacionadas, os valores $I_{\mathcal{X};\mathcal{Y}}(x_k, y_j)$ são iguais a $\log_2 \frac{1}{P_{\mathcal{X}}(x_k)}$ e a informação mútua é igual a entropia das variáveis aleatórias. Vale observar que as entropias das variáveis são idênticas nestes casos.

Outro conceito importante é o de entropia condicional:

$$H(\mathcal{X} | \mathcal{Y}) = \Leftrightarrow \sum_{k=0}^{m_x-1} \sum_{j=0}^{m_y-1} P_{\mathcal{X}\mathcal{Y}}(x_k, y_j) \log_2 P_{\mathcal{X}|\mathcal{Y}}(x_k | y_j) \quad (3.13)$$

A entropia condicional mede a informação que resta à variável \mathcal{X} dado que a variável \mathcal{Y} é conhecida.

Partindo destas definições e da definição de entropia não é difícil chegar às equações:

$$I_{\mathcal{X};\mathcal{Y}} = H(\mathcal{X}) \Leftrightarrow H(\mathcal{X} | \mathcal{Y}) \quad (3.14)$$

$$= H(\mathcal{Y}) \Leftrightarrow H(\mathcal{Y} | \mathcal{X}) \quad (3.15)$$

$$H(\mathcal{Y}) = I_{\mathcal{X};\mathcal{Y}} + H(\mathcal{Y} | \mathcal{X}) \quad (3.16)$$

$$H(\mathcal{X}) = I_{\mathcal{X};\mathcal{Y}} + H(\mathcal{X} | \mathcal{Y}) \quad (3.17)$$

Estas equações têm uma interpretação intuitiva interessante: A informação contida numa variável aleatória (sua entropia) pode ser decomposta numa parcela correspondente à informação que outra variável carrega sobre a primeira (informação mútua) e numa outra parcela correspondente à informação que resta à primeira variável após o conhecimento da segunda.

O lema abaixo estabelece limites para o valor da informação mútua entre duas variáveis aleatórias:

Lema 3.2 *Sejam \mathcal{X} e \mathcal{Y} duas variáveis aleatórias com alfabetos finitos definidas sobre o mesmo espaço de probabilidade. Então:*

$$0 \leq I_{\mathcal{X};\mathcal{Y}} \leq \min(H(\mathcal{X}), H(\mathcal{Y}))$$

Demonstração:

Como a entropia de qualquer variável aleatória é não negativa, de 3.14 e 3.15 conclui-se respectivamente que

$$I_{\mathcal{X};\mathcal{Y}} \leq H(\mathcal{X}) \quad (3.18)$$

$$I_{\mathcal{X};\mathcal{Y}} \leq H(\mathcal{Y}) \quad (3.19)$$

e, portanto,

$$I_{\mathcal{X};\mathcal{Y}} \leq \min(H(\mathcal{X}), H(\mathcal{Y})) \quad (3.20)$$

Para provar que a informação mútua é uma grandeza não negativa é preciso voltar à sua definição: 3.11. Cada termo $P_{\mathcal{X}\mathcal{Y}}(x_k, y_j)I_{\mathcal{X};\mathcal{Y}}(x_k, y_j)$ do somatório é formado pelo produto de uma probabilidade (por definição não negativa) e de $I_{\mathcal{X};\mathcal{Y}}(x_k, y_j)$. Mas $I_{\mathcal{X};\mathcal{Y}}(x_k, y_j)$, definido em 3.10, é também não negativa porque

$$P_{\mathcal{X}|\mathcal{Y}}(x_k | y_j) \geq P_{\mathcal{X}}(x_k)$$



A noção de informação mútua é importante quando se codifica uma variável aleatória. Uma *codificação* é uma função $f : \mathcal{A}_X \rightarrow \mathcal{A}_Y$ que leva o alfabeto \mathcal{A}_X da variável \mathcal{X} no alfabeto de codificação \mathcal{A}_Y . O resultado da codificação $f(\mathcal{X})$ é outra variável aleatória. A informação mútua neste caso mede o quanto de informação sobre \mathcal{X} a codificação $f(\mathcal{X})$ carrega.

A medida de informação mútua não é tão intuitiva quanto a entropia ou a taxa de entropia. No entanto ela é útil para o prosseguimento do desenvolvimento teórico.

3.2.4 Critério de Fidelidade

Diversas vezes durante o decorrer deste trabalho menciona-se a qualidade de uma representação de imagens. A caracterização de qualidade é também um tópico importante na teoria da codificação.

Para definir a qualidade de representação da fonte $\{\mathcal{X}_n\}_n$ através da fonte $\{\mathcal{Y}_n\}_n$ (onde $\mathcal{Y}_n = f(\mathcal{X}_n)$) define-se inicialmente uma distorção entre um símbolo x e sua representação $y = f(x)$. Esta distorção pode ser entendida como o custo de aproximar x por $f(x)$.

É importante salientar que esta distorção é representada através de uma função $d : \mathcal{A}_X \times \mathcal{A}_Y \rightarrow R^+$. Esta função não precisa ser uma distância no sentido topológico, isto é, não precisa ser simétrica em relação a suas coordenadas e nem precisa atender à desigualdade triangular. Esta função deve quantificar a distorção na representação de um símbolo x por outro y . Deseja-se que a função distorção atenda a três requisitos:

- Ela deve ser tratável sob o ponto de vista teórico para que se possa desenvolver teoria a partir dela.
- Ela não deve requerer um custo computacional elevado.
- Ela deve traduzir em números a distorção perceptual entre 2 símbolos de uma fonte.

Estes três requisitos são muito difíceis de serem atendidos simultaneamente. A maior dificuldade situa-se no terceiro item. No caso de imagens é bastante difícil quantificar a distorção perceptual entre duas imagens. Existem vários estudos objetivando determinar uma função de distorção para a visão humana. Alguns destes estudos encontram-se em [162], [109] e [70].

A função de distorção varia não só entre fontes de imagens diferentes. Até mesmo dentro da classe de fontes de imagens esta função pode variar. O que acontece é que diferentes aplicações podem ter diferentes critérios para avaliar a distorção entre imagens.

Sob o ponto de vista teórico é suficiente modelar uma distorção como uma função $d : \mathcal{A}_X \times \mathcal{A}_Y \rightarrow R^+$. Pode-se medir distorções em dimensões maiores. Nestes casos a função de distorção $d_n : \mathcal{A}_X^n \times \mathcal{A}_Y^n \rightarrow R^+$.

Um critério de fidelidade $\{\rho_n\}$ é uma sequência de funções de distorção $d_n : \mathcal{A}_X^n \times \mathcal{A}_Y^n \rightarrow R^+$. A fidelidade entre dois vetores de símbolos X^n e Y^n é definida aplicando a distorção d_n , isto é, $d_n(X^n, Y^n)$. O critério de fidelidade entre duas amostras X e Y dos processos estocásticos $\{\mathcal{X}_n\}$ e $\{\mathcal{Y}_n\}$ respectivamente é definido por

$$\rho(X, Y) = d_\infty(X, Y) = \limsup_{n \rightarrow \infty} \frac{1}{n} d_n(X^n, Y^n)$$

Neste trabalho considera-se as funções de distorção d_n aditivas, isto é, as distorções são baseadas na distorção d_0 de forma que :

$$d_n(X^n, Y^n) = \sum_{i=0}^{n-1} d_0(X_i, Y_i)$$

Nestes casos o critério de fidelidade existe caso os dois processos $\{\mathcal{X}_n\}$ e $\{\mathcal{Y}_n\}$ sejam estacionários ([72]).

Definido o critério de fidelidade pode-se definir a distorção da aproximação do processo $\{\mathcal{X}_n\}$ pelo processo $\{\mathcal{Y}_n\}$ como sendo a esperança matemática do critério de fidelidade ρ quando a sequência amostrada X varia no processo $\{\mathcal{X}\}$.

$$D = E_X[\rho(x, f(x))]$$

Segue de [72] que a distorção está bem definida quando os dois processos são estacionários e a distorção é aditiva.

3.3 Função Taxa-Distorção

Uma ferramenta importante na teoria da codificação é a função Taxa-Distorção. Esta função estabelece uma relação entre a quantidade de informação sobre o processo estocástico $\{\mathcal{X}_n\}_n$ que um processo $\{\hat{\mathcal{X}}_n\}_n$ (tipicamente uma codificação de $\{\mathcal{X}_n\}_n$) carrega e o mínimo de distorção a que se pode chegar representando $\{\mathcal{X}_n\}_n$ por $\{\hat{\mathcal{X}}_n\}_n$.

Seja o processo $\{\mathcal{X}_n\}_n$ estacionário no sentido estrito com distribuição μ para cada variável aleatória \mathcal{X}_n . O processo tem alfabeto \mathcal{A}_X . Considere-se também um processo $\{\hat{\mathcal{X}}_n\}_n$ com o alfabeto $\hat{\mathcal{A}}_X$. Seja um critério de fidelidade ρ com funções de distorção definidas em $\mathcal{A}_X^n \times \hat{\mathcal{A}}_X^n$.

Seja $R_n(R, \mu^n)$ a coleção de todas as distribuições p^n para os vetores aleatórios $\mathcal{X}_n \times \hat{\mathcal{X}}_n$ tais que:

1. p^n tem como distribuição marginal na primeira coordenada μ^n , isto é, $p^n(F \times \hat{\mathcal{A}}_X^n) = \mu^n(F) \forall F \in B_{\mathcal{A}_X}^n$. $B_{\mathcal{A}_X}^n$ é o conjunto de Borel do alfabeto \mathcal{A}_X^n .

2. A informação mútua satisfaz:

$$\frac{1}{n} I_{p^n}(\mathcal{X}_n; \hat{\mathcal{X}}_n) \leq R$$

A interpretação de $R_n(R, \mu^n)$ é a seguinte: $R_n(R, \mu^n)$ contém as distribuições de probabilidades de todos os vetores aleatórios $\mathcal{X}_n \times \hat{\mathcal{X}}_n$ que preservam a distribuição marginal μ^n e mantém a informação mútua limitada por R . Implicitamente nesta coleção estão todas as codificações cujas informações mútuas com \mathcal{X}_n são limitadas por R .

Seja o valor $D_n(R, \mu^n)$ definido por:

$$D_n(R, \mu^n) = \inf_{p^n \in R_n(R, \mu^n)} E_{p^n} \rho_n(\mathcal{X}_n, \hat{\mathcal{X}}_n)$$

Caso $R_n(R, \mu^n) = \emptyset \Rightarrow D_n(R, \mu^n) = \infty$.

D^n é chamada de função de taxa-distorção de ordem n .

A função de taxa-distorção é definida por

$$D(R, \mu) = \limsup_{n \rightarrow \infty} D_n(R, \mu^n) \quad (3.21)$$

Outra propriedade interessante de D e das D_n provada em [72] é:

Teorema 3.2 $\forall \mu$ as funções $D_n(R, \mu^n)$ e $D(R, \mu)$ são não negativas, convexas e, portanto, contínuas em relação à variável R .

Qual o significado da função $D(R, \mu)$?

A função tem dois parâmetros: R e μ . R é a taxa máxima a que se pretende codificar a fonte $\{\mathcal{X}_n\}_n$, cujas variáveis aleatórias \mathcal{X}_n têm distribuição μ .

O conjunto $R_n(R, \mu^n)$ usado na construção da definição de $D(R, \mu)$, tem como objetivo agregar implicitamente os códigos $\hat{\mathcal{X}}_n$ com informação mútua limitada com \mathcal{X}_n . As distribuições p^n podem ser entendidas como as probabilidades de se associar um vetor $\hat{\mathcal{X}}_n$ ao \mathcal{X}_n para codificá-lo.

O que se perde na aproximação de $\{\mathcal{X}_n\}_n$ por $\{\hat{\mathcal{X}}_n\}_n$? É justamente isto que é avaliado em $D_n(R, \mu^n)$. Este último valor é o ínfimo das distorções quando \mathcal{X}_n é aproximado por $\hat{\mathcal{X}}_n$, com as limitações impostas a $\hat{\mathcal{X}}_n$ dadas pela definição do conjunto $R_n(R, \mu^n)$.

Quando n cresce $D_n(R, \mu^n)$ tende ao comportamento de longo prazo da fonte, isto é, $D(R, \mu)$ é o mínimo custo, ou mínimo de distorção, que se consegue na aproximação de uma fonte por um código mantendo a informação mútua entre os dois limitada por R .

3.3.1 Aplicações para Imagens

A teoria da codificação estabelece uma função contínua relacionando a taxa máxima em que se deseja codificar a fonte e a menor distorção possível.

A determinação desta função é bem difícil mesmo para fontes e critérios de fidelidade simples. No caso geral de fontes e critérios de fidelidade a determinação das funções é impraticável.

No caso das imagens, além das dificuldades de cálculo, os modelos de fontes (especificados por suas distribuições μ) e os critérios de fidelidade não são conhecidos. Além disto, o

conhecimento da função $D(R, \mu)$ não diz nada sobre como obter codificações eficientes. A função apenas estabelece o resultado ótimo teórico da codificação.

Apesar de não trazer resultados práticos imediatos a teoria da codificação é útil porque evidencia os pontos importantes para a codificação de uma fonte:

- o conhecimento das características estatísticas da fonte (entropia, informação mútua, ...)
- o conhecimento do critério de fidelidade usado para medir a qualidade da aproximação da fonte pela codificação.

Desta forma fica clara a importância dos estudos de modelos de fontes de imagens e de critérios de fidelidade para imagens. Os dois próximos capítulos são dedicados respectivamente a estes dois estudos.

A utilização da teoria de codificação de forma apenas indireta não é um fenômeno isolado na compressão de imagens. Em várias outras situações a teoria consegue caracterizar critérios de otimização, mas na prática dificuldades de cálculo ou limitações no conhecimento dos modelos inviabilizam a obtenção de valores ótimos. No entanto, assim como no caso da teoria de codificação, muitas vezes o conhecimento da teoria permite caracterizar os pontos mais importantes e direcionar as aplicações práticas.

Capítulo 4

Modelos de Fontes de Imagens

4.1 Introdução

A caracterização probabilística completa de uma fonte de imagens $\{\mathcal{X}_n\}_n$ requer o conhecimento das distribuições de probabilidade de todos os vetores aleatórios $(\mathcal{X}_m, \dots, \mathcal{X}_{m+q})$. Isto significa que para cada dimensão q , para cada instante inicial m e para todo elemento $\vec{a} \in \mathcal{A}_X^q$ é necessário conhecer a probabilidade $P_{(m,q)}(\mathcal{X}_m, \dots, \mathcal{X}_{m+q} = \vec{a})$, isto é, a probabilidade do vetor aleatório $(\mathcal{X}_m, \dots, \mathcal{X}_{m+q})$ gerar o elemento $\vec{a} \in \mathcal{A}_X^q$.

O conhecimento destas distribuições de probabilidade permite o cálculo da taxa de entropia ou entropia média da fonte. Além disto o modelo caracteriza as probabilidades de ocorrência das imagens ou seqüências imagens. Portanto o comportamento das fontes pode ser analisado tanto sob o ponto de vista de variáveis aleatórias isoladas \mathcal{X}_n (imagens estáticas) quanto sob o ponto de vista de vetores aleatórios multidimensionais $(\mathcal{X}_n)_{n=m}^{m+q}$ (imagens dinâmicas).

Infelizmente a determinação das distribuições de probabilidades não é viável na prática.

Primeiramente porque a quantidade de elementos do conjunto \mathcal{A}_X^q é muito grande mesmo para q 's pequenos. Assim a quantidade de imagens necessárias para permitir uma estimativa boa das probabilidades é muito grande.

Além disto não há um consenso sobre qual seria uma fonte de imagens típica.

Na prática usa-se bastante a modelagem local das fontes. Mais especificamente, modela-se, por exemplo, a influência de uma cor (ou intensidade luminosa) sobre seus pontos vizinhos (no tempo e no espaço). A caracterização do comportamento local das fontes fornece apenas indicações sobre o comportamento global. O fato que a presença de uma cor num ponto aumenta a probabilidade de ocorrência de cores próximas nos pontos vizinhos indica que imagens com áreas com poucas variações de cores são mais prováveis que imagens com muitas variações.

Outra abordagem comum na prática é o uso de modelos paramétricos para caracterizar as fontes de imagem. As fontes são observadas inicialmente. Em seguida um modelo probabilístico paramétrico $M(\vec{\theta})$ é sugerido para a fonte. Depois realiza-se uma estimação do parâmetro $\vec{\theta}$ (o parâmetro pode ser multidimensional). A estimação é baseada em amostras da fonte (imagens isoladas e seqüências de imagens). Por fim testa-se a validade do modelo

resultante.

A utilização de modelos paramétricos restringe de forma drástica os graus de liberdade dos modelos de fontes. Nestes casos a estimação de parâmetros equivale à determinação da melhor aproximação da fonte dentre o conjunto dos modelos parametrizados. A figura 4.1 ilustra esta aproximação.

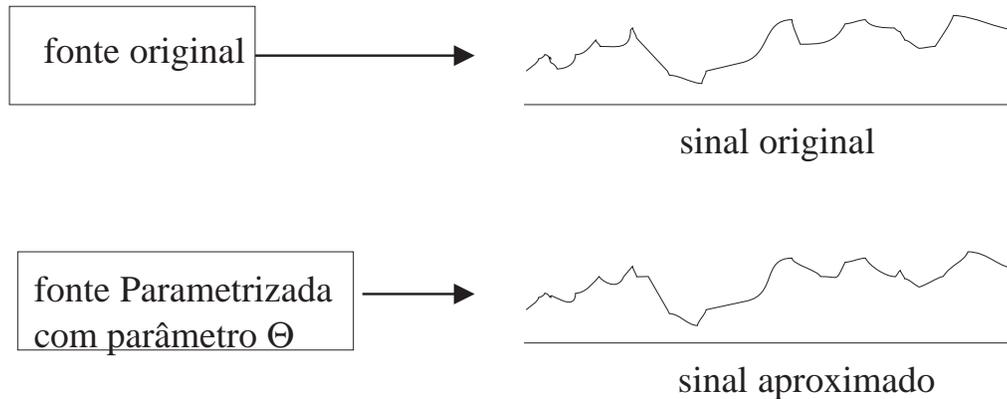


Figura 4.1: Aproximação de uma fonte através de um modelo paramétrico

A combinação de modelos locais e paramétricos fornece informação bem mais pobre sobre as imagens do que o conhecimento das distribuições $P_{(m,q)}(\mathcal{X}_m, \dots, \mathcal{X}_{m+q} = \vec{a})$, $\vec{a} \in \mathcal{A}_{\mathcal{X}}^q$. No entanto, essa é a informação que pode ser agregada sobre as fontes.

Os modelos locais são mais versáteis e podem ser utilizados em ampla gama de fontes de imagens. Primeiramente porque os mesmos modelos locais podem ser usados para fontes com diferentes comportamentos globais.

Além disto, estes modelos podem ser utilizados mesmo em casos de fontes não-estacionárias. De fato as fontes de imagens apresentam não-estacionariedades, isto é, seu comportamento probabilístico varia com o tempo e com a posição espacial. Regiões com movimentos bruscos e áreas com arestas têm comportamentos bastante diferentes de respectivamente regiões sem movimentos e áreas sem arestas, por exemplo. Mesmo assim, seu comportamento é caracterizado de forma bastante adequada por uma estacionariedade local ou quase-estacionariedade. Isto é, embora as fontes não tenham comportamento constante globalmente, seu comportamento é aproximadamente constante em regiões pequenas.

Por outro lado, a quantidade de dados necessária para estimar os modelos locais parametrizados é bem menor que a necessária para os modelos genéricos.

Além disto, as características locais são de importância prática. Isto porque grande parte dos algoritmos utilizados em codificação de imagens dividem as imagens em regiões, que são processadas separadamente. Esta divisão em regiões é útil para reduzir o esforço computacional envolvido na aplicação dos algoritmos nas imagens completas.

Desta forma, as aproximações usadas para modelar as fontes viabilizam pelo menos alguma caracterização estatística de imagens. Esta é mais uma situação em que se faz concessões no modelo ideal em prol da viabilidade prática das aplicações.

Este capítulo procura apresentar os diversos modelos locais sugeridos para as imagens. A apresentação começa com os modelos preditores de imagens estáticas (modelos unidimensionais e bidimensionais). Em seguida vêm os modelos de imagens dinâmicas. Depois são apresentados os modelos de Covariância.

É importante observar que até este ponto os modelos apresentados servem para imagens monocromáticas, isto é, imagens em preto-e-branco, com apenas uma coordenada de cor. A última seção do capítulo (seção 4.5) traz um modelo de correlação de cores. Com base neste modelo de cores, os modelos de imagens coloridas recaem (através de transformação de coordenadas) em n_c modelos de imagens monocromáticas independentes (onde n_c é o número de coordenadas de cor). O modelo de correlação de cores fornece a transformação de coordenadas responsável pelo desacoplamento das coordenadas de cor.

Assim, os algoritmos de codificação de imagens representam as cores em coordenadas decorrelacionadas (ou pelo menos coordenadas próximas das decorrelacionadas), e realizam seus métodos nas n_c coordenadas de forma independente.

4.1.1 Modelos Locais

Dentre os modelos de comportamento local de imagens utilizados destacam-se os de predição de sinais. Estes modelos fazem uma predição dos sinais de imagem em um ponto a partir dos pontos vizinhos (no tempo e/ou no espaço) já conhecidos. A diferença entre o sinal gerado pela fonte e a predição constitui um sinal de erro.

Os algoritmos preditores são parametrizados. Os parâmetros têm que ser estimados para que a predição se aproxime do valor efetivamente gerado pela fonte.

A caracterização estatística das fontes é composta dos parâmetros do modelo e do comportamento estatístico do sinal de erro.

Quando os modelos se adequam bem às fontes de sinais modeladas as aplicações para codificação são imediatas. Isto porque o conhecimento dos sinais a partir de seu passado é grande (graças à predição). Assim, a distribuição de probabilidades da variável aleatória ($\mathcal{X} \mid$ vizinhança de \mathcal{X}) fica bastante concentrada em volta de 0. Portanto sua entropia é bem menor que a da variável \mathcal{X} . Assim, a codificação dos parâmetros das fontes e dos sinais de erro pode ser bem mais econômica que a codificação da fonte original. Pode-se, também, entender esta redução de entropia de outra forma: a representação de um sinal através da diferença entre seus valores e as predições (sinal de erro) elimina a informação mútua entre as amostras do sinal e seu passado.

Outra abordagem bastante utilizada é a que procura caracterizar o comportamento das fontes de imagens através de modelos sobre alguma(s) medida(s) da fonte. A medida mais comumente modelada é a covariância (a ser definida nas próximas seções).

Esta modelagem da covariância de fontes é útil na obtenção de transformadas que decorrelacionem as diversas dimensões do sinal de imagem. Isto porque a transformação linear Φ que diagonaliza a covariância é tal que decorrelaciona as dimensões do sinal de imagens (este resultado será comprovado na continuidade do capítulo).

Tais transformadas decorrelacionadoras do sinal de imagem podem ser de difícil implementação prática devido aos custos computacionais envolvidos. No entanto, através da modelagem da covariância, além da obtenção das transformadas decorrelacionadoras, estabelece-se

uma base probabilística para a aproximação destas transformadas por outras de implementação mais vantajosa (transformadas rápidas, por exemplo).

Maiores informações sobre a transformada decorrelacionadora, chamada de *Karhunen-Loève*, estão apresentadas nas próximas seções.

Conforme mencionado anteriormente, o comportamento das fontes de imagens não é estacionário. As fontes são bem caracterizadas como quase-estacionárias, isto é, possuem comportamento local aproximadamente estacionário. Este fato é levado em conta na modelagem local de imagens através da consideração de múltiplas fontes para as diversas regiões das imagens. Considera-se um conjunto de Q possíveis fontes $\{F_0, \dots, F_{Q-1}\}$ para uma dada região. A probabilidade da fonte F_i modelar uma dada região é dada por $P_{\text{fonte}}(i)$. Exemplos desta abordagem podem ser encontrados em [189], [190] e [114].

Esta modelagem é útil primeiramente porque aproxima melhor o comportamento das fontes de imagens. Em particular permite caracterizar o comportamento de regiões com ou sem arestas através do mesmo modelo. Sob o ponto de vista da implementação este modelo permite a utilização de um conjunto de métodos de codificação de regiões de imagens. Inicialmente uma imagem é segmentada em regiões caracterizadas pelas diferentes fontes F_i 's. Em seguida as regiões são codificadas de acordo com os respectivos métodos de codificação.

As próximas seções destinam-se à apresentação detalhada dos modelos preditivos e de caracterizações de medidas de imagens.

4.2 Modelos Preditivos

A modelagem por preditores consiste inicialmente na formulação de um modelo paramétrico para o sinal. O(s) parâmetro(s) do modelo são estimados tão logo existam amostras do sinal.

A base da utilização de modelos preditivos em codificação de sinais consiste na transformação de representação que leva o sinal original em um sinal residual. A codificação por predição considera os sinais decompostos em duas partes: uma parte predita a partir dos valores conhecidos e dos parâmetros estimados do modelo; e outra parte contendo a diferença entre a predição e o sinal original, conhecida por resíduo, ou sinal residual. A nova representação é feita através dos parâmetros de modelagem e do resíduo.

Caso a estimativa realmente se aproxime bastante do sinal original, o sinal de resíduo apresenta grande concentração em volta de 0. Desta forma remove-se a redundância do sinal original (informação mútua entre pontos vizinhos das imagens) utilizando-se um sinal com entropia mais baixa.

Os preditores utilizados na prática são lineares. O artigo [107] traz uma boa revisão sobre preditores lineares.

Uma das primeiras áreas de utilização de preditores lineares foi a codificação de voz ([111]). Na verdade, o emprego de preditores lineares estabeleceu-se de forma sólida para sinais unidimensionais (voz por exemplo) antes de ser aplicado a imagens. Por isto, as primeiras aplicações de preditores em imagens foram unidimensionais, conforme apresenta a próxima seção.

4.2.1 Modelos Unidimensionais

A caracterização das imagens através de modelos unidimensionais baseia-se na representação unidimensional produzida pela varredura das imagens. Através da varredura as imagens em preto-e-branco (sinais bidimensionais) são representadas por sinais unidimensionais.

A varredura estabelece uma ordem na matriz de pontos aonde está definida a imagem. Em seguida as intensidades luminosas (valores da matriz) são dispostas numa seqüência de acordo com a ordem estabelecida. Quando as seqüências de imagens sucessivas são encadeadas obtém-se um sinal unidimensional para representar imagens.

Existem duas varreduras muito comuns para televisores e computadores: varredura entrelaçada e varredura progressiva.

A varredura entrelaçada é a utilizada em televisores. Para definir a ordenação dos pontos inicialmente define-se uma ordenação para as linhas. Antes da ordenação, porém as linhas recebem uma numeração: as linhas são numeradas de forma crescente da parte superior das imagens em direção à parte inferior. A linha superior da imagem recebe o número 1, a linha logo abaixo recebe o número 2, e assim por diante até a linha inferior da imagem.

Na ordenação as linhas ímpares antecedem as pares. Dentro do mesmo conjunto de linhas (pares ou ímpares) as linhas são ordenadas de forma crescente. Assim, a ordenação de linhas fica:

$$(1, 3, 5, \dots, 2n \Leftrightarrow 1, 2, 4, 6, \dots, 2n)$$

se o número de linhas for par ($2n$), ou

$$(1, 3, 5, \dots, 2n \Leftrightarrow 1, 2n + 1, 2, 4, 6, \dots, 2n)$$

caso o número de linhas seja ímpar ($2n + 1$). Dentro de uma mesma linha os pontos são ordenados da esquerda para a direita.

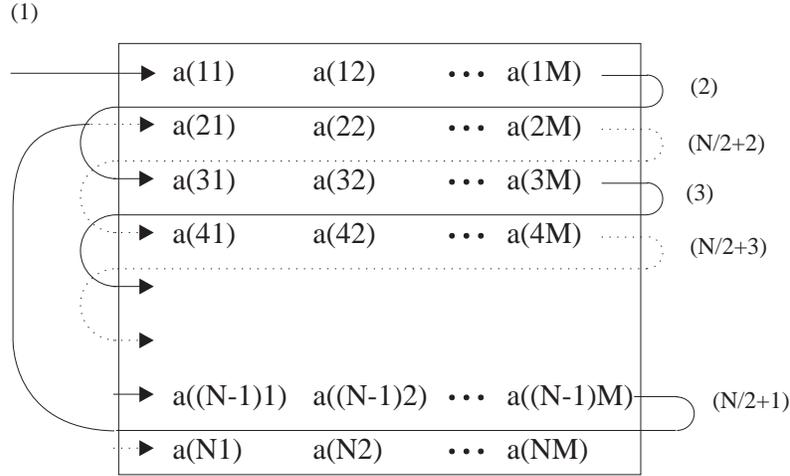
Portanto, os quadros (ou imagens) são divididos em 2 campos. Cada campo é uma sub-amostragem do quadro. Em cada campo os pontos são ordenados da esquerda para a direita (numa mesma linha) e de cima para baixo (dentre as linhas do campo). A figura 4.2 ilustra esta varredura.

A varredura progressiva, por sua vez, é muito utilizada para computadores. Nela os pontos também são ordenados da esquerda para a direita e de cima para baixo. A diferença em relação à varredura entrelaçada é que as linhas são ordenadas de forma crescente. Não há uma divisão de um quadro em dois campos sub-amostrados. A figura 4.3 ilustra esta varredura.

Apesar da diferença, as duas varreduras têm em comum o fato de pontos adjacentes numa mesma linha serem também adjacentes na representação unidimensional do sinal (com exceção dos pontos extremos das linhas). Esta é a característica importante para a predição unidimensional de sinais de imagem.

Neste caso as diversas linhas da imagem são tratadas como sinais unidimensionais independentes. Assim a predição numa linha não leva em conta informação de outras linhas, como ilustra a figura 4.4.

Esta característica faz com que a predição unidimensional tenha uma potencialidade reduzida. Isto porque ela não modela a correlação existente entre pontos vizinhos mas em linhas diferentes.



Seqüência de pontos da varredura:
 $a(11), a(12), \dots, a(1M),$
 $a(31), a(32), \dots, a(3M),$
 $a((N-1)1), a((N-1)2), \dots, a((N-1)M),$
 $a(21), a(22), \dots, a(2M),$
 $a(41), a(42), \dots, a(4M),$
 $a(N1), a(N2), \dots, a(NM)$

Figura 4.2: Varredura entrelaçada transformando imagens (matrizes bidimensionais de pontos) em sinais unidimensionais.

No entanto, este modelo tem sua importância prática motivada por alguns fatores:

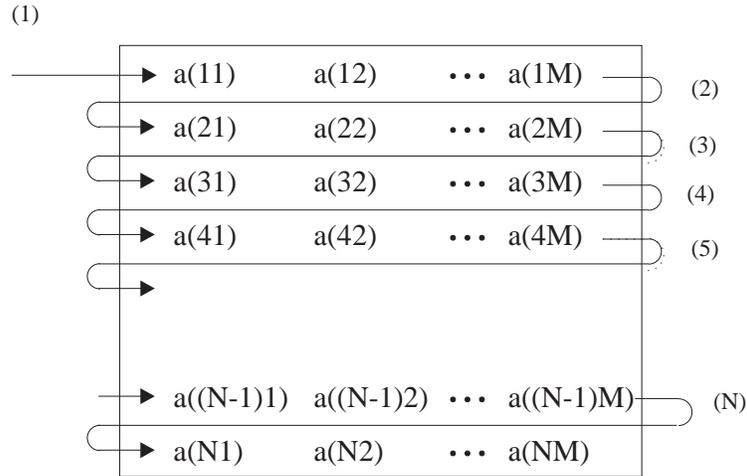
- A representação de imagens como sinais unidimensionais através de varredura é muito comum devido à televisão. O fato da modelagem utilizar a informação tal como ela é representada reduz os custos computacionais do processamento de imagens.
- Os modelos preditores unidimensionais são bastante utilizados em outras áreas (como o processamento de voz). Desta forma existe uma cultura sedimentada no uso destes modelos.

Os modelos lineares mais usados para fontes de imagens são apresentados nas próximas sub-seções.

Modelos Autoregressivos (AR)

Um processo estocástico discreto (ou sinal) $u(n)$ é dito autoregressivo de ordem p quando ele pode ser gerado como a saída de um sistema linear

$$u(n) = \sum_{k=1}^p a(k)u(n \Leftarrow k) + \epsilon(n) \quad (4.1)$$



Seqüência de pontos da varredura:
 $a(11), a(12), \dots, a(1M),$
 $a(21), a(22), \dots, a(2M),$
 $a(31), a(32), \dots, a(3M),$
 $a(41), a(42), \dots, a(4M),$
 $a((N-1)1), a((N-1)2), \dots, a((N-1)M),$
 $a(N1), a(N2), \dots, a(NM)$

Figura 4.3: Varredura progressiva transformando imagens (matrizes bidimensionais de pontos) em sinais unidimensionais.

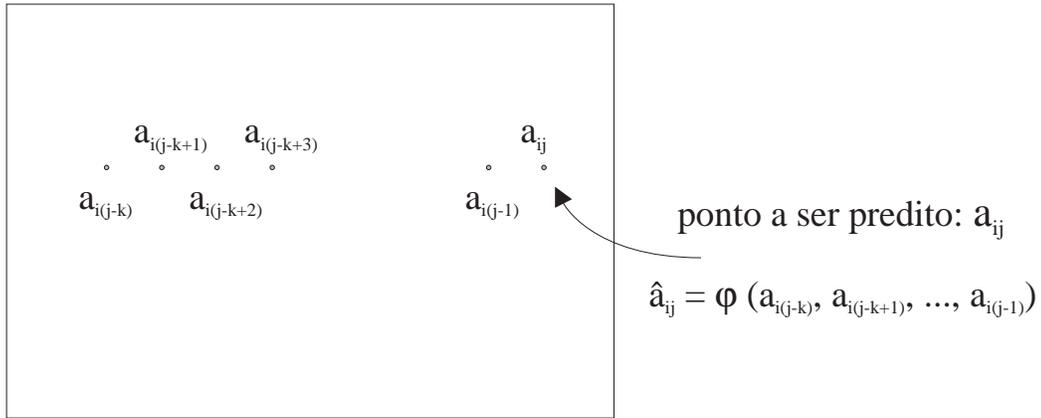


Figura 4.4: Pontos utilizados para a predição unidimensional

sendo que o sinal de excitação $\epsilon(n)$ é estacionário, tem média 0 e é independente do estado anterior do sinal $u(n)$. Em linguagem matemática as restrições ao sinal $\epsilon(n)$ são:

$$E[\epsilon(n)] = 0,$$

$$\begin{aligned} E[\epsilon(n)\epsilon(m)] &= \beta^2 \delta(n \Leftrightarrow m), \\ E[\epsilon(n)u(m)] &= 0, m < n \end{aligned} \quad (4.2)$$

O preditor do sinal é dado por

$$\overline{u(n)} = \sum_{k=1}^p a(k)u(n \Leftrightarrow k) \quad (4.3)$$

Seguindo esta notação, o sinal é dado por

$$u(n) = \overline{u(n)} + \epsilon(n) \quad (4.4)$$

O preditor $\overline{u(n)}$ tal que o sinal residual $\epsilon(n)$ satisfaz as condições 4.2 é ótimo. Isto porque ele minimiza o valor $E\{[u(n) \Leftrightarrow \overline{u(n)}]^2\}$, isto é, a potência do sinal residual, dentre todos os preditores lineares baseados nos últimos p valores do sinal $u(n)$.

A transformação de representação que leva o sinal original ao resíduo decorre diretamente de 4.4

$$\epsilon(n) = u(n) \Leftrightarrow \overline{u(n)} = u(n) \Leftrightarrow \sum_{k=1}^p a(k)u(n \Leftrightarrow k) \quad (4.5)$$

Esta transformação equivale a uma filtragem do sinal $u(n)$ cujo filtro tem função de transferência:

$$A(z) = 1 \Leftrightarrow \sum_{k=1}^p a(k)z^{-k} \quad (4.6)$$

O filtro com estas características é chamado de filtro branqueador do sinal $u(n)$, pois gera um ruído branco $\epsilon(n)$ a partir de $u(n)$.

Caso o sinal $u(n)$ não tenha média 0, como de fato ocorre com imagens, o modelo tem de ser um pouco alterado. É necessário criar um processo estocástico discreto auxiliar $x(n)$, tal como mostram as equações abaixo:

$$x(n) = \sum_{k=1}^p a(k)u(n \Leftrightarrow k) + \epsilon(n) \quad (4.7)$$

$$u(n) = x(n) + \mu \quad (4.8)$$

onde μ é a média do sinal $u(n)$.

A identificação dos parâmetros $\{a_1, \dots, a_p\}$ começa com a escolha de p , isto é a ordem do preditor. O artigo [107] discorre sobre a escolha de tal parâmetro para sinais genéricos. Uma vez que p esteja escolhido existem dois métodos clássicos para estimação dos parâmetros: método da Autocorrelação e método da Covariância. Estes métodos estão descritos em [107] e [110].

A aproximação das linhas de imagens por sinais autoregressivos é bastante boa para regiões sem arestas. De fato, o alto grau de correlação entre pontos vizinhos pode muito bem ser modelado por uma combinação linear da intensidade luminosa dos pontos vizinhos mais um resíduo com comportamento de ruído branco (exceto pela média μ). Para este fim poucos

pontos vizinhos precisam ser considerados (p é pequeno). A literatura relata que apenas um ou dois pontos na mesma linha são suficientes para fazer a predição linear dos valores dos pontos de imagens ([114], [110]).

Por outro lado, a aproximação fica pobre nas vizinhanças de arestas. Isto porque arestas implicam em variações bruscas de intensidade luminosa completamente independentes das intensidades dos pontos anteriores. Nas regiões com arestas o sinal residual assume valores altos. Este fenômeno pode ser observado na literatura. Os exemplos de sinais residuais apresentados em [80], [100], [114] e [110] têm valores mais elevados nas regiões nas quais os respectivos sinais de imagem apresentam arestas. Ainda assim esta modelagem é útil para caracterizar imagens.

Outros Modelos

Existem outros modelos de predição linear que têm aplicação bem mais restrita na modelagem de imagens. Ei-los:

- Modelos “Moving Average” (MA): Um sinal $u(n)$ é dito um processo “moving average” de ordem q caso possa ser escrito como a soma ponderada de processos descorrelacionados:

$$u(n) = \sum_{k=0}^q b(k)\epsilon(n \Leftrightarrow k) \quad (4.9)$$

- Modelos Auto-regressivos e “Moving Average” (ARMA): Um sinal $u(n)$ é dito um processo auto-regressivo e “moving average” caso possa ser escrito como a uma estimativa linear de seu passado mais uma soma ponderada de processos descorrelacionados:

$$u(n) = \sum_{k=1}^p a(k)u(n \Leftrightarrow k) + \sum_{k=0}^q b(k)\epsilon(n \Leftrightarrow k) \quad (4.10)$$

Este modelo generaliza os modelos AR e MA. Caso todos os parâmetros $b(k)$ sejam nulos o modelo recai no AR. Por outro lado, se todos os parâmetros $a(k)$ se anularem o modelo recai no MA.

- Modelos de Variáveis de Estado : Esta modelagem utiliza um processo auxiliar $X(n)$ para caracterizar o processo principal $Y(n)$. O uso desse processo auxiliar, chamado de estado do processo $Y(n)$ é dado por:

$$\begin{aligned} X(n+1) &= A_n Y(n) + B_n \epsilon(n) \\ Y(n) &= C_n X(n) + \eta(n) \end{aligned} \quad (4.11)$$

Cada elemento do processo $X(n)$ é um vetor aleatório $m \times 1$. Os elementos de $Y(n)$, $\epsilon(n)$ e $\eta(n)$, por outro lado são matrizes $p \times 1$. As matrizes de parâmetros A_n , B_n e C_n têm dimensões respectivamente $m \times m$, $m \times p$ e $p \times m$. O vetor $Y(n)$ pode ter diversos significados, como por exemplo a intensidade luminosa num conjunto de pontos. Também neste caso a modelagem exige condições para os processos de excitação $\epsilon(n)$ e

$\eta(n)$: $\epsilon(n)$ tem de ser um vetor composto por variáveis aleatórias independentes e com média 0, enquanto que $\eta(n)$ tem que ser um ruído branco. Em linguagem matemática as condições ficam:

$$E[\eta(n)] = 0, \quad (4.12)$$

$$E[\eta(n)\eta(n')^T] = Q_n\delta(n \Leftrightarrow n'), \quad (4.13)$$

$$E[\epsilon(n)] = 0$$

$$E[\eta(n)\epsilon(n')^T] = 0, \quad (4.14)$$

$$E[\epsilon(n)\epsilon(n')^T] = P_n\delta(n \Leftrightarrow n'), \quad (4.15)$$

$$\forall n, n' \quad (4.16)$$

Este modelo é uma generalização do modelo ARMA.

4.2.2 Modelos Bidimensionais

Os modelos preditores bidimensionais usam o mesmo princípio que os unidimensionais. Novamente neste caso os modelos auto-regressivos são os que apresentam maiores e melhores aplicações.

No entanto, o conjunto de pontos usados para fazer a predição não está restrito apenas à linha do ponto a ser predito. Devido à esta flexibilidade não há um único conjunto de pontos naturalmente usado para predição.

No caso de sinais unidimensionais a evolução dos sinais está normalmente associada ao tempo. O sinal x no instante $n\Delta t$, onde Δt é o período de amostragem, fica representado por $x(n)$. Portanto, a seqüência de pontos do sinal está associada à sua ordem cronológica de ocorrência. Desta forma, é natural considerar que o valor do sinal x no ponto n , isto é, no instante $n\Delta t$, dependa dos valores nos instantes anteriores $\dots, x(n \Leftrightarrow p), x(n \Leftrightarrow (p \Leftrightarrow 1)), \dots, x(n \Leftrightarrow 1)$. A dependência de um sinal com seu passado apenas, e não com seu futuro, é chamada de princípio de causalidade.

No caso dos modelos bidimensionais de imagens, a correlação existe entre os pontos e seus vizinhos. Esta correlação não está associada à ordem de ocorrência de eventos, e sim às características das imagens. Todos os pontos de uma imagem são essencialmente amostrados ao mesmo tempo. Quando há correlação entre pontos vizinhos isto não acontece porque a ocorrência de um valor em um ponto veio a influenciar outro ponto na seqüência temporal.

Portanto não há causalidade intrínseca nos modelos bidimensionais de imagens. Em princípio toda a vizinhança de um ponto, e não apenas uma parcela que acontece “anteriormente” segundo alguma ordenação, influencia o ponto.

No entanto, existem aplicações em que a predição dos pontos é feita de forma recursiva. Isto significa que os pontos são ordenados, normalmente da esquerda para a direita e de cima para baixo, conforme as varreduras entrelaçadas e progressivas. Devido à recursividade, a predição de um ponto $u(n, m)$ só pode utilizar os pontos já conhecidos (acima e na mesma linha à esquerda). Os modelos utilizados nestas aplicações são chamados de *modelos causais*. Neles o conjunto de pontos utilizados para predição está limitado por $W_1 = \{u(p, q) \in \text{imagem} \mid q < m \text{ ou } (q = m \text{ e } p < n)\}$, conforme ilustra a figura 4.5.

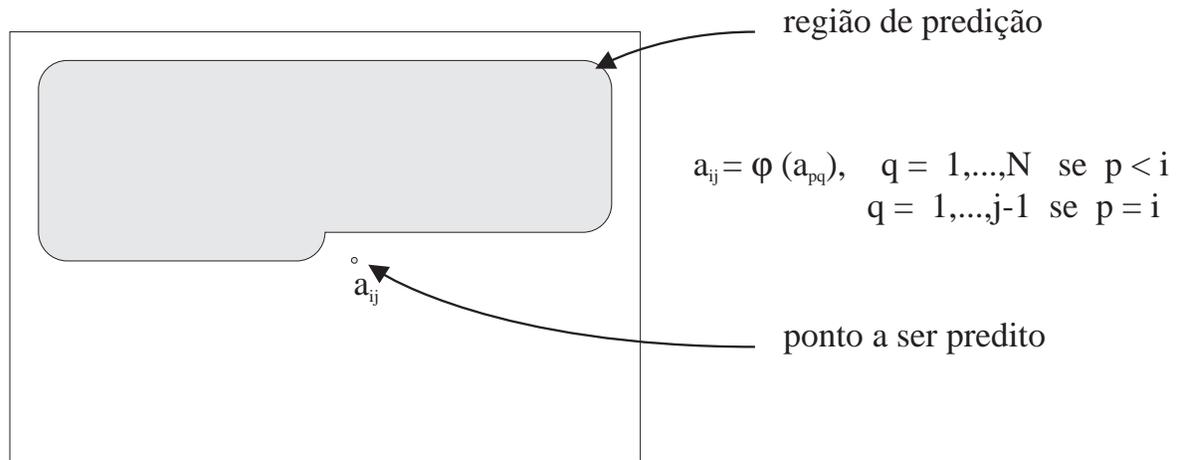


Figura 4.5: Conjunto restrição dos pontos de predição de modelos causais.

Um caso particular dos modelos causais, chamado de *modelo fortemente causal*, utiliza um subconjunto de W_1 como conjunto de predição. Este subconjunto $W'_1 \subset W_1$ contém apenas pontos acima e à esquerda do ponto a ser predito: $W'_1 = \{u(p, q) \in \text{imagem} \mid (q < m \text{ e } p \leq n) \text{ ou } (q = m \text{ e } p < n)\}$. A figura 4.6 ilustra o conjunto W'_1 .

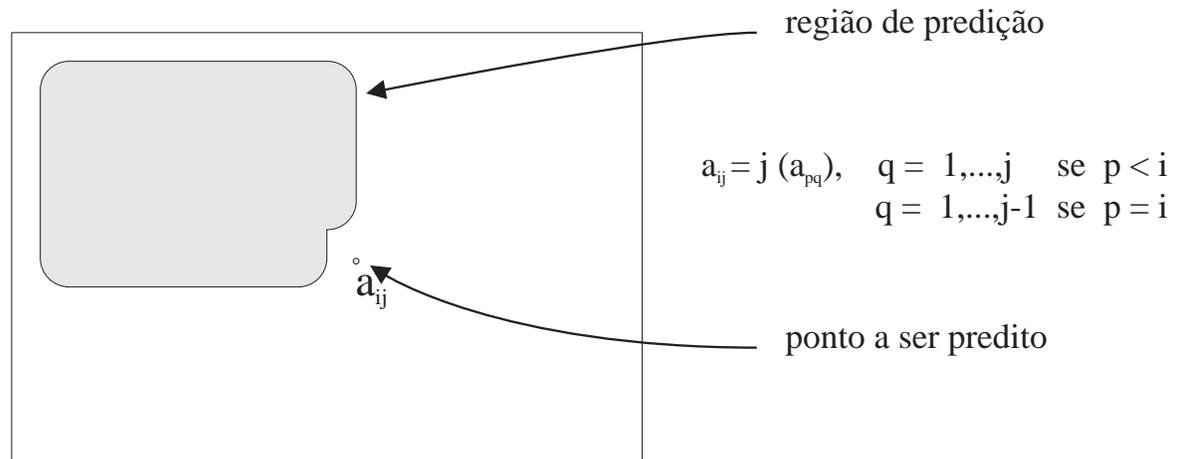


Figura 4.6: Conjunto restrição dos pontos de predição de modelos fortemente causais.

Conforme mencionado anteriormente, os modelos causais e fortemente causais são úteis para as codificações que utilizam predições recursivas a partir da ordenação embutida na varredura de imagens.

Modelos semi-causais são aqueles que usam para prever o valor da imagem num ponto todas as linhas anteriores à linha na qual o ponto está (incluindo a linha do ponto). Neste modelo, além dos pontos utilizados no modelo causal, usa-se também os pontos na mesma

linha e à direita do ponto a ser predito. O conjunto restrição dos pontos de predição é dado por: $W_2 = \{u(p, q) \in \text{imagem} \mid p \leq n\}$, e ilustrado pela figura 4.7.

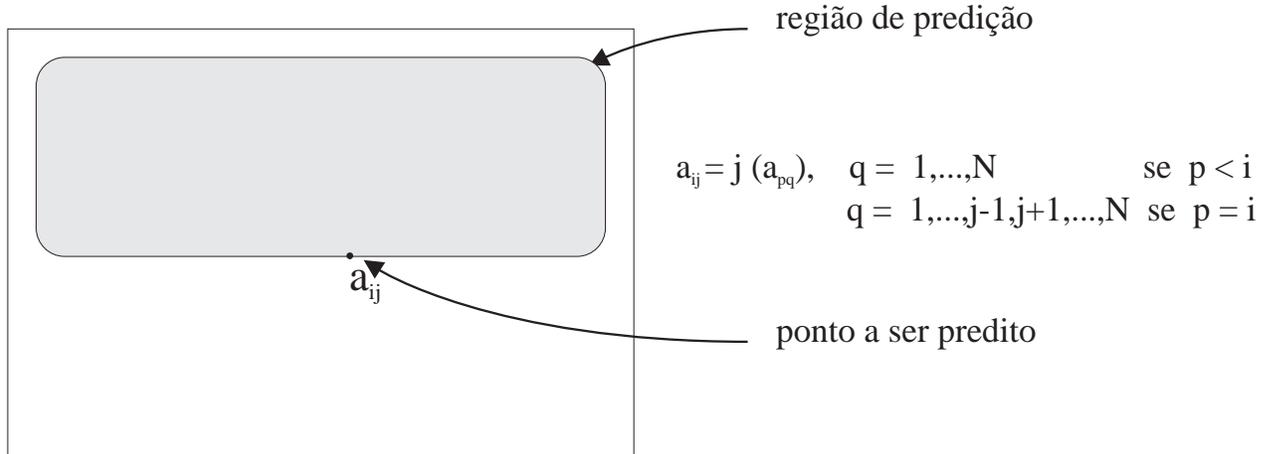


Figura 4.7: Conjunto restrição dos pontos de predição de modelos semi-causais.

Estes modelos semi-causais costumam ser utilizados em codificações híbridas de imagens. Estas caracterizam-se por utilizar métodos de transformadas para decorrelacionar os sinais em uma dimensão (nas linhas, por exemplo) e posteriormente métodos de predição linear para retirar as redundâncias entre as linhas. A figura 4.8 esquematiza um método típico de codificação híbrida.

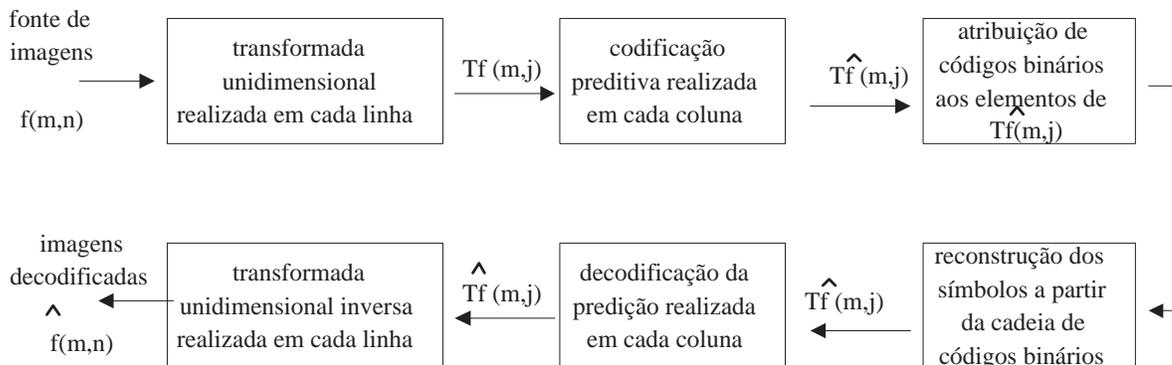


Figura 4.8: Esquema de codificação híbrida de imagens.

Os modelos mais completos são os ditos *não-causais*. Segundo eles, a predição linear de um ponto pode ser feita de acordo com todos os outros pontos da imagem, sem restrição. O conjunto de predição é dado por: $W_3 = \{u(p, q) \in \text{imagem} \mid (p, q) \neq (n, m)\}$ e ilustrado pela figura 4.9.

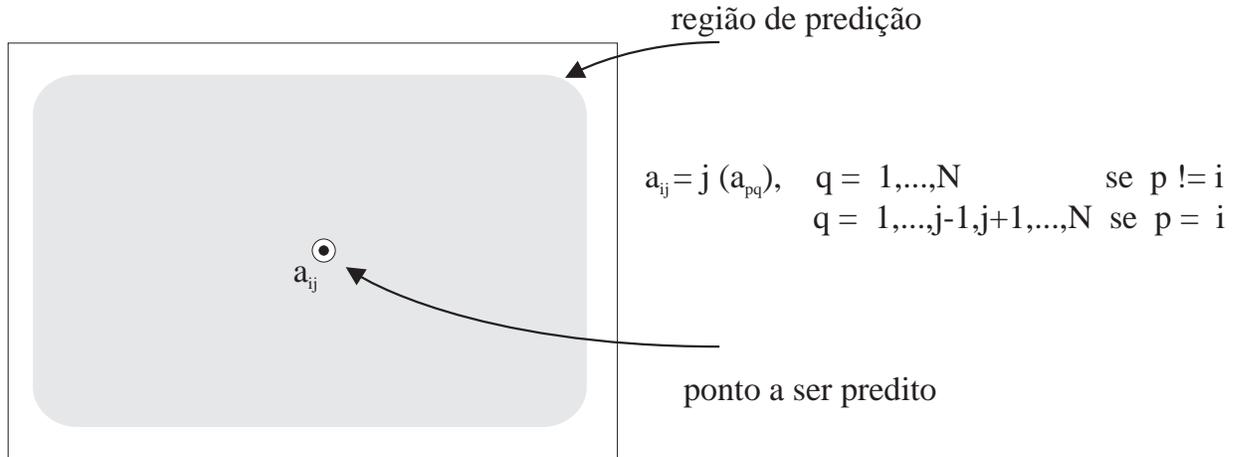


Figura 4.9: Conjunto restrição dos pontos de predição de modelos não-causais.

Estes modelos não-causais costumam ser usados nos métodos de codificação por transformadas. Nestes métodos as imagens são processadas em blocos e não de acordo com a seqüência induzida pela varredura. Em cada bloco os pontos estão cercados de vizinhos (exceto os pontos pertencentes às bordas dos blocos).

De acordo com a aplicação utilizada escolhe-se um modelo bidimensional. Em seguida, escolhe-se um conjunto W_x de pontos para servir como preditores dos pontos na imagem. Este conjunto preditor é um subconjunto do conjunto de restrição (W_1 , W'_1 , W_2 ou W_3). O preditor linear do ponto $u(n, m)$ é dado pela equação 4.17.

$$\overline{u(n)} = \sum_{(p,q) \in W_x} a(p, q) u(n \Leftrightarrow p, m \Leftrightarrow q) \quad (4.17)$$

A estimação dos parâmetros $a(p, q)$, $(p, q) \in W_x$ está descrita em [80] e [110].

Da mesma forma que no caso de modelos unidimensionais, a literatura relata que apenas poucos pontos vizinhos são suficientes para fazer a predição linear dos valores pontos de imagens ([114] e [110])

4.2.3 Modelos de Predição de Movimentos

Nas seções anteriores foram apresentados modelos para imagens estáticas. Mais especificamente, os modelos uni e bidimensionais procuram caracterizar a correlação existente entre pontos vizinhos de uma mesma imagem.

No entanto, além da correlação espacial, as fontes de imagens apresentam bastante correlação temporal. Modelos preditores tri-dimensionais podem ser utilizados para caracterizar a correlação temporal. A base para esta modelagem é a interpretação do sinal de imagem como um sinal tridimensional (duas coordenadas espaciais (n, m) e uma temporal t).

A predição de um ponto $u[(n, m), t]$ é feita através de uma combinação linear de pontos vizinhos. Define-se uma vizinhança para os pontos da imagem. A vizinhança tem papel

análogo ao do conjunto W_x usado nos modelos bidimensionais.

Esta vizinhança implica na ordem do preditor e nas suas características de causalidade. Na dimensão temporal existe causalidade entre imagens sucessivas.

Na prática usa-se com bastante freqüência a técnica da *predição de movimentos* para reduzir a redundância temporal de uma seqüência de imagens. Esta técnica pertence a um contexto mais geral de *estimação de movimentos* que tem aplicação não apenas na codificação de imagens como também nas áreas de análise de imagens e restauração de imagens ([45]).

No caso de codificação de imagens a predição de movimentos é implementada através de técnicas de casamento de blocos “*block matching techniques*”. A predição não é feita ponto-a-ponto. Inicialmente as imagens são divididas em blocos (normalmente retangulares $N \times M$ ou quadrados N^2). Estes blocos podem ser determinados de forma adaptativa, como proposto em [45] e [32]. A predição é realizada a nível de blocos. Para cada bloco existe uma região de procura. Pesquisa-se qual a melhor aproximação de cada bloco dentre os blocos $N \times M$ dentro da região de procura das imagens vizinhas (normalmente da anterior). Esta melhor aproximação é utilizada como predição do bloco. Esta predição a nível de blocos procura aproveitar o fato que objetos pertencentes à uma cena muitas vezes continuam pertencendo às cenas subsequêntes com pequenas alterações, como ilustra a figura 4.10.

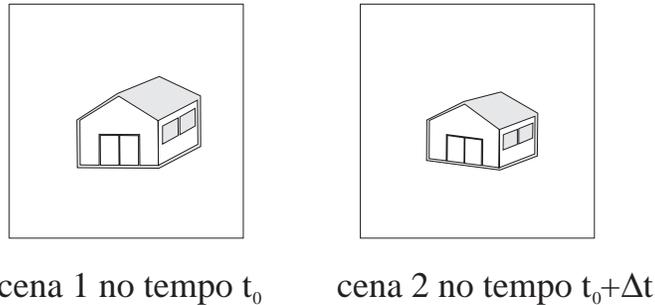


Figura 4.10: Objeto M sendo focalizado em diferentes regiões de pixels em duas imagens adjacentes no tempo

A região de procura normalmente é um retângulo de dimensões $(N + \delta N) \times (M + \delta M)$ centrado no retângulo original, como ilustra a figura 4.11.

Nos retângulos próximos às bordas das imagens o retângulo de procura ultrapassa os limites das imagens. Nestes casos restringe-se as regiões de procura para que fiquem dentro da imagem.

Para que a predição de movimento seja implementada é preciso definir uma medida de distorção entre dois blocos de imagens. Segundo esta medida pode-se determinar qual a melhor aproximação de um bloco dentro de seu conjunto de predição (união das regiões de predição das imagens vizinhas). As medidas de distorção mais utilizadas são: média do erro quadrático (MSE) e média do erro absoluto (MAE). Suas definições são dadas a seguir. Nestas definições vale ressaltar que o bloco a ser predito é $B_{nmt} = \{u[(p, q), t] \mid (p, q) \in [n, n + (N \Leftrightarrow 1)] \times [m, m + (M \Leftrightarrow 1)]\}$.

