



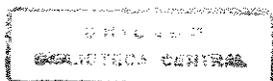
Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Comunicações

Transmissão de Vídeo em Pacotes através de Redes de Faixa Larga

Autor: Luiz Rômulo Mendes
Orientador: Max Henrique Machado Costa

Dissertação submetida à Faculdade de Engenharia Elétrica e de Computação da UNICAMP, como parte dos requisitos para obtenção do Título de Mestre em Engenharia Elétrica

Maio 1998



1169/87

UNIDADE	BC
N.º CHAMADA:	Umicamp
	M522t
V. Ex.	
TOMBO B/	34699
PROC.	395/98
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	R\$ 11,00
DATA	11/08/98
N.º CPD	

CM-00114996-0

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

M522t

Mendes, Luiz Rômulo

Transmissão de vídeo em pacotes através de redes de
faixa larga. / Luiz Rômulo Mendes. --Campinas, SP: [s.n.],
1998.

Orientador: Max Henrique Machado Costa
Dissertação (mestrado) - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Sistemas de comunicação em banda larga.
2. Transmissão de imagem. 3. Compressão de dados
(Telecomunicações). 4. Processamento de imagens.
5. Sistemas de transmissão de dados. I. Costa, Max
Henrique Machado. II. Universidade Estadual de Campinas.
Faculdade de Engenharia Elétrica e de Computação. III.
Título.

Dedicatória

Aos meus pais, Rômulo e Edina.

"Não há virtude nem vitória mais bela do que
comandar e vencer a si próprio."

Brantôme

Agradecimentos

Ao Prof. Max, pela orientação, apoio e paciência que foram cruciais para a realização deste trabalho.

Ao Prof. Waldman, que possibilitou meu ingresso nesta Universidade.

Ao Prof. Dalton, pelas sugestões proíficas.

Ao Prof. Michel e seus orientados, Paula, Fred, e Kyn.

A todos os amigos do DECOM, que tornaram agradável a execução deste trabalho, em especial, Renê, Fabbryccio, Nalon, Marcelo e Mylene, cujas discussões foram muito proveitosas.

Aos meus pais, à Andreia, e aos meus irmãos, pelo apoio e compreensão.

À CAPES e ao CPqD, pelo apoio financeiro oferecido.

Resumo

Este trabalho apresenta comparações e análises de desempenho de sistemas hierárquicos de codificação de vídeo digital para transmissão através de redes de pacotes com possíveis perdas de células. Nestes sistemas, o escalonamento dos dados codificados tem como objetivo reduzir os efeitos dos erros de canal, particularmente no caso de perdas de células causadas por congestionamentos da rede.

Desenvolve-se uma plataforma de simulação de um sistema de codificação e decodificação de vídeo digital que inclui um modelo de transmissão com perdas de células. O codec implementado é baseado no Padrão MPEG-2, e possibilita o escalonamento da seqüência codificada através de particionamento de dados, requantização, ou separação espectral fixa.

Inicialmente são apresentadas técnicas de compreensão de imagens e vídeo digital. Em seguida, apresenta-se um visão geral dos padrões atuais de compressão de vídeo, com atenção especial para o MPEG-2. Vários aspectos a serem considerados quando se transmite vídeo digital codificado em redes de pacotes são discutidos. Finalmente, apresentam-se resultados obtidos com as simulações e extraem-se conclusões sobre o desempenho dos métodos de escalonamento investigados.

As principais contribuições deste trabalho consistem no desenvolvimento do pacote de programas de simulações, e nos resultados obtidos com tais simulações.

Abstract

This thesis presents performance analysis of hierarchical video compression systems with transmissions over lossy packet networks. In these systems, scalability is used to reduce the effects of cell losses, typically caused by network congestion.

A simulation platform was developed that includes a packet transmission model with loss. The implemented codec is based on MPEG-2 Standard, and allows for scalability by data partitioning, requantization, or spectral separation techniques.

Initially, image and video digital compression techniques are introduced. Secondly, a general overview of current video compression standards is presented, with special attention to MPEG-2. Following, general aspects related to transmission of digital video over packet networks are discussed. Finally, simulation results are presented, and conclusions on the performance of the investigated scalability techniques are drawn.

The main contributions of this work are the development of the simulation package, and results obtained with the simulations.

Sumário

<i>Capítulo 1 - Introdução</i>	<i>1</i>
<i>Capítulo 2 - Técnicas Utilizadas em Compressão de Imagens e Vídeo</i>	<i>3</i>
2.1 Introdução	3
2.2 Codificação Preditiva	6
2.2.1 Otimização do quantizador.....	9
2.2.2 Estimação e compensação de movimento.....	11
2.2.2.1 Busca logarítmica	15
2.2.2.2 Busca em três passos	16
2.2.2.3 Busca em direções conjugadas	17
2.2.2.4 Busca telescópica.....	18
2.3 Codificação por Transformadas Ortogonais	19
2.3.1 Transformada de Karhunen-Loève	21
2.3.2 Transformada discreta de Fourier.....	22
2.3.3 Transformada discreta do coseno	22
2.3.4 Transformada de Walsh-Hadamard	23
2.3.5 Estratégias de Codificação	24
2.4 Codificação por Subbandas	25
2.4.1 Analogia entre codificação por subbandas e codificação por transformadas.....	27
2.5 Quantização Vetorial	28
2.6 Codificação de Entropia	30
2.7 Codificação Hierárquica	31
2.7.1 Hierarquias de resolução variável.....	32
2.7.2 Hierarquias de resolução fixa	34
<i>Capítulo 3 - Padrões de Compressão de Imagens e Vídeo Digital</i>	<i>35</i>
3.1 Introdução	35
3.2 O Padrão JPEG	35
3.3 O Padrão MPEG-2	36
3.3.1 Histórico.....	36

3.3.2	Processo de desenvolvimento do Padrão MPEG-2	37
3.3.3	Partes que compõe o Padrão ISO 13818	38
3.3.3.1	Parte 1: Sistemas.....	38
3.3.3.2	Parte 2: Vídeo.....	39
3.3.3.3	Parte 3: Áudio.....	39
3.3.3.4	Parte 4: Testes de conformidade.....	40
3.3.3.5	Parte 5: Software.....	40
3.3.3.6	Parte 6: DSM-CC.....	40
3.3.3.7	Parte 7: Codificação de áudio avançada	41
3.3.3.8	Parte 8: Vídeo com 10 bits de resolução amplitude (cancelado).....	41
3.3.3.9	Parte 9: Interface de tempo real (RTI – “Real-Time Interface”).....	41
3.3.3.10	Parte 10: Testes de conformidade do DSM-CC.....	42
3.3.4	Descrição dos processos de codificação e decodificação	42
3.3.5	Tipos de quadros	45
3.3.6	Reordenamento de quadros.....	46
3.3.7	Codificação de um macrobloco	47
3.3.8	Escalonamento.....	51
3.3.8.1	Escalonamento Temporal	52
3.3.8.2	Escalonamento Espacial	52
3.3.8.3	Escalonamento SNR	52
3.3.8.4	Particionamento de Dados (“data partitionig”).....	53
3.4	Outros Padrões de Compressão de Vídeo Digital	53
<i>Capítulo 4 - Transmissão de Vídeo em Redes de Faixa Larga.....</i>		55
4.1	Introdução.....	55
4.2	Redes ATM	55
4.3	Técnicas para Minimizar os Efeitos dos Erros de Canal	57
<i>Capítulo 5 - Simulações e Resultados.....</i>		59
5.1	Introdução.....	59
5.2	Codificador	59
5.2.1	Codificador com separação espectral	64
5.2.2	Codificador com requantização dos coeficientes.....	68
5.3	Modelagem do Congestionamento de Tráfego	70
5.4	Simulações e Análise dos Resultados Obtidos	72

5.4.1	Simulações considerando perda total do canal de baixa prioridade	80
5.4.2	Simulações considerando perda de células do canal de baixa prioridade.....	83
Capítulo 6 - Conclusões e Trabalhos Futuros		88
Apêndice A - Resultados Complementares.....		91
Apêndice B.....		98
Referências Bibliográficas		102

Lista de Figuras

Fig. 2.1 - Diagrama de blocos de um sistema DPCM	7
Fig. 2.2 - Disposição dos pixels utilizados pelo preditor	8
Fig. 2.3 - Quantizadores uniformes	9
Fig. 2.4 - Falsa estimação de movimento	12
Fig. 2.5 - Estimação de movimento.....	13
Fig. 2.6 - Busca logarítmica 2-D	16
Fig. 2.7 - Busca em três passos.....	17
Fig. 2.8 - Busca em direções conjugadas.....	18
Fig. 2.9 - Busca telescópica.....	19
Fig. 2.10 - Exemplo de transformação unitária de 2 pontos.....	21
Fig. 2.11 - Periodicidade implícita da DFT e da DCT	23
Fig. 2.12 - Exemplos de alocação de bits (extraído de [9]).....	25
Fig. 2.13 - Diagrama geral de um sistema de codificação por subbandas.....	26
Fig. 2.14 - Par de filtros espelhados em quadratura	27
Fig. 2.15 - Relação entre transformada e subbandas	28
Fig. 2.16 - Exemplo de quantização vetorial.....	29
Fig. 2.17 - Modos de varredura dos coeficientes do bloco transformado	31
Fig. 2.18 - Exemplo de codificação hierárquica com estrutura piramidal.....	33
Fig. 3.1 - Diagrama geral de um codificador MPEG-2	43
Fig. 3.2 - Estrutura de um quadro não entrelaçado de formato 4:2:0	44
Fig. 3.3 - Exemplo de estrutura temporal de seqüência	45
Fig. 3.4 - Relação entre coeficientes originais e coeficientes recuperados para blocos não intracodificados (a), e para blocos intracodificados (b).	49
Fig. 3.5 - Matrizes de quantização recomendadas pelo Padrão MPEG-2, para blocos intracodificados (a) e para blocos não intracodificados (b).....	50
Fig. 3.6 - Exemplo de particionamento de dados	53
Fig. 4.1 - Célula ATM	56
Fig. 5.1 - Relação entre taxas para $ivc=1$ e $ivc=0$	63

Fig. 5.2 - Sequência codificada segmentada em duas partições	65
Fig. 5.3 - Bloco codificado	66
Fig. 5.4 - Codificador com separação espectral.....	67
Fig. 5.5 - Codificador com requantização dos coeficientes.....	69
Fig. 5.6 – Exemplo de requantização de coeficientes	70
Fig. 5.7 – Exemplo de tráfego agregado para simulação de perdas de células.....	72
Fig. 5.8 - Primeiro quadro da seqüência "Kiel".....	73
Fig. 5.9 - Primeiro quadro da seqüência "Mobile"	74
Fig. 5.10 - Primeiro quadro da seqüência "Tennis".....	74
Fig. 5.11 – Relação $\frac{\text{taxa AP}}{\text{taxa total}}$ versus razão $\frac{Q1}{Q2}$, para a seqüência "Mobile"	77
Fig. 5.12 - Razão entre taxas para a seqüência "Kiel".....	79
Fig. 5.13 - Razão entre taxas para a seqüência "Mobile"	79
Fig. 5.14 - Razão entre taxas para a seqüência "Tennis".....	80
Fig. 5.15 - Curvas de distorção x taxa para casos de 0% de perda, e perda total do canal de baixa prioridade (BPP), para a seqüência "Kiel".....	81
Fig. 5.16 - Curvas distorção x taxa para casos de 0% de perda, e perda total do canal de baixa prioridade (BPP), para a seqüência "Mobile"	81
Fig. 5.17 - Curvas de distorção x taxa para casos de 0% de perda, e perda total do canal de baixa prioridade (BPP), para a seqüência "Tennis"	82
Fig. 5.18 - Curvas de distorção x taxa para a seqüência "Tennis", taxa de perda: 1%	84
Fig. 5.19 - Curvas de distorção x taxa para a seqüência "Tennis", taxa de perda: 5%	84
Fig. 5.20 - Curvas de distorção x taxa para a seqüência "Tennis", taxa de perda: 10%	85
Fig. 5.21 - Curvas de distorção x taxa para a seqüência "Tennis", taxa de perda: 20%	85
Fig. 5.22 - Curvas de distorção x taxa para a seqüência "Tennis", taxa de perda: 30%	86
Fig. 5.23 - Curvas de distorção x taxa para a seqüência "Tennis", taxa de perda: 40%	86
Fig. A.1 - Curvas de distorção x taxa para seqüência "Kiel", taxa de perda: 1%	92
Fig. A.2 - Curvas de distorção x taxa para seqüência "Kiel", taxa de perda: 5%	92
Fig. A.3 - Curvas de distorção x taxa para seqüência "Kiel", taxa de perda: 10%	93
Fig. A.4 - Curvas de distorção x taxa para seqüência "Kiel", taxa de perda: 20%	93
Fig. A.5 - Curvas de distorção x taxa para seqüência "Kiel", taxa de perda: 30%	94
Fig. A.6 - Curvas de distorção x taxa para seqüência "Kiel", taxa de perda: 40%	94

Fig. A.7 - Curvas de distorção x taxa para seqüência “Mobile”, taxa de perda: 1%.....	95
Fig. A.8 - Curvas de distorção x taxa para seqüência “Mobile”, taxa de perda: 5%.....	95
Fig. A.9 - Curvas de distorção x taxa para seqüência “Mobile”, taxa de perda: 10%.....	96
Fig. A.10 - Curvas de distorção x taxa para seqüência “Mobile”, taxa de perda: 20%.....	96
Fig. A.11 - Curvas de distorção x taxa para seqüência “Mobile”, taxa de perda: 30%.....	97
Fig. A.12 - Curvas de distorção x taxa para seqüência “Mobile”, taxa de perda: 40%.....	97

Lista de Tabelas

Tabela 5.1 - Valores de Q1/Q2 e de N	78
Tabela B.1 - Tabela de Huffman para codificação dos coeficientes da DCT (Tabela B.14, extraída do Padrão MPEG-2[2]).....	98
Tabela B.2 - Tabela para codificação do par “corrida de zeros / nível do coeficiente”, quando utiliza-se o código de Escape (Tabela B.16, extraída do Padrão MPEG-2 [2])	101

Capítulo 1

Introdução

Atualmente, os sistemas de transmissão de vídeo popularmente difundidos são predominantemente analógicos. Entretanto, o atual estágio de desenvolvimento da tecnologia digital e as novas necessidades dos consumidores propiciaram uma tendência de migração para sistemas de transmissão de vídeo digital. Para viabilizar economicamente estes sistemas, é necessário a utilização de técnicas de compressão digital. Diversas técnicas de compressão digital têm sido desenvolvidas. No Capítulo 2 são apresentadas algumas destas técnicas com vistas à compressão de imagens e vídeo digital.

Um sistema de compressão de imagens ou vídeo digital nada mais é do que um conjunto destas técnicas de compressão, ajustadas para determinada aplicação ou conjunto de aplicações. Vários sistemas de codificação de imagens e vídeo têm sido padronizados (JPEG, MPEG-1 [33], MPEG-2 [2], H.263 [31], H261 [32]) para permitir a interoperabilidade entre equipamentos de diversos fabricantes, e facilitar a massificação da utilização de aplicações que envolvem compressão de vídeo e imagens. Alguns destes padrões são apresentados no Capítulo 3, em especial o padrão de codificação de vídeo digital conhecido como MPEG-2 [2,4,25].

O Padrão MPEG-2 requer taxas de transmissão e qualidade de serviço que não são fornecidos pela maioria das redes convencionais atualmente instaladas (e.g. a rede telefônica possibilita transmissão a taxas de, no máximo, 64 Kbits/s). A transmissão de vídeo digital com qualidade igual (TV digital), ou superior (e.g., HDTV) à TV convencional (PAL-M e NTSC), através de redes digitais, tornou-se possível graças ao advento das redes digitais de serviços integrados de faixa larga (RDSI-FL), e devem ocupar boa parte da banda de transmissão disponível nestas redes. Por outro lado, o modo de transferência assíncrono (ATM - “Asynchronous Transfer Mode”) tem se firmado como uma boa opção para implementação destas redes. No Capítulo 4 é apresentada uma sucinta visão geral das redes RDSI-FL e do Padrão ATM. Neste capítulo, também são abordados os problemas envolvidos quando se transmite vídeo digital em redes RDSI-FL baseadas no Padrão ATM. Um destes

problemas é a perda de células. São apontadas algumas técnicas que visam reduzir os efeitos destas perdas, quando se transmite vídeo MPEG-2 através destas redes. Uma das soluções apontadas é o escalonamento do vídeo codificado em camadas, com proteção especial para a camada base.

No Capítulo 5, três modos de escalonamento de vídeo digital em duas camadas são testados: requantização dos coeficientes espectrais (“SNR scalability”), particionamento de dados (“data partitioning”), e separação espectral fixa. Os dois primeiros são recomendados pelo Padrão MPEG-2, enquanto que o terceiro é proposto neste trabalho, e é bastante semelhante ao particionamento de dados.

Nestas simulações, cada um dos três sistemas de escalonamento pode utilizar os dados referentes às duas camadas na etapa de estimação e compensação de movimento (EM/CM), ou somente os dados de alta prioridade (ou seja, da camada base) nesta etapa (EM/CM). Considerando esta diferenciação, são seis os sistemas testados. É utilizado um modelo de transmissão através de redes ATM que garante a integridade da camada base na recepção, enquanto a camada superior é sujeita a perda de células. Os dados produzidos por cada sistema de codificação são transmitidos, considerando-se este modelo, e os resultados, dados em curvas de distorção x taxa, são analisados e comparados. Finalmente, as conclusões finais são apresentadas no Capítulo 6.

Capítulo 2

Técnicas Utilizadas em Compressão de Imagens e Vídeo

2.1 Introdução

Uma imagem digital pode ser considerada como uma matriz de pontos chamados pixels ou pels. A palavra pixel vem de “picture element”, e representa o elemento básico de uma cena. Cada pixel pode ser representado por um valor ou um conjunto de valores discretos. Quando o pixel é representado por um único valor, este geralmente corresponde à luminância da imagem naquele ponto. No caso de imagens coloridas, usualmente o pixel é representado por três valores. Uma representação muito utilizada de uma imagem colorida é obtida por suas componentes básicas RGB, onde R representa a componente vermelha (“Red”), G a componente verde (“Green”), e B a componente azul (“Blue”). Outra possível representação é através da decomposição YCrCb, onde o Y representa a luminância e Cr e Cb são sinais de crominâncias. Além destas duas representações, várias outras são possíveis. Em geral, elas procuram representar bem todas as cores visíveis pelo olho humano.

Uma imagem pode ainda representar outras faixas de frequência que apesar de invisíveis ao olho humano, como o infravermelho e o ultravioleta, são de interesse para aplicações específicas. Por exemplo, satélites de sensoriamento remoto captam imagens em faixas de frequências no infravermelho para identificação de determinados minerais, solos ou vegetações.

Uma seqüência de vídeo nada mais é do que um conjunto ordenado de imagens, coloridas ou monocromáticas e, geralmente, possuem um sinal de áudio associado. Cada imagem de uma seqüência de vídeo é chamada de quadro. Os quadros são temporalmente espaçados em intervalos iguais e estes intervalos estão relacionados com o número de quadros por segundo. A taxa de quadros deve ser tal que dê a impressão de movimento contínuo ao observador humano. Para se conseguir obter esse efeito com taxas de quadros mais baixas, os

sinais convencionais de TV (NTSC/PAL) utilizam uma técnica conhecida como entrelaçamento de campos. Cada quadro é subdividido em campos de linhas pares e ímpares. Na reprodução do sinal, os dois campos de cada quadro são exibidos alternadamente a uma taxa de 60 campos por segundo, resultando em uma taxa de 30 quadros por segundo. Já nos cinemas a reprodução é feita a uma taxa de 24 quadros por segundo¹, pois a sensibilidade do olho humano ao efeito de “flickering” (percepção da descontinuidade temporal da cena) varia com a iluminação do ambiente, sendo menor em ambientes menos iluminados.

A importância da compressão de um sinal de vídeo digital pode ser ilustrada tomando como exemplo uma seqüência em cores de 30 quadros por segundo com cada quadro tendo uma dimensão de 720 X 525 pixels [3]. Se cada pixel é representado por três componentes RGB, e cada componente pode assumir 256 valores (o que equivale a 8 bits por componente, e 24 bits por pixel), são necessários $720 \times 525 \times 24$ bits para representar um quadro. Considerando-se a taxa de 30 quadros por segundo, chega-se a taxa de $720 \times 525 \times 24 \times 30$ bits/s, ou aproximadamente 272 Mbits/s. A transmissão de uma seqüência de vídeo a esta taxa é possível graças ao advento das redes óticas de alta velocidade. Entretanto, para tornar-se economicamente viável, o canal deve ser compartilhado por vários usuários e permitir transmissão de vários canais de vídeo. Neste ponto surge a necessidade de técnicas de compressão. Vários padrões de compressão de vídeo digital têm sido estabelecidos, correspondendo a diferentes aplicações, como videoconferência (H.261 [32]), videofonia (H.263 [31]), multimídia (MPEG-4) e TV digital (MPEG-1 [33], MPEG-2 [2]). Dentre estes padrões, o conhecido como MPEG-2 merece atenção especial por possibilitar uma vasta gama de aplicações, incluindo TV digital de alta definição (HDTV).

Quando se refere à taxa de compressão, dada pela razão entre a taxa de bits original e a obtida após o processo de compressão, é importante considerar as características da seqüência envolvida no processo. Por exemplo, se uma seqüência analógica é digitalizada, inicialmente em quadros de dimensão 512 x 512 pixels, e em seguida em quadros de 256 x 256 pixels, são obtidas duas seqüências semelhantes. A primeira tem uma taxa de amostragem duas vezes maior que a segunda, tanto na horizontal quanto na vertical. Se as duas seqüências forem comprimidas de maneira que a qualidade das seqüências recuperadas seja a mesma, a taxa de

¹ Na verdade, a taxa efetiva é de 48 quadros por segundo, pois cada quadro é apresentado duas vezes consecutivas.

compressão obtida para a primeira seqüência será maior que para a segunda, o que dá a falsa impressão que o processo foi mais eficiente no primeiro caso. Isso ocorre porque a primeira seqüência possui mais redundância espacial que a segunda, que pode ser considerada como resultado de uma “compressão” da primeira, através do processo de subamostragem. Análise semelhante pode ser feita se o número de amostras por quadro é o mesmo mas há uma diferença no número de bits por amostra. Por isso, para quantificarmos a compressão de maneira mais coerente, utilizaremos bits por amostra como figura de mérito para compressão.