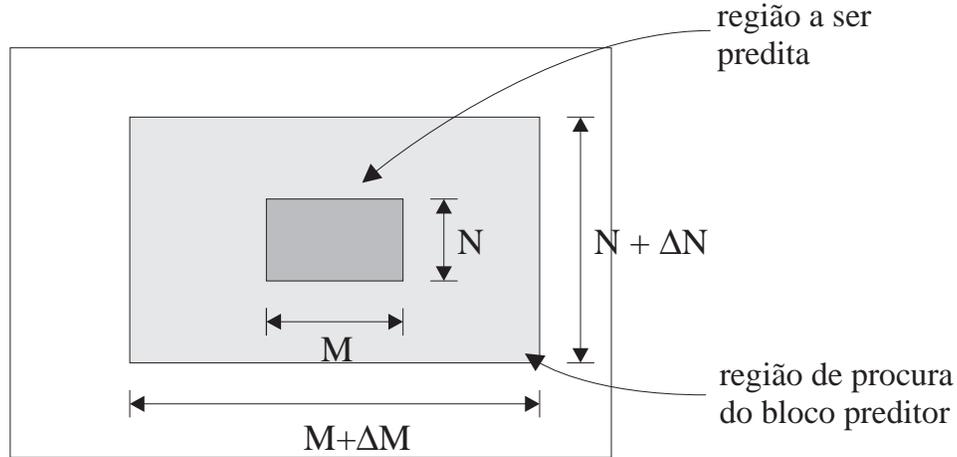


Figura 4.11: Região de procura de um retângulo $N \times M$

- Caso MSE

$$\begin{aligned}
 M(B_{nmt}, B_{n_1 m_1 t_1}) &= \frac{1}{N \times M} \sum_{i=1}^N \sum_{j=1}^M \{u[(n+i, m+j), t] \\
 &\Leftrightarrow u[(n_1+i, m_1+j), t_1]\}^2
 \end{aligned} \tag{4.18}$$

- Caso MAE

$$\begin{aligned}
 M(B_{nmt}, B_{n_1 m_1 t_1}) &= \frac{1}{N \times M} \sum_{i=1}^N \sum_{j=1}^M |u[(n+i, m+j), t] \\
 &\Leftrightarrow u[(n_1+i, m_1+j), t_1]|
 \end{aligned} \tag{4.19}$$

A determinação da melhor aproximação é um procedimento bastante caro computacionalmente. Existem alguns algoritmos de determinação sub-ótima de predições que sacrificam a aproximação em prol de viabilidade de implementação ([65], [82] e [102]).

4.3 Modelos de Covariância

Dentre os modelos que caracterizam o comportamento de medidas das fontes de imagens destacam-se os modelos de Covariância.

A covariância entre duas variáveis aleatórias X e Y definidas no mesmo espaço de probabilidade é dada por:

$$\text{Cov}_{XY} = r_{XY} = E\{[X \Leftrightarrow \mu_X][Y \Leftrightarrow \mu_Y^*]\} \tag{4.20}$$

onde μ_X e μ_Y são respectivamente os valores médios de X e Y .

Quando um sinal $u(n)$ (seqüência de variáveis aleatória) é analisado define-se uma covariância para cada par de variáveis aleatórias $u(n)$ e $u(n')$:

$$\text{Cov} [u(n), u(n')] = E\{[u(n) \Leftrightarrow \mu(n)][u^*(n') \Leftrightarrow \mu^*(n')]\} \quad (4.21)$$

Neste caso a covariância pode ser entendida como uma função $\text{Cov} : Z \times Z \rightarrow R$.

Caso o processo $u(n)$ seja estacionário o comportamento da fonte do sinal não varia com o deslocamento (n ou n'). Sendo assim,

$$\begin{aligned} \text{Cov} [u(n), u(n')] &= E\{[u(n) \Leftrightarrow \mu(n)] \\ &\quad [u^*(n + \delta n) \Leftrightarrow \mu^*(n + \delta n)]\} \\ &= E\{[u(n') \Leftrightarrow \mu(n')] \\ &\quad [u^*(n' + \delta n) \Leftrightarrow \mu^*(n' + \delta n)]\} \end{aligned}$$

Portanto, a covariância só depende da diferença entre os seus argumentos. Nestas circunstâncias pode-se definir uma função de covariância a um parâmetro $r_u(\delta n)$ para um sinal unidimensional. O parâmetro é a diferença entre os seus argumentos da covariância, tal como definida em 4.21.

O modelo de covariância pode ser utilizado para caracterizar imagens estáticas ou dinâmicas (lembrar que nestes casos os sinais são reais, portanto a conjugação complexa na fórmula de cálculo da covariância não exerce papel importante).

4.3.1 Covariância para Imagens Estáticas

No caso de imagens estáticas, considera-se cada ponto $u(n, m)$ das imagens como uma variável aleatória. A função de covariância tem como argumentos os pontos (n, m) e (n', m') que definem a posição das variáveis aleatórias consideradas dentro da imagem:

$$\text{Cov} [u(n, m), u(n', m')] = E\{[u(n, m) \Leftrightarrow \mu(n, m)] \quad (4.22)$$

$$[u^*(n', m') \Leftrightarrow \mu^*(n', m')]\} \quad (4.23)$$

Se a fonte de imagens for estacionária, a função de covariância escreve-se $r_u(\delta n, \delta m)$, onde δn e δm são respectivamente as diferenças entre os seus argumentos n e n' e m e m' da covariância, tal como definida em 4.23.

No caso de imagens dinâmicas, as variáveis aleatórias distribuem-se no tempo e nas dimensões das matrizes de imagens: $u[(n, m), t]$. A função de covariância tem como argumentos os pontos $[(n, m), t]$ e $[(n', m'), t']$:

$$\text{Cov} \{u[(n, m), t], u[(n', m'), t']\} = E\{[u[(n, m), t] \Leftrightarrow \mu[(n, m), t]] \quad (4.24)$$

$$[u^*[(n', m'), t'] \Leftrightarrow \mu^*[(n', m'), t']]\} \quad (4.25)$$

Novamente neste caso o modelo fica bastante simplificado se a fonte for estacionária. Para cada dimensão em que a fonte for estacionária (uma das duas dimensões espaciais ou o tempo), a covariância pode ser representada como uma função da diferença entre os respectivos argumentos da definição 4.25 naquela dimensão. Portanto, se a fonte for estacionária

nas três dimensões, isto é, caso seu comportamento for invariante com o tempo e com os deslocamentos dentro das coordenadas das imagens, então a função de covariância tem três parâmetros.

Existem dois modelos bastante usados para a covariância de imagens estáticas. Ambos consideram que as fontes de imagens são estacionárias. Ei-los:

- Modelo Separável:

$$r_u(n, m) = \sigma^2 \rho_1^{|n|} \rho_2^{|m|} \quad (4.26)$$

com $|\rho_1| < 1$ e $|\rho_2| < 1$.

- Modelo Não-Separável:

$$r_u(n, m) = \sigma^2 e^{-\sqrt{\alpha_1 n^2 + \alpha_2 m^2}} \quad (4.27)$$

Caso $\alpha_1 = \alpha_2 = \alpha$, a covariância pode ser escrita em função da distância $d = \sqrt{n^2 + m^2}$ no plano das imagens:

$$r_u(n, m) = \sigma^2 \rho^d \quad (4.28)$$

onde $\rho = e^{-|\alpha|}$. Uma função de covariância com esta característica é dita *isotrópica* ou *circularmente simétrica*.

O modelo não-separável é menos restritivo que o separável. Suas aproximações das funções de covariância são em geral melhores do que as obtidas por modelos separáveis. Os modelos separáveis, no entanto, aproximam as funções de covariância razoavelmente e são de manuseio mais fácil.

A estimação dos parâmetros é similar para ambos os modelos. Inicialmente estima-se a variância do sinal, isto é, $r_u(0, 0) = \sigma^2$. Em seguida estima-se as correlações entre pontos vizinhos nas duas dimensões espaciais da imagem: $\rho_1 = r_u(1, 0)/\sigma^2$, $\rho_2 = r_u(0, 1)/\sigma^2$. Eventualmente pode-se calcular as correlações entre pontos não adjacentes: $r_u(a, b)/\sigma^2$, com $a > 1$ e/ou $b > 1$.

A estimação destes parâmetros é feita usando dados de alguma imagem, isto é, dados de alguma amostra da fonte de imagens. Esta estimação é apenas aproximada. Primeiramente porque se baseiam em medidas realizadas com um conjunto pequeno de dados. Além disto, as medidas obtidas não são esperanças de variáveis aleatórias, e sim médias de alguns valores amostrados destas variáveis.

Uma vez estimados σ^2 , ρ_1 e ρ_2 (e eventualmente algumas outras correlações entre pontos próximos) o modelo está “completo”. Se o modelo for separável todos os parâmetros necessários estão determinados. Caso o modelo seja não separável, os parâmetros α_1 e α_2 relacionam-se diretamente com ρ_1 e ρ_2 através das fórmulas:

$$\alpha_1 = \Leftrightarrow \ln \rho_1 \quad (4.29)$$

$$\alpha_2 = \Leftrightarrow \ln \rho_2 \quad (4.30)$$

Na verdade esta estimação é bastante pobre. Além das limitações do número reduzido de dados e da necessidade da hipótese de ergodicidade para formalizar a estimação, os valores da função de correlação usados são normalmente apenas $r_u(1, 0)$ e $r_u(0, 1)$. Os valores de

r_u com argumentos mais elevados ($(a, b) \mid \sqrt{a^2 + b^2} \gg 1$) normalmente não são levados em consideração.

Apesar das simplificações, estes modelos apresentam boa aproximação com as fontes de imagens encontradas na prática. A literatura ([80]) reporta valores variando entre 0.8 e 0.99, com valor médio em 0.95, para as correlações entre pontos vizinhos ρ_1 e ρ_2 .

4.3.2 Covariância para Imagens Dinâmicas

No caso de imagens dinâmicas, além dos modelos separáveis e não-separáveis, existe o modelo em que as dimensões de espaço são separadas das de tempo, mas não separáveis entre si. Os possíveis modelos, já com a dimensão tempo acrescentada estão apresentados em seguida.

- Modelo Separável:

$$r_u[(n, m), t] = \sigma^2 \rho_1^{|n|} \rho_2^{|m|} \rho_3^{|t|} \quad (4.31)$$

com $|\rho_1| < 1$, $|\rho_2| < 1$ e $|\rho_3| < 1$.

- Modelo Não-Separável:

$$r_u(n, m) = \sigma^2 e^{-\sqrt{\alpha_1 n^2 + \alpha_2 m^2 + \alpha_3 t^2}} \quad (4.32)$$

- Modelo Não-Separável no Espaço, mas Separável no Tempo:

$$r_u[(n, m), t] = \sigma^2 e^{-\sqrt{\alpha_1 n^2 + \alpha_2 m^2}} \rho_3^{|t|} \quad (4.33)$$

com $|\rho_3| < 1$.

Tal como no caso de imagens estáticas, na prática apenas as correlações entre pontos próximos são utilizadas para estimar os parâmetros dos modelos. O artigo [30] apresenta gráficos com as correlações entre pontos vizinhos no tempo para seqüências de imagens com deslocamentos simples.

A utilização prática de modelos de covariância para a coordenada tempo de imagens dinâmicas não é muito difundida. Os métodos de codificação por transformadas são a principal motivação para o uso destes modelos (tal como veremos na próxima seção). A utilização de métodos de transformadas na coordenada tempo possui 2 desvantagens:

- Para que se possa realizar transformadas na coordenada tempo é preciso que se armazene vários quadros de uma seqüência de imagens antes de realizar a codificação. Isto acarreta um atraso indesejável entre a geração e a codificação de imagens dinâmicas.
- A melhor aproximação do comportamento de uma seqüência dinâmica de imagens é obtida não através da correlação entre pontos vizinhos no tempo, e sim através da correlação entre pontos que se sucedem na direção de um movimento da cena observada. Isto significa que a modelagem através de predição de movimentos adequa-se melhor ao comportamento dinâmico das imagens do que a modelagem de covariância.

4.3.3 Modelos de Covariância e Transformadas

Seja $X = (x(0), x(1), \dots, x(N \Leftrightarrow 1))$ um vetor aleatório. A função de covariância para este vetor pode ser escrita através de uma matriz quadrada R_x de dimensão N^2 , onde $\{R_x\}_{n,m}$ é a covariância existente entre as variáveis aleatórias $x(n)$ e $x(m)$ (coordenadas respectivamente n e m do vetor X). Através da própria definição de covariância (4.20), não é difícil de deduzir que esta matriz R_x é simétrica, com coeficientes reais e, em caso de vetor estacionário, Toeplitz. Portanto, existe uma matriz unitária Φ que reduz R_x a sua forma diagonal, ie, $\exists \Phi$ matriz unitária $N \times N$ tal que

$$\Phi^T R_x \Phi = \Lambda$$

sendo Λ uma matriz diagonal.

O vetor aleatório X pode ser escrito em alguma outra base de R^N . Para isto basta efetuar uma mudança de coordenadas.

Seja a transformação de coordenadas

$$Y = \Phi^T X$$

onde Φ é a matriz que diagonaliza R_x . Esta transformação é chamada de *Transformada de Karhunen-Loève*, ou *Transformada KL*, ou simplesmente *KLT*. Esta transformada é de importância fundamental no processamento de sinais porque a nova representação Y do vetor aleatório é tal que as coordenadas $y(0), \dots, y(N \Leftrightarrow 1)$ são não correlacionadas (matriz de autocorrelação de Y é $\Phi^T R_x \Phi = \Lambda$ que é diagonal).

Além de descorrelacionar as coordenadas de um vetor aleatório, a transformada KL otimiza a compactação de energia em coeficientes de representação do vetor. O lema a seguir formaliza esta afirmação.

Lema 4.1 *Seja $X = (x(0), x(1), \dots, x(N \Leftrightarrow 1))$ um vetor aleatório estacionário com matriz de correlação R_x . Seja $Y_T = (y_T(0), y_T(1), \dots, y_T(N \Leftrightarrow 1))$ o vetor aleatório obtido através da aplicação da transformação linear T sobre X , isto é, $Y_T = T[X]$. Seja $E_{(T,M)}$ o erro médio quadrático devido à truncagem de Y_T nas M componentes de mais alta variância. A transformação KL é dentre todas as transformações lineares a que minimiza $E_{(T,M)}$ para todo $M = 1, 2, \dots, N$.*

Demonstração:

A demonstração será feita em dois passos:

- Passo 1: Prova que para cada valor de M a transformação que minimiza $E_{(T,M)}$ é ortogonal (e sem perda de generalidade ortonormal)
- Passo 2: Prova que para cada valor de M a transformação KL apresenta $E_{(T,M)}$ menor que qualquer outra transformação ortonormal
- Passo 1:

Seja $U = (u_0, u_1, \dots, u_{N-1})$ uma base de R^N . Seja \vec{b} um vetor $\in R^N$. A representação de \vec{b} na base U é dada por:

$$\vec{b} = \sum_{i=0}^{N-1} c_i u_i$$

Sem perda de generalidade pode-se considerar que os vetores u_n estão ordenados de forma que as variâncias de seus respectivos coeficientes c_n formam uma n-upla decrescente:

$$(\text{variância}(c_0), \text{variância}(c_1), \dots, \text{variância}(c_{N-1}))$$

Em outras palavras, $\text{variância}(c_i) > \text{variância}(c_j)$ se $i < j$.

Quando se trunca a representação de \vec{b} na base U considerando apenas as $(N \Leftrightarrow 1)$ primeiras componentes obtém-se o vetor $t_{N-1}(\vec{b})$. Este vetor pertence ao sup-espaço gerado pelos vetores $(u_0, u_1, \dots, u_{N-2})$, denotado por U_0^{N-2} . Portanto o erro de aproximação de \vec{b} por $t_{N-1}(\vec{b})$ é minimizado quando $t_{N-1}(\vec{b})$ for a projeção ortogonal de \vec{b} em U_0^{N-2} , o que só acontece se $u_{N-1} \perp U_0^{N-2}$.

Desta forma, se u_{N-1} não for perpendicular à U_0^{N-2} pode-se obter uma nova base $U' = (u_0, u_1, \dots, u_{N-1} \Leftrightarrow \sum_{i=0}^{N-2} \langle u_i, u_{N-1} \rangle u_i)$ tal que $u'_{N-1} = u_{N-1} \Leftrightarrow \sum_{i=0}^{N-2} \langle u_i, u_{N-1} \rangle u_i \perp \{U_0^{N-2}\}' = U_0^{N-2}$. Nesta nova base a aproximação de \vec{b} por $t'_{N-1}(\vec{b})$ será melhor do que por $t_{N-1}(\vec{b})$.

Como este raciocínio não depende do vetor $\vec{b} \in R^N$ pode-se concluir que a esperança da distância entre \vec{b} e $t_{N-1}(\vec{b})$, $E[\vec{b}, t_{N-1}(\vec{b})]$ só será minimizada por uma base U caso $u_{N-1} \perp U_0^{N-2}$.

Esta argumentação estende-se para as demais dimensões indutivamente. Com efeito, considere-se que os últimos k vetores da base U são todos l.i. e geram um espaço $U_{N-k}^{N-1} \perp U_0^{N-k-1}$. Caso o vetor u_{N-k-1} não seja perpendicular ao sub-espaço $U_0^{N-k-2} \Rightarrow t_{N-k}(\vec{b})$ não será a melhor aproximação de \vec{b} em U_0^{N-k-1} . Portanto haverá margem para a obtenção de uma nova base $U' = (u_0, u_1, \dots, u_{N-k-2}, u_{N-k-1} \Leftrightarrow \sum_{i=0}^{N-k-2} \langle u_i, u_{N-k-1} \rangle u_i, u_{N-k}, \dots, u_{N-1})$ tal que $u'_{N-k-1} = u_{N-k-1} \Leftrightarrow \sum_{i=0}^{N-k-2} \langle u_i, u_{N-k-1} \rangle u_i \perp \{U_0^{N-k-2}\}' = U_0^{N-k-2}$. Logo a esperança da aproximação incorrida na truncagem de coordenadas das dimensões da representação de vetores na base U só será minimizada caso $u_{N-k-1} \perp U_0^{N-k-2}$.

Portanto, por indução, todos os vetores u_n têm de ser perpendiculares aos respectivos sub-espaços U_0^{n-1} , o que equivale a dizer que a base U é ortogonal.

Vale observar que o erro incorrido na aproximação de \vec{b} por $t_{N-k}(\vec{b})$ corresponde à projeção ortogonal de \vec{b} em R^N / U_0^{N-k-1} , que só depende das direções dos vetores u_n , e não de seus módulos. Desta forma, sem perda de generalidade pode-se restringir a pesquisa da base que minimiza a esperança do erro de truncagem ao conjunto de bases ortonormais.

- Passo 2:

Seja U uma transformação ortonormal. A aplicação de U em um vetor aleatório X corresponde à transformação de coordenadas de X para uma base $(u_0, u_1, \dots, u_{N-1})$. As novas coordenadas de X são dadas por:

$$y_u(n) = u_n^* X \quad n = 0, \dots, (N \Leftrightarrow 1) \quad (4.34)$$

O erro médio quadrático incorrido quando apenas M componentes de Y_u são considerados é dado por:

$$\begin{aligned}
 \text{EMQ} &= \frac{1}{N} \sum_{n \geq M} E |y_u(n)|^2 \\
 &= \frac{1}{N} \sum_{n \geq M} E \{u_n^* X u_n^* X^*\} \\
 &= \frac{1}{N} \sum_{n \geq M} E \{u_n^* X X^* u_n\} \\
 &= \frac{1}{N} \sum_{n \geq M} u_n^* E \{X X^*\} u_n \\
 &= \frac{1}{N} \sum_{n \geq M} u_n^* R_x u_n
 \end{aligned} \tag{4.35}$$

Os vetores u_n podem ser escritos na base de vetores da transformação KL (k_0, k_1, \dots, k_{N-1}) (ordenados de forma que os auto-valores de R_x correspondentes aos vetores k_i formem uma n -upla ordenada de forma decrescente ($\lambda_0, \lambda_1, \dots, \lambda_{N-1}$)):

$$u_n = \sum_{i=0}^{N-1} w_{in} k_i \tag{4.36}$$

onde

$$w_{in} = k_i^* u_n \tag{4.37}$$

De forma análoga os vetores da base KL podem ser escritos na base (u_0, u_1, \dots, u_{N-1}):

$$k_i = \sum_{n=0}^{N-1} w_{in}^* u_n \tag{4.38}$$

As decomposições dos vetores u_n na base KL (4.36) são úteis para relacionar o erro médio quadrático devido à truncagem de X na base U com o erro de truncagem na base KL. Para isto escreve-se o fator $u_n^* R_x u_n$ em termos dos k_i 's:

$$u_n^* R_x u_n = u_n^* R_x \left[\sum_{i=0}^{N-1} w_{in} k_i \right] \tag{4.39}$$

mas como os vetores k_i são os auto-vetores de R_x pela definição de transformação KL,

$$u_n^* R_x u_n = u_n^* \sum_{i=0}^{N-1} w_{in} \lambda_i k_i \tag{4.40}$$

onde λ_i é o auto-valor associado ao auto-vetor k_i . Decompondo u_n^* em função dos k_i 's na equação 4.40 tem-se:

$$\begin{aligned} u_n^* R_x u_n &= \left\{ \sum_{j=0}^{N-1} w_{jn} k_j \right\}^* \sum_{i=0}^{N-1} w_{in} \lambda_i k_i \\ &= \sum_{i=0}^{N-1} |w_{in}|^2 \lambda_i \end{aligned} \quad (4.41)$$

Como U é ortonormal,

$$\sum_{i=0}^{N-1} |w_{in}|^2 = 1 \quad (4.42)$$

e a equação 4.42 pode ser escrita da seguinte forma:

$$u_n^* R_x u_n = \lambda_M + \sum_{i=0}^{N-1} |w_{in}|^2 (\lambda_i \Leftrightarrow \lambda_M) \quad (4.43)$$

$$\begin{aligned} &= \lambda_M + \sum_{i=0}^{M-1} |w_{in}|^2 (\lambda_i \Leftrightarrow \lambda_M) + \\ &\quad \sum_{i=M+1}^{N-1} |w_{in}|^2 (\lambda_i \Leftrightarrow \lambda_M) \end{aligned} \quad (4.44)$$

Lembrando que os auto valores λ_i são ordenados de forma decrescente, os termos $(\lambda_i \Leftrightarrow \lambda_M)$ são todos positivos se $i < M$ e, portanto,

$$\sum_{i=0}^{M-1} |w_{in}|^2 (\lambda_i \Leftrightarrow \lambda_M) > 0 \quad (4.45)$$

Considerando esta inequação e 4.44 obtem-se um limite inferior para $u_n^* R_x u_n$:

$$u_n^* R_x u_n \geq \lambda_M + \sum_{i=M+1}^{N-1} |w_{in}|^2 (\lambda_i \Leftrightarrow \lambda_M) \quad (4.46)$$

Esta inequação, por sua vez, estabelece um limite inferior do erro médio quadrático EMQ (vide equação 4.35):

$$\begin{aligned} \text{EMQ} &\geq \frac{1}{N} \sum_{n \geq M} \left[\lambda_M \Leftrightarrow \sum_{i=M+1}^{N-1} |w_{in}|^2 (\lambda_M \Leftrightarrow \lambda_i) \right] \\ &= \frac{N \Leftrightarrow M}{N} \lambda_M \Leftrightarrow \frac{1}{N} \sum_{i \geq M} (\lambda_M \Leftrightarrow \lambda_i) \sum_{n \geq M} |w_{in}|^2 \end{aligned} \quad (4.47)$$

Mas como $\sum_{n \geq M} |w_{in}|^2 \leq 1$ e $(\lambda_M \Leftrightarrow \lambda_i) \geq 0$ para $i \geq M$,

$$\begin{aligned} \frac{1}{N} \sum_{i \geq M} (\lambda_M \Leftrightarrow \lambda_i) \sum_{n \geq M} |w_{in}|^2 &\leq \\ &= \frac{1}{N} \sum_{i \geq M} (\lambda_M \Leftrightarrow \lambda_i) \\ &= \frac{N \Leftrightarrow M}{N} \lambda_M \Leftrightarrow \sum_{i \geq M} \lambda_i \end{aligned} \quad (4.48)$$

Das inequações 4.47 e 4.48 conclui-se que:

$$\text{EMQ} \geq \frac{N \Leftrightarrow M}{N} \lambda_M \Leftrightarrow \left(\frac{N \Leftrightarrow M}{N} \lambda_M \Leftrightarrow \sum_{i \geq M} \lambda_i \right) \quad (4.49)$$

$$\Rightarrow \text{EMQ} \geq \sum_{i \geq M} \lambda_i \quad (4.50)$$

mas $\sum_{i \geq M} \lambda_i$ é exatamente o erro médio quadrático em que se incorre ao se considerar apenas os M primeiros componentes da transformação de X pela KLT. Portanto, 4.49 prova que a transformação KLT é a que minimiza o erro médio quadrático incorrido ao se considerar apenas os M primeiros termos de um vetor aleatório transformado. ♠

Na codificação de sinais a transformada KL é importante como passo anterior à quantização. O decorrelacionamento das coordenadas facilita a quantização. Isto porque neste caso a quantização escalar (que é bem mais simples e de menor custo computacional) é tão eficiente quanto a vetorial. Além disto, a compactação ótima de energia permite uma divisão também ótima de níveis de quantização entre as coordenadas do vetor transformado.

Para obter a transformada KL é preciso primeiramente conhecer a matriz de autocovariância R_x do vetor aleatório. No caso de imagens o conhecimento a respeito desta matriz é grosseiro, mas ainda assim permite a obtenção de importantes conclusões utilizadas na codificação por transformadas.

Inicialmente modela-se as imagens de forma separável, isto é, possuindo características estatísticas independentes em cada coordenada (normalmente as coordenadas espaciais). Para cada coordenada a matriz de covariância $R_x(\text{coordenada})$ pode ser descrita em função da variância σ das variáveis aleatórias $x(i)$ e dos coeficientes de correlação entre pontos vizinhos ρ_k definidos da seguinte forma:

$$\rho_k = \frac{1}{N} \sum_{i=0}^{N-k} (x(i) \Leftrightarrow \bar{x})(x(i+k) \Leftrightarrow \bar{x})$$

onde \bar{x} é a esperança das variáveis aleatórias $x(i)$.

$$R_x(\text{cada coordenada}) = \sigma^2 \begin{bmatrix} 1 & \rho_1 & \rho_2 & \dots & \rho_{N-1} \\ \rho_1 & 1 & \rho_1 & \dots & \rho_{N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{N-1} & \rho_{N-2} & \rho_{N-3} & \dots & 1 \end{bmatrix}$$

A simplificação do modelo separável descrito nas seções anteriores implica no fato que as variáveis aleatórias $x(i)$ são processos de Markov de ordem 1 em cada coordenada. Assim sendo, todos os coeficientes de correlação ρ_k dependem diretamente do coeficiente ρ_1 :

$$\rho_k = \rho_1^{|k|}$$

e a matriz de covariância resume-se em cada coordenada a:

$$R_x(\text{cada coordenada}) = \sigma^2 \begin{bmatrix} 1 & \rho_1 & \rho_1^2 & \dots & \rho_1^{N-1} \\ \rho_1 & 1 & \rho_1 & \dots & \rho_1^{N-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \rho_1^{N-1} & \rho_1^{N-2} & \rho_1^{N-3} & \dots & 1 \end{bmatrix}$$

A matriz de covariância a duas dimensões especiais é obtida através do produto direto das matrizes de covariância horizontais e verticais:

$$R_x = R_x(\text{horizontal}) \otimes R_x(\text{vertical})$$

Uma vez obtido o modelo simplificado resta validá-lo em um conjunto de imagens reais. A literatura relata que, embora o modelo não seja precisamente adequado para os conjuntos de imagens estudados, a aproximação é razoável. Os coeficientes de correlação entre pontos vizinhos estimados situam-se numa faixa entre 0.8 e 0.95.

A importância deste modelo vem do fato que para esta faixa de parâmetros a KLT é bastante bem aproximada pela transformada cosseno discreta, DCT (*Discrete Cosine Transform*) ([28] e [152]). O livro [28] apresenta comparações entre as duas transformadas para diferentes imagens tomando como parâmetro de comparação a eficiência quanto a compactação da energia dos coeficientes transformados.

Os resultados apresentados mostram que a aproximação da KLT pela DCT é muito boa para regiões de imagens sem arestas. Os resultados nas regiões com arestas são piores porque, tal como será visto na próxima seção, as fontes de imagens apresentam comportamento diferenciado em regiões com ou sem arestas. Portanto o modelo de matriz de covariância que resultou na aproximação da KLT pela DCT perde sua validade nas regiões com arestas.

Mesmo assim, a transformada DCT apresenta características muito importantes para efeito de codificação de imagens:

- Aproximado decorrelacionamento das coordenadas transformadas. Isto significa que a utilização de quantização escalar dos coeficientes transformados não implica em grandes perdas quando comparada à quantização vetorial.
- Boa eficiência na compactação da “energia” dos coeficientes transformados ([28]).
- Unitarismo da transformada. Esta propriedade assegura que a “energia” dos coeficientes transformados preserva (a menos de uma constante de proporcionalidade) a “energia” dos coeficientes originais (“energia” definida como a soma dos quadrados dos valores dos coeficientes).

- Significado físico dos coeficientes transformados. O coeficiente transformado c_{mn} corresponde aproximadamente à componente de frequência horizontal m e frequência vertical n dentro do espectro de frequências das variáveis aleatórias $x(i)$. Este significado físico dos coeficientes pode ser utilizado conjuntamente com as propriedades do olho humano para melhorar a codificação de imagens.
- Implementação através de algoritmos rápidos (mesmo princípio utilizado na FFT).

As três últimas características apresentadas são comuns a muitas transformadas utilizadas na área de processamento de sinais, tais como a transformada de Fourier (FFT), transformada seno (DST - *Discrete Sine Transform*), Transformada de Walsh-Hadamard e a Transformada de Haar. Embora estas outras transformadas com características similares também sejam utilizadas para codificar imagens, a DCT é a que tem mais ampla utilização por causa da aproximação com a KLT (duas primeiras características da lista acima).

4.4 Não Estacionariedade das Imagens

Os modelos preditores e de covariância apresentados neste capítulo pressupõem fontes de imagens estacionárias.

Entretanto as fontes de imagens não são estacionárias. Em particular, as fontes apresentam comportamento bem distinto em regiões vizinhas a arestas e em regiões sem arestas próximas. Além disto, o comportamento temporal também apresenta grandes perturbações quando movimentos bruscos ocorrem.

Uma forma de encarar esta situação consiste em determinar o comportamento mais característico das fontes de imagens e usar este comportamento para modelar as fontes como se estas fossem estacionárias. Esta abordagem, bastante utilizada, permite uma modelagem simplificada das fontes de imagens, mas acarreta discordâncias entre os modelos e as fontes em algumas situações. Estas discordâncias levam a uma codificação ineficiente (informação redundante) ou a uma perda de qualidade na representação das imagens.

Outra abordagem consiste em considerar as fontes de imagens compostas por “sub-fontes” com comportamento distinto. Cada um dos comportamentos possíveis para as fontes forma uma “sub-fonte”. A figura 4.12 ilustra esta abordagem:

A probabilidade de escolha de uma das sub-fontes para descrever o comportamento das fontes é dada pela probabilidade de ocorrência do comportamento correspondente à “sub-fonte”.

O trabalho [114] usa este conceito para modelagem preditiva. O comportamento das fontes de imagens é caracterizado por 6 modelos ou “sub-fontes” preditivas (autoregressivas) e mais uma “sub-fonte” com pontos de imagem descorrelacionados. As “sub-fontes” preditivas tem todas a mesma probabilidade de ocorrência, enquanto que a “sub-fonte” descorrelacionada tem probabilidade de ocorrência bem menor.

O artigo [190] também usa este conceito decompondo as fontes de imagens estáticas em uma “sub-fonte” para regiões sem arestas, outra para regiões com arestas orientadas para uma direção e finalmente outra para regiões com arestas orientadas para mais de uma direção.

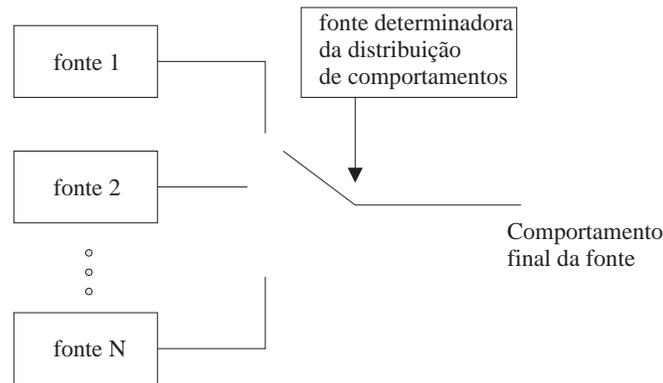


Figura 4.12: Fonte de imagens composta por múltiplas “sub-fontes”.

O comportamento temporal das fontes também pode ser dividido em sub-modelos. De fato, [75] divide o comportamento temporal das fontes de imagens em regiões estacionárias (sem movimento ou com movimentos lentos) e regiões com movimentos rápidos.

Esta abordagem de caracterizar as fontes como uma composição de “sub-fontes” tem maior potencialidade que a alternativa de fontes estacionárias. Isto porque as regiões das imagens podem ser classificadas quanto ao seu comportamento probabilístico e, em seguida, a codificação mais adequada aquele comportamento utilizada na região. Entretanto, a classificação de comportamentos pode trazer desvantagens relacionadas a aumento de complexidade e redução de robustez na codificação.

4.5 Modelos de Correlação das Cores

Nas seções anteriores deste capítulo foram apresentadas características de imagens monocromáticas. No entanto, muitas aplicações utilizam imagens coloridas. É preciso conhecer as características probabilísticas das componentes de cores das imagens.

Conforme estabelecido no Capítulo 1, as cores num ponto são representadas por n_c coordenadas de cores. Na prática o número de coordenadas de cores é 3.

Qual será a melhor representação para as cores? Como as características probabilísticas das matrizes de imagens em cada coordenada de cor se relacionam com as características das imagens monocromáticas?

4.5.1 Classificação das Representações de Cores

Para classificar as representações de cores é preciso saber quais os critérios a serem utilizados:

- Descorrelação estatística
- Aproveitamento das características de percepção da visão humana
- Compatibilidade com os sistemas de cores já existentes

Quanto mais descorrelacionadas forem as coordenadas de cores melhor é a representação. Seja $\{c(n)\}_n$ uma fonte de cores. No instante m a cor gerada é $c(m) = \{c_1(m), c_2(m), c_3(m)\}^T$. Caso as coordenadas c_1 , c_2 e c_3 sejam descorrelacionadas o desempenho da quantização escalar em cada coordenada aproximar-se-á do desempenho da quantização vetorial da fonte. Como a quantização escalar é consideravelmente mais simples que a vetorial, o descorrelacionamento das coordenadas é bastante vantajoso.

Como a transformação de coordenadas de um sistema de representação para outro é feita através de transformações lineares ([200], [101]), a transformação de Karhunen-Loève é a que descorrelaciona as coordenadas. Uma vantagem adicional da transformação KL é a compactação de energia nas coordenadas transformadas. Desta forma pode-se otimizar a alocação de níveis de quantização para cada uma das coordenadas de cor.

Existem estudos na literatura que obtêm as coordenadas descorrelacionadas, chamadas de K_1 , K_2 e K_3 [141], [192] e [158]. Neles a estimação da transformada KL é obtida através de algumas amostras de imagens. A literatura ([101]) relata que a coordenada K_1 se aproxima da luminância Y .

Figura	Sistema de Coordenadas	Energia(%) 1a Comp.	Energia(%) 2a Comp.	Energia(%) 3a Comp.
GAROTA	RGB	45,14	35,41	19,45
	YIQ	78,32	17,54	4,14
	K_1, K_2, K_3	85,84	12,10	2,06
CASAL	RGB	51,55	31,09	17,36
	YIQ	84,84	13,81	1,35
	K_1, K_2, K_3	92,75	6,46	0,79

É interessante observar a tabela anterior. Nela estão apresentadas as percentagens de energias contidas nas coordenadas dos sistemas de representação RGB, YIQ e $K_1K_2K_3$ para duas imagens.

Embora este não seja o capítulo reservado para a descrição das características da visão humana, vale a pena antecipar uma destas características que ajuda a determinar melhor representação de cores.

Experiências envolvendo percepção humana indicam que à medida em que detalhes em imagens vão ficando mais finos, chega um momento em que a visão perde a noção de cor,

ainda que consiga identificar o detalhe. Esta experiência é de comprovação simples. Basta observar um fio de costura. Afastando-se o fio do olho ele vai ocupando cada vez uma parte menor do campo visual, isto é, vai se tornando um detalhe mais fino. Chega uma hora em que o fio ainda pode ser observado, mas sua cor não pode ser identificada. Isto significa que o nível de detalhes espaciais das coordenadas de crominância pode ser menor do que o da coordenada de luminância. Em outras palavras, as coordenadas de crominância podem ser amostradas no espaço de forma mais grosseira do que a coordenada de luminância.

Desta forma, as representações de cores que separam luminância de crominância (YIQ e YUV, por exemplo), tem a chance de aproveitar uma característica da visão humana para reduzir consideravelmente a informação necessária para imagens coloridas.

Estas representações também têm outra vantagem importante: compatibilidade com os sistemas de televisão. Quando os padrões de televisão colorida foram especificados (NTSC, PAL), sistemas de cores contendo a luminância como uma coordenada foram utilizados. Isto foi feito para que o sinal de televisão continuasse a conter a informação necessária para os receptores preto-e-branco. Assim, a informação de cor foi adicionada ao sinal de televisão preto-e-branco.

Portanto, os sistemas de cores que contém a luminância como uma das coordenadas de cores são comumente utilizados para codificar imagens. Eles apresentam compactação de energia e decorrelação nas coordenadas de cor próximas das ótimas (obtidas por transformação de Karhunen-Loève). Além disto a separação de luminância e crominância permite melhor aproveitamento das características de percepção de cor humana.

Alguns trabalhos ([101], [158] e [141]) apresentam estudos a respeito da escolha das coordenadas de crominância a serem utilizadas para minimizar a correlação entre as coordenadas. Embora os resultados destes estudos não sejam conclusivos para escolher as melhores coordenadas de crominância, conclui-se que as redundâncias entre as coordenadas YIQ (usadas no padrão NTSC) e YUV (usadas no padrão PAL) são ambas pequenas. Desta forma, não se perde muito realizando quantização escalar nas três coordenadas de cores separadamente, ao invés de se realizar quantização vetorial.

4.5.2 Comportamento Probabilístico de Imagens Coloridas

A luminância de uma imagem é sua representação monocromática. Portanto, todas as características probabilísticas estudadas para imagens monocromáticas valem para a coordenada de luminância das imagens coloridas. O que acontece com as outras coordenadas ?

As cores presentes nas imagens estão relacionadas com os objetos observados. Estes objetos, por sua vez, tem na grande maioria dos casos, cores com componentes em todas as coordenadas. De fato, um objeto não apresenta componente numa determinada coordenada de cor caso sua cor seja uma combinação das outras duas coordenadas.

Sejam $u_Y(n, m)$, $u_{C_1}(n, m)$ e $u_{C_2}(n, m)$ as matrizes cujos pontos representam respectivamente luminância (Y), coordenada C_1 de crominância e coordenada C_2 de crominância. Como as características probabilísticas dos três sinais (das três matrizes) dependem de uma fonte comum (objetos observados), há uma forte correlação entre elas. Esta correlação faz com que os modelos probabilísticos utilizados para imagens monocromáticas também sejam usados para as coordenadas de cores das imagens coloridas.

Em resumo, o modelo probabilístico de imagens coloridas reduz-se essencialmente a três instâncias do modelo de imagens monocromáticas, uma instância para cada coordenada de cor.

Capítulo 5

Percepção e Codificação de Imagens

5.1 Introdução

Este capítulo apresenta características da visão humana que podem ser exploradas a fim de melhorar a compressão de imagens. A utilização de tais características está baseada no refinamento de detalhes das imagens (com a consequente utilização de maior número de bits para representação) nas regiões em que a visão humana apresenta maior sensibilidade. De forma análoga, o refinamento de detalhes pode ser menor em regiões aonde a visão humana apresenta menor sensibilidade.

O conhecimento das características da visão humana serve de base para o desenvolvimento de métodos de compressão com perdas/erros. Tal conhecimento permite concentrar as perdas onde a visão humana terá menor sensibilidade para percebê-las.

O ser humano não é o único usuário final de imagens. Em diversas aplicações os computadores são os usuários finais. Pode-se citar como exemplo a navegação automatizada. Nestes casos, o projeto de um algoritmo de compressão de imagens otimizado exige o conhecimento das características importantes das imagens sob o ponto de vista do processamento realizado pelos computadores. Em outras palavras, o critério de fidelidade usado para as imagens não é necessariamente o critério da visão humana.

Por outro lado, há aplicações que não permitem a existência de erros na representação de imagens comprimidas. Como exemplos pode-se citar: imagens médicas, imagens espaciais e imagens de objetos de valor histórico. Nestes casos, a representação comprimida tem de conter toda a informação original das imagens. Portanto, nestas situações não se pode usar as características da visão humana para representar de forma diferenciada regiões mais ou menos perceptíveis das imagens. Os algoritmos de compressão usados nestes casos são ditos *sem perdas*.

Ainda que a visão humana não seja o único padrão de qualidade de imagens, e ainda que distorções possam não ser aceitas em certos casos, na grande maioria das situações, os usuários finais são seres humanos e perdas são aceitáveis na representação de imagens. Este capítulo destina-se exclusivamente ao estudo das características da visão humana. O estudo de características de visão usadas em algoritmos específicos de processamento de imagens por computadores está fora do escopo deste trabalho.

Conforme a seção 3.2.4, a definição de um critério de fidelidade baseado na distância perceptual entre imagens é algo bastante difícil. No entanto, o conhecimento das principais características da visão humana já é suficiente para orientar os algoritmos de compressão de imagens.

5.2 Componentes da Visão Humana

A visão humana é composta basicamente por órgãos receptores (olhos), uma unidade de processamento (cérebro), e por nervos que transmitem informação dos olhos para o cérebro.

O olho é o órgão responsável por traduzir estímulos luminosos em impulsos nervosos. A figura 5.1 mostra de forma esquematizada a seção horizontal do olho humano.

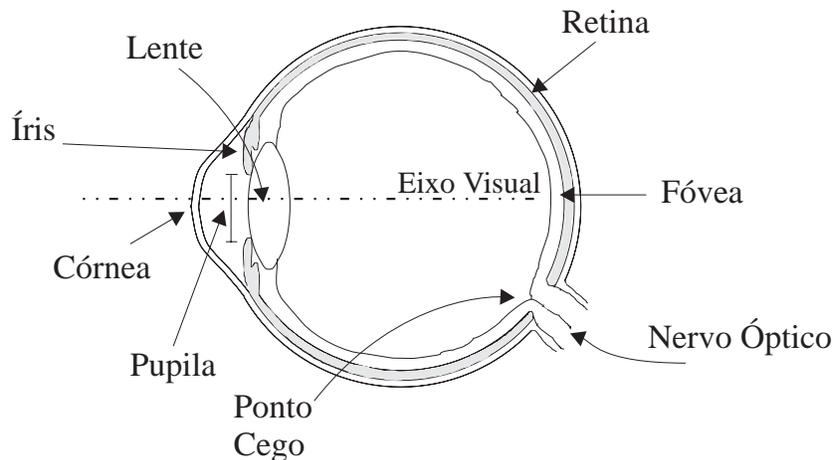


Figura 5.1: Seção reta horizontal do olho humano direito. A seção reta horizontal do olho esquerdo é simétrica.

A luz que vem de um cena chega inicialmente na *córnea*. Esta age como uma lente focalizadora da luz incidente. Em seguida a luz passa por um líquido transparente, chamado *humor aquoso*. Uma parte da luz incidente é barrada pela *íris*. A íris é um anel de musculatura lisa, cujo círculo interior, chamado de *pupila*, pode aumentar ou diminuir de tamanho, controlando assim a quantidade de luz que efetivamente entra no interior do olho.

A luz que passa pela pupila chega a uma lente (*crystalino*). Esta lente é responsável pela focalização dinâmica do olho. O cristalino é uma estrutura biconvexa, elástica e transparente. Através da ação dos *músculos ciliares* seu raio de curvatura pode ser alterado, possibilitando a focalização de objetos situados a distâncias variadas. Quando um objeto distante é observado a lente se alonga e seu poder de convergência diminui, focalizando adequadamente o objeto. O contrário acontece quando um objeto próximo é observado. Esta alteração no formato da lente para focalizar objetos em diferentes distâncias é chamada de *acomodação*.

Após passar pela lente a luz atravessa outro meio transparente: o *humor vítreo*.

Por fim a luz chega à *retina*, induzindo na mesma o aparecimento de impulsos nervosos.

A retina é composta de camadas de células. Uma das camadas é formada de células foto-receptoras, enquanto que as outras camadas, formadas por outras células e seus processos, são responsáveis por conduzir a informação visual até as fibras que compõem o *nervos ópticos*.

Conforme mencionado no capítulo 2, existem dois tipos de células foto-receptoras: os bastonetes e os cones. Os bastonetes existem em maior número (aproximadamente 120 milhões). Eles apresentam sensibilidade mesmo a níveis baixos de iluminação, no entanto não são capazes de detectar cores. Os cones (aproximadamente 7 milhões de células), por outro lado, tem limiar de sensibilidade maior, isto é, necessitam de maior quantidade de iluminação ambiente para produzirem impulsos nervosos. Porém os cones são responsáveis pela visão de cores e nitidez de imagens. Os bastonetes e cones têm natureza aproximadamente complementar. Em cenas bastante iluminadas a visão é dominada pela estimulação dos cones. Nestes casos a visão é dita *fotoptica*. Em cenas com pouca iluminação a visão é essencialmente devida aos bastonetes. Nestes casos a visão é dita *escotópica*.

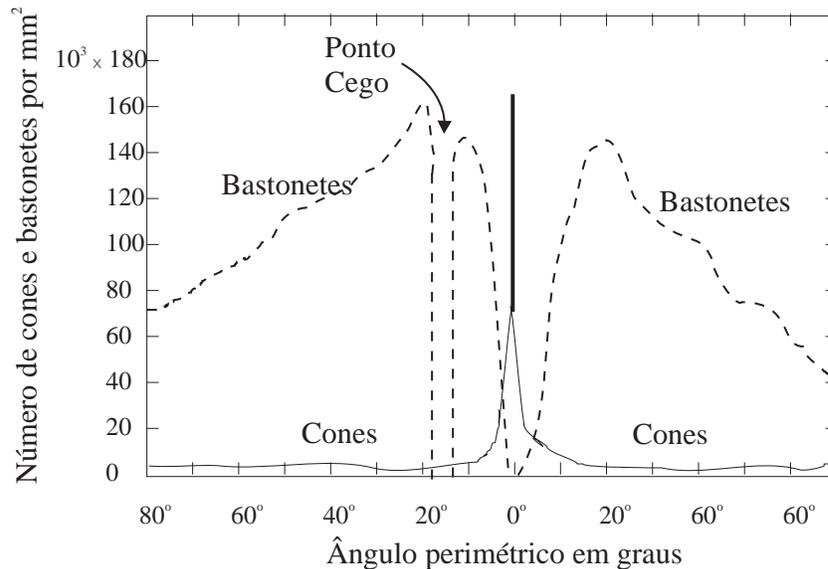


Figura 5.2: Distribuição de cones e bastonetes na seção reta horizontal do olho direito humano. A distribuição é simétrica no olho esquerdo.

Os cones e bastonetes não estão distribuídos de forma uniforme na retina. A figura 5.2 mostra como são estas distribuições.

Os cones estão extremamente concentrados na região central do olho, chamada *fóvea*. De fato, esta é a região mais sensível a cores e menos sensível a baixas iluminações do olho humano.

Após os impulsos nervosos serem gerados, eles são conduzidos pelos nervos ópticos até o sistema nervoso central. Existem cerca de 1 milhão de fibras formadoras do nervo óptico no olho humano. Como há cerca de 130 milhões de células foto-receptoras na retina, para cada fibra óptico há, na média, mais de 100 células receptoras. As fibras do nervo óptico

recebem a composição dos impulsos de um grupo de células receptoras. O número de células receptoras que estimulam uma fibra do nervo óptico varia bastante na retina. Na fóvea, região onde se projeta o eixo visual, o número de células receptoras por nervo óptico é bem menor que no resto da retina. Há fibras ópticas na fóvea que são estimuladas por apenas uma célula foto-receptora. Desta forma, a fóvea é responsável não apenas pela percepção de cor, mas também pela visão detalhada. É interessante observar que embora toda a superfície da retina seja responsável pela visão, quando desejamos observar um objeto com nitidez fixamos o nosso olhar de forma a posicionar e focalizar o objeto na fóvea.

As fibras que constituem o nervo óptico agrupam-se numa região do olho, chamada de *ponto cego*, ao emergir para o cérebro. A região do ponto cego é totalmente ocupada pelo feixe de fibras. Desta forma os objetos de uma cena que são focalizados no ponto cego não são vistos efetivamente.

O fato de não percebermos um ponto de cegueira no nosso campo visual parece contradizer a existência do ponto cego. A visão humana na verdade completa a informação luminosa correspondente ao ponto cego com a informação de regiões vizinhas. O livro ([153]) explica este processo de completamento e apresenta uma experiência comprovando a existência do ponto cego.

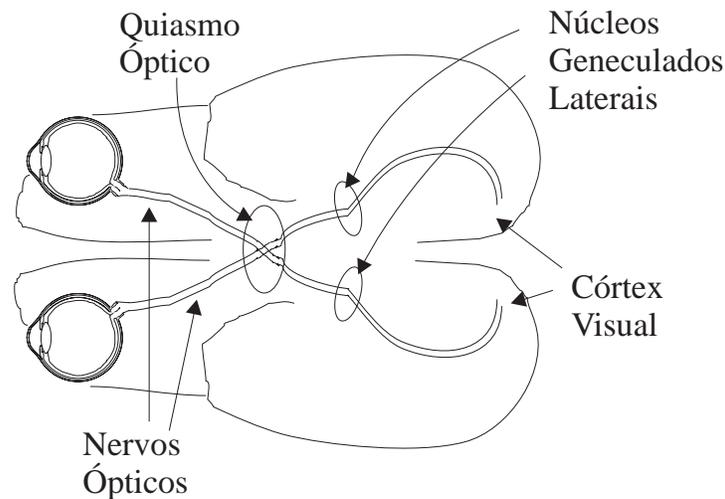


Figura 5.3: Caminho dos feixes de nervos ópticos em direção ao cérebro

A informação que segue pelo nervo óptico percorre um caminho até chegar ao *córtex visual* (região do cérebro diretamente relacionada à visão). Este caminho está ilustrado na figura 5.3.

Os feixes de fibras dos dois olhos se encontram no *quiasma óptico*. Na verdade os feixes de ambos os olhos são divididos em duas partes. Cada parte transporta informação obtida por uma parcela de retina. A seção reta vertical que corta o centro do olho humano, também chamada de seção sagital, divide a retina em duas porções: a *nasal*, mais próxima do nariz; e *temporal*, mais próxima da lateral da cabeça. As informações provenientes da porção temporal

de cada retina projetam-se no córtex visual no mesmo lado (esquerdo ou direito) da cabeça ocupado pela retina (informação temporal da retina do olho esquerdo projeta-se no lado esquerdo do córtex visual, enquanto que a informação temporal da retina do olho direito projeta-se no lado direito do córtex visual). As informações provenientes da porção nasal de cada retina, por outro lado, projetam-se em hemisférios opostos do córtex visual.

Esta mistura de feixes de fibras ópticas permite ao córtex visual processar duas amostras de uma cena. A combinação dos estímulos provenientes das duas retinas é justamente umas das duas principais transformações de informação realizadas pelo córtex visual ([76]).

A outra principal transformação de informação realizada no córtex visual é uma filtragem direcional. O córtex é dividido em grupos de células sensíveis a diferentes orientações da informação luminosa. Cada grupo tem um comportamento correspondente a um filtro passa-faixa bidimensional com faixa de passagem restrita a um conjunto limitado de orientações dentro de um ângulo entre 20° e 40° .

Após estas transformações no córtex os impulsos nervosos prosseguem em seu caminho através do sistema nervoso central. Os demais estágios deste caminho irão traduzir o conjunto de estímulos visuais na visão propriamente dita (esfera consciente).

5.3 Aproximação de Imagens Levando em Consideração a Visão Humana

A visão humana pode ser compreendida como um processo dividido em duas etapas. Numa primeira etapa, realizada pelos olhos, a estimulação luminosa provoca, em última instância, o aparecimento de impulsos nervosos nas fibras do nervo óptico. O resultado desta etapa é uma descrição da cena observada na linguagem elétrica que o cérebro é capaz de entender (potenciais de ação). A segunda etapa consiste no entendimento da cena. Nesta segunda etapa distinguimos se um objeto observado é uma árvore, uma casa ou um cachorro. A primeira etapa é exclusivamente sensorial, enquanto que a segunda etapa é cognitiva. As duas etapas da visão estão ilustradas na figura 5.4.

Existem duas abordagens diferentes para caracterizar a qualidade da aproximação de uma imagem por outra.

A primeira abordagem consiste em avaliar a qualidade da aproximação das imagens sob o ponto de vista sensorial. Em outras palavras, esta abordagem corresponde a avaliar a distância entre os resultados das primeiras etapas da visão das duas imagens. Dentro desta abordagem duas imagens são consideradas próximas caso uma reproduza a outra com boa fidelidade.

A segunda abordagem consiste em avaliar a qualidade da aproximação das imagens sob o ponto de vista cognitivo. Esta abordagem corresponde a avaliar a distância entre os resultados das respectivas segundas etapas da visão das duas imagens. Nesta abordagem duas imagens são próximas caso forneçam ao observador informações parecidas sobre o conteúdo da cena observada.

Desta forma, a determinação de um critério de fidelidade de imagens (ou de forma equivalente, a determinação dos fatores mais importantes na percepção de imagens) depende da abordagem escolhida.

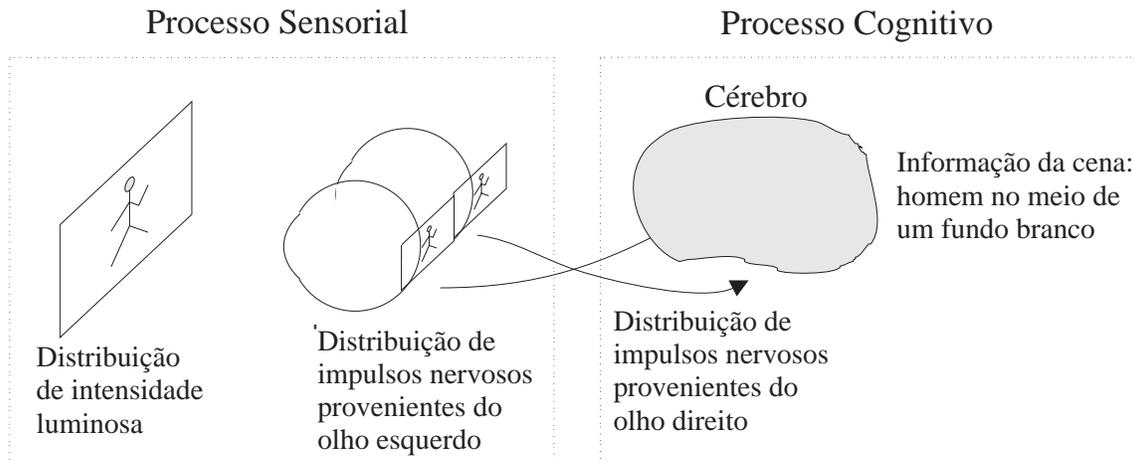


Figura 5.4: O processo visual dividido em duas etapas: uma sensorial e uma cognitiva.

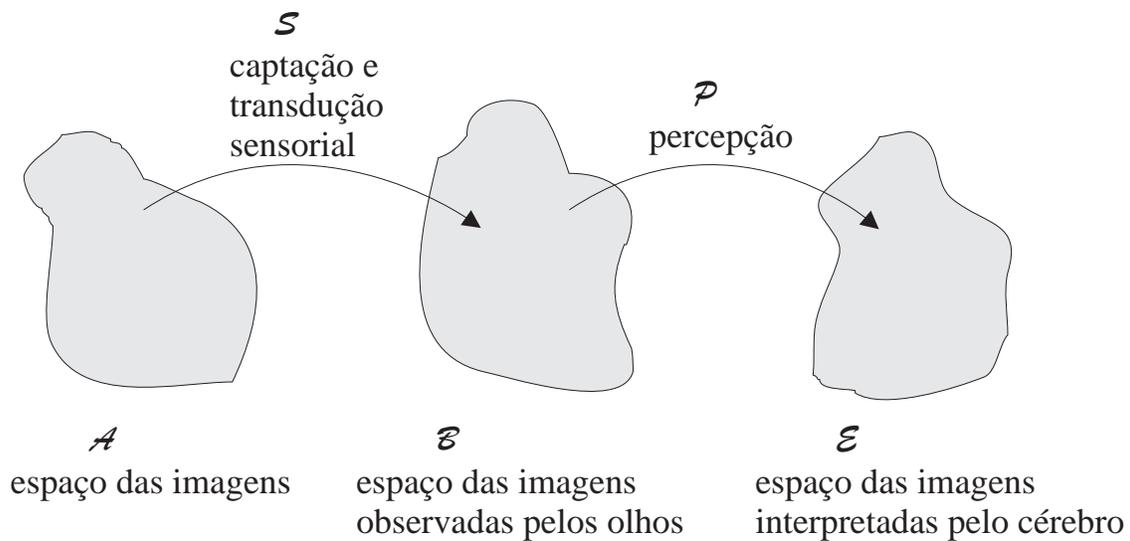


Figura 5.5: O processo visual visto da óptica matemática

Uma notação matemática pode ajudar o entendimento de como o critério de fidelidade se encaixa nos processos da visão. Seja o espaço de imagens \mathcal{A} , o espaço de imagens observadas pelos olhos \mathcal{B} e o processo sensorial $\mathcal{S} : \mathcal{A} \rightarrow \mathcal{B}$. Por outro lado, seja \mathcal{E} o espaço de imagens interpretadas pelo cérebro e $\mathcal{P} : \mathcal{B} \rightarrow \mathcal{E}$ o processo cognitivo. A figura 5.5 ilustra esta notação matemática da visão.

Quando se deseja aproximar uma imagem I_1 por outra I_2 segundo a primeira abordagem, deseja-se reduzir a distância entre $\mathcal{S}(I_1)$ e $\mathcal{S}(I_2)$ no espaço \mathcal{B} . Considere-se $\text{dist}_{\mathcal{B}}$ a distância perceptual entre imagens sob o ponto de vista sensorial e $\text{dist}_{\mathcal{E}}$ a distância perceptual entre

imagens sob o ponto de vista cognitivo.

Seja $B_\varepsilon[\mathcal{S}(I_1)]$ a bola de raio ε centrada no ponto $\mathcal{S}(I_1)$. Uma boa aproximação da imagem I_1 , segundo a abordagem sensorial é qualquer imagem $I_2 \in \mathcal{S}^{-1}\{B_\varepsilon[\mathcal{S}(I_1)]\}$.

Caso o processo \mathcal{P} seja uniformemente contínuo (segundo as distâncias $\text{dist}_{\mathcal{B}}$ e $\text{dist}_{\mathcal{E}}$), uma boa aproximação sensorial implica numa boa aproximação cognitiva. De fato, $\forall \varepsilon > 0, \exists \delta > 0 \mid \forall b_1, b_2 \in \mathcal{B}, \text{dist}_{\mathcal{B}}(b_1, b_2) < \delta \Rightarrow \text{dist}_{\mathcal{E}}(\mathcal{P}(b_1), \mathcal{P}(b_2)) < \varepsilon$. Portanto, se uma distância máxima ε entre imagens em \mathcal{E} for determinada como padrão de qualidade de aproximação cognitiva, basta determinar δ como padrão de qualidade sensorial. Logo, segundo esta hipótese, uma aproximação sensorial implica numa aproximação cognitiva.

Embora não se possa comprovar a continuidade uniforme do processo cognitivo em todo seu domínio \mathcal{B} , a experiência prática mostra que na grande maioria dos casos uma boa aproximação sensorial realmente implica numa boa aproximação cognitiva. Por este motivo, e também porque o processo cognitivo é bem mais complexo e menos conhecido que o sensorial, a abordagem sensorial é a mais utilizada na caracterização da qualidade da aproximação entre duas imagens. Em outras palavras, a maior parte dos algoritmos de compressão procura representar uma imagem através de outra procurando maximizar a fidelidade de informação luminosa.

No entanto, os algoritmos de compressão que procuram maximizar a fidelidade cognitiva das imagens são mais promissores (sob o ponto de vista de compressão de dados). Isto porque o critério de fidelidade cognitivo é menos rigoroso que o sensorial. Podemos entender o conteúdo de uma cena mesmo que seus componentes sejam representados de forma grosseira. Existem trabalhos na literatura que exploram o conhecimento que se tem sobre o processo cognitivo da visão ([135], [173] e [4]).

É importante entender que ambas as abordagens de caracterização de fidelidade (sensorial e cognitiva) são importantes e têm sua faixa de atuação. Quando se deseja comprimir imagens de vídeo, por exemplo, procura-se manter uma fidelidade das imagens (não apenas do conteúdo das imagens). Quando, por outro lado, deseja-se comprimir imagens para que elas possam ser transmitidas por linhas telefônicas e, com isto, possibilitar a comunicação telefônica entre deficientes auditivos, procura-se manter apenas uma fidelidade cognitiva.

5.4 Características da Visão Humana

5.4.1 Caracterização de Cores em Três Dimensões

Algumas características da visão humana já foram utilizadas neste trabalho na determinação do modelo de fontes de imagens. A primeira delas é a representação de cores por um número finito de coordenadas.

Na seção 2.4 do capítulo 2 está descrito o porquê da aproximação do espaço de cores por R^3 . É importante observar a compressão de informação imbutida nesta representação. Uma densidade espectral de energia luminosa, isto é, uma função com domínio no intervalo de luz visível é representada através de três parâmetros. Há uma redução de dimensão no processo sem que haja perda significativa da qualidade da imagem resultante.

5.4.2 Filtragem Passa-Baixas no Tempo

A segunda característica da visão humana importante na codificação de imagens também já foi objeto de explicação no capítulo 2 (na seção 2.5.2). Trata-se da filtragem passa-baixas no tempo. Conforme o descrito no capítulo 2, a informação de uma cena observada em dado instante permanece ativa no sistema visual durante um intervalo de tempo. A percepção humana funciona como um filtro passa baixas capaz de responder apenas a variações de baixas frequências. Em termos práticos isto significa que a visão humana é capaz de perceber um certo número de diferentes observações de uma cena por intervalo de tempo.

Desta forma, a representação de uma cena é realizada através de uma seqüência de imagens amostradas no tempo a uma taxa de N imagens por segundo.

A determinação do número de amostras por segundo depende da banda passante do olho humano. Dois fatores da visão humana são importantes na caracterização da banda-passante temporal da visão humana: percepção de continuidade e cintilação. Estes fatores foram bastante estudados no início do cinema.

O primeiro fator diz respeito ao número mínimo de imagens apresentadas por segundo para que uma cena amostrada no tempo seja percebida como contínua. Este número foi levantado experimentalmente e, como já mencionado, chegou-se à conclusão de que 24 amostras por segundo são suficientes para garantir uma sensação de continuidade de movimentos de uma cena. Este número, no entanto, não é utilizado em todas as aplicações. O padrão de televisão M (utilizado no Brasil e nos EUA, por exemplo), especifica aproximadamente 30 amostras por segundo. Por outro lado, existem métodos de compressão de imagens dinâmicas que utilizam taxas de amostragem mais baixas. Estes métodos, como é o caso da recomendação CCITT H.261 ([104]), resultam em taxa de compressão mais elevadas, porém, em detrimento da continuidade dos movimentos.

O segundo fator está relacionado com a mínima frequência de apresentação de informação luminosa necessária para que um observador não tenha a impressão que a fonte está cintilando. Pode-se entender bem este fator imaginando a seguinte experiência:

- Seja um ponto emissor de luz intermitente. A luz permanece acesa durante um intervalo $\frac{1}{2f_c}$, é apagada em seguida, ficando apagada também por um intervalo $\frac{1}{2f_c}$. O processo de “acende e apaga” processegue periodicamente. Em outras palavras, a fonte de luz tem dois estados: acesa e apagada, havendo uma troca de estado a cada $\frac{1}{2f_c}$ segundos. À medida que o valor f_c aumenta, isto é, aumenta a frequência de mudança de estados da fonte de luz, a visão humana tem mais dificuldade de perceber estas mudanças de estado da luz (cintilações). Quando f_c é suficientemente alta, a visão humana é incapaz de perceber que a luz cintila. Isto é o que acontece, por exemplo, com as lâmpadas fluorescentes. Estas lâmpadas variam seu estado (acesas/apagadas) com uma frequência de 120 variações por segundo (duas vezes maior que a frequência da rede de energia elétrica). No entanto percebe-se a iluminação proveniente das lâmpadas como sendo constante (sem cintilação). Qual seria o valor de f_c (conhecida como CFF “*critical fusion frequency*”, ou “frequência crítica de fusão”) para o qual não mais se percebem visualmente a mudança de estados da fonte de luz?

Verificou-se que esta frequência, denominada frequência crítica de fusão de cintilação, é

proporcional ao logaritmo da iluminação ambiente média (lei de Ferry Porter [184]).

Analisando esta lei pode-se constatar que quanto maior for a iluminação ambiente, maior será a frequência de percepção de cintilação. O cinema usa este princípio. Como as salas de apresentação de filmes são escuras, a frequência de cintilação é menor do que no caso de iluminação natural. Portanto a taxa com que a iluminação dos pontos da tela precisa ser atualizada pode ser menor do que seria numa apresentação à luz do dia.

O cinema usa uma taxa de apresentação de 48 telas por segundo. Na verdade, 24 amostras de uma cena são coletadas a cada segundo. Na hora de realizar a apresentação, cada amostra é apresentada duas vezes em seqüência. Desta forma o cinema consegue garantir percepção de continuidade, evitando a cintilação nas imagens apresentadas. A figura 5.6 ilustra a apresentação das imagens $I_1, I_2, I_3, \dots, I_n$ no cinema.

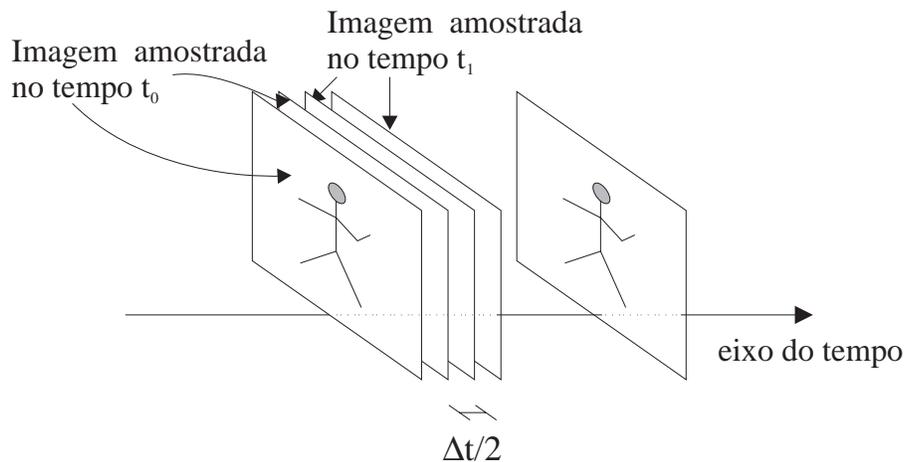


Figura 5.6: Esquema da apresentação de imagens no caso do cinema

A televisão, por outro lado, é geralmente utilizada em ambientes bem mais iluminados que uma sala de cinema. Em um ambiente bem iluminado a apresentação de 48 telas por segundo é insuficiente para evitar cintilação das imagens. Por isto os padrões de representação de imagens usados nas televisões utilizam taxas mais altas de atualização de telas (aproximadamente 60 campos por segundo no caso do padrão M usado no Brasil e nos EUA).

Novamente, em televisão, a frequência relevante para evitar cintilação é duas vezes maior que a frequência relevante para garantir continuidade de movimento. Cada quadro amostrado possui N linhas. A apresentação de cada quadro é feita em duas etapas. Numa primeira etapa as linhas ímpares são apresentadas. Posteriormente as linhas pares são apresentadas. Esta forma de apresentação recebe o nome de varredura entrelaçada, como já mencionado anteriormente. Como pontos correspondentes de duas linhas adjacentes são muito próximos, a visão humana percebe duas atualizações de luz em cada ponto a cada quadro. Desta forma, a cada atualização do conteúdo de imagens (a cada quadro) há duas atualizações das fontes de luz da tela (os pontos da tela de televisão).

5.4.3 Não-Linearidade na Percepção de Intensidade Luminosa

A percepção de intensidade luminosa pela visão humana é estudada através de experiências com observadores humanos.

Uma experiência bastante conhecida é a da determinação do limiar de percepção de diferença entre duas intensidades luminosas. Nesta experiência uma imagem contendo duas áreas de intensidade uniforme é apresentada para um observador. Uma área tem intensidade I , enquanto que a outra área tem intensidade $I + \Delta I$. A diferença ΔI começa inicialmente em 0 e é incrementada até que o observador perceba que a imagem é composta de duas áreas, isto é, perceba a diferença entre as duas intensidades luminosas. Quando chega a este ponto a diferença ΔI é chamada de limiar de discriminação (*just noticeable difference*).

Para determinar o comportamento da visão humana esta experiência é repetida diversas vezes para valores diferentes de I . O comportamento de $\Delta I/I$ à medida em que I varia é dado pelo gráfico da figura 5.7.

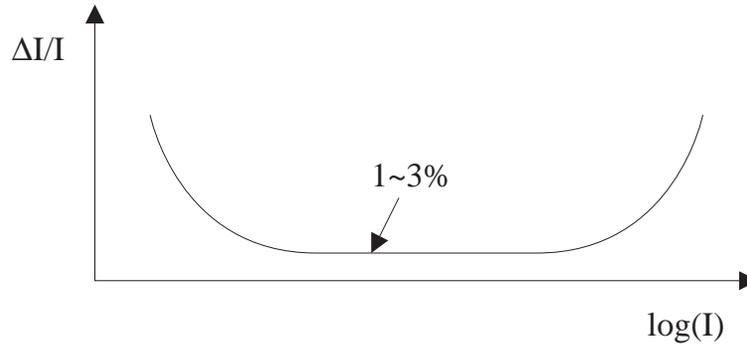


Figura 5.7: Comportamento de $\Delta I/I$ à medida que I varia

Este comportamento foi modelado inicialmente através da *Lei de Weber*:

$$\frac{\Delta I}{I} = \text{constante}$$

Segundo esta lei, a diferença perceptual de intensidade ΔB obedece à lei $\Delta B = \alpha \Delta I/I$, onde α é uma constante de proporcionalidade. Integrando-se esta lei diferencial chega-se à seguinte fórmula: $B = \alpha \log I + \beta$, onde β é uma constante de integração. Esta equação estabelece uma relação entre a intensidade luminosa I e a percepção de intensidade pela visão B . Estudos mais recentes chegaram a uma relação mais precisa entre intensidade luminosa e sua percepção:

$$B = \alpha I^\gamma$$

onde γ vale $1/2$ ou $1/3$ (maiores detalhes em [91], [70]).

Portanto a percepção de intensidade luminosa pelo olho humano não é linear. Variações de intensidade luminosa são mais perceptíveis caso ocorram em regiões escuras (intensidade luminosa baixa). Em outras palavras, um ruído uniformemente espalhado por uma imagem será mais percebido em regiões escuras do que em regiões claras.

Pode-se entender este fenômeno levando-se em consideração a relação sinal ruído nas proximidades de uma região. Em regiões escuras o nível de ruído tem que ser mais baixo para que a relação sinal ruído seja mantida constante.

Esta característica da visão humana oferece um potencial a ser explorado nos métodos de codificação de imagens. Pode-se representar uma imagem reservando-se maior quantidade de bits para caracterizar regiões escuras do que regiões claras.

5.4.4 Filtragem Passa-Faixa no Espaço

A visão humana não apresenta a mesma sensibilidade para todas as freqüências espaciais. Isto é, a visão humana apresenta sensibilidade diferenciada para detalhes mais finos ou mais grosseiros de uma imagem. Isto acontece porque a resposta espectral da visão humana não é uniforme. Mais especificamente, a resposta espectral do olho humano tem uma característica passa-faixas. Esta característica é causada pela composição de um efeito passa-baixas devido ao sistema de lentes e a um efeito passa-altas devido a estrutura das ligações das células nervosas do olho.

O sistema de lentes (córnea + cristalino) do olho age como um filtro passa-baixas com banda passante de aproximadamente 60 ciclos/grau ([162]). Este fato é bastante importante para o correto funcionamento de nossa visão. Isto porque a distribuição de células foto-receptoras em pontos discretos da retina implica numa amostragem espacial do sinal de imagem (aproximadamente 120 células foto-receptoras por grau). O sistema de lentes do olho, portanto, provê um filtro *anti-aliasing* anterior à amostragem espacial da retina (maiores detalhes sobre esta característica podem ser encontrados em [162]). Sakrison comentou este fato de forma interessante em seu artigo [162]: “O projetista do sistema (visão humana) obviamente conhecia com familiaridade o teorema da amostragem”.

As ligações entre células nervosas na retina, por outro lado, são tais que os estímulos causados por uma célula exercem uma influência inibidora sobre as células vizinhas. Este efeito, chamado de inibição lateral, confere à visão uma característica passa-altas (ver [91] para maiores detalhes).

A composição destas duas características resulta num comportamento passa faixas para a visão humana. A resposta espectral da visão humana está ilustrada na figura 5.8.

Esta resposta espectral da visão é muito importante para a codificação de imagens. Uma técnica bastante utilizada para codificar imagens consiste em decompor as imagens em freqüências espaciais. Em seguida a imagem é codificada reservando-se uma quantidade maior de bits para caracterizar as faixas de freqüências em que a visão apresenta maior resposta espectral (maior sensibilidade).

5.4.5 Banda Passante Diferenciada nas Coordenadas de Cores

A literatura ([70]) relata que as cores são percebidas por três tipos de células cones gerando três componentes de cor para a informação visual. Posteriormente estas três componentes são combinadas antes de seus estímulos serem enviados do olho para o cérebro. Esta combinação nada mais é do que uma transformação de coordenadas no espaço de cores. Tal como na representação da TV a cores, uma das novas coordenadas é acromática (luminância), enquanto

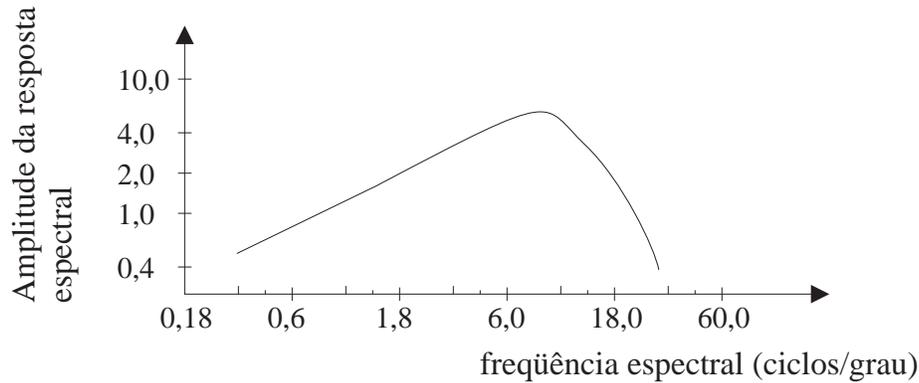


Figura 5.8: Resposta espectral da visão humana

que as duas outras são cromáticas (vermelho menos verde e vermelho menos azul).

As três componentes fluem por canais distintos do olho até o cérebro. Cada canal tem sua resposta espectral. Consta-se na prática que os três canais cortam altas frequências espaciais (portanto são passa-baixas ou passa-faixa). Normalmente estes canais são modelados como filtros passa-baixas espaciais e a banda passante do canal acromático é bem maior do que a dos canais cromáticos. Novamente como comprovação prática simples vale lembrar a observação de fios de costura coloridos citada na seção 4.5.1.

Esta característica da visão humana é de grande utilidade para a codificação de imagens coloridas. Em conjunto com a modelagem de imagens coloridas apresentada na seção 4.5, ela significa que o problema de codificação de imagens coloridas recai na codificação de três componentes distintas da imagem: uma acromática contendo o mesmo número de amostras espaciais da imagem original e duas componentes cromáticas sub-amostradas no espaço (graças a menor banda passante dos canais cromáticos da visão humana).

Este fato é convenientemente usado na codificação de sinais de TV a cores, por exemplo, onde a banda passante (que influencia diretamente na resolução espacial) sinal de luminância (acromático) é de $4,2\text{MHz}$, enquanto que as bandas passantes dos sinais de crominância são respectivamente $1,3\text{MHz}$ e $0,6\text{MHz}$.

5.4.6 Importância das Bordas no Entendimento das Imagens

Experimentos a respeito da percepção de imagens revelaram que a visão humana não é capaz de detectar objetos que permaneçam completamente parados na retina ([97], [154]).

Esta revelação parece um tanto absurda inicialmente, pois conseguimos observar objetos parados quando fixamos nosso olhar. Percebeu-se, no entanto, que os olhos permanecem em constante movimentação involuntária, chamada movimentação *sacádica*. Para se comprovar a não visibilidade de objetos parados na retina realizou-se experiências em que imagens acompanharam os movimentos sacádicos dos olhos. Desta forma os pontos das imagens foram focalizados sempre nas mesmas regiões da retina. Nestas circunstâncias as imagens de teste passaram a não mais serem observadas, ou seja, após alguns segundos necessários para

a “estabilização” da percepção das imagens, os observadores passaram a enxergar imagens uniformemente cinza.

Estas experiências evidenciaram a importância fundamental das variações de intensidade no processo de visão. Estas variações de intensidades correspondem a bordas de imagens estáticas e a bordas e movimentos de imagens dinâmicas.

Uma variação desta experiência, realizada inicialmente por Cornsweet ([31]) e posteriormente aperfeiçoada por Crane e outros ([36] e [37]), manteve uma imagem de teste fixa em relação à retina (com compensação do movimento sacádico) apenas em uma direção (horizontal, por exemplo). A imagem de teste era composta por um retângulo vermelho cercado por fundo verde, como mostra a figura 5.9. Os observadores submetidos ao teste não conseguiram detectar as separações laterais (verticais) entre o retângulo e fundo verde. As partes da imagem correspondentes ao fundo verde posicionado na lateral do retângulo (indicadas por V_1 e V_2 na figura 5.9), não foram vistas com precisão pelos observadores, que ora relataram estarem vendo-as como verde ora como vermelho, mas nunca como uma mistura de cores.

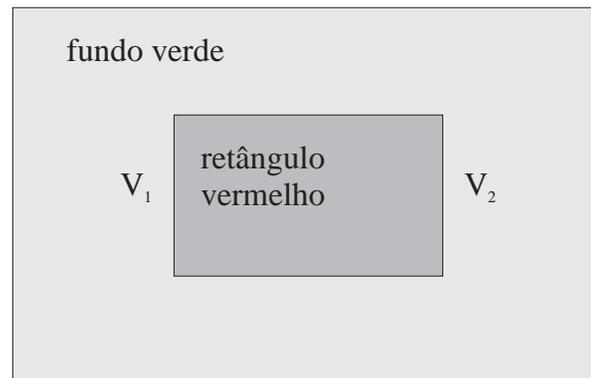


Figura 5.9: Imagem de teste das experiências de Cornsweet e Crane

Esta variante da experiência enfatiza a importância das bordas na percepção de imagens. Além disto ela sugere que o processo de visão humana identifica primeiramente as bordas das imagens e posteriormente completa o restante da informação a partir de pouca informação sobre as regiões entre as bordas.

Os resultados destas experiências são de grande importância para a codificação de imagens. Isto porque eles mostram que o processo de visão realiza uma grande compressão de dados ao desprezar as informações fora das bordas ou movimentos das imagens.

Em relação a abordagem de qualidade de imagens do ponto de vista sensorial (fidelidade na reprodução de imagens), os resultados indicam que caso se consiga reproduzir o processamento da visão humana (detecção de bordas e posterior completamento das regiões entre as bordas) e também se consiga codificar estas informações de bordas de forma reduzida, pode-se obter uma codificação bastante eficiente de imagens.

Os resultados são também de grande relevância para a codificação da informação cognitiva de imagens (qualidade do ponto de vista cognitivo). Isto porque eles revelam quais as porções das imagens que são mais importantes para a percepção (bordas e movimentos). Portanto

também neste caso a codificação das bordas e movimentos torna-se muito importante.

Vários trabalhos têm sido realizados com o objetivo de codificar imagens separando suas informações em duas componentes: bordas e não-bordas (texturas). O capítulo 14 é dedicado a este tipo de codificação.

5.4.7 Conteúdo de Informação das Bordas

Dado o bordo de um objeto, quais são os pontos mais importantes para caracterizá-lo na visão humana? Para responder a esta pergunta Attneave ([9]) realizou a seguinte experiência: forneceu imagens formadas apenas por contornos para diferentes observadores e pediu a eles que escolhessem os N pontos que melhor as caracterizassem. O número de pontos de caracterização N foi convenientemente escolhido para cada imagem da experiência.

Em seguida Attneave dividiu os contornos em pequenos arcos, contou as incidências de pontos por arcos e elaborou histogramas para quantificar o resultado da experiência.

O resultado obtido foi que os pontos de caracterização foram posicionados próximos aos pontos onde o valor absoluto da curvatura assumia máximos locais. Isto evidencia a importância dos pontos de maior curvatura na caracterização da informação de bordas pela visão humana. Este fato pode ser usado na codificação de imagens com taxas de compressão muito altas para classificar a importância das bordas a serem codificadas.

5.4.8 Comportamento Anisotrópico da Percepção Visual em Regiões Próximas de Arestas

Em regiões de intensidade luminosa uniforme (regiões sem arestas próximas), a percepção de imperfeições (ruído) na representação das imagens é aproximadamente isotrópica (independente de direção).

Em regiões próximas de arestas, por outro lado, a percepção de ruído é anisotrópica. Isto se deve à importância das arestas na interpretação das imagens. Em particular a caracterização das arestas é mais importante que da caracterização de áreas uniformes na percepção das imagens. Desta forma, em regiões próximas às arestas é mais importante caracterizar com precisão o posicionamento as arestas do que a intensidade luminosa dos pontos da vizinhança. Portanto, a percepção de ruído é maior nas direções das arestas do que nas direções ortogonais às arestas.

Esta característica da visão oferece um grande potencial para a área de codificação de imagens. Isto porque pode-se representar uma região próxima a uma aresta essencialmente com a informação necessária para caracterizar o posicionamento da aresta. Os demais pontos da vizinhança da aresta podem ser representados de forma bem grosseira, através de uma pequena quantidade de bits.

Este fenômeno pode ser chamado de mascaramento da percepção da visão humana em regiões próximas a arestas. Ele foi utilizado, por exemplo, em [189] e [190].

5.4.9 Redução de Sensibilidade em Áreas em Movimento

De forma análoga ao mascaramento anisotrópico de imperfeições em regiões próximas a arestas, existe também um efeito de mascaramento da percepção visual em regiões onde há

movimento.

Capítulo 6

Concepção Geral da Codificação de Imagens

6.1 Introdução

Este capítulo apresenta uma concepção geral dos métodos de codificação de imagens segundo a qual os métodos são decompostos em três fases:

- Transformação de representação de imagens
- Quantização
- Atribuição de códigos binários a elementos de alfabetos de fontes

Esta decomposição será utilizada nos capítulos seguintes para analisar as diferentes técnicas e ferramentas existentes para realizar cada uma das três fases.

6.2 Estrutura Geral para o Estudo de Codificação de Imagens

Conforme visto no capítulo 2, a codificação de imagens pode ser compreendida de forma bastante adequada através do modelo de sistemas de comunicações. A figura 2.1 no capítulo 2 ilustra o modelo de sistemas de comunicações.

O capítulo 2 teve como objetivo estudar as fontes de imagens. Como resultado as fontes de imagens foram modeladas como processos estocásticos discretos \mathcal{X}_n com alfabeto comum \mathcal{A} . \mathcal{A} é o conjunto de matrizes $i \times j \times n_c$ de números inteiros dentro de um conjunto finito $I_{intensidade} \subset Z$ com k elementos. Os valores i e j são respectivamente o número de linhas e colunas de cada uma das n_c componentes de cor das imagens. Desta forma, uma fonte de imagens gera objetos com dimensão ijn_c a uma taxa de P imagens por unidade de tempo. Portanto a taxa de bits para representar a fonte sem compressão de informação é $ijn_c P \log_2(k)$ bits por unidade de tempo.

Os codificadores de imagens têm como objetivo receber as imagens geradas por uma fonte \mathcal{X}_n e transformá-las numa seqüência de bits a ser enviada através de um canal de

comunicações. No receptor a seqüência de bits é novamente convertida em imagens $\tilde{\mathcal{X}}_n$. A qualidade das imagens resultantes do processo de codificação será medida através de um critério de fidelidade perceptual entre \mathcal{X}_n e $\tilde{\mathcal{X}}_n$.

Um codificador será tão mais eficiente quanto menor for a taxa de bits necessária para a transmissão de imagens codificadas mantendo um determinado padrão de qualidade. De forma equivalente, a eficiência de um codificador será medida através do padrão de qualidade das imagens codificadas mantendo uma determinada taxa de bits para comunicação.

A figura 6.1 apresenta o ambiente típico de codificação de imagens.

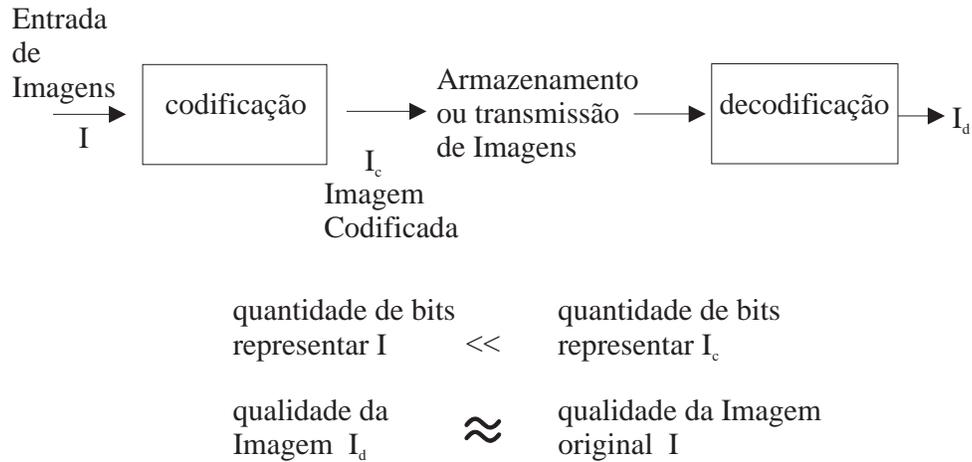


Figura 6.1: Ambiente de codificação de imagens

Os métodos de codificação de imagens podem ser decompostos em três fases distintas, conforme visto na seção anterior:

- Fase 1: Transformação da representação das imagens.
- Fase 2: Quantização ou agrupamento de imagens “próximas” em imagens representativas
- Fase 3: Atribuição de códigos binários a elementos do alfabeto resultante.

A figura 6.2 ilustra as três fases da codificação de imagens.

Na primeira fase uma transformação de representação das imagens faz com que a fonte de imagens seja alterada. Nesta etapa podem ser (e normalmente são) modificados a taxa de geração de informação, a dimensão de cada informação gerada e o significado das coordenadas da informação.

Os próximos capítulos trazem descrições das diversas transformações de representações encontradas hoje em dia. Como um exemplo pode-se considerar a transformação que leva uma seqüência de matrizes ijn_c com elementos pertencentes a um conjunto finito de cardinalidade k geradas a uma taxa de P matrizes por unidade de tempo numa seqüência de inteiros pertencentes a um conjunto finito de cardinalidade k_{n_c} gerados a uma taxa de ijP inteiros

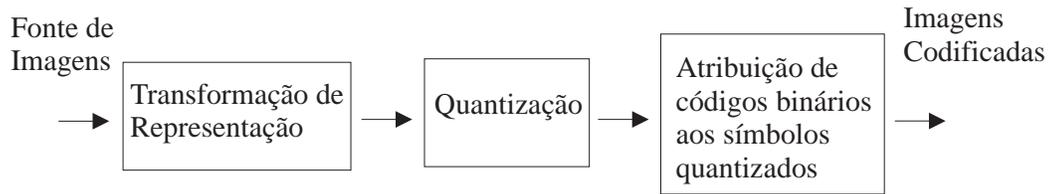


Figura 6.2: As fases da codificação de imagens

por unidade de tempo. Esta transformação divide o período de geração de matrizes em ij intervalos. Em cada intervalo um inteiro é gerado. No intervalo m considera-se a n_c -upla formada pelos elementos $(a_{m/i, m\%i, 0}, \dots, a_{m/i, m\%i, n_c-1})$ e gera-se um inteiro a partir dela: $I_m = \sum_{r=0}^{n_c-1} a_{m/i, m\%i, r} \times k^r$. $m\%i$ denota o valor de m módulo i . A seqüência I_m descreve a fonte de imagens original com periodicidade, dimensão e significado diferentes para a informação gerada.

Uma característica importante do exemplo de transformação de informação apresentado é que ele não redunde em perda de informação. Dito de outra forma, toda a informação da fonte \mathcal{X}_n está contida na fonte I_m . Nem sempre esta característica é válida para as transformações de representação das fontes de imagens.

A próxima fase do processo de codificação de imagens consiste na QUANTIZAÇÃO. A quantização é um processo que reduz a cardinalidade do espaço dos sinais observados. Ela caracteriza-se por uma decomposição celular finita do espaço do sinal original. Todos os sinais contidos dentro de uma mesma célula passam a ser representados por um único sinal. Este sinal representante de toda a célula é também chamado de sinal quantizado.

O papel da quantização neste instante é o de agregar imagens suficientemente próximas em imagens representantes ou quantizadas.

A quantização nesta fase não é a primeira realizada no sinal de imagem. A quantização no espaço de cores tem que ser realizada previamente para que as imagens possam ser representadas por computadores.

A combinação da transformação de representação com a quantização pode trazer bons resultados de compressão de imagens. Isto porque a transformação pode resultar numa representação na qual informações importantes para a percepção humana de imagens fiquem concentradas em alguns coeficientes, enquanto que outras informações de menor relevância perceptual fiquem concentradas em outros coeficientes. Desta forma, a transformação de representação pode “separar” ou “desacoplar” informações de grande relevância de informações pouco importantes. A quantização realizada em seguida pode caracterizar com maior detalhamento a informação relevante do que a informação irrelevante. Em termos de decomposição celular do espaço de informações isto significa considerar células com dimensões pequenas nas direções dos eixos de informação mais relevantes. Este procedimento também pode ser entendido através da utilização de um maior número de bits para caracterizar informação relevante do que para caracterizar informação menos relevante.

É importante observar que as imagens quantizadas formam um conjunto finito. Portanto as imagens resultantes do processo de quantização podem ser interpretadas como a saída de

uma fonte de informação com alfabeto finito.

A última fase da codificação de imagens consiste na atribuição de códigos binários a cada uma das mensagens resultantes do processo de quantização. Nesta etapa existe o potencial para mais alguma redução de taxa de bits necessária para representação de imagens. Pode-se entender intuitivamente esta potencial redução imaginando que a mensagens muito freqüentes associa-se códigos com poucos bits, enquanto que a mensagens pouco freqüentes associa-se códigos com maior número de bits. Em outras palavras, reserva-se códigos binários com poucos bits para mensagens muito freqüentes. Desta forma, a taxa média de bits necessária para representar a fonte de mensagens pode ser reduzida.

Os problemas envolvidos nas fases de quantização e atribuição de códigos binários a mensagens são bastante conhecidos e não apresentam tantas diferentes abordagens quanto os problemas da fase de transformação de representação de fontes de imagens. Os dois próximos capítulos apresentam os detalhamentos (problemas, características e diferentes abordagens) destas duas fases da codificação de imagens.

A fase de transformação de representação, por outro lado, apresenta um leque bem maior de abordagens e é fruto de mais ativa pesquisa. Por isto um capítulo não é suficiente para apresentar seu detalhamento. A apresentação será feita através de um capítulo inicial, contendo as características comuns aos diversos métodos da fase de transformação de representação, seguido de seis outros capítulos correspondendo às particularidades de cada método.

Capítulo 7

Quantização

A quantização consiste no mapeamento do alfabeto \mathcal{A} de uma fonte em um conjunto $\mathcal{A}_Q \subset \mathcal{A}$ de representantes dos elementos de \mathcal{A} . Em outras palavras a quantização é uma função $Q : \mathcal{A} \rightarrow \mathcal{A}_Q$.

No caso geral \mathcal{A} é o alfabeto de uma fonte arbitrária e \mathcal{A}_Q é um sub-conjunto qualquer de \mathcal{A} (embora seja normalmente um finito). No caso da compressão de imagens \mathcal{A} é o conjunto dos símbolos que compõem a saída de etapa de transformação de representação de imagens. Ainda no caso das imagens, o conjunto \mathcal{A}_Q — alfabeto na saída da quantização — é finito. A seus elementos serão atribuídos códigos binários na etapa posterior da codificação de imagens.

A quantização divide o alfabeto da fonte em conjuntos disjuntos. Para cada elemento $b \in \mathcal{A}_Q$ existe um conjunto correspondente:

$$Q^{-1}(b) = \{a \in \mathcal{A} \mid Q(a) = b\}$$

Nas aplicações práticas de quantização os conjuntos $Q^{-1}(b)$ são conexos e formam células. Por isto a quantização equivale a uma decomposição celular do alfabeto da fonte.

A avaliação da perda ou distorção do processo de quantização é feita através do conceito de critério de fidelidade, apresentado no Capítulo 4. Seja uma fonte de informação $\{\mathcal{F}_n\}_n$ e uma função $d : \mathcal{A} \times \mathcal{A} \rightarrow R^+$ que quantifica a distorção causada pela quantização. Intuitivamente esta função d calcula a distância entre um elemento do alfabeto da fonte $a \in \mathcal{A}$ e seu representante após a quantização $Q(a) \in \mathcal{A}_Q$. Entretanto, esta função não precisa ser uma distância, conforme mencionado na seção 3.2.4 do capítulo 3.

No caso de fontes estacionárias e com função de distorção invariante no tempo o critério de fidelidade na quantização é dado por:

$$D = E_{\mathcal{A}}[a, Q(a)]$$

onde $E_{\mathcal{A}}$ é a esperança matemática calculada em relação à fonte $\{\mathcal{F}_n\}_n$.

A quantização inicialmente pode ter um dos dois seguintes objetivos:

- Minimizar o critério de fidelidade D dado que o conjunto \mathcal{A}_Q tem um número fixo de elementos.

- Minimizar o número de elementos de \mathcal{A}_Q dado que o critério de fidelidade D é menor ou igual a um valor fixo.

Na prática estes objetivos podem se misturar.

Algumas restrições podem ser feitas às funções de quantização Q . Pode-se desejar apenas células $Q^{-1}(b_i), b_i \in \mathcal{A}_Q$ retangulares, por exemplo. É claro que este tipo de restrição reduz o universo de funções candidatas à otimizar a quantização. No entanto simplifica-se a busca de quantizadores ótimos e o próprio processamento dos quantizadores.

A otimização na quantização depende

- do modelo da fonte
- da função de distorção d
- das restrições que são feitas às funções de quantização.

7.1 Exemplo de Quantização: Caso Unidimensional

Neste instante convém apresentar um exemplo. Seja um sinal unidimensional $\{f_n\}_n$ (alfabeto R) com distribuição de probabilidade de ocorrência p_{f_n} . O critério de fidelidade vale:

$$D = \int_R d(f, Q(f)) p_{f_n}(f) df$$

No caso de sinais unidimensionais a decomposição em células conexas implica necessariamente em decomposição em intervalos. Desta forma o conjunto R é decomposto em L intervalos disjuntos:

$$R = I_0 \cup I_1 \cup \dots \cup I_{L-1}$$

onde

$$\begin{aligned} I_0 &= (-\infty, r_0] \\ I_1 &= (r_0, r_1] \\ &\vdots \\ I_{L-2} &= (r_{L-3}, r_{L-2}] \\ I_{L-1} &= (r_{L-2}, \infty] \end{aligned}$$

Para cada intervalo I_k será determinado um representante q_k .

Portanto a função $Q : \mathcal{A} \rightarrow \mathcal{A}_Q$ associa todos os elementos de um intervalo ao representante do intervalo. Q pode ser escrita como:

$$Q = \sum_{k=0}^{L-1} q_k \mathcal{X}_{I_k}$$

onde \mathcal{X}_{I_k} é a função característica do intervalo I_k .

A otimização da quantização consiste em achar as bordas $\{r_0, r_1, \dots, r_{L-2}\}$ e os representantes $\{q_0, q_1, \dots, q_{L-1}\}$ dos intervalos de forma que D seja minimizado (L fixo).

A função de distorção mais utilizada é o quadrado da diferença, isto é,

$$d(a, b) = (a \Leftrightarrow b)^2$$

Com esta função de distorção o critério de fidelidade fica:

$$D = \int_R (f, Q(f))^2 p_{f_n}(f) df \quad (7.1)$$

$$= \sum_{k=0}^{L-1} \int_{I_k} (f \Leftrightarrow q_k)^2 p_{f_n}(f) df \quad (7.2)$$

Neste contexto D é um campo real a $2L \Leftrightarrow 1$ variáveis: $r_0, r_1, \dots, r_{L-2}, q_0, q_1, \dots, q_{L-1}$. Uma condição necessária para pontos de mínimo do campo D é que o valor do gradiente ∇D nestes pontos seja 0, ou seja:

$$\frac{\partial D}{\partial q_k} = 0, \quad 0 \leq k \leq L \Leftrightarrow 1 \quad (7.3)$$

$$\frac{\partial D}{\partial r_k} = 0, \quad 0 \leq k \leq L \Leftrightarrow 2 \quad (7.4)$$

Desenvolvendo-se as equações 7.3 e 7.4 chega-se ao seguinte sistema de equações não lineares:

$$q_k = \frac{\int_{I_k} f p_{f_n}(f) df}{\int_{I_k} p_{f_n}(f) df} \quad 0 \leq k \leq L \Leftrightarrow 1 \quad (7.5)$$

$$r_k = \frac{q_k + q_{k+1}}{2} \quad 0 \leq k \leq L \Leftrightarrow 2 \quad (7.6)$$

Este sistema não tem solução analítica para distribuições de probabilidade p_{f_n} genéricas. No entanto, o problema foi resolvido para distribuições uniformes, Gaussianas e Laplacianas, e nestes casos os pontos de anulamento do gradiente de fato são pontos de mínimo do campo D . O livro [100] traz tabelas com os valores ótimos de $r_0, r_1, \dots, r_{L-2}, q_0, q_1, \dots, q_{L-1}$ para as três distribuições de probabilidade citadas com valores de $L = 2, 4, 8, 16$. É interessante observar que a quantização uniforme (aquela que divide o conjunto aonde o sinal toma valores em intervalos de mesmo comprimento) é de fato a quantização ótima para o caso de distribuições uniformes.

7.2 Caso Multidimensional

Na maioria das aplicações de quantização na codificação de imagens os sinais a serem quantizados não são unidimensionais. Um exemplo típico ocorre com os coeficientes de transformadas do sinal de imagem. Estes coeficientes formam uma matriz (maiores detalhes serão apresentados no próximo capítulo).

O processo de quantização nestes casos tem duas abordagens:

- **Quantização Escalar:** na qual cada dimensão do sinal é considerada separadamente. Um processo de quantização unidimensional é realizado para cada dimensão do sinal.

- Quantização Vetorial: na qual a otimização é realizada considerando todas as dimensões do sinal de forma agregada.

No caso da quantização escalar quando várias dimensões são agregadas é preciso tomar cuidado para determinar o número de intervalos de quantização para cada dimensão. Nestes casos não mais se tem L intervalos de quantização que devem ser determinados para minimizar a distorção. Tem-se, por outro lado, uma quantidade máxima de paralelepípedos que podem ser utilizados. Se cada uma das n dimensões do sinal contribuir com L_j intervalos, o número total de células resultantes será $\mathcal{L} = \prod_{j=0}^{n-1} L_j$. Os L_j precisam ser determinados de forma a minimizar a distorção no sinal n -dimensional e satisfazendo a condição $\mathcal{L} \leq L$.

O problema de determinar os valores L_j é intratável analiticamente para sinais arbitrários. No entanto, para o caso das dimensões possuírem as mesmas distribuições de probabilidades, diferindo apenas pela variância, existe uma regra prática bastante útil:

- Os valores L_j devem ser feitos proporcionais às variâncias dos sinais nas n dimensões.

A quantização escalar tem como vantagem principal a simplicidade. A decomposição de R em intervalos é consideravelmente mais simples que a decomposição celular de R^n .

Outro aspecto que torna o uso desta quantização atraente é a fácil modelagem geométrica das células da decomposição. No caso unidimensional a representação das células é feita através dos bordos dos intervalos de quantização. Quando várias quantizações unidimensionais são agregadas as células tornam-se paralelepípedos retangulares (produtos cartesianos de intervalos reais). A representação destas células é simples.

No caso da quantização vetorial as células podem ter formatos arbitrários. A representação de conjuntos arbitrários em R^n é bastante mais complicada do que no caso de paralelepípedos. Além disto, a caracterização da pertinência de elementos da fonte às células da quantização também fica mais difícil. Por isto os algoritmos de quantização vetorial são mais complexos e mais caros computacionalmente.

A quantização vetorial tem duas vantagens sobre a escalar:

- Pode-se considerar a dependência (linear ou não-linear) entre as diferentes dimensões do sinal quantizado.
- A decomposição de R^n em células não retangulares pode ser mais eficiente do que em células retangulares. Em outras palavras, pode-se aproveitar a flexibilidade dos formatos de células para dividir o R^n em um menor número de células mantendo limitada a máxima distância entre os representantes e os demais pontos das respectivas células. Para exemplificar esta vantagem basta lembrar que o R^2 é decomposto em hexágonos de forma mais eficiente do que em retângulos.

Dentre estas duas vantagens a primeira é a principal. Esta vantagem é tanto maior quanto mais forem dependentes os sinais quantizados. Portanto, se as dimensões de sinal a ser quantizado são bastante independentes, a quantização vetorial não oferece grande vantagem sobre a escalar.

O problema de obtenção de quantizadores vetoriais é insolúvel analiticamente para a grande maioria dos casos. As abordagens mais utilizadas baseiam-se em métodos numéricos de otimização.

Na prática os algoritmos de determinação de quantizadores vetoriais fixam o número de representantes de sinais a serem encontrados. Este número equivale ao número de células da decomposição do alfabeto. Os algoritmos partem de uma estimativa inicial dos representantes e executam iterativamente procedimentos de convergência da estimativa para um mínimo local da distorção entre o alfabeto e os representantes.

As células são determinadas indiretamente. Seja $A = \{x_0, \dots, x_{N-1}\}$ um conjunto de representantes do alfabeto. As células são dadas por:

$$C'_i = \{a \in \mathcal{A} \mid d(a, x_i) \leq d(a, x_j), j \in [0, N \Leftrightarrow 1] \Leftrightarrow \{i\}\}$$

onde d é a função de distorção para o alfabeto \mathcal{A} .

Segundo esta definição podem haver pontos pertencendo a mais de uma célula. Nestes casos faz-se uma escolha arbitrária da célula a qual o ponto pertence. Após as eventuais escolhas as células são denotadas por C_i (sem o \prime).

Esta determinação implícita de células é muito vantajosa sob o ponto de vista de modelagem geométrica. O modelo das células no R^n é composto dos representantes da fonte mais a função de distorção. A avaliação de pertinência de um ponto a numa célula C_i consiste na verificação das condições:

- $a \in C'_i$, isto é, $d(a, x_i) \leq d(a, x_j) \forall j \in [0, N \Leftrightarrow 1] \Leftrightarrow \{i\}$
- Se a pertence a mais de um C'_k x_i foi entre os possíveis candidatos o escolhido para representar a .

O processo de determinação dos representantes pode ser feito a partir de diferentes estimativas. Desta forma pode-se escolher o mais favorável dentre os mínimos locais encontrados.

Dois condições necessárias de otimização são usadas para pautar os métodos de projeto de quantizadores vetoriais.

- Condição de Pertinência: $a \in C'_i \Leftrightarrow d(a, x_i) \leq d(a, x_j) \forall j \in [0, N \Leftrightarrow 1] \Leftrightarrow \{i\}$
- Condição de Centróide: x_i tem de minimizar a distorção $E[d(a, x_i) \mid a \in C_i]$, isto é, x_i tem de ser o centróide de C_i em relação à distribuição de probabilidade da fonte.

Em princípio pode-se iniciar o projeto de quantizadores com uma estimativa $A_0 = \{x_0^{(0)}, \dots, x_{N-1}^{(0)}\}$. Em seguida determina-se as células C_i . A partir das células calcula-se os respectivos centróides, $A_1 = \{x_0^{(1)}, \dots, x_{N-1}^{(1)}\}$ que servirão como nova estimativa de representantes.

A utilização do método tal como apresentado acima tem dois problemas práticos:

- Caso o alfabeto seja muito grande (ou ilimitado), fica muito caro (ou inviável) computacionalmente determinar todos os elementos das diversas células.
- Muitas vezes não se dispõe de distribuições de probabilidade da fonte para poder calcular os centróides.

O algoritmo LBG, proposto por Y. Linde, A. Buzo e R.M. Gray ([103]), utiliza exatamente esta idéia básica evitando os problemas práticos.

O algoritmo utiliza apenas uma amostra do alfabeto para efetuar as iterações. Tal amostra $B = \{y_0, \dots, y_{M-1}\}$ é tomada de forma que $M \gg N$, isto é, a cardinalidade da amostra é muito maior que a cardinalidade do conjunto de representantes.

O primeiro problema prático é “resolvido” porque a condição de pertinência é testada apenas no conjunto amostra e não no alfabeto inteiro. O segundo problema é “resolvido” porque todos os elementos dentro de uma célula são considerados equiprováveis. O algoritmo na verdade não resolve os problemas práticos. Ele evita os problemas usando uma aproximação do modelo da fonte. O algoritmo cede em otimização para ganhar em praticidade.

O conjunto de representantes da fonte é chamado de *livro-código* – *codebook*.

Maiores detalhes sobre o algoritmo de LBG, bem como outras variantes do método podem ser encontradas no próprio artigo [103] ou no livro [100].

7.3 Aplicações da Quantização

A quantização é uma ferramenta de aproximação de elementos de um alfabeto por elementos de um de seus subconjuntos. Dentro deste contexto, ela se aplica a qualquer grandeza resultante do processo de transformação de representação.

Muitos trabalhos com o objetivo de aplicar eficientemente a quantização na codificação de imagens têm sido desenvolvidos. As próximas seções apresentam maiores detalhes sobre as principais áreas de aplicação, a saber, Quantização Vetorial no Espaço e Quantização de Coeficientes Transformados.

7.3.1 Quantização Vetorial no Espaço

Esta aplicação da quantização é realizada diretamente nas intensidades luminosas da imagem, sem que haja nenhuma transformação de representação anterior. Os vetores a serem quantizados são subconjuntos de imagens com tamanho típico 4×4 , costumeiramente chamados de blocos.

A base teórica para a compressão de dados consiste na otimização da aproximação dos vetores do universo original por um conjunto finito de representantes. É importante perceber que a quantização por si só não se responsabiliza por reduzir a redundância estatística das fontes.

O primeiro passo da codificação é a determinação do *livro-código* (*codebook*). Uma vez que este esteja determinado e seja de conhecimento tanto do codificador quanto do decodificador, um bloco será codificado através do índice de seu representante no livro-código. A codificação de uma imagem completa é composta pelos índices dos representantes dos diversos blocos ordenados de acordo com a varredura da imagem. A figura 7.1 ilustra esta codificação.

Para que a codificação tenha resultados ótimos é preciso que a função de distorção utilizada na escolha dos representantes dos blocos de imagens represente a distorção percebida pela visão humana. Como tal medida de distorção perceptual é, na prática, muito difícil de ser obtida, usa-se normalmente outras distorções mais simples e compensa-se esta perda com o uso de outras técnicas, como a classificação das imagens antes da quantização. As próxima

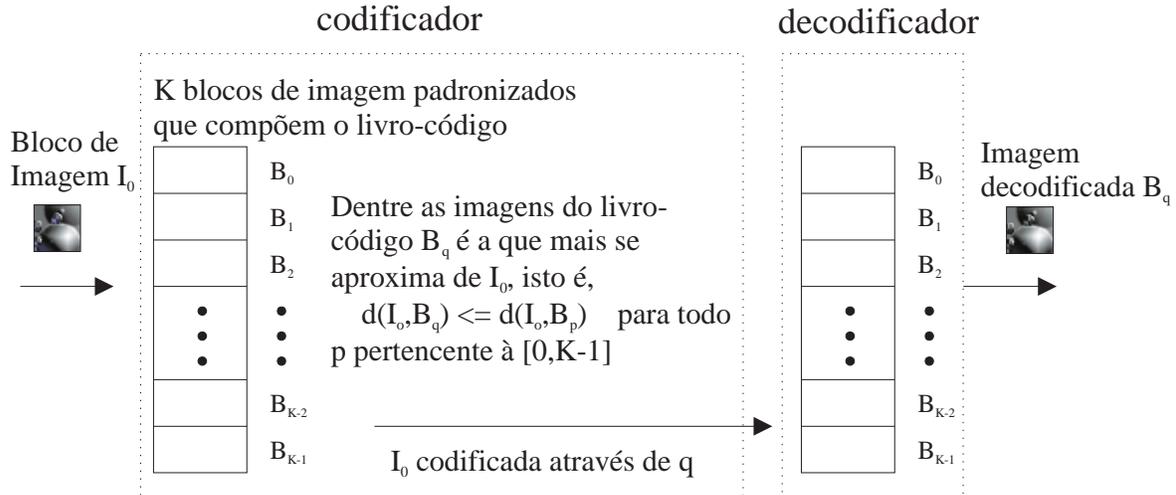


Figura 7.1: Codificação Vetorial através de livros-códigos

seções apresentam algumas abordagens utilizadas para aumentar a eficiência da quantização vetorial no domínio do espaço.

7.3.2 Quantização Adaptativa

Na quantização não adaptativa a escolha dos representantes do livro-código tem de ser feita levando-se em consideração todos os tipos de imagens. O universo considerado é muito amplo. Para que se obtenha um determinado valor de distorção máxima é preciso haver representantes suficientemente próximos de cada bloco de imagem deste universo considerado.

Uma alternativa a esta abordagem é a quantização adaptativa ([67]). Através dela o próprio grupo de blocos de imagens a serem codificados é utilizado na determinação do livro-código. Evidentemente este livro-código será mais adequado à codificação sendo realizada que outro obtido para o universo inteiro de imagens. Entretanto, a codificação da(s) imagem(ns) a que os blocos pertencem requer a caracterização do livro-código.

Existe um compromisso entre a adequação do livro-código a situações locais de imagens e a necessidade de informação adicional na codificação. Quanto mais frequentemente um novo livro-código for calculado, mais específico às características locais ele será. Assim, menor será o número de representantes para a mesma distorção entre as imagens e seus representantes e, portanto, menor será o número de bits necessários para codificar cada bloco de imagem. A quantidade de informação adicional, por outro lado, aumentará.

Uma contribuição importante nesta área foi dada por Gersho e Yano ([64]). Eles desenvolveram um método adaptativo no qual os livros-códigos não são alterados completamente a cada grupo de blocos de imagem. Apenas uma parte do livro-código correspondente às características locais não contempladas no livro anterior é alterada. Este método reduz a quantidade de informação adicional necessária para os livros-códigos utilizando-se da redundância existente entre livros-códigos vizinhos. Esta técnica, que é uma aplicação do *reabastecimento*

condicional (*conditional replenishment*) na quantização vetorial, pode ser utilizada para reduzir redundâncias entre livros-códigos de blocos vizinhos não apenas no espaço, mas também no tempo (imagens dinâmicas).

7.3.3 Classificação de Imagens antes da Quantização

Esta técnica consiste em classificar os blocos de imagens de acordo com suas características antes da quantização ([149]). A quantização é feita de acordo com a classe do bloco. Através dela, classes de blocos de imagens que exijam maiores detalhamentos do ponto de vista de percepção visual podem ser quantizadas através de uma densidade maior de representantes. Desta forma, a utilização desta técnica ameniza o fato da medida de distorção perceptual não estar sendo utilizada para determinar os representantes do livro-código.

A motivação original desta técnica foi melhorar a quantização vetorial em regiões próximas a arestas ([149]). Diferentes números de classes têm sido propostas na literatura, desde apenas duas classes (regiões com e sem arestas) até uma classe para cada orientação.

A decomposição da imagem em classes e posterior codificação pode ser estendida para os mais diversos tipos de codificação. Cada classe pode ser codificada através do método mais adequado a ela. Um exemplo de decomposição em classes com hibridismo é o trabalho [148]. Nele os blocos que pertencem a classe “sem arestas” são codificados através de métodos de transformadas, enquanto que os demais blocos são codificados através de quantização vetorial.

Os blocos de imagens a serem classificados não precisam ser todos do mesmo tamanho. Uma técnica chamada “Quantização Vetorial Adaptativa Hierárquica”, introduzida por Nasrabadi ([123]), utiliza blocos de tamanho variável. Inicialmente a imagem a ser codificada é decomposta em segmentos retangulares de comportamento aproximadamente constante (nenhuma das dimensões do retângulo pode ser menor que 2). Estes segmentos são caracterizados através de uma estrutura de dados “quad-tree”, conforme mostra a figura 7.2. Cada segmento é dividido no menor número possível de blocos quadrados de largura 2, 4, 8 ou 16. Os quadrados grandes (larguras 8 e 16) pertencem a regiões com comportamento estável, para as quais a codificação por transformadas apresenta bons resultados. Portanto os quadrados grandes são codificados através de métodos de transformadas. Os quadrados pequenos (larguras 2 e 4), pertencendo a regiões de comportamento não estável, são classificados e posteriormente codificados através de livros-códigos específicos.

É importante notar a importância dos algoritmos de classificação e segmentação das imagens na implementação destas técnicas.

7.3.4 Máquina de Estado de Quantização Vetorial

Esta técnica foi desenvolvida para aproveitar a redundância estatística das imagens para obter taxas ainda maiores de compressão através da quantização vetorial. A idéia fundamental consiste em utilizar a informação dos blocos anteriores para codificar novos blocos ([46], [60], [8] e [11]).

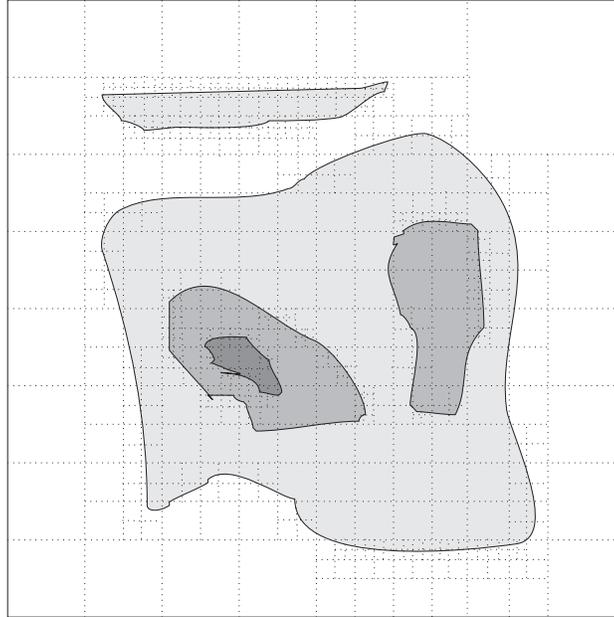


Figura 7.2: Decomposição de uma imagem em segmentos com comportamento aproximadamente constantes

7.3.5 Codificação por “Truncagem de Blocos” (BTC)

A quantização realizada neste método ([43] e [96]) não utiliza livros-códigos. Ela se baseia nos seguintes princípios:

- Um quantizador de dois níveis é utilizado para quantizar todos os pontos de um bloco
- O quantizador é projetado de forma que o representante de cada bloco (bloco quantizado) preserve as medidas estatísticas de 1^a e 2^a ordem do bloco
- A regra de quantização de cada ponto é

$$Q(x_i) = \begin{cases} a & \text{se } x_i < x_{\text{limiar}} \\ b & \text{se } x_i \geq x_{\text{limiar}} \end{cases} \quad (7.7)$$

onde $x_{\text{limiar}} = \bar{x}$

Os valores de a e b são resultantes das regras acima e podem ser determinados em função da média e da variância da luminosidade no bloco, \bar{x} e σ respectivamente, do número de elementos do bloco que ultrapassam a média q e do número total de elementos no bloco m :

$$\begin{aligned} a &= \bar{x} - \sigma \sqrt{\left(\frac{q}{m - q}\right)} \\ b &= \bar{x} + \sigma \sqrt{\left(\frac{m - q}{q}\right)} \end{aligned}$$

A quantização de um bloco é descrita através das dimensões do bloco (normalmente fixas em 4×4 e consideradas implicitamente conhecidas), de suas medidas estatísticas \bar{x} e σ e de uma matriz de bits de dimensão igual à dimensão do bloco quantizado. Um ponto da matriz de bits é 1 caso a intensidade luminosa do ponto correspondente no bloco seja maior que a intensidade de limiar x_{limiar} . Caso contrário o bit vale 0.

Esta quantização é bem mais simples que a vetorial baseada em livros-códigos justamente porque prescinde da fase inicial de determinação dos representantes dos blocos.

Uma característica importante desta quantização é que ela tem bom desempenho para blocos com arestas. Isto porque as arestas são caracterizadas por saltos de luminosidade nas direções a elas perpendiculares. Como os blocos utilizados são pequenos, as médias de luminosidade no bloco normalmente situam-se nos intervalos dos saltos de luminosidade. Desta forma, os pontos de um lado de uma aresta têm intensidade luminosa maior que a média \bar{x} e, portanto, são aproximados por b . No outro lado da aresta os pontos têm intensidade menor que \bar{x} e são aproximados por a . Logo as arestas tendem a ser adequadamente representadas por esta quantização.

Esta quantização, por outro lado, não explora as redundâncias estatísticas que existem entre pontos vizinhos de imagens. Além disto, 1 bit por ponto de imagem (fora a informação adicional de medidas do bloco) é muita informação para ser codificada por um algoritmo de compressão. Para reduzir a quantidade de informação necessária para a codificação pode-se realizar uma quantização vetorial da matriz de bits da quantização [5].

7.3.6 Quantização dos Coeficientes Transformados

O método de codificação por transformadas tem como objetivo representar as imagens em um novo sistema de coordenadas no qual os coeficientes sejam descorrelacionados estatisticamente. Este objetivo é satisfatoriamente atingido, tal como está apresentado na seção 11.3.1. Uma vez realizada a transformação de coordenadas é preciso que seja feita a quantização dos coeficientes transformados. As transformações são realizadas em blocos de imagens, normalmente quadrados 8×8 ou 16×16 . Os coeficientes transformados também formam uma matriz 8×8 ou 16×16 respectivamente.

A seção 11.2.2 apresenta detalhes sobre o processo de transformação de coordenadas. Para efeito de quantização as informações importantes sobre os coeficientes transformados são as seguintes:

- Os coeficientes são suficientemente descorrelacionados.
- O coeficiente de índice $(0, 0)$, $c_{0,0}$ corresponde ao nível médio de intensidade luminosa. Ele é chamado de coeficiente DC do bloco. Os demais coeficientes são conhecidos como coeficientes AC.
- Os coeficientes DC são uma versão sub-amostrada das imagens, devido ao fato de representarem a média da intensidade luminosa em suas regiões (filtragem passa-baixas)
- Existe forte correlação entre os coeficientes DC de regiões vizinhas. Isto pode ser entendido uma vez que os níveis médios de luminosidade não costumam mudar abruptamente entre regiões vizinhas.