Uma cena real pode ser considerada um sinal de quatro dimensões: um eixo temporal e três espaciais. A captura desta cena por uma câmara digital é feita através de uma amostragem tridimensional, ou seja vários conjuntos bidimensionais de amostras (pixels) igualmente espaçados no tempo. Cada conjunto de amostras constitui um quadro, ou uma projeção bidimensional da cena real em um dado instante. Para perfeita reprodução, cada cena deve ser amostrada pelo menos a taxa de Nyquist [9], o que geralmente é conseguido através de uma pré-filtragem do sinal (por exemplo, utilizando um filtro passa-baixa bidimensional). Considerem-se amostras da luminância do sinal. Cada amostra pode ser quantizada e representada por k bits, que identificam 2^k níveis. Esta técnica é conhecida como codificação PCM (“pulse code modulation”). Este processo de aquisição da cena invariavelmente envolve perda de informação. Entretanto, a rede de amostragem e o número de bits por amostra podem ser ajustados de forma que esta perda de informação não seja perceptível ao olho humano. A quantidade de dados gerados por este processo é consideravelmente alta, como já foi exposto em exemplo. Entretanto, o sinal de vídeo digital gerado possui muita redundância, tanto espacial quanto temporal, e várias técnicas de compressão podem ser utilizadas para se reduzir estas redundâncias. Algumas das principais técnicas utilizadas são listadas a seguir:

- Codificação preditiva,
- Codificação por transformadas ortogonais,
- Codificação por sub-bandas,
- Quantização vetorial,
- Codificação de entropia,
- Codificação hierárquica.

Nas seções seguintes são apresentados aspectos gerais de cada uma destas técnicas de compressão de vídeo digital.

2.2 Codificação Preditiva

Os pixels de cada quadro de uma seqüência de vídeo apresentam alto grau de correlação, tanto temporal quanto espacial. Esta correlação representa redundância de informação. Devido a esta correlação, os pixels vizinhos podem ser utilizados para se obter uma estimativa (predição) do pixel atual. Considere uma seqüência unidimensional que apresente correlação entre suas amostras, sendo $x(m)$ a $m^{\text{ésima}}$ amostra desta seqüência. Uma estimativa desta amostra, $\hat{x}(m)$, pode ser obtida através da Eq. (2.1), onde c_i é o fator de ponderabilidade (coeficiente do preditor) da $i^{\text{ésima}}$ amostra utilizada na predição.

$$\hat{x}(m) = \sum_{i=0}^{m-1} c_i x(i) \quad (2.1)$$

A técnica que explora este processo de predição é conhecida como codificação DPCM (“Differential Pulse Code Modulation”). O diagrama geral de um sistema DPCM é apresentado na Fig. 2.1. Neste sistema, somente a diferença quantizada, $e'(n)$, entre a amostra, $x(n)$, e a sua predição, $\hat{x}(n)$, é enviada através do canal. Antes de ser enviada, esta diferença pode passar por etapas que envolvem codificação de entropia e codificação de canal.

Note que o receptor só tem acesso aos valores dos pixels reconstruídos. Já que a quantização do sinal diferença, $e(n)$, introduz erro, os valores reconstruídos diferem dos originais. Para assegurar que sejam formadas predições idênticas, tanto no transmissor quanto no receptor, o transmissor também baseia suas predições em valores reconstruídos. Para isso, o quantizador é posicionado dentro do laço de predição. Se esta medida não é adotada, há propagação de erros até que os preditores sejam reinicializados.

O trunfo da codificação DPCM é que a seqüência diferença, $e(n)$, possui tanto a variância quanto a correlação menores que as da seqüência de entrada $x(n)$, propiciando uma compressão mais eficiente. Por outro lado, uma desvantagem desta técnica é que, quando há perda de uma amostra da seqüência diferença, ocorre propagação de erros no receptor, até que os preditores sejam reinicializados.

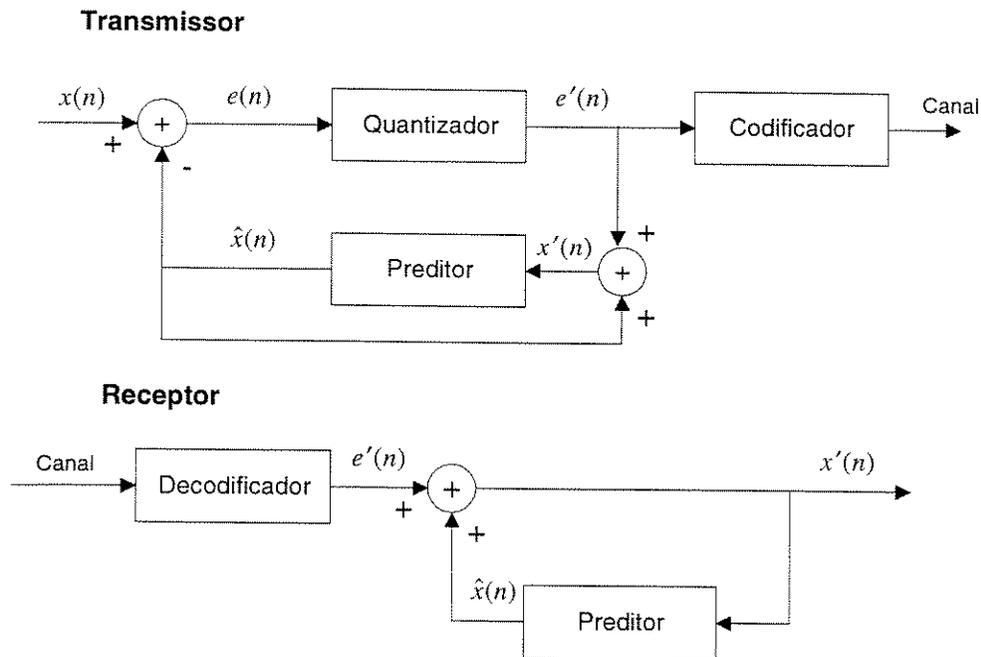


Fig. 2.1 - Diagrama de blocos de um sistema DPCM

O diagrama apresentado na Fig. 2.1 utiliza uma seqüência unidimensional como entrada. Entretanto, a extensão para o caso de seqüências bidimensionais (e.g., seqüências de vídeo) é simples. Por exemplo, $x(n)$ pode representar uma seqüência de varreduras horizontais de cada imagem transmitida. A Fig. 2.2 ilustra uma possível disposição de pixels (A, B, C, e D) utilizados na estimação do pixel X. Alguns exemplos típicos de preditores [8] são listados a seguir:

$$\hat{X} = 0.97A ,$$

$$\hat{X} = 0.5A + 0.5C ,$$

$$\hat{X} = 0.9A - 0.81B + 0.9C ,$$

$$\hat{X} = 0.75A - 0.5B + 0.75C ,$$

$$\hat{X} = A - B + C .$$

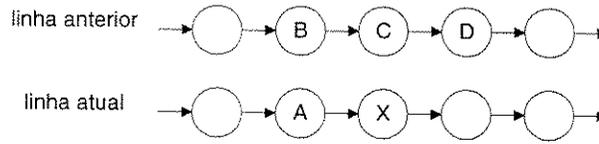


Fig. 2.2 - Disposição dos pixels utilizados pelo preditor

O conjunto de coeficientes do preditor pode ser fixo para todas as imagens (predição global), ou pode variar de imagem para imagem (predição local), ou ainda pode variar dentro de uma mesma imagem (predição adaptativa). Um critério muito utilizado [8,9] para se otimizar os coeficientes do preditor (para determinada imagem) é a minimização do erro quadrático médio, i. e., minimizar:

$$\text{MSE} = E \left\{ \left[x(m) - \sum_{i=0}^{m-1} c_i x(i) \right]^2 \right\} \quad (2.2)$$

Para sinais estacionários (pelo menos de 2ª ordem), isto é feito tornando o erro de predição ortogonal aos dados. Dessa forma, os coeficientes ótimos podem ser encontrados resolvendo o seguinte sistema de equações [8]:

$$E \left\{ \left(x(m) - \sum_{i=0}^{m-1} c_i x(i) \right) x(i) \right\} = 0, \quad i = 0, 1, \dots, m-1. \quad (2.3)$$

A expansão destas equações resulta em termos envolvendo valores de autocorrelação da imagem. A necessidade de se calcular os valores de autocorrelação para cada imagem inviabiliza a utilização de predição local para a maioria das aplicações em tempo real. Entretanto, o ganho de desempenho do preditor local sobre o global não é muito significativo [8]. Dessa forma, a predição global é utilizada na maioria das aplicações.

Outra importante etapa de um sistema DPCM é a quantização da imagem diferença, que influencia substancialmente a taxa de compressão alcançada pelo sistema. Essencialmente, um quantizador é uma função escada que mapeia os valores de entrada em um número finito de valores de saída (níveis do quantizador). A quantização invariavelmente envolve perda de informação. Esta perda de informação pode ser reduzida pelo número e posicionamento dos níveis de quantização, considerando-se a densidade de probabilidade do sinal de entrada. O quantizador mais simples é o uniforme, que tem as diferenças entre níveis de quantização vizinhos, e as diferenças entre limiares de decisão vizinhos, constantes. A

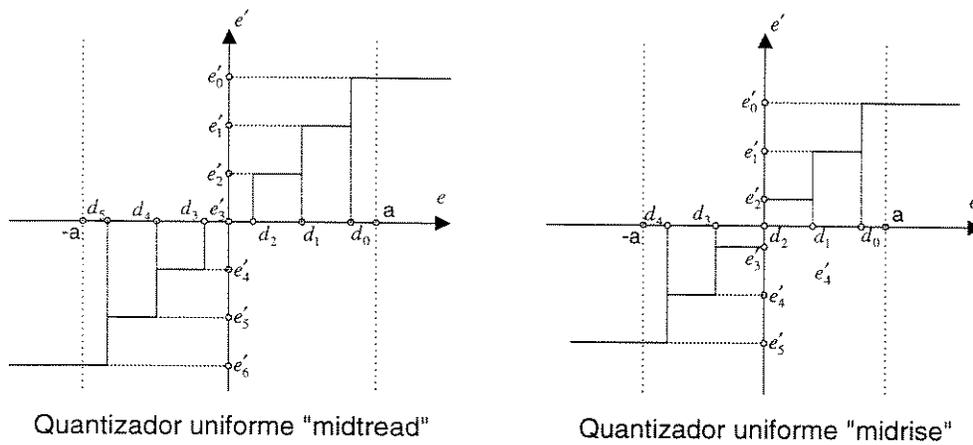


Fig. 2.3 - Quantizadores uniformes

Fig. 2.3 apresenta dois exemplos de quantizadores uniformes. Nestes exemplos, e , o sinal de entrada que pode assumir valores reais, é mapeado em valores discretos de saída, e'_i . Os limiares de decisão são identificados por d_i . No primeiro exemplo, o zero é um nível de saída, e por isso, este quantizador é chamado de "midtread". O segundo, é conhecido como quantizador "midrise", por ter o zero como limiar de decisão. Nestes dois exemplos, se o sinal de entrada não for limitado ao intervalo $[-a, a]$, ocorrerá saturação dos quantizadores quando e extrapolar este intervalo. Nestas situações o erro de quantização pode se tornar muito grande.

2.2.1 Otimização do quantizador

Fixado o número de níveis N e conhecendo-se a estatística do sinal de entrada, pode-se projetar um quantizador otimizado de acordo com algum critério. Um critério bastante utilizado é a minimização do MSE. Outro aspecto a ser considerado é se os níveis de saída serão codificados utilizando-se palavras-código de comprimento fixo ou variável (e.g., código de "Huffman"). No caso de palavras código de comprimento fixo, a taxa de bits é proporcional a $\log_2 N$. Se o critério utilizado é a minimização do MSE, o quantizador ótimo é conhecido como quantizador de Lloyd-Max [8,9].

Os níveis de reconstrução, e'_i , e os limiares de decisão, d_i , do quantizador de Lloyd-Max são encontrados resolvendo-se o seguinte conjunto de equações não lineares:

$$d_i = \frac{e'_{i-1} + e'_i}{2} \quad \text{com } i \in \{0, 1, \dots, N\} \quad (2.4)$$

$$e'_i = \frac{\int_{d_i}^{d_{i+1}} e \cdot p_e(e) de}{\int_{d_i}^{d_{i+1}} p_e(e) de} \quad \text{com } i \in \{0, 1, \dots, N\} \quad (2.5)$$

onde $p_e(e)$ é a densidade de probabilidade de e .

Para solucionar este conjunto de equações, pode ser utilizado o seguinte método iterativo [8]:

passo 1: é escolhido um conjunto arbitrário de valores d_i

passo 2: são calculados valores de e'_i de acordo com a Eq. (2.5)

passo 3: a partir dos valores encontrados no passo 2 para e'_i , são calculados valores de d_i de acordo com a Eq. (2.4). Volta-se ao passo 2 até que a diferença entre as duas aproximações esteja abaixo de um limiar preestabelecido.

Por outro lado, se as palavras código que representam os níveis de quantização são de comprimento variável, a taxa de bits é limitada pela entropia da saída do codificador. Neste caso, a otimização do quantizador é feita minimizando-se o MSE com restrições de entropia. Para um sinal de entrada de distribuição Laplaciana (que pode ser uma boa aproximação para uma imagem diferença típica) o quantizador uniforme resulta em bom desempenho. É interessante notar que, fixada uma taxa de bits, para uma distribuição Laplaciana e um grande número de níveis de quantização, a utilização de um quantizador uniforme, associado a codificação VLC (“variable length coding”), apresenta um ganho de desempenho [8] de até 5.6 dB sobre um sistema DPCM baseado no quantizador de Lloyd-Max. Além disso, a implementação de um quantizador uniforme é muito simples, podendo ser realizada através de uma divisão seguida de arredondamento em direção ao inteiro mais próximo. Para tornar ainda mais trivial esta implementação, pode-se restringir o tamanho do passo de quantização (diferença entre dois níveis de quantização vizinhos) a potências de dois. Dessa forma, a quantização equivale a um deslocamento de bits com descarte do(s) bit(s) menos significativo(s).

O sistema DPCM apresentado considera tanto o preditor quanto o quantizador fixos. É possível fazer as características do preditor e do quantizador variarem durante o processo de codificação. Estes sistemas adaptativos são conhecidos como ADPCM (“Adaptative DCPM”) [8,9], e apresentam um desempenho superior às custas de uma maior complexidade.

2.2.2 Estimação e compensação de movimento

No preditor descrito anteriormente, a estimativa de um pixel da imagem é baseada em outros pixels desta mesma imagem. Esta predição é conhecida como predição intraquadro (“intraframe prediction”). Quando se trata de uma seqüência de vídeo, a correlação temporal pode ser explorada utilizando-se pixels de outros quadros (quadros de referência) para predizer o pixel atual. Neste caso, tem-se predição interquadros (“interframe prediction”).

Um exemplo simples de predição interquadros é a utilização de um quadro anterior (quadro de referência) como estimativa do quadro atual. Dessa forma cada pixel do quadro atual é predito pelo pixel de mesma posição espacial do quadro de referência. Este método é eficiente em regiões da cena em que não há movimento. Entretanto, nas áreas onde há movimento a correlação entre pixels de mesma posição espacial geralmente é pequena. Para se obter melhores preditores são utilizadas técnicas de estimação e compensação de movimento.

As técnicas de estimação e compensação de movimento podem, ser divididas em quatro grupos principais:

- técnicas de gradiente,
- técnicas de pixel recursivas (“pel-recursive”),
- técnicas de “casamento” de blocos (“block matching”),
- técnicas no domínio da frequência.

Neste trabalho só serão discutidas técnicas de casamento de blocos, já que estas são as mais utilizadas em codificação de vídeo digital. Pode-se citar alguns padrões de compressão que recomendam, ainda que de forma implícita, a utilização destas técnicas: MPEG-2, H261, H263, entre outros.

A rigor, as técnicas de estimação de movimento, baseadas em casamento de blocos, não procuram estimar o deslocamento real dos objetos de uma cena, mas sim a variação do fluxo óptico (“optical flow”) [10]. Neste contexto, fluxo óptico é o campo associado às variações espaço-temporais de intensidade (de cada componente de cor, para o caso de imagens coloridas). Isto pode ser melhor compreendido através do exemplo ilustrado na Fig. 2.4.

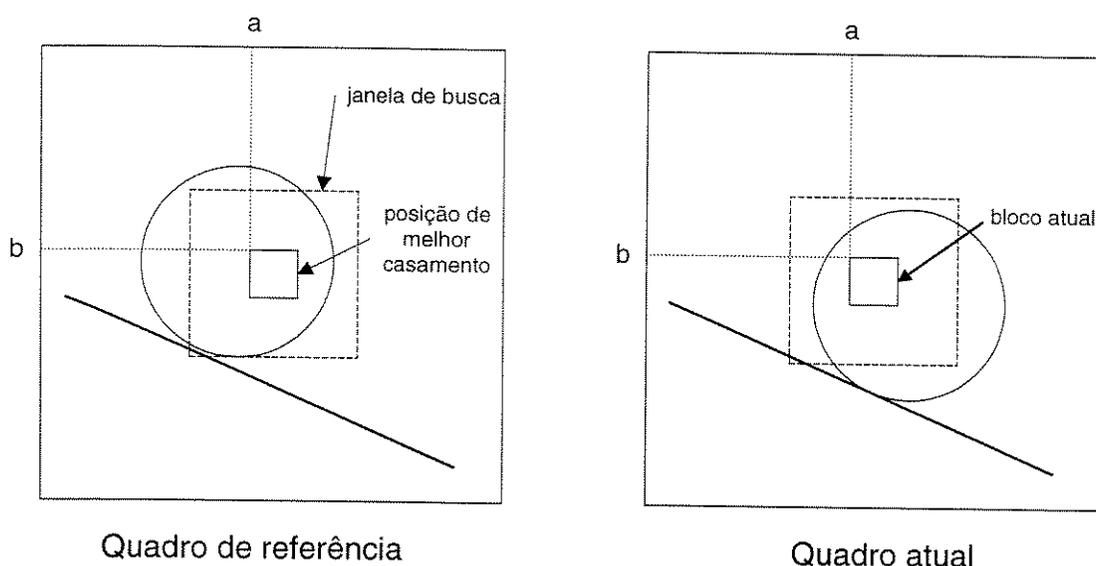


Fig. 2.4 - Falsa estimação de movimento

Neste exemplo são apresentados dois quadros de uma cena em que há um objeto, de cor constante, em movimento translacional. O cálculo de deslocamento do bloco de posição (a,b) resulta em deslocamento nulo, visto que a mesma região espacial do quadro anterior tem pixels com valores constantes, idênticos aos do bloco atual. Portanto, para este bloco, o deslocamento do objeto não foi detectado. Este fato não prejudica o desempenho do sistema de compressão, pois o que se deseja é obter um bom estimador, e não quantificar o deslocamento de cada objeto da cena.

Em cenas reais digitalizadas, as mudanças espaço-temporais da intensidade podem ocorrer devido a três principais causas. A primeira é o movimento da câmara de vídeo. Ainda que não ocorram movimentos na cena real, o movimento da câmara (“pan”, zoom, etc.) é captado como um deslocamento global da cena. A segunda causa são os movimentos intrínsecos dos objetos da cena (translações, rotações, deformações, etc.). A terceira causa são as mudanças nas condições de iluminação das cenas. Em geral, técnicas de compensação de movimento não consideram as mudanças nas condições de iluminação das cenas. Além disso, geralmente estas técnicas não fazem distinção entre deslocamentos locais e globais. Os deslocamentos globais são considerados na estimativa dos deslocamentos locais.

Na estimação e compensação de movimento, o termo movimento é utilizado em um sentido genérico, podendo significar fluxo óptico. Quando o bloco contém detalhes ou contornos, geralmente, o deslocamento estimado é aproximadamente o deslocamento real do

objeto mais o deslocamento global da cena. A Fig. 2.5 apresenta um exemplo em que isto ocorre. Neste exemplo, há um deslocamento da câmara para a direita (correspondendo à um deslocamento da cena para a esquerda), e um deslocamento do avião em relação à nuvem. No cálculo do deslocamento do bloco, \vec{d} , considera-se tanto o deslocamento da cena quanto o deslocamento do avião. Isto ocorre pois o vetor \vec{d} representa a diferença entre a posição do bloco do quadro de referência que propiciou o melhor casamento e a posição do bloco atual.

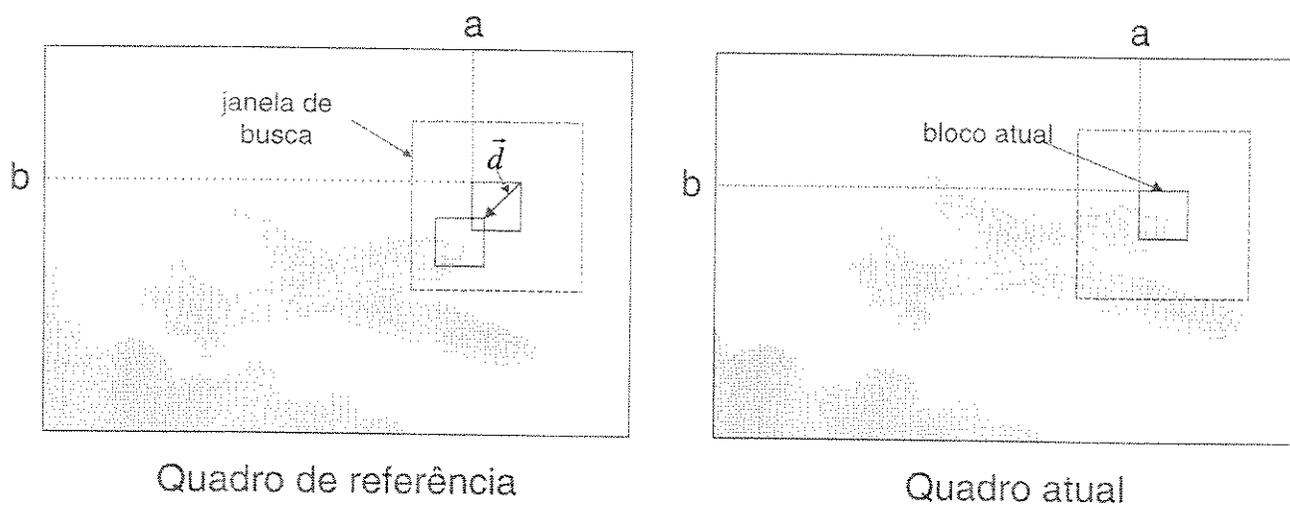


Fig. 2.5 - Estimação de movimento

Técnicas de casamento de blocos assumem como modelo que a imagem é composta de objetos rígidos com movimentos translacionais. Embora esse modelo seja restrito, ele é justificado pelo fato que movimentos complexos (e.g., deformação) podem ser decompostos em uma soma de movimentos translacionais. Estes algoritmos tem como objetivo minimizar uma medida de disparidade entre os blocos do quadro atual e do quadro de referência. Primeiro, o quadro atual é subdividido em blocos de tamanho $N \times M$ pixels. Em seguida, para cada bloco, é realizada uma busca no quadro de referência pela região que realize o melhor casamento, ou seja, que minimize uma medida de disparidade (distância métrica) entre os blocos dos dois quadros. Duas medidas de disparidade muito utilizadas [4,10,11] são: o erro quadrático médio (MSE - “Mean Square Error”), definido pela Eq. (2.6), e a média da diferença absoluta entre pixels (MAD - “Mean Absolute Difference”), definida pela Eq. (2.7).

$$MSE(i, j) = \frac{1}{M \cdot N} \sum_{m=a}^{M+a} \sum_{n=b}^{N+b} [Q_a(m, n) - Q_{ref}(m + i, n + j)]^2, \quad (2.6)$$

$$MAD(i, j) = \frac{1}{M \cdot N} \sum_{m=a}^{M+a} \sum_{n=b}^{N+b} |Q_a(m, n) - Q_{ref}(m+i, n+j)|, \quad (2.7)$$

onde, $Q_a(x, y)$ é o quadro atual,

$Q_{ref}(x, y)$ é o quadro de referência,

(i, j) é o deslocamento entre os dois blocos ($\vec{d} = (i, j)$ ótimo),

(a, b) é a posição do bloco atual.