- O coeficiente DC possui média positiva, já que representa valores médios de uma grandeza sempre positiva (intensidade luminosa)
- Os coeficientes AC possuem médias aproximadamente 0. Suas distribuições são aproximadamente simétricas em torno de 0, aonde formam um pico. O trabalho [17] propôs o modelo de distribuições Laplacianas para os coeficientes AC. No entanto, não há concordância da literatura quanto ao modelo a ser adotado devido aos diferentes resultados obtidos em estimações destas distribuições (vide livro [28]).
- Os coeficientes correspondem a uma decomposição espectral dos blocos de imagens. Mais especificamente, o coeficiente de índice i, j , denotado por $c_{i,j}$, corresponde aproximadamente à componente de frequência horizontal i e vertical j do espectro do bloco de imagem. Este significado físico é importante por causa da variação de sensibilidade ao ruído da visão em função da frequência espacial (resposta espectral da visão). Portanto, do ponto de vista perceptual, os coeficientes de altas frequências podem ser quantizados de forma mais grosseira (menos níveis de quantização e, conseqüentemente, menor número de bits para codificação).
- A “energia” dos coeficientes transformados concentra-se nos primeiros coeficientes. Esta característica ocorre porque as transformadas utilizadas para descorrelacionamento são tais que maximizam a compactação de energia dos componentes. Como os coeficientes AC têm médias aproximadamente nulas, as variâncias dos coeficientes são também concentradas.

O tipo de quantização mais utilizado para os coeficientes transformados é a quantização escalar de cada coeficiente. Isto por causa do descorrelacionamento entre os coeficientes. Uma outra abordagem para a quantização, proposta em [161], divide os coeficientes em faixas, agrupa faixas de blocos vizinhos e realiza quantização vetorial para estas faixas. A próxima seção trata da quantização escalar dos coeficientes, enquanto que a seção posterior discorre sobre a quantização vetorial inter-bloco.

7.3.7 Quantização Escalar dos Coeficientes

Para que a quantização escalar possa ser feita para cada coeficiente é preciso que se determine a distribuição dos níveis de quantização para os coeficientes, bem como a natureza dos quantizadores. O conhecimento estatístico que se tem acerca dos coeficientes, entretanto, não é suficiente para que se determine os níveis e os quantizadores otimizados.

Uma vez que os quantizadores ótimos não podem ser determinados, usa-se geralmente quantizadores uniformes para os coeficientes. Os coeficientes DC, que formam uma subamostragem da imagem original, costumam ser representados com a mesma precisão utilizada para a intensidade luminosa da imagem original, isto é, quantização uniforme dividida em 256 níveis (8 bits para a codificação).

Apesar da impossibilidade de precisão na determinação dos níveis de quantização dos coeficientes AC, sabe-se que os coeficientes de índices mais altos devem ser quantizados de forma mais grosseira que os de índice menores. Isto porque, além daqueles concentrarem

uma menor parcela da variância dos coeficientes (devido à compactação de energia), eles correspondem a faixas de frequência observadas com menor sensibilidade pela visão.

Existem duas abordagens iniciais para a distribuição de níveis de quantização entre os coeficientes: codificação zonal e codificação por limiar.

Na codificação zonal os coeficientes espalhados nas N^2 posições do bloco transformado são divididos em zonas. A cada zona corresponde um número de níveis de quantização. Esta é, na verdade, a abordagem clássica da quantização escalar em múltiplas dimensões que, conforme visto acima, não se adequa bem aos coeficientes transformados visto que suas distribuições não são adequadamente modeladas. Uma melhoria importante para esta abordagem consiste na classificação dos blocos de imagens em grupos com características similares e utilização de uma divisão zonal para cada grupo. Para blocos contendo muitas componentes de altas frequências horizontais por exemplo, a decomposição zonal privilegia as zonas correspondentes a estas altas frequências. A figura 7.3 ilustra algumas distribuições de níveis de quantização por zonas.

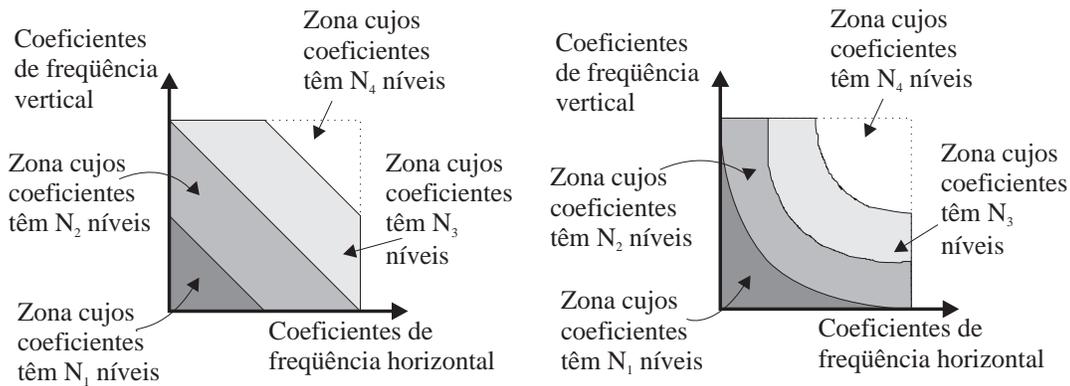


Figura 7.3: Distribuições típicas de níveis de quantização em zonas de coeficientes transformados de blocos de imagens

Na codificação por limiar não há pré-alocação de níveis para cada coeficiente transformado. Ao invés disto, o número de níveis para cada coeficiente depende de seu valor. O nome da abordagem vem do fato de existir um limiar abaixo do qual os coeficientes são considerados nulos. Esta abordagem adequa-se mais às características de cada bloco. Entretanto ela requer que as posições e números de bits dos coeficientes que ultrapassam o limiar façam parte da codificação.

As duas abordagens podem ser utilizadas em conjunto. Pode-se pré-alocar níveis de quantização para coeficientes de baixas frequências, enquanto que os coeficientes de altas frequências são quantizados através da abordagem de limiar.

Uma diferença prática entre as abordagens é que a zonal garante de antemão o número de bits utilizados para codificar cada bloco. Na codificação por limiar, por outro lado, o número de bits para a quantização de cada bloco depende dos valores dos coeficientes. Esta variação na quantidade de informação necessária para codificação de cada bloco requer a utilização de técnicas de controle de fluxo (bufferização) para a comunicação de imagens através de canais

com taxas de transmissões constantes.

7.3.8 Quantização Vetorial Inter-Bloco

Nos métodos de codificação por transformadas, as transformadas são aplicadas a blocos de imagens. Elas são bastante eficientes na tarefa de descorrelacionar os coeficientes que descrevem os blocos. As correlações das imagens, porém, freqüentemente ultrapassam os limites dos blocos e pode haver correlação entre coeficientes de blocos vizinhos.

Uma possibilidade adicional de redução de quantidade de informação na codificação de imagens consiste na utilização de quantização vetorial inter-bloco ([33] e [168]). Coeficientes de mesmo índice de blocos vizinhos são agrupados formando os vetores a serem quantizados. Para que os livros-códigos não aumentem exageradamente a quantidade de informação adicional na codificação, os coeficientes podem ser divididos em zonas e um livro-código utilizado para cada zona ([125]).

Capítulo 8

Atribuição de Códigos a Fontes Discretas

8.1 Introdução

Seja uma fonte $\{\mathcal{F}_n\}_n$ com alfabeto finito $\mathcal{A} = \{a_0, \dots, a_{n-1}\}$ contendo n elementos. A forma mais simples de atribuir códigos binários a estes n elementos consiste na atribuição de códigos com o mesmo número de bits para todos os elementos do alfabeto. Para realizar isto é preciso calcular qual o menor número de bits m necessário para representar de forma única todos os n elementos de \mathcal{A} . O cálculo é simples: $m = \lceil \log_2 n \rceil$, onde $\lceil \cdot \rceil$ significa a função cujo resultado é o menor inteiro não menor que o argumento. Em seguida escolhe-se um inteiro entre 0 e $2^m - 1$ para cada elemento a_i . Este inteiro c_i é o código do elemento a_i . É claro que a escolha tem de ser feita de forma que $c_i \neq c_j$ se $i \neq j$. Caso contrário a codificação seria ambígua, pois representaria dois elementos a_i, a_j através do mesmo código. Denomina-se este tipo de codificação por codificação com códigos ou palavras com número uniforme de bits.

Caso a fonte $\{\mathcal{F}_n\}_n$ gere P elementos a cada unidade de tempo, a taxa de bits necessária para codificar a fonte seria mP .

Esta taxa pode ser maior que a taxa necessária para codificar a fonte. De fato o “Teorema da Codificação sem Perdas”, descrito no Capítulo 2, estabelece uma taxa mínima intrínseca da fonte, independente da codificação. Esta taxa mínima intrínseca é a taxa de entropia $\overline{H}[\mathcal{F}_n]$. Mais especificamente o teorema estabelece que dada a fonte $\{\mathcal{F}_n\}_n$ e um real $\epsilon > 0$, existe uma codificação para a fonte tal que a taxa de bits média necessária para representação seja menor que $\overline{H}[\mathcal{F}_n] + \epsilon$. Por outro lado, o teorema também estabelece que não é possível encontrar uma codificação que redunde na representação da fonte por uma taxa menor que $\overline{H}[\mathcal{F}_n]$.

Conforme mencionado anteriormente, a idéia básica das codificações cujas taxas de representação aproximam a entropia das respectivas fontes é a atribuição de códigos com pequenas quantidades de bits para elementos com freqüências de ocorrências altas. Consequentemente os elementos menos prováveis serão representados por códigos com maiores quantidades de bits.

Foi exatamente seguindo esta idéia que surgiram as codificações com códigos ou palavras

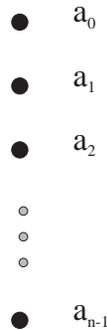
com número variável de bits. O primeiro destes códigos a ficar famoso foi o de Huffman, publicado em 1952 ([77]).

8.2 Códigos de Huffman

A codificação de Huffman foi projetada inicialmente para fontes estacionárias no sentido estrito. Nestes casos a probabilidade de ocorrência de um determinado elemento a_i de um alfabeto \mathcal{A} ($P[a_i] = p_i$) não varia com o tempo.

O algoritmo que associa um código c_i a um elemento a_i funciona da seguinte maneira:

Os elementos $\{a_0, \dots, a_{n-1}\}$ do alfabeto \mathcal{A} são ordenados de forma decrescente em relação a suas probabilidades de ocorrência, tal como ilustra a figura 8.1.



com $p(a_0) \geq p(a_1) \geq \dots \geq p(a_{n-1})$

Figura 8.1: Elementos do alfabeto ordenados de acordo com suas probabilidades de ocorrência

A atribuição de bits aos códigos dos elementos começa pelos menos prováveis $a_{m_{n-1}}$ e $a_{m_{n-2}}$. De forma arbitrária escolhe-se um símbolo (0 ou 1) para ser o menos significativo do código de $a_{m_{n-1}}$, código este denotado por $c_{m_{n-1}}$. O outro símbolo (1 ou 0, respectivamente), será o menos significativo do código de $a_{m_{n-2}}$ ($c_{m_{n-2}}$). Nenhum símbolo é atribuído aos demais elementos ainda. Em seguida estes dois elementos menos prováveis são agrupados em um “super-elemento” $S_0 = (\{a_{m_{n-1}}, a_{m_{n-2}}\}, p_{m_{n-1}} + p_{m_{n-2}})$.

Os “super-elementos” são na verdade pares contendo um sub-conjunto do alfabeto \mathcal{A} e uma probabilidade correspondente a soma das probabilidades dos elementos constituintes do sub-conjunto.

Os próximos passos do algoritmo consistem em reordenar os $n \Leftarrow 1$ elementos $\{a_{m_0}, a_{m_1}, \dots, S_0\}$ de acordo com suas probabilidades e atribuir os símbolos 0 e 1 aos códigos dos dois elementos menos prováveis. Caso S_0 seja um destes elementos menos prováveis ambos os códigos $c_{m_{n-1}}$ e $c_{m_{n-2}}$ recebem o símbolo atribuído a S_0 .

Os processos de ordenação de elementos, atribuição de símbolos aos códigos dos menos prováveis e agrupamentos destes elementos menos prováveis formam uma etapa. Ao final de cada etapa o número de elementos do conjunto considerado é reduzido de 1 unidade.

Portanto, ao final de $n \Leftrightarrow 1$ etapas todos os elementos de \mathcal{A} possuirão um código.

Uma vez que a idéia do algoritmo está apresentada, convém escrevê-lo de forma menos intuitiva, porém mais precisa. O propósito desta descrição menos intuitiva é evitar a mistura de tipos de elementos nos conjuntos de elementos utilizados nas etapas do algoritmo. A idéia básica desta nova apresentação consiste em considerar que a i -ésima etapa age sobre um conjunto de elementos $\{P^{(i)}[k]\}_{k=0}^{n-(i+1)}$ cada um contendo um subconjunto de elementos de \mathcal{A} e uma probabilidade (soma das probabilidades dos elementos do sub-conjunto). Os elementos iniciais $\{P^{(0)}[k]\}_{k=0}^{n-1}$ confundem-se com os próprios elementos de \mathcal{A} , pois $P^{(0)}[k] = (\{a_k\}, p_k)$. A descrição completa está apresentada nos parágrafos abaixo.

Seja $P^i[k]$ um elemento contendo uma probabilidade e um subconjunto de \mathcal{A} (alfabeto da fonte). A probabilidade e o subconjunto de \mathcal{A} do elemento $P^i[k]$ são denotados respectivamente por $P^i[k].p$ e $P^i[k].a$.

O algoritmo usa $n(n \Leftrightarrow 1)/2$ destes elementos. Na verdade o algoritmo é composto de $n \Leftrightarrow 1$ fases. Para cada fase i existem $(n \Leftrightarrow i)$ elementos $P^i[0]$, $P^i[1]$, ..., $P^i[(n \Leftrightarrow i) \Leftrightarrow 1]$. O índice k é referente à ordenação decrescente dos elementos $P^i[k]$ em relação as probabilidades $P^i[k].p$ a eles associadas. Em outras palavras, $P^i[0].p \geq P^i[1].p \geq \dots \geq P^i[(n \Leftrightarrow i) \Leftrightarrow 1].p$.

Na fase inicial, $i = 0$, os subconjuntos $P^0[k].a$ são unitários. Os elementos de \mathcal{A} são ordenados de forma decrescente de probabilidade de ocorrência. Seja $(a_{m_0}, a_{m_1}, \dots, a_{m_{n-1}})$ esta ordenação, isto é, $P[a_{m_0}] \geq P[a_{m_1}] \geq \dots \geq P[a_{m_{n-1}}]$. Os elementos $P^0[k]$ são tais que:

$$P^0[k].a = \{a_{m_k}\}$$

e

$$P^0[k].p = P[a_{m_k}]$$

No início do algoritmo todos os elementos de \mathcal{A} possuem um código nulo, isto é, não têm código algum. Toda vez que um elemento $a_i \in \mathcal{A}$ “recebe” um novo bit (0 ou 1), este bit é posicionado à esquerda do código antigo. Desta forma um elemento cujo código é 001 fica com os códigos 0001 e 1001 ao “receber” respectivamente os bits 0 e 1.

Em cada fase i , os 2 elementos com menores probabilidades $P^i[(n \Leftrightarrow i) \Leftrightarrow 2]$ e $P^i[(n \Leftrightarrow i) \Leftrightarrow 1]$ são agregados. Todos os elementos do conjunto $P^i[(n \Leftrightarrow i) \Leftrightarrow 2].a$ recebem um bit (0 ou 1), enquanto que todos os elementos do conjunto $P^i[(n \Leftrightarrow i) \Leftrightarrow 1].a$ recebem outro bit (1 ou 0). A designação de 0 ou 1 para cada um dos conjuntos é arbitrária (desde que os dois conjuntos recebam bits distintos). Na verdade esta escolha altera os códigos resultantes dos elementos a_i , mas não altera o número de bits de cada código.

Em seguida é criado um elemento temporário $P^i[n \Leftrightarrow i]$ cujos campos valem:

$$P^i[n \Leftrightarrow i].a = P^i[(n \Leftrightarrow i) \Leftrightarrow 1].a \cup P^i[(n \Leftrightarrow i) \Leftrightarrow 2].a$$

e

$$P^i[n \Leftrightarrow i].p = P^i[(n \Leftrightarrow i) \Leftrightarrow 1].p + P^i[(n \Leftrightarrow i) \Leftrightarrow 2].p$$

A última etapa da fase i consiste em ordenar os $(n \Leftrightarrow i) \Leftrightarrow 1$ elementos $P^i[0]$, $P^i[1]$, ..., $P^i[(n \Leftrightarrow i) \Leftrightarrow 3]$, $P^i[n \Leftrightarrow i]$ de acordo com suas probabilidades. O resultado desta ordenação é a nova lista de $(n \Leftrightarrow i) \Leftrightarrow 1$ elementos $P^{i+1}[0]$, $P^{i+1}[1]$, ..., $P^{i+1}[(n \Leftrightarrow (i+1)) \Leftrightarrow 1]$.

Estas etapas são realizadas sucessivamente até que reste apenas 1 elemento $P^{n-1}[0]$. Neste instante todos os elementos de \mathcal{A} terão um código.

Convém apresentar um exemplo para sedimentar as idéias. Seja o alfabeto \mathcal{A} com 6 elementos: $\{a_0, a_1, \dots, a_5\}$, cujas probabilidades de ocorrência são respectivamente $p_0 = 1/2$, $p_1 = 1/8$, $p_2 = 1/32$, $p_3 = 3/32$, $p_4 = 1/16$, $p_5 = 3/16$. Na fase inicial ($i = 0$) teremos os seguintes elementos:

$$\begin{array}{ll} P^0[0].p = 1/2 & P^0[0].a = \{a_0\} \\ P^0[1].p = 3/16 & P^0[1].a = \{a_5\} \\ P^0[2].p = 1/8 & P^0[2].a = \{a_1\} \\ P^0[3].p = 3/32 & P^0[3].a = \{a_3\} \\ P^0[4].p = 1/16 & P^0[4].a = \{a_4\} \\ P^0[5].p = 1/32 & P^0[5].a = \{a_2\} \end{array}$$

Os elementos $P^0[4]$ e $P^0[5]$ são agregados. Antes porém os elementos de $P^0[4].a$ recebem o bit 0 e os elementos de $P^0[5].a$ recebem o bit 1. O novo elemento $P^0[6]$ terá os seguintes campos:

$$P^0[6].p = 3/32 \quad P^0[6].a = \{a_2, a_4\}$$

Após a fase 0 os códigos dos elementos de \mathcal{A} são:

$$c_0 = \emptyset \quad c_1 = \emptyset \quad c_2 = 0 \quad c_3 = \emptyset \quad c_4 = 1 \quad c_5 = \emptyset$$

e os novos elementos $P^1[k]$ serão:

$$\begin{array}{ll} P^1[0].p = 1/2 & P^1[0].a = \{a_0\} \\ P^1[1].p = 3/16 & P^1[1].a = \{a_5\} \\ P^1[2].p = 1/8 & P^1[2].a = \{a_1\} \\ P^1[3].p = 3/32 & P^1[3].a = \{a_3\} \\ P^1[4].p = 3/32 & P^1[4].a = \{a_2, a_4\} \end{array}$$

Ao final da fase 1 existem os seguintes códigos e elementos:

$$c_0 = \emptyset \quad c_1 = \emptyset \quad c_2 = 00 \quad c_3 = 1 \quad c_4 = 01 \quad c_5 = \emptyset$$

$$\begin{array}{ll} P^2[0].p = 1/2 & P^2[0].a = \{a_0\} \\ P^2[1].p = 3/16 & P^2[1].a = \{a_5\} \\ P^2[2].p = 3/16 & P^2[2].a = \{a_2, a_3, a_4\} \\ P^2[3].p = 1/8 & P^2[3].a = \{a_1\} \end{array}$$

Ao final da fase 2 se tem os seguintes códigos e elementos:

$$c_0 = \emptyset \quad c_1 = 0 \quad c_2 = 100 \quad c_3 = 11 \quad c_4 = 101 \quad c_5 = \emptyset$$

$$\begin{array}{ll} P^3[0].p = 1/2 & P^3[0].a = \{a_0\} \\ P^3[1].p = 5/16 & P^3[1].a = \{a_1, a_2, a_3, a_4\} \\ P^3[2].p = 3/16 & P^3[2].a = \{a_5\} \end{array}$$

Ao final da fase 3 se tem os seguintes códigos e elementos:

$$c_0 = \emptyset \quad c_1 = 10 \quad c_2 = 1100 \quad c_3 = 111 \quad c_4 = 1101 \quad c_5 = 0$$

$$\begin{aligned} P^4[0].p &= 1/2 & P^4[0].a &= \{a_0\} \\ P^4[1].p &= 1/2 & P^4[1].a &= \{a_1, a_2, a_3, a_4, a_5\} \end{aligned}$$

Ao final da fase 4 o algoritmo está encerrado, resultando nos códigos

$$c_0 = 1 \quad c_1 = 010 \quad c_2 = 01100 \quad c_3 = 0111 \quad c_4 = 01101 \quad c_5 = 00$$

e no elemento

$$P^5[0].p = 1 \quad P^5[0].a = \{a_0, a_1, a_2, a_3, a_4, a_5\} = \mathcal{A}$$

É interessante observar que o algoritmo sempre termina com os códigos atribuídos aos elementos do alfabeto \mathcal{A} e com o elemento $P^{n-1}[0]$ tal que:

$$P^{n-1}[0].p = 1 \quad P^{n-1}[0].a = \mathcal{A}$$

No caso do exemplo a taxa de entropia da fonte pode ser calculada diretamente da entropia (caso de fontes estacionárias). A entropia vale:

$$\begin{aligned} H[\mathcal{F}_n] &= \Leftrightarrow \sum_{i=0}^5 p_i \log_2(p_i) \\ &= 2.054 \end{aligned} \tag{8.1}$$

Por outro lado, a taxa de bits média necessária para codificar a fonte usando o código de Huffman é:

$$\begin{aligned} \text{Taxa média} &= \Leftrightarrow \sum_{i=0}^5 p_i \text{comp}(c_i) \\ &= 67/32 \\ &= 2.094 \end{aligned} \tag{8.2}$$

onde “comp” é a função que associa um código ao seu comprimento em número de bits.

Portanto pode-se observar que a codificação de Huffman realmente reduziu a taxa média de representação da fonte de 3 bits por símbolo para 2.094 bits por símbolo (muito próximo da entropia da fonte).

Os códigos de Huffman, tal como são construídos são perfeitamente decodificáveis. Seja uma seqüência qualquer $\{b_n\}_n$ de elementos de \mathcal{A} e $c(b_n)$ a seqüência binária formada pela concatenação dos códigos dos elementos de $\{b_n\}_n$: $c(b_0) | c(b_1) | \dots$ ($|$ simboliza a operação de concatenação). Esta seqüência será decodificada de forma única por um decodificador de Huffman.

Isto ocorre porque os códigos de Huffman obedecem a condição de prefixação (são “prefix condition codes”). Esta condição estabelece que nenhum código $c_i = c(a_i)$, $a_i \in \mathcal{A}$ é prefixo de algum outro código $c_j = c(a_j)$, $a_j \neq a_i$, $a_j \in \mathcal{A}$.

Além desta característica de decodificação os códigos de Huffman são importantes porque são ótimos. A otimização é no sentido de que para uma dada fonte $\{\mathcal{F}_n\}_n$ estacionária com alfabeto finito \mathcal{A} não há outro conjunto de palavras código $\{c'_0, c'_1, \dots, c'_{n-1}\}$ que permita um tamanho médio de palavra-código menor que o atingido pelo código de Huffman. Isto significa que nestas condições os códigos de Huffman minimizam a taxa média de representação de uma fonte dentre todos os códigos decodificáveis que associam uma palavra a cada elemento do alfabeto da fonte.

A codificação de Huffman tal como foi descrita tem suas limitações. De fato a taxa de representação da fonte apresentada no exemplo foi maior que a entropia da fonte. Portanto, segundo o Teorema da Codificação sem Perdas, existe código mais eficiente que o Huffman apresentado.

Para que uma codificação se aproxime do valor ótimo dentre todas as codificações sem perdas, isto é, sua taxa de representação da fonte se aproxime da entropia, é necessário que os comprimentos dos códigos $c(a_i)$ sejam próximos de $\lceil \log_2 p_i \rceil$. Como os comprimentos são necessariamente números inteiros e os logaritmos das probabilidades não, pode haver uma diferença razoável entre $\lceil \log_2 p_i \rceil$ e $\log_2 p_i$. Na verdade a codificação de Huffman produz resultados tão mais próximos da entropia da fonte quanto as probabilidades p_i , $i \in [1, n]$ aproximem-se de potências negativas de 2.

Uma forma de reduzir os efeitos desta limitação é atribuir códigos a cadeias de elementos do alfabeto da fonte. A saída de uma fonte pode ser compreendida como uma seqüência de cadeias de elementos do alfabeto original. O primeiro passo para esta abordagem consiste na determinação do alfabeto de cadeias a serem codificadas. Pode-se, por exemplo, considerar as cadeias de comprimento menor ou igual a 2. No caso do exemplo apresentado o novo alfabeto teria 42 elementos. As cadeias de comprimento 1 são utilizadas nas situações em que uma seqüência de número ímpar de elementos precisar ser codificada.

$$\begin{aligned} &\{(a_0), (a_1), \dots, (a_5), \\ &(a_0, a_0), (a_0, a_1), \dots, (a_0, a_5), \\ &(a_1, a_0), (a_1, a_1), \dots, (a_1, a_5), \dots \\ &(a_5, a_0), (a_5, a_1), \dots, (a_5, a_5)\} \end{aligned}$$

Outros alfabetos de seqüências de elementos do alfabeto original da fonte podem ser utilizados. Pode-se usar cadeias de comprimento variado de elementos.

Outra limitação da codificação de Huffman apresentada consiste na utilização de um modelo pré-fixado para a fonte $\overline{H[\mathcal{F}_n]}$. Na maioria das aplicações práticas a determinação das probabilidades de elementos do alfabeto de uma fonte é uma tarefa muito difícil ou até mesmo impraticável. Além disto, muitas fontes encontradas nas aplicações não são estacionárias.

A codificação de Huffman pode ser adaptativa, isto é, pode usar modelos de fontes que se adaptam à medida em que os elementos do alfabeto vão sendo codificados/decodificados. A adaptabilidade reduz os efeitos desta segunda limitação, mas traz consigo um incremento na complexidade dos algoritmos de codificação/decodificação e uma maior vulnerabilidade a erros que possam ocorrer entre os processos de codificação e decodificação (ruído numa linha de comunicação, por exemplo).

Ainda que possua suas limitações a codificação de Huffman tem várias aplicações, como por exemplo a codificação de imagens em dois tons utilizada para comunicação de FAX.

Além das aplicações práticas, esta codificação tem grande importância teórica, dado ao seu caráter ótimo dentro de uma classe de codificações. Maiores informações sobre codificação de Huffman podem ser encontradas em [77] e [63].

Um outro método de codificação sugerido inicialmente nos trabalhos de Shannon tomou corpo ao longo de décadas e hoje apresenta-se como bastante eficiente: Codificação Aritmética. Na verdade pode-se dizer que a codificação aritmética generaliza a de Huffman, sendo particularmente mais adequada para lidar com as limitações apresentadas nos parágrafos anteriores.

8.3 Codificação Aritmética

A codificação aritmética baseia-se na idéia de representar uma cadeia de elementos através de um intervalo $I = [a, b) \subset [0, 1)$ ([92] e [191]). A determinação do intervalo de representação de uma cadeia de elementos é feita através da realização sucessiva de um processo de decomposição de intervalos.

Considere-se novamente uma fonte $\{\mathcal{F}_n\}_n$ com alfabeto finito $\mathcal{A} = \{a_0, \dots, a_{n-1}\}$ contendo n elementos. Seguindo a notação já utilizada, a probabilidade de ocorrência do elemento a_i é dada por $P(a_i) = p_i$ (a fonte é estacionária no sentido estrito). A primeira decomposição de intervalo acontece com $[0, 1)$. Este intervalo é dividido em n sub-intervalos:

$$[0, 1) = \cup_{i=0}^{n-1} [P_i, P_{i+1})$$

onde $P_i = \sum_{j < i} p_j$ para $i \in [0, n \Leftrightarrow 1]$ e $P_n = 1$, conforme mostra a figura 8.2.

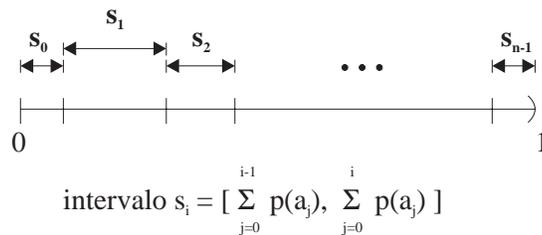


Figura 8.2: Decomposição do intervalo $[0, 1)$ em n sub-intervalos

Associa-se a cada elemento a_i do alfabeto o sub-intervalo $[P_i, P_{i+1})$. Este sub-intervalo tem comprimento igual a probabilidade de ocorrência do elemento a_i .

A codificação de uma cadeia de elementos é realizada com o auxílio de um intervalo de trabalho. Este intervalo é iniciado como $[0, 1)$. Para cada elemento novo considerado na cadeia obtém-se um novo intervalo de trabalho a partir do antigo. Caso o intervalo de trabalho seja $[a, b)$ em um dado instante e o novo elemento a se considerar na cadeia seja a_i , o novo intervalo de trabalho será $[a + (b \Leftrightarrow a) * P_i, a + (b \Leftrightarrow a) * P_{i+1})$. Vale dizer que este novo intervalo está para o intervalo $[a, b)$ assim como o intervalo $[P_i, P_{i+1})$ está para o intervalo $[0, 1)$.

Este procedimento de decomposição de intervalos é realizado recursivamente até que todos os elementos da cadeia a ser codificada sejam considerados. O intervalo de trabalho resultante representa de forma única a seqüência de elementos da cadeia.

Um exemplo pode ilustrar bem as idéias. Considere-se novamente a fonte da seção anterior com seus 6 elementos $\{a_0, a_1, a_2, a_3, a_4, a_5\}$. A codificação da cadeia (a_1, a_3, a_0, a_5) é feita seguindo os seguintes passos:

- Passo 1: • Intervalo de trabalho = $[0, 1)$
 • Elemento considerado = a_1
 • Intervalo associado = $[P_1, P_2) = [1/2, 11/16)$
 • Novo intervalo de trabalho = $[P_1, P_2) = [1/2, 11/16) = [0.5, 0.6875)$
- Passo 2: • Intervalo de trabalho = $[P_1, P_2) = [1/2, 11/16)$
 • Elemento considerado = a_3
 • Intervalo associado = $[P_3, P_4) = [13/16, 29/32) = [0.8125, 0.90625)$
 • Novo intervalo de trabalho = $[0.65234375, 0.669921875)$
- Passo 3: • Intervalo de trabalho = $[0.65234375, 0.669921875)$
 • Elemento considerado = a_0
 • Intervalo associado = $[P_0, P_1) = [0, 1/2) = [0.0, 0.5)$
 • Novo intervalo de trabalho = $[0.65234375, 0.6611328125)$
- Passo 4: • Intervalo de trabalho = $[0.65234375, 0.6611328125)$
 • Elemento considerado = a_5
 • Intervalo associado = $[P_5, P_6) = [31/32, 1) = [0.96875, 1.0)$
 • Novo intervalo de trabalho = $[0.6319580078125, 0.6611328125)$

Portanto a seqüência (a_1, a_3, a_0, a_5) é representada pelo intervalo $[0.6319580078125, 0.6611328125)$. Este intervalo é escrito como $[0.10000111001, 0.1010011011)$ na representação binária, ou como $[1081/2048, 667/1024)$ na forma fracionária. A figura 8.3 ilustra o processo de obtenção do intervalo.

Os trabalhos [136], [116], [118] e [117] apresentam detalhes sobre outro exemplo de implementação de codificação aritmética.

Uma característica importante do método de determinação dos intervalos é a não interseção entre intervalos distintos representando cadeias de mesmo número de elementos. De fato, sejam duas cadeias distintas $(b_i)_i$ e $(d_i)_i$ ambas com m elementos. Como as cadeias são distintas existe um primeiro elemento diferente, isto é, $\exists k \in [0, m \Leftrightarrow 1] \mid b_k \neq d_k$ e $b_j = d_j \forall j \in [0, k \Leftrightarrow 1]$. Por construção os intervalos de trabalho das duas cadeias são iguais até que sejam considerados os elementos b_k e d_k . Seja $[a, b)$ o intervalo comum de trabalho após os $k \Leftrightarrow 1$ primeiros passos. Em seguida a decomposição de intervalos das cadeias $(b_i)_i$ e $(d_i)_i$ redundarão respectivamente nos intervalos $[a + (b \Leftrightarrow a) * P_{ac}(b_k), a + (b \Leftrightarrow a) * (P_{ac}(b_k) + p_{b_k}))$ e $[a + (b \Leftrightarrow a) * P_{ac}(d_k), a + (b \Leftrightarrow a) * (P_{ac}(d_k) + p_{d_k}))$. $P_{ac}(x)$ é a probabilidade acumulada do

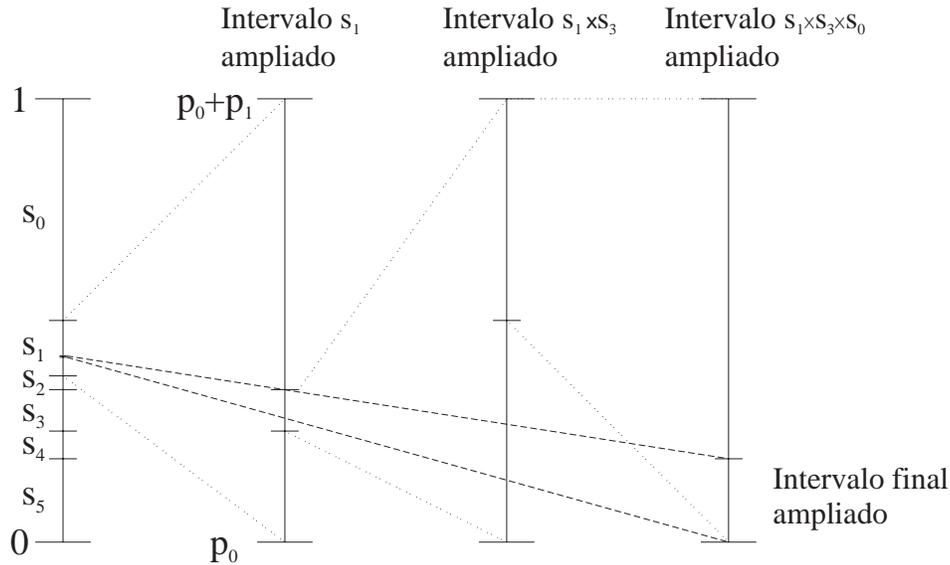


Figura 8.3: Representação da Cadeia (a_1, a_3, a_0, a_5) em codificação aritmética

elemento x . Como $x \in \mathcal{A}$, $\exists i \in [0, n \Leftrightarrow 1] \mid x = a_i$ ($\mathcal{A} = \{a_0, \dots, a_{n-1}\}$). $P_{ac}(x) = \sum_{j < i} p(a_j)$. Logo, como b_k e d_k são disjuntos e todos os elementos de \mathcal{A} têm probabilidade não nula $P_{ac}(b_k) + p_{b_k} \leq P_{ac}(d_k)$ ou $P_{ac}(d_k) + p_{d_k} \leq P_{ac}(b_k) \Rightarrow$ os intervalos de trabalho após a fase k são distintos. Como os intervalos de trabalho das fases posteriores estão contidos nos respectivos intervalos após a fase k , os intervalos finais de representação das cadeias têm interseção nula.

Esta propriedade assegura que a representação de um intervalo relativo a uma cadeia com número conhecido de elementos não precisa ser feita através de suas fronteiras. Na verdade ela pode ser feita através de qualquer ponto do intervalo. A representação dos intervalos por pontos pode ser feita também para cadeias com um número arbitrário de elementos. Basta que um elemento do alfabeto seja escolhido como terminador de cadeias. Neste caso, o codificador insere o terminador ao final da cadeia que deseja codificar. A determinação da cadeia de elementos pode ser feita por qualquer ponto do intervalo resultante após a codificação do terminador. O decodificador recebe o ponto codificado e obtém a seqüência de elementos da cadeia até que seja encontrado um terminador. O processo de decodificação de cadeias a partir de pontos no intervalo $[0, 1)$ está ilustrado na figura 8.4.

Voltando ao exemplo apresentado, à medida em que a cadeia de elementos cresce o intervalo de trabalho fica cada vez menor e (mais importante) requer maior quantidade de bits para sua representação. Desta forma, a representação cresce com o número de elementos representados.

O crescimento da representação dos intervalos não é igual para todos os elementos de \mathcal{A} . Quanto menor for a probabilidade de um elemento maior será a redução de tamanho por ele provocada no intervalo de trabalho (o comprimento do intervalo de trabalho é mul-

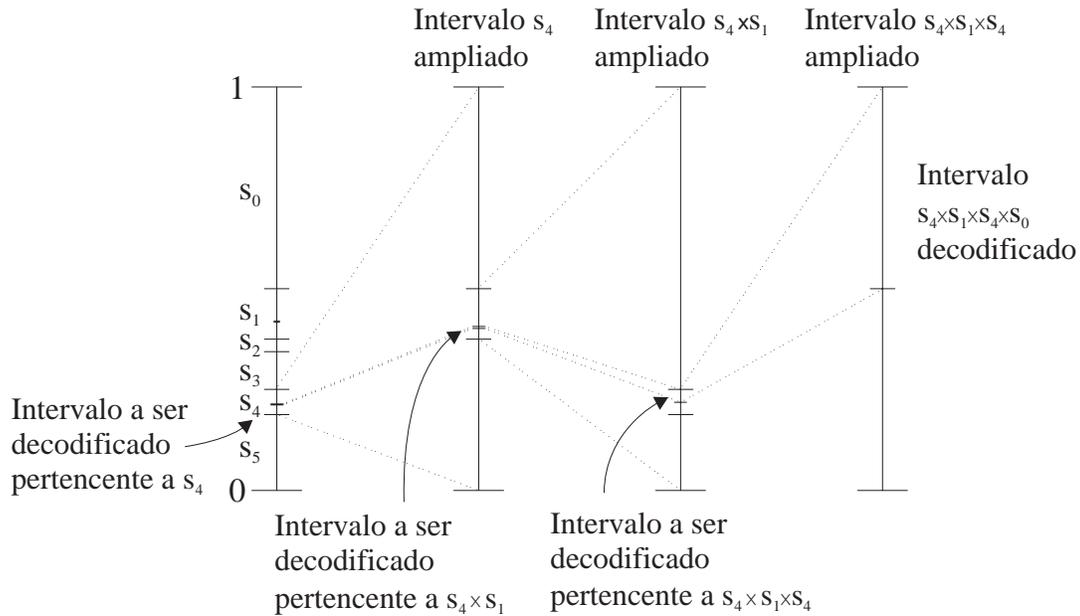


Figura 8.4: Decodificação de uma cadeia de elementos codificada através de codificação aritmética

tipicado pela probabilidade do elemento). A presença de um elemento com probabilidade alta, por outro lado, implica numa alteração menor no comprimento do intervalo de trabalho. Analisando de forma grosseira, quanto menor for a alteração no comprimento do intervalo de trabalho, menor será a necessidade de utilização de mais informação para caracterizar o novo intervalo. Desta forma a codificação aritmética utiliza implicitamente uma atribuição de palavras-código com número variado de bits para os elementos de um alfabeto.

8.3.1 Base Teórica da Codificação Aritmética

A noção intuitiva de que a codificação aritmética utiliza quantidades diferentes de bits para codificar elementos com diferentes probabilidades e, portanto, pode aproximar a entropia de uma fonte é insuficiente para justificar sua utilização. É preciso estabelecer uma relação mais precisa entre o comprimento dos intervalos de codificação e o número de bits necessários para representar as respectivas cadeias de elementos. Tal relação foi determinada por Shannon no precursor artigo [164]: Considere-se cadeias contendo N elementos. A cada cadeia s associa-se um intervalo l_s de comprimento p_s . Seja b_s a expansão binária do limite inferior de l_s . A figura 8.5 ilustra esta decomposição do intervalo $[0, 1)$ representando cadeias de N elementos.

Os intervalos são representados por uma aproximação de b_s com precisão finita B_s (considerando-se apenas uma quantidade finita de bits de b_s). Deseja-se obter um limite superior para a quantidade de bits necessária para os B_s 's tal que cada intervalo l_s ainda possa ser identificado unicamente. Este limite será obtido através de uma argumentação em três etapas:

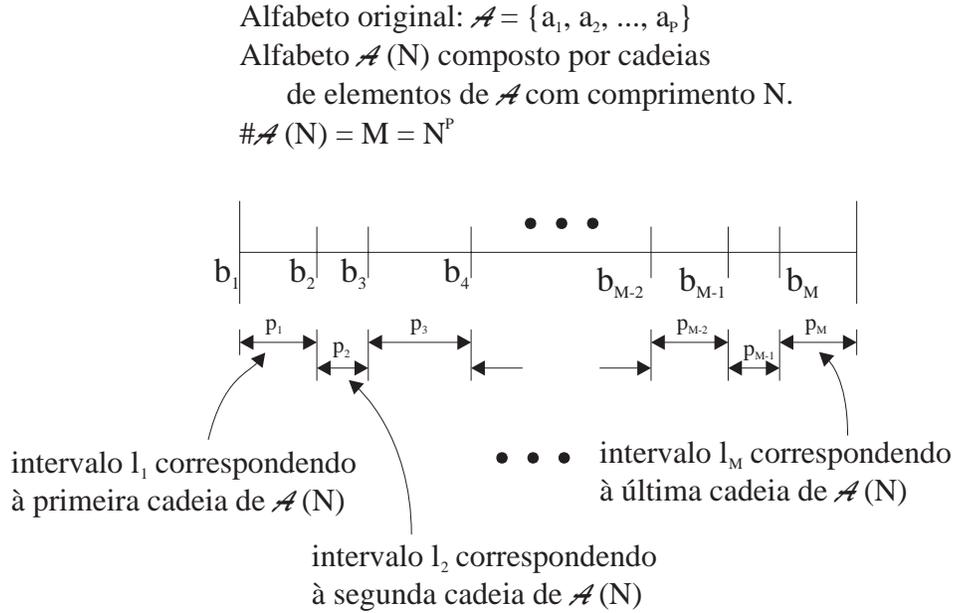


Figura 8.5: Decomposição do intervalo $[0, 1)$ utilizada para representar cadeias de N elementos através da codificação aritmética

1. Apresentação de um critério de escolha de precisão de B_s
2. Demonstração que os códigos gerados a partir deste critério obedecem à condição de prefixação, isto é, são “*prefix condition codes*”. Em outras palavras, nenhum código B_{s_i} é prefixo de outro B_{s_j} para $i \neq j$. Isto garante que cada intervalo, e portanto cada cadeia de elementos N , é unicamente determinado por B_s .
3. Determinação de qual o número máximo de bits necessários para B_s em função do comprimento de l_s e, posteriormente, de um limite superior para a média de bits necessária para codificar a fonte de cadeias de N elementos.

• Passo 1:

Utilize-se para B_s a precisão p_s tal que existam pelo menos 2 pontos do conjunto $\{\frac{m}{2^{p_s}}\}$, $m = 0, \dots, (p_s \Leftrightarrow 1)$ em l_s . B_s será o menor dos pontos $\{\frac{m}{2^{p_s}}\} \cap l_s$. A figura 8.6 ilustra esta escolha de B_s .

• Passo 2:

Por contradição seja B_{s_i} um código que coincida integralmente com os p_{s_i} primeiros bits de B_{s_j} . Desta forma a diferença entre B_{s_j} e B_{s_i} será menor que $2^{p_{s_i}}$ e, portanto, $B_{s_j} \in l_{s_i}$ pois por construção $[B_{s_i}, B_{s_i} + 2^{p_{s_i}}) \subset l_{s_i}$. Como este resultado contradiz a construção dos códigos B_s ($B_s \in l_s$ e $l_{s_i} \cap l_{s_j} = \emptyset$ se $i \neq j$) a hipótese inicial não pode ser verdadeira $\Rightarrow B_{s_i}$ não pode ser um prefixo de B_{s_j} para $i \neq j$.

Três exemplos de intervalos para os quais a precisão p_s de suas representações vale $(k+1)$

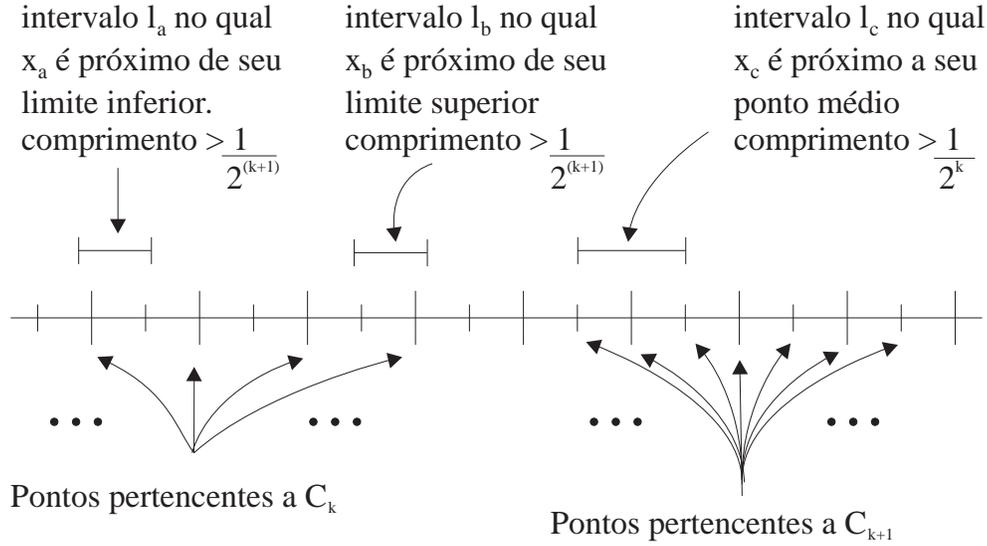


Figura 8.6: Escolha da precisão de B_s em três casos diferentes de l_s

- Passo 3:

Seja a família de conjuntos

$$C_k = \left\{ \frac{m}{2^k} \right\}, m = 0, \dots, (2^k \Leftrightarrow 1) \quad (8.3)$$

Seja k_s o menor inteiro tal que $C_{k_s} \cap l_s \neq \emptyset$. $C_{k_s} \cap l_s$ tem que ser um conjunto unitário, pois caso contrário pelo menos um de seus pontos teria de pertencer a $C_{k_s-1} \Rightarrow C_{k_s-1} \cap l_s \neq \emptyset$, o que contraria a definição de k_s . Seja, portanto, o ponto $x_s = C_{k_s} \cap l_s$. De acordo com a construção dos códigos definida no passo 1, a precisão de B_s (p_s) será o menor inteiro tal que cardinalidade $[C_{p_s} \cap l_s] > 1$. Isto requer que $x_s \Leftrightarrow \frac{1}{2^{p_s}} \geq b_s$ ou $x_s + \frac{1}{2^{p_s}} < b_{s+1}$, ou seja, $\frac{1}{2^{p_s}} < \max(b_{s+1} \Leftrightarrow x_s, x_s \Leftrightarrow b_s)$, o que no pior caso significa $\frac{1}{2^{p_s}} < (\text{comprimento}(l_s)/2)$ (vide figura 8.6). Entretanto, mesmo para este pior caso vale

$$\frac{1}{2^{p_s}} > \frac{\text{comprimento}(l_s)}{4} \quad (8.4)$$

pois caso contrário $\frac{1}{2^{p_s-1}} < (\text{comprimento}(l_s)/2)$ e, portanto cardinalidade $[C_{p_s-1} \cap l_s] > 1$, o que contradiz a definição de p_s . Logo, aplicando-se logaritmo na base 2 na equação 8.4 obtem-se um limite superior para a precisão p_s do código B_s de cada cadeia de elementos s a partir do comprimento de l_s :

$$p_s < 2 \Leftrightarrow \log_2(\text{comprimento}(l_s)) \quad (8.5)$$

Uma vez determinada a relação 8.5 não é difícil calcular um limite superior para a taxa média de bits necessária para codificar a fonte de cadeias de N elementos. Basta para isto lembrar que por construção dos intervalos l_s , comprimento (l_s) é a probabilidade de geração da cadeia s pela fonte. A taxa média de bits necessária para codificar a fonte H_1 é dada pela equação 8.6:

$$H_1 = \sum_{s \in \mathcal{A}} p_s \text{ comprimento } (l_s) \quad (8.6)$$

onde \mathcal{A} é o alfabeto da fonte (conjunto de cadeia de N elementos). Juntando esta equação com a relação 8.5 obtem-se uma limitação para H_1 :

$$H_1 < \sum_{s \in \mathcal{A}} [2 \Leftrightarrow \log_2(\text{comprimento } (l_s))] \text{ comprimento } (l_s) \quad (8.7)$$

$$\begin{aligned} &< [\Leftrightarrow \sum_{s \in \mathcal{A}} \text{comprimento } (l_s) \log_2(\text{comprimento } (l_s)) \\ &\quad + 2 \sum_{s \in \mathcal{A}} \text{comprimento } (l_s)] \quad (8.8) \end{aligned}$$

$$< H_f(N) + 2 \quad (8.9)$$

onde $H_f(N)$ é a entropia da fonte de cadeias de N elementos.

A taxa média de bits por elemento codificado fica, portanto, limitada à

$$\frac{H_1}{N} < \frac{H_f(N)}{N} + \frac{2}{N} \quad (8.10)$$

Como $\frac{H_f(N)}{N}$ tende para a entropia da fonte de elementos à medida em que N cresce, o número médio de bits por elemento necessários para codificar uma fonte através da codificação aritmética pode aproximar tanto quanto desejado a entropia da fonte.

8.4 Codificação Universal e Métodos de Lempel-Ziv

8.4.1 Introdução

Na maioria das aplicações de codificações de Huffman e aritméticas o processo de codificação é alimentado com ocorrências de elementos isolados. No primeiro caso a cada elemento gera-se um novo código, enquanto que no segundo a cada elemento realiza-se uma divisão do intervalo de representação da cadeia de elementos. Neste tipo de aplicação os métodos de atribuição de códigos não aproveitam o histórico das fontes, isto é, as dependências estatísticas que possam existir entre seus elementos.

Este tipo de dependência pode ser verificado nos textos da língua portuguesa. Considere-se a da letra “u”, por exemplo. A probabilidade de ocorrência desta letra dado que a anterior foi “q” é muito maior que sua probabilidade de ocorrência dado que qualquer outra letra tenha a precedido.

Nada impede, em tese, que as codificações de Huffman e aritmética aproveitem estas dependências. De fato, os elementos do alfabeto de uma fonte podem ser agrupados em cadeias e posteriormente a estas cadeias atribuídos códigos. Existem, porém, limitações de ordem prática que dificultam bastante o aproveitamento do histórico das fontes por estes tipos de codificações.

Quando cadeias de elementos de fontes e não os elementos propriamente ditos são utilizados para alimentar os codificadores a cardinalidade do alfabeto a ser considerado cresce exponencialmente com o número de elementos das cadeias.

O primeiro aspecto prejudicial deste aumento exponencial de cardinalidade é que, mesmo para alfabetos pequenos e cadeias de pequeno comprimento, o número de elementos a serem codificados torna-se virtualmente intratável. Considere-se, por exemplo, o alfabeto mais utilizado: o código ASCII contendo 256 elementos. Neste caso, para cadeias de 4 elementos, o número de cadeias a serem codificadas cresce a $256^4 = 2^{32} \sim 4000000000$.

Este aspecto do desconhecimento dos modelos poderia ser contornado através da utilização de versões adaptativas dos métodos de codificação. O próprio uso de adaptabilidade, entretanto, fica prejudicado, uma vez que a coleta de dados necessária para alimentar um modelo adaptativo de elevado número de elementos (2^{32} no exemplo acima) é também muito grande.

Estas limitações impedem a utilização de históricos de fontes pelas codificações de Huffman e aritmética. Apesar disto, existem outras alternativas de codificação que são capazes de utilizar o histórico de fontes para melhorar a codificação. A área da teoria da informação que estuda este problema foi proposta por Komolgorov e é denominada Codificação Universal ([87], [57], [58], [41], [156]).

Um código é dito universal quando atende a dois requisitos:

- A codificação é realizada em blocos;
- O desempenho é assintoticamente ótimo em algum sentido.

Segundo esta formulação os codificadores universais prestam-se muito bem para codificar fontes com características estatísticas desconhecidas de antemão e também mutáveis com o tempo. Isto porque a codificação realiza-se em blocos. Toda a informação relativa à caracterização estocástica da fonte tem que ser extraída do bloco que está sendo codificado, e não do conhecimento a priori ou do passado inteiro da fonte.

Os trabalhos publicados por Jacob Ziv e Abraham Lempel no final da década de 70 revolucionaram a codificação universal. Hoje em dia quase todos os compressores de dados de uso genérico (compactadores de arquivos, compactadores de dados para armazenamento em fita/disco, compressores de dados para comunicação via modem, por exemplo) baseiam-se em um dos métodos propostos por Ziv e Lempel.

Embora estes métodos não sejam muito utilizados nos algoritmos atualmente mais promissores de compressão de imagens, vale a pena conhecê-los para que se aumente o rol de ferramenta úteis para novos algoritmos de codificação.

8.4.2 Métodos de Ziv e Lempel

Os métodos de Ziv e Lempel revolucionaram a codificação universal principalmente pela eficiência na utilização dos dados históricos das fontes de informação. Além disto, a simplicidade dos métodos fez com que suas implementações passassem a ser corriqueiras nas manipulações de dados do dia-a-dia (exemplos: PKZIP, STACKER, compressão de dados para fitas e modems, “superstore” e “doublepace”).

O primeiro método, chamado LZ 1 ou LZ77, foi publicado em 1977 ([205]), enquanto o segundo, denominado LZ 2 ou LZ78 foi publicado em 1978 ([206]). Ambos baseiam-se na idéia de armazenar o passado recente de uma fonte, e codificar cadeias de elementos através de identificadores da ocorrência anterior na cadeia nesse passado recente armazenado. As duas próximas seções apresentam os algoritmos LZ 1 e LZ 2 de forma sucinta. Maiores detalhes podem ser encontrados nos próprios artigos de Lempel e Ziv ([205] e [206]) e em trabalhos posteriores mais voltados para implementações práticas ([128], [187]).

8.4.3 LZ 1

O método LZ 1 armazena o passado recente numa janela de texto. Esta janela é composta dos elementos já gerados pela fonte e já codificados. Estes elementos são ordenados de acordo com sua ordem de geração pela fonte. Os elementos novos são armazenados numa área de trabalho antes de serem codificados. Tanto a janela de texto quanto a área de trabalho têm tamanhos máximos denotados, respectivamente, por t e s . A figura 8.7 ilustra o processo de codificação.



Figura 8.7: Processo de codificação LZ1

A janela de texto pode ser entendida como um dicionário no qual as cadeias novas procurarão antigas ocorrências no histórico da fonte.

A área de trabalho vai sendo preenchida enquanto a cadeia de elementos que ela contém for encontrada em alguma posição da janela de texto, e enquanto o tamanho da cadeia for menor que s .

Tão logo uma destas condições seja violada, a cadeia da área de trabalho é codificada, seus elementos passam a fazer parte da janela de texto e a área de trabalho se renova para receber novos elementos. A codificação de uma cadeia é feita por intermédio de uma tripla contendo os seguintes campos:

- Identificação do posicionamento da cadeia de elementos da janela de texto que coincide com a cadeia na área de trabalho
- Número de elementos coincidentes nas cadeias da janela de texto e da área de trabalho

- Último elemento inserido na área de trabalho

Os parágrafos a seguir descrevem a codificação com mais detalhes.

Inicialmente um elemento a gerado pela fonte é inserido na área de trabalho. Caso o elemento exista na janela de texto o método prossegue. Caso contrário o elemento é codificado isoladamente, e é transferido da área de trabalho para a janela de texto (passa a fazer parte do histórico da fonte). Esta codificação isolada é feita através de uma tripla: $(0,0,a)$.

Seja uma cadeia S de comprimento $0 < l(S) < s$ (l é a função que atribui a uma cadeia de elementos o seu comprimento) que se encontra na área de trabalho. Pela definição do método esta cadeia existe na janela de texto. Os próximos passos do processo de codificação dependem do novo elemento gerado a pela fonte e do comprimento da cadeia $l(S)$:

- Caso a seja tal que $S | a$, isto é, a cadeia formada pela concatenação de S e a , ainda esteja na janela de texto E $l(S | a) < s$ então o elemento a é incorporado pela cadeia S e o processo continua.
- Caso $S | a$ ainda esteja na janela de texto E $l(S | a) = s$ então o elemento a é incorporado pela cadeia S e esta é codificada através da tripla (endereço da cadeia igual a S na janela de texto, $l(S)$, a). A janela de texto é deslocada de $l(S)$ elementos para que a cadeia S seja a ela concatenada. A área de trabalho é esvaziada para que possa receber novos elementos
- Caso $S | a$ não esteja mais janela de texto o novo elemento a é incorporado à S , que é codificada através da tripla (endereço da cadeia igual a S na janela de texto, $l(S) \leftrightarrow 1$, a). A janela de texto é deslocada de $l(S)$ elementos para que a cadeia S seja a ela concatenada. A área de trabalho fica novamente vazia após a codificação de S .

Desta forma percebe-se que a codificação é realizada de duas maneiras distintas: para cadeias encontradas na janela de texto e para elementos isolados. Para que possa haver a decodificação é necessário existir um campo a mais nos códigos informando o tipo de código (código de cadeia ou de elemento isolado).

Os comprimentos das janelas de texto e da área de trabalho são dois parâmetros do projeto do codificador LZ1. Quanto maior for t maior será o passado considerado da fonte. Portanto maior será a chance de casamento de cadeias novas com o passado da fonte. Quanto maior for s , por sua vez, menor será a chance de se codificar uma cadeia por causa do enchimento da área de trabalho ($l(S) = s$, ou seja, sub-utilização do codificador).

Estes parâmetros, por outro lado, não podem ser aumentados indiscriminadamente. Primeiramente porque a complexidade computacional da busca de um encontro da área de trabalho na janela de texto cresce linearmente com s e, na melhor hipótese, logaritmicamente com t .

Além disto, os números de bits utilizados para representar o endereço da cadeia igual à codificada na janela de texto e o comprimento da cadeia dependem respectivamente de $\lceil \log_2 t \rceil$ e $\lceil \log_2 s \rceil$. Caso s e t cresçam de forma desnecessária a própria codificação binária das duplas correspondentes às cadeias codificadas crescerá desnecessariamente.

8.4.4 LZ 2

Tal como o LZ1 o LZ2 procura casar cadeias novas (armazenadas na área de trabalho) com ocorrências anteriores no histórico da fonte. Também de forma similar as cadeias são codificadas através de um identificador de ocorrência anterior no histórico.

A diferença entre os dois métodos reside na forma com que o histórico da fonte é armazenado. Ao invés de uma janela de texto armazena-se um dicionário de frases ocorridas. Cada entrada do dicionário contém a seqüência de elementos da frase e seu comprimento.

Um detalhe de implementação também contribui para diferenciar os dois métodos. O dicionário de frases é preenchido inicialmente com frases de comprimento 1 correspondendo a todos os elementos do alfabeto. Assim não há necessidade de codificação especial para elementos isolados.

A área de trabalho vai sendo preenchida enquanto a cadeia de elementos que ela contém for encontrada em alguma entrada do dicionário. No momento em que um novo elemento a adicionado à área de trabalho fizer com que a cadeia nela armazenada não seja mais encontrada no dicionário, esta cadeia é codificada através de um par:

- Índice da frase do dicionário coincidente com a cadeia anterior à adição do último elemento a
- elemento a

8.4.5 Aplicações dos Métodos LZ em Codificação de Imagens

Tal como foi dito anteriormente os métodos de Ziv e Lempel não são muito utilizados nos métodos de compressão de imagens mais promissores hoje em dia.

Isto porque seu maior ganho consiste na utilização eficiente do histórico de fontes desconhecidas na codificação. O caso de imagens é bastante particular porque o comportamento das fontes não é completamente desconhecido e, portanto, pode-se usar métodos projetados especificamente para estes casos. Tais métodos mais específicos podem aproveitar melhor o histórico das fontes. De fato o trabalho de aproveitamento do histórico das fontes é geralmente realizado conjuntamente nas etapas de transformação de representação e quantização. Desta forma, em geral não cabe à etapa de atribuição de códigos aos símbolos de saída da quantização a tarefa de maximizar a codificação através do uso de informações históricas das fontes.

De qualquer forma estes algoritmos de Ziv e Lempel podem ser de grande utilidade para novos métodos a serem desenvolvidos.

Capítulo 9

Transformações de Representação

9.1 Introdução

Este capítulo inicia uma seqüência de capítulos sobre a fase de transformação de representação. Inicialmente são apresentados os diferentes tipos de métodos de transformação de representação de imagens. Em seguida existe uma seção sobre as bases comuns à maioria dos métodos de transformação de representação, divididas nas áreas: 1) propriedades estatísticas de imagens e 2) características da visão humana.

9.2 Classificação dos Métodos de Codificação

A classificação dos métodos de codificação de imagens proposta baseia-se no princípio utilizado para compressão de informação. A divisão proposta é a seguinte:

- Codificação Preditiva
- Codificação por Transformadas
- Codificação por Decomposição de Imagens em Sub-bandas
- Codificação por Fractais
- Codificação Baseada em Informações das Bordas Existentes nas Imagens (ou através de métodos de segunda geração)
- Codificação por Modelagem

Este trabalho destina um capítulo para cada classe de codificação citada acima. Estes capítulos estão organizados nas seguintes seções:

- Descrição do Método
- Base Teórica da Codificação
 - Princípios utilizados para redução de redundância das fontes

- Características da visão humana exploradas
- Possibilidade de emprego de adaptabilidade
- Possibilidade de emprego de métodos híbridos
- Possibilidade de emprego de codificação sem perdas
- Complexidade de implementação
- Faixa de compressão atingida

9.3 Base Comum para as Transformações de Representação

Esta seção apresenta os aspectos comuns à grande maioria dos métodos de transformações de representação. Desta forma evita-se repetir estes aspectos nas exposições das diferentes classes de métodos.

Na primeira subseção desta seção as características estatísticas das imagens globalmente utilizadas são apresentadas. A seção seguinte concentra-se nas características da visão humana. Estas seções não se aprofundam na descrição das características apresentadas, uma vez que elas estão apresentadas em maiores detalhes nos capítulos 4 e 5 deste trabalho.

9.3.1 Características Estatísticas de Imagens

Em termos de estatísticas de imagens, existem três características utilizadas na maioria dos métodos de transformações de representação:

- Fontes de imagens com comportamento globalmente não estacionário, mas aproximadamente estacionário em âmbito local. Esta característica justifica a modelagem local das imagens.
- Decomposição das coordenadas de cor em luminância e duas coordenadas de crominância é muito próxima da decomposição em coordenadas principais (através da transformada de Karhunen-Loève). Isto significa que as coordenadas de luminância e crominância não são muito correlacionadas e apresentam boa compactação de “energia” (variância) na coordenada de luminância.
- As coordenadas de cor das imagens apresentam aproximadamente as mesmas características estatísticas espaciais e temporais.

As duas últimas características estatísticas das imagens levam à conclusão (já descrita no capítulo 4) que os métodos de codificação de imagens coloridas se resumem essencialmente a três instâncias dos métodos de codificação de imagens monocromáticas, uma instância para cada coordenada de cor.

9.3.2 Características da Visão Humana

As primeiras características da visão humana globalmente utilizadas são aquelas que servem como base para as aproximações que levam às imagens digitais:

- Percepção passa-baixas no tempo \Rightarrow amostragem temporal das imagens
- Percepção passa-baixas no espaço \Rightarrow amostragem espacial das imagens
- Caracterização de Cores em R^3

Estas características, somadas à quantização no espaço de cores, compõem os elementos que permitem considerar as imagens como matrizes de vetores em Z^3 . Os elementos das matrizes de imagens correspondem às cores (em Z^3) em pontos das imagens, chamados de elementos de imagens ou *pixels*. Também segundo estas aproximações as imagens dinâmicas, isto é, cenas observadas durante um intervalo de tempo, são representadas por uma seqüência de matrizes de imagens. Cada matriz corresponde a uma imagem amostrada no tempo.

Estas aproximações, já descritas no capítulo 2, são utilizadas na grande maioria, senão em todos os métodos de codificação de imagens utilizados hoje em dia. Na verdade, na maioria das aplicações as imagens são consideradas de início como sendo seqüências de matrizes de cores em pixels.

Outra característica bastante usada é a diferenciação de percepção das informações de cores. Uma cor é caracterizada através de três coordenadas de cor. Uma primeira questão a ser resolvida é saber quais as coordenadas que devem ser usadas para especificar as cores.

Existem alguns estudos na literatura que objetivam obter os melhores eixos coordenados sob o ponto de vista de correlação estatística ([101]). Em outras palavras, estes estudos buscam eixos no espaço de cores de forma a minimizar a correlação entre as coordenadas de cor. A seção 4.5.1 resume estes estudos. A conclusão é que as cores devem ser representadas por uma componente de luminosidade (a grosso modo as imagens em preto-e-branco) e duas componentes puramente relacionadas com informação de cor. A estas componentes dá-se o nome de luminância e crominância, respectivamente.

Uma característica muito importante da percepção humana de cores é que o nível de detalhamento espacial necessário para a informação de luminância é maior do que o necessário para as informações de crominância. Isto significa que a amostragem espacial das coordenadas de crominância pode ser mais grosseira que a da coordenada de luminância. Para aplicar esta característica na codificação de imagens, estas são consideradas não mais como matrizes de elementos em Z^3 , mas sim como três matrizes de elementos inteiros. A matriz de informação de luminância tem mais elementos que as de crominância. Isto porque cada elemento corresponde a um ponto amostrado na imagem. Desta forma amostragem mais grosseira corresponde a menos pontos amostrados \Rightarrow menos elementos nas matrizes. Em geral as dimensões das matrizes de luminância são múltiplas das dimensões das matrizes de crominância.

Esta consideração de diferentes dimensões para as matrizes de luminância e crominância também é bastante utilizada. De fato ela é usada nas aplicações em que as cores e uma imagem (e não os objetos contidos numa cena) são codificadas.

Capítulo 10

Codificação Preditiva

10.1 Descrição do Método

A codificação preditiva baseia-se na modelagem preditiva das imagens segundo a qual as fontes de imagens podem ser decompostas numa componente predita e em outra componente residual com entropia menor que a fonte original. Desta forma a fonte de imagens passa a ser descrita através dos parâmetros de modelagem mais o sinal residual, isto é, a diferença entre a componente predita e o valor efetivamente produzido pela fonte. Em outras palavras, a transformação de representação realizada na codificação preditiva é aquela que transforma intensidades luminosas em parâmetros de modelagem e em sinais residuais.

Para ilustrar a idéia, seja um ponto P_m de uma imagem monocromática. Sejam seus vizinhos (tanto no tempo quanto no espaço) denotados por $P_{m-k}, P_{m-(k-1)}, \dots, P_{m-1}$. As respectivas intensidades luminosas destes pontos são: $u(P_{m-k}), u(P_{m-(k-1)}), \dots, u(P_{m-1}), u(P_m)$. Uma estimativa de $u(P_m)$ baseada em seus vizinhos é calculada:

$$\Phi[u(P_{m-1}), \dots, u(P_{m-(k-1)})]$$

A diferença entre $u(P_m)$ e $\Phi[u(P_{m-1}), \dots, u(P_{m-(k-1)})]$ é chamada de resíduo. Uma vez que a função Φ seja conhecida, os valores de $u(P_m)$ podem ser determinados recursivamente através de seus vizinhos e dos resíduos $\epsilon(m)$:

$$u(P_m) = \Phi[u(P_{m-1}), \dots, u(P_{m-(k-1)})] + \epsilon(m) \quad (10.1)$$

A codificação tende a ser eficiente à medida em que pouca informação seja necessária para caracterizar os parâmetros e o sinal residual. Isto requer inicialmente que os parâmetros de modelagem (que determinam a função Φ) não sejam muitos e também não variem muito na codificação. Além disto o sinal residual tem que ter entropia menor que o sinal original. Intuitivamente pode-se compreender porque o resíduo teria entropia menor que o sinal original: Caso a predição seja bem feita o resíduo se concentra bastante no 0, isto é, a distribuição de probabilidade do resíduo concentra-se no 0. Conforme visto no capítulo 2, quanto mais concentrada for a distribuição de probabilidade de um sinal (ou processo estocástico discreto), menor será sua entropia.

A figura 10.1 ilustra a idéia básica da codificação preditiva.

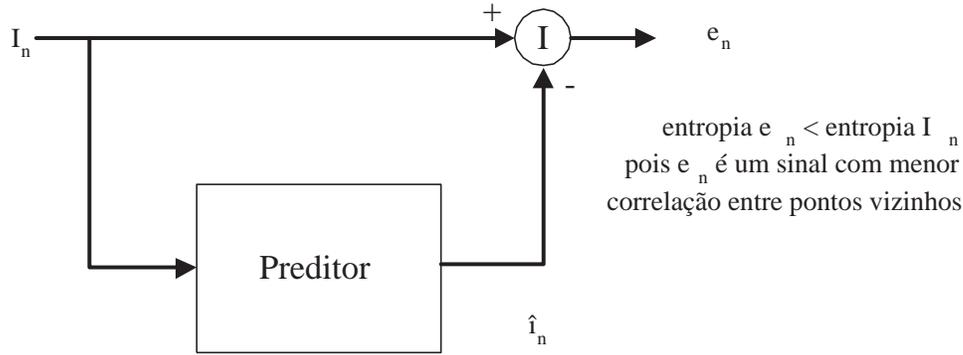


Figura 10.1: Codificação Preditiva

10.2 Base Teórica da Codificação

10.2.1 Fundamentos para a Redução de Redundância

A idéia de representar uma fonte como a soma de uma componente predita e um sinal residual de menor quantidade de informação origina-se nos modelos autoregressivos de fontes. Conforme definido no capítulo 3, uma fonte $\{u_n\}_n$ é dita autoregressiva (AR) caso possa ser escrita como

$$\begin{aligned} u(n) &= \overline{u(n)} + \epsilon(n) \\ u(n) &= \sum_{k=1}^p a(k)u(n \Leftrightarrow k) + \epsilon(n) \end{aligned} \quad (10.2)$$

sendo o sinal de excitação $\epsilon(n)$ estacionário, com média 0 e independente do estado anterior do sinal $u(n)$. Isto significa que:

$$\begin{aligned} E[\epsilon(n)] &= 0, \\ E[\epsilon(n)\epsilon(m)] &= \beta^2 \delta(n \Leftrightarrow m), \\ E[\epsilon(n)u(m)] &= 0, \\ m &< n \end{aligned} \quad (10.3)$$

Segundo esta definição apenas as fontes com média 0 podem ser AR. Apesar de não serem diretamente contemplados na definição, os sinais com média não nula podem ter um comportamento essencialmente autoregressivo. Isto acontece quando eles puderem ser escritos como a soma de um sinal AR e uma constante, isto é:

$$\begin{aligned} x(n) &= y(n) + \bar{x} \\ y(n) &= \sum_{k=1}^p a(k)y(n \Leftrightarrow k) + \epsilon(n) \end{aligned}$$

$$\begin{aligned}
 E[\epsilon(n)] &= 0, \\
 E[\epsilon(n)\epsilon(m)] &= \beta^2\delta(n \Leftrightarrow m), \\
 E[\epsilon(n)y(m)] &= 0, \\
 & m < n
 \end{aligned}
 \tag{10.4}$$

Sob o ponto de vista da codificação esta representação dos sinais AR é importante porque o sinal $u(n)$ pode passar a ser descrito pelo preditor linear mais o sinal residual que apresenta características mais adequadas para a codificação:

- o sinal residual $\epsilon(n)$ minimiza o valor médio quadrático: $E\{[\epsilon(n)]^2\}$. Sendo assim, $\{\epsilon(n)\}_n$ é um sinal bastante concentrado em 0 e, indiretamente, apresenta entropia baixa.
- $\epsilon(n)$ não apresenta correlação entre amostras (comportamento de ruído branco), o que significa que a redundância existente entre as amostras do sinal $u(n)$ foi eliminada ao subtrair-se $\overline{u(n)}$ de $u(n)$.

Para que a codificação possa ser realizada é preciso determinar uma forma de conversão entre os sinais $\{u(n)\}_n$ e $\{\epsilon(n)\}_n$. Esta forma de conversão está implícita na própria definição dos sinais AR:

$$\begin{aligned}
 u(n) &= \overline{u(n)} + \epsilon(n) \\
 \epsilon(n) &= u(n) - \overline{u(n)}
 \end{aligned}
 \tag{10.5}$$

Para terminar a determinação da conversão é preciso definir a aproximação linear $\overline{u(n)}$ de $u(n)$. Esta, por sua vez, depende da natureza do sinal AR (ordem do preditor e valores dos parâmetros usados na predição).

A operação que leva o sinal AR original no seu resíduo $\{\epsilon(n)\}_n$ é chamada de *filtragem branqueadora*, pois o sinal resultante tem características de ruído branco (independência entre amostras). A figura 10.2 ilustra a filtragem branqueadora.

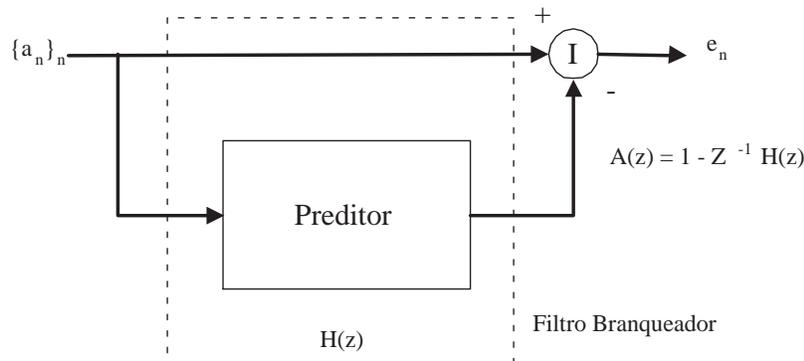


Figura 10.2: Filtragem branqueadora

Na prática a consideração de modelos AR para imagens é uma aproximação. Não existe comprovação de que os sinais provenientes das imagens sejam AR. No entanto vale a idéia de buscar preditores de forma que o sinal residual tenha entropia menor que o original. Em outras palavras, preditores tais que seus respectivos filtros branqueadores realmente eliminem (ou pelo menos reduzam) a redundância estatística entre amostras do sinal. Uma comprovação prática da eficiência na utilização de filtros preditores está apresentada no histograma normalizado do sinal residual de um preditor unidimensional de ordem 3 exemplificado no livro [157]. Este histograma está transcrito para este trabalho na figura 10.3. Pode-se perceber a forte concentração do sinal residual em torno de 0.

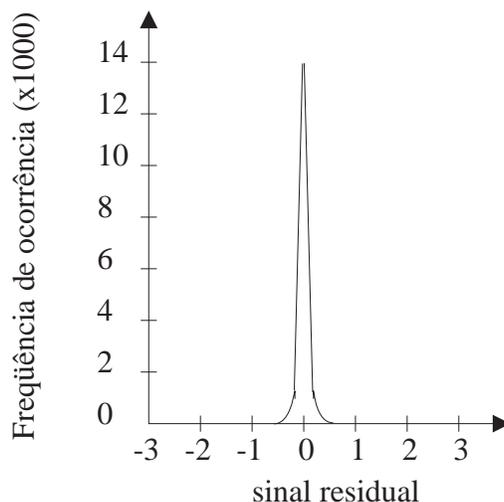


Figura 10.3: Histograma do sinal residual de um preditor 1-D de ordem 3

10.2.2 Características da Visão Humana Exploradas

Os métodos de codificação preditiva baseiam-se principalmente na redução de redundância estatística das fontes para reduzir a quantidade de informação necessária para representar imagens. As características de visão humana normalmente utilizadas são somente aquelas relacionadas à aproximação das imagens por imagens digitais (vide seção 9.3).

10.2.3 Codificação Preditiva para Imagens

Conforme visto anteriormente, a idéia da codificação preditiva consiste em determinar um preditor para as imagens. O objetivo do preditor é antever de forma mais fiel possível o comportamento de uma fonte com base nas amostras do passado da fonte. Se a predição for realmente boa o sinal residual concentra-se muito no valor 0 e, indiretamente, tem entropia baixa.

Alguns problemas práticos ocorrem quando se deseja utilizar esta abordagem no caso de imagens:

- Determinação dos sinais a serem preditos
- Determinação das vizinhanças dos elementos a serem preditos
- Determinação dos preditores e estimação de parâmetros
- Quantização do sinal residual

As próximas subseções destinam-se a analisar as abordagens utilizadas para resolver estes problemas.

10.2.4 Determinação dos Sinais a serem Preditos

O primeiro problema a ser resolvido na aplicação deste método para imagens é determinar quais os elementos a serem preditos: matrizes de imagens, linhas ou colunas de imagens, blocos de imagens ou pontos de imagens (pixels). Em outras palavras, é preciso escolher qual será o significado das variáveis aleatórias $u(n)$ que formam o sinal a ser predito. No fundo esta escolha de elementos implica na determinação de como as imagens são decompostas em sinais para efeito de predição.

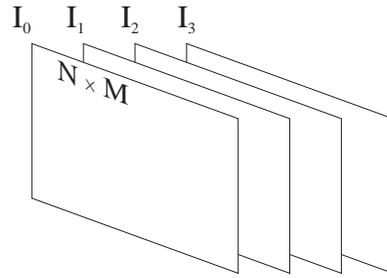
Em princípio as fontes de imagens originais (gerando matrizes de pixels) poderiam ser utilizadas para predição. No entanto verifica-se que o comportamento local das imagens apresenta uma previsibilidade bem maior que o global. Isto significa que é mais fácil prever o comportamento de um ponto (ou uma pequena região de pontos) a partir de seus vizinhos do que prever uma imagem inteira a partir de suas antecessoras. Portanto, as abordagens mais comuns são aquelas em que a predição está baseada em pixels ou pequenas regiões, ao invés de imagens inteiras. Mais especificamente, a predição baseada em pixels é a mais comumente utilizada. A predição de pequenas regiões, apesar de não tão comum, tem um exemplo importante: a predição de movimentos realizada para eliminar a redundância temporal de seqüências de imagens.

Para que esta abordagem seja possível os pixels ou as regiões de uma imagem tem de ser ordenados de forma com que a concatenação sequencial dos elementos de diversas imagens forme um sinal. Em outras palavras o sinal multi-dimensional original tem de se transformar num sinal unidimensional (no caso de predição baseada em pixels) ou bidimensional (no caso de predição de pequenas regiões). A ordenação dos pixels ou regiões não é nada mais do que uma varredura das imagens.

A figura 10.4 ilustra como a ordenação determinada por uma varredura transforma uma seqüência de imagens numa seqüência de pixels.

As varreduras mais utilizadas são as progressiva e entrelaçada, ilustradas respectivamente nas figuras 4.3 e 4.2 do capítulo 4.

A ordenação dos elementos é importante porque ela determina como as relações de vizinhança das imagens originais se refletem no sinal 1-D (ou 2-D) a ser predito. As relações de vizinhança, por sua vez, são importantes justamente porque as predições de sinais são realizadas tomando por base os pontos vizinhos. Portanto, caso os pontos vizinhos no sinal original de imagens permaneçam vizinhos no sinal 1-D obtido após a varredura, as predições do sinal 1-D também levarão em conta pontos vizinhos. Caso contrário, as predições dos

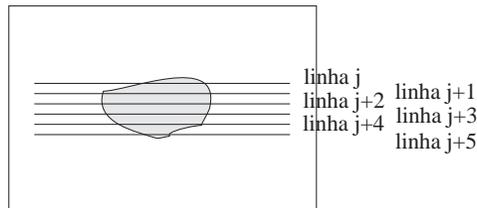


Seqüência da imagem I_0 , Seqüência da imagem I_1 , ...

$a_{011}, a_{012}, \dots, a_{01M}, a_{021}, a_{022}, \dots, a_{02M}, \dots, a_{0N1}, a_{0N2}, \dots, a_{0NM},$
 $a_{111}, a_{112}, \dots, a_{11M}, a_{121}, a_{122}, \dots, a_{12M}, \dots, a_{1N1}, a_{1N2}, \dots, a_{1NM}, \dots$

Figura 10.4: Sequenciamento herdado pela determinação de uma varredura

sinais 1-D que pretendam levar em conta toda a vizinhança (no tempo e no espaço) de um ponto têm de considerar pontos não vizinhos vide figura 10.5.



Seqüência de Pixels:

linha j	linha j+1	linha j+2	linha j+3	linha j+4	linha j+5
vizinhança na linha j	vizinhança na linha j+1	vizinhança na linha j+2	vizinhança na linha j+3	vizinhança na linha j+4	vizinhança na linha j+5

Figura 10.5: Localização da vizinhança de um ponto após a transformação do sinal de multi para unidimensional

Quando a varredura progressiva é utilizada, por exemplo, pontos vizinhos situados na mesma linha continuam vizinhos após a conversão da seqüência de imagens em seqüência de pixels. No entanto, pontos próximos em linhas diferentes ficam separados uns dos outros no novo sequenciamento. O mesmo acontece com pontos próximos separados por um instante de tempo, conforme mostra a figura 10.6.

Com o objetivo de mapear vizinhanças bidimensionais de imagens em vizinhanças unidimensionais de seqüências de pixels pode-se utilizar varreduras derivadas de “curvas pre-

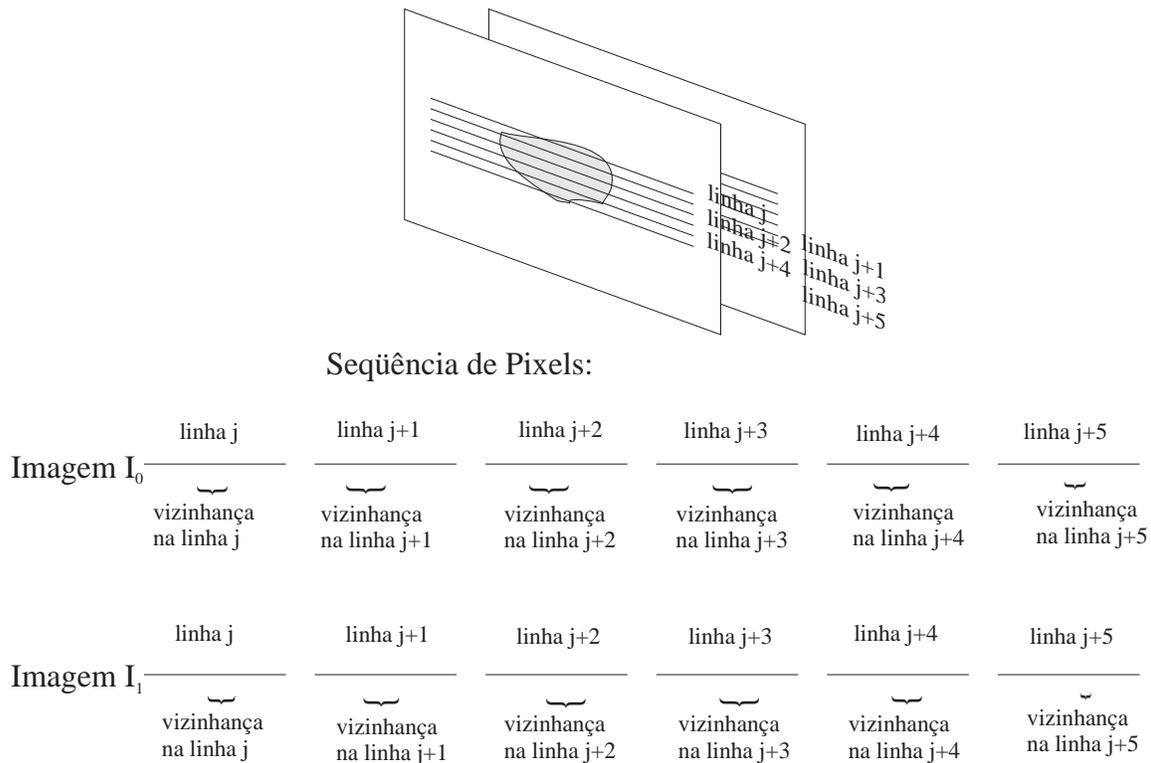


Figura 10.6: Relações de vizinhança após a ordenação dos pontos das imagens

enchedoras de área” (*space filling curves*), como as curvas de Peano (figura 10.7). Apesar desta potencialidade, este tipo de varredura não tem sido utilizado na prática. Isto porque o atrativo da codificação preditiva é a simplicidade de implementação, sendo a varredura derivada diretamente da varredura dos dispositivos de captação de imagens (câmeras). Desta forma, uma nova varredura implicaria em um aumento de complexidade nas câmeras (economicamente impraticável) ou na necessidade de armazenamento de um quadro inteiro para re-varredura (perda do atrativo de simplicidade).

Portanto, o sinal a ser predito é normalmente composto de pixels e obtido através de uma ordenação particular dos pixels nas imagens (varredura).

10.2.5 Determinação das vizinhanças dos elementos a serem preditos

Após definir qual o sinal a ser predito cabe definir quais os pontos a serem utilizados para realizar a predição de um dado ponto.

Existem três fatores importantes para determinar quais os pontos usados para predição:

- Comportamento do Sinal: é importante determinar a vizinhança cujos pontos influenciam na predição de um dado ponto.

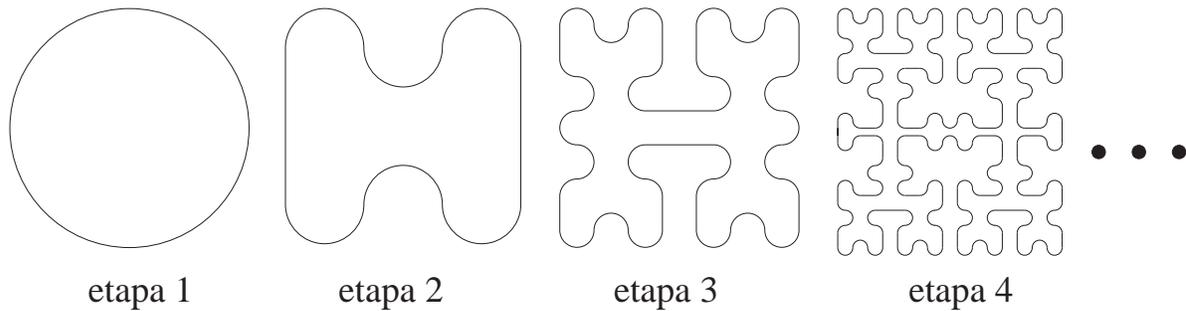


Figura 10.7: Curva de Peano

- Causalidade na Predição: a predição de um dado ponto tem de levar em conta apenas os pontos anteriores a ele (anteriores em relação à ordenação imposta pela varredura).
- Complexidade da Implementação: de acordo com a complexidade permitida para a implementação os preditores podem levar em consideração diferentes vizinhanças.

Em relação ao comportamento do sinal, existem três tipos de vizinhanças que podem ser consideradas:

- vizinhança espacial em uma dimensão (vizinhança na mesma linha)
- vizinhança espacial em duas dimensões
- vizinhança temporal

A vizinhança espacial em uma dimensão foi a primeira a ser utilizada. Ela é uma aplicação direta dos métodos de predição de sinais unidimensionais (sinal de voz, por exemplo). Sua implementação (quase sempre baseada em varredura progressiva) é simples porque a vizinhança na mesma linha é preservada na transformação das imagens em sinais 1-D (a menos, é claro, das extremidades das linhas).

Observou-se que poucos vizinhos são necessários para prever o comportamento de um pixel. A explicação desta observação encontra-se na próxima seção (10.2.6).

A predição 1-D apresenta uma limitação clara: ela não leva em conta a vizinhança de um pixel fora de sua linha. O conhecimento de pontos vizinhos (e até adjacentes) situados em outras linhas não é utilizado na predição dos pixels. Por isto a predição 1-D tem um potencial menor que a predição espacial bidimensional.

A predição bidimensional (2-D), por sua vez, requer mais recursos na sua implementação que a 1-D. De fato a predição de um pixel requer o armazenamento das linhas anteriores para que os vizinhos situados nestas linhas anteriores possam ser usados na predição. Também neste caso vale a observação que poucos pixels compõem a vizinhança utilizada para predição.

As predições espaciais procuram remover a redundância existente entre pixels vizinhos dentro de uma imagem (processamento intra-quadro). A predição temporal, por outro lado, procura remover a redundância temporal das imagens. Neste caso a vizinhança de um pixel

ultrapassa a barreira de sua imagem (ou quadro). Portanto trata-se de um processamento inter-quadro. A implementação requer ainda mais recursos computacionais devido ao armazenamento de pelo menos um quadro inteiro e ao processamento/estimação de preditores com quantidades maiores de pixels vizinhos.

Reverendo a fórmula básica de predição de sinais, equação 10.1, verifica-se que a predição é realizada recursivamente: o conhecimento dos pontos anteriores é utilizado para prever o próximo ponto. Portanto existe uma relação de causalidade no uso dos pontos para predição.

Entretanto, como a ordenação do sinal 1-D usado para predição está relacionada com a vizinhança dos pixels da imagem original, a causalidade tem implicações na determinação da vizinhança preditora dos pixels. Mais especificamente, nem toda vizinhança de um pixel pode ser usada para sua predição sem que a causalidade seja desrespeitada. Sob o ponto de vista da causalidade os preditores recebem a seguinte classificação:

- **Preditores Causais:** Os pixels utilizados para predição sempre antecedem o pixel a ser predito na ordenação 1-D. O significado desta restrição para o caso de varredura progressiva é o seguinte: dado um pixel no quadro Q , linha L e coluna C , sua predição pode utilizar todos os pixels dos quadros anteriores ao Q , todas as linhas anteriores à L dentro do quadro Q e todas as colunas anteriores à C dentro da linha L
- **Preditores Não-Causais:** Não há restrições devido à causalidade para vizinhança usada para predição de um pixel. Estes preditores são utilizados quando a predição é feita em blocos ou regiões de imagens. No caso de varredura progressiva (o caso mais importante e utilizado) existe uma sub-classe de preditores dentro da classe não-causal:
 - **Preditores Semi-Causais:** A vizinhança de predição inclui os pontos anteriores (vizinhança causal) mais os pixels situados na mesma linha do pixel a ser predito. Continuando com o exemplo do pixel P situado no quadro Q , linha L e coluna C , os pixels que podem ser utilizados para predição são os seguintes: todos os pixels dos quadros anteriores ao Q e todas as linhas anteriores ou iguais à L dentro do quadro Q (exceto obviamente o ponto P) Este tipo de predição se aplica quando linhas inteiras são preditas (maiores detalhes na seção 10.4).

A figura 10.8 ilustra as restrições às vizinhanças devido à causalidade para os três tipos de preditores:

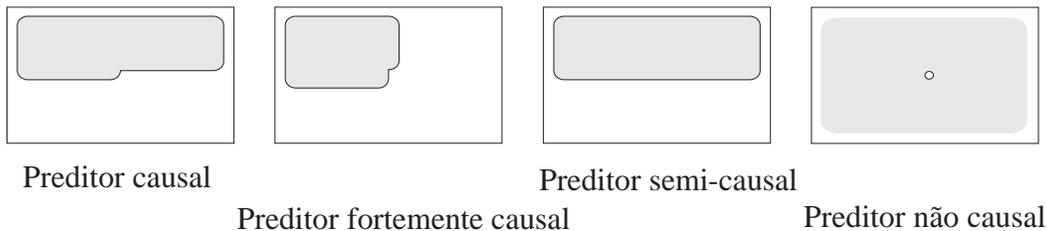


Figura 10.8: Restrições às vizinhanças usadas para predição de imagens

A complexidade de implementação também pode impor restrições às vizinhanças preditoras de imagens. Isto porque quanto maiores forem as vizinhanças maiores serão os recursos computacionais (capacidade de memória e processamento) necessários para implementar a predição. Um exemplo deste tipo de limitação ocorre na predição de movimentos, utilizada para reduzir a redundância temporal das imagens. Para cada bloco de imagem (normalmente um quadrado 8×8 ou 16×16), procura-se a melhor aproximação dentro de uma vizinhança contida no quadro anterior. A avaliação de proximidade de dois blocos 8×8 ou 16×16 requer uma quantidade considerável de operações aritméticas (ordem de n^2 onde n é 8 ou 16). Se a vizinhança for grande o número de vezes que a avaliação de proximidade terá que ser realizada também será grande.

Em verdade a quantidade de dados envolvidos no processamento de imagens é muito grande. Portanto, apesar de todos os avanços tecnológicos na informática, os algoritmos de processamento de imagens não podem se descuidar da sua eficiência computacional pois neles incide o risco de se tornarem impraticáveis.

10.2.6 Determinação dos Preditores e Estimação de Parâmetros

Os preditores têm de prever o comportamento futuro dos sinais com base nos acontecimentos do passado. A base para a maioria dos preditores é a constatação que existe uma forte correlação entre os pixels vizinhos (no espaço e no tempo) de imagens (vide capítulo 4).

A forma mais simples de prever um sinal com esta característica é considerar como predição de um pixel $P(m)$ seu pixel anterior $P(m \leftrightarrow 1)$. De fato este tipo de predição simples é bastante utilizada, tanto para predição espacial 1-D quanto para predição temporal. A ordem deste tipo de preditor é obviamente 1 (apenas um pixel anterior é utilizado na predição do próximo pixel).

Uma forma mais precisa de predição é obtida considerando os coeficientes de correlação horizontal e vertical entre pixels adjacentes (ρ_h e ρ_v respectivamente). Estes coeficientes são normalmente estimados levando-se em consideração uma grande massa de dados (várias imagens). Em seguida calcula-se a matriz de correlação da fonte de imagens de acordo com os coeficientes ρ_h e ρ_v . Por fim calcula-se os preditores da seguinte forma:

- Inicialmente determina-se a vizinhança a ser utilizada para predição. Para esta determinação deve-se ter em mente os fatores expostos na seção anterior.
- Em seguida calcula-se os pesos de cada ponto da vizinhança na predição. Este cálculo envolve a solução de um sistema de N equações a N incógnitas, onde N é o número de pontos na vizinhança de predição. Maiores detalhes sobre este sistema de equações podem ser encontrados em [107].
- A predição é realizada através de uma média ponderada dos pontos da vizinhança. Os pesos são dados pelos coeficientes calculados no passo anterior.

Vale explicar como se determina vizinhança de predição tendo em vista a adequação do preditor às características do sinal de imagem. Considera-se de início preditores de diferentes vizinhanças e ordens. Os valores de coeficientes para os diferentes preditores são determinados

através do mesmo modelo de imagens (que por exemplo pode ser o modelo baseado em correlação entre pontos adjacentes, conforme visto acima). Em seguida, para cada vizinhança V_k calcula-se o valor médio quadrático do sinal residual nas imagens: E_{V_k} .

Os valores E_{V_k} tendem a diminuir à medida em que a vizinhança V_k cresce. Isto porque quanto mais se sabe a respeito do passado de um sinal (maior a vizinhança V_k), melhor pode ser a previsão de um próximo ponto. Logo, menor será a incerteza do sinal residual e portanto menor será o erro de aproximação do sinal original por sua representação preditiva.

Sabe-se, por outro lado, que o comportamento das imagens é bem caracterizado localmente, isto é, existe uma correlação grande entre pontos de uma vizinhança, mas esta correlação não se mantém quando a vizinhança cresce. Portanto, não adianta considerar vizinhanças V_k muito grandes. V_k é considerada uma boa vizinhança de predição caso o modelo de imagens obtido com ela possa prever o sinal original com uma precisão bem melhor do que os modelos das vizinhanças menores, porém pouco ou nada pior que os modelos das vizinhanças maiores.

Este raciocínio vale tanto para vizinhanças 1-D (preditores espaciais 1-D) quanto para vizinhanças 2-D (preditores espaciais 2-D). A literatura de codificação preditiva ([114]) relata que vizinhanças de entre 2 a 4 pontos anteriores na mesma linha e entre 2 a 3 pontos na linha superior são suficientemente boas para obter uma boa predição de imagens. O livro [157] apresenta uma figura (figura 17 do capítulo 5) ilustrando a evolução E_{V_k} à medida em que a vizinhança V_k cresce (vizinhança 1-D). Apesar dos dados desta figura, transcrita para a figura 10.9 deste trabalho, referirem-se à codificação de apenas uma imagem específica, eles coincidem com as conclusões apresentadas em outras referências ([114]) e podem ser considerados típicos.

A figura 10.10 ilustra o método de determinação das vizinhanças dos preditores de imagens.

Além dos preditores simples de primeira ordem e dos baseados em correlações horizontal e vertical, existem preditores mais sofisticados. Diversos são os fatores de sofisticação que podem existir sozinhos ou combinados nos preditores de imagens:

- Predição Adaptativa: [114]
- Predição Não-linear
- Predição Anisotrópica: [190]

Estas sofisticações exigem maiores recursos computacionais tanto para a determinação dos preditores/estimação de parâmetros quanto para o cálculo dos valores preditos.

10.2.7 Quantização do Sinal Residual

Em princípio a quantização do resíduo $\{\epsilon(n)\}_n$ é bastante simples, conforme ilustrada na figura 10.11 :

Um aspecto importante deste esquema inicial, também chamado de *feedforward*, é o fato de que a quantização se encontra fora do laço de cálculo do resíduo $\{\epsilon(n)\}_n$. Isto implica que a quantização em um instante n não leva em consideração os erros (ou ruído) de quantização

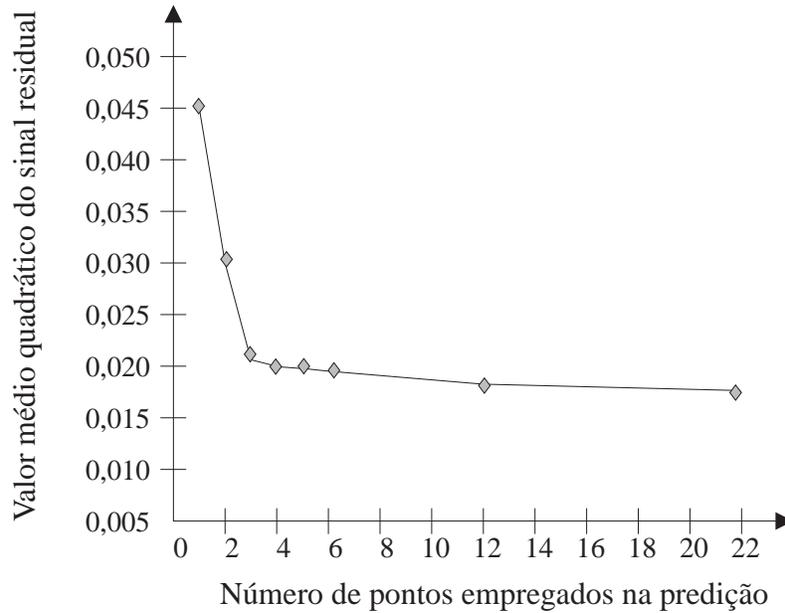


Figura 10.9: Evolução do valor médio quadrático do sinal residual na codificação preditiva da imagem exemplo do capítulo 5 de [157] à medida em que o número de pontos da vizinhança de predição 1-D cresce

que possam ter ocorrido nos instantes anteriores. Com isto erros de quantização podem se propagar indefinidamente.

Existe outro esquema de quantização em que esta pertence ao laço de cálculo do resíduo. Este esquema, chamado de *quantização com realimentação*, está ilustrado na figura 10.12:

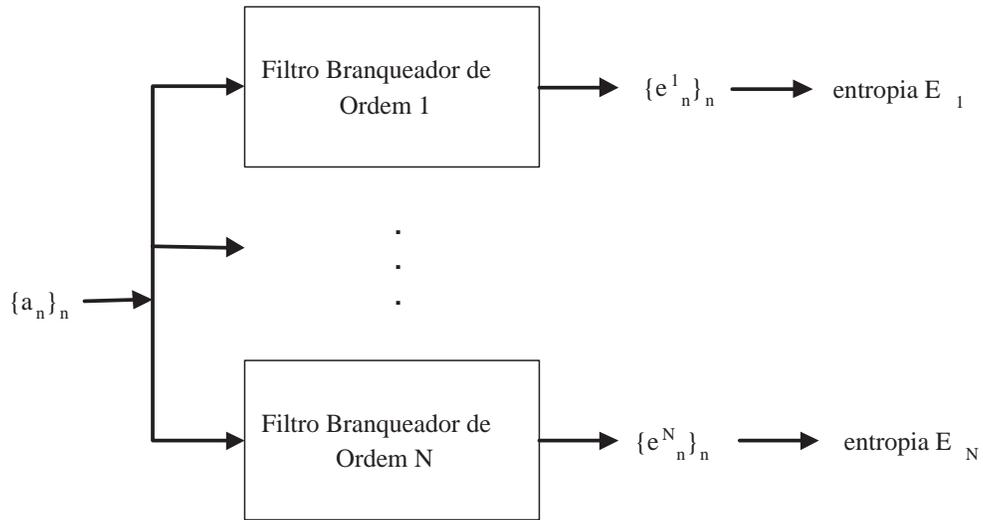
A grande vantagem da quantização com realimentação é que erros de quantização são compensados à medida em que as amostras vão sendo calculadas. O esquema evita propagação de erros de quantização. Neste ponto vale a pena apresentar um exemplo para clarear as idéias:

Seja um sinal $u(n)$ com a seguinte sequência de amostras:

$$\begin{aligned}
 u(0) &= 100 \\
 u(1) &= 102 \\
 u(2) &= 120 \\
 u(3) &= 120 \\
 u(4) &= 120 \\
 u(5) &= 118
 \end{aligned}$$

Seja seu preditor $\overline{u(n)} = u(n \Leftarrow 1)$. Por fim seja o quantizador do resíduo dado pela curva característica da figura 10.13.

A tabela abaixo resume a reconstrução do sinal $u(n)$ após uma codificação preditiva para



ordem mais adequada para o filtro branqueador (e conseqüentemente o preditor) é aquela a partir da qual $E(\text{ordem}) \sim E(\text{ordem}+K)$, $K>0$

Figura 10.10: Método utilizado para determinar as vizinhanças dos preditores de imagens

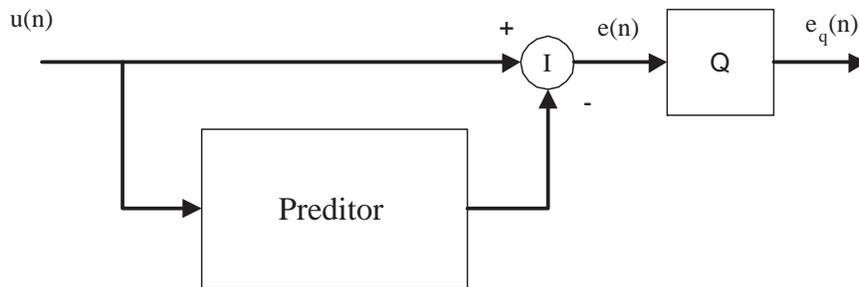


Figura 10.11: Abordagem inicial para a quantização dos resíduos de predição de imagens

ambos os casos de quantização: com realimentação e com *feedforward*:

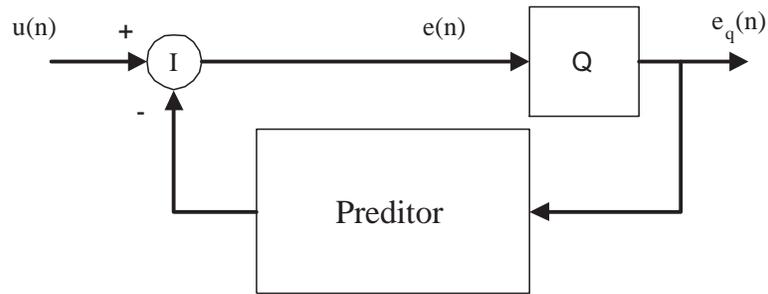


Figura 10.12: Esquema de quantização com realimentação

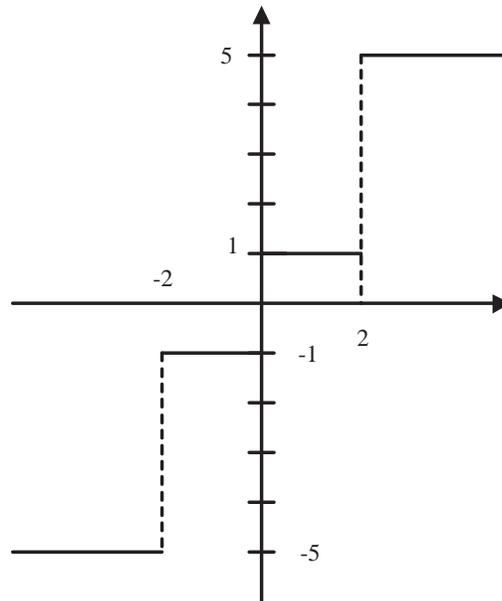


Figura 10.13: Curva característica do quantizador do resíduo no exemplo

n	Entrada			Realimentação				Feedforward			
	$u(n)$	$u_q(n)$	$\epsilon(n)$	$\epsilon_q(n)$	$u_q(n)$	$\delta u(n)$	$u(n)$	$\epsilon(n)$	$\epsilon_q(n)$	$u_q(n)$	$\delta u(n)$
0	100	—	—	—	100	0	—	—	—	100	0
1	102	100	2	1	101	1	100	2	1	101	1
2	120	101	19	5	106	14	102	18	5	106	14
3	120	106	14	5	111	9	120	0	-1	105	15
4	120	111	9	5	116	4	120	0	-1	104	16
5	118	116	2	1	117	1	120	-2	-5	99	19

Pode-se verificar na tabela acima que após uma situação de descontinuidade (amostras pulando de 102 para 120) os preditores falham em prever o comportamento do sinal e o resíduo

apresenta componentes mais altas. Depois da descontinuidade o erro acumulado tende a se anular no caso de quantização com realimentação o que não acontece na quantização com *feedforward*. Portanto é aconselhável incluir a quantização no laço de cálculo do resíduo.

10.3 Possibilidade de Emprego de Adaptabilidade

A adaptabilidade pode ser utilizada em dois processos na codificação preditiva:

- Determinação/Estimação de Preditores
- Quantização do Resíduo

Quanto aos preditores, a idéia básica da adaptabilidade consiste em ajustar os preditores de acordo com o comportamento observado do sinal. Esta idéia é interessante primeiramente porque permite ajustar os modelos de imagens de acordo com as reais características do sinal sendo codificado. Além disto, ela permite uma flexibilidade dos preditores de sinais. Com efeito, a adaptação permite que os preditores sejam capazes de acompanhar variações localizadas no comportamento dos sinais. Esta característica é bastante interessante para o caso de sinais de imagem. Isto porque o comportamento localizado das imagens altera-se consideravelmente quando se passa de uma região sem arestas para uma região com arestas.

Existem duas abordagens para a implementação desta adaptação: preditores paramétricos e escolha de um preditor dentre um conjunto de possíveis preditores.

A primeira abordagem consiste na determinação de preditores que são função de um ou mais parâmetros. À medida em que os sinais vão sendo processados o(s) parâmetro(s) são estimados a partir das amostras de sinal observadas. Com isto os preditores são alterados de acordo com as características dos sinais.

A segunda abordagem requer um conjunto de preditores ajustados para os diferentes comportamentos locais dos sinais. Conforme os sinais são processados avalia-se qual o melhor preditor dentre as possibilidades pré-determinadas. Mais especificamente os sinais são segmentados e um melhor preditor é escolhido e utilizado para cada segmento. Com este esquema a codificação dos sinais requer informações de controle para que tanto a segmentação quanto os melhores preditores possam ser restaurados. Um exemplo de codificador apresentando esta abordagem pode ser encontrado em [114].

A adaptabilidade na quantização é utilizada para ajustar os limites de quantização durante o decorrer dos sinais. Os sinais de resíduo normalmente concentram-se no valor 0. Isto é tão mais verdade quanto melhores forem os preditores de sinais. Entretanto, mesmo com preditores bons é inevitável que existam situações em que estes não consigam prever o comportamento dos sinais. Afinal de contas apesar de uma razoável previsibilidade os sinais continuam sendo processos estocásticos e, portanto apresentam uma flutuação aleatória em torno do comportamento previsível.

Se a quantização do resíduo for muito fina nas proximidades de 0 ela será muito boa para as situações em que os sinais se ajustarem bem ao seu comportamento “padrão”. No entanto, quando os sinais apresentarem desvios razoáveis do comportamento previsto (como no caso da presença de arestas), os sinais residuais serão maiores e, assim, a quantização fina não será adequada. Caso, por outro lado a quantização seja mais grosseira ela adequar-se-á para as

regiões com resíduos maiores, mas será inadequada para regiões com comportamento próximo do previsível. Esta não adequação da quantização para todas as situações fica agravada para a codificação preditiva porque seus quantizadores são geralmente muito simples, apresentando poucos níveis de quantização.

A quantização adaptativa procura superar esta limitação. A idéia básica consiste em verificar a distância entre o sinal original e o reproduzido através da predição mais a adição do resíduo quantizado. Quando a diferença for grande o sinal original está se comportando de forma menos previsível e, portanto os limites de quantização devem ser ampliados. Quando, por outro lado, a diferença for pequena os limites de quantização devem ser reduzidos.

10.4 Possibilidade de Emprego de Metodos Híbridos

Métodos híbridos são aqueles que empregam mais de um tipo de codificação (preditiva, por transformada, por aproximação de blocos da imagem ou por modelagem de cenas) para representar as imagens.

A utilização de codificação preditiva em métodos híbridos é relativamente simples: ela é utilizada para reduzir a redundância de uma ou mais das coordenadas do sinal de imagens.

Um primeiro exemplo ocorre quando usa-se uma transformada 1-D para cada coluna de uma imagem. Em seguida realiza-se codificação preditiva linha-a-linha no quadro. Assim usa-se dois métodos para redução de redundâncias: um para cada dimensão de uma imagem. Este exemplo tem pouca aplicação prática hoje-em-dia. Sua aplicação justifica-se em aplicações em que a complexidade computacional das transformadas 2-D seja proibitiva, o que não ocorre com as transformadas mais utilizadas atualmente (todas transformadas rápidas).

Outro exemplo, este com muita aplicação prática, é o de métodos de codificação que utilizam predição para reduzir a redundância temporal em imagens dinâmicas. De fato esta é a abordagem mais comum para codificação inter-quadro.

10.5 Possibilidade de Emprego de Codificação sem Perdas

A única etapa da codificação preditiva que pode apresentar perdas ao sinal codificado é a quantização. Portanto, se a quantização for realizada sem perdas toda a codificação será sem perdas.

A ausência de perdas na quantização não é uma exigência muito forte. Com efeito, o sinal predito tem a mesma precisão do original. Portanto o resíduo (subtração dos dois sinais anteriormente citados), necessita de apenas um bit a mais que o sinal original para que não apresente erros. Apesar da existência de um bit a mais para cada amostra o método continua a apresentar compressão de informação, pois os sinais concentram-se no zero e, portanto, apresentam menores entropias. É claro que para realizar o potencial de compressão de informação é preciso haver uma fase de atribuição de novos códigos binários para os valores do resíduo (vide capítulo 8).

10.6 Complexidade da Implementação

Conforme mencionado na seção 10.2.4, o grande atrativo da codificação preditiva é sua baixa complexidade computacional. De fato o método clássico de codificação preditiva é a filtragem linear branqueadora, que é bem mais barata computacionalmente que os demais métodos de codificação (vide capítulos subsequentes).

Entretanto quando predições mais sofisticadas que a linear isotrópica são utilizadas pode-se ter um aumento considerável na complexidade computacional da codificação.

Mesmo com preditores lineares a complexidade computacional pode crescer. Quando se utiliza predição de movimento (seção 10.7.5), por exemplo, as predições envolvem pesquisas de melhor aproximação em dimensões grandes.

10.7 Variantes da Codificação

Esta seção tem como objetivo apresentar diversas implementações práticas de codificação preditiva. Antes de começar as implementações convém alertar ao leitor que a utilização da codificação preditiva apresenta duas vertentes normalmente separadas:

- Predição espacial ou intra-quadro
- Predição temporal ou inter-quadro

A predição inter-quadro procura prever o comportamento dos pixels com base nas ocorrências dos pixels vizinhos dentro do mesmo quadro. A codificação baseada neste tipo de predição tem como objetivo reduzir a redundância espacial existente nos quadros de imagens (ou imagens estáticas).

A predição temporal, por outro lado, modela o comportamento dos pixels levando em consideração os valores dos pixels vizinhos em quadros anteriores. Desta forma a codificação baseada neste tipo de predição pretende reduzir a redundância temporal das imagens.

10.7.1 Modulação Delta

A *modulação delta* é uma forma muito simples de codificação preditiva de sinais unidimensionais. A predição do sinal $\{u(n)\}_n$ no ponto k $\overline{u(k)}$ é simplesmente o sinal codificado no ponto anterior: $u_q(k \Leftrightarrow 1)$. A quantização do sinal residual também é muito simples: $u_q(k) = u_q(k \Leftrightarrow 1) + \delta$ se $u(k) \geq \overline{u(k)}$ ou $u_q(k) = u_q(k \Leftrightarrow 1) - \delta$ caso contrário.

Pode-se observar, portanto, que a quantização baseia-se num parâmetro δ , que por sinal inspira o nome do método de codificação.

Este método de codificação apresenta três problemas bastante conhecidos:

- Sobrecarga de Inclinação
- Ruído Granular
- Instabilidade em Presença de Erros

Quando o sinal apresenta um crescimento ou uma queda acentuada, isto é, uma inclinação forte, a seqüência $u_q(n) = u_q(m_i) + (n \Leftrightarrow m_i) * \delta$ resultante da codificação pode não ser capaz de acompanhar o sinal. A figura 10.14 ilustra este tipo de situação, conhecida por *Sobrecarga de Inclinação*.

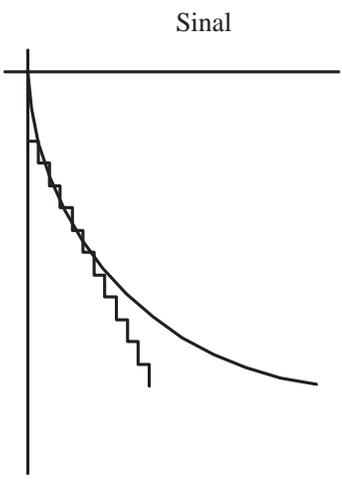


Figura 10.14: Situação de sobrecarga de inclinação

Considere-se, por outro lado, a situação em que o sinal se estabiliza num valor não múltiplo de δ ($u(n) = L$ para $n \in [N_i, N_f]$). Seja $u_q(n)$ um valor do sinal quantizado tal que $u_q(n) < u(n)$. O próximo valor quantizado será $u_q(n+1) = [u_q(n) + \delta] > u(n+1)$. Em seguida, o outro valor será $u_q(n+2) = [u_q(n+1) \Leftrightarrow \delta] = u_q(n)$. Portanto o sinal codificado oscila entre os valores $u_q(n)$ e $u_q(n+1)$. A este fenômeno de oscilação do sinal codificado a partir de um sinal original estável dá-se o nome de Ruído Granular. A figura 10.15 ilustra este fenômeno.

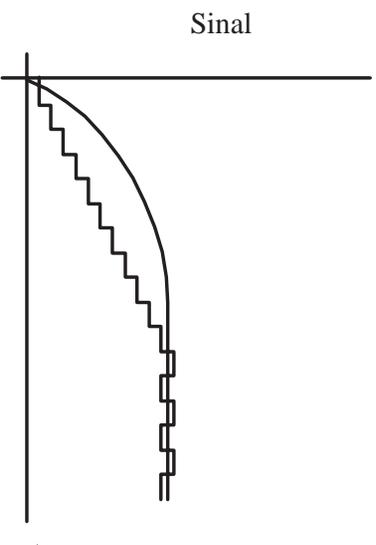


Figura 10.15: Ruído Granular

Valores grandes de δ reduzem a chance de sobrecarga de inclinação, porém aumentam o ruído granular. δ 's menores, por outro lado, reduzem o ruído granular mas aumentam a chance de sobrecarga de inclinação. Portanto a escolha do parâmetro δ envolve um compromisso entre sobrecarga de inclinação e ruído granular.

A sobrecarga de inclinação também pode ser reduzida com o aumento da taxa de amostragem do sinal original. Desta forma num mesmo intervalo podem existir mais amostras de sinal codificado e, portanto, um maior número de degraus de altura δ para acompanhar a inclinação do sinal original. A figura 10.16 mostra como o aumento da taxa de amostragem pode reduzir o efeito de sobrecarga de inclinação.

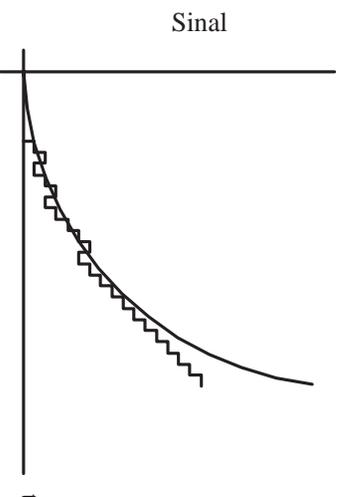


Figura 10.16: Aumento da taxa de amostragem reduzindo o efeito da sobrecarga de inclinação

Uma forma simples de reduzir o ruído granular consiste na sofisticação do quantizador do resíduo: $u_q(k) = u_q(k \Leftrightarrow 1)$ se $u(k)$ for suficientemente próximo de $u_q(k \Leftrightarrow 1)$. Caso contrário, $u_q(k) = u_q(k \Leftrightarrow 1) + \delta$ se $u(k) \geq \overline{u(k)}$ ou $u_q(k) = u_q(k \Leftrightarrow 1) \Leftrightarrow \delta$. Esta codificação tem como resultado três níveis a serem codificados:

- Sinal mantido inalterado
- Sinal incrementado de δ
- Sinal decrementado de δ

Tanto o aumento da taxa de amostragem quanto a sofisticação da quantização são opções que aumentam a qualidade da modulação δ em troca de um acréscimo de informação codificada. De fato, no caso do aumento da taxa de amostragem, necessita-se de mais informação por unidade de tempo para representar o sinal, dado que há uma quantidade maior de amostras por unidade de tempo. No caso da quantização em três níveis, por outro lado, existe um nível a mais (*Sinal mantido inalterado*) na saída do quantizador para cada amostra. Estes são dois exemplos do compromisso existente entre taxa de compressão de informação e qualidade da representação utilizada.

O terceiro problema da modulação δ é a instabilidade em presença de erros. Como a codificação é baseada na representação do sinal a partir de valores anteriores, erros podem se propagar até que haja um reinício da codificação dos sinais.

10.7.2 Modulação Delta Adaptativa

Uma alternativa melhorada da modulação delta é sua versão adaptativa. A diferença entre as versões normais e adaptativas da modulação delta reside no fato que o salto de quantização é variável para o caso adaptativo. Esta variação resolve na prática o problema de sobrecarga de inclinação.

10.7.3 Modulação DPCM (“*Differential Pulse Code Modulation*”)

Esta modulação é a preditiva tradicional com ordem 1 e previsão do próximo ponto igual ao ponto atual, isto é:

$$u(\tilde{n}) = u(n \Leftrightarrow 1)$$

Esta predição apresenta como vantagem sua simplicidade e possui razoável eficiência porque consegue eliminar a entropia condicional entre intensidades luminosas de pixels vizinhos. Observando a figura 10.9, entretanto, pode-se observar que a eficiência pode crescer significativamente caso a ordem de predição seja aumentada para 2 ou 3.