Outra medida que pode ser utilizada para determinar o casamento de blocos é a correlação cruzada normalizada (NCC - “Normalized Cross-Correlation”), definida pela Eq. (2.8). Neste caso, deve-se maximizar (e não minimizar, como nos casos do MSE e do MAD) a figura de mérito (NCC) para se obter o melhor casamento. Apesar de apresentar um bom desempenho no sentido de encontrar o melhor casamento, esta medida não é muito utilizada em implementações práticas por apresentar complexidade maior que o MSE e o MAD.

$$NCC(i, j) = \frac{\sum_{m=a}^{M+a} \sum_{n=b}^{N+b} Q_a(m, n) \cdot Q_{ref}(m+i, n+j)}{\sqrt{\sum_{m=a}^{M+a} \sum_{n=b}^{N+b} Q_a^2(m, n)} \cdot \sqrt{\sum_{m=a}^{M+a} \sum_{n=b}^{N+b} Q_{ref}^2(m+i, n+j)}} \quad (2.8)$$

Entre as três medidas apresentadas, o MAD tem menor complexidade computacional. Por outro lado, se um sistema de compressão de vídeo digital tem como objetivo minimizar o MSE para uma determinada taxa de bits, pode ser vantajosa a utilização do MSE como figura de mérito na etapa de estimação de movimento.

Em grande parte das implementações práticas, a busca pelo melhor casamento em todo o quadro de referência (FS - “Full Search”) apresenta uma complexidade que poderia inviabilizar a utilização deste método. Além disso, geralmente, o deslocamento de um objeto entre o quadro atual e o quadro de referência é pequeno. Limita-se então a área de busca a uma janela centrada no bloco atual. Por exemplo, se o bloco tem dimensões de 16 x 16 pixels e o deslocamento máximo assumido é de ± 8 pixels, tem-se uma janela de busca de tamanho 32 x 32 pixels. Na escolha do tamanho da janela de busca deve-se considerar os

deslocamentos máximos dos objetos entre o quadro de referência e o quadro atual. Quanto maior for a distância temporal entre estes quadros, maior deve ser a janela de busca.

Mesmo limitando-se a busca a uma determinada região, o número de operações efetuadas é considerável. Para se diminuir ainda mais a complexidade do processo de busca, pode-se utilizar vários algoritmos disponíveis na literatura [4,10-12]. Em geral, estes algoritmos são baseados na hipótese que a superfície de erro é unimodal, ou seja, o erro cresce monotonicamente à medida que a procura se distancia da posição de erro mínimo global. Na verdade, esta hipótese nem sempre é verdadeira. Entretanto, ela serve como uma boa aproximação e estes algoritmos apresentam desempenhos próximos àquele obtido quando se utiliza busca completa (FS). Alguns destes algoritmos são apresentados a seguir.

2.2.2.1 Busca logarítmica

O algoritmo de busca logarítmica bidimensional foi apresentado por Jain e Jain [35] em 1981. Neste algoritmo, a busca é baseada na direção de distorção mínima. A métrica de distorção utilizada é o MSE, pois, de acordo com este artigo [35], o MSE apresenta melhores resultados que a correlação cruzada quando a área é pequena e o movimento não é de translação. No primeiro passo, são comparadas cinco posições, uma central (posição do bloco atual), e quatro posições ortogonais equidistantes. No passo seguinte, o ponto que apresentou o menor MSE torna-se a posição central. Caso o menor MSE corresponda à posição central, a distância entre as posições comparadas é reduzida por um fator de 2. Este processo se repete até que a distância seja igual a 1 pixel. A Fig. 2.6 ilustra um exemplo de busca logarítmica 2-D em que o vetor de deslocamento encontrado foi $\vec{d} = (3,-5)$.

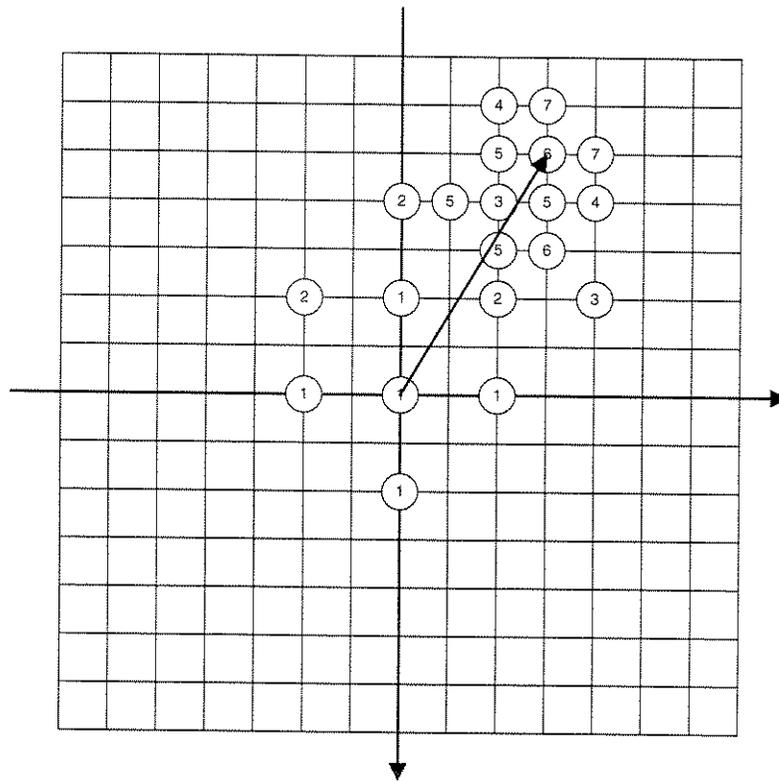


Fig. 2.6 - Busca logarítmica 2-D

2.2.2.2 Busca em três passos

O algoritmo de busca em três passos (TSS – “Three-Steep Search”) foi publicado por Koga, et. al.[36], quase simultaneamente com o artigo de Jain e Jain. Neste trabalho, o MAD foi utilizado como medida de distorção. No primeiro passo, são feitas comparações considerando nove posições, dispostas como apresentado no exemplo da Fig. 2.7. No segundo passo, a posição que apresentou o menor MAD é tomada como posição central, e a distância entre as novas posições restantes é dividida por 2. Em seguida, fazem-se novas comparações. No terceiro passo, que é semelhante ao segundo, é encontrado o ponto de melhor casamento (menor MAD). No exemplo da Fig. 2.7, o vetor de deslocamento encontrado foi $\vec{d} = (3,-5)$.

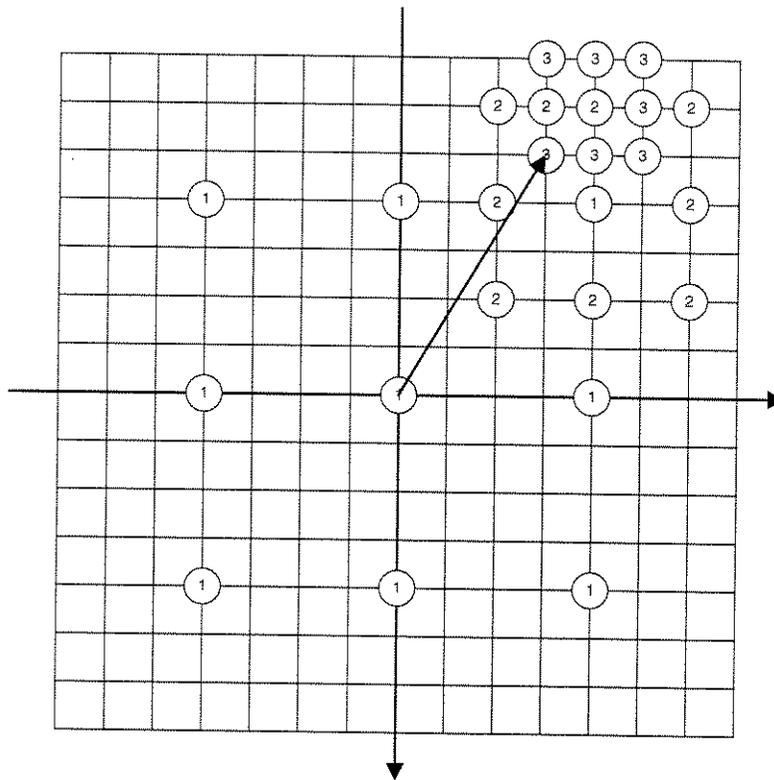


Fig. 2.7 - Busca em três passos

2.2.2.3 Busca em direções conjugadas

A busca em direções conjugadas (CDS – “Conjugated Direction Search”) procura por um mínimo local, primeiro na horizontal e, em seguida, na vertical. Um exemplo deste algoritmo é apresentado na Fig. 2.8 no qual foi encontrado um deslocamento $\vec{d} = (3,-5)$.

Entre os três algoritmos apresentados, este é o que possui maior eficiência computacional, pois realiza a busca analisando um número menor de posições. Entretanto ele possui um desempenho inferior na busca pelo mínimo global, quando comparado aos outros dois algoritmos.

Uma variação deste algoritmo é o OTS (“One-at-a-Time Search”) que alterna buscas na horizontal e na vertical até que seja encontrado um ponto de mínimo em relação as posições ortogonais vizinhas.

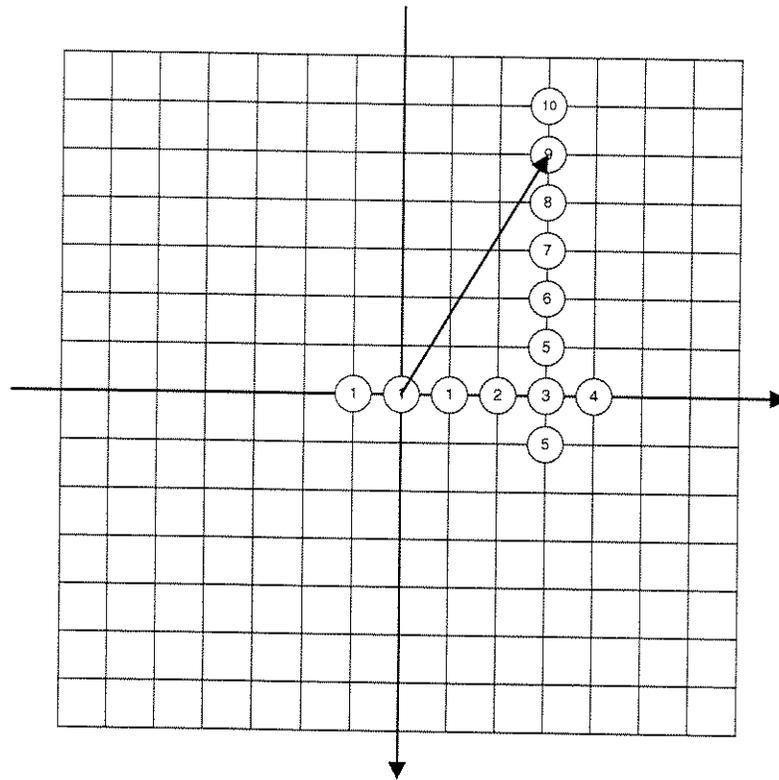


Fig. 2.8 - Busca em direções conjugadas

2.2.2.4 Busca telescópica

A busca telescópica se baseia na continuidade temporal do movimento. Os vetores de deslocamento de um quadro são utilizados para prever os vetores de deslocamento de quadros vizinhos. Esta técnica pode ser implementada de diferentes maneiras. Um exemplo é apresentado na Fig. 2.9. Considerem-se três quadros de uma seqüência de vídeo, sendo que o primeiro, Q_{ref} , serve como referência na predição dos seguintes, P1 e P2. Dado que para o bloco de P1, de posição (a,b), foi encontrado um vetor de deslocamento \vec{d} , o bloco de mesma posição espacial em P2 terá o vetor $2\vec{d}$ como estimativa inicial do vetor de deslocamento. A multiplicação de \vec{d} por um fator de 2 deve-se ao fato que P2 está a uma distância temporal duas vezes maior de Q_{ref} do que P1. Dessa forma, a janela de busca é deslocada de $2\vec{d}$ e inicia-se a busca pelo melhor casamento. Neste caso, o tamanho da janela de busca pode ser o mesmo que o utilizado em P1. Se a busca telescópica não fosse utilizada seria necessária uma janela de busca maior, já que P2 é temporalmente mais distante de Q_{ref} do que P1. Portanto,

através desta técnica pode-se diminuir a área de busca. Entretanto, ela fornece apenas uma estimativa inicial para o vetor de deslocamento.

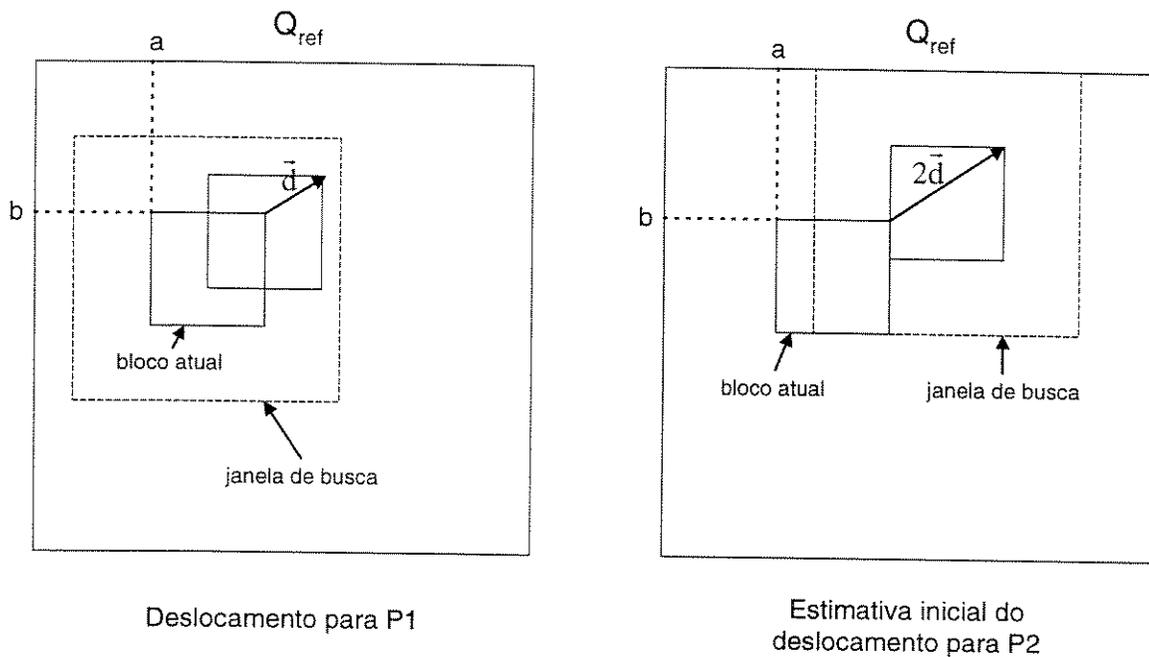


Fig. 2.9 - Busca telescópica

2.3 Codificação por Transformadas Ortogonais

Como já foi apresentado, um modo de se reduzir a correlação entre pixels de uma imagem no domínio espacial é a utilização de um sistema DPCM. Outra forma de se reduzir esta correlação é a utilização de transformadas ortogonais. Em geral, a codificação por transformadas ortogonais apresenta um desempenho superior ao de um sistema DPCM, para imagens comprimidas a taxas de 2 ou 3 bits por pixel [4].

Um esquema geral de codificação por transformadas ortogonais envolve a subdivisão da imagem em blocos de $N \times N$ pixels aos quais é aplicada uma transformação, que consiste em operação linear reversível. Os blocos de pixels são convertidos do domínio espacial para o domínio da transformada. Neste processo, consegue-se descorrelacionar o sinal original e redistribuir sua energia de maneira que poucos coeficientes concentrem grande parte desta energia. Desta forma, alguns coeficientes podem ser descartados no processo de quantização. Além disso, pode-se considerar fatores de percepção do olho humano na quantização dos coeficientes, objetivando melhor qualidade subjetiva da imagem reconstruída.

Quando a transformada é aplicada em blocos de $N \times N$ pixels ela é chamada de bidimensional (transformada 2-D). Se ela é aplicada a linhas ou colunas do bloco (ou seja, um conjunto de N pontos) é uma transformada unidimensional (1-D). Uma transformada bidimensional é separável se o seu “kernel” também é separável. Neste caso, a transformada 2-D pode ser implementada primeiro aplicando-se transformações 1-D de N pontos às linhas e, em seguida, às colunas.

Na codificação por transformadas ortogonais, um bloco de $N \times N$ pixels é representado por funções de base ortogonais, ponderadas por um conjunto de coeficientes. A transformação é unitária se as funções de base da transformada são ortonormais. Neste caso, a energia do sinal é conservada após a transformação.

Considere uma linha de n pixels de um bloco como um ponto em um espaço n -dimensional, sendo que cada pixel do bloco representa a posição do ponto em relação a um dos eixos cartesianos deste espaço. A transformação unitária pode ser interpretada como uma rotação n -dimensional dos eixos cartesianos deste espaço. Como exemplo, considere que uma imagem típica é dividida em conjuntos de pares de pixels vizinhos (x_1, x_2) . Dessa forma, tem-se um conjunto de vetores X que podem ser dispostos em um espaço bidimensional, onde x_1 é o eixo horizontal, e x_2 o eixo vertical. A Fig. 2.10 apresenta a região que concentra o maior número destes vetores. Esta concentração se deve à correlação entre pixels vizinhos. Uma transformação unitária que converte os vetores X em vetores Y é dada pela Eq. (2.9). Esta transformação corresponde a uma rotação de 45° dos eixos cartesianos x_1 e x_2 , como apresentado na Fig. 2.10. Antes da transformação a energia do sinal é aproximadamente igualmente distribuída nas direções x_1 e x_2 . Após a transformação ocorre uma concentração da energia ao longo da direção y_1 (ou seja, x_1 rotacionado de 45°). Este é um exemplo de empacotamento de energia (“energy packing”).

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (2.9)$$

ou

$$Y = T \cdot X \quad (2.10)$$

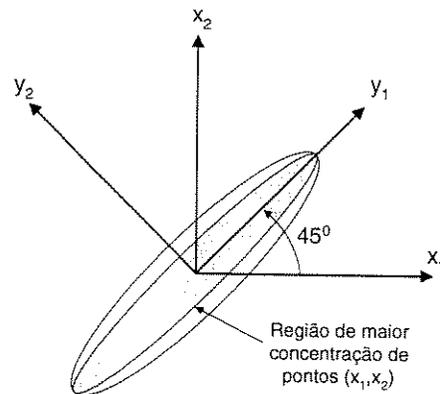


Fig. 2.10 - Exemplo de transformação unitária de 2 pontos

A transformação inversa é dada pela Eq. (2.11). É interessante notar que, para transformadas unitárias $T^{-1} = T^{*T}$. Portanto se as funções de base são reais, como é o caso deste exemplo, $T^{-1} = T^T$.

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (2.11)$$

As três características básicas desejáveis em uma transformada, quando ela é utilizada no processo de compressão de imagens, são as seguintes:

- descorrelacionar a imagem;
- possuir funções de base independentes da imagem;
- algoritmos de implementação rápida.

A seguir são apresentadas algumas transformadas ortogonais utilizadas em processamento e codificação de imagens e vídeo.

2.3.1 Transformada de Karhunen-Loève

A transformada de Karhunen-Loève (KLT - “Karhunen-Loève Transform”) descorrelaciona completamente as amostras do sinal de entrada e realiza um empacotamento ótimo da energia deste sinal. Ela se caracteriza por possuir como funções de base os autovetores da matriz de correlação do sinal. Portanto, ela é dependente do sinal de entrada, o que prejudica sua utilização em sistemas de compressão de imagens em tempo real. Além disso, só existem algoritmos KLT rápidos se o sinal for um processo autoregressivo de

primeira ordem com condições de contorno fixadas, pois, neste caso, a transformada KLT é igual à transformada discreta do seno [37], para a qual existem algoritmos rápidos.

2.3.2 Transformada discreta de Fourier

A transformada discreta de Fourier (DFT – “Discrete Fourier Transform”) tem como funções de base senos e cosenos de diferentes frequências e, por isto, é interpretada como uma decomposição espectral da imagem original. Sendo assim, a DFT é bastante utilizada em processos de filtragem e análise espectral.

Para um bloco de $n \times n$ pixels, a DFT direta é definida pela Eq. (2.12), e sua inversa pela Eq. (2.13).

$$F(u, v) = \frac{1}{n} \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} f(x, y) \exp\left\{\frac{-2\pi j(ux + vy)}{n}\right\} \quad (2.12)$$

$$f(x, y) = \frac{1}{n} \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} F(u, v) \exp\left\{\frac{2\pi j(ux + vy)}{n}\right\} \quad (2.13)$$

onde $j = \sqrt{-1}$

Como o “kernel” da DFT 2-D é separável, esta transformada pode ser implementada aplicando-se DFT’s 1-D nas linhas e nas colunas do bloco. Vários algoritmos de implementação rápida (FFT – “Fast Fourier Transform”) existem na literatura. Em geral, estes algoritmos tem uma ordem de complexidade de $n \cdot \log_2(n)$ para uma transformada de n pontos.

À DFT apresenta duas desvantagens em aplicações de codificação de imagens. A primeira é que os coeficientes da transformada são números complexos, o que dificulta o armazenamento e a manipulação destes coeficientes. A segunda, é o aparecimento de frequências espectrais espúrias, geradas devido à periodicidade implícita dos blocos da imagem. Uma alternativa para solucionar este problema é a utilização da transformada discreta do coseno (DCT – “Discrete Cosine Transform”), que é apresentada na seção seguinte.

2.3.3 Transformada discreta do coseno

A DCT 2-D é definida pela Eq. (2.14), e sua inversa pela Eq. (2.15).

$$F(u, v) = \frac{2C(u)C(v)}{n} \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} f(x, y) \cos\left\{\frac{(2x+1)u\pi}{2n}\right\} \cos\left\{\frac{(2y+1)v\pi}{2n}\right\} \quad (2.14)$$

$$f(x, y) = \frac{2}{n} \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} C(u)C(v)F(u, v) \cos\left\{\frac{(2x+1)u\pi}{2n}\right\} \cos\left\{\frac{(2y+1)v\pi}{2n}\right\} \quad (2.15)$$

onde $C(u), C(v) = \begin{cases} 1/\sqrt{2} & \text{para } u, v = 0 \\ 1 & \text{para } u, v \neq 0 \end{cases}$.