10.7.4 Reabastecimento Condicional

Este método de predição se aplica à codificação inter-quadro, ou seja, a imagens dinâmicas. A idéia básica consiste em comparar os valores de grandezas correspondentes em imagens vizinhas (intensidades luminosas em pixels correspondentes, por exemplo) e codificar os valores da imagem nova somente se estes valores ultrapassarem um limite pré-determinado.

A motivação para o método vem do fato de muitos pixels correspondentes em imagens sucessivas numa seqüência de quadros apresentarem valores muito próximos ou iguais. De fato as imagens podem ser divididas em regiões com variações temporais e regiões aproximadamente estáticas. Como é bastante comum em seqüências de imagens que as regiões estáticas sejam grandes, obtem-se compressão maior se apenas as regiões dinâmicas forem codificadas.

Outras grandezas podem ser codificadas através deste princípio de reabastecimento condicional. Um exemplo diferente dos valores de pixels correspondentes em imagens vizinhas é o do conjunto de vetores que formam o livro-código de uma quantização vetorial adaptativa (vide seção 7.3.2 do capítulo 7).

10.7.5 Predição de Movimentos

Esta técnica também se aplica unicamente no caso de imagens dinâmicas. Sua idéia básica, tal como descrito na seção 4.2.3, é que existe muita correlação entre imagens vizinhas numa seqüência dinâmica. Esta correlação ocorre porque objetos observados em um instante também serão (com grande probabilidade) observados na cena seguinte (vide figura 4.10).

Além de adequar-se muito bem ao comportamento dinâmico das imagens, esta técnica de redução de redundância temporal de seqüências de imagens pode ser aplicada de forma híbrida junto com diferentes métodos de codificação intra-quadro. Este é o caso, por exemplo, do padrão de codificação MPEG ([62]) que utiliza (entre outras técnicas) a predição de

movimentos para redução de redundância temporal e DCT para codificação estática (intra-quadro).

Capítulo 11

Codificação por Transformada

11.1 Introdução

Tal como visto no capítulo 2, as imagens digitais são originalmente descritas através das intensidades luminosas de cada componente de cor nos pontos amostrados. Esta descrição é uma representação da imagem em um sistema de coordenadas. A base deste sistema é formada por vetores (no caso imagens) apresentando intensidade luminosa de valor 1 em apenas um ponto amostrado (pixel) e 0 em todos os demais pontos, como ilustra a figura 11.1.

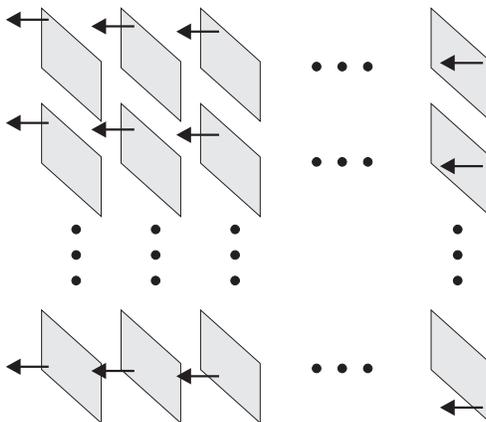


Figura 11.1: Base do sistema de coordenadas de representação de imagens. As setas indicam os pontos nas imagens (vetores da base do sistema de coordenadas) em que as intensidades são 1

O propósito da codificação por transformadas é representar as imagens em outro sistema de coordenadas tal que os novos coeficientes caracterizadores das imagens sejam mais “propícios” para a codificação. Os novos coeficientes serão propícios para a codificação caso facilitem o trabalho da fase de quantização (próxima fase no processo de codificação de imagens). Eis algumas características desejáveis para os novos coeficientes:

- Descorrelacionamento estatístico
- Compactação de “energia”, ou seja, poucos coeficientes concentrarem grande parte da informação das imagens
- Significado físico diretamente relacionado com as propriedades da visão humana

As duas primeiras características referem-se à redução de redundância das fontes de imagens, enquanto que a terceira está relacionada com o aproveitamento do conhecimento sobre visão para minimização da distorção entre uma imagem original e sua versão codificada. Nas próximas seções estão apresentados os passos do método de transformação de coordenadas, bem como as razões pelas quais os coeficientes dos sistemas de coordenadas utilizados possuem aproximadamente as características desejáveis.

11.2 Descrição do Método

11.2.1 Decomposição da Imagem em Blocos

As transformações de coordenadas das imagens podem ser realizadas considerando-se todos os seus pontos de uma vez. Esta abordagem, no entanto, não é muito adequada.

Inicialmente porque as imagens geralmente não apresentam uma uniformidade em seu comportamento, sendo melhor modeladas através de modelos locais. Além disto as operações de transformação de coordenadas ficam mais baratas computacionalmente caso sejam realizadas em dimensões menores.

Portanto, antes de serem transformadas, as imagens são decompostas em blocos. Posteriormente as mudanças de coordenadas são feitas em cada um dos blocos.

Embora possam assumir quaisquer formato ou dimensão, os blocos costumam ser quadrados 8×8 ou 16×16 .

Existe um compromisso entre a melhor adequação dos blocos às características locais das imagens e a necessidade de armazenamento de informação adicional para determinar os blocos. A determinação da decomposição da imagem em blocos quadrados de tamanho fixo necessita de muito pouca informação. No entanto não apresenta flexibilidade para se adequar a variações de comportamento de imagens em regiões menores que as dimensões do quadrado. Uma decomposição muito flexível, por outro lado, requer muita informação para ser caracterizada. Embora a grande maioria das aplicações realmente utilize blocos quadrados, decomposições retangulares flexíveis, utilizando árvore quaternária (*quad-trees*) por exemplo, podem ser empregadas.

11.2.2 Transformação de Coordenadas

Seja \mathcal{B} um bloco de imagem de dimensões $M \times M$. As intensidades luminosas do bloco são determinadas pelos coeficientes $b_{ij}, i \in [0, M \Leftrightarrow 1], j \in [0, M \Leftrightarrow 1]$. Uma mudança de coordenadas é uma transformação linear \mathcal{L} que leva o bloco inicial \mathcal{B} à nova representação da imagem \mathcal{B} com coeficientes $c_{pq}, p \in [0, M \Leftrightarrow 1], q \in [0, M \Leftrightarrow 1]$.

As transformações lineares utilizadas para as mudanças de coordenadas são unitárias,

$$\mathcal{L}\mathcal{L}^* = \mathcal{I}$$

onde \mathcal{I} é a identidade. Por questões de praticidade faz-se outras restrições às transformações lineares:

- Elas devem ser reais. Juntamente com o unitarismo esta simplificação implica no fato que as transformadas têm inversas, que são exatamente suas transformações transpostas.
- Elas devem ser separáveis. Desta forma a transformação de coordenadas pode ser escrita como a composição de duas transformações em R_N : T_1 e T_2 :

$$\mathcal{G} = T_1\{\mathcal{B}\}T_2^T$$

O retorno às coordenadas iniciais é realizado através da composição das transformações inversas:

$$\mathcal{B} = T_1^T\{\mathcal{G}\}T_2$$

Cada transformada (T_1 ou T_2) corresponde à versão unidimensional da transformação de coordenadas escolhida para ser realizada em duas dimensões (Uma terceira dimensão pode ser adicionada se a variável tempo for considerada).

As transformações mais utilizadas, bem como as definições dos elementos das matrizes que as descrevem estão apresentadas abaixo:

- Transformada Walsh-Hadamard [152]:

$$T_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & \Leftrightarrow 1 \end{bmatrix}$$

$$T_N = \frac{1}{\sqrt{2}} \begin{bmatrix} T_{N/2} & T_{N/2} \\ T_{N/2} & \Leftrightarrow T_{N/2} \end{bmatrix}$$

Evidentemente esta transformação está definida apenas para valores de N potências de 2.

- Transformada Cosseno [152]

– DCT-I

$$[T_N]_{pq} = \frac{2}{(N \Leftrightarrow 1)}^{1/2} [k_p k_q \cos(\frac{pq\pi}{N \Leftrightarrow 1})]$$

– DCT-II

$$[T_N]_{pq} = \frac{2}{(N \Leftrightarrow 1)}^{1/2} [k_p \cos(\frac{p(q + 1/2)\pi}{N \Leftrightarrow 1})]$$

– DCT-III

$$[T_N]_{pq} = \frac{2}{(N \Leftrightarrow 1)}^{1/2} [k_q \cos(\frac{(p+1/2)q\pi}{N \Leftrightarrow 1})]$$

– DCT-IV

$$[T_N]_{pq} = \frac{2}{(N \Leftrightarrow 1)}^{1/2} [\cos(\frac{(p+1/2)(q+1/2)\pi}{N \Leftrightarrow 1})]$$

onde $p, q \in [0, N \Leftrightarrow 1]$ e

$$k_p = 1 \text{ se } p \neq 0 \text{ ou } p \neq N$$

$$k_p = \frac{1}{\sqrt{2}} \text{ se } p = 0 \text{ ou } p = N$$

• Transformada Seno [152]

$$[T_N]_{pq} = \frac{2}{(N+1)}^{1/2} [\sin(\frac{(p+1)(q+1)\pi}{N+1})]$$

Uma vez realizada a transformação de coordenadas a representação de uma imagem é feita através da matriz \mathcal{G} .

11.2.3 Quantização dos Coeficientes Transformados

O próximo passo da codificação consiste em quantizar os coeficientes transformados. Esta quantização depende das características estocásticas dos coeficientes. Na seção 11.3 estas características estão apresentadas, enquanto que as diferentes abordagens do problema de quantização de coeficientes quantizados foram apresentadas na seção 7.3.6.

11.2.4 Atribuição de Códigos Binários aos Símbolos da Quantização

Uma vez definido o alfabeto de saída dos quantizadores utiliza-se algum método de atribuição de bits aos elementos do alfabeto. Os métodos mais utilizados são o de Huffman e o aritmético em suas versões adaptativas (vide seções 8.2 e 8.3).

11.3 Base teórica para a Codificação

11.3.1 Fundamentos para a Redução de Redundância - Desacoplamento Estatístico entre Coeficientes

A escolha do novo sistema de coordenadas das imagens tem como base o modelo de covariância das imagens. Uma vez conhecida a covariância de fontes de imagens pode-se determinar o sistema de coordenadas que desacopla estatisticamente os coeficientes através da transformada Karhunen-Loève (seção 4.3.3).

Conforme provado no capítulo 4, além de descorrelacionar os coeficientes, este sistema de coordenadas é tal que maximiza a compactação de energia dos coeficientes.

Portanto, do ponto de vista teórico existe um sistema de coordenadas ótimo para representar as imagens desde que se possa obter a auto-covariância da fonte de imagens.

Na prática, porém, as fontes de imagens não são estacionárias, apresentando comportamento variável. Desta forma o modelo de covariância não descreve precisamente as fontes e, portanto, não se pode obter as transformações ótimas. Apesar disto, as imagens têm comportamento próximo do estacionário quando observadas em âmbito local (pequenas vizinhanças). Portanto, a aproximação do modelo de covariância localmente é satisfatória.

Seguindo este modelo local de covariância, uma possível abordagem para a codificação consistiria nos seguintes passos:

- decomposição das imagens em regiões com o mesmo comportamento local
- estimação das auto-covariâncias para cada uma das regiões
- determinação das transformadas KL
- transformação dos coeficientes segundo as KL's de cada região
- codificação da informação necessária para reproduzir a imagem (regiões decompostas, transformadas KL e coeficientes para cada região)

De fato estes são essencialmente os passos do método de codificação por transformadas, descrito na seção anterior, utilizando as seguintes considerações práticas:

- a decomposição das imagens é feita através de uma malha de quadrados $M \times M$
- Aproxima-se a transformada KL por alguma outra (normalmente DCT) tomando por base uma estimativa global (menos precisa que a local) da auto-covariância
- codifica-se os quadrados da decomposição (número de quadrados na imagem e sua dimensão M) e os coeficientes em cada quadrado

Várias transformadas, tais como a de Fourier, seno, cosseno, Walsh-Hadamard e Haar ([152]) têm sido propostas e utilizadas em codificação de imagens. A ênfase, no entanto, concentra-se na transformada cosseno (DCT), devido à proximidade de seu comportamento ao da transformada KL para o caso de fontes de imagens (vide seção 4.3.2).

Resumindo, do ponto de vista estatístico, a utilidade destes métodos de transformadas para a codificação de imagens está baseada:

- na adequação do modelo de covariância para descrever localmente as fontes de imagens e
- na boa aproximação da transformada KL pela DCT para as fontes de imagens.

Além destes pontos, há aspectos práticos importantes que reduzem os custos computacionais destes métodos, tornando-os implementáveis mesmo em tempo real (com a tecnologia de hoje):

- A existência de algoritmos rápidos para implementar muitas das transformadas utilizadas (FFT, DCT, DST, Walsh-Hadamard e Haar)

- A aplicação das transformadas em blocos pequenos de imagens (normalmente retângulos 4×4 , 8×8 , 16×16 ou 32×32). Além de permitir a adequação do modelo de covariância ao comportamento local das imagens, a utilização destes blocos pequenos restringe os custos computacionais envolvidos nas transformadas.

11.3.2 Fundamentos para a Redução de Redundância - Comportamento dos Coeficientes

O decorrelacionamento estatístico entre os coeficientes transformados significa que não há grande vantagem na utilização da quantização vetorial em relação à escalar. Resta saber como devem ser distribuídos os níveis de quantização para cada um dos coeficientes e como deve ser realizada a quantização em cada coordenada.

Vários trabalhos publicados na literatura procuraram caracterizar as distribuições de probabilidades dos coeficientes ([28]).

Antes de apresentar as conclusões é importante estabelecer uma notação. Em muitas transformadas (cosseno, Hadamard e Haar, por exemplo), os coeficientes estão diretamente relacionados com o espectro de potência da imagem. Isto porque os vetores das bases (imagens) relacionam-se com os vetores que formam a base da transformada de Fourier. Mais especificamente, a imagem-base com índice p , q apresenta p oscilações na direção horizontal e q na vertical (vide exemplo com $p = 3$ e $q = 4$ na figura 11.2). Portanto o coeficiente c_{pq} está diretamente relacionado com a componente de potência da imagem nas frequências horizontais e verticais p e q respectivamente. Nestes casos o coeficiente 0,0 armazena a média da intensidade luminosa no retângulo transformado. Ele é chamado de coeficiente DC. Os demais coeficientes são chamados de AC (esta denominação vem da Engenharia Elétrica).

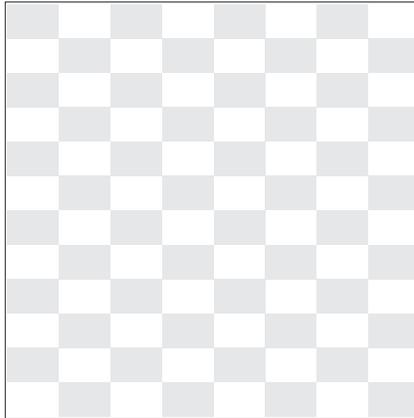


Figura 11.2: Imagem base com 3 oscilações na direção horizontal e 4 na vertical

Estabelecida a nomenclatura, pode-se apresentar as conclusões relatadas na literatura:

- Os coeficientes DC são uma versão sub-amostrada das imagens. De fato eles representam a média da intensidade luminosa em suas regiões (filtragem passa-baixas)

- Existe forte correlação entre os coeficientes DC de regiões vizinhas. Isto pode ser entendido uma vez que os níveis médios de luminosidade não costumam mudar abruptamente entre regiões vizinhas.
- O coeficiente DC possui média positiva, já que representa valores médios de uma grandeza sempre positiva (intensidade luminosa)
- Os coeficientes AC possuem médias aproximadamente 0. Suas distribuições são aproximadamente simétricas em torno de 0, aonde formam um pico. O trabalho [17] propôs o modelo de distribuições Laplacianas para os coeficientes AC. No entanto, não há concordância da literatura quanto ao modelo a ser adotado devido aos diferentes resultados obtidos em estimações destas distribuições (vide livro [28]).

Estes dados evidenciam um comportamento bastante diferenciado dos coeficientes DC e AC. Justamente por isto os métodos de codificação de imagens normalmente quantizam e armazenam os coeficientes de duas formas bem distintas (vide seção 7.3.6 do capítulo 7).

11.3.3 Características da Visão Humana Exploradas

Um aspecto muito importante dos coeficientes obtidos através das transformadas comumente utilizadas na codificação de imagens é seu significado físico, isto é, sua relação com o espectro de potências das imagens. A razão da importância deste significado físico é que ele permite estabelecer uma relação entre a percepção de ruído da visão humana e os coeficientes.

A sensibilidade ao ruído da visão humana varia no espectro de frequências espaciais (vide seção 5.4.4). Assim, a variação de sensibilidade a ruído induz no espectro medidas de distorção diferenciadas para cada frequência e, conseqüentemente, para cada coeficiente transformado. Portanto, pode-se adequar os níveis de quantização (implicitamente os ruídos de quantização) de cada coeficiente à variação de sensibilidade de ruído da visão na frequência.

11.4 Possibilidade de Emprego de Adaptabilidade

A motivação para o uso de adaptabilidade nestes métodos vem da própria modelagem local utilizada para fundamentá-los. Idealmente, através do modelo do comportamento das imagens determina-se

- a transformada descorrelacionadora de coeficientes,
- a atribuição de níveis de quantização ótimos para os coeficientes,
- os quantizadores a serem utilizados e, finalmente,
- os códigos binários a serem utilizados por cada símbolo codificado.

Cada determinação citada acima pode ser feita de forma adaptativa à medida em que a imagem é analisada. Para isto basta modelar localmente as imagens, estimar os parâmetros do modelo em cada região e utilizar estes parâmetros para finalmente determinar os itens acima.

11.4.1 Validação Local da Aproximação da Transformada KL por Transformadas Rápidas

O primeiro ítem que pode ser determinado de acordo com o modelo local das imagens é a decomposição da imagem em regiões com comportamento constante. Embora a decomposição não trivial (não necessariamente composta por uma malha retangular) de imagens seja usada em outros métodos, ela não é muito usada nos métodos de codificação por transformadas.

As transformadas são o próximo ítem a ser determinado adaptativamente. Uma primeira abordagem para aproximar a transformada KL adaptativamente consiste em modelar a covariância, estimar os parâmetros do modelo a partir de dados da região sendo codificada e, em seguida, calcular a transformada KL que diagonaliza a covariância resultante.

Outra abordagem para esta adaptabilidade consiste em permitir um leque de possibilidades para as transformadas, calcular os coeficientes obtidos por todas as transformadas e verificar qual o conjunto de coeficientes mais propício para a codificação.

Estas abordagens não são muito usadas porque a utilização da transformada DCT em retângulo é uma aproximação suficientemente boa da KLT em regiões de comportamento constante.

11.4.2 Quantização dos Coeficientes de Acordo com as Características Locais das Imagens

A quantização dos coeficientes é a etapa mais importante do método de codificação por transformadas. É nesta etapa que faz-se a parte mais importante da compressão dos dados ao se quantizar coeficientes menos importantes de forma grosseira. Justamente por isto, esta etapa é responsável pela qualidade das imagens codificadas.

Conforme visto anteriormente na seção 7.3.6 do capítulo 7, não existe uma solução para o problema de quantização dos coeficientes transformados que seja adequada para todos os tipos de imagens. Por isto, os métodos adaptativos são de grande proveito nesta área.

A codificação por limiar (vide seção 7.3.7) por si só já é adaptativa. Além disto, ela pode ser utilizada em conjunto com a codificação zonal.

11.4.3 Atribuição Adaptativa de Códigos Binários aos Símbolos Codificados

Esta adaptabilidade faz parte da fase de “Atribuição de Códigos Binários” aos símbolos da saída da quantização. Normalmente a adaptação é feita a nível de imagem completa e não bloco a bloco. Em outras palavras, as probabilidades de ocorrência dos símbolos são estimadas durante o decorrer de toda a codificação da imagem. Com base nestas probabilidades códigos binários são atribuídos aos símbolos. É claro que nestas situações o conjunto de probabilidades dos símbolos tem de ser enviado para o decodificador antes do início da decodificação.

11.5 Possibilidade de Emprego de Métodos Híbridos

Os métodos de codificação por transformadas têm sido utilizados conjuntamente com outros métodos (hibridismo) há décadas.

Tal utilização foi motivada inicialmente porque os custos computacionais das transformadas eram muito altos para que estas pudessem ser realizadas em duas dimensões para codificação de imagens em tempo real (pelo menos 20 imagens por segundo). Para reduzir o número de operações necessárias realizava-se transformadas em uma dimensão e os coeficientes resultantes eram codificados de forma preditiva na outra dimensão.

Este tipo de hibridismo tinha como motivação puramente uma limitação tecnológica e caiu em desuso quando a tecnologia deixou de ser um gargalo para a execução de transformadas 2-D em tempo real.

Existem outras possibilidades de hibridismo que têm como motivação a melhor adequação ao modelo de imagens, e não limitações tecnológicas:

- Codificação preditiva na coordenada tempo
- Codificação de regiões com arestas através de outro métodos

11.5.1 Codificação Preditiva na Coordenada Tempo

Os métodos preditivos e de transformadas utilizam essencialmente o mesmo princípio de correlação entre pontos vizinhos para redução de redundância estatística de fontes.

As transformadas, porém, têm como característica a utilização de modelos englobando toda a vizinhança dos pontos de imagens (modelos não causais). Nas dimensões espaciais esta característica é uma vantagem importante. Na dimensão temporal, por outro lado, as transformadas não apresentam grande aplicação. Primeiramente porque as alterações temporais de imagens são melhor modeladas através de preditores de movimento (vide seção 10.7.5). Além disto, a realização de transformadas na dimensão tempo requer o armazenamento de diversos quadros de uma seqüência de imagens antes de sua codificação. Este armazenamento causa um atraso indesejável entre a geração de imagens por uma fonte e sua codificação.

Portanto a utilização de métodos preditivos para reduzir a redundância inter-quadro (imagens dinâmicas) associada ao uso de métodos de transformadas para tratamento das imagens estáticas é uma abordagem bastante utilizada.

11.5.2 Codificação Especial para Regiões com Arestas

Tanto do ponto de vista de redução de redundância estatística de fontes quanto do ponto de vista de utilização de características de visão, o método de transformadas adequa-se muito melhor a regiões sem arestas do que a regiões com arestas.

Uma forma de hibridismo muito promissora consiste na decomposição das imagens em regiões com e sem arestas (dominadas por componentes altas e baixas frequências respectivamente). As regiões sem arestas são codificadas através de métodos de transformadas, enquanto que as outras regiões são codificadas por outros métodos mais propícios para elas.

11.6 Possibilidade de Emprego de Codificação Sem Perdas

Os métodos de transformadas não são muito convenientes para codificação sem perdas.

Para que não haja perdas no processo de codificação é necessário que os coeficientes transformados sejam calculados e codificados com precisão absoluta. O cálculo de cada coeficiente de representação de um bloco $N \times N$ corresponde a um produto escalar de dois vetores de elementos representados com b bits com dimensão N . Portanto, para garantir a precisão absoluta nos cálculos, cada coeficiente transformado tem de ser representado com $3b$ bits, o que aumenta sobremaneira a quantidade de bits necessários para representar cada bloco.

Além deste grande aumento de informação na representação de blocos, a utilização de transformadas para codificação sem perdas foge a um de seus propósitos iniciais, a saber, a busca de um sistema de coordenadas em que as imagens possam ser representadas por um conjunto pequeno de coeficientes caracterizados precisamente e pela maior parte dos coeficientes caracterizados de forma grosseira (poucos bits utilizados para representá-los).

11.7 Complexidade de Implementação

A complexidade de implementação foi um obstáculo importante para a codificação por transformadas inicialmente. Durante a década de 70, por exemplo, a tecnologia digital apresentava grau de integração muito menor do que se vê hoje em dia. Os avanços tecnológicos de lá para cá, no entanto, permitiram que esse obstáculo fosse superado.

Hoje em dia existem diversos componentes (processadores digitais de sinais e circuitos integrados dedicados) capazes de permitir a implementação de métodos de transformadas para codificação de imagens com qualidade de vídeo em tempo real. Os mais importantes padrões de codificação de imagens da atualidade são baseados em métodos de transformadas (JPEG [181], MPEG [62], CCITT H.261 [104]).

Um aspecto importante da complexidade computacional destes métodos é que o codificador e o decodificador apresentam essencialmente a mesma complexidade e as mesmas operações. Desta forma pode-se utilizar o mesmo equipamento (hardware e software) para se fazer tanto a codificação quanto a decodificação. Esta característica permite o barateamento mais rápido dos equipamentos devido à massificação (principalmente dos decodificadores) e a utilização deste método para codificar comunicação entre pares (*peer-to-peer*), isto é, comunicação em que os participantes são ora transmissores (realizando codificação) ora receptores (realizando decodificação).

11.8 Faixa de Compressão Atingida

Os métodos de codificação por transformadas têm dois representantes bastante típicos: os padrões JPEG e MPEG. Estes padrões têm sido bastante utilizados na prática e seus resultados quanto a taxa de compressão são verdadeiros padrões de comparação (*benchmarks*) tanto do ponto de vista prático quanto teórico.

No caso de imagens estáticas, consegue-se codificação de boa qualidade (JPEG) a uma taxa de 0.5 bps (bits por segundo), o que significa uma taxa de compressão de 16:1 (8 bits por pixel é a quantidade de informação original).

Já para imagens dinâmicas (MPEG), a taxa de bits por segundo necessária para se obter imagens com boa qualidade (aproximadamente imagem de vídeo cassetes) é de 1.4 Mbps, o que significa uma taxa de compressão de 128:1 (30 quadros por segundo, cada quadro com 512×512 pixels e 24 bits de cor por pixel \Rightarrow 180Mbps).

Capítulo 12

Codificação em Sub-bandas

12.1 Introdução

Os métodos de compressão através da decomposição em sub-bandas foram utilizados inicialmente no processamento de voz [38]. Posteriormente sua aplicação foi estendida também para o caso de imagens. Eles estão baseados na decomposição do sinal original (voz ou imagens) em sub-bandas que correspondem às componentes do sinal em diferentes faixas de frequências.

Após a decomposição, cada sub-banda do sinal tem seus coeficientes quantizados e finalmente codificados. Portanto a transformação de representação realizada leva as imagens originais aos coeficientes que as descrevem em diversas sub-bandas. O apelo do método está na utilização de quantização diferenciada nas diversas sub-bandas, já que estas são projetadas para corresponderem a bandas de diferentes percepções de ruído da visão humana. Um maior número de níveis de quantização pode ser utilizado para bandas nas frequências de maior sensibilidade de visão, enquanto que as bandas de menor sensibilidade podem ser quantizadas de forma mais grosseira. Esta adequação dos níveis de quantização à variação de sensibilidade da visão é conhecida por *modelamento do ruído* (*noise shaping*).

12.2 Descrição do Método

12.2.1 Decomposição da Imagem em Sub-bandas

A primeira etapa do método consiste na decomposição da imagem inicial em bandas. Cada banda, obtida através da passagem da imagem por um filtro e de sua posterior sub-amostragem, corresponde a uma faixa espectral da imagem. É interessante observar que estas bandas são um meio termo entre o sinal original, descrito interiramente no domínio do espaço (e do tempo para o caso de imagens dinâmicas), e sua representação através de coeficientes transformados, interiramente no domínio da frequência. Cada banda informa as localizações espaciais (e temporais) das componentes da faixa de frequências a que corresponde.

A viabilidade e o projeto dos filtros responsáveis por esta decomposição é um ponto de vital importância para o método de codificação. Estes filtros devem satisfazer a duas condições:

- Discriminação de freqüências: Cada sub-banda resultante da decomposição tem de corresponder a uma faixa espectral do sinal original
- Recuperação sem perdas: O sinal original tem de poder ser recuperado sem perdas a partir das sub-bandas

A sub-amostragem realizada, conhecida por decimação, tem como objetivo manter constante o número de coeficientes utilizados para descrever imagens. Um dos trabalhos inovadores deste método de codificação ([3]) realiza uma decomposição bidimensional de imagens estáticas com sub-amostragem apenas das componentes de baixas freqüências. Assim o número total de coeficientes para representar as imagens aumenta por um fator aproximadamente igual a $\frac{4}{3}$, o que é indesejável.

Na seção (12.3) estão apresentadas duas abordagens para obtenção de filtros com as características descritas acima, também conhecidos como *filtros de reconstrução perfeita* (PR, *perfect-reconstruction*). Para a continuidade da descrição do método é necessário e suficiente citar que tais filtros existem.

Existem várias alternativas de utilização de filtros de reconstrução perfeita. A mais comum baseia-se em um par de filtros unidimensionais com respostas espectrais complementares: um passa-baixas e outro passa-altas. Neste caso a decimação é feita com fator 2, ou seja, após a filtragem considera-se apenas um a cada dois pontos do sinal resultante.

Dentro desta alternativa, quando se deseja decompor sinais com dimensões maiores, (2 para o caso de imagens estáticas e 3 para o caso de imagens dinâmicas), os filtros são compostos formando filtros multi-dimensionais separáveis. Neste caso a decimação é feita com fator 2 para cada dimensão envolvida. No caso de filtragem bi-dimensional, por exemplo, considera-se apenas 1 a cada 4 pontos de cada banda do sinal filtrado pois há duas dimensões envolvidas.

Quando a decomposição requer um número maior de bandas, o filtro é utilizado seguidamente em cascata, isto é, a saída de uma etapa do filtro entra em outra etapa, tal como mostra a figura 12.1.

A decomposição através de filtros assimétricos ([48]) é uma alternativa à abordagem de filtros unidimensionais simétricos. Através dela as larguras de faixas dos filtros passa-baixas e passa-altas são diferentes, correspondendo respectivamente às seguintes frações da largura de faixa do sinal original: $\frac{1}{k_l}$ e $\frac{1}{k_h}$. Neste caso as decimações também têm de ser diferenciadas para as duas componentes: fator k_l para as baixas freqüências e k_h para as altas. Para que estas decimações sejam precisas e, portanto, possa haver a reconstituição perfeita do sinal original, é necessário que os fatores k_l e k_h sejam racionais.

Uma outra decomposição possível é a que divide o sinal inicial em M ($M > 2$) sub-bandas de mesma largura espectral. Cada sub-banda é decimada por um fator M . Esta alternativa não tem sido muito utilizada no caso de decomposição de imagens.

Outra alternativa importante à abordagem inicial é a utilização de filtros não separáveis para a decomposição de sinais multi-dimensionais ([106] e [2]).

Existem diversas formas de decompor um sinal através de cascatas de filtros/decimadores. Uma possibilidade, utilizada por Woods e O'Neil em [196], decompõe imagens em 16 sub-bandas ou canais de mesma largura espectral. Utiliza-se duas camadas de filtros simétricos para realizar esta decomposição, como mostra a figura 12.2.

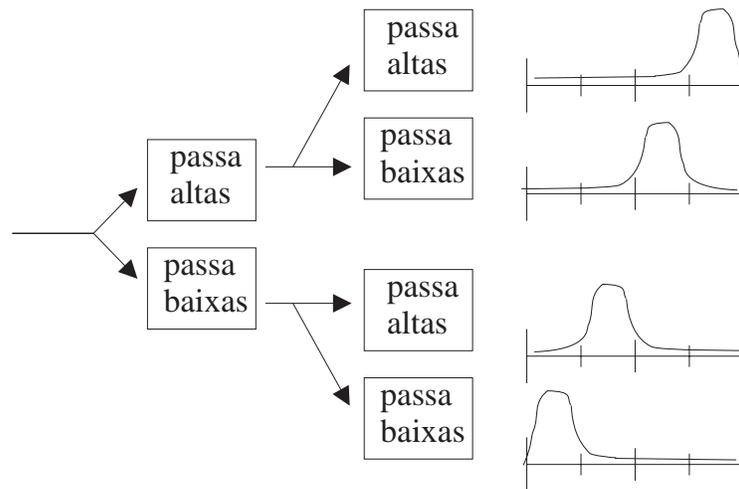


Figura 12.1: Aplicação dos filtros em cascata para decomposição em mais de duas sub-bandas

Outra decomposição em canais, motivada pela teoria de *Análise Multi-Escala*, introduzida por Mallat ([108]), divide os sinais em sub-bandas com larguras iguais em escala logarítmica. Esta divisão, ilustrada na figura 12.3, leva mais em consideração o conhecimento da visão humana, porque nesta a decomposição das imagens é feita através de canais com a mesma largura de de faixa em escala logarítmica.

12.2.2 Quantização dos Coeficientes das Sub-bandas

O próximo passo do método consiste em quantizar e codificar os coeficientes das diversas sub-bandas da imagem. Como a filtragem não tem como objetivo o decorrelacionamento dos coeficientes, e sim a decomposição sem perdas em canais de frequências, os coeficientes resultantes apresentam correlação maior do que os obtidos por transformadas cosseno. Por isto, a quantização vetorial apresenta vantagem considerável sobre a quantização escalar ([7], [188]). De fato, a quantização vetorial tem se consagrado na literatura para aplicação no caso de codificação em sub-bandas.

Os vetores considerados podem ser conjuntos de pontos vizinhos em cada sub-banda ou pontos e conjunto de pontos pertencentes a diferentes bandas. A primeira abordagem ([7]) permite que os livros-códigos sejam projetados de acordo com a variação de sensibilidade da visão ao ruído (maiores detalhes na seção 12.3). A segunda, [188], permite que se explore redundâncias que podem haver entre componentes de diferentes bandas.

12.2.3 Atribuição de Códigos Binários aos Símbolos da Quantização

Tal como no caso de codificação por transformadas, esta etapa corresponde simplesmente à aplicação de algum método de atribuição de bits aos elementos do alfabeto de saída do quantizador.

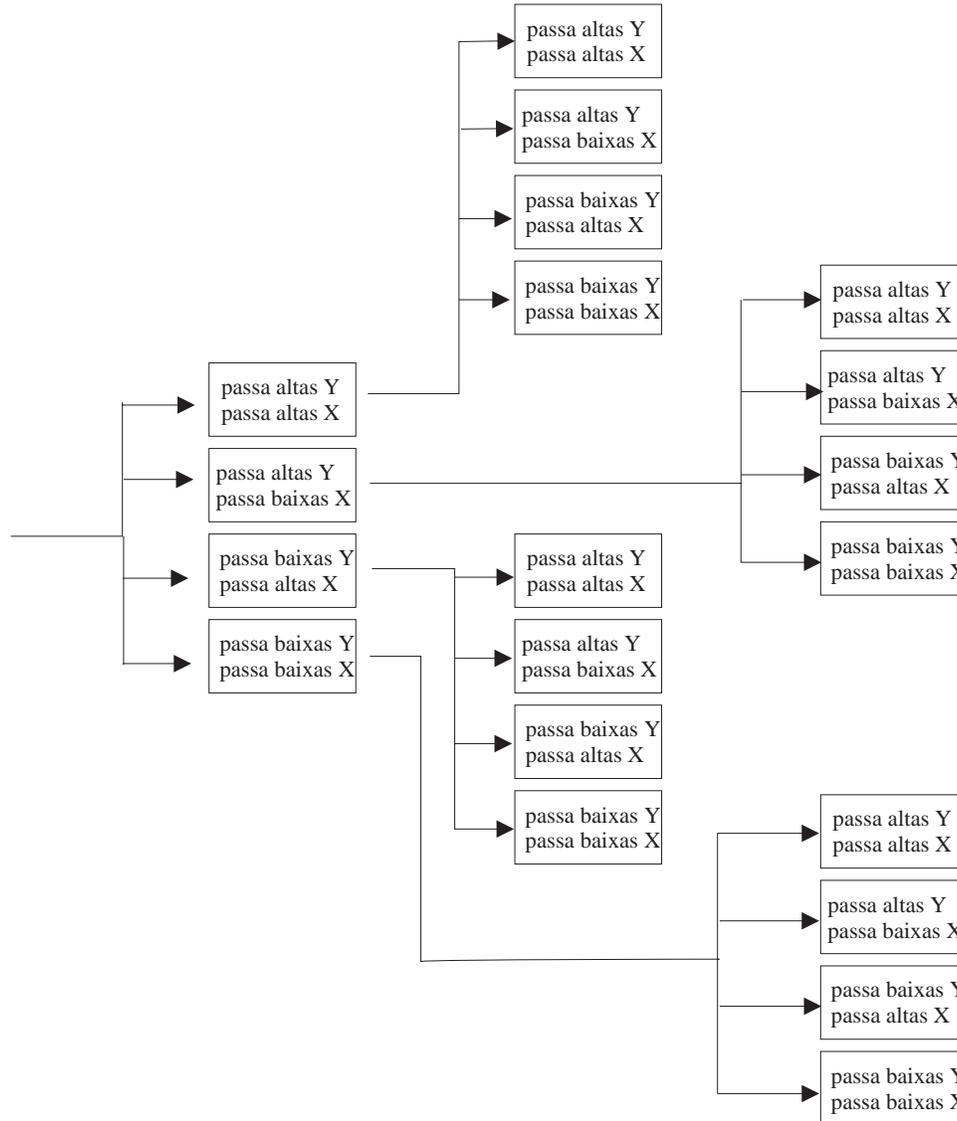


Figura 12.2: Decomposição de imagens em 16 canais de mesma largura espectral

12.3 Base Teórica para a Codificação

12.3.1 Filtros de Reconstrução Perfeita em Dimensão 1

Seja um sinal unidimensional $x(n)$. Deseja-se decompor o sinal através de um banco de M filtros H_0, \dots, H_{M-1} e decimadores acoplados às saídas dos diversos filtros. Os sinais resultantes formam as M sub-bandas ou canais $d_i(m)$, $i = 0, 1, \dots, M \Leftrightarrow 1$. Para recuperar o sinal original faz-se inicialmente o procedimento inverso da decimação: interpolação. Entre cada amostra do sinal $d_i(m)$ insere-se $M \Leftrightarrow 1$ zeros. Em seguida cada componente passa por

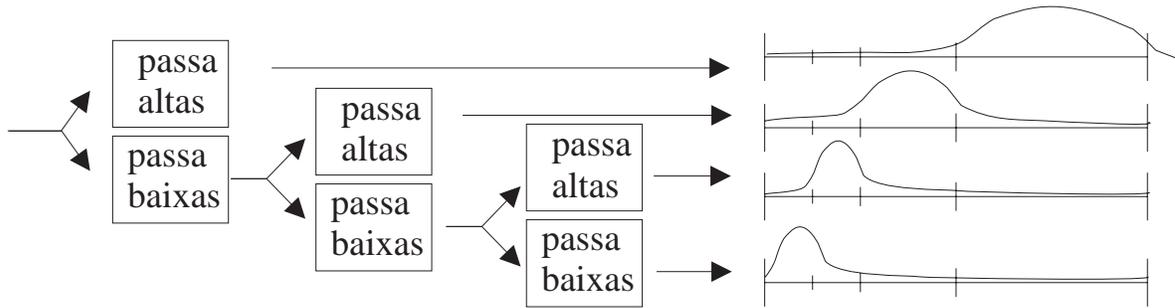


Figura 12.3: Decomposição de imagens em diferentes escalas

um filtro recuperador G_i e as resultantes são somadas para formar o sinal $y(n)$. A figura 12.4 ilustra este processamento.

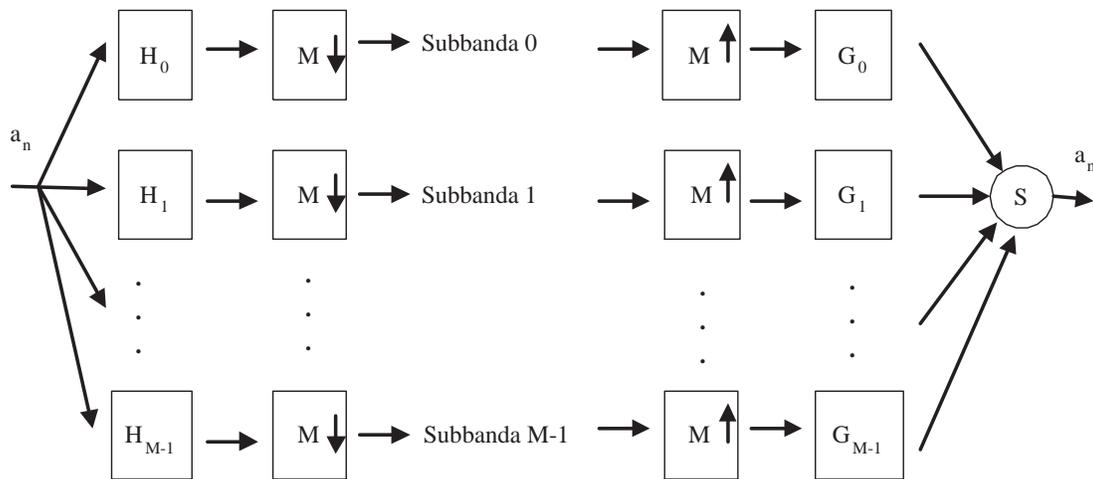


Figura 12.4: Decomposição e reconstrução de um sinal em M filtros

A reconstrução do sinal será perfeita se e somente se $y(n)$ for igual a $x(n)$ a menos de um deslocamento de amostras, isto é, $\exists \Delta n \in Z \mid y(n \leftrightarrow \Delta n) = x(n)$. Para que isto ocorra é preciso impor condições aos filtros H_i e G_j . Com o objetivo de descobrir estas condições os próximos parágrafos estabelecem uma relação entre os sinais $x(n)$ e $y(n)$. A ferramenta básica utilizada é a transformada Z , que leva um sinal de seu domínio original para o domínio das frequências digitais. A relação entre $x(n)$ e $y(n)$ será obtida através de duas etapas:

- determinação dos sinais intermediários $D_i(z)$ em função de $X(z)$
- determinação de $Y(z)$ em função dos sinais $D_i(z)$

Inicialmente define-se a seguinte decomposição dos sinais $x(n)$ e $y(n)$:

$$x_j(n) = \begin{cases} x(n) & \text{se } (n \bmod M) = j \\ 0 & \text{caso contrário} \end{cases} \quad (12.1)$$

$$y_j(n) = \begin{cases} y(n) & \text{se } (n \bmod M) = j \\ 0 & \text{caso contrário} \end{cases} \quad (12.2)$$

Tem-se trivialmente que $x(n) = \sum_{k=0}^{M-1} x_k(n)$ e $y(n) = \sum_{k=0}^{M-1} y_k(n)$.

As transformadas Z dos sinais $x(n)$ e $y(n)$, dadas respectivamente por:

$$X(z) = \sum_{p \in Z} z^{-p} x(p) \quad (12.3)$$

e

$$Y(z) = \sum_{p \in Z} z^{-p} y(p) \quad (12.4)$$

podem ser escritas em função dos sinais $x_j(n)$ e $y_j(n)$:

$$X(z) = \sum_{k=0}^{M-1} z^k X_k(z^M) \quad (12.5)$$

$$Y(z) = \sum_{k=0}^{M-1} z^k Y_k(z^M) \quad (12.6)$$

A transformada Z dos filtros também podem ser decompostas de forma similar:

$$H_i(z) = \sum_{k=0}^{M-1} z^{-k} H_{i,k}(z^M) \quad (12.7)$$

$$G_i(z) = \sum_{k=0}^{M-1} z^k G_{i,k}(z^M) \quad (12.8)$$

onde

$$H_{i,k}(z) = \sum_{j \in Z} z^{-j} h_i(Mj + k) \quad (12.9)$$

$$G_{i,k}(z) = \sum_{j \in Z} z^{-j} g_i(Mj \Leftrightarrow k) \quad (12.10)$$

e os coeficientes h_i e g_i representam respectivamente as respostas impulsiais dos filtros H_i e G_i .

O sinal decimado $d_i(m)$ é o resultado das seguintes operações no domínio das frequências digitais:

$$\begin{aligned} D_i(z) &= \text{Decimação}[H_i(z)X(z)] \\ &= \text{Decimação}[[\sum_{k=0}^{M-1} z^{-k} H_{i,k}(z^M)][\sum_{j=0}^{M-1} z^j X_j(z^M)]] \end{aligned}$$

O que a decimação causa no domínio das freqüências? Seja $b(n)$ um sinal e $b_{\text{dec}}(m)$ sua decimação de fator M , ou seja, $b_{\text{dec}}(m) = b(mM)$. A transformada Z de $b_{\text{dec}}(m)$ é dada por

$$\begin{aligned} B_{\text{dec}}(z) &= \sum_{k \in Z} z^{-k} b_{\text{dec}}(k) \\ &= \sum_{k \in Z} z^{-k} b(kM) \end{aligned} \quad (12.11)$$

A relação entre a transformada Z do sinal decimado e a do original é dada através do seguinte procedimento: no somatório que compõe a transformada original coleta-se os termos que são potências de z^M . Os coeficientes destes termos serão os coeficientes da transformada do sinal decimado, mas suas ordens (potência de z a eles multiplicadas) serão diferentes. Mais especificamente, o coeficiente de ordem kM , que na transformada original multiplica z^{kM} , passa a multiplicar z^k na transformada do sinal decimado. Os demais coeficientes (que multiplicam termos que não são potências de z^M) são desprezados.

Assim, os sinais $d_i(m)$ são descritos por:

$$\begin{aligned} D_i(z) &= \text{Decimação [} \\ &\quad [H_{i,0}(z^M)X_0(z^M) + (zH_{i,0}(z^M)X_1(z^M) + \\ &\quad z^2 H_{i,0}(z^M)X_1(z^M) + \\ &\quad \dots + \\ &\quad z^{M-1} H_{i,0}(z^M)X_{M-1}(z^M))] + \\ &\quad [H_{i,1}(z^M)X_1(z^M) + (zH_{i,1}(z^M)X_0(z^M) + \\ &\quad z^2 H_{i,1}(z^M)X_2(z^M) + \\ &\quad \dots + \\ &\quad z^{M-2} H_{i,1}(z^M)X_{M-2}(z^M))] + \\ &\quad \vdots \\ &\quad [H_{i,M-1}(z^M)X_{M-1}(z^M) + (z^{M-1} H_{i,M-1}(z^M)X_0(z^M) + \\ &\quad z^{M-2} H_{i,M-1}(z^M)X_1(z^M) + \\ &\quad \dots \\ &\quad z H_{i,M-1}(z^M)X_{M-2}(z^M))] \\ &= \sum_{k=0}^{M-1} H_{i,k}(z)X_k(z) \end{aligned} \quad (12.12)$$

pois nenhum termo entre parênteses é potência de z^M . Esta, portanto, é a relação entre $D_i(z)$ e $X(z)$.

A próxima relação a ser obtida é a entre os sinais $Y(z)$ e $D(z)$. Antes, porém, é necessário saber o efeito de uma interpolação no domínio da freqüência. Novamente considere-se um sinal $b(n)$. Ao se interpolar com fator M este sinal obtém-se um novo sinal $b_{\text{int}}(m)$ com a seguinte transformada Z:

$$\begin{aligned} B_{\text{int}}(z) &= \sum_{k \in Z} z^{-k} b_{\text{int}}(k) \\ &= \sum_{j \in Z} z^{-jM} b_{\text{int}}(jM) \end{aligned}$$

$$\begin{aligned}
&= \sum_{j \in Z} z^{-jM} b(j) \\
&= B(z^M)
\end{aligned} \tag{12.13}$$

Logo, $Y(z)$ escreve-se da seguinte forma a partir dos $D_i(z)$:

$$\begin{aligned}
Y(z) &= \sum_{i=0}^{M-1} D_i(z^M) G_i(z) \\
&= \sum_{i=0}^{M-1} D_i(z^M) \sum_{k=0}^{M-1} z^k G_{i,k}(z^M) \\
&= \sum_{k=0}^{M-1} z^k \left[\sum_{i=0}^{M-1} D_i(z^M) G_{i,k}(z^M) \right]
\end{aligned} \tag{12.14}$$

A equação 12.4 estabelece uma decomposição similar para $Y(z)$. Igualando os termos de mesmo grau obtém-se a seguinte relação entre os sinais decimados $D_k(z)$ e os componentes $Y_k(z)$ do sinal $Y(z)$:

$$Y_k(z) = \sum_{i=0}^{M-1} D_k(z) G_{i,k}(z)$$

Para relacionar $X(z)$ e $Y(z)$ convém apresentar as equações 12.12 e 12.14 de forma matricial. Para isto define-se as *matrizes de componentes polifase*, $H_P(z)$ e $G_P(z)$:

$$H_P(z) = \begin{bmatrix} H_{0,0}(z) & H_{0,1}(z) & \dots & H_{0,M-1}(z) \\ H_{1,0}(z) & H_{1,1}(z) & \dots & H_{1,M-1}(z) \\ \vdots & \vdots & \vdots & \vdots \\ H_{1,0}(z) & H_{1,1}(z) & \dots & H_{1,M-1}(z) \end{bmatrix}$$

$$G_P(z) = \begin{bmatrix} G_{0,0}(z) & G_{0,1}(z) & \dots & G_{0,M-1}(z) \\ G_{1,0}(z) & G_{1,1}(z) & \dots & G_{1,M-1}(z) \\ \vdots & \vdots & \vdots & \vdots \\ G_{1,0}(z) & G_{1,1}(z) & \dots & G_{1,M-1}(z) \end{bmatrix}$$

Define-se também a forma matricial de $X(z)$, $Y(z)$ e $D(z)$:

$$X_P(z) = \begin{bmatrix} X_0(z) & X_1(z) & \dots & X_{M-1}(z) \end{bmatrix}^T$$

$$Y_P(z) = \begin{bmatrix} Y_0(z) & Y_1(z) & \dots & Y_{M-1}(z) \end{bmatrix}^T$$

$$D_P(z) = \begin{bmatrix} D_0(z) & D_1(z) & \dots & D_{M-1}(z) \end{bmatrix}^T$$

Segundo as definições acima, as equações 12.12 e 12.14 condensam-se da seguinte forma:

$$D(z) = H_P(z) X_P(z)$$

$$Y_P(z) = G_P^T(z) D(z)$$

e desta forma fica fácil de relacionar os sinais x e y :

$$Y_P(z) = G_P^T(z) H_P(z) X_P(z) \tag{12.15}$$

A equação 12.15 é a principal da teoria de filtros com reconstrução perfeita.

Para que os sinais $y(n)$ e $x(n)$ difiram apenas por um atraso é preciso que o produto $G_P^T(z)H_P(z)$ seja $z^q\mathcal{I}$ para algum $q \in Z_+$, onde \mathcal{I} é a matriz identidade de ordem M .

Para o caso de decomposição em duas componentes ($M = 2$), por exemplo, a reconstrução perfeita requer:

$$\begin{bmatrix} G_{0,0}H_{0,0} + G_{1,0}H_{1,0} & G_{0,0}H_{1,0} + G_{1,0}H_{1,1} \\ G_{1,0}H_{0,0} + G_{1,1}H_{1,0} & G_{1,0}H_{1,0} + G_{1,1}H_{1,1} \end{bmatrix} = z^q \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Além destas condições, por uma questão prática, deseja-se que os filtros H_i e G_i sejam FIR, isto é, filtros de resposta impulsional finita, ou ainda, que possam ser escritos da seguinte forma:

$$H_i[x](n) = \sum_{k=0}^{N_{H_i}-1} h_i(k)x(n \Leftrightarrow k) \quad (12.16)$$

$$G_i[x](n) = \sum_{k=0}^{N_{G_i}-1} g_i(k)x(n \Leftrightarrow k) \quad (12.17)$$

Os maiores índices de coeficientes não nulos, N_{H_i} e N_{G_i} para os casos acima, são chamados de ordens dos filtros H_i e G_i .

O artigo [176] apresenta duas abordagens de projeto de filtros FIR satisfazendo 12.15 e com as características espectrais desejadas. A primeira baseia-se na teoria de circuitos sem perdas ([177]), para os quais o determinante da matriz $H_P(z)$ é um atraso: z^q , $q \in Z$. Exemplos de filtros projetados segundo esta abordagem podem ser encontrados em [178].

Esta primeira abordagem apresenta um problema: apenas para o caso trivial de ordem 2, conhecido como filtro Haar, onde

$$h_{0,0} = h_{0,1} = \sqrt{2}$$

$$h_{1,0} = \Leftrightarrow h_{1,1} = \sqrt{2}$$

os filtros obtidos através dela apresentam linearidade de fase. Todos os demais conjuntos de filtros H_i , G_i projetados através deste método apresentam pelo menos 1 de seus filtros sem linearidade de fase.

A linearidade de fase de um filtro significa fisicamente que um sinal que por ele passe terá atraso de grupo constante, ou seja, caso o sinal tenha suas componentes espectrais concentradas na faixa de passagem do filtro, a resultante após a filtragem será uma versão atrasada e atenuada do sinal original.

No caso de bancos de filtros com reconstrução perfeita, a presença de filtros intermediários sem a linearidade de fase não impede que o processamento como um todo apresente linearidade de fase. Para que isto ocorra, entretanto, é preciso que haja implicitamente uma compensação de fase para ajustar de forma exata as bandas obtidas através dos diversos filtros.

A importância da linearidade de fase nos filtros intermediários H_i e G_i advém do fato que os sinais intermediários $d_i(m)$ são aproximados durante a fase de quantização. Assim, a compensação realizada para contornar a não linearidade de fase de filtros H_i e G_i é executada

sobre aproximações dos sinais $d_i(m)$. Isto compromete a compensação de fase e, conseqüentemente, a linearidade de fase do processo como um todo. Na prática esta característica implica em distorções mais perceptíveis do que as obtidas através da decomposição por bancos de filtros com linearidade de fase ([176]).

O segundo método de projeto de filtros apresentado em [176] é uma variação do primeiro onde não se requer que a matriz $H_P(z)$ seja sem perdas, mas, por outro lado, faz-se a restrição dos filtros intermediários apresentarem linearidade de fase. O artigo [131] apresenta exemplos de filtro projetados através deste método.

Os filtros projetados através destes métodos descritos em [131] e [176] decompõem os sinais em duas componentes ($M = 2$) simétricas. A maior parte dos trabalhos nesta área segue esta linha de decomposição simétrica de sinais. O trabalho [48], por outro lado, propõe a utilização de um par de filtros assimétricos, sendo o filtro passa-altas bem mais estreito que o passa-baixas. O método de obtenção dos coeficientes, descrito no artigo, naturalmente tem de ser diferente dos apresentados em [176].

A decomposição de sinais em mais de 2 componentes ($M > 2$), por sua vez, não tem sido muito utilizada.

12.3.2 Filtros de Reconstrução Perfeita em Dimensões Maiores

Quando se deseja utilizar a decomposição de imagens em sub-bandas é necessário que se utilize filtros de mais de uma dimensão. Caso a decomposição seja feita apenas no espaço, como no caso de imagens estáticas, os filtros utilizados têm de ser bidimensionais. Caso a decomposição leve em consideração também a coordenada temporal, passa a ser necessária a utilização de filtros tridimensionais.

A maneira mais direta de obtenção de filtros multi-dimensionais consiste na composição dos filtros unidimensionais nas diversas dimensões isoladamente. Desta forma obtém-se filtros separáveis. Seja, por exemplo, um conjunto de filtros unidimensionais H_0 , H_1 , G_0 e G_1 , tal como descritos na seção anterior ($M = 2$). Sem perda de generalidade pode-se considerar o filtro H_0 passa-baixas e o H_1 passa-altas. Os filtros de decomposição bidimensional são obtidos da seguinte forma:

- Passa-baixas horizontal e vertical: $H_{(0,0)}(x, y) = H_0(x)H_0(y)$. Este filtro separa as componentes de baixas frequências tanto horizontais quanto verticais.
- Passa-baixas horizontal e passa-altas vertical: $H_{(0,1)}(x, y) = H_0(x)H_1(y)$. Este filtro separa as componentes de baixas frequências horizontais e altas frequências verticais, tais como arestas horizontais.
- Passa-altas horizontal e passa-baixas vertical: $H_{(1,0)}(x, y) = H_1(x)H_0(y)$. Este filtro tem característica complementar ao anterior, ou seja, separa as componentes de altas frequências horizontais e baixas frequências verticais, tais como arestas verticais.
- Passa-altas horizontal e vertical: $H_{(1,1)}(x, y) = H_1(x)H_1(y)$. Este filtro separa as componentes de altas frequências tanto horizontais quanto verticais, tais como arestas nas direções diagonais.

No caso tridimensional a divisão é similar. Normalmente divide-se uma seqüência de imagens em uma componente de altas freqüências temporais, correspondendo às regiões com movimentos, e outra de baixas freqüências temporais, correspondendo às regiões estáticas. Em seguida faz-se a decomposição das sub-bandas temporais em suas respectivas sub-sub-bandas espaciais.

Existe, entretanto, uma diferença importante entre os filtros usados para as coordenadas espaciais e a temporal. Enquanto nas coordenadas espaciais usa-se filtros de ordens pequenas (para evitar custos computacionais elevados), mas maiores que 2 (para se obter filtros mais suaves e com respostas espectrais próximas às dos filtros ideais), na coordenada temporal é comum a utilização dos filtros de Haar, de ordem 2. Embora neste caso de ordem 2 a decomposição espectral das altas e baixas freqüências não seja tão boa, ela é utilizada para evitar a necessidade de armazenamento de muitos quadros antes da codificação de uma seqüência de imagens. De fato, um filtro temporal de ordem N_T requer o armazenamento de N_T quadros para a posterior codificação, o que causa um atraso de até N_T na seqüência de codificação dos quadros. Apesar da simplicidade, o filtro Haar justifica sua presença nestes casos por atuar de forma similar a um preditor temporal. Isto porque a componente de baixas freqüências armazena a média de dois quadros, enquanto que a de altas armazena a diferença entre eles (erro de predição).

A decomposição em sub-bandas pode ser feita através de filtros não separáveis. Alguns trabalhos procuram obter filtros não separáveis que decomponham as imagens ainda mais de acordo com o processamento da visão do que através de filtros separáveis. Dentre estes destacam-se os trabalhos de Mahesh e Pearlman [106] e Simonceli e Adelson [2], baseados em amostragem hexagonal de imagens, e o trabalho de Bamberger e Smith [12], baseado em amostragem convencional (retangular). Uma abordagem interessante, apresentada em [172], utiliza filtro separáveis, mas com as direções principais moldadas às características das imagens sendo codificadas. Mais precisamente, dada uma imagem verifica-se quais as direções que correspondem ao maior número de arestas. Antes da decomposição tradicional em sub-bandas as imagens são re-amostradas nessas direções. Em seguida faz-se o processamento tradicional (com filtros separáveis). O decodificador, evidentemente, precisa realizar a re-amostragem inversa da realizada inicialmente para recuperar a imagem em sua forma original. Com esta decomposição adaptativa Taubman and Zakhor objetivam reduzir a energia das sub-bandas diagonais e, conseqüentemente, diminuir a perda durante a quantização grosseira destas sub-bandas.

12.3.3 Análise Multi-Escala

A análise multi-escala, introduzida por Mallat [108], fornece um caminho alternativo para se obter filtros de reconstrução perfeita. Ela se baseia na decomposição do espaço de funções $L^2(R)$ em sub-espacos, cada qual correspondendo a uma escala ou resolução. Um sinal $x(n)$ é interpretado como a seqüência de componentes da projeção de uma função em um espaço tomado como o inicial. A decomposição do sinal é feita através das projeções da função representada pelo sinal nos sub-espacos de resolução menor que o sub-espaco inicial.

As motivações para o uso da análise multi-escala na filtragem de reconstrução perfeita são as seguintes:

- Simplicidade conceitual da decomposição de sinais em bandas.
- Relacionamento direto entre as projeções nos espaços componentes de $L^2(R)$ e as filtragens dos sinais.
- Embasamento matemático capaz de fornecer classes de filtros recuperáveis com determinada regularidade (ordem do decaimento da resposta espectral do filtro à medida que as frequências crescem).

Para estabelecer matematicamente o conceito de decomposição do espaço de funções $L^2(R)$ em sub-espaços de diferentes escalas definem-se condições que devem ser atendidas por uma família de sub-espaços $\{V_i\}_{i \in Z}$:

$$\dots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \dots \quad (12.18)$$

$$\overline{\cup V_j}_{j \in Z} = L^2(R) \quad (12.19)$$

$$\overline{\cap V_j}_{j \in Z} = \{0\} \quad (12.20)$$

$$f \in V_j \Leftrightarrow r(x) = f(2^j x) \in V_0 \quad (12.21)$$

$$f \in V_0 \Rightarrow s(x) = f(x \Leftrightarrow n) \in V_0 \quad \forall n \in Z \quad (12.22)$$

$$\exists \phi \in V_0 \mid \{\phi_{0,n}\}_{n \in Z} \text{ é uma base ortonormal de } V_0 \quad (12.23)$$

onde $\phi_{m,n}(x) = \phi(2^{-m}x \Leftrightarrow n)$, $m, n \in Z$.

Seja P_j a projeção ortogonal sobre V_j . A primeira e a segunda condições juntas implicam que quanto menor for a escala do espaço V_j melhor será a aproximação de f em V_j , dada por $P_j(f)$. Em particular, $\lim_{j \rightarrow -\infty} P_j(f) = f$. A quarta condição estabelece a natureza multi-escala da família $\{V_i\}_{i \in Z}$.

A última condição define as funções utilizadas na decomposição de uma função f em cada espaço V_j . Na verdade esta condição pode ser um pouco enfraquecida pois o desenvolvimento matemático da análise multi-escala exige apenas que $\{\phi_{0,n}\}_{n \in Z}$ seja uma base de Riesz de V_0 , cuja definição é a seguinte:

Definição 12.1 $\{\phi_{0,n}\}_{n \in Z}$ é uma base de Riesz de V_0 se e só se $\exists A > 0$ e $B < \infty$ tais que $\forall (c_k)_{k \in Z} \in l^2(Z)$

$$A \sum_{k \in Z} |c_k|^2 \leq \left\| \sum_{k \in Z} c_k \phi_{0,k} \right\|^2 \leq B \sum_{k \in Z} |c_k|^2 \quad (12.24)$$

Nesta apresentação a exigência de ortonormalidade será mantida primeiro por causa da simplicidade nas projeções de funções nas bases e também porque tais bases ortonormais existem.

As condições 12.21 e 12.23 garantem que as funções $\{\phi_{j,n}\}_{n \in Z}$ formam uma base ortonormal de $V_j \quad \forall j \in Z$.

Um resultado importante da análise multi-escala relaciona uma família $\{V_i\}_{i \in Z}$ a uma base ortonormal de $L^2(R)$ formada por *wavelets* $\{\psi_{m,n}\}_{m,n \in Z}$. Para entender este resultado é preciso definir o conceito de *wavelets*:

Definição 12.2 *Wavelets são funções $\psi^{a,b}$ geradas a partir de translações e dilatações de uma única função ψ , chamada de wavelet mãe, que satisfaz a seguinte condição:*

$$\int_R \psi(t) dt = 0 \tag{12.25}$$

A definição precisa das wavelets $\psi^{a,b}$ a partir da wavelet mãe é a seguinte:

$$\psi^{a,b}(t) = \sqrt{|a|} \psi\left(\frac{t-b}{a}\right)$$

O resultado importante citado acima estabelece que dada uma família $\{V_i\}_{i \in Z}$ de sub-espacos de $L^2(R)$ satisfazendo as condições 12.18, 12.19, 12.20, 12.21, 12.22 e 12.23, existe uma base ortonormal de $L^2(R)$ formada por *wavelets*

$$\psi^{m,n}(t) = \sqrt{2} \psi(2^{-m}t \Leftrightarrow n)$$

($a = 2$ e $b = 1$) tal que

$$P_{j-1} = P_j + \sum_k \langle f, \psi_{j,k} \rangle \psi_{j,k} \tag{12.26}$$

Seja W_j o espaco gerado pelas *wavelets* definidas acima com j constante $\{\psi_{jn}\}_{n \in Z}$. É interessante observar que os sub-espacos W_j decompõem $L^2(R)$ em soma direta. Com efeito, pelo fato das *wavelets* serem ortonormais os espacos W_j são ortogonais entre si, isto é, $W_i \perp W_j$ se $i \neq j$. Além disto as *wavelets* geram $L^2(R)$, logo:

$$L^2(R) = \bigotimes_{j \in Z} W_j \tag{12.27}$$

A equação 12.26 implica que o sub-espaco gerado pelas *wavelets* $\{\psi^{j,n}\}_{n \in Z}$, denotado por W_j , é o complemento ortogonal de V_j em V_{j-1} :

$$V_{j-1} = W_j \otimes V_j \tag{12.28}$$

Esta decomposição de V_{j-1} em V_j e W_j pode prosseguir de forma recursiva:

$$\begin{aligned} V_{j-1} &= W_j \otimes V_j \\ V_{j-1} &= W_j \otimes W_{j+1} \otimes V_{j+1} \\ V_{j-1} &= W_j \otimes W_{j+1} \otimes W_{j+2} \otimes V_{j+2} \\ V_{j-1} &= W_j \otimes W_{j+1} \otimes \dots \otimes W_{j+N} \otimes V_{j+N} \end{aligned}$$

A figura 12.5 ilustra esta decomposição aplicada a uma função $f \in V_{j-1}$.

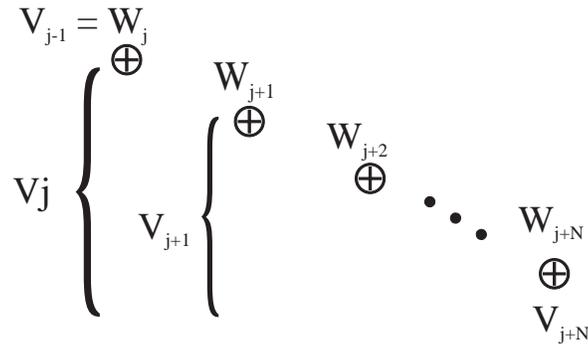


Figura 12.5: Decomposição de V_{j-1} em sub-espços de resolução menor

12.3.4 Conexão entre Filtragem de Recuperação Perfeita e Análise Multi-Escala

A figura 12.5 sugere a conexão entre a decomposição multi-escala e a filtragem. A idéia básica para esta conexão é escolher os V_j tais que as projeções P_j e $(P_{j-1} \Leftrightarrow P_j) = Q_j$ correspondam a filtros decompositores e, desta forma, as melhores aproximações de uma função $f \in V_{j-1}$ nos espaços $W_j, W_{j+1}, \dots, W_{j+N}$ e V_{j+N} correspondam a suas respectivas bandas. Para completar a conexão, entretanto, é preciso resolver algumas questões:

- Como relacionar os sinais com funções nos espaços V_j ?
- Como garantir que o número de coeficientes utilizados para representar um sinal qualquer através de suas projeções em sub-espços não exceda o número de pixels inicial do sinal?
- Como recuperar um sinal a partir das projeções $W_j, W_{j+1}, \dots, W_{j+N}$ e V_{j+N} ?
- Como determinar a resposta espectral das projeções P_j e Q_j ?

O ponto de partida para as respostas é um sinal uni-dimensional $x(n)$. Para relacionar $x(n)$ com uma função do $L^2(\mathbb{R})$ escolhe-se uma escala inicial que, sem perda de generalidade, pode ser V_0 . $x(n)$ corresponde à função dada por

$$\sum_k x(k)\phi_{0k} \tag{12.29}$$

A pesquisa das projeções multi-escala que aproximem o comportamento ideal dos filtros de decomposição está muito bem exposta em [7]. Um pequeno resumo desta exposição está apresentado nos próximos parágrafos para poder responder às próximas questões.

Como é a representação dos elementos das bases ortonormais de V_{j+1} e W_{j+1} na base ortonormal de V_j ? Tanto $\phi_{j+1,0}$ quanto $\psi_{j+1,0}$ podem ser decompostos na base V_j :

$$\begin{aligned} \phi_{j+1,0} &= \sum_n h_n \phi_{j,n} \\ \text{onde } h_n &= \langle \phi_{j+1,0}, \phi_{j,n} \rangle \end{aligned} \tag{12.30}$$

$$\begin{aligned}
&= \langle \phi(2 \Leftrightarrow j \Leftrightarrow 1x), \phi(2 \Leftrightarrow jx \Leftrightarrow n) \rangle \\
&= \langle \phi(x/2), \phi(x \Leftrightarrow n) \rangle
\end{aligned} \tag{12.31}$$

Lembrando da fórmula de dilatação e translação dos elementos $\phi_{m,n}$, a equação 12.30 pode ser escrita somente em função de ϕ :

$$\phi_{j+1,0}(x) = \sqrt{2} \sum_n h_n \phi_{j+1,0}(2x \Leftrightarrow n) \tag{12.32}$$

Esta fórmula é importante porque aplicando-se a ela a transformada de Fourier pode-se estabelecer uma relação entre os espectros de $\phi_{m,n}$ em diferentes dilatações no domínio da frequência:

$$\Phi(\omega) = \frac{1}{\sqrt{2}} \sum_n h_n e^{-i\omega n/2} \Phi(\omega/2) \tag{12.33}$$

Como $\Phi(\omega/2)$ não depende de n no lado esquerdo da equação 12.33, os termos do somatório podem ser agrupados numa função m_0 :

$$m_0(\omega) = \frac{1}{\sqrt{2}} \sum_n h_n e^{-i\omega n} \tag{12.34}$$

Com este agrupamento a equação 12.33 fica re-escrita da seguinte forma:

$$\Phi(\omega) = m_0(\omega/2) \Phi(\omega/2) \tag{12.35}$$

Esta igualdade é importante e significa que a projeção de V_j em V_{j+1} corresponde a uma filtragem passa-baixas que contrai o espectro do sinal filtrado por um fator 2. É importante observar que pela própria construção da função m_0 , sua resposta impulsional é dada pela seqüência $(h_n)_{n \in \mathbb{Z}}$.

O complemento do sinal, isto é, as componentes de altas frequências, é projetado em W_{j+1} . A resposta espectral do filtro passa-altas está relacionada com o de passa-baixas. O artigo [7] apresenta de forma detalhada as contas que estabelecem esta relação. A base destas contas é a decomposição de $\psi_{j+1,0}$ na base ortonormal de V_j .

$$\psi_{j+1,0} = \sum_n (\Leftrightarrow 1)^n h_{-n+1} \phi_{j,n} \tag{12.36}$$

$$= \sum_n g_n \phi_{j,n} \tag{12.37}$$

onde $g_n = (\Leftrightarrow 1)^n h_{-n+1}$. Estes coeficientes $\{g_n\}_n$ formam a resposta impulsional do filtro de passa-altas, que será denotado por $n_0(\omega)$. Sua resposta espectral é:

$$n_0(\omega) = \frac{1}{\sqrt{2}} \sum_n g_n e^{-i\omega n} \tag{12.38}$$

$$= \frac{1}{\sqrt{2}} \sum_n (\Leftrightarrow 1)^n h_{-n+1} e^{-i\omega n} \tag{12.39}$$

A projeção de $x(n)$ em v_1 e W_1 pode ser escrita em termos de h_n e g_n respectivamente:

$$\begin{aligned}
P_1(X) &= P_1 \left[\sum_n x(n) \phi_{0,n} \right] & (12.40) \\
&= \sum_n x(n) P_1 [\phi_{0,n}] \\
&= \sum_n x(n) P_1 [\phi_{0,n}] \\
&= \sum_n x(n) \sum_k \langle \phi_{0,n}, \phi_{1,k} \rangle \phi_{1,k} \\
&= \sum_n x(n) \sum_k \langle \phi_{0,n}, \phi_{1,k} \rangle \phi_{1,k} & (12.41)
\end{aligned}$$

Mas

$$\begin{aligned}
\langle \phi_{0,n}, \phi_{1,k} \rangle &= \langle \phi(x \Leftrightarrow n), \phi(x/2 \Leftrightarrow k) \rangle \\
&= \langle \phi[x \Leftrightarrow (n \Leftrightarrow 2k)], \phi(x/2) \rangle \\
&= \overline{\langle \phi(x/2), \phi[x \Leftrightarrow (n \Leftrightarrow 2k)] \rangle} \\
&= \overline{h_{n-2k}} & (12.42)
\end{aligned}$$

Logo,

$$P_1(X) = \sum_n x(n) \sum_k \overline{h_{n-2k}} \quad (12.43)$$

Contas análogas chegam ao valor da projeção Q_1 :

$$Q_1(X) = \sum_n x(n) \sum_k \overline{g_{n-2k}} \quad (12.44)$$

Estas duas equações implicam que as projeções de um sinal $X \in V_0$ nos espaços V_1 e W_1 são feitas através das convoluções do sinal $x(n)$ com as seqüências $(h_n)_{n \in Z}$ e $(g_n)_{n \in Z}$ respectivamente, sucedidas de decimação. Convolução com a seqüência $(a_n)_{n \in Z}$ no domínio original dos sinais equivale no domínio da freqüência ao produto pelo filtro que tem $(a_n)_{n \in Z}$ como resposta impulsional. Portanto as projeções P_j e Q_j são compostas por uma filtragem (pelo $m_0(\omega)$ passa-baixas para P_j e pelo $n_0(\omega)$ passa-altas para Q_j) e por uma decimação com fator 2. A presença desta decimação responde à segunda questão, ou seja, garante que o número de coeficientes para descrever uma imagem não aumenta quando se projeta um sinal em duas componentes de escala mais grosseira.

A reconstrução dos sinais também pode ser descrita em função dos coeficientes $(h_n)_{n \in Z}$ e $(g_n)_{n \in Z}$. De fato, seja um sinal $X \in V_j$ decomposto em duas parcelas $X_\phi \in V_{j+1}$ e $X_\psi \in W_{j+1}$. Para obter o sinal X novamente basta somar as duas componentes X_ϕ e X_ψ e calcular os coeficientes nas coordenadas da base de V_j :

$$\begin{aligned}
X &= X_\phi + X_\psi & (12.45) \\
&= \sum_m X_\phi(m) \phi_{1,m} + \sum_m X_\psi(m) \psi_{1,m} \\
&= \sum_m X_\phi(m) \sum_k \langle \phi_{1,m}, \phi_{0,k} \rangle \phi_{0,k} + \\
&\quad \sum_m X_\psi(m) \sum_k \langle \psi_{1,m}, \phi_{0,k} \rangle \phi_{0,k}
\end{aligned}$$

$$\begin{aligned}
&= \sum_m X_\phi(m) \sum_k \langle \phi_{1,0}, \phi_{0,k-2m} \rangle \phi_{0,k} + \\
&\quad \sum_m X_\psi(m) \sum_k \langle \psi_{1,0}, \phi_{0,k-2m} \rangle \phi_{0,k} \\
&= \sum_m X_\phi(m) \sum_k h_{k-2m} \phi_{0,k} + \\
&\quad \sum_m X_\psi(m) \sum_k g_{k-2m} \phi_{0,k} \\
&= \sum_k \left[\sum_m (X_\phi(m) h_{k-2m} + X_\psi(m) g_{k-2m}) \right] \phi_{0,k}
\end{aligned}$$

Portanto a reconstrução é feita através da convolução de X_ϕ e X_ψ interpolados (um 0 inserido entre cada amostra) com os filtros com respostas espectrais $(h_n)_{n \in Z}$ e $(g_n)_{n \in Z}$. Isto responde à terceira questão.

Para determinar a resposta espectral dos filtros (quarta questão a ser resolvida para relacionar análise multi-escala e filtragem de reconstrução perfeita) é preciso verificar quais as restrições que a decomposição multi-escala impõem à função m_0 . [39] e [7] apresentam uma formulação que adiciona a estas restrições a necessidade dos filtros terem resposta impulsional finita e uma restrição de regularidade, isto é, convergência de decaimento quando $\omega \rightarrow \infty$. O resultado é uma classe de funções candidatas a serem m_0 para uma decomposição multi-escala. Estes artigos foram além e determinaram condições necessárias e suficientes a serem testadas para verificar se uma função candidata m_0 corresponde a uma decomposição multi-escala. Pode-se escolher dentre as candidatas aquelas com melhores respostas espectrais (mais próximas às respostas dos filtros ideais) e, em seguida, verificar as condições de m_0 “gerar” uma decomposição multi-escala.

Os filtros obtidos através da aplicação do método descrito acima não apresentam linearidade de fase (a menos do filtro trivial de Haar). Isto acontece porque não se consegue sintetizar *wavelets* simétricas através deste método ([7]). *Wavelets* não simétricas implicam em seqüências $\{h_n\}_n$ e $\{g_n\}_n$ não simétricas e, conseqüentemente, filtros sem linearidade de fase.

Esta dificuldade pode ser ultrapassada através de uma variante do método ([7]) em que as funções $\{\phi_{mn}\}_n$ não precisam ser bases ortonormais dos sub-espacos gerados por elas. Basta que sejam bases de Reisz. Filtros bastante adequados têm sido obtidos através desta variante do método ([167]).

Durante toda esta seção trabalhou-se com sinais unidimensionais. O resultado do trabalho é um método de obtenção de filtros de reconstrução perfeita com duas componentes ($M = 2$) simétricas. Este resultado pode ser estendido para maiores dimensões de forma simples através da composição dos filtros unidimensionais em cada uma das coordenadas, tal como exposto na seção 12.3.2. Os filtros multi-dimensionais assim obtidos são separáveis.

12.3.5 Características da Visão Humana Exploradas

A principal característica da visão humana utilizada para a codificação em sub-bandas é a variação de sensibilidade a ruído nas diferentes sub-bandas das imagens. Uma vez realizada a decomposição em sub-bandas, cada banda pode ser quantizada de acordo com a sensibilidade a ruído da visão na faixa de frequência em que se encontra.

Outra característica importante é a variação de percepção do ruído em diferentes orientações. A visão é mais sensível a ruído nas direções horizontal e vertical e menos nas diagonais. Desta forma, as sub-bandas correspondentes a faixas de frequências orientadas nas diagonais podem ser quantizadas de forma mais grosseira que as bandas horizontais e verticais.

A literatura denomina esta adequação da precisão da quantização às diversas sub-bandas como modelamento do ruído (*noise shaping*).

12.4 Possibilidade de Emprego de Adaptabilidade

12.4.1 Ajuste de Direções de Filtragem Adaptativa

O trabalho de Egger ([49]), tal como citado anteriormente, propõe um ajuste dos filtros de decomposição das imagens às características das imagens. O primeiro passo do ajuste consiste na determinação das direções nas quais há maiores componentes de frequências (normalmente maior quantidade de arestas). Em seguida faz-se uma re-amostragem da imagem em um reticulado formado por estas direções. Após esta etapa a codificação é feita normalmente (filtragem, decimação, quantização, interpolação e filtragem). Finalmente faz-se a re-amostragem inversa da imagem.

12.4.2 Quantização Vetorial Adaptativa

Os livros-códigos utilizados na quantização vetorial são obtidos através da aplicação do método de LBG a um conjunto de imagens de treinamento. À medida com que novas imagens sejam codificadas os livros-códigos podem variar para se adequarem às características dessas novas imagens.

12.4.3 Filtragem em Sub-Bandas Adaptativa

O filtro de decomposição de imagens em sub-bandas pode ser projetado de forma a otimizar a relação entre a compressão de informação e a distorção da codificação ([150] e [151]). Para isto define-se inicialmente uma função custo $J = D + \lambda R$ (onde D e R são as funções que associam a cada filtro escolhido a distorção e a quantidade de informação médias verificadas na codificação de um conjunto de imagens). A minimização corresponde à otimização da relação entre compressão e distorção.

Esta otimização pode ser feita considerando como conjunto de imagens no qual J (e implicitamente D e R) deve ser minimizada um conjunto representativo do espaço de imagens a serem codificadas. Neste caso o único filtro escolhido é utilizado para decompor blocos de todas as imagens a serem codificadas ([150]).

Outra alternativa, mais flexível porém mais custosa computacionalmente, consiste na escolha do filtro ótimo para decompor cada bloco de imagem ([151]).

12.5 Possibilidade de Emprego de Métodos Híbridos

12.5.1 Codificação Preditiva na Coordenada Tempo

A maior parte das aplicações do método de codificação em sub-bandas ou canais obtêm os canais através de filtragem espacial bi-dimensional. No caso de imagens dinâmicas, métodos preditivos podem ser utilizados para explorar a redundância temporal, enquanto se usa a decomposição em sub-bandas para codificação intra-quadro (estática).

12.5.2 Codificação Preditiva Entre Diferentes Escalas

No trabalho [155] os autores utilizam a sub-banda de resolução m para prever a sub-banda de resolução $m \Leftrightarrow 1$. Esta idéia é interessante para remover a redundância estatística que há entre coeficientes de diferentes sub-bandas, já que as sub-bandas costumam ser quantizadas isoladamente.

12.5.3 Codificação Preditiva dos Livros-Códigos

Outra possibilidade do emprego de hibridismo é a utilização de métodos preditivos para a codificação dos livros-códigos (reabastecimento condicional, por exemplo). Assim, pode-se reduzir a quantidade de informação para a codificação através da redução da redundância existente entre conjuntos de imagens.

12.6 Possibilidade de Emprego de Codificação sem Perdas

Este método é bastante propício à codificação sem perdas. Isto porque o processo de decomposição das imagens em sub-bandas não redundam necessariamente em perdas. Em verdade as perdas que podem existir decorrem das aproximações de operações reais com aritmética de precisão finita. Caso a decomposição em sub-bandas seja feita através de aritmética inteira, como proposto em [160], a representação de uma imagem por suas sub-bandas é absolutamente precisa, sem perdas. É importante, porém, lembrar que a codificação dos coeficientes que caracterizam as sub-bandas também têm de ser feita sem perdas. Para isto a quantização dos coeficientes tem de ser exata, isto é, sem aproximações.

A compressão de dados neste caso é originada exclusivamente pela redução de redundância estatística entre os coeficientes.

Um aspecto interessante da decomposição em sub-bandas para o caso de codificação sem perdas é a possibilidade de utilização de codificação progressiva. Para caracterizar este tipo de codificação seja $\mathcal{C}_N = (c_0, c_1, \dots, c_{N-1})$ a seqüência de N coeficientes que descrevem uma imagem I . Para cada inteiro $k < N$ define-se a aproximação de \mathcal{C}_N considerando apenas os k primeiros coeficientes de \mathcal{C}_N :

$$\mathcal{C}_k = (c_0, c_1, \dots, c_{k-1}, 0, \dots, 0)$$

A decodificação de \mathcal{C}_k define uma imagem I_k , que é a reprodução intermediária de I . Uma codificação é dita progressiva se existir um conjunto de imagens intermediárias

$\{I_{k_0}, I_{k_1}, \dots, I_{k_{M-1}}\}$ tais que I_{k_n} seja uma aproximação da I em todas as regiões da imagem. Além disto as aproximações tem de convergir para I à medida em que k_n cresce.

A motivação para a codificação progressiva é a possibilidade de apresentações abreviadas de imagens através de suas reproduções intermediárias de I_{k_n} . Uma aplicação imediata é a consulta de imagens em banco de dados. Um usuário pode selecionar as imagens que deseja visualizar através de uma consulta abreviada e posteriormente obter a apresentação completa das imagens. Para isto um nível de reprodução intermediária é escolhido para apresentação das aproximações a serem selecionadas. Uma vez selecionada, uma imagem é finalmente apresentada com maior nível de detalhes.

No caso de codificação em sub-bandas cada banda corresponde à representação da imagem original numa escala (ou resolução). Seja I_{q_j} a reprodução intermediária correspondente à resolução da imagem sendo conhecidas as j primeiras sub-bandas da imagem. A seqüência de reproduções $\{I_{q_j}\}$ satisfaz às condições impostas nos parágrafos anteriores e, portanto, a codificação em sub-bandas pode ser feita progressiva. Para isto basta que os coeficientes de cada sub-banda sejam agrupados na N -upla $(c_0, c_1, \dots, c_{N-1})$ que representa a imagem codificada.

12.7 Complexidade de implementação

Estes métodos apresentam complexidade um pouco maior do que a dos métodos baseados em transformadas, sendo, portanto, bem mais complexos do que os métodos preditivos.

As filtragens apresentam aproximadamente a mesma complexidade computacional das transformadas, cabendo à quantização vetorial a razão destes métodos serem um pouco mais caros computacionalmente que os métodos por transformadas.

12.8 Faixa de Compressão Atingida

Este método é objeto de intensa pesquisa atualmente. Existem várias novas abordagens sendo trabalhadas e propostas. Ainda não se pode dizer que exista uma faixa de compressão típica para eles. Nos trabalhos recentes ([7], [188]) têm-se obtido razoável qualidade de imagens monocromáticas a menos de 0,5 bpp (bits por pixel). Existe uma boa expectativa quanto a estes métodos devido

- aos avanços obtidos recentemente (este método de codificação foi proposto a apenas 10 anos ([196]))
- ao foco que tem sido recebido nas pesquisas e
- à possibilidade de utilização de todo o ferramental já desenvolvido para a codificação que ainda não foi totalmente explorado (quantização vetorial, diversos tipos de predição, transformadas para codificar componentes de baixas frequências).

Espera-se que estes métodos atinjam taxas de compressão mais altas que os métodos baseados em transformadas (codificação intra-quadro) e predição de movimento (codificação inter-

quadro). De fato já existem trabalhos onde consegue-se boa qualidade de imagem com entre 0.15 e 0.36 bpps ([155], [172]).

Capítulo 13

Codificação Através de Fractais

13.1 Introdução

A codificação de itrimagens através de fractais passou a receber grande atenção a partir dos artigos publicados por seu precursor Barnsley ([15]). As taxas de compressão apregoadas (1000:1) eram muito maiores que as atingidas pelos outros métodos sendo utilizados (aproximadamente 10:1 para métodos de transformadas) ou pesquisados (não mais que 70:1, para métodos da chamada segunda geração de codificação de imagens).

Outro chamariz de atenção foi o fato dos artigos iniciais não terem descrito completamente os métodos de codificação ([15], por exemplo). Alguns anos passaram até que em 1992 tornou-se público o tão esperado método de codificação via fractais, inicialmente através do artigo [79] de Jacquin, e posteriormente também por meio de outras publicações, como por exemplo [55] (1992) e [56] (1995).

As bases do método são a modelagem da redundância das imagens através de auto-similaridades e da representação das imagens através de pontos fixos de contrações. Embora as taxas de compressão obtidas não sejam tão fantásticas quanto o que se imaginava inicialmente, o método apresenta bons resultados, comparáveis com os métodos de codificação por transformadas.

13.2 Descrição do Método

O método consiste, de forma simplificada, na representação de uma imagem I_k por uma contração f_k tal que o ponto fixo de f_k aproxima I_k . A transformação de representação neste caso leva as intensidades luminosas dos pontos da imagem nos coeficientes que descrevem a contração f_k . Na próxima sub-seção os conceitos de distância entre imagens e contração estão definidos.

O processo de codificação requer a resolução do “problema inverso da teoria das transformadas iteradas”:

- Dada uma imagem qualquer I_k , determinar uma contração f_k tal que:
 - seu ponto fixo aproxima I_k

– a descrição de f_k requer menor quantidade de informação que a descrição de I_k

Sem que alguma simplificação seja feita este problema é de resolução bastante difícil. Na prática várias simplificações, descritas nas seções subseqüentes, são utilizadas.

A decodificação é um processo simples. Para que ela seja feita basta considerar qualquer imagem I_0 como ponto de partida e iterar a contração até que $f_k^{(n)}(I_0)$ seja próxima de $f_k^{(n+1)}(I_0)$.

13.2.1 Contrações no Espaço das Imagens

Antes de se prosseguir na descrição do método de compressão convém apresentar os conceitos de distâncias entre imagens e regiões de imagens e o de contrações no espaço das imagens.

O conceito de distância entre regiões de imagem e imagens é de grande importância para esta codificação. Primeiramente porque a distância fornece a medida da aproximação entre regiões auto-similares. Além disto para que se defina as contrações é preciso antes definir uma distância entre imagens.

As distâncias perceptuais, isto é, que medem a aproximação entre imagens de acordo com a percepção da visão, são as ideais para serem utilizadas. Entretanto não se conseguiu quantificá-las até hoje. Outras alternativas devem ser usadas. Dentre as distâncias práticas a mais utilizada é a do erro médio quadrático.

Considere-se imagens formadas por $M \times N$ pixels (M linhas e N colunas). Seja um conjunto de pixels formando uma região R . A distância entre as regiões R de duas imagens I_1 e I_2 é dada por:

$$\frac{1}{\#\mathcal{R}} \sum_{p \in \mathcal{R}} (I_1(p) \leftrightarrow I_2(p))^2$$

onde $\#\mathcal{R}$ é o número de elementos de \mathcal{R} , $I_1(p)$ e $I_2(p)$ são respectivamente as intensidades luminosas das imagens I_1 e I_2 no ponto $p \in R$.

Esta distância entre regiões pode ser estendida de forma simples para imagens completas:

$$\text{dist}(I_1, I_2) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I_1(i, j) \leftrightarrow I_2(i, j))^2 \quad (13.1)$$

Uma *contração* em um espaço métrico X com distância d é uma função $f : X \rightarrow X$ tal que $\exists s < 1 \mid \forall x, y \in X, d(f(x), f(y)) < d(x, y)$. O menor s para que vale a regra acima recebe o nome de fator de contração de f . Para o caso de imagens X é o conjunto de matrizes $M \times N$ com valores inteiros entre 0 e 2^k e a distância é dada pela equação 13.1.

13.2.2 Contrações Compostas por Transformações Afins Locais

A primeira simplificação usada para reduzir a complexidade do “problema inverso da teoria das transformadas iteradas” restringe o universo das contrações candidatas a terem pontos fixos aproximando a imagem original. Estas têm de ser compostas por “transformações afins locais”.

A idéia desta composição consiste em dividir uma imagem em regiões, chamadas células imagens $\{R_i\}_i$, tais que para cada uma destas regiões haja uma região maior D_i (ainda

pertencente à imagem), chamada região domínio, e uma transformada afim contrativa w_i tais que $w_i(D_i)$ se aproxime de R_i . A contração final é a composição das transformações afins nas diversas células domínio. (**OBS:** embora a grande maioria das aplicações de codificação por fractais utilize transformações afins para mapear as células domínio $\{D_i\}_i$ nas células imagem $\{R_i\}_i$, pode-se também utilizar classes mais amplas de funções, como por exemplo mapeamentos conformes [140], para este fim).

As principais motivações para esta restrição de contrações são: 1) elas adequam-se ao modelo de imagens segundo o qual regiões com auto-similaridades são muito comuns e 2) as transformações afins são simples de caracterizar e preservam as características das regiões domínio (paralelogramas são transformados em paralelogramas). A simplicidade na caracterização das transformações, dos domínios e das células imagens é importante porque através dela requer-se uma menor quantidade de informação para descrever os novos parâmetros de representação das imagens, isto é, as transformações locais.

Uma vez descrita a idéia básica do universo de contrações a ser considerado, bem como as motivações para a escolha deste universo, resta apresentar os detalhes necessários para formalizar a idéia, ou seja, definir de forma precisa as funções contrativas compostas por transformações afins locais.

Seja X o espaço métrico das imagens (distância d). Seja F o conjunto das funções compostas por transformações afins contrativas locais.

Definição 13.1 *Uma função $f \in F$ é caracterizada por:*

- *Uma decomposição celular dos pixels das imagens, isto é, de conjuntos de pixels $\{R_i\}_{i \in \Lambda}$ formando as regiões imagem tais que*

$$R_i \cap R_j = \emptyset \quad i \neq j \quad (13.2)$$

$$I = \bigcap_{i \in \Lambda} R_i \quad (13.3)$$

onde I é o reticulado $M \times N$ formando o conjunto total dos pixels.

- *Um conjunto de pares $\{(D_i, w_i)\}_{i \in \Lambda}$ sendo D_i uma região de pixels (região domínio) e w_i uma transformação afim contrativa que mapeia D_i em R_i . Uma restrição adicional é feita às w_i : estas têm de ser compostas por duas transformações independentes, uma geométrica (g_i) e outra de intensidade luminosa (l_i).*

A figura figura 13.1 ilustra os componentes da caracterização de f .

O mapeamento de D_i em R_i através de w_i pode ser entendido melhor de forma contínua que discreta. Para isto uma imagem deve ser considerada um gráfico de intensidade luminosa definido em um retângulo (normalmente quadrado $[0, 1] \times [0, 1]$). Neste contexto os $M \times N$ pixels da imagem são amostras deste gráfico. As regiões R_i e D_i pertencem ao quadrado $[0, 1] \times [0, 1]$. A transformação geométrica g_i transforma os pontos de D_i em pontos de R_i , enquanto que a de intensidade luminosa determina como as intensidades luminosas dos pontos de R_i são transformadas para se obter $w_i(D_i)$. A figura figura 13.2 resume esta transformação de D_i em $g_i(D_i)$.

Desta forma, $w_i(D_i)$ é um gráfico de intensidade luminosa restrito à região $g_i(D_i) \subseteq R_i$. Na prática restringe-se ainda mais o modelo e considera-se células e transformações tais que

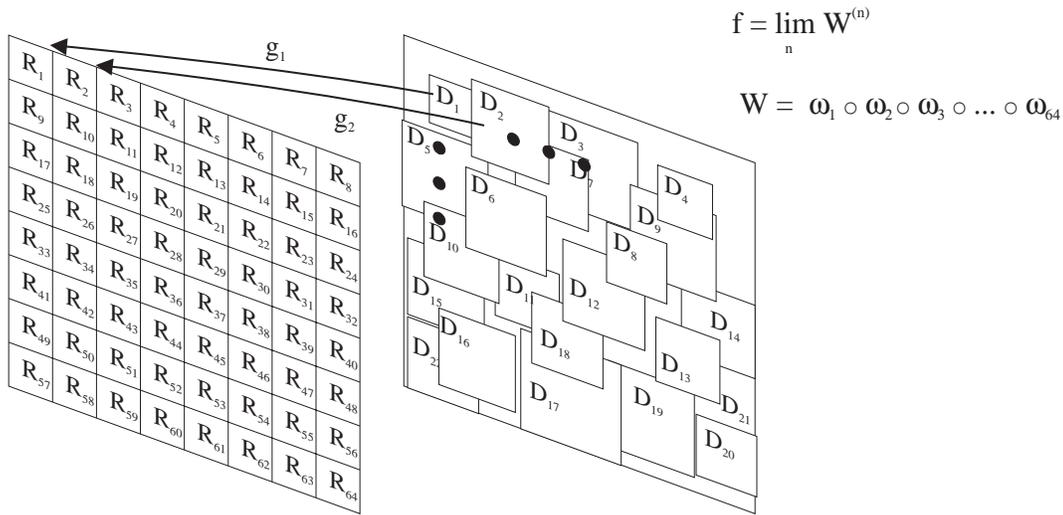


Figura 13.1: Caracterização de uma função $f \in F$

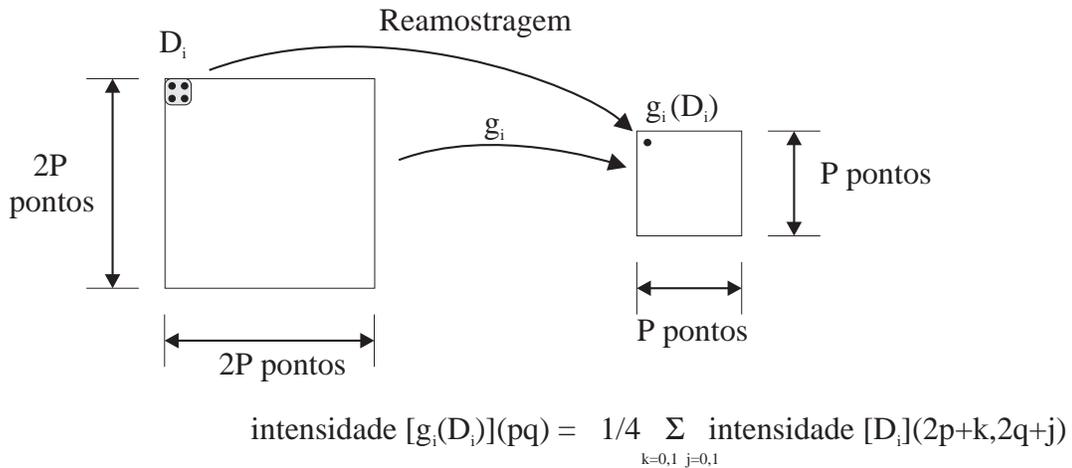


Figura 13.2: Transformação de uma região domínio D_i em $w_i(D_i)$

$g_i(D_i) = R_i$. Portanto R_i e $w_i(D_i)$ são dois gráficos definidos na mesma região. A distância entre estes dois gráficos pode ser calculada de forma similar ao cálculo de distância entre regiões de imagens:

$$d_{\text{gráficos}}(R_i, w_i(D_i)) = \int_{R_i} ([R_i](x) \leftrightarrow [w_i(D_i)](x))^2 dx \tag{13.4}$$

onde $[R_i](x)$ é a intensidade luminosa de R_i no ponto $x \in R_i$ e, analogamente onde $[w_i(D_i)](x)$ é a intensidade luminosa de $w_i(D_i)$ no mesmo ponto.

Voltando ao caso discreto (imagens digitais), o fato de $g_i(D_i) = R_i$ implica que $g_i(D_i)$ e R_i têm os mesmos pixels. O cálculo da distância entre $w_i(D_i)$ e R_i pode ser feito, portanto, da seguinte forma:

$$d_{\text{regiões de imagens}}(R_i, w_i(D_i)) = \sum_{p \in R_i} ([R_i](p) \Leftrightarrow [w_i(D_i)](p))^2 \quad (13.5)$$

onde p é um pixel de R_i . Um pré-requisito importante para este cálculo é a determinação das intensidades luminosas de $w_i(D_i)$ nos pixels de R_i . Esta determinação não é trivial porque os pixels de D_i não são transformados em pixels de R_i através de g_i pois esta transformação tem de ser contrativa. Para resolver este problema faz-se uma re-amostragem de D_i nos pontos que formam a imagem inversa dos pixels de R_i segundo g_i , ou seja, nos pontos $\{g_i^{-1}(p)\}_p$ pixel de R_i .

O processo de re-amostragem envolve vários detalhes técnicos que estão bem apresentados em [193], [34] e [68]. No caso das transformações afins g_i utilizadas para fractais o problema fica simplificado por causa de restrições feitas aos formatos das células domínio e imagem. Por uma questão de ordem dos assuntos estas restrições estão apresentadas na próxima seção. Por ora basta saber que o problema de re-amostragem tem solução.

Assim, a distância entre $w_i(D_i)$ e R_i pode ser calculada através da equação 13.5.

O cálculo de f em uma imagem consiste na soma das intensidades dos diversos sub-gráficos $w_i(D_i)$. Esta soma fica simplificada uma vez que $g_i(D_i) = R_i$, $R_i \cap R_j = \emptyset$ se $i \neq j$ e $I = \bigcap_{i \in \Lambda} R_i$, pois cada pixel de I pertence a uma e somente uma célula R_i , que, por sinal, é influenciada por uma e apenas uma célula domínio D_i . Portanto, para cada pixel p existe apenas uma intensidade $[w_i(D_i)](p)$ não nula, correspondendo à $l_{i_p}([D_{i_p}](g_{i_p}^{-1}(p)))$, onde $p \in R_{i_p}$. Logo, $f(p) = l_{i_p}([D_{i_p}](g_{i_p}^{-1}(p)))$.

Com esta restrição feita sobre as contrações o “problema inverso da teoria das transformadas iteradas” fica reduzido à busca de uma coleção de trincas $\{(R_i, D_i, w_i)\}_{i \in \Lambda}$ com as características descritas na definição 13.1.

13.2.3 Restrições nos Formatos e Tamanhos de Células Imagem e Domínio

Em princípio o formato e o tamanho das células imagem e domínio têm apenas as restrições implícitas no fato de D_i ser transformado em R_i através de uma transformação afim contrativa:

- D_i tem de ter área maior que R_i
- D_i e R_i tem de ser semelhantes pois paralelogramos são mapeados em paralelogramos através de g_i

Com o objetivo de reduzir a complexidade do “problema inverso da teoria das transformadas iteradas” restrições adicionais costumam ser feitas nos tamanhos e formatos das células. Há várias abordagens (bastante semelhantes entre si) para estas restrições adicionais. Os formatos mais utilizados para as células são quadrados, retângulos e triângulos. Os tamanhos podem ser fixos ou variarem de acordo com as imagens a serem codificadas (decomposição celular adaptativa). Diversas diferentes abordagens estão descritas abaixo. Estas abordagens

incorrem em simplificações que podem parecer muito restritivas (de fato algumas delas são mesmo, conforme será descrito). Não se deve deixar de lembrar, entretanto, que a simplificação do problema inverso é crucial para a viabilidade do processo de codificação. Isto porque o problema de achar uma coleção de trios $\{(R_i, D_i, w_i)\}_{i \in \Lambda}$ com as características descritas na seção anterior é extremamente caro computacionalmente pois exige uma busca no conjunto de todas as coleções de trios, cuja cardinalidade é enorme. Portanto as simplificações não são apenas justificáveis, mas também necessárias.

- Células imagem e domínio quadradas de tamanho fixo: Neste caso os pixels de imagens são divididos em um reticulado de quadrados com tamanho fixo, $B \times B$ pixels. Para que as transformações afins sejam contrativas os quadrados candidatos a serem células domínio têm de ter tamanho maior que B . Usa-se normalmente quadrados $2B \times 2B$ para este propósito. A vantagem deste valor está na simples resolução do problema de re-amostragem. Neste caso as transformações g_i reduzem cada dimensão de D_i por um fator de 2. Isto significa que D_i deve ser sub-amostrada em cada dimensão por um fator de 2, ou seja, os valores de intensidade luminosa de D_i na nova amostragem são a média aritmética das intensidades em 4 pixels vizinhos, como mostra a figura 13.3. Outra restrição que costuma ser utilizada refere-se ao posicionamento dos domínios.

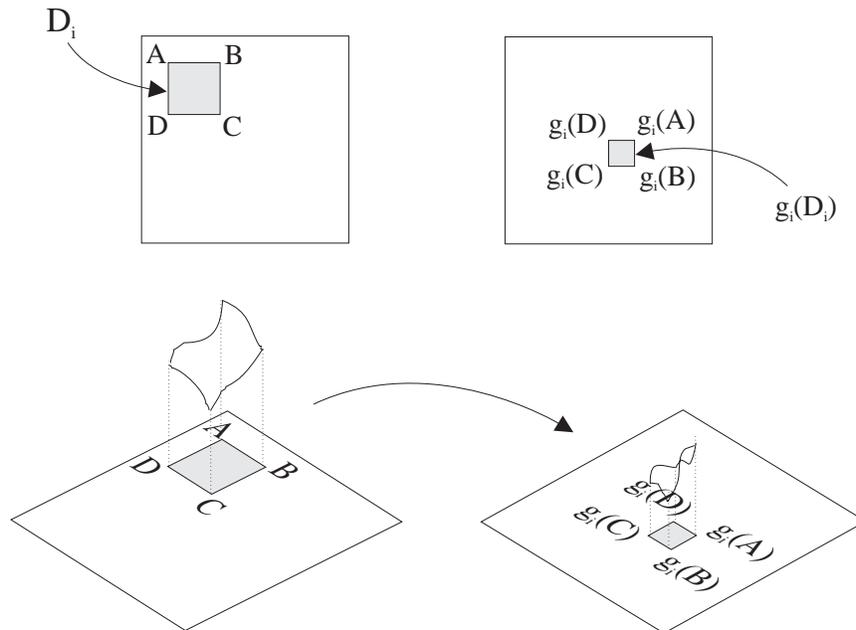


Figura 13.3: Re-amostragem no caso de células domínio 2 vezes maiores que as células imagem em cada dimensão

Nem todos os quadrados de pixels $2B \times 2B$ são candidatos a serem células domínio. Os candidatos restringem-se aos quadrados cujos pixels superiores esquerdos pertencem a um reticulado $A \times A$, onde A vale tipicamente B ou $B/2$. As restrições simplificam

não apenas o processo de busca do melhor conjunto de trios $\{(R_i, D_i, w_i)\}_{i \in \Lambda}$, como também a descrição dos próprios trios. O número de células imagem é bem determinado: $((M/B) \times (N/B))$, assim como a ordenação destas células (varredura progressiva). Portanto para descrever o conjunto de trios basta concatenar as descrições das duplas $\{(D_i, w_i)\}_{i \in \Lambda}$ de acordo com a ordenação induzida pela varredura progressiva. As transformações geométricas g_i são caracterizadas por uma translação e uma transformação unitária (rotação e/ou simetria em relação a algum eixo dos quadrados). As translações são caracterizadas de forma única pelas posições dos D_i . De acordo com a restrição de seu posicionamento, estas células domínio podem ter seus pontos superiores esquerdos numa malha $B \times B$ ou $(B/2) \times (B/2)$. Portanto há $(M/B) \times (N/B)$ ou $4(M/B) \times (N/B)$ possíveis posições para os D_i 's, o que significa que são necessários $\lceil \log_2((M/B) \times (N/B)) \rceil$ ou $\lceil \log_2(4(M/B) \times (N/B)) \rceil$ bits para caracterizar estes conjuntos. Substituindo M , N e B por valores típicos (respectivamente 512, 512 e 8) nas expressões acima verifica-se que entre 12 e 14 bits para representar os D_i 's e, conseqüentemente a componente de translação das transformações g_i . Já as transformações unitárias existem em 8 diferentes possibilidades (4 rotações e 4 rotações com simetria), o que significa que uma transformação específica pode ser caracterizada por 3 bits. Para completar a caracterização do par (D_i, w_i) resta determinar a transformação de intensidade luminosa, que costuma ser representada por 8 bits. Logo cada célula necessita tipicamente entre 23 e 25 bits para ser representada. Considerando que uma célula é composta por $B \times B$ pixels com 8 bits cada a taxa compressão obtida é de aproximadamente $\frac{8B \times B}{25} = 20.4$.

Esta abordagem é muito restritiva principalmente porque não permite nenhum tipo de adaptação das malhas de células imagem e domínio às características das imagens. O parâmetro B é determinado de ante-mão e é fixo para todas as imagens. Ele não pode ser grande demais porque senão muitas células imagens serão aproximadas de forma muito grosseira, nem pode ser pequeno demais porque senão a quantidade de informação necessária para descrever as transformações será muito grande.

- Células imagem e domínio quadradas de 2 possíveis tamanhos: O processo de codificação neste caso é similar a o do item anterior, com a diferença que os tamanhos dos quadrados têm uma pequena flexibilidade para se adaptarem às características das imagens sendo codificadas. Os pixels de imagens são inicialmente divididos no mesmo reticulado de quadrados com tamanho fixo do item anterior. Para cada célula imagem R_i faz-se o mesmo processo de busca do melhor par (D_i, w_i) para aproximar R_i . A diferença entre os métodos reside no fato que se a distância entre a imagem original definida em R_i e a imagem obtida por $w_i(D_i)$ superar um limite pré-estabelecido esta célula não será determinada através do par (D_i, w_i) . Ao invés disto a célula é dividida em 4 células menores $((B/2) \times (B/2))$ e o processo de busca dos pares (D_i, w_i) é feito para cada uma destas células menores. Esta flexibilidade adicional aumenta a carga computacional do método de codificação visto que o processo de busca pode vir a ser feito em um número maior de células. Além disto a busca realizada para células $(B/2) \times (B/2)$ é mais cara computacionalmente que a busca para células $B \times B$ pois existem 2 vezes mais possíveis posições para os D_i 's em cada coordenada. Outro aspecto

importante desta flexibilidade no tamanho das células é que as representações dos pares (D_i, w_i) exigem uma informação adicional que é o tipo de célula sendo codificada ($B \times B$ ou $(B/2) \times (B/2)$). Com isto a taxa de compressão reduz-se e esta redução varia com o tipo de imagem sendo codificada. Isto porque imagens em que muitas células R_i 's precisam ser divididas para serem codificadas precisarão de mais pares (D_i, w_i) para serem representadas. Desta forma a taxa de compressão varia de imagem para imagem. Apesar deste aumento de complexidade computacional e redução na taxa de compressão a literatura ([79] e [56]) relata que o tamanho variável das células, mesmo que o número de possibilidades seja restrito a células $B \times B$ ou $(B/2) \times (B/2)$, resulta em um grande ganho na qualidade das imagens codificadas.

- Células imagem e domínio representadas através de árvores quaternárias (*quad-trees*): Esta abordagem é uma extensão da abordagem anterior. Neste caso as células podem ter seus tamanhos variando em uma faixa mais larga e não apenas 2 diferentes possibilidades. Novamente as células iniciam com os maiores tamanhos possíveis e são decompostas à medida em que suas aproximações por $w_i(D_i)$ não forem satisfatórias.
- Células imagem e domínio retangulares: Esta abordagem ainda é similar às anteriores no sentido em que as células são divididas até que suas aproximações por $w_i(D_i)$ sejam suficientemente boas ou até que um tamanho mínimo seja atingido. Novamente as células R_i pertencem inicialmente a um reticulado de quadrados de tamanho fixo. A diferença deste para os métodos anteriores é que quando uma célula imagem R_i quadrada tem que ser dividida duas novas células são geradas, e não 4. Mais especificamente a divisão tem como resultado duas células retangulares e não 4 quadradas. Quando um retângulo tem que ser dividido o eixo de corte é perpendicular ao eixo de corte usado para o quadrado cuja divisão gerou o retângulo. Desta forma o resultado da divisão do retângulo é um par de quadrados. A figura 13.4 ilustra as divisões celulares desta abordagem de formatos e tamanhos de células imagem e domínio. Nesta abordagem o problema de re-amostragem das células domínio continua sendo resolvido da mesma forma trivial dos casos anteriores pois as células domínio continuam sendo exatamente 2 vezes maiores que as células imagem em cada dimensão.
- Células imagem e domínio triangulares Esta decomposição também tem como objetivo determinar as células imagem e domínio de forma adaptativa, isto é, as células imagem vão sendo divididas até que sejam suficientemente bem aproximadas por uma célula domínio ou um tamanho mínimo seja atingido. A diferença desta para as demais decomposições previamente citadas é que nesta as células imagem e domínio são triangulares. Inicialmente a imagem original é dividida numa malha triangular simples, como a ilustrada na figura 13.5. As células triangulares resultantes desta divisão são as primeiras candidatas a células imagem. Em seguida estas células são analisadas uma a uma: sempre que uma célula for suficientemente bem aproximada através de transformação afim de uma célula domínio, ela será considerada como uma célula imagem pertencente à decomposição final. Caso contrário a célula é dividida em outras células triangulares, nas quais o processo de análise será aplicado.

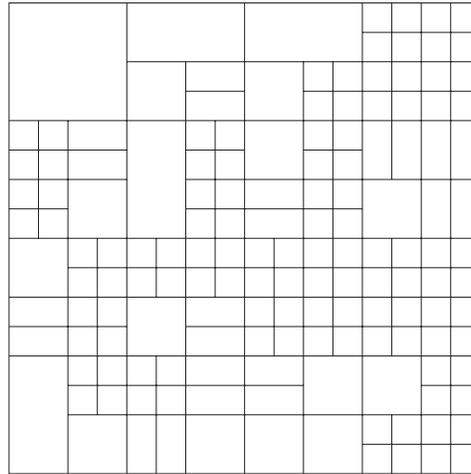


Figura 13.4: Divisão de células imagem quadradas e retangulares

A divisão de uma célula triangular em outras menores pode ser feita de diversas maneiras. Em [56] sugere-se uma divisão feita pelos segmentos que ligam os 3 pontos médios dos lados da célula original, tal como ilustra a figura 13.6. Uma característica importante desta divisão é a manutenção da retangularidade dos triângulos, ou seja, os 4 triângulos gerados a partir de um triângulo retângulo também são retângulos. Como a decomposição original normalmente é feita somente com triângulos retângulos, esta divisão resulta em triângulos imagem retângulos. Este fato simplifica o mapeamento dos pixels de células domínio em células imagem.

Em [42] a divisão proposta é através da *triangularização de Delaunay* [185]. Esta triangularização caracteriza-se pela não pertinência de seus vértices aos círculos circunscritos aos triângulos formados pelos demais vértices da triangularização. Decorre desta característica o fato que esta é a triangularização que maximiza o menor ângulo interno de todos os triângulos da triangularização. Este fato é importante para evitar problemas numéricos causados pela presença de triângulos com ângulos internos muito pequenos. Embora esta forma de decomposição ainda não esteja completamente explorada, ela apresenta características que a qualificam como promissora:

- A decomposição triangular é mais flexível que a retangular
 - As imperfeições causadas pelas aproximações das células imagem pelos domínios, que se acentuam nas bordas das imagens, distribuem-se em diversas direções (não são apenas horizontais e verticais). Isto faz com que o efeito perceptual das imperfeições seja menor.
- Células imagem e domínio em forma de paralelepípedo (3D): A modelagem de redundância caracterizada através de auto-similaridades pode ser estendida para seqüências de imagens ([47]). Neste caso as células imagem e domínio são tridimensionais e as

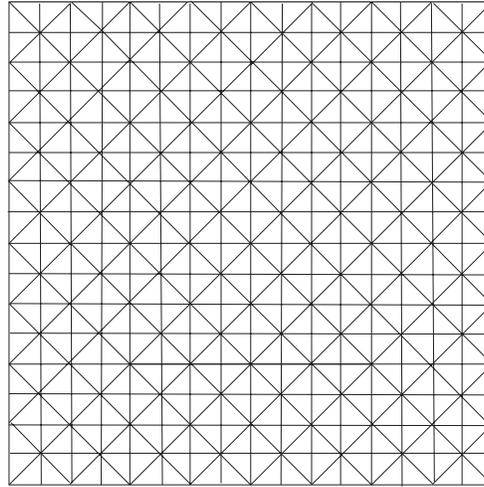
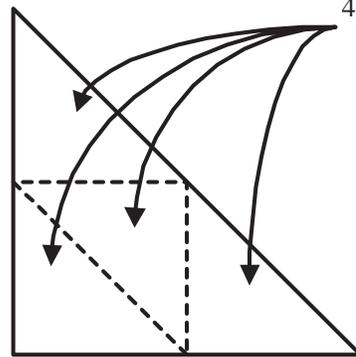


Figura 13.5: Decomposição de uma imagem em células triangulares

triângulo original



4 novos triângulos gerados a partir da divisão do triângulo original

Figura 13.6: Divisão de célula imagem triangular em quatro novas células

técnicas de segmentação de seqüências de imagens em células imagem devem ser estendida para três dimensões.

13.2.4 Classificação das Células Imagem e Domínio

Esta simplificação objetiva reduzir o esforço computacional do processo de busca do melhor par (D_i, w_i) para cada R_i . A idéia básica consiste na extração de parâmetros que representem a natureza das células e na posterior classificação das células de acordo com estes parâmetros. Esta simplificação tem o potencial de reduzir bastante os custos computacionais da busca dos melhores pares (D_i, w_i) para as células R_i 's. Isto porque, apesar da existência de uma etapa de preparação na qual todas as células domínio tem que ser classificadas e armazenadas em

estruturas de dados propícias para a recuperação posterior, a pesquisa do melhor par (D_i, w_i) para uma dada célula R_i é feita exclusivamente no conjunto de células domínio de mesma classe de R_i . Como o número de células R_i 's e D_i 's é grande, aproximadamente 4000 de cada tipo por imagem, a redução da varredura na pesquisa é bastante significativa para a diminuição da complexidade computacional.

Um exemplo deste tipo de simplificação está apresentado em [79]. A classificação das células é feita de acordo com a proposta de Ramamurthi e Gersho ([149]) para quantização vetorial classificada: as células são divididas em

- células sombreadas - não contém componentes de altas frequências espaciais
- células com arestas - contém componentes de altas frequências espaciais com uma orientação dominante
- células de textura - contém componentes de altas frequências espaciais sem uma orientação dominante

Além de efetuar a busca apenas em células domínio pertencentes à mesma classe das células imagem, Jacquin restringe as possibilidades de transformações de intensidade para cada classe. Antes de apresentar as diferentes possibilidades de transformações vale lembrar que uma transformação afim unidimensional, escrita como $l(x) = \alpha x + \Delta g$ é composta por um fator multiplicativo α e um deslocamento Δg . Para células sombreadas as transformações permitidas são as constantes, isto é, transformam qualquer intensidade em uma intensidade fixa (igual à intensidade luminosa média da célula imagem). No caso de células de textura os fatores multiplicativos α são restritos aos valores 0.7, 0.8, 0.9, 1.0 e os deslocamentos Δg são calculados para que as médias de intensidade luminosa dos pixels de R_i e $w_i(D_i)$ sejam aproximadamente as mesmas. Já para as células com arestas a transformação afim de intensidade luminosa é determinada de tal forma que os saltos de luminância que caracterizam as arestas sejam aproximadamente iguais em R_i e $w_i(D_i)$.

Outro exemplo de utilização de classificação das células foi proposto por McGregor e outros em [115]. Vários parâmetros representativos das células, tais como os coeficientes da transformada de Fourier ou da transformada cosseno, ou os momentos de diversas ordens (média e desvio padrão, por exemplo) das intensidades dos pixels das células, são extraídos. As células possíveis são armazenadas numa estrutura de dados que implementa memória associativa, que permite a busca de um dado a partir de parte de seu conteúdo. A parte do conteúdo utilizada para a identificação das células domínio é o conjunto de parâmetros delas extraídos. Para cada célula imagem R_i extrai-se os parâmetros e busca-se na estrutura de dados contendo os domínios quais os que apresentam parâmetros próximos dos de R_i . A pesquisa do melhor par (D_i, w_i) é feita apenas neste limitado conjunto de células domínio.

Mais um exemplo foi proposto por Boss e Jacobs no capítulo 4 do livro ([56]).

13.3 Base Teórica para a Codificação

13.3.1 Teorema da Colagem

Uma vez obtida a decomposição celular e os domínios/transições de cada célula imagem obtem-se uma contração f definida em toda imagem. Para cada imagem I_k esta contração foi obtida de forma que $f(I_k)$ aproxime I_k . O que pode garantir, contudo, que o ponto fixo de f se aproxime de I_k ? A resposta está no teorema da colagem:

Teorema 13.1 *Seja (X, d) um espaço métrico completo com métrica d . Seja $I_k \in X$ e $f : X \leftarrow X$ uma contração com fator de contração $0 \leq s < 1$ tais que*

$$d(I_k, f(I_k)) \leq \varepsilon$$

Então a distância entre I_k e o ponto fixo \bar{f} de f satisfaz

$$d(I_k, \bar{f}) \leq \frac{\varepsilon}{1 - s}$$

O conjunto das imagens (matrizes bidimensionais de intensidade luminosa) associado à distância do erro médio quadrático forma um espaço métrico completo.

Sejam s_i os fatores de contração das transformações τ_i . O fator de contração de τ é o máximo entre todos os s_i . Portanto, como as funções τ_i são escolhidas com fatores de contração menores que 1, o fator de contração de τ será menor que 1. Logo o teorema da colagem limita a distância entre uma imagem I_k e o ponto fixo da transformação τ que a representa.

13.3.2 Características da Visão Humana Exploradas

13.3.3 Importância Perceptual das Arestas

As versões deste método que utilizam classificação de células imagem e domínio têm a possibilidade de privilegiar a descrição de regiões que contenham arestas. Isto porque, tal como no codificador proposto por Jacquín ([79]), as transformações utilizadas para regiões de baixas frequências (células sombreadas) podem ser bastante simplificadas, enquanto que as transformações das células com arestas podem ser mais complexas.

Este privilégio tem como objetivo preservar uma parcela maior de informação para descrever células com arestas, enquanto que pouca informação é utilizada para descrever regiões sombreadas. Esta modelagem de ruído (*noise shaping*) baseia-se na importância das arestas na percepção das imagens pela visão humana.

13.4 Possibilidade de Emprego de Adaptabilidade

13.4.1 Adaptação de Tamanhos e Formatos de Regiões

A adaptabilidade para este método de codificação é utilizada quando se permite que os tamanhos e formatos das células imagem e domínio variem de acordo com as características locais das imagens sendo codificadas (ver seção 13.2.3).