Esta transformada soluciona os dois principais problemas apresentados pela DFT para aplicações em compressão de imagens: os coeficientes são reais, e não há geração de componentes espectrais espúrias. Além disso, para imagens típicas, a DCT apresenta desempenho próximo ao da transformada ótima (KLT) e possui algoritmos de implementação rápida de mesma ordem de complexidade que a FFT. Por estes motivos, a DCT tem sido a transformada ortogonal mais utilizada em sistemas de compressão de imagens e vídeo.

Para ilustrar o problema de aparecimento de componentes espectrais espúrias, considere uma seqüência de n pontos. No cálculo da DFT desta seqüência há uma periodicidade implícita da mesma, como apresentado na Fig. 2.11a. Esta periodicidade é resultado da amostragem no domínio da freqüência e produz o aparecimento de bordas. Para representação destas bordas, são necessárias componentes de alta freqüência que não fazem parte do sinal original (freqüências espúrias). A fim de se evitar estas bordas, a seqüência original pode ser duplicada espelhando-a em relação ao eixo vertical, como apresentado na Fig. 2.11b. O cálculo da DFT desta nova seqüência é equivalente ao cálculo da DCT da seqüência original. Por isto, ao contrário da DFT, a DCT não gera componentes espúrios de freqüência.

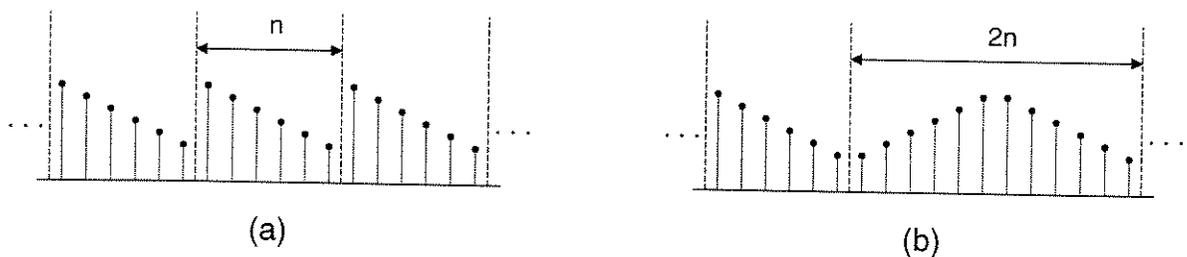


Fig. 2.11 - Periodicidade implícita da DFT e da DCT

2.3.4 Transformada de Walsh-Hadamard

A maior vantagem da transformada de Walsh-Hadamard (WHT – “Walsh-Hadamard Transform”) é a sua simplicidade de implementação, já que as funções de base são

constituídas de valores +1 e -1. Por outro lado, a WHT possui uma capacidade de empacotamento de energia consideravelmente inferior à da DCT, para imagens típicas. A matriz de Hadamard de 2 x 2 pontos é dada pela Eq. (2.16). Para se obter matrizes maiores pode ser utilizado um processo recursivo dado pela Eq. (2.17), onde H_{2n} é uma matriz de ordem $2n \times 2n$.

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (2.16)$$

$$H_{2n} = \frac{1}{\sqrt{2}} \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix} \quad (2.17)$$

2.3.5 Estratégias de Codificação

A operação de transformação, se considerada isoladamente, não representa nenhuma compressão. Os benefícios que ela proporciona são a decorrelação das amostras e o empacotamento da energia, que permitem que a codificação seja mais eficiente. Duas são as estratégias mais utilizadas para selecionar os coeficientes a serem codificados:

- seleção de amostras por zona (“zonal sample selection”);
- seleção de amostras por limiar (“threshold sample selection”).

A seleção de amostras por zona consiste em preservar os coeficientes localizados em uma zona previamente especificada do bloco transformado, e tornar nulos todos os demais coeficientes. Como as imagens típicas apresentam uma característica espectral passa baixas, geralmente os coeficientes de alta frequência são descartados.

Quando se utiliza a estratégia de seleção de amostras por zona, podem existir coeficientes fora da zona preestabelecida que contenham um parte significativa da energia do sinal. Para solucionar este problema, utiliza-se a estratégia de seleção por limiar. Nesta técnica, são preservados os coeficientes cujos valores, em módulo, sejam superiores a um determinado limiar. A desvantagem desta técnica é que os endereços (posições dentro do macrobloco) dos coeficientes selecionados também devem ser transmitidos, já que estes endereços variam de bloco para bloco. Uma variação desta técnica é enviar os L coeficientes mais significativos de cada bloco.

O próximo passo, para qualquer uma das duas estratégias de seleção, é a quantização e a codificação dos coeficientes selecionados. Pode-se utilizar o mesmo quantizador para todos os coeficientes, entretanto, a eficiência do processo é maior se o passo de quantização, ou seja

o número de bits alocados, varia para cada coeficiente. Se a alocação de bits visa minimizar o erro de quantização total, o número de níveis do quantizador é proporcional à variância do coeficiente. Neste caso, o espectro do erro é, idealmente, plano.

Na Fig. 2.12 são apresentados gráficos de dois exemplos práticos [9] de alocação de bits. Os eixos horizontais representam coeficientes espectrais (k), e os eixos verticais representam o logaritmo na base 2 da variância de cada um destes coeficientes ($\log_2 \sigma_k^2$). Os limiares K_k definem o número de bits alocados para cada coeficiente (R_k), e são espaçados entre si de 6 dB. Estes limiares são deslocados para cima ou para baixo até que se encontre a taxa de bits total desejada. No exemplo da Fig. 2.12a, o espectro do erro de quantização é aproximadamente plano pois, neste caso, a alocação de bits visa minimizar o erro médio quadrático de quantização total. Por outro lado, no exemplo da Fig. 2.12b, são considerados fatores de percepção visual humana, e a cada coeficiente é associado um fator de ponderação, W_k de acordo com sua importância. Desta forma, o espectro do erro de quantização é equalizado de acordo com os valores W_k , como apresentado na Fig. 2.12b.

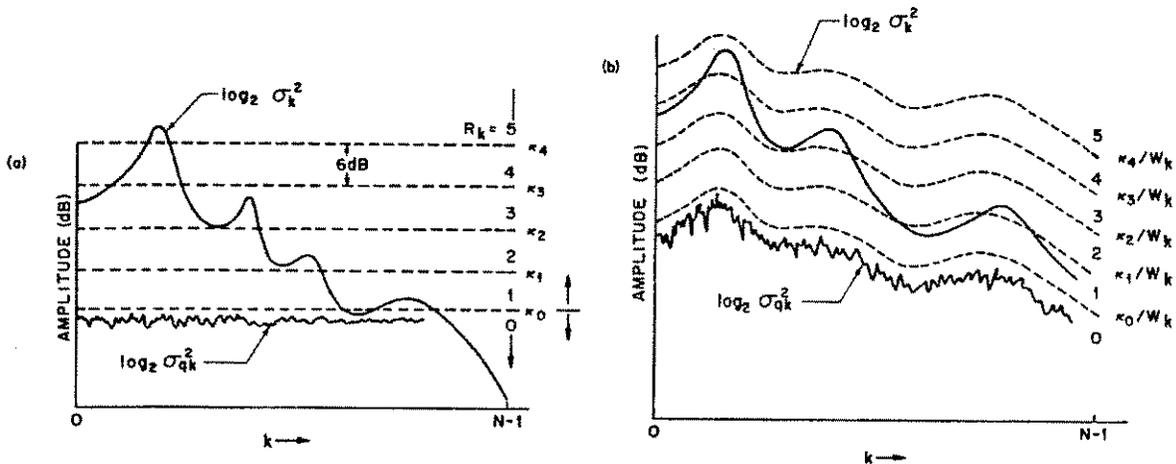


Fig. 2.12 - Exemplos de alocação de bits (extraído de [9])

Na codificação dos coeficientes são utilizadas técnicas de codificação de entropia, por exemplo códigos de Huffman e corrida de zeros (“runlength coding”). Estas técnicas são apresentadas na Seção 2.6.

2.4 Codificação por Subbandas

Nesta técnica, a imagem original é processada em um banco de filtros que gera um conjunto de imagens, chamadas subbandas. Cada subbanda é limitada à uma faixa de

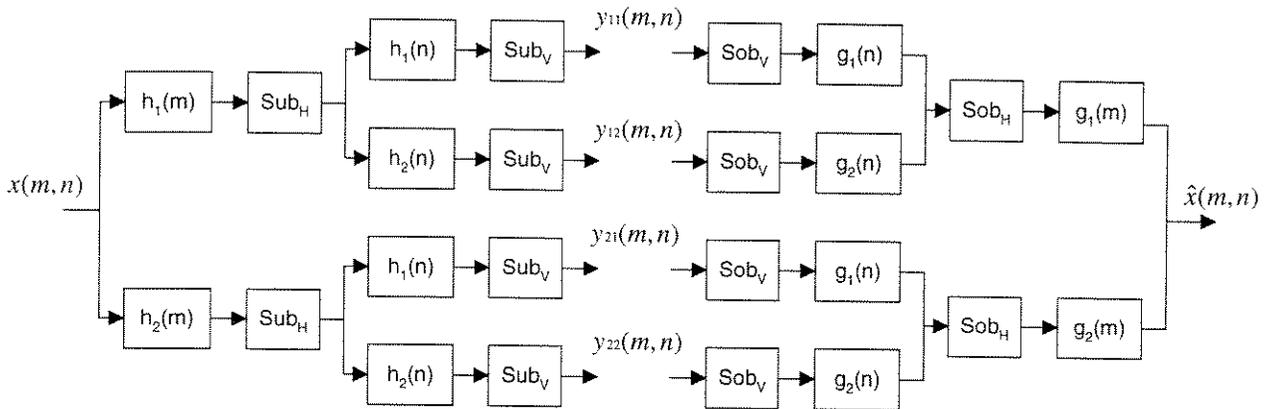


Fig. 2.13 - Diagrama geral de um sistema de codificação por subbandas

freqüências espaciais, e por isso, pode ser subamostrada. O processo de filtragem e subamostragem é chamado de análise. Em seguida, as subbandas são codificadas e enviadas. No decodificador, as subbandas decodificadas são sobreamostradas e passam por uma filtragem inversa à aplicada no processo de análise. As subbandas filtradas são somadas e a imagem é reconstruída. O processo de reconstrução da imagem é chamado de síntese. A Fig. 2.13 apresenta o diagrama geral de um exemplo no qual uma imagem é subdividida em quatro subbandas. Neste diagrama, os filtros de análise, $h_1(n)$ e $h_2(n)$, são passa baixas e passa altas, respectivamente, e $g_1(n)$ e $g_2(n)$ são os filtros de síntese correspondentes. Se o número de subbandas desejado é maior que 4, o processo de análise pode ser aplicado repetidamente à uma ou mais subbandas.

A segmentação da imagem em subbandas, assim como a aplicação de transformadas ortogonais, não representa nenhuma compressão. Entretanto, a cada subbanda podem ser aplicadas diferentes técnicas de compressão (e.g., quantização vetorial ou DPCM) de maneira mais eficiente do que se estas técnicas fossem diretamente aplicadas na imagem original.

No processo de filtragem, freqüentemente, utilizam-se filtros espelhados em quadratura (QMF's – "Quadrature Mirror Filters"), que permitem uma reconstrução sem "aliasing", quando não há erros de transmissão. Esta reconstrução sem "aliasing" é possível porque os filtros de análise e síntese são projetados de modo que o "aliasing" introduzido por um dos filtros seja exatamente cancelado pelo "aliasing" introduzido pelo seu correspondente. A Fig. 2.14 apresenta um par de filtros QMF's de análise ideais. Outros tipos de filtros podem também ser utilizados, como por exemplo, os filtros de quadratura conjugados (CQF's – "Conjugate Quadrature Filters").

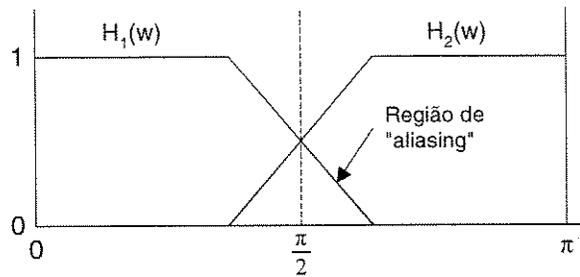


Fig. 2.14 - Par de filtros espelhados em quadratura

2.4.1 Analogia entre codificação por subbandas e codificação por transformadas

Existe uma analogia entre a divisão de uma imagem em subbandas e aplicação de uma transformação a segmentos desta imagem, que é utilizada na codificação por transformadas. Nas duas técnicas a imagem é representada no domínio da frequência. Basicamente, a diferença entre estas técnicas é a maneira como os dados de saída são organizados.

Esta analogia pode ser melhor compreendida através de um exemplo. Considere uma transformada (e.g., DFT ou DCT) que é aplicada a blocos de tamanho 2×2 de uma imagem. O resultado desta transformação é apresentado na Fig. 2.15a, onde os coeficientes a_n são os de mais baixa frequência e os d_n , os de mais alta frequência. O índice n está relacionado com a posição espacial do coeficiente. Agora considere que a imagem original é dividida em quatro subbandas, através de filtragem 2×2 , seguida de subamostragem nas duas direções, horizontal e vertical, por um fator de 2. As subbandas resultantes são apresentadas na Fig. 2.15b. Os filtros de análise utilizados podem ser escolhidos de forma que a divisão em subbandas corresponda exatamente à aplicação da transformada [8], com a diferença que os coeficientes são agrupados de maneira diferente.

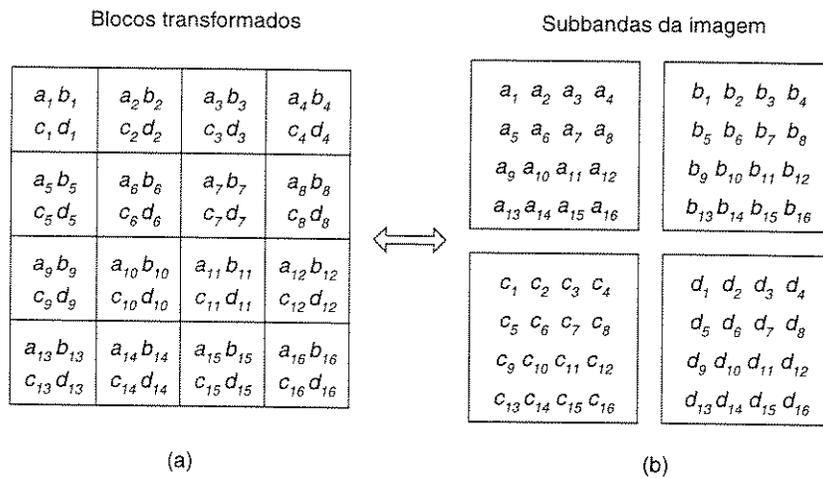


Fig. 2.15 - Relação entre transformada e subbandas

No exemplo apresentado, os coeficientes de mais baixa frequência de cada bloco transformado podem ser agrupados de maneira a constituírem a subbanda de mais baixa frequência da Fig. 2.15b, composta por elementos a_n . Da mesma forma, os outros coeficientes dos blocos podem ser agrupados de forma a constituírem as outras três subbandas. O processo inverso também é possível, isto é, pode-se obter cada um dos blocos transformados agrupando-se os coeficientes de mesma posição espacial de cada subbanda. A equivalência entre estas técnicas permite algumas observações. Por exemplo, a estratégia de seleção de amostras por zona, utilizada na codificação por transformadas, corresponde a selecionar determinadas subbandas para serem codificadas.

O exemplo apresentado na Fig. 2.15 é relativamente simples, entretanto, quando o filtro de análise tem um comprimento maior que o fator de dizimação (e.g., um filtro de 5 taps e um fator de dizimação de 2 em cada direção), a busca por uma transformação equivalente é mais complexa, e devem ser utilizadas transformadas LOT (“Lapped Orthogonal Transform”), nas quais há sobreposição dos blocos transformados.

2.5 Quantização Vetorial

Na Seção 2.2, que trata de codificação preditiva, foi introduzido o conceito de quantização escalar. Este conceito, que é aplicável a um elemento, pode ser generalizado para o caso de quantização vetorial (VQ – “Vector Quantization”). Na quantização vetorial um grupo de elementos é quantizado ao mesmo tempo, através da representação destes por um vetor-código.

Um exemplo de quantização vetorial é apresentado na Fig. 2.16, onde um conjunto de vetores bidimensionais \vec{x} , uniformemente distribuídos na região sombreada do gráfico, são representados por vetores-código \vec{x}'_i , $i = 0, \dots, 9$, que compõem um “codebook”. As linhas tracejadas delimitam as regiões de decisão. No centro de cada região de decisão se encontra o vetor representativo de todos os outros vetores que pertencem a esta região. Como os vetores deste exemplo tem uma distribuição uniforme e estão dispostos em uma região retangular, a utilização de quantização escalar, aplicada individualmente a cada elemento do vetor, teria um desempenho idêntico. Entretanto, geralmente a quantização vetorial apresenta melhor desempenho que a escalar, a custo de maior complexidade. Neste exemplo, foram utilizados vetores bidimensionais, mas podem também ser utilizados vetores multidimensionais.

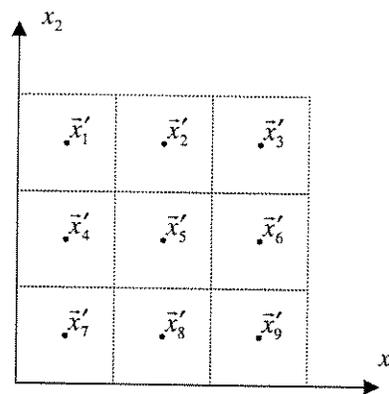


Fig. 2.16 - Exemplo de quantização vetorial

No caso de codificação de imagens, a imagem é decomposta em vetores n -dimensionais, por exemplo, dividindo-se a imagem em regiões de 3×3 pixels, ou seja, em vetores de dimensão 9. A quantização vetorial também pode ser aplicada em vetores de erros de predição, ou de coeficientes resultantes de alguma transformação. Cada vetor é comparado com um conjunto de vetores-código que compõem o “codebook”, previamente gerado. Nesta comparação, utiliza-se alguma regra de distorção mínima, e.g. MSE, para encontrar o vetor-código que melhor representa cada vetor da imagem. Isto equivale a verificar à qual região de decisão pertence cada vetor. O próximo passo é o envio dos índices dos vetores-código. No decodificador, os vetores-código são recuperados através de uma “look-up table”. Quanto à complexidade, este processo é bastante assimétrico, já que a decodificação é muito mais simples que a codificação.

O “codebook” pode ser otimizado para cada imagem, ou podem ser utilizados “codebooks” globais. No primeiro caso, geralmente o ganho de compressão é maior. Entretanto, a complexidade também é maior, já que deve-se gerar um “codebook” para cada imagem. Além disso, deve-se enviar um “codebook” para cada imagem, e isto representa “overhead” adicional. Os “codebooks” globais são gerados utilizando-se um conjunto de imagens de treinamento que representam bem o universo de imagens que serão codificadas. Um algoritmo muito utilizado para gerar VQ “codebooks” é o algoritmo de Linde-Buzo-Gray (LBG), que é uma generalização do algoritmo de Lloyd-Max para quantização escalar, apresentado na Seção 2.2.1.

2.6 Codificação de Entropia

As técnicas de codificação de entropia buscam retirar a redundância estatística do sinal. O conjunto mais importante destas técnicas é o que utiliza códigos de comprimento variável (VLC - "Variable length coding"), e.g. códigos de Huffman. Nos códigos de Huffman utilizam-se palavras-código binárias para representar cada símbolo da fonte. O comprimento da palavra-código atribuída a cada símbolo é proporcional ao logaritmo na base 2 do inverso da probabilidade de ocorrência deste símbolo ($\log_2(1/p(s_n))$), onde $p(s_n)$ é a probabilidade de ocorrência do símbolo s_n). Desta forma, quanto maior a frequência de ocorrência do símbolo, menor será o comprimento da palavra-código atribuída a ele.

A codificação “runlength” é uma técnica de codificação de entropia que apresenta um bom desempenho quando um determinado símbolo ocorre em seqüência muito frequentemente. Ao invés de se codificar cada um dos símbolos de uma seqüência individualmente, codifica-se um número relativo a quantidade de repetições consecutivas deste símbolo. Vários padrões de codificação de imagens e vídeo utilizam esta técnica, por exemplo o JPEG e o MPEG. Nestes padrões a codificação “runlength” é aplicada em conjunto com codificação de Huffman na compressão dos coeficientes gerados pela transformação DCT. Após a quantização vários destes coeficientes tornam-se nulos, o que viabiliza a utilização da codificação “runlength.” Os coeficientes espectrais estão inicialmente dispostos em blocos de 8x8 pixels. Estes coeficientes são ordenados em zig-zag, aproximadamente de acordo com sua importância psico-visual, como apresentado na Fig. 2.17a. Esta ordem de varredura é utilizada pelos padrões JPEG, MPEG-1, MPEG2, entre outros. O Padrão MPEG-2 também possibilita outro tipo de ordenação, apresentado na Fig. 2.17b. Nesta ordenação, as

freqüências verticais são privilegiadas, em detrimento das freqüências horizontais. A utilização desta varredura alternada (“alternate scan”) se justifica em alguns casos, especialmente quando as linhas pares e ímpares do bloco original pertencem a campos diferentes (quando o quadro original é entrelaçado). Neste caso, quando há deslocamento na cena, ocorre o aparecimento de altas freqüências na vertical devido a distância temporal entre os dois campos.

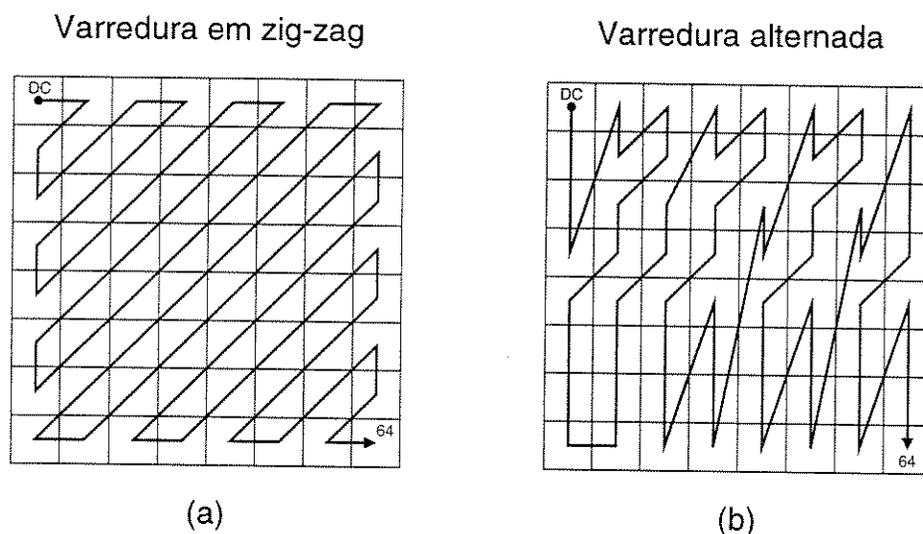


Fig. 2.17 - Modos de varredura dos coeficientes do bloco transformado

2.7 Codificação Hierárquica

A codificação hierárquica é uma classe de técnicas em que a imagem é codificada em vários níveis de qualidade ou resolução. O termo hierárquica é utilizado pois os dados referentes à imagem são organizados em ordem de importância, ou seja, em vários níveis de hierarquia, onde cada nível corresponde a uma imagem reconstruída a um determinado nível de qualidade ou resolução. Nas técnicas que utilizam codificação hierárquica, algumas das características desejáveis são listadas a seguir:

- a reconstrução da imagem em determinado nível deve aproveitar ao máximo as informações já transmitidas para os níveis inferiores;
- o escalonamento dos dados em níveis de hierarquia deve prejudicar o mínimo possível o ganho de compressão, em relação ao obtido quando não há o compromisso de se possibilitar a reprodução da imagem em vários níveis de qualidade ou resolução;

- devem existir algoritmos de codificação e decodificação relativamente rápidos e de fácil implementação em hardware.

Uma das aplicações para esta técnica é a busca por determinada imagem, em um banco de imagens. Primeiro, uma versão de baixa qualidade da imagem é transmitida. Em seguida, progressivamente mais dados são enviados de forma a melhorar a qualidade da imagem, até que o usuário possa identificá-la, e decidir se continua a recebê-la (se for a imagem desejada) ou a descarta (caso contrário). Este esquema é chamado de transmissão progressiva, e não há variação de resolução espacial entre as versões da imagem. O fato do usuário poder interromper a qualquer instante a transmissão da imagem representa uma redução dos dados efetivamente transmitidos. Esta redução é chamada de compressão efetiva (“effective compression”), e é uma importante característica da transmissão progressiva. Outra característica deste esquema é que a quantidade de dados, necessária para se passar de um nível para o próximo, é aproximadamente constante.

Outra aplicação importante da codificação hierárquica é em ambientes de múltiplo uso (“multiuse environment”), em que as imagens podem ser apresentadas em telas de diferentes resoluções. Neste caso, a quantidade de dados para se alcançar o próximo nível, ao contrário da transmissão progressiva, cresce exponencialmente.

2.7.1 Hierarquias de resolução variável

Nas técnicas de hierarquia de resolução espacial variável, é comum a interpretação do conjunto de níveis como uma estrutura piramidal, em que a base representa a imagem com resolução total. Por exemplo, cada nível da pirâmide pode ser formado tomando-se a média de blocos de 2x2 pixels, como apresentado no exemplo da Fig. 2.18. Nesta figura são mostradas as conexões entre os pixels de mesma região espacial. Geralmente, o pixel do nível imediatamente superior é utilizado como estimador na codificação dos quatro pixels de mesma posição espacial que, na Fig. 2.18, estão a ele conectados.

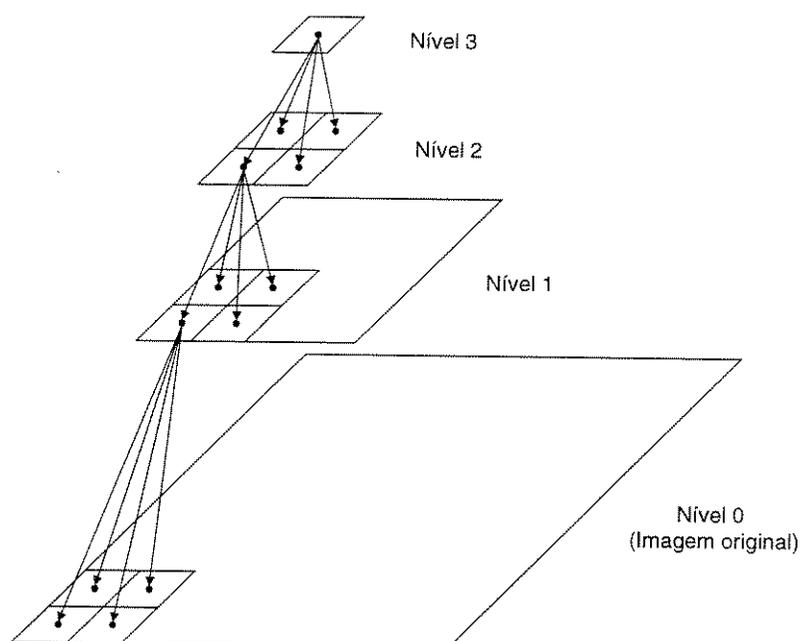


Fig. 2.18 - Exemplo de codificação hierárquica com estrutura piramidal

Os valores dos pontos de cada nível da pirâmide podem ser obtidos de várias outras formas. Por exemplo, dizimando-se a imagem nas duas direções, horizontal e vertical. Dessa forma, só os outros três pixels precisam ser enviados. Esta técnica apresenta a vantagem de manter a mesma quantidade de pontos que a imagem original, entretanto a dizimação introduz “aliasing” que prejudica a visualização dos níveis superiores. Além disso, o ponto escolhido pode não representar bem a região da qual foi tomado.

Para diminuir o efeito de “aliasing” pode-se tomar a média dos quatro pontos como representante da região, e a informação complementar é obtida, por exemplo, tomando-se a diferença entre os pixels e a média destes. Outras técnicas semelhantes podem ser também utilizadas, como a pirâmide de somas reduzidas (“reduced-sum pyramid”), a pirâmide de diferenças reduzidas (“reduced-difference pyramid”), a transformada S (“S-transform”), entre outras.

A codificação por subbandas utilizando filtros espelhados em quadratura, naturalmente possibilita a codificação hierárquica. Isto porque a subbanda de frequências mais baixas representaria o nível imediatamente superior na pirâmide. A esta subbanda é novamente aplicada a divisão em subbandas até que se obtenha o número de níveis desejados.

Quando se trata de vídeo pode-se realizar, além do escalonamento espacial, um escalonamento temporal, em que a taxa de quadros por segundo varia para cada nível da

hierarquia. Um padrão de codificação de vídeo digital largamente utilizado, conhecido como MPEG-2, possibilita este tipo de escalonamento (“temporal scalability”), bem como o escalonamento espacial (“spatial scalability”). Este padrão também contempla duas outras formas de codificação hierárquica que são: a separação espectral (“data partitioning”) e a requantização dos coeficientes espectrais (“SNR scalability”).

2.7.2 Hierarquias de resolução fixa

Uma forma de se enviar gradativamente uma imagem é segmentar esta imagem em planos de bits. Por exemplo se uma imagem em níveis de cinza é representada por pixels de tamanho 8 bits, esta imagem pode ser enviada em 8 etapas correspondendo a 8 planos de bits. Os bits mais significativos de cada pixel são enviados primeiro, e em seguida, cada plano de bit restante. Este processo equivale a enviar primeiro uma imagem grosseiramente quantizada utilizando-se um quantizador uniforme que arredonda valores em direção ao zero. Por exemplo, se cada pixel da imagem é representado por 8 bits, o passo de quantização inicial é 128, e qualquer valor abaixo de 128 é arredondado para zero. Em seguida, é calculada a diferença entre a imagem enviada recuperada (aplicando-se a quantização inversa) e a imagem original. Esta diferença é também quantizada e enviada, entretanto o passo de quantização é dividido por 2. Repete-se o processo até que o passo de quantização seja igual a 1. Este tipo de escalonamento de dados pode também ser aplicado no domínio da transformada. Neste caso os planos de bit são formados pelos bits dos coeficientes da transformada.

Outra forma de se hierarquizar os dados de uma imagem no domínio da transformada é enviar primeiro os coeficientes de maior importância, seja do ponto de vista psico-visual ou em relação ao MSE. O JPEG, que é um padrão de codificação de imagens, utiliza esta técnica para transmissão progressiva, ordenando os coeficientes da transformada em zig-zag, como disposto na Fig. 2.17a, onde são enviados primeiro aqueles de mais baixa frequência espectral.

A quantização vetorial também possibilita hierarquização da imagens em níveis. Neste caso, são enviados vetores-código intermediários para cada nível da árvore que é utilizada para decodificação. À medida que se passa para o nível seguinte, tem-se vetores-código que são melhores aproximações dos vetores originais.

Capítulo 3

Padrões de Compressão de Imagens e Vídeo Digital

3.1 *Introdução*

Um sistema de compressão de imagens ou vídeo digital é, basicamente, um conjunto de técnicas de compressão, tais como as apresentadas no Capítulo 2, que têm como objetivo atender determinada aplicação, ou conjunto de aplicações. A padronização destes sistemas de compressão é necessária para permitir inteligibilidade e interoperabilidade entre equipamentos de diversos fabricantes. Além disso, a padronização geralmente representa uma tentativa de otimização destas técnicas para as aplicações alvo. A primeira iniciativa de padronização de um sistema de comunicações amplamente difundida foi, provavelmente, o código Morse. Desde então, vários padrões de comunicação, cada vez mais sofisticados, têm sido estabelecidos, entre eles, padrões de compressão de imagem e vídeo digital.

3.2 *O Padrão JPEG*

O JPEG (“Joint Photographic Experts Group”) é um grupo de trabalho da ISO/IEC JTC1 que desenvolveu um padrão de codificação de imagens, ISO 10918 (mais conhecido como Padrão JPEG), baseado na transformada DCT. A estrutura do algoritmo proposto é composta de três principais componentes. O primeiro é um sistema básico que é adequado para a maioria das aplicações de compressão de imagens. O segundo é um conjunto de funcionalidades adicionais do sistema que permitem que o sistema básico satisfaça uma maior gama de aplicações. Entre estas funcionalidades pode-se citar: codificação de imagens originais de 12 bits/pixel, codificação sequencial progressiva, e codificação aritmética. O terceiro componente é um método independente de compressão sem perdas para aplicações que necessitam desta característica.

Apesar de objetivar a codificação de imagens paradas, este padrão tem sido utilizado também para codificação de vídeo, sendo, neste caso, conhecido como “motion JPEG”. Os estudos realizados para o desenvolvimento deste padrão foram o ponto de partida para o desenvolvimento de vários outros padrões de codificação de vídeo digital, tais como o H.261, o MPEG-1, e o MPEG-2.

3.3 O Padrão MPEG-2

Oficialmente, o Padrão MPEG-2 chama-se “Codificação Genérica de Quadros em Movimento e Informação de Áudio Associada: Vídeo” (“Generic Coding of Moving Pictures and Associated Audio Information: Video”), e foi publicado como padrão internacional ISO/IEC 13818-2 (ou, equivalentemente ITU-T 262). O termo “Generic” é utilizado pois este padrão não tem como objetivo uma aplicação específica, mas apresenta um conjunto de algoritmos e ferramentas que podem ser usados para variadas aplicações em diferentes condições operacionais, isto é, com diferentes taxas, diferentes meios de transmissão ou armazenamento, etc. A seguir é dada uma lista de aplicações nas quais pode-se utilizar o Padrão MPEG-2:

- TV - Transmissão broadcasting, via satélite, ou via cabo
- HDTV - Transmissão broadcasting, via satélite, ou via cabo
- Vídeo em meios de armazenamento digital - CD-ROM, CD's de alta densidade, DVD
- Vídeo em computadores - e-mail vídeo, sistemas de informação multimídia
- “Video on Demand”
- Vídeo em redes - Vídeo transmitido através de redes ATM, Ethernet e LAN's
- HDTV em camadas - Transmissão compatível de TV e HDTV, HDTV progressivo (“higher progressive HDTV”)
- Vídeo profissional - Edição não linear, pós-produção de estúdio (“studio post production”)

Na seção seguinte é apresentado um breve histórico do desenvolvimento do Padrão MPEG-2.

3.3.1 Histórico

Em 1988, o comitê técnico da ISO/IEC (JTC 1 - Joint ISO/IEC Technical Committee) em tecnologia da informação criou o grupo de trabalho “Moving Pictures Experts Group” (MPEG), com a finalidade de desenvolver padrões para a representação codificada de vídeo e

áudio associado. Originalmente, três padrões, que almejavam taxas de até 1.5, 10 e 40 Mbits/s, estavam em estudos. Este padrões eram conhecidos como MPEG-1, MPEG-2, MPEG-3, respectivamente.

Originalmente, o Padrão MPEG-2 tinha como objetivo compressão de vídeo digital a taxas de até 10 Mbits/s [34]. Para aplicações com taxas superiores a 10 Mbits/s, e.g. HDTV, outro padrão também em desenvolvimento, conhecido como MPEG-3, seria utilizado [34]. Entretanto, em julho de 1992, o desenvolvimento do Padrão MPEG-3 foi interrompido, pois percebeu-se que o Padrão MPEG-2 poderia atender a aplicações que requerem taxas acima de 10 Mbits/s com alta qualidade de reprodução. Atualmente, o Padrão MPEG-2 é utilizado na etapa de compressão de vídeo do sistema HDTV.

Em 11 de novembro de 1994, foi aprovado o Padrão MPEG-2, oficialmente chamado ISO 13818. Em 10 de julho de 1995 o ITU-T aprovou o Padrão H.262 que equivale às três primeiras partes do ISO 13818 (sistemas, áudio e vídeo).

3.3.2 Processo de desenvolvimento do Padrão MPEG-2

O desenvolvimento do Padrão MPEG-2 foi dividido em duas fases, a fase competitiva e a fase colaborativa. O objetivo da fase competitiva foi procurar pelos melhores esquemas de codificação de vídeo e testá-los sob algumas condições padronizadas, para avaliar performance, complexidade de implementação, possibilidade de acesso randômico, atrasos de codificação e decodificação e robustez a erros. O objetivo da fase colaborativa foi refinar interativamente a performance dos algoritmos e ferramentas de codificação de vídeo que mostraram ser promissores na primeira fase para encontrar a melhor ferramenta, quando existem várias ferramentas voltadas para a mesma aplicação. Na fase colaborativa, foi determinado que, para atender as mais variadas aplicações, o padrão seria estruturado de maneira hierárquica, de acordo com níveis (“levels”) e perfis (“profiles”). Os níveis determinam restrições aos parâmetros do codificador, por exemplo, impondo limites superiores para taxa de bits, tamanho do quadro, tamanho do buffer, etc. Os perfis estão relacionados com limitações na sintaxe que implicam, por exemplo, restrições aos tipos de

quadros utilizados¹ (I, P ou B), ao tipo de subamostragem da crominância² (4:2:0, 4:2:2 ou 4:4:4), e aos modos de escalonamento permitidos.

3.3.3 Partes que compõe o Padrão ISO 13818

O Padrão MPEG-2, formalmente conhecido como ISO 13818, é dividido em dez partes [4,13] que são apresentadas nas seções seguintes.

3.3.3.1 Parte 1: Sistemas

Esta parte define como é feita a multiplexação de vários sinais de áudio e vídeo. Também são definidos sinais de sincronismo, tais como:

- STC (“System Time Clock”), utilizado para sincronismo no decodificador;
- PTSs (“Presentation Time Stamps”), utilizados para controlar a saída do decodificador, ou seja, a apresentação dos quadros.

O sistema MPEG-2 define dois tipos de “streams”: a “stream” de programa (“program stream”) e a “stream” de transporte. As “streams” de programa são voltadas para aplicações não sujeitas a erros (e.g., DVD – “Digital Versatile Disc”), e são adequadas para aplicações que envolvem processamento em software. Os pacotes gerados são, geralmente, longos e têm tamanhos que podem variar entre 1 e 64K bytes.

Por outro lado, a “stream” de transporte oferece robustez necessária para transmissão através de canais ruidosos, e possibilita a inclusão de vários programas em uma mesma “stream”. Desta forma, ela é adequada para transmissão de vídeo digital através de “transponders”, redes de cabos coaxiais, broadcasting terrestre, e redes ATM. Os pacotes têm comprimento fixo de 188 bytes, o que possibilita que cada pacote seja colocado em quatro células ATM, restando 4 bytes que podem ser utilizados, por exemplo, pela camada de adaptação AAL - “ATM Adaptation Layer” (Obs.: $4 \times 48 = 192$; $192 - 188 = 4$ bytes). Além disso, estes pacotes de comprimento fixo facilitam o processamento em hardware e a implementação de esquemas de correção de erros.

O PES (“Packetized Elementary Stream”) é a estrutura básica na organização tanto da “stream” de programa, quanto da “stream” de transporte.

^{1,2} Os tipos de quadros (I, P, e B) são definidos na Seção 3.3.5, e os tipos de subamostragem da crominância (formatos 4:4:4, 4:2:2, e 4:2:0), na Seção 3.3.4.

3.3.3.2 *Parte 2: Vídeo*

Define a sintaxe da seqüência de bits gerada pelo codificador e o processo de decodificação. É interessante salientar que o processo de codificação não é especificado pelo Padrão ISO 13818, ficando a cargo do projetista os detalhes da implementação. Busca-se portanto, especificar o mínimo a fim de garantir a interoperabilidade entre os diversos codecs implementados. Para isto, é suficiente a definição de uma sintaxe comum, e a descrição da semântica do decodificador. Uma visão geral do processo de decodificação é apresentada com mais detalhes em outras seções deste capítulo.

O ITU-T colaborou com o desenvolvimento, tanto desta parte do padrão, quanto da parte que trata de sistemas (ISO 13818-1). Desta forma, o ITU-T publicou recomendações que são equivalentes a estas duas partes do Padrão ISO 13818, a saber:

- ITU-T H.262 [2], que foi aprovado em 10 de julho de 1995 e corresponde ao ISO 13818-2;
- e o H.222.0 [30], que corresponde ao ISO 13818-1.

3.3.3.3 *Parte 3: Áudio.*

O áudio no Padrão MPEG-2 é tratado em duas partes. A primeira, especificada nesta parte do padrão (ISO 13818-3), apresenta compatibilidade total com o Padrão MPEG-1. Por outro lado, a segunda, especificada no ISO 13818-7 (parte 7), apresenta apenas compatibilidade “forward”, ou seja, um decodificador de áudio MPEG-2 pode decodificar uma seqüência de áudio MPEG-1, mas o inverso não é possível. Neste contexto, compatibilidade total (“backward” e “forward”) significa que um decodificador de áudio MPEG-1 pode decodificar até dois canais de uma seqüência MPEG-2, e que um decodificador de áudio MPEG-2 pode decodificar uma seqüência de áudio MPEG-1 como se fosse um decodificador de áudio MPEG-1. Quanto ao número de canais, a diferença entre os dois padrões é que o MPEG-1 pode decodificar no máximo dois canais de áudio (possibilitando áudio estéreo), enquanto o MPEG-2 permite a decodificação de vários canais, permitindo efeito “surround”.

A compatibilidade “backward” é uma restrição muito forte e compromete parcialmente a eficiência da codificação. Entretanto, ela permite a migração para o uso de vários canais (áudio multicanal) sem tornar obsoletos os decodificadores MPEG-1 existentes.

3.3.3.4 *Parte 4: Testes de conformidade.*

O ISO 13818-4 determina a metodologia para verificar a conformidade com o padrão de um decodificador ou de uma seqüência gerada por um codificador. No caso da seqüência gerada por um codificador, verifica-se se a sintaxe está correta, ou seja, se esta é uma seqüência válida. Para o decodificador, os testes são realizados a fim de verificar se ele atende aos requerimentos especificados nas partes 1, 2 e 3 do padrão.

Para testar os decodificadores, são geradas seqüências, conhecidas como “evil bitstreams” que exploram as dificuldades potenciais para obtenção de conformidade com o padrão. Estas seqüências são decodificadas no decodificador testado e também em um decodificador de referência, que é assumido como um decodificador em perfeita conformidade com o padrão. As saídas são comparadas e verifica-se se há diferença entre os pixels reconstruídos e se houve perda temporária de sincronismo de bit no decodificador testado, ou seja, se ocorreu imprecisão ou erro no processo de decodificação.

3.3.3.5 *Parte 5: Software.*

É uma implementação, em linguagem C, de softwares de simulação baseados nas três primeiras partes do padrão, referentes a sistemas, áudio e vídeo. A implementação inclui tanto o codificador quanto o decodificador. A parte de sistemas consiste em um software que codifica e decodifica “streams” de programa e “streams” de transporte. O software da parte de vídeo consiste em um codec de vídeo não escalonável. O software de áudio consiste em um codec de áudio que utiliza dois diferentes modelos psico-acústicos.

3.3.3.6 *Parte 6: DSM-CC.*

O MPEG-2 DSM-CC (“Digital Storage Media Command and Control”) especifica um conjunto de protocolos que fornecem funções de controle e operações específicas para o manuseio de seqüências MPEG-2. Alguns exemplos de funções são: conexão, reprodução em tela, edição, armazenamento, reprodução reversa, etc.

A necessidade de se estabelecer estes protocolos se deve ao fato que cada meio de armazenamento digital (DSM – “Digital Storage Media”) tem sua própria linguagem de comandos e controle. Isto é um problema para o usuário que deseja utilizar diferentes DSM, pois, a princípio, ele deveria conhecer a linguagem de cada um deles. Com o DSM-CC o usuário não precisa conhecer detalhes de cada DSM. Os comandos e operações são

transparentes para o usuário, ainda que ele esteja utilizando um DSM remotamente (e.g., através de uma rede).

3.3.3.7 Parte 7: Codificação de áudio avançada

A codificação de áudio avançada (AAC – “Advanced Audio Coding”), ao contrário da especificação 13818-3, não apresenta compatibilidade “backward” em relação ao Padrão MPEG-1. Sem este compromisso de compatibilidade foi possível aprimorar o ganho de compressão de maneira que a AAC apresenta aproximadamente a mesma performance que o áudio do Padrão MPEG-1, com metade da taxa de bits.

A necessidade de compatibilidade “backward”, que é característica da especificação 13818-3, surgiu para atender a várias aplicações, como por exemplo broadcasting de TV. Entretanto, algumas aplicações não necessitam desta compatibilidade e este foi o estímulo para a elaboração desta parte do padrão (ISO 13818-7).

3.3.3.8 Parte 8: Vídeo com 10 bits de resolução amplitude (cancelado).

Esta parte do padrão atenderia a aplicações profissionais que requerem alta qualidade. Argumentava-se que estas aplicações necessitariam de maior resolução espacial para a crominância (requerimento atendido pela “profile” 4:2:2, estabelecida em janeiro de 1996), e de maior resolução de amplitude para a luminância e para a crominância, ou seja, a precisão da amostra de vídeo seria de 10 bits e não de 8 bits como é usual. Os trabalhos nesta parte do padrão foram interrompidos, pois as indústrias de vídeo que haviam requisitado esta funcionalidade mudaram seus interesses para outros domínios. Considerou-se que vídeo de 10 bits (“10-bit video”) serviria para aplicações muito restritas e especializadas, e este item foi oficialmente removido do programa de trabalho do MPEG-2.

3.3.3.9 Parte 9: Interface de tempo real (RTI – “Real-Time Interface”).

Esta parte foi aprovada como padrão em julho de 1996, e especifica a interface de tempo real entre os decodificadores de “stream” de transporte e os adaptadores de canal. Com esta especificação, é possível construir camadas de adaptação de rede que garantem a performance de tempo requerida, e que também garantem que os decodificadores terão um comportamento correto em termos de buffers e recuperação de sincronismo temporal (“timing recovery”).