13.5 Possibilidade de Emprego de Métodos Híbridos

13.5.1 Reabastecimento Condicional

A transformação de representação que leva uma imagem à contração cujo ponto fixo aproxima a imagem não é contínua, isto é, imagens próximas podem gerar contrações bem diferentes. Portanto não é interessante utilizar métodos preditivos para os coeficientes que descrevem as contrações.

Uma abordagem híbrida útil é o uso de reabastecimento condicionado (*conditional re-planishment*) para considerar apenas as regiões alteradas na codificação de seqüências de imagens.

13.6 Possibilidade de Emprego de Codificação sem Perdas

Este método não é adequado para codificação sem perdas uma vez que se baseia desde o princípio na aproximação de uma imagem pelo ponto fixo de uma contração. Para que a codificação seja sem perdas é preciso que se calcule exatamente o ponto fixo de uma contração. Isto é necessário tanto para a etapa de codificação (garantia da igualdade entre a imagem e o ponto fixo), quanto na etapa da decodificação (representação precisa da imagem). A obtenção precisa do ponto fixo de uma contração só é, entretanto, praticável para contrações simples. Na prática utiliza-se aproximações nas etapas de codificação (aproximação justificada pelo teorema da colagem) e decodificação (iteração da contração sobre qualquer imagem inicial para obtenção da imagem codificada).

13.7 Complexidade de Implementação

Este método apresenta uma assimetria muito grande no que tange a complexidade de implementação. O processo de codificação é bastante complexo pois requer uma extensa pesquisa de qual o melhor domínio para aproximar uma região. Várias simplificações têm de ser feitas à formulação original para que este processo possa ter sua complexidade computacional reduzida. Ainda assim a complexidade computacional é muito grande para que se possa realizar codificação em tempo real com a tecnologia de hoje.

Uma vez codificada a imagem, seu processo de decodificação é bastante mais simples, pois requer apenas a iteração da função que descreve a imagem (composta por uma transformações afins restritas às região da imagem).

13.8 Faixa de Compressão Atingida

A literatura relata a codificação de imagens monocromáticas com boa qualidade a entre 0.5 e 1 bpps.

Capítulo 14

Segunda Geração de Métodos de Codificação de Imagens

14.1 Introdução

A divisão dos métodos de codificação de imagens em duas diferentes gerações foi proposta por Kunt, Ikonomopoulos e Kocher em ([91]).

A primeira geração, composta por métodos chamados de codificadores de forma de onda, procura obter compressão através de redução de redundância das fontes de imagens e de algum conhecimento sobre o processo de visão (principalmente a variação da sensibilidade ao ruído em função da frequência). Dela fazem parte os métodos apresentados até agora neste trabalho:

- Codificação Preditiva
- Codificação por Transformadas
- Codificação por Quantização Vetorial
- Codificação em Sub-bandas
- Codificação Fractal

As melhores versões destes métodos apresentam taxas de compressão entre 10:1 e 20:1.

Os métodos da segunda geração procuram explorar mais a estrutura dos objetos pertencentes às imagens e o conhecimento da visão para obter taxas de compressão mais altas. A importância das arestas na determinação dos bordos dos objetos pertencentes às imagens é a característica mais importante do ponto de vista de estruturação das imagens. Já no que tange às características da visão mais utilizadas, destacam-se a importância das arestas na percepção das imagens e o completamento das áreas sem arestas.

Mais especificamente, dois foram os principais motivadores para os métodos da segunda geração de codificadores de imagens:

- a importância das arestas na percepção das imagens

- a observação de que é justamente nas proximidades de arestas que os métodos de codificação tradicionais (principalmente a codificação por transformadas) apresentam maiores distorções.

Com o objetivo de explorar as informações de arestas vários métodos de codificação foram propostos. Apesar destes métodos apresentarem técnicas e particularidades próprias, todos eles baseiam-se na decomposição das imagens em parcelas com e sem arestas.

O trabalho precursor desta nova geração de métodos foi o “*Synthetic Highs*” [163]. A proposta do trabalho foi dividir as imagens em duas componentes, altas e baixas frequências, e codificar cada componente separadamente. A componente de altas frequências recebe atenção especial devido à importância das arestas (presentes nas altas frequências espaciais) na percepção das imagens. A componente de baixas frequências pode ser codificada de forma mais grosseira.

Os trabalhos que sucederam o *Synthetic Highs* passaram a separar as imagens de forma similar, mas não exatamente igual. A classificação básica passou a ser:

- Componente devida aos bordos ou arestas, e
- Componente sem arestas ([202], [22] e [78]).

Esta última componente costuma ainda ser dividida em mais duas componentes: textura (altas frequências, mas sem arestas), e sombreados ou *Background* (baixas frequências). A decomposição mais geral, portanto, é a seguinte:

- Arestas
- Textura
- Sombreado

14.2 Descrição dos Métodos

A base dos métodos pertencentes à segunda geração de codificadores de imagens consiste na decomposição de imagens nas componentes descritas na introdução: bordos, textura e sombreados (observação: nem todos os métodos desta geração distinguem textura de sombreados). As seções subseqüentes apresentam as possíveis classificações destes métodos.

14.2.1 Classificação quanto ao tipo de técnica de detecção de arestas

Quanto à detecção de arestas os métodos da segunda geração podem ser classificados em detetores de arestas através de informações espectrais ou detetores de arestas através dos crescimento de regiões com características comuns.

Detecção de arestas através de características espectrais

A detecção de arestas através de características espectrais das imagens é mais utilizada ([78] e [27]). Nestes casos os pontos com arestas são obtidos através de processos similares à obtenção de máximos locais do gradiente de imagem (exemplo em [20]).

Detecção de arestas através de crescimento de regiões com características comuns

Outra alternativa de detecção de arestas, proposta em [86], procura dividir uma imagem em regiões com características comuns (com a mesma textura, por exemplo). Uma vez realizada tal divisão, as bordas de cada região correspondem às arestas das imagens (figura 14.1).



Figura 14.1: Processo de detecção de arestas através de crescimento de regiões com características comuns

14.2.2 Classificação quanto ao uso das arestas na codificação

Uma vez que as arestas tenham sido determinadas, existem basicamente duas formas de utilizá-las na codificação:

- Aumento da proporção do número de bits utilizados para codificar as arestas dentre o total de bits utilizados para a codificação das imagens completas
- Utilização de informações das imagens nos pontos de arestas para inferência do restante das imagens

14.2.3 Aumento proporcional da importância das arestas na codificação

A primeira forma de utilização das informações de arestas é a menos complexa delas. O aumento da importância proporcional das arestas na codificação das imagens é motivado por sua importância perceptual. Este aumento pode ser realizado por diferentes formas.

Em [78] uma imagem é decomposta em n componentes de altas frequências direcionais e 1 de baixas frequências. De cada componente de altas frequências codifica-se apenas os pontos com arestas (pontos de cruzamento de zero, isto é, pontos onde o sinal(+/-) da intensidade da imagem filtrada é diferente do sinal da intensidade da imagem filtrada em pontos vizinhos). A componente de baixas frequências é codificada de forma grosseira através de transformadas. Desta forma, o método aumenta a importância proporcional das arestas na alocação de bits

para codificação das imagens em detrimento da componente de baixas frequências e das informações de altas frequências que não constituem arestas.

O aumento de importância das arestas foi implementado de forma diferente em [190]. Este trabalho propõe um método de codificação preditiva na qual o preditor leva em consideração as informações de arestas das imagens. Portanto o erro de predição tende a ser menor nas regiões de arestas.

14.2.4 Inferência das imagens a partir de suas informações nos bordos

Outra forma de utilização das informações de arestas na codificação é mais sofisticada. Ela toma como base o fato da visão humana utilizar primordialmente as informações de variações de intensidade luminosa na observação de imagens (vide seção 5.4.6). A idéia por trás do método é codificar as imagens através de informações concentradas nos pontos com arestas e, na decodificação, recuperar uma aproximação da imagem original a partir das informações codificadas.

Em outras palavras, estes métodos pretendem entender, quantificar e, conseqüentemente, sintetizar o processo de visão humana, que entende as imagens essencialmente através de seus pontos de variação de intensidade luminosa (contornos, bordos ou arestas). De fato, se a visão é capaz de entender as imagens essencialmente a partir de seus bordos, também pode, em princípio, um método de decodificação recuperar grande parte da informação de uma imagem a partir de seus bordos. Para isto basta que esta decodificação seja capaz de realizar um processo de recuperação de imagens semelhante ao da visão humana.

Um método seguindo esta linha foi proposto em [27]. Os pontos de arestas são determinados através de análise multi-escala (vide seção 12.3.3). Além da determinação da localização dos pontos com arestas a caracterização das imagens proposta requer os valores das transformadas *wavelets* nestes pontos em diferentes escalas. O processo de recuperação de imagens a partir das transformadas *wavelets* corresponde à busca de uma função (cuja projeção na escala 2^0 é a imagem desejada) com as transformadas *wavelets* coincidentes nos pontos de arestas para as diferentes escalas e também satisfazendo aos seguintes critérios:

- a função deve ser o mais suave possível,
- a função deve ter o mínimo possível de oscilações.

Para quantificar estes critérios [27] propõe que a função candidata minimize uma norma Sobolev. O trabalho [27] também apresenta um algoritmo para obtenção de tal função a partir das informações de arestas.

Um aspecto importante deste método é a ausência de prova de que a solução do problema de minimização da norma de Sobolev com restrições nos pontos de arestas aproxima a imagem original. Portanto, não se pode, em princípio, esperar que o algoritmo de solução deste problema vá aproximar as imagens reais em todos os casos. Em outras palavras, não há garantias que o algoritmo funciona robustamente para qualquer tipo de imagem.

Para completar a codificação o codificador realiza este processo de recuperação, calcula a diferença entre a imagem original e a recuperada e codifica esta diferença através de métodos de transformada.

14.3 Base Teórica para a Codificação

14.3.1 Características da Visão Humana Exploradas

Esta geração de métodos procura ir além da anterior no que tange à utilização de características da visão com intuito de compressão de imagens.

A característica da visão humana mais importante para estes métodos é a importância dos bordos na percepção das imagens. A primeira forma de utilização desta característica consiste na caracterização mais detalhada de regiões com bordos do que regiões sem bordos. Outra forma de utilização desta característica é a utilização das informações dos bordos para inferir as informações luminosas no restante das imagens. De fato o completamento das informações de regiões sem bordos a partir das luminosidades nos bordos e de muito pouca informação original destas regiões sem bordos é sugerido pelas experiências de Cornsweet e Crane (vide seção 5.4.6 no capítulo 5).

14.4 Possibilidade de Emprego de Adaptabilidade

Estes métodos são bastante adaptativos em sua essência, uma vez que as informações de bordos são particulares de cada imagem.

14.5 Possibilidade de Emprego de Métodos Híbridos

Os métodos desta geração são tipicamente híbridos. Nos métodos em que as informações de arestas servem para aumentar a importância proporcional destas arestas na codificação a componente sem arestas é codificada através de métodos de primeira geração (principalmente codificação por transformadas).

Já nos métodos em que as informações de arestas servem como base de inferência das imagens ([27]), as diferenças entre as imagens reais e as recuperadas a partir dos bordos são também codificadas através de métodos de primeira geração (novamente codificação por transformadas).

14.6 Possibilidade de Emprego de Codificação sem Perdas

Os métodos pertencentes à segunda geração de codificadores de imagens não objetivam codificação sem perdas. Uma codificação sem perdas separada em cada componente (com e sem arestas) de uma imagem pode até ser feita (posteriormente a decodificação poderia recuperar cada componente separadamente e juntá-las obtendo uma decodificação sem perdas).

No entanto o princípio básico destes métodos consiste em codificar com boa precisão apenas as arestas das imagens. Mesmo dentre o conjunto de pontos com arestas os codificadores procuram obter quais as arestas mais importantes e estas sim definir precisamente. As parcelas restantes de uma imagem (textura e sombreamento, ou baixas frequências e partes das altas frequências que não correspondem à arestas) são codificadas sem muita precisão.

Portanto esta geração de métodos não é adequada para codificação sem perdas.

14.7 Complexidade de Implementação

A implementação destes métodos exige uma capacidade computacional maior do que os métodos clássicos da primeira geração (métodos preditivos e por transformadas).

A complexidade de implementação começa no algoritmo de detecção de arestas. No caso do uso das arestas para inferência do restante das imagens a complexidade computacional ainda aumenta devido ao algoritmo de busca da função que minimiza a norma Sobolev com restrições nos pontos de arestas.

14.8 Faixa de Compressão Atingida

Estes métodos de codificação ainda são objeto de pesquisas. Não se pode dizer que suas faixas de compressão estão bem caracterizadas ainda porque a eficiência dos métodos depende muito da quantidade de arestas detectadas nas imagens.

Os trabalhos publicados apregoam altas taxas de compressão: 30:1 em [78], mais de 16:1 em [190], 50:1 em [86] e 40:1 em [27]. Estas taxas, entretanto, foram obtidas através da codificação de poucas amostras de imagens. Para que se possa ter uma estimativa da taxa típica de compressão atingida por estes métodos é preciso codificar amostras mais significativas do conjunto de imagens possíveis.

Capítulo 15

Codificação por Modelos

15.1 Introdução

A abordagem de utilizada na codificação por modelos é bastante diferente das usadas nos demais métodos. Ao invés de procurar uma forma eficiente de representar as intensidades luminosas dos diversos pontos das imagens (pixels), a codificação por modelos procura descrever eficientemente o conteúdo das cenas cujas observações resultam nas imagens.

A representação de uma seqüência de imagens através de seu conteúdo é comum em computação gráfica (imagens sintéticas). Uma animação gráfica, por exemplo, pode ser representada através de seus objetos componentes e da iluminação ambiente. Para que a animação seja vista é necessário que as imagens resultantes da observação dos objetos pertencentes à cena sejam sintetizadas.

Numa seqüência de imagens naturais (não sintéticas), os objetos componentes, seus movimentos e a iluminação incidente têm de ser estimados. Estas estimações são a representação das imagens originais através da codificação por modelos. O nome “codificação por modelos” decorre do fato que a estimação de objetos, movimentos e níveis de iluminação genéricos constitui um problema muito complexo, que passa a ser tratável apenas quando os objetos componentes são conhecidos e modelados. Este é o caso de cenas compostas por pessoas falando em frente à uma câmera, como numa video-conferência. Estas cenas são compostas basicamente pela face e pelos ombros de uma pessoa. A reprodução das imagens codificadas requer a sintetização das cenas compostas pelos objetos estimados.

A potencialidade do método passa a ser grande caso os modelos dos objetos sejam realistas e necessitem pouca informação para caracterizar movimentos e alterações de iluminação. Um exemplo de codificação é a representação de uma cena composta por um sólido realizando movimentos rígidos com iluminação constante. A representação necessita de informação inicial para caracterizar o sólido (geometria, textura, refletância) e as fontes de iluminação. Em seguida os quadros da cena podem ser representados apenas pelos parâmetros que caracterizam o movimento do sólido: 3 parâmetros de translação e 3 de rotação. Embora este exemplo seja simples, ele mostra a grande potencialidade de redução de informação necessária para representar uma seqüência de imagens restritas a um modelo.

15.2 Descrição do Método

A idéia básica da codificação é representar as imagens através de parâmetros que caracterizem os objetos pertencentes à cena observada. Esta caracterização pressupõe a existência de um modelo paramétrico para representar os objetos da cena. Este modelo tem de ser do conhecimento tanto do codificador quanto do decodificador antes do início da codificação. De posse do modelo o método de codificação é responsável por extrair os parâmetros representativos dos objetos a partir dos dados disponíveis da cena. A decodificação, por outro lado, é responsável por sintetizar as imagens a partir do modelo pré-determinado e dos parâmetros representativos dos objetos em cada cena específica.

Desta forma a transformação de representação envolvida no método de codificação leva as intensidades luminosas das imagens em parâmetros dos modelos de cenas. Para que esta transformação possa ser realizada três problemas precisam ser resolvidos:

- restrição do universo de objetos pertencentes as cenas
- adoção de um modelo para os objetos
- extração dos parâmetros que caracterizam os objetos em uma cena específica

As técnicas e métodos utilizados nas resoluções destes problemas estão apresentadas nas seções sub-seqüentes.

15.2.1 Restrição do Universo de Objetos Pertencentes às Cenas

O grande ganho que o método de codificação por modelos pode trazer vêm da caracterização sucinta de objetos através de aproximações pertencentes a um modelo paramétrico. Para que esta caracterização seja sucinta é necessário que os modelos sejam capazes de representar os possíveis objetos com um pequeno conjunto de parâmetros. Isto requer que os modelos sejam bastante adequados aos objetos pertencentes às cenas. Assim, para que haja compressão de informação ao representar uma imagem pelo modelo de seu conteúdo é preciso que o modelo seja específico para os objetos contidos na cena. Em outras palavras, os requisitos de generalidade do modelo de objetos e redução da quantidade de informação necessária para caracterizar imagens são conflitantes.

Além deste fator de compressão de dados, a especialização do modelo de objetos é muito importante para reduzir a complexidade do problema de analisar os dados da cena original e convertê-los em parâmetros de um modelo que aproxime os objetos da cena. Este problema já é bastante complexo para modelos restritos e só aumenta de complexidade a medida em que o modelo cresce em generalidade.

Desta forma, para que a codificação por modelos seja implementada na prática é preciso inicialmente escolher um conjunto de objetos que poderão fazer parte das cenas a serem codificadas. O modelo utilizado para a representação das cenas deve ser específico para este conjunto de objetos escolhido.

A restrição dos objetos pertencentes às cenas é muito forte para cenas em geral, mas existe pelo menos um importante tipo de imagem para o qual ela se aplica bem: imagens de rostos humanos. Estas imagens estão presentes em video-conferências, por exemplo. Elas

são compostas basicamente por cabeças e ombros (*head and shoulders*), objetos para os quais existem modelos razoavelmente adequados. De fato é justamente este tipo de imagens que tem sido a meta da codificação por modelos ([4], [173], [85], [98] e [134]).

15.2.2 Modelos de Faces Humanas

O objetivo primordial do modelo na codificação é permitir a sintetização de imagens o mais realistas possível a partir dos parâmetros do modelo. Existem vários níveis de modelagem de objetos tri-dimensionais:

- modelagem geométrica: Descrição dos contornos dos objetos
- modelagem de iluminação: Determinação da luz incidente na câmera em função dos pontos de iluminação da cena, da geometria, refletância e textura dos objetos
- modelagem de movimentos faciais

Estas modelagens vem sendo objeto de estudo de áreas como computação gráfica (síntese de imagens) e psicologia (expressões faciais). Logo a codificação por modelos dispõe de técnicas já utilizadas em outras áreas para atingir o objetivo de modelar faces humanas.

A geometria da face humana é modelada através de um poliedro com faces triangulares. Adota-se um poliedro base para faces genéricas, que é adaptado para as dimensões do rosto da pessoa cujas imagens serão codificadas. Diferentes poliedros podem ser utilizados como base para a modelagem facial. A literatura relata que entre 100 e 500 faces devem ser utilizados para que o modelo tenha um grau de detalhamento necessário para sintetizar expressões faciais [4]. Dentre os modelos mais utilizados destaca-se o “Candide”, desenvolvido inicialmente na universidade de Linköping por Rydfalk [120], que vai além do poliedro base por conter também regras de movimentação do poliedro a partir de expressões faciais.

O ajuste do modelo à face de um indivíduo pode ser feito através de um *scanner* tridimensional (Cyberware). Este dispositivo permite também detectar a textura da face, que pode então ser mapeada no modelo. Este mapeamento da textura original da face melhora bastante o realismo do modelo.

A iluminação ambiente e a refletância da face são modeladas normalmente de forma simples: refletância considerada Lambertiana e iluminação difusa [173]. Desta forma a intensidade luminosa incidente em um ponto p da câmera imaginária utilizada para geração da imagem sintética (*ray tracing*) corresponde à cor do ponto f da face projetado em p .

Uma vez ajustado o modelo à pessoa cujas imagens serão codificadas, a sintetização das imagens do modelo é resolvida de forma bastante conhecida na área de computação gráfica, pois o problema fica reduzido à geração de imagens formadas por poliedros com textura conhecida.

Uma observação importante sobre a codificação por modelagem é que esta requer uma fase inicial de ajuste do modelo antes da codificação das cenas propriamente ditas. Outro fato importante é que esta fase inicial necessita de informação que vai além das imagens originais. De fato, esta fase requer dados geométricos tridimensionais e de textura sobre os objetos presentes na cena que não podem ser obtidos simplesmente através das imagens das cenas. É justamente para obter estes dados que o “scanner” tridimensional é utilizado.

Para completar o modelo é preciso descrever os movimentos da face. Os movimentos podem ser decompostos numa componente de movimento rígido, correspondente ao movimento da cabeça como um todo, e em outra não-rígida, correspondente às expressões faciais. A descrição do movimento rígido da cabeça, também conhecido por movimento global, é feita simplesmente através de 6 parâmetros: 3 de deslocamento e 3 de rotação em relação aos eixos x - y - z de R^3 .

Existem diferentes abordagens para a descrição dos movimentos não-rígidos ou locais.

A primeira abordagem, conhecida por “método do recorte-e-colagem” (*clip-and-paste method*) [4], consiste na determinação do posicionamento de apenas algumas regiões expressivas da face, como por exemplo os olhos e a boca. As demais regiões, como a testa e as bochechas permanecem inalteradas. Esta primeira abordagem é bastante simplista e não é muito utilizada.

A segunda abordagem consiste na consideração que os movimentos do poliedro facial são compostos por uma combinação linear de movimentos provenientes de expressões faciais. Desta forma uma seqüência de movimentos faciais é descrita através de uma seqüência de expressões faciais.

Esta abordagem tem a vantagem de reduzir o grau de liberdade dos movimentos faciais. De fato as expressões faciais são modeladas adequadamente através de um conjunto de unidades de ação, conhecidas na literatura por AU's (*action units*). Este conjunto, chamado FACS (*facial action coding system*), é composto por 44 AU's e foi desenvolvido por Ekman e Friesen [51]. Desta forma, restringir os movimentos faciais à uma combinação de movimentos devidos às AU's implica em determinar a movimentação das centenas de vértices do modelo através das intensidades relativas de dezenas de expressões faciais.

Esta restrição não é muito forte porque na prática existem restrições mecânicas no relacionamento entre os vértices (elasticidade da pele) e na velocidade de deslocamento do poliedro (acionamento dos músculos que comandam a face).

Outra vantagem grande desta abordagem é a relação entre a descrição dos movimentos e o significado do objeto codificado (especificamente expressão facial). Embora não seja de utilidade para a codificação, esta relação é importante para as animações, nas quais imagens sintéticas são geradas a partir da criação dos autores. É muito mais direto para um autor descrever os movimentos através de expressões faciais do que através de deslocamentos dos vértices que compõem o modelo da face.

A outra abordagem de descrição de movimentos consiste na modelagem física do tecido facial e dos músculos que agem para deslocar este tecido realizando as expressões faciais [173]. Cada músculo é modelado por meio de uma corda presa por um lado à estrutura facial (pontos fixos em relação à cabeça) e por outro a um pontos do tecido facial. Uma vez que o modelo físico do tecido facial seja conhecido tanto pelo codificador quanto pelo decodificador, a movimentação facial fica determinada pelo deslocamento dos músculos.

15.2.3 Extração dos Parâmetros do Modelo

Uma série de parâmetros do modelo deve ser determinada antes do início da codificação propriamente dita:

- Nível de iluminação ambiente

- Adequação do modelo geométrico à face da pessoa cujas imagens serão codificadas
- Mapeamento da textura da face original no modelo

Em um ambiente de teleconferência estas parametrizações podem ser feitas através de um *scanner* 3-D antes do início da conversação. O restante da parametrização necessária refere-se ao movimento facial.

A determinação dos parâmetros de movimentação de faces a partir das seqüências de imagens é objeto de pesquisa nos dias de hoje. Algumas abordagens para resolver este problema merecem destaque:

- Determinação de movimentos a partir de princípio de conservação do fluxo ótico incidente na câmera. A partir da equação de conservação do fluxo relaciona-se as intensidades luminosas da imagem e suas derivadas com os movimentos dos objetos na cena ([98]).
- Acompanhamento de regiões significativas das faces (olhos, sombrancelhas, contorno do cabelo, boca, nariz, queixo) através de métodos de minimização de potencial associado à variação de intensidade luminosa. O movimento das demais regiões fica automaticamente determinado a partir do movimento das regiões significativas e das restrições físicas do modelo de faces (geometria e elasticidade). O artigo [173] apresenta a descrição desta abordagem.
- Determinação dos parâmetros do modelo através de um ciclo de realimentação envolvendo análise e síntese de objetos observados na cena (faces e ombros). Nesta abordagem ([134] e [85]), alguma abordagem é utilizada para prover uma estimativa inicial dos parâmetros do modelo. Em seguida a imagem correspondente aos modelos estimados é sintetizada. A diferença entre esta imagem sintetizada e a original é utilizada para realimentar a estimação dos parâmetros do modelo.

15.3 Possibilidade de Emprego de Métodos Híbridos

Embora este método seja bem diferente dos demais métodos de codificação de imagens, existem formas de utilizá-lo conjuntamente com outro método. Uma possibilidade de utilização de hibridismo consiste em obter a inicialmente representação I_{modelo} de uma imagem original I_0 através da codificação por modelos. Em seguida calcula-se a diferença entre a imagem original e I_{modelo} e esta, denotada por I_{diff} , é codificada por algum método voltado para redução de redundância, como a codificação por transformadas.

15.4 Possibilidade de Emprego de Codificação sem Perdas

Este método não objetiva codificação sem perdas. Ele é totalmente desaconselhável para codificações que não admitam perdas.

15.5 Complexidade de Implementação

A complexidade computacional deste método é bastante grande. Primeiramente porque os algoritmos de extração de parâmetros faciais são complexos e têm de ser realizados em tempo real para as aplicações de teleconferência, por exemplo.

Além disto a sintetização de imagens por si só já é bastante cara computacionalmente. Apesar dos algoritmos de sintetização e apresentação de imagens já serem bem conhecidos na área de computação gráfica, máquinas de grande poder computacional são necessárias para que a geração de imagens seja realizada em tempo real.

15.6 Faixa de Compressão Atingida

Este método é o que apresenta taxas de compressão mais elevadas. A literatura relata experiências nas quais seqüências de imagens são codificadas a 9600 bits/s ([4]). Considerando que as imagens originais sejam matrizes 256×256 com 8 bits de cores e apresentadas a uma taxa de 30 imagens por segundo, a taxa de bits necessária para representar as imagens antes da codificação é de aproximadamente 15,7 Mbps. Logo a ordem de grandeza da taxa de compressão atingida é 20000:1, o que é muito maior que as taxas de compressão atingidas pelos outros métodos de compressão. A título de ilustração vale lembrar que a taxa de bits necessária para representar imagens dinâmicas segundo o padrão MPEG é de aproximadamente 1,4 Mbps, isto é, mais de cem vezes maior que a taxa obtida no protótipo de codificador por modelos de Huang, Reddy e Aizawa ([4]).

Para atingir taxas de compressão tão altas, esta codificação abdica não apenas da fidelidade às imagens originais, mas também do realismo, visto que as imagens são codificadas através de animações dos objetos contidos nas cenas.

Capítulo 16

Conclusão

Este trabalho apresenta uma visão bastante abrangente da codificação de imagens utilizando como ponto de partida bases comuns para os diversos métodos específicos de codificação existentes.

Estas bases comuns, a saber, a Teoria da Codificação e a decomposição do processo de codificação em fases permitem entender quais as bases teóricas utilizadas pelos diversos métodos existentes e quais os fatores importantes para a codificação.

A visão abrangente inclui a apresentação da descrição, das bases teóricas, das possibilidades de expansão e da complexidade de implementação das técnicas de cada uma das três fases da codificação. Tal apresentação do assunto, devido a sua abrangência e estruturação, pode ser utilizada para comparação dos métodos e escolha daquele mais adequado para uma utilização específica.

O trabalho apresenta uma bibliografia farta, podendo servir como base para estudos mais profundos sobre assuntos específicos da codificação de imagens. Isto possibilita a existência de um grande leque de continuações possíveis. Cada etapa da codificação, por exemplo, possui seus métodos próprios, que podem ser estudados de forma mais específica.

Referências Bibliográficas

- [1] K. Jain A. *Fundamentals of Digital Image Processing*. Prentice Hall International, Inc, 1989.
- [2] E.H. Adelson and F.P. Simoncelli. Nonseparable extensions of quadrature mirror filters to multiple dimensions. *Proceedings of the IEEE*, 78(4):652–664, Abril 1990.
- [3] P.J. Burt E.H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31(4):532–540, Abril 1983.
- [4] K. Aizawa and T.S. Huang. Model-based image coding: Advanced video coding techniques for very low bit-rate applications. *Proceedings of the IEEE*, 83(2):259–271, Fevereiro 1995.
- [5] A. Alcaim and L.V. Oliveira. Quantização vetorial binária de imagens codificadas por btc. *Anais do SIBGRAPI VI*, pages 75–80, Outubro 1993.
- [6] L.A. Anson. Fractal image compression. *BYTE*, pages 195–202, Outubro 1993.
- [7] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1(2):205–220, Abril 1992.
- [8] A. Aravind and A. Gersho. Low-rate image coding with finite-sate vector quantization. *in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 137–140, Março 1986.
- [9] F. Attneave. Some informational aspects of visual perception. *Psychophysics Review*, 61:183–193, 1954.
- [10] A. Averbuch, D. Lazar, and M. Israeli. Image compression using wavelet transform and multiresolution decomposition. *IEEE Transactions on Image Processing*, 5(1):4–15, Janeiro 1996.
- [11] R.L. Baker and H.H. Shen. A finite-sate vector quantizer for low-rate image sequence coding. *in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 760–763, Março 1987.
- [12] R.H. Bamberger and M. Smith. A filter bank for the directional decomposition of images: Theory and design. *IEEE Transactions on Signal Processing*, 40(4):882–893, Abril 1992.

- [13] M. Barlaud, P. Sole, T. Gairdon, M. Antonini, and P. Mathieu. Pyramidal lattice vector quantization for multiscale image coding. *IEEE Transactions on Image Processing*, 3(4):367–381, Julho 1994.
- [14] M.F. Barnsley and L.P. Hurd. *Fractal Image Compression*. AK Peters, Ltd., 1995.
- [15] M.F. Barnsley and A.D. Sloan. A better way to compress images. *BYTE*, pages 215–223, Janeiro 1988.
- [16] J.M. Beaumont. Image data compression using fractal techniques. *BT Technical Journal*, 9(8):93–109, Outubro 1991.
- [17] K.A. Birney and T.R. Ficher. On the modeling of dct and subband image data for compression. *IEEE Transactions on Image Processing*, 4(2):186–193, Fevereiro 1995.
- [18] R.F. Brammer. Unified image computing on fractals and chaos model techniques. *Optical Engineering*, 28(7):726–734, Julho 1989.
- [19] Z.L. Budrikis. Visual fidelity criterion and modeling. *Proceedings of the IEEE*, 60(7):771–779, Julho 1972.
- [20] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.
- [21] A.B. Carlson. *Communication Systems - An Introduction to Signals and Noise in Electrical Communication*. McGraw-Hill, 1986.
- [22] S. Carlsson. Sketch based coding of gray level images. *Signal Processing North-Holland*, 15(1):57–83, 1988.
- [23] D. Chen and A.C. Bovik. Visual pattern image coding. *IEEE Transactions on Communications*, 38(12):2137–2146, Dezembro 1990.
- [24] D. Chen and A.C. Bovik. Hierarchical visual pattern image coding. *IEEE Transactions on Communications*, 40(4):671–675, Abril 1992.
- [25] B. Chitprasert. Human visual weighted progressive image transmission. *IEEE Transactions on Communications*, 38(7):1040–1044, Julho 1990.
- [26] C.K. Chui, editor. *Waveletes: A Tutorial in Theory and Applications*. Academic Press, INC., 1992.
- [27] C.K. Chui, editor. *Wavelets - A Tutorial in Theory and Applications*, chapter Second Generation Compact Image Coding with Wavelets, pages 655–678. Academic Press, Inc, 1992.
- [28] R.J. Clarke. *Transform Coding of Images*. Academic Press, INC., 1985.
- [29] D.J. Connor, R.C. Brainard, and J.O. Limb. Intraframe coding for picture transmission. *Proceedings of the IEEE*, 60(7):779–791, Julho 1972.

- [30] D.J. Connor and J.O. Limb. Properties of frame-difference signals generated by moving images. *IEEE Transactions on Communications*, COM-22(10):1564–1575, Outubro 1974.
- [31] T.N. Cornsweet. Changes in the appearance of stimuli of very high luminance. *Psychophysics Review*, 69:257–273, 1962.
- [32] L. Corte-Real and A.P. Alves. A very low bit rate video coder based on vector quantization. *IEEE Transactions on Image Processing*, 5(2):263–273, Fevereiro 1996.
- [33] P.C. Cosman, R.M. Gray, and M. Vetterli. Vector quantization of image subbands: A survey. *IEEE Transactions on Image Processing*, 5(2):202–225, Fevereiro 1996.
- [34] B. Costa, L. Darsa, and J. Gomes. Image methamorphosis. *Proceedings of SIBGRAPI V*, pages 19–27, 1992.
- [35] C.A.R. Costa and P.L. de Geus. Um esquema de compressão de sequências de imagens baseado na dct. *Anais do SIBGRAPI VI*, pages 45–54, Outubro 1993.
- [36] H.D. Crane and M.R. Clark. Three-dimensional visual stimulus deflector. *Applied Optics*, 17:706–714, 1978.
- [37] H.D. Crane and C.M. Steele. Accurate three-dimensional eyetracker. *Applied Optics*, 17:691–705, 1978.
- [38] R.E. Crochiere. Sub-band coding. *Bell Systems Technical Journal*, pages 1633–1653, Setembro 1981.
- [39] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41:909–996, Novembro 1988.
- [40] L.D. Davisson. Rate-distortion theory and application. *Proceedings of the IEEE*, 60(7):800–808, Julho 1972.
- [41] L.D. Davisson. Universal noiseless coding. *IEEE Transactions on Information Theory*, IT-19(6):783–795, Novembro 1973.
- [42] F. Davoine, M. Antonini, J.M. Chassery, and M. Barlaud. Fractal image compression based on delaunay triangularization and vector quantization. *IEEE Transactions on Image Processing*, 5(2):338–346, Fevereiro 1996.
- [43] E.J. Delp and O.R. Mitchel. Image compression using block truncation coding. *IEEE Transactions on Communications*, COM-27(9):1335–1342, Setembro 1979.
- [44] R.A. DeVore, B. Jawerth, and B.J. Lucier. Image compression through wavelet transform coding. *IEEE Transactions on Information Theory*, 38(2):719–746, Março 1992.
- [45] F. Dufaux and F. Moscheni. Motion estimation techniques for digital tv: A review and a new contribution. *Proceedings of the IEEE*, 83(6):858–876, Junho 1995.

- [46] M. Dunham and R. Gray. An algorithm for the design of labelled-transition finite-state vector quantizer. *IEEE Transactions on Communications*, COM-33(1):83–89, Janeiro 1985.
- [47] T. Ebrahimi, E. Reusens, and W. Li. New trends in very low bitrate video coding. *Proceedings of the IEEE*, 83(6):877–891, Junho 1995.
- [48] O. Egger and W. Li. Subband coding of images using asymmetrical filter banks. *IEEE Transactions on Image Processing*, 4(4):478–485, Abril 1995.
- [49] O. Egger, W. Li, and M. Kunt. High compression image coding using an adaptive morphological subband decomposition. *Proceedings of the IEEE*, 83(2):272–287, Fevereiro 1995.
- [50] J.D. Eggerton and M.D. Srinath. A visually weighted quantization scheme for image bandwidth compression at low data rates. *IEEE Transactions on Communications*, COM-38(8):840–846, Agosto 1986.
- [51] P. Ekman and W. Friesen. Facial action coding system, 1977.
- [52] P.M. Farrelle and A.K. Jain. Recursive block coding — a new approach to transform coding. *IEEE Transactions on Communications*, COM-34(2):161–179, Fevereiro 1986.
- [53] J.A. Ferwerda and D.P. Greenberg. A psychophysical approach to assessing the quality of antialiased images. *IEEE Computer Graphics and Applications*, pages 85–95, Setembro 1988.
- [54] T.R. Fischer. A pyramid vector quantizer. *IEEE Transactions on Information Theory*, IT-32(4):568–583, Julho 1986.
- [55] Yuval Fisher. Fractal image compression. In *SIGGRAPH*, 1992. SIGGRAPH 92 Course Notes.
- [56] Yuval Fisher, editor. *Fractal Image Compression: Theory and Applications*. Springer-Verlag, 1995.
- [57] B.M. Fitingof. Optimal coding in the case of unknown and changing message statistics. *Probl. Inform. Transm.*, 2(2):1–7, 1966.
- [58] B.M. Fitingof. The compression of discrete information. *Probl. Inform. Transm.*, 3(3):22–29, 1967.
- [59] R. Forchheimer and T. Kronander. Image coding — from waveforms to animation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(12):2008–2023, Dezembro 1989.
- [60] J. Foster, R.M. Gray, and M. Dunham. Finite-state vector quantization of waveform coding. *IEEE Transactions on Information Theory*, IT-31:348–355, Maio 1985.

- [61] W. Frei and B. Baxter. Rate-distortion coding simulation for color images. *IEEE Transactions on Communications*, COM-25(11):1385–1392, Novembro 1977.
- [62] D.L. Gall. Mpeg: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):46–58, Abril 1991.
- [63] R.G. Gallager. Variations on a theme by huffman. *IEEE Transactions on Information Theory*, IT-24(6):668–673, Setembro 1978.
- [64] A. Gersho and M. Yano. Adaptive vector quantization by progressive code-vector replacement. *Proc. IEEE Int. Conf. Acoust.*, pages 133–136, 1985.
- [65] M. Ghanbari. The cross-search algorithm for motion estimation. *IEEE Transactions on Communications*, 38(7):950–953, Julho 1990.
- [66] H. Gish and J. Pierce. Asymptotically efficient quantizing. *IEEE Transactions on Information Theory*, IT-14:676–681,683, Setembro 1968.
- [67] M. Goldberg, P.R. Boucher, and S. Shlien. Image compression using adaptive vector quantization. *IEEE Transactions on Communications*, COM-34(2):180–187, Fevereiro 1986.
- [68] J. Gomes and L. Velho. *Computação Gráfica: Imagem*. Instituto de Matemática Pura e Aplicada - Sociedade Brasileira de Matemática, 1994.
- [69] R. Gonzalez and C. Wintz. *Digital Image Processing*. Addison Wesley, 1987.
- [70] D.J. Granrath. The role of human visual models in image processing. *Proceedings of the IEEE*, 69(5):552–561, Maio 1981.
- [71] H.G. Grassman. Theory of compound colors. *The Philosophic Magazine*, 4(7):254–264, 1854.
- [72] R.M. Gray. *Entropy and Information Theory*. Springer-Verlag, 1990.
- [73] O.G. Guleryuz and M.T. Orchard. Optimized nonorthogonal transforms for image compression. *IEEE Transactions on Image Processing*, 6(4):507–522, Abril 1997.
- [74] A. Habibi. Survey of adaptive image coding techniques. *IEEE Transactions on Communications*, COM-25(11):1275–1284, Novembro 1977.
- [75] B.G. Haskell, F.W. Mounts, and J.C. Candy. Interframe coding of videotelephone pictures. *Proceedings of the IEEE*, 60(7):792–800, Julho 1972.
- [76] D. Hubel and T.N. Wiesel. Brain mechanisms of vision. *Scientific American*, pages 150–162, Setembro 1978.
- [77] D.A. Huffman. A method for construction of minimum-redundancy codes. *Proc. IRE*, 40:1098–1101, Setembro 1952.

- [78] A. Ikonomopoulos and M. Kunt. High compression image coding via directional filtering. *Signal Process*, 8(1):179–203, Maio 1985.
- [79] A. E. Jacquin. Image coding based on fractal theory of iterated contractive image transform. *IEEE Transactions on Image Processing*, 1(1):18–30, Janeiro 1992.
- [80] A.K. Jain. Advances in mathematical models for image processing. *Proceedings of the IEEE*, 69(5):502–528, Maio 1981.
- [81] A.K. Jain. Image data compression: A review. *Proceedings of the IEEE*, 69(3):349–389, Março 1981.
- [82] J.R. Jain and A.K. Jain. Displacement measurement and its application in interframe image coding. *IEEE Transactions on Communications*, 29(12):1799–1808, Dezembro 1981.
- [83] W.C. Kellogg. Information rates in sampling and quantization. *IEEE Transactions on Information Theory*, IT-13:506–511, Julho 1967.
- [84] Y.H. Kim and J.W. Modestino. Adaptive entropy coded subband coding of images. *IEEE Transactions on Image Processing*, 1(1):31–48, Janeiro 1991.
- [85] R. Koch. Dynamic 3-d scene analysis through synthesis feedback control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):556–568, Junho 1993.
- [86] M. Kocher and M. Kunt. A contour-texture approach to picture coding. In *Proc. ICASSP-82*, pages 436–440, Maio 1982.
- [87] A. N. Komolgorov. Three approaches to the quantitative definition of information. *Probl. Inform. Transm.*, 1(1):1–7, 1965.
- [88] F. Kossentini, W.C.L. Chung, and M.J. Smith. A jointly optimized subband coder. *IEEE Transactions on Image Processing*, 5(9):1311–1323, Setembro 1996.
- [89] E.R. Kretzmer. Statistics of television signals. *Bell Systems Technical Journal*, pages 751–763, Julho 1952.
- [90] T. Kronander. Sampling of bandpass pyramids. *Optical Engineering*, 28(7):125–127, Julho 1989.
- [91] M. Kunt, A. Ikonomopoulos, and M. Kocher. Second generation image coding techniques. *Proceedings of the IEEE*, 73(4):549–574, Abril 1985.
- [92] G.G. Langdon. An introduction to arithmetic coding. *IBM Journal of Research and Development*, 28(2):135–149, Março 1984.
- [93] G.G. Langdon and J. Rissanen. Compression of black—white images with arithmetic coding. *IEEE Transactions on Communications*, COM-29(6):858–867, Junho 1981.

- [94] P.J. Laurent, A.L. Méhauté, and L.L. Schumaker, editors. *Wavelets, Images and Surface Fitting*. AK Peters, Ltd., 1994.
- [95] J.P. Leduc, J.M. Odobez, and C. Labit. Adaptive motion-compensated wavelet filtering for image sequence coding. *IEEE Transactions on Image Processing*, 6(6):862–878, Junho 1997.
- [96] M.D. Lema and O.R. Mitchel. Absolute moment block truncation coding and application to color images. *IEEE Transactions on Communications*, COM-32(10):1148–1157, Outubro 1984.
- [97] J.Y. Lettvin, H.R. Maturana, W.S. McCulloch, and W.H.Pitts. What the frog's eye tells the frog's brain. *Proceedings of the IRE*, 47:1940–1959, 1959.
- [98] H. Li, P. Roivainen, and R. Forchheimer. 3-d motion estimation in model-based facial image coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):545–555, Junho 1993.
- [99] C.C. Lien, C.L. Huang, and J.G. Chen. Complex-subband transform for subband-based motion estimation/compensation and coding. *IEEE Transactions on Image Processing*, 6(5):693–702, Maio 1997.
- [100] J.S. Lim. *Two-Dimensional Signal and Image Processing*. Prentice Hall International, Inc, 1990.
- [101] J.O. Limb, B. Rubinstein, and J.E. Thompson. Digital coding of color video signals — a review. *IEEE Transactions on Communications*, COM-25:1349–1384, Novembro 1977.
- [102] Y.C. Lin and S.C. Tai. Fast full-search block-matching algorithm for motion-compensated video compression. *IEEE Transactions on Communications*, 45(5):527–531, Maio 1997.
- [103] Y. Linde, A. Buzo, and R.M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, COM-28(1):84–95, Janeiro 1980.
- [104] M.L. Liou. Overview of the p×64 kbps video coding standard. *Communications of the ACM*, 34(4):59–63, Abril 1991.
- [105] S.P. Lloyd. A sampling theorem for stationary (wide sense) stochastic processes. *Computer Vision, Graphics and Image Processing*, COM-25(11):1–12, Novembro 1977.
- [106] B. Mahesh and W.A. Pearlman. Hexagonal sub-band coding for images. *International Conference on Acoustics, Speech and Signal Processing*, 3:1953–1956, Maio 1989.
- [107] J. Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580, Abril 1975.

- [108] S.G. Mallat. Multifrequency channel decomposition of images and wavelet models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(12):2091–2110, Dezembro 1989.
- [109] J.L. Mannos and D.J. Sakrison. The effects of a visual fidelity criterion on the encoding of images. *IEEE Transactions on Information Theory*, IT-20(4):525–536, Julho 1974.
- [110] P.A. Maragos, R.W. Schafer, and R.M. Mersereau. Two-dimensional linear prediction and its application to adaptive predictive coding of images. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-32(6):1213–1229, Dezembro 1984.
- [111] J.D. Markel and A.H. Gray Jr, editors. *Linear Prediction Of Speech*. Springer-Verlag, 1976.
- [112] D. Marr. *Vision*. Freeman, 1982.
- [113] J. Max. Quantizing for minimum distortion. *IRE Transactions on Information Theory*, 6:7–12, 1960.
- [114] N.F. Maxemchuk and J.A. Stuller. An adaptive intraframe dpcm codec based upon nonstationary image model. *Bell Systems Technical Journal*, 58(6):1395–1412, Julho-Agosto 1979.
- [115] D.R. McGregor, R.J. Fryer, P. Cockshott, and P. Murray. Faster fractal compression. *Dr Dobb's Journal*, pages 34–41, Janeiro 1996.
- [116] J.L. Mitchell and W.B. Pennebaker. Optimal hardware and software arithmetic coding procedures for the q-coder. *IBM Journal of Research and Development*, 32(6):727–736, Novembro 1988.
- [117] J.L. Mitchell and W.B. Pennebaker. Probability estimation for the q-coder. *IBM Journal of Research and Development*, 32(6):737–752, Novembro 1988.
- [118] J.L. Mitchell and W.B. Pennebaker. Software implementations of the q-coder. *IBM Journal of Research and Development*, 32(6):753–774, Novembro 1988.
- [119] D.M. Monro and F. Dudbridge. Fractal block coding of images. *Electronics Letters*, 28(11):1053–1055, Maio 1992.
- [120] M. Rydfalk. Candide: A parametrized face. Technical Report Rep. LiTH-ISY-I-0866, Dep. Elec. Eng. - Linkoping University, Outubro 1987.
- [121] T. Murakami, K. Asai, and E. Yamazaki. Vector quantizer of video signals. *Electronics Letters*, 3:1005–1006, Novembro 1982.
- [122] G. Musmann, P. Pirsch, and H.J. Grallert. Advances in picture coding. *Proceedings of the IEEE*, 73(4):523–548, Abril 1985.

- [123] N.M. Nasrabadi. Use of vector quantizers in image coding. *in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 125–128, Março 1985.
- [124] N.M. Nasrabadi and Y. Feng. Image compression using address-vector quantization. *IEEE Transactions on Communications*, 38(12):2166–2173, Dezembro 1990.
- [125] N.M. Nasrabadi and R.A. King. Image compression using vector quantization in the transform domain. *Pattern Recognition Letters*, pages 323–329, 1983.
- [126] N.M. Nasrabadi and R.A. King. Image coding using vector quantization: A review. *IEEE Transactions on Communications*, 36(8):957–971, Agosto 1988.
- [127] V.E.I. Neagoe. Predictive ordering technique and feedback transform coding for data compression of still pictures. *IEEE Transactions on Communications*, 40(2):385–396, Fevereiro 1992.
- [128] M. Nelson and J.L. Gailly. *The Data Compression Book*. M&T Books, second edition, 1996.
- [129] A.N. Netravali and B.G. Haskell. *Digital Pictures: Representation and Compression*. Plenum Press, 1988.
- [130] A.N. Netravali and J.O. Limb. Picture coding: A review. *Proceedings of the IEEE*, 68(3):366–405, Março 1980.
- [131] T.Q. Nguyen and P.P. Vaidyanathan. Two channel perfect reconstruction fir qmf structures which yield linear phase fir analysis and synthesis filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37:676–690, Maio 1989.
- [132] J.P. Oakley and M.J. Cunningham. A function space model for digital image sampling and its applications in image reconstruction. *Computer Vision, Graphics and Image Processing*, COM-25(11):171–197, Novembro 1977.
- [133] A. Ortega and M. Vetterli. Adaptive scalar quantization without side information. *IEEE Transactions on Image Processing*, 6(5):665–676, Maio 1997.
- [134] D.E. Pearson. Developments in model-based video coding. *Proceedings of the IEEE*, 83(6):892–906, Junho 1995.
- [135] D.E. Pearson and J.A. Robinson. Visual communication at very low data rates. *Proceedings of the IEEE*, 73(4):795–812, Abril 1985.
- [136] W.B. Pennebaker, J.L. Mitchell G.G. Langdon, and R.B. Arps. An overview of the basic principles of the q-coder adaptive binary arithmetic coder. *IBM Journal of Research and Development*, 32(6):717–726, Novembro 1988.
- [137] W.B. Pennebaker and J.L. Mitchell. *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, 1993.

- [138] M.G. Perkins. A comparison of the hartley, cas-cas, fourier and discrete cosine transforms for image coding. *IEEE Transactions on Communications*, 36(6):758–761, Junho 1988.
- [139] C.L. Podilchuk, N.S. Jayant, and N. Farvardin. Three-dimensional subband coding of video. *IEEE Transactions on Image Processing*, 4(2):125–139, Fevereiro 1995.
- [140] D.C. Popescu, A. Dimca, and H. Yan. A nonlinear model for fractal image coding. *IEEE Transactions on Image Processing*, 3(7):373–382, Março 1997.
- [141] W.K. Pratt. Spatial transform coding of color images. *IEEE Transactions on Communications*, COM-19:980–992, Dezembro 1971.
- [142] W.K. Pratt. *Digital Image Processing*. John Wiley & Sons, 1978.
- [143] R.L. Queiroz and R. Eschbach. Fast downscaled inverses for image compressed with m-channel lapped transforms. *IEEE Transactions on Image Processing*, 6(6):794–807, Junho 1997.
- [144] R.A. Quinnell. Image compression part 1. *Electronic Design*, pages 62–71, Janeiro 1993.
- [145] R.A. Quinnell. Image compression part 2. *Electronic Design*, pages 120–126, Março 1993.
- [146] R.A. Quinnell. Image compression part 3. *Electronic Design*, pages 114–120, Maio 1993.
- [147] M. Rabbani and P.W. Jones. *Digital Image Compression Techniques*. Spie Press, 1991.
- [148] B. Ramamurthi. *Vector Quantizer of Images Based on a Composite Source Model*. PhD thesis, University of California, Santa Barbara, Abril 1985.
- [149] B. Ramamurthi and A. Gersho. Classified vector quantization of images. *IEEE Transactions on Communications*, COM-34(11):1105–1115, Novembro 1986.
- [150] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Transactions on Image Processing*, 2:160–175, Abril 1993.
- [151] K. Ramchandran, M. Vetterli, and C. Herley. Wavelets, subband coding, and best bases. *Proceedings of the IEEE*, 84(4):541–560, Abril 1996.
- [152] K.R. Rao and P. Yip. *Discrete Cosine Transform*. Academic Press, INC., 1990.
- [153] H.L. Resnikoff. *The Illusion of Reality*. Springer-Verlag, 1989.
- [154] L.A. Riggs, F. Ratliff, J.C. Cornsweet, and T.N. Cornsweet. The disappearance of steadily fixated visual test objects. *Journal of the Optical Society of America*, 43:495, 1953.

- [155] R. Rinaldo and G. Calvagno. Image coding by black prediction of multiresolution subimages. *IEEE Transactions on Image Processing*, 4(7):909–920, Julho 1995.
- [156] J. Rissanen. A universal data compression system. *IEEE Transactions on Information Theory*, IT-29(5):656–664, Setembro 1983.
- [157] A. Rosenfeld and A.C. Kak. *Digital Picture Processing*. Academic Press, INC., 1982.
- [158] B. Rubinstein and J.O. Limb. Statistical dependence between components of differentially quantized color signals. *IEEE Transactions on Communications*, COM-20:890–899, Outubro 1972.
- [159] J.A. Saghri, P.S. Cheatham, and A. Habibi. Image quality measure based on a human visual system model. *Optical Engineering*, 28(7):813–818, Julho 1989.
- [160] A. Said and W.A. Pearlman. An image multiresolution representation for lossless and lossy compression. *IEEE Transactions on Image Processing*, 5(9):1303–1310, Setembro 1996.
- [161] T. Saito, H. Takeo, K. Aizawa, H. Harashima, and H. Miyakawa. Adaptive discrete cosine transform image coding using gain/shape vector quantization. *in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 129–132, Abril 1986.
- [162] D.J. Sakrison. On the role of the observer and a distortion measure in image transmission. *IEEE Transactions on Communications*, COM-25(11):1251–1267, Novembro 1977.
- [163] W.F. Schreiber, C.F. Knapp, and N.D. Kay. Synthetic highs, an experimental tv bandwidth reduction system. *Journal of SMPTE*, 68:525–537, Agosto 1959.
- [164] C.E. Shannon. The mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423,623–656, Outubro 1948.
- [165] G. Sharma and H.J. Trussell. Digital color imaging. *IEEE Transactions on Image Processing*, 6(7):901–932, Julho 1997.
- [166] E. Shusterman and M. Feder. Image compressing via improved quadtree decomposition algorithms. *IEEE Transactions on Image Processing*, 3(2):207–215, Março 1994.
- [167] E.A.B. Silva and M. Ghanbari. On the performance of linear phase wavelet transforms in low bit-rate image coding. *IEEE Transactions on Image Processing*, 5(5):689–704, Maio 1996.
- [168] E.A.B. Silva, D.G. Sampson, and M. Ghanbari. A successive approximation vector quantizer for wavelet transform image coding. *IEEE Transactions on Image Processing*, 5(2):299–310, Fevereiro 1996.

- [169] M.J.T. Smith and W.C.L. Chung. Recursive time-varying filter banks for subband image coding. *IEEE Transactions on Image Processing*, 4(7):885–895, Julho 1995.
- [170] M.J.T. Smith and S.L. Eddins. Subband coding of images with octave band tree structures. *Proc. ICASSP*, 3:1382–1385, Dezembro 1987.
- [171] P. Sriram and M.W. Marcellin. Image coding using wavelet transform and entropy-constrained trellis-coded quantization. *IEEE Transactions on Image Processing*, 4(6):725–733, Junho 1995.
- [172] D. Taubman and A. Zakhor. Orientation adaptive subband coding of images. *IEEE Transactions on Image Processing*, 3(4):421–437, Julho 1994.
- [173] D. Terzopoulos and K. Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):569–579, Junho 1993.
- [174] W.F. Schreiber D.E. Troxel. Transformation between continuous and discrete representation of images: A perceptual approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(2):178–186, Março 1985.
- [175] H.C. Tseng and T.R. Fischer. Transform and hybrid transform/dpcm coding of images using pyramid vector quantization. *IEEE Transactions on Communications*, COM-35(1):79–86, Janeiro 1987.
- [176] P.P. Vaidyanathan. Multirate digital filters, filter banks, polyphase networks, and applications: A tutorial. *Proceedings of the IEEE*, 78(1):56–93, Janeiro 1990.
- [177] P.P. Vaidyanathan and Z. Doganata. The role of lossless systems in modern digital signal processing: A tutotial. *IEEE Trans. Education*, pages 181–197, Agosto 1989.
- [178] P.P. Vaidyanathan and P.Q. Hoang. Lattice structures for optimal design and robust implementation of two-channel perfect reconstruction qmf banks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(1):81–94, Janeiro 1988.
- [179] M. Vetterli. Multi-dimensional sub-band coding: Some theory and algorithms. *Signal Process*, 6(2):97–112, Abril 1984.
- [180] M. Vetterli and J. Koracevic. *Wavelets and Subband Coding*. Prentice Hall International, Inc, 1995.
- [181] G.K. Wallace. The jpeg still compression standard. *Communications of the ACM*, 34(4):30–44, Abril 1991.
- [182] L. Wang and M. Goldberg. Progressive image transmission using vector quantization on images in pyramid form. *IEEE Transactions on Communications*, 37(12):1339–1349, Dezembro 1989.

- [183] L. Wang and M. Goldberg. Block transform image coding by multistage vector quantization with optimal bit allocation. *IEEE Transactions on Communications*, 39(9):1360–1369, Setembro 1991.
- [184] A.B. Watson. Temporal sensitivity. In K.R. Boff, L. Kauffman, and J.P. Thomas, editors, *Handbook of Perception and Human Performance*. John Wiley & Sons, 1986.
- [185] D.F. Watson. Computing the n-dimensional delaunay tessellation with application to voronoi polytopes. *Computer Journal*, 24(2):167–172, 1981.
- [186] E.J. Wegman and D.J. DePriest, editors. *Statistical Image Processing and Graphics*. Marcel Dekker, INC., 1986.
- [187] T. Welch. A technique for high-performance data compression. *IEEE Computers*, 17(6):8–19, Junho 1984.
- [188] P.H. Westerink, D.E. Boekee, J. Biemond, and J.W. Woods. Subband coding of images using vector quantization. *IEEE Transactions on Communications*, 36(6):713–719, Junho 1988.
- [189] R. Wilson, H.E. Knutsson, and G.H. Granlund. Anisotropic nonstationary image estimation and its applications: Part i — restoration of noisy images. *IEEE Transactions on Communications*, COM-31:388–397, Março 1983.
- [190] R. Wilson, H.E. Knutsson, and G.H. Granlund. Anisotropic nonstationary image estimation and its applications: Part ii — predictive image coding. *IEEE Transactions on Communications*, COM-31:398–406, Março 1983.
- [191] I.H. Witten, R.M. Neal, and J.G. Cleary. Arithmetic coding to data compression. *Communications of the ACM*, 30(6):520–540, Junho 1987.
- [192] W. Chen W.K. Pratt and L.R. Welch. Slant transform image coding. *IEEE Transactions on Communications*, COM-22:1075–1093, Agosto 1974.
- [193] G. Wolberg. *Digital Image Warping*. IEEE Computer Society, 1988.
- [194] R.C. Wood. On optimum quantization. *IEEE Transactions on Information Theory*, IT-15:248–252, Março 1969.
- [195] J.W. Woods, editor. *Subband Image Coding*. Kluwer, 1991.
- [196] J.W. Woods and S.D. O’Neil. Subband coding images. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34:1278–1288, Outubro 1986.
- [197] G. Wu and J.W.Mark. Multiuser variable rate subband coding incorporating dsi and buffer control. *IEEE Transactions on Communications*, 38(12):2159–2165, Dezembro 1990.

- [198] S.W Wu and A. Gersho. Improved decoder for transform coding with applications to the jpeg baseline system. *IEEE Transactions on Communications*, 40(2):251–254, Fevereiro 1992.
- [199] Y. Wu and D.C. Coll. Btc-vq-dct hybrid coding of digital images. *IEEE Transactions on Communications*, 39(9):1283–1287, Setembro 1991.
- [200] G. Wyszecki and W.S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. John Wiley & Sons, second edition, 1982.
- [201] Z. Xiong, K. Ramchandran, and M. T. Orchard. Space-frequency quantization for wavelet image coding. *IEEE Transactions on Image Processing*, 6(5):676–693, Maio 1997.
- [202] J.K. Yan and D.J. Sakrison. Encoding of images based on a two—component source model. *IEEE Transactions on Communications*, COM-25(11):1315–1322, Novembro 1977.
- [203] K. Zeger and A. Gersho. Pseudo—gray coding. *IEEE Transactions on Communications*, 38(12):2147–2158, Dezembro 1990.
- [204] J. Ziv. Coding theorems for individual sequences. *IEEE Transactions on Information Theory*, IT-24(4):405–412, Julho 1978.
- [205] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, IT-23(3):337–343, Maio 1977.
- [206] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, IT-24(5):53–536, Setembro 1978.