3.3.3.10 Parte 10: Testes de conformidade do DSM-CC.

Os testes de conformidade do DSM-CC (“DSM-CC Conformance”) definem meios para verificar se uma implementação específica de DSM-CC está em conformidade com a parte 6 do Padrão MPEG-2 (ISO 13818-6).

3.3.4 Descrição dos processos de codificação e decodificação

O Padrão MPEG-2 comprime seqüências de vídeo digitalizadas basicamente explorando os três tipos de redundância presentes nestas seqüências:

- (1) redundância espacial: através da transformada DCT combinada com codificação de entropia;
- (2) redundância temporal: através de codificação preditiva entre quadros;
- (3) redundância psico-visual: através de subamostragem da crominância.

Uma visão geral das técnicas utilizadas para se reduzir estas redundâncias foi apresentada no Capítulo 2. Esta seção apresenta como estas técnicas são utilizadas pelo Padrão MPEG-2.

O diagrama geral de um codificador MPEG-2 é apresentado na Fig. 3.1. Cada etapa deste processo será brevemente discutida, entretanto, primeiro é apresentada a estrutura adotada para a seqüência de vídeo, ou seja, a forma como os dados são divididos para serem codificados.

A estrutura de um quadro da seqüência é apresentada na Fig. 3.2. Cada quadro é dividido em fatias (“slices”) de macroblocos, e estes são divididos em blocos.

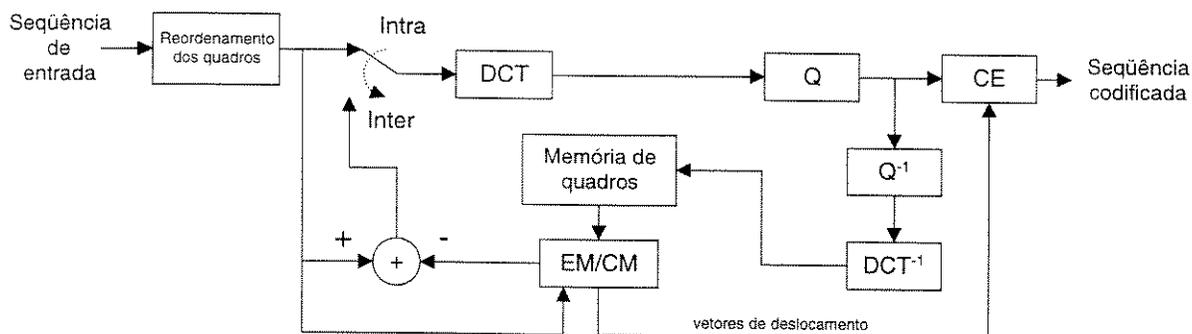


Fig. 3.1 - Diagrama geral de um codificador MPEG-2

Legenda:

- CE - Codificação de entropia (Huffman e corrida de zeros).
- DCT - Transformação DCT de blocos de 8x8 pixels
- DCT^{-1} - Transformação DCT inversa
- Q - Quantizador
- Q^{-1} - Quantizador inverso²
- EM/CM - Estimativa e compensação de movimento

O bloco é a menor unidade de codificação no algoritmo MPEG. Ele é utilizado na aplicação da transformada DCT, e tem tamanho 8x8 pixels. O bloco pode ser de luminância (Y), ou crominância (Cr ou Cb).

O macrobloco é a unidade básica de codificação no algoritmo MPEG. Ele consiste de segmentos de 16 x 16 pixels, e é a unidade utilizada nas etapas de estimativa e compensação de movimento. Na Fig. 3.2 é apresentado um exemplo de macrobloco de formato 4:2:0, composto de quatro blocos de luminância e dois de crominância.

O formato 4:2:0 refere-se a forma como subamostra-se a crominância do sinal. Três formatos são possíveis:

- 4:2:0 - a crominância é subamostrada por um fator de dois nas duas direções, horizontal e vertical.
- 4:2:2 - a crominância é subamostrada por um fator de dois somente na horizontal, e necessita-se de 8 blocos para representar cada macrobloco.
- 4:4:4 - a crominância não é subamostrada, e, portanto, necessita-se de 12 blocos para representar cada macrobloco.

² A utilização do símbolo Q^{-1} para representar quantização inversa é um abuso de notação, já que a quantização não é uma operação inversível. Entretanto, esta notação já foi consagrada pela literatura.

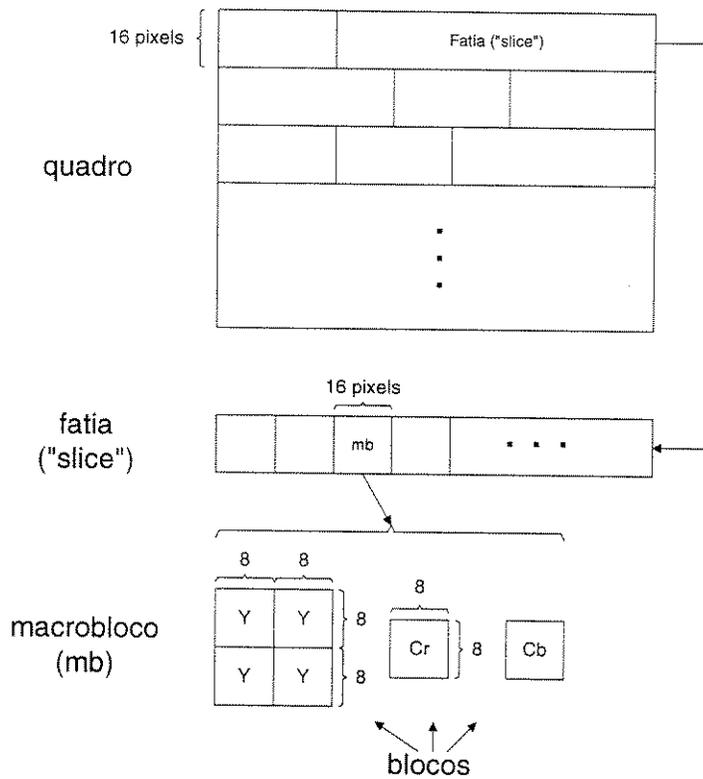


Fig. 3.2 - Estrutura de um quadro não entrelaçado de formato 4:2:0

O “slice” é uma fatia horizontal de macroblocos e serve como unidade de resincronismo. Uma fatia horizontal de macroblocos de um quadro pode conter mais de um slice. Além disso, podem existir áreas do quadro onde não há slices e que, portanto, não são codificadas. Entretanto, todos os perfis (“profiles”) do padrão utilizam uma estrutura de slice restrita (“restricted slice structure”), em que todo o quadro é segmentado em slices, como no exemplo da Fig. 3.2.

Grupo de quadros (GOP - “Group of pictures”) é um conjunto de quadros que possibilita acesso randômico. No Padrão MPEG-2, o conceito de GOP é um pouco diferente do Padrão MPEG-1. No primeiro caso sua utilização é opcional, enquanto no segundo é obrigatória. Tipicamente, 6 a 15 quadros são agrupados em um GOP e é estabelecida uma ordem de tipos de quadros (I, P, ou B) que se repete ao longo da seqüência. Os tipos de quadros são apresentados na seção seguinte.

3.3.5 Tipos de quadros

Os quadros podem ser divididos em três tipos, de acordo com a codificação preditiva utilizada:

- quadro I (intracodificado): este quadro é codificado de maneira independente dos outros quadros;
- quadros P (predito): este quadro é codificado utilizando-se o quadro de referência anterior como preditor;
- quadros B (interpolado): o preditor utilizado na codificação deste quadro é baseado em dois quadros de referência, um anterior e outro posterior.

Cada quadro P ou B é codificado tomando-se a diferença entre os quadros originais e uma estimativa deste quadro. Para o cálculo desta estimativa, no caso de quadros P, o quadro de referência anterior é utilizado em um processo que envolve estimação e compensação de movimento (EM/CM). Para os quadros B o processo é semelhante só que, neste caso, são utilizados dois quadros de referência (um anterior e outro posterior) que, após o processo de EM/CM, são interpolados para formarem o quadro estimado. Os quadros de referência são os quadros P ou I temporalmente mais próximos do quadro atual, um anterior e outro posterior. Estes quadros são recuperados no codificador e armazenados na memória de quadros para serem utilizados no processo de EM/CM. Os quadros do tipo B nunca são usados como referência. A Fig. 3.3 apresenta um exemplo de uma seqüência de quadros, ordenados temporalmente. Neste exemplo, para cada quadro, é dado o tipo de predição e os quadros de referência relacionados.

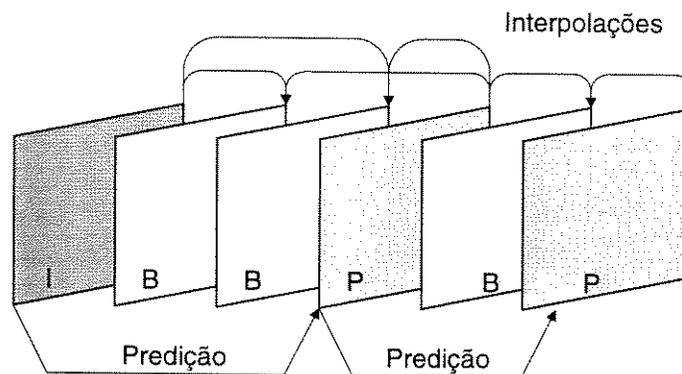


Fig. 3.3 - Exemplo de estrutura temporal de seqüência

3.3.6 Reordenamento de quadros.

Os quadros a serem codificados, geralmente estão dispostos em ordem temporal, ou seja, de acordo com a ordem de apresentação na tela quando a seqüência é reproduzida. Quando os quadros são codificados, eles são incluídos na “bitstream” de acordo com a ordem em que são decodificados, este ordenamento é chamado ordem de codificação.

Quando a seqüência não contém quadro do tipo B, a ordem de codificação é igual a ordem de temporal. Entretanto, se existem quadros B na seqüência, a ordem de codificação é diferente da ordem temporal. Neste caso, a saída do processo de decodificação deve obedecer às seguintes regras:

- se o quadro atual decodificado é do tipo B, e não é o último da seqüência, o quadro de saída é o quadro atual;
- se o quadro atual é do tipo B, e é o último da seqüência, o quadro atual é enviado para a saída e, em seguida, é enviado o último quadro de referência anteriormente decodificado;
- se o quadro atual decodificado é do tipo I ou P, e este quadro não é o primeiro nem o último da seqüência, a saída é o último quadro I ou P anteriormente decodificado que foi previamente armazenado em um “buffer” para servir como referência para outros quadros;
- se o quadro atual decodificado é do tipo I ou P, e é o primeiro da seqüência, nenhum quadro é enviado para a saída;
- se o quadro atual decodificado é do tipo I ou P, e é o último da seqüência, o último quadro I ou P anteriormente decodificado é enviado para a saída e, em seguida, o quadro atual decodificado também é enviado.

Na codificação, deve-se ordenar os quadros a serem codificados de maneira que estas regras de ordenamento na decodificação sejam válidas. Basicamente, quando o quadro atual na ordem temporal é do tipo B, deve-se codificar primeiro sua referência posterior. Em seguida, o quadro atual e os quadros seguintes do tipo B, que utilizam a mesma referência posterior, são codificados.

Um exemplo de ordenamento na codificação e reordenamento na decodificação é dado a seguir. Neste exemplo, tem-se uma seqüência de quadros T_n , onde T é o tipo do quadro (I, P ou B) e n é a posição temporal deste quadro. T_n é utilizado para representar os quadros originais, os quadros codificados e os quadros reconstruídos.

Na entrada do codificador tem-se:

$$I_1 \ B_2 \ B_3 \ P_4 \ B_5 \ B_6 \ P_7 \ P_8 \ B_9 \ I_{10} \ P_{11}$$

Na saída do codificador e na entrada do decodificador tem-se:

$$I_1 \ P_4 \ B_2 \ B_3 \ P_7 \ B_5 \ B_6 \ P_8 \ I_{10} \ B_9 \ P_{11}$$

Na saída do decodificador tem-se:

$$I_1 \ B_2 \ B_3 \ P_4 \ B_5 \ B_6 \ P_7 \ P_8 \ B_9 \ I_{10} \ P_{11}$$

3.3.7 Codificação de um macrobloco

Quanto à forma de predição utilizada, basicamente, um macrobloco pode ser codificado de quatro modos:

- intracodificado;
- predito utilizando-se o quadro de referência anterior;
- predito utilizando-se o quadro de referência posterior;
- ou interpolado (utiliza os dois quadros de referência para formar a predição).

O Padrão MPEG-2 [2] contempla algumas variações destes quatro modos básicos, por exemplo, o “dual-prime” e o 16x8 MC. Neste último o macrobloco é dividido em dois e o processo de EM/CM é realizado para cada uma das partes de tamanho 16x8 pixels. Este modo só é utilizado quando os campos do quadro são codificados individualmente.

Os processos de predição e interpolação envolvem estimação e compensação de movimento. Estas técnicas foram descritas na Seção 2.1.2. Se o macrobloco atual é predito, seja em relação ao quadro de referência anterior ou posterior, o macrobloco do quadro de referência que apresentou melhor casamento é utilizado como predição. No caso de macroblocos interpolados, a predição é formada tomando-se a média entre os macroblocos de melhor casamento de cada um dos quadros de referência.

Os dados a serem codificados, referentes a um macrobloco predito ou interpolado, são resultantes da diferença entre este macrobloco e a sua predição. Tem-se então um macrobloco-diferença que é codificado de maneira semelhante a um macrobloco intracodificado, utilizando-se transformada DCT e codificação de Huffman (VLC - “Variable-Length Code”). Os vetores de deslocamento resultantes do processo de EM/CM são codificados e incluídos na “bitstream”.

Todos os macroblocos de um quadro I são intracodificados. Nos quadros P pode-se ter macroblocos preditos em relação ao quadro de referência anterior, ou macroblocos intracodificados, o que equivale a utilizar um macrobloco nulo (todos elementos iguais a zero) como predição. Finalmente, nos quadros B pode-se ter macroblocos interpolados, macroblocos preditos (em relação ao quadro de referência anterior ou posterior), ou macroblocos intracodificados. A escolha de como é formada a predição para um macrobloco fica a cargo do projetista do codificador e, geralmente, depende da correlação entre o macrobloco atual e o macrobloco de melhor casamento de cada quadro de referência. Nos quadros B e P, existe a possibilidade de não se codificar determinado macrobloco. Neste caso, o macrobloco é reconstruído no decodificador utilizando-se somente os quadros de referência e considerando nulos os vetores de movimento, se o quadro atual é do tipo P, ou tomando-se os vetores de movimento do último macrobloco codificado, se o quadro atual é do tipo B.

Os dados resultantes deste processo, sejam macroblocos originais ou um macroblocos-diferença, são divididos em blocos de tamanho 8x8. A cada bloco é aplicada uma transformação DCT bidimensional. Os coeficientes resultantes são quantizados. O coeficiente DC de macroblocos intra é quantizado de forma diferente que os outros coeficientes. O Padrão MPEG-2 só define o processo de quantização inversa, permitindo flexibilidade no projeto do quantizador. A quantização inversa dos coeficientes DC dos blocos intracodificados é dada pela equação (3.1), onde $CQ[x][y]$ são os coeficientes quantizados, $C'[x][y]$ os coeficientes recuperados, e $intra_dc_mult$ o passo de quantização. Os coeficientes restantes são recuperados (inversamente quantizados) de acordo com a equação (3.2), onde $W[x][y]$ é a matriz de quantização utilizada.

$$C'[x][y] = intra_dc_mult \times CQ[x][y] , \quad (3.1)$$

$$C'[x][y] = \frac{(2 \times CQ[x][y] + k) \times W[x][y] \times quantiser_scale}{32} , \quad (3.2)$$

onde

$$k = \begin{cases} 0 & \text{para blocos intracodificados} \\ \text{Sinal}(CQ[x][y]) & \text{para blocos não intracodificados} \end{cases} ,$$

e

$$\text{Sinal}(CQ[x][y]) = \begin{cases} 0 & \text{se } CQ[x][y] = 0 \\ 1 & \text{se } CQ[x][y] > 0 \\ -1 & \text{se } CQ[x][y] < 0 \end{cases}$$

No codec desenvolvido, a quantização é realizada de forma que a relação entre os coeficientes originais, $C[x][y]$, e os coeficientes recuperados, $C'[x][y]$, é de acordo com a Fig. 3.4a, para blocos não intracodificados, e de acordo com a Fig. 3.4b, para blocos intracodificados. Nestas quantizadores, L representa o passo de quantização e assume os seguintes valores:

- $L = \text{intra_dc_mult}$, para coeficientes DC de MB intra;
- $L = \frac{W[x][y] \times \text{quantiser_scale}}{16}$, para os demais coeficientes.

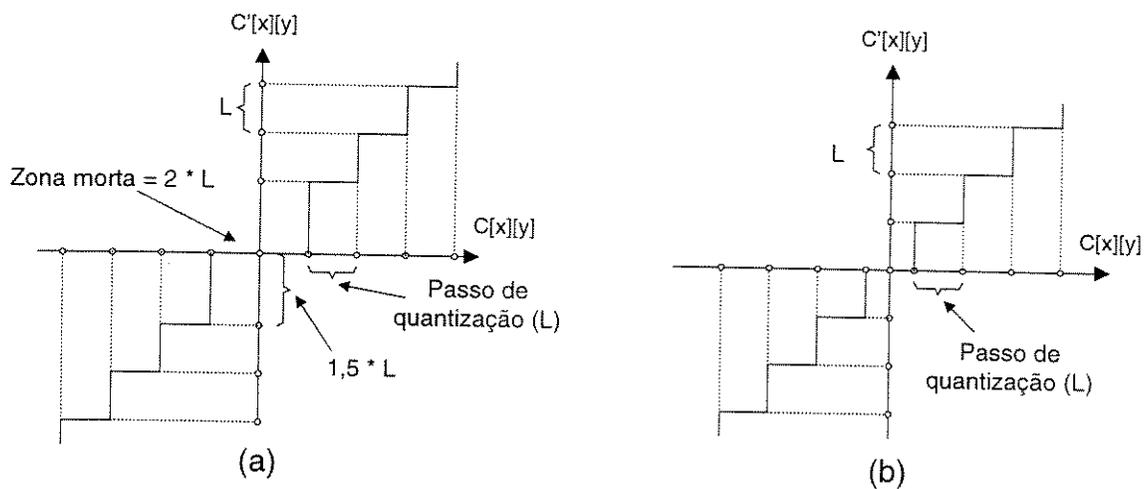


Fig. 3.4 – Relação entre coeficientes originais e coeficientes recuperados para blocos não intracodificados (a), e para blocos intracodificados (b).

Para os formatos 4:4:4 ou 4:2:2, quatro matrizes de quantização podem ser utilizadas. Por outro lado, no formato 4:2:0 somente duas matrizes de quantização são utilizadas, uma para blocos intracodificados e outra para blocos não intracodificados (e.g., preditos). A Fig. 3.5 apresenta as duas matrizes de quantização, $W[x][y]$, recomendadas pelo Padrão MPEG-2, a primeira (Fig. 3.5a) para blocos intracodificados, e a segunda (Fig. 3.5b) para blocos não intracodificados. Além destas matrizes, podem ser utilizadas outras que, neste caso, devem ser enviadas junto com os dados codificados (e.g., no cabeçalho de um quadro).

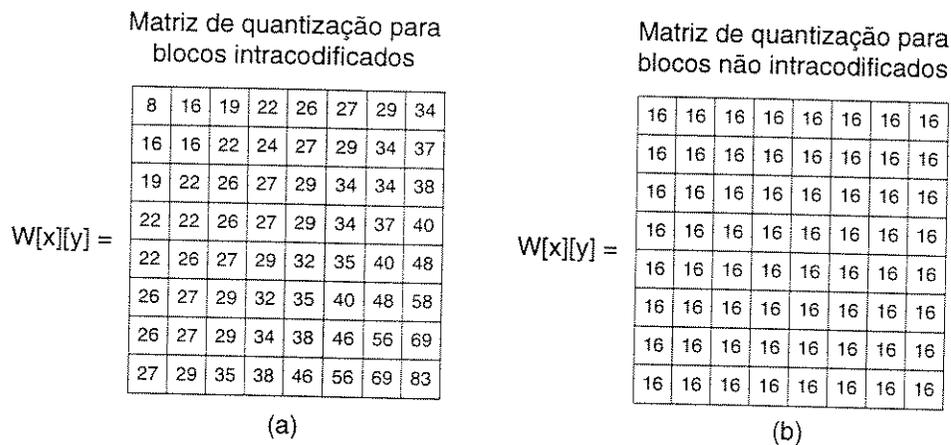


Fig. 3.5 - Matrizes de quantização recomendadas pelo Padrão MPEG-2, para blocos intracodificados (a) e para blocos não intracodificados (b).

O próximo passo é a codificação dos coeficientes quantizados. Os coeficientes DC dos blocos intra (intracodificados) são codificados diferentemente dos demais. Para os coeficientes DC dos blocos intra, é utilizado um esquema DPCM, no qual codifica-se a diferença entre o coeficiente DC do bloco atual e o coeficiente DC do bloco anterior. A esta diferença é atribuída uma palavra-código que indica o número de bits necessários para se representar esta diferença.

Os coeficientes restantes de cada bloco são ordenados de acordo com uma das duas matrizes de varredura apresentadas na Fig. 2.17. Em seguida, são codificados através de uma combinação de codificação por corrida de zeros (“runlength”) e codificação VLC. O Padrão MPEG-2 recomenda duas tabelas VLC para codificação destes coeficientes [2]. No Anexo B uma destas tabelas é apresentada (Tabela B.1). A cada coeficiente não nulo é atribuído uma palavra-código que representa o nível (módulo) do coeficiente e o número de coeficientes nulos entre o coeficiente atual e o último coeficiente não nulo codificado deste bloco. Esta palavra-código é seguida de um bit que representa o sinal do coeficiente. Na tabela VLC, existem duas palavras-código especiais. Uma delas serve para indicar fim de bloco, isto é, indica que não há mais nenhum coeficiente não nulo no bloco atual. A outra representa um código de escape (“scape code”), que indica que são utilizadas palavras de comprimento fixo para representar o comprimento da corrida de zeros (6 bits) e o coeficiente com sinal (12 bits), de acordo com a tabela B.2 do Anexo B.

O código de escape é utilizado, pois o número de combinações entre corridas de zeros e os níveis de coeficientes é muito grande. Considerando as corridas de zeros pode variar entre 0 e 63, e o número de níveis pode variar entre 1 e 2048, seriam necessárias 13.107.264 (64x2048) palavras-código para representar todas as combinações possíveis, tornando impraticável este esquema de codificação. Entretanto, a grande maioria destas combinações ocorre com uma frequência muito baixa, e são representadas utilizando-se o código de escape, como descrito anteriormente. As combinações estatisticamente mais frequentes (um pouco mais que 100, de acordo com as tabelas VLC do Padrão MPEG-2) são representadas individualmente por palavras-código. Mesmo com o uso do código de escape, tipicamente, a codificação VLC proporciona um fator de compressão de, pelo menos, dois, quando comparada com códigos de comprimento fixo.

3.3.8 Escalonamento

Em algumas situações, deseja-se possibilitar a representação da seqüência de vídeo em diferentes resoluções (espaciais, temporais ou de amplitude), de acordo com as limitações de taxa de transmissão, e com os equipamentos utilizados em uma determinada aplicação. Por exemplo, deseja-se transmitir um sinal de vídeo para receptores de HDTV e de TV convencional. Uma possibilidade é enviar a mesma seqüência através de dois canais (“multicast”), cada um satisfazendo os requisitos de cada receptor. Entretanto, neste caso, o meio de transmissão é mal utilizado, pois as informações enviadas para o receptor de TV convencional podem ser aproveitadas na codificação do sinal de HDTV, possibilitando uma redução da taxa de transmissão. Este é um exemplo de escalonamento (ou “escalabilidade”), em que a seqüência codificada é enviado em dois canais (“simulcast”), sendo o primeiro essencial para a decodificação desta seqüência, e o segundo para melhorar a resolução espacial, temporal ou de amplitude (relacionada com o número de bits utilizados para representar cada pixel).

Outra importante aplicação do escalonamento é aumentar a robustez do sistema em canais sujeitos a erros de transmissão. Neste caso, a separação em vários canais é feita de forma a garantir que os dados essenciais para a decodificação, que possibilitam reprodução da seqüência com qualidade aceitável, sejam recebidos sem erros pelo receptor. Os outros canais são para melhorar a qualidade da seqüência reproduzida no receptor (e.g., acrescentando informações de realce).

O escalonamento consiste em a codificar a seqüência em vários níveis de hierarquia, sendo que a decodificação em determinado nível desta hierarquia depende da disponibilidade de dados de todos os níveis inferiores a este nível. Técnicas de codificação hierárquica foram apresentadas no Capítulo 2.

O Padrão MPEG-2 possibilita quatro modos de escalonamento:

- temporal,
- espacial,
- SNR,
- particionamento de dados (“data partitionig”).

Nas seções seguinte serão apresentados cada um destes modos de escalonamento.

3.3.8.1 Escalonamento Temporal

O escalonamento temporal consiste em codificar a seqüência a uma determinada taxa de quadros, constituindo a camada base e, em seguida, acrescentar informações em camadas superiores para possibilitar a reprodução a taxas de quadros/s maiores. Um exemplo simples é quando a camada base e a camada superior são codificadas, cada uma, à metade da taxa de quadros por segundo da seqüência original. O escalonamento temporal pode ser utilizada para se obter interoperabilidade entre vídeo entrelaçado e progressivo, em aplicações tais como HDTV, TV digital, entre outras.

3.3.8.2 Escalonamento Espacial

Nesta técnica, todas as camadas, inclusive a camada base, têm a mesma resolução temporal. Entretanto, a resolução espacial é incrementada quando se passa de determinada camada para a camada superior. Uma das aplicações deste método de escalonamento é transmissão "simulcast" de TV digital e HDTV, em que a resolução da TV deve ser incrementada para atender aos requisitos do sistema HDTV.

3.3.8.3 Escalonamento SNR

Neste caso, o escalonamento dos dados é obtido através de requantização dos coeficientes. Tanto a camada base quanto as camadas superiores possuem a mesma resolução temporal e espacial. O que muda de uma camada para outra é a qualidade de reprodução, que esta relacionada com o SNR. Por isto, esta técnica é chamada escalonamento SNR.

3.3.8.4 Particionamento de Dados (“data partitioning”)

Esta técnica consiste em segmentar a sequência de vídeo codificada em duas ou mais partições (camadas) para transmissão ou armazenamento. Através desta técnica, pode-se obter escalonamento por separação espectral, apesar de não ser possível estabelecer uma mesma “frequência de corte” para todos os bloco codificados de um quadro.

A Fig. 3.6 apresenta um exemplo de particionamento de dados aplicado a um bloco de coeficientes quantizados. Neste exemplo, os primeiros 5 coeficientes (N=5) são enviados na camada 0 (de alta prioridade), e os coeficientes restantes na camada 1 (de baixa prioridade).

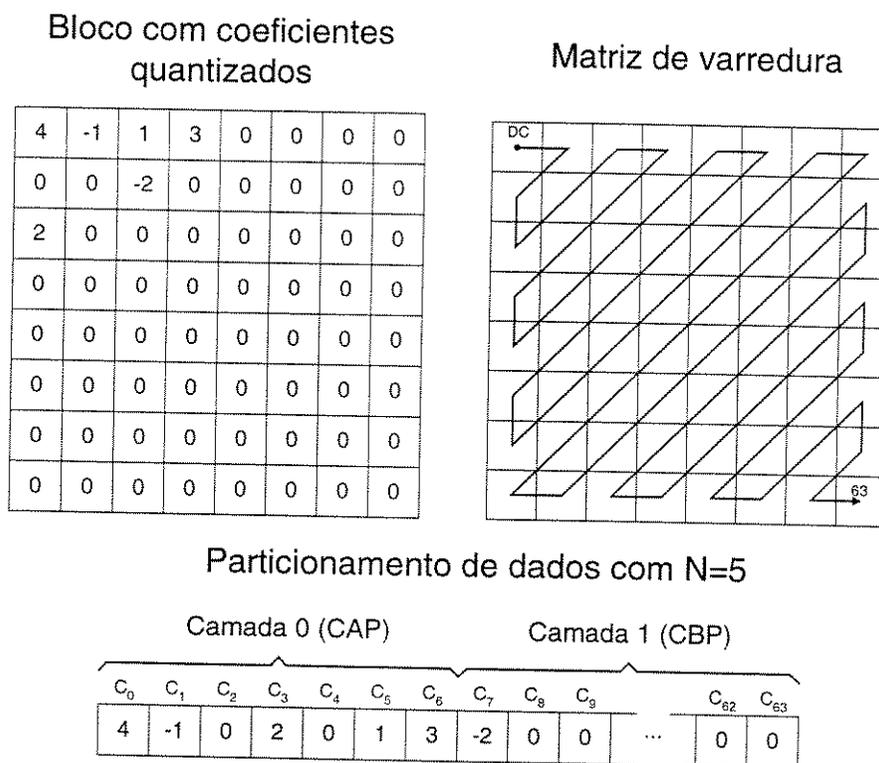


Fig. 3.6 - Exemplo de particionamento de dados

3.4 Outros Padrões de Compressão de Vídeo Digital

Na década de 80, foram desenvolvidos pelo CCITT, que atualmente é conhecido como ITU-T, padrões de compressão digital para vídeo-conferência. Por exemplo, o Padrão H.320, que tem como finalidade aplicações de vídeo-conferência em redes RDSI (“Redes digitais de serviços Integrados”). A parte deste padrão que trata da codificação de vídeo é chamada H.261

e codifica seqüência de quadros em formato CIF (“Common Intermediate Format”) de 352x288 pixels, a taxas de transmissão múltiplas de 64 Kbits/s.

Atualmente, o grupo MPEG trabalha no desenvolvimento do Padrão MPEG-4 que trata da codificação de vídeo a baixa taxas para comunicações multimídia. De acordo com previsões [13], o MPEG-4 se tornará padrão internacional em novembro de 1998. Neste padrão, são utilizadas novas técnicas de compressão, por exemplo, codificação baseada em modelos e codificação baseada em regiões do objeto.

Capítulo 4

Transmissão de Vídeo em Redes Digitais de Faixa Larga

4.1 Introdução

Neste capítulo são apresentados aspectos a serem considerados quando se transmite vídeo através de redes digitais de serviços integrados de faixa larga (RDSI-FL) baseadas no Padrão ATM. Também são apresentadas técnicas para reduzir os efeitos dos erros de canal quando se transmite vídeo digital comprimido através destas redes, particularmente, quando se utiliza o Padrão MPEG-2.

4.2 Redes ATM

O modo de transferência assíncrono (ATM - “Asynchronous Transfer Mode”) [6,7] tem se firmado como uma boa opção para implementação de redes digitais de serviços integrados de faixa larga (BISDN - “Broadband Integrated Services Digital Networks”). Um dos serviços oferecidos por este tipo de rede, que deve ocupar boa parte da banda de transmissão disponível, é a transmissão de vídeo.

A transmissão de dados na rede ATM é realizada através de pacotes, denominados células, de tamanho fixo de 53 bytes dos quais 5 bytes são de cabeçalho e os 48 restantes, de dados do usuário. O cabeçalho de uma célula ATM é apresentado na Fig. 4.1.

Neste tipo de rede é possível transmissão a taxas variáveis (VBR - “Variable Bit Rate”), pois a alocação de canais é feita considerando-se multiplexação estatística. Esta característica é favorável para a transmissão de vídeo, pois com taxa variável é possível que a qualidade da reprodução do sinal de vídeo se mantenha aproximadamente constante, independente das variações de complexidade das cenas. Quando se utiliza uma taxa de transmissão constante (CBR - “Constant Bit Rate”) a qualidade pode sofrer grandes variações, por exemplo, quando ocorrem mudanças de cenas.

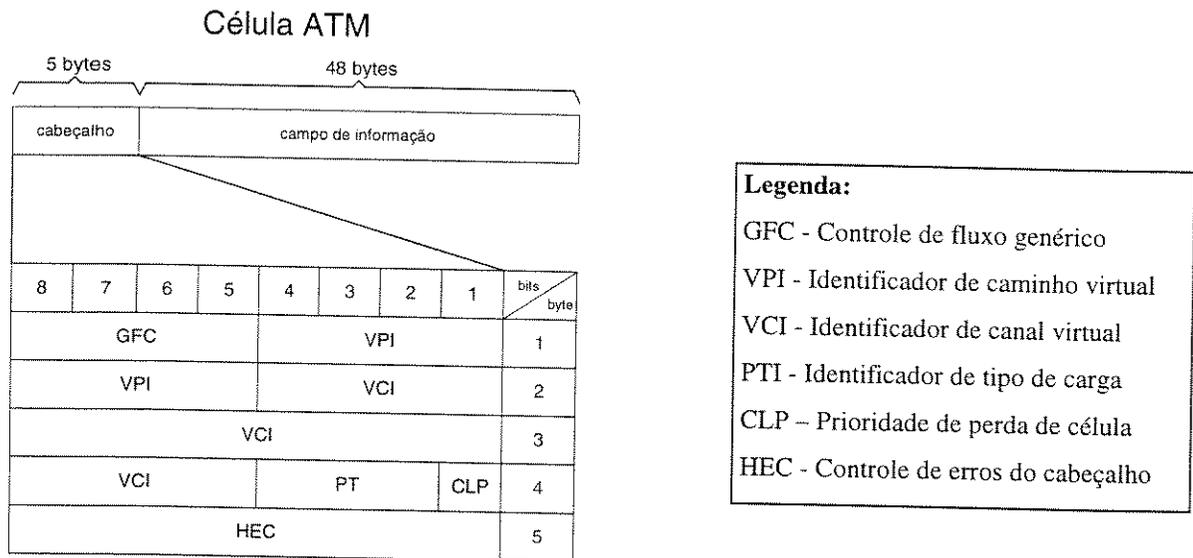


Fig. 4.1 - Célula ATM

Um dos problemas da utilização de multiplexação estatística é que pode ocorrer uma sobrecarga em um nó da rede ATM quando o número de células que chegam ao nó é superior a sua capacidade de processamento. Quando isto acontece este nó descarta algumas células até que se volte a condições normais de operação. Quanto a este aspecto, há células com dois níveis de prioridade. O nível de prioridade é determinado por um bit do cabeçalho da célula, indicado por CLP ("Cell Loss Priority") na Fig. 4.1. As células de baixa prioridade são descartadas primeiro, e garante-se, às células de alta prioridade, uma probabilidade desprezível de descarte. Esta característica pode ser explorada para transmissão de vídeo utilizando dois canais de diferentes níveis de prioridade, onde a informação essencial para a decodificação é enviada no canal de alta prioridade e no canal de baixa prioridade é enviada informação suplementar, de realces [1,15-17]. Esta informação suplementar pode estar relacionada com maior definição de altas frequências, melhor resolução espacial ou temporal, ou maior precisão no valor das amostras.

Como a transmissão de vídeo em redes ATM é uma aplicação em tempo real, deve-se considerar também, além da perda de células, os atrasos de transmissão e o "jitter" (i.e., a variação dos atrasos de transmissão). Vários trabalhos têm sido desenvolvidos objetivando a transmissão de vídeo em redes ATM [15,18-22]. Nesta dissertação, é abordado o problema de perdas de células e a utilização de técnicas de escalonamento para reduzir os efeitos deste problema. O tratamento do problema e os resultados são apresentados no Capítulo 5.

4.3 *Técnicas para Minimizar os Efeitos dos Erros de Canal*

O Padrão MPEG-2 não especifica codificação de canal para proteção contra erros de transmissão. Entretanto, algumas características deste padrão podem ser exploradas para diminuir a sensibilidade da seqüência codificada a erros, quando transmitida ou armazenada em meios ruidosos (ou seja, sujeitos a erros de transmissão ou armazenamento).

Tipicamente [4], quando os erros são detectados na recepção, o decodificador considera os dados perdidos como macroblocos saltados (“skipped macroblocks”), até o início da próxima fatia de macroblocos (“slice”). Desta forma, cada macrobloco perdido é substituído pelo de mesma posição espacial do quadro de referência anterior. Quando isto acontece em um quadro I, o erro se propaga por todo o GOP, ou seja, até que seja recebido um novo quadro I.

A solução apresentada pelo Padrão MPEG-2 para amenizar este problema consiste em enviar, opcionalmente, vetores de deslocamento codificados para quadros I. Estes vetores são utilizados quando há erros no receptor. Por exemplo, quando ocorrem erros em um determinado “slice” de um quadro (I, P ou B), os vetores de deslocamento do “slice” superior podem ser utilizados na substituição dos macroblocos perdidos por pixels do quadro de referência anterior (I ou P), considerando-se compensação de movimento. De acordo com B. G. Haskell et al. [4], em simulações realizadas, este método apresentou eficiência alta para taxas de erro de bits de até 10^{-4} , e eficiência aceitável para taxas de até 10^{-3} . Nestas simulações, os vetores de deslocamento para os macroblocos perdidos foram derivados dos vetores de deslocamento do “slice” superior. Pode-se também utilizar vetores de deslocamento do “slice” inferior, ou vetores de deslocamento do quadro anterior, neste caso, utilizando-se técnicas de busca telescópica, apresentadas na Seção 2.1.2.4 do Capítulo 2.

Como os “slices” são a menor unidade de sincronismo, pode-se definir “slices” menores para aumentar o número de pontos de sincronismo. No caso de redes ATM, pode-se utilizar “slices” de tamanhos adaptativos, de forma que o primeiro macrobloco de cada célula seja precedido por um cabeçalho de “slice” que esteja totalmente contido nesta célula. Desta forma, quando ocorre a perda de determinada célula, o erro não se propaga para as células seguintes, já que a próxima célula terá um ponto de resincronismo (cabeçalho de “slice”).

Outra forma de se recuperar um macrobloco perdido é através de interpolação espacial, utilizando-se os macroblocos vizinhos disponíveis. Esta técnica é especialmente interessante

quando ocorrem erros de transmissão em mudanças de cenas pois, nestes casos, o quadro de referência anterior não oferece condições de ser utilizado como preditor.

Um maior número de “slices” intracodificados ou de quadros I (GOP menores) pode ser utilizado para diminuir a propagação temporal do erro na seqüência, a custo de aumento de taxa.

As técnicas apresentadas podem ser utilizadas em conjunto e, neste caso, são razoavelmente eficientes para taxas de perda de células de até 10^{-3} [2]. Para taxas de perdas de células maiores, devem ser utilizadas técnicas de codificação em camadas, oferecendo maior proteção à camada base para garantir a reprodução de, pelo menos, uma versão de qualidade aceitável da seqüência de vídeo. Como já foi apresentado no Capítulo 3, o Padrão MPEG-2 possibilita a codificação em camadas de quatro diferentes modos: particionamento de dados, requantização dos coeficientes (SNR), escalabilidade espacial, e escalabilidade temporal. Todos estes modos podem ser utilizados para reduzir os efeitos de perdas de células, entretanto, dois deles são especialmente interessantes por apresentar menor complexidade de implementação e bom desempenho em situações de perda de informação das camadas superiores. São eles: o particionamento de dados e a requantização. Vários trabalhos [1,15-17] têm apresentado comparações entre a performance destes dois modos de escalonamento para transmissão através de canais com diferentes níveis de proteção contra erros, entretanto, ainda há muito a ser investigado.

Além dos meios que o Padrão MPEG-2 oferece para diminuir a sensibilidade a erros de transmissão, podem também ser utilizadas técnicas de codificação de canal. Por exemplo, o Padrão HDTV, que utiliza o MPEG-2 na etapa de codificação de vídeo, recomenda a utilização de códigos de treliça para proteção contra erros de bits, e técnicas de embaralhamento de dados aliadas a codificação Reed-Solomon para proteção contra perdas em “bursts” (longa seqüência de erros de bits). A codificação de Reed-Solomon é também aplicável no caso de transmissão de vídeo através de redes ATM, para minimizar o efeito de perda de células.

Estas técnicas de codificação de canal estão fora do escopo deste trabalho, pois aqui são consideradas situações em que as taxas de perda são superiores à capacidade de correção destas técnicas, tornado-as ineficientes. Para estas altas taxas de perdas, técnicas de escalonamento são utilizadas.

Capítulo 5

Simulações e Resultados

5.1 *Introdução*

No Capítulo 4 foram apresentadas técnicas para minimizar os efeitos dos erros de canal quando se transmite vídeo digital em redes RDSI-FL. Dentre estas técnicas, o escalonamento de dados mostra-se especialmente interessante para aplicações em redes ATM e por isso é objeto de estudo mais detalhado. Neste capítulo são apresentados resultados de simulações de sistemas que exploram o escalonamento de dados para transmissão de vídeo digital em redes ATM. Três diferentes esquemas de escalonamento foram implementados: o primeiro com separação espectral fixa, o segundo com particionamento de dados, e o último com requantização dos coeficientes espectrais. Cada um deles foi analisado em dois casos. No primeiro caso, os dados relativos aos dois canais (de alta e baixa prioridade) são utilizados na etapa de predição. Alternativamente, no segundo caso somente os dados relativos ao canal de alta prioridade são utilizados nesta etapa. Inicialmente consideram-se as situações em que os dois canais são recebidos sem erros no receptor, ou somente o canal de alta prioridade é recebido (i.e., quando há perda do canal de baixa prioridade). Posteriormente consideram-se casos onde o canal de baixa prioridade é sujeito a perda de células, enquanto a recepção do canal de alta prioridade é garantida. O desempenho destes sistemas são comparados utilizando-se curvas de distorção x taxa.

5.2 *Codificador*

Foi implementado, utilizando a linguagem ANSI C como ferramenta, um codec baseado no Padrão MPEG-2. O codificador implementado possibilita o escalonamento da seqüência digital codificada em duas camadas utilizando os seguintes critérios:

- (1) particionamento de dados (“data partitioning”);
- (2) separação espectral fixa;

(3) requantização dos coeficientes (“SNR scalability”).

A primeira camada (de alta prioridade) é essencial para decodificação, enquanto a camada de baixa prioridade representa informação adicional (e.g., de realce, ou detalhamento) para uma melhor qualidade da seqüência reconstruída.

Na implementação do codec não houve preocupação em enquadrá-lo em determinado nível (“level”) ou perfil (“profile”) do Padrão MPEG-2, já que o particionamento de dados não é suportado por nenhum dos níveis ou perfis estabelecidos no padrão [2], constituindo-se de pós-processamento e pré-processamento em relação ao codec do padrão. Algumas restrições foram feitas para simplificar a implementação do codec, e são citadas a seguir:

- 1)- A codificação é sempre realizada no modo progressivo (“progressive_sequence” = ‘1’ [2]). Caso a seqüência a ser codificada seja entrelaçada, os campos par e ímpar são unidos para formarem quadros, e na recepção, se necessário, é realizado processo inverso para se recuperar os campos separadamente.
- 2)- A compensação de movimento é sempre baseada em quadro (“frame-based prediction”) e para isso atribui-se “progressive_frame” = ‘1’ e “frame_pred_frame_dct” = ‘1’ [2]. Portanto, o modo “dual prime” não é utilizado. Da mesma forma, o modo 16 x 8 MC também não se aplica, pois ele só pode ser utilizado quando a compensação de movimento é baseada em campo (“field-based prediction”).
- 3)- O cálculo da transformada DCT é baseado em quadro (“frame DCT coded”) o que também é garantido atribuindo “progressive_frame” = ‘1’ e “frame_pred_frame_dct” = ‘1’ na extensão do cabeçalho de quadro (“picture coding extension”).
- 4)- No codificador implementado, o critério de busca pelo vetor de deslocamento é o casamento de macroblocos, utilizando-se uma figura de mérito que pode ser a média da diferença entre pixels (MAD), o erro quadrático médio (MSE) ou ainda a correlação cruzada normalizada (NCC) entre os macroblocos. A busca do melhor casamento pode ser exaustiva ou em três passos. Nas simulações realizadas, foram utilizados o erro quadrático médio, como figura de mérito, e a busca em três passos, como estratégia de busca. O tamanho da janela de

busca foi de 46 x 46 pixels, limitando as dimensões, horizontais e verticais, dos vetores de deslocamento ao intervalo [-15,15].

5)- A decisão de se codificar ou não um macrobloco de quadros do tipo P ou B é feita calculando-se o MSE entre o macrobloco original e a sua predição, e comparando-se o valor obtido com um limiar predeterminado. Se o MSE for maior que o limiar o macrobloco é codificado, caso contrário ele é saltado (“skipped macroblock”). Nas simulações realizadas o limiar foi de 80% do valor do MSE do primeiro quadro da seqüência.

6)- Dado que um macrobloco de um quadro B é codificado, ele pode ser predito (em relação ao quadro de referência anterior), interpolado, ou intracodificado. Esta escolha é feita de modo a minimizar a energia média dos dados a serem codificados. Sejam $Q_a(x,y)$ o quadro atual, $Q_{ref}(x,y)$ o quadro de referência posterior, (a,b) a posição do macrobloco atual, $(d1x,d1y)$ o deslocamento do macrobloco em relação à Q_{ref1} , $(d2x,d2y)$ o deslocamento do macrobloco em relação à Q_{ref2} , calcula-se:

$$MSE_{ref1} = \frac{1}{16 \cdot 16} \sum_{m=a}^{a+16} \sum_{n=b}^{b+16} [Q_a(m,n) - Q_{ref1}(m+d1x, n+d1y)]^2 \quad (5.1)$$

$$MSE_{ref2} = \frac{1}{16 \cdot 16} \sum_{m=a}^{a+16} \sum_{n=b}^{b+16} [Q_a(m,n) - Q_{ref2}(m+d2x, n+d2y)]^2 \quad (5.2)$$

$$MSE_B = \frac{1}{16 \cdot 16} \sum_{m=a}^{a+16} \sum_{n=b}^{b+16} \left[Q_a(m,n) - \frac{Q_{ref1}(m+d1x, n+d1y) + Q_{ref2}(m+d2x, n+d2y)}{2} \right]^2 \quad (5.4)$$

$$E_m = \frac{1}{16 \cdot 16} \sum_{m=a}^{a+16} \sum_{n=b}^{b+16} [Q_a(m,n)]^2 \quad (5.5)$$

A escolha do modo que cada macrobloco de um quadro B é codificado é feita da seguinte forma:

- se $\min[MSE_{ref1}, MSE_{ref2}, MSE_B, E_m] = MSE_{ref1}$, o macrobloco é predito;
- se $\min[MSE_{ref1}, MSE_{ref2}, MSE_B, E_m] = MSE_{ref2}$, o macrobloco é predito em relação ao quadro de referência posterior;
- se $\min[MSE_{ref1}, MSE_{ref2}, MSE_B, E_m] = MSE_B$, o macrobloco é interpolado;
- se $\min[MSE_{ref1}, MSE_{ref2}, MSE_B, E_m] = E_m$, o macrobloco é intracodificado;

onde $\min[a, b, c, d]$ é uma função que retorna o menor de seus argumentos, a , b , c ou d .

Nas simulações realizadas, a figura de mérito utilizada na etapa de estimação de movimento é o MSE, então os valores de MSE_{ref1} e MSE_{ref2} são calculados na etapa de estimação de movimento. Desta forma, somente MSE_B e E_m precisam ser calculados na presente etapa.

Para macroblocos dos quadros P, a escolha do modo de codificação é semelhante ao caso apresentado, entretanto o macrobloco só pode ser predito ou intracodificado. Neste caso, somente E_m precisa ser calculado.

7)- A ordenação dos coeficientes é feita utilizando-se a matriz apresentada na Fig. 2.17a, que corresponde a primeira das duas matrizes de ordenação especificadas no Padrão MPEG-2 [2]. Dessa forma, faz-se “alternate_scan” = 0 [2].

8)- O Padrão MPEG-2 recomenda duas tabelas VLC (no padrão [2], tabelas B.14 e B.15) para a codificação dos coeficientes. No caso de coeficientes intracodificados, qualquer uma das duas pode ser utilizada. Esta escolha é transmitida através da variável “intra_vlc_code” (ivc), que é transmitida no cabeçalho de cada quadro. Por outro lado, na codificação dos coeficientes de blocos não intracodificados, somente a tabela B.14 (ver apêndice B) deve ser utilizada. Nas simulações realizadas foi utilizada a mesma tabela VLC para os coeficientes intracodificados¹ e não intracodificados (“intra_vlc_code” = ‘0’). Esta escolha foi baseada nos resultados de simulações apresentados nos gráficos da Fig. 5.1, onde, para taxas abaixo de 0.5 bits/pixel, observa-se uma clara vantagem quando utiliza-se “intra_vlc_code” = ‘0’.

¹ Os coeficientes DC dos blocos intracodificados são codificados a parte, como apresentado no Capítulo 3.

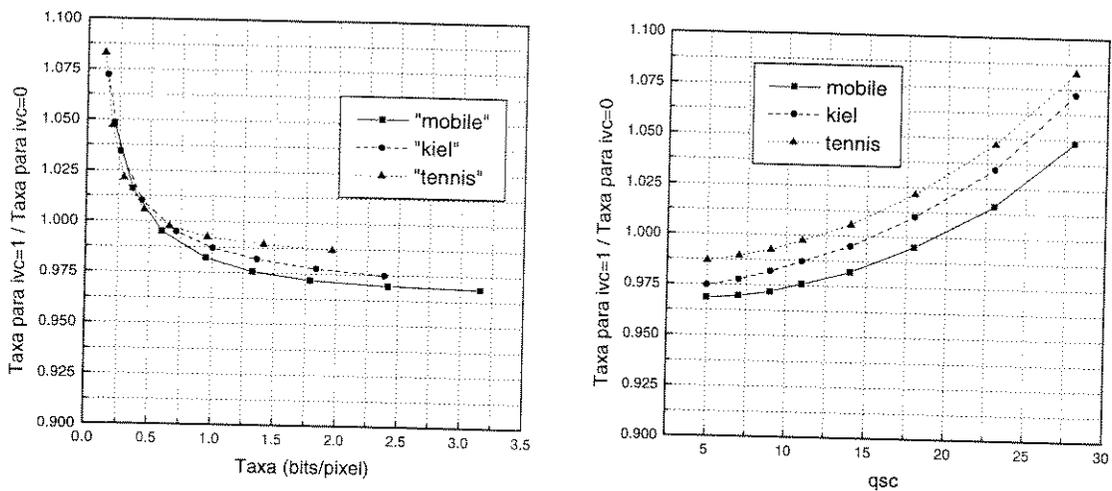


Fig. 5.1 - Relação entre taxas para ivc=1 e ivc=0

Algumas destas restrições prejudicam o ganho de codificação do sistema, porém não comprometem os resultados globais que consistem em comparações de desempenho entre os modos de escalonamento investigados.

O codificador implementado não possui controle de taxa, pois deseja-se taxa de transmissão variável (VBR - "variable bit rate") para se obter qualidade de reprodução aproximadamente constante. Para se minimizar as variações na qualidade de reprodução é necessário um sistema que controle os parâmetros do codificador objetivando esta finalidade. Entretanto, para evitar que os resultados ficassem dependentes do desempenho deste sistema de controle, e para simplificar a implementação do codificador, foi feita uma aproximação onde fixa-se os passos de quantização para cada tipo de quadro, I, P, e B. Esta aproximação tem sido utilizada em trabalhos [1,15,17] que tratam de sistemas de codificação de vídeo com requantização e transmissão a taxas variáveis (VBR). Com isto, obtém-se uma qualidade de reprodução aproximadamente constante para determinada seqüência, desde que esta não apresente grandes mudanças em suas características. Por exemplo, quando ocorrem mudanças de cena, pode haver variação significativa na qualidade de reprodução da seqüência.

Os passos de quantização são determinados pelas matrizes de quantização e pela variável "quantiser_scale", que depende de outras duas variáveis, QSC ("quantiser_scale_code") e "q_scale_type". A variável "q_scale_type" determina qual tabela (Tabela 7.6 do Padrão H.262 [2]) é utilizada para derivar "quantise_scale" a partir de QSC. Nas simulações realizadas a tabela utilizada foi a logarítmica, "q_scale_type" = 1. Os valores

de QSC são escolhidos para cada tipo de quadro, I, P, e B e são utilizados para todos os quadros da seqüência.

Outra característica do codec implementado é que ele utiliza o formato 4:2:0 (“croma_format” = ‘01’ [2]). Neste formato, os sinais de crominância são dizimados, tanto na horizontal quanto na vertical, por um fator de 2 antes de serem codificados. Os filtros utilizados neste processo foram os seguintes:

- dizimação horizontal: $(1 \ 2 \ 1) / 4$,
- dizimação vertical: $(5 \ 11 \ -21 \ -37 \ 70 \ 228 \ 228 \ 70 \ -37 \ -21 \ 11 \ 5) / 512$,
- interpolação vertical: $(3 \ 7 \ -16 \ -32 \ 67 \ 227 \ 227 \ 67 \ -32 \ -16 \ 7 \ 3) / 256$,
- interpolação horizontal: $(1 \ 2 \ 1) / 2$.

A seguir são apresentados os modos de escalonamento do codec implementado, com uma breve descrição de cada um deles e seus respectivos diagramas de bloco.

5.2.1 Codificador com separação espectral

No Padrão MPEG-2 a separação espectral é obtida através de uma técnica chamada “data partitionig” (particionamento de dados). Nesta técnica o escalonamento de dados é feito através da inserção de um ponto de quebra (“priority breakpoint”), no cabeçalho de cada “slice”, para indicar o coeficiente de transição entre as duas camadas. Para se conseguir a separação espectral, a variável “priority_breakpoint” deve assumir valores $63+j$ onde $j \in \{1,2,\dots,64\}$, e representa o número máximo de coeficientes não nulos de um bloco que são alocados na camada de alta prioridade. Os coeficientes restantes de cada bloco são enviados na camada de baixa prioridade. Nesta técnica as camadas de alta e baixa prioridade são chamadas de partição 0 e partição 1, respectivamente.

Alguns elementos sintáticos pertencentes a partição 0 são incluídos na partição 1, de forma redundante, para facilitar a recuperação do sinal quando ocorrem erros. Essa informação redundante consiste de vários cabeçalhos tais como os cabeçalhos de seqüência (“sequence header”), de grupo de quadros (“group of pictures header”) e de quadro (“picture header”). O indicador de fim de seqüência (“sequence end code”) também é repetido nas duas partições. Estes elementos sintáticos redundantes e outros cabeçalhos que fazem parte da partição 1, como por exemplo os cabeçalhos de “slice”, representam um aumento de taxa quando se compara o “data partitioning” com um sistema sem escalonamento. Entretanto este aumento de taxa é pequeno, tipicamente, da ordem de 1-2% [1]. Esta é a maior vantagem de

se utilizar este esquema de escalonamento. Por outro lado, uma desvantagem deste esquema é que, fixada uma determinada taxa para a camada base (camada de alta prioridade), a qualidade de reprodução desta camada (quando só ela é recebida) é inferior à obtida quando o escalonamento é realizado através de requantização dos coeficientes [1]. A Fig. 5.2 ilustra um exemplo de particionamento de dados onde “priority breakpoint” é igual a 65, o que corresponde a, no máximo, 2 coeficientes não nulos alocados na camada base. Os coeficientes AC apresentados nesta figura são os não nulos, sendo que ACn representa o código referente ao par corrida de zeros/nível mais o bit de sinal deste coeficiente. As setas numeradas indicam a ordem de decodificação.

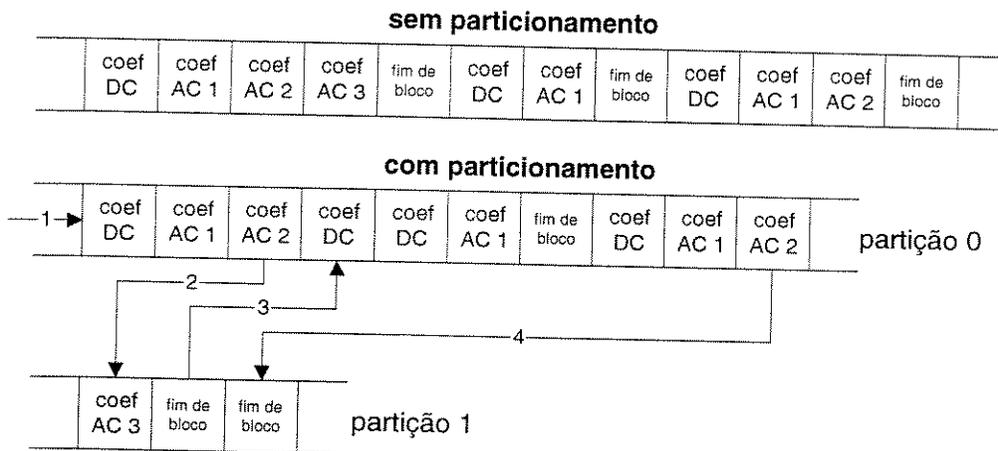


Fig. 5.2 - Seqüência codificada segmentada em duas partições

Outra forma de se realizar a separação espectral é fixando o número de coeficientes, N, a serem alocados em cada camada. Neste caso, os N primeiros coeficientes de cada bloco (de acordo com ordenamento em zig-zag) são colocados na camada de alta prioridade e os N-64 restantes são colocados na camada de baixa prioridade. Para diferenciá-lo do “data partitioning”, este esquema será chamado de separação espectral fixa. A realização de separação espectral fixa através de “data partitioning” só seria possível se a variável “priority breakpoint” fosse alterada para cada bloco. Já que, de acordo com o Padrão MPEG-2, esta variável só é alterada no início de cada “slice”, então não é possível realizar separação espectral fixa através de “data partitionig”. Como a única maneira de se realizar separação

espectral, de acordo com o Padrão MPEG-2, é o “data partitioning”, este padrão não suporta separação espectral fixa. Entretanto, ao codec implementado foi adicionado esse modo de escalonamento (separação espectral fixa) para fins de comparações. O overhead adicional deste esquema de separação são os símbolos de fim de bloco (EOB - “end of block”) que deverão ser enviados na camada de alta prioridade sempre que o $N^{\text{ésimo}}$ coeficiente do bloco for nulo.

Em algumas situações, a introdução dos símbolos de fim de bloco adicionais pode representar não um aumento, mas sim uma redução de taxa. Isto ocorre quando a soma do número de bits da palavra-código EOB com o número de bits da próxima palavra-código (que representa o primeiro coeficiente não nulo alocado na camada de baixa prioridade) é menor que o número de bits da palavra-código que seria utilizada caso o fim de bloco adicional não fosse inserido, especialmente quando, neste último caso, o código de “escape” é utilizado. Esta situação pode ser exemplificada supondo-se que determinado bloco de coeficientes, já quantizados e ordenados, de um quadro predito é composto pelo primeiro coeficiente com valor 4, o décimo primeiro com valor 3 e os coeficientes restantes nulos. Utilizando-se as Tabelas B.1 e B.2 do Apêndice B (Tabelas B.14 e B.16 do Padrão MPEG-2 [2]) para codificar estes coeficientes, são necessários 18 bits caso esteja sendo utilizado o método da separação espectral fixa, com 10 coeficientes alocados na camada de alta prioridade. Por outro lado, se o “data partitioning” for utilizado (ou se não houver escalonamento), são necessários 34 bits para codificar este mesmo bloco. A Fig. 5.3 apresenta as palavras código utilizadas em cada caso. Apesar do exemplo exposto, em média, a inserção destes fins de bloco adicionais representam pequeno overhead.

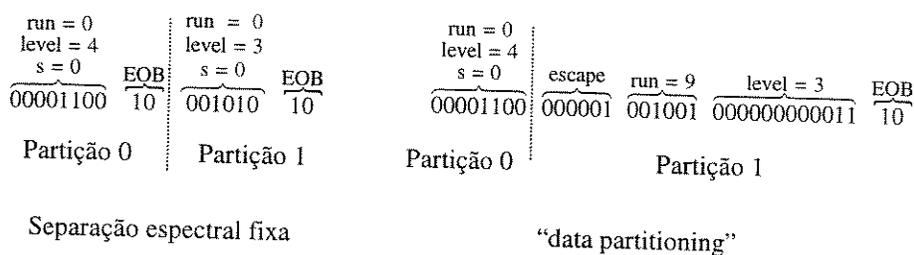


Fig. 5.3 - Bloco codificado

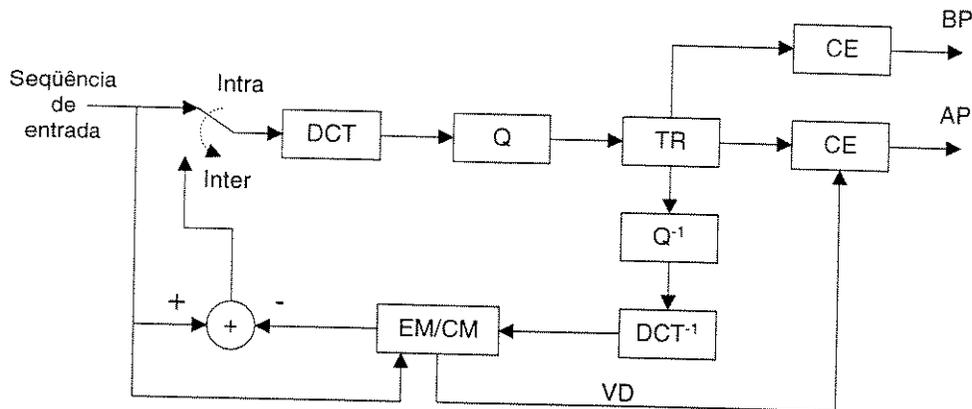


Fig. 5.4 - Codificador com separação espectral

O diagrama geral de um codificador com separação espectral é apresentado na Fig. 5.4. Este diagrama representa tanto o esquema com separação espectral fixa quanto o esquema com “data partitioning”, sendo que as diferenças entre estes dois esquemas estão nas etapas de triagem (TR) dos coeficientes espectrais, bem como na codificação de entropia (CE). Cada quadro da seqüência de entrada pode ser codificado de dois modos: codificação intraquadro (Intra) ou codificação interquadros (Inter). Neste último só é codificada a diferença entre o quadro e a sua previsão. A previsão pode ser feita utilizando-se o quadro de referência anterior ou dois quadros de referência, um anterior e outro posterior. No primeiro caso o quadro codificado é chamado de predito (quadro P), enquanto que no segundo é chamado de quadro interpolado (quadro B). Somente os quadros intracodificados e os preditos podem ser utilizados como referência na etapa de previsão.

O primeiro quadro de uma seqüência é sempre intracodificado, visto que ainda não há quadros de referência disponíveis no preditor. Este quadro é dividido em blocos de 8 x 8 pixels, e a cada bloco é aplicada a transformação DCT. Em seguida os coeficientes são quantizados (Q). O próximo passo é o processo de triagem (TR) dos coeficientes. A forma de escolha dos coeficientes que são enviados em cada canal (de alta prioridade, AP, ou baixa prioridade, BP) já foi descrita, tanto para a separação espectral fixa quanto para o “data partitioning”. Os coeficientes passam então por um processo de codificação de entropia (CE)

que envolve codificação de Huffman e codificação “runlength”. Finalmente, os coeficientes são enviados através de seus respectivos canais.

Na etapa de triagem também é feita a escolha dos dados que serão utilizados na etapa de predição. Estes dados podem ser referentes às camadas de alta e baixa prioridade ou somente referentes à camada de alta prioridade. É bom lembrar que o decodificador descrito pelo Padrão MPEG-2 com “data partitionig” utiliza, na etapa de predição, os dados das partições 0 e 1, que correspondem aos canais de alta e baixa prioridade, respectivamente. Entretanto, no decodificador implementado a escolha dos dados a serem utilizados na etapa de predição é feita de maneira similar a do codificador, possibilitando a utilização exclusiva de dados de alta prioridade para estimar o preditor. Os coeficientes que serão utilizados na etapa de predição passam, então, por um processo de quantização inversa (Q^{-1}) seguido de uma transformação DCT inversa (DCT^{-1}). Neste ponto tem-se um quadro reconstruído que é usado como referência, por outros quadros, nas etapas de estimação e compensação de movimento.

O processo que foi descrito é para codificação intraframe de um quadro. Caso a codificação seja “interframe” o processo é semelhante. Neste caso, apenas a diferença entre o quadro atual e a sua predição é codificada. Se o quadro codificado é do tipo B não é necessário que ele seja recuperado no codificador, já que ele não é utilizado como referência na etapa de estimação e compensação de movimento.

Os vetores de deslocamento (VD) resultantes da etapa de estimação e compensação de movimento são codificados utilizando-se código de Huffman (codificação de entropia - CE) e em seguida são enviados no canal de alta prioridade.

5.2.2 Codificador com requantização dos coeficientes

A requantização dos coeficientes da DCT também é um método utilizado para se codificar uma seqüência de vídeo digital em duas camadas a fim de se oferecer proteção contra erros de transmissão (e.g., através de redes ATM ou transmissão “broadcasting”). A primeira camada, que contém coeficientes grosseiramente quantizados, é essencial para a decodificação e será chamada camada de alta prioridade. Por outro lado, a segunda camada contém dados para refinamento destes coeficientes e corresponde a camada de baixa prioridade. Já que a camada de baixa prioridade tem como finalidade melhorar a relação sinal ruído (SNR - “signal to noise ratio”) da seqüência reconstruída, este esquema de separação em

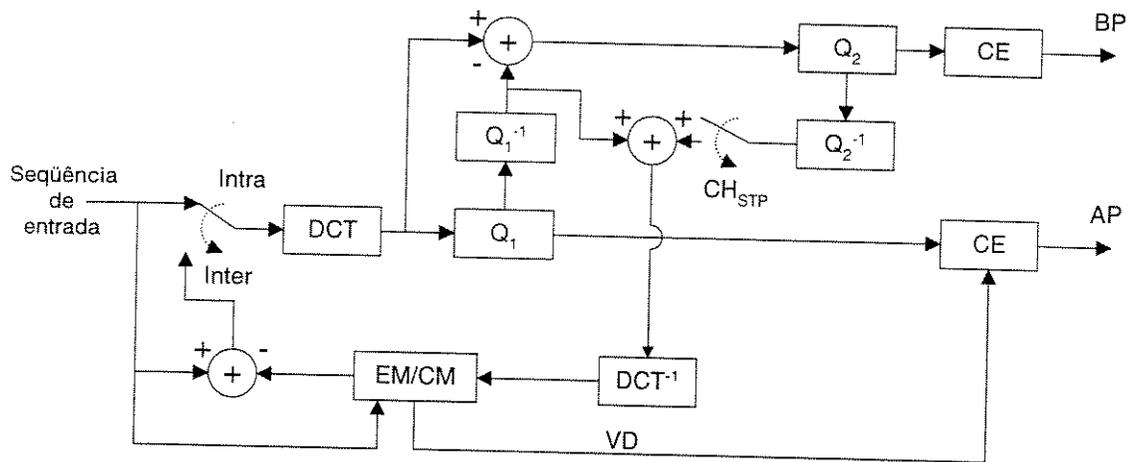


Fig. 5.5 - Codificador com requantização dos coeficientes

camadas é chamado, no Padrão MPEG-2, de escalonamento SNR (“SNR scalability”). A desvantagem deste esquema é que o "overhead" de taxa é maior que o do “data partitioning”.

Na Fig. 5.5 é apresentado o diagrama geral de um codificador com requantização dos coeficientes. O processo de predição de quadros, que envolve estimação e compensação de movimento, é semelhante ao do codificador com separação espectral. Neste caso, os dados a serem codificados (o quadro atual ou a diferença entre este e sua predição) passam pela etapa de transformação DCT e os coeficientes resultantes são quantizados grosseiramente por Q_1 . Estes coeficientes quantizados passam pela etapa de codificação de entropia (CE) e são enviados no canal de alta prioridade (AP). Estes mesmos coeficientes são recuperados e a diferença entre estes e os coeficientes originais é calculada. Esta diferença é quantizada em Q_2 e os dados resultantes são enviados, após a codificação de entropia, através canal de baixa prioridade (BP). Se a etapa de predição é baseada nos dados de alta e baixa prioridade, estes dados são recuperados em Q_2^{-1} , somados com a saída de Q_1^{-1} e esta soma é encaminhada para a etapa de transformação DCT inversa (DCT^{-1}). Deste ponto, os coeficientes reconstruídos alimentam o preditor. A escolha dos dados utilizados na estimação do preditor (dados referentes aos canais BP e AP ou somente referentes ao canal AP) é feita através da chave seletora de tipo de predição (CH_{STP}).

A Fig. 5.6 apresenta um exemplo de requantização de coeficientes onde são considerados quatro coeficientes de um bloco intracodificado. Neste exemplo, utiliza-se $L_1=8$, como passo de quantização de Q_1 , e $L_2=2$, como passo de quantização em Q_2 .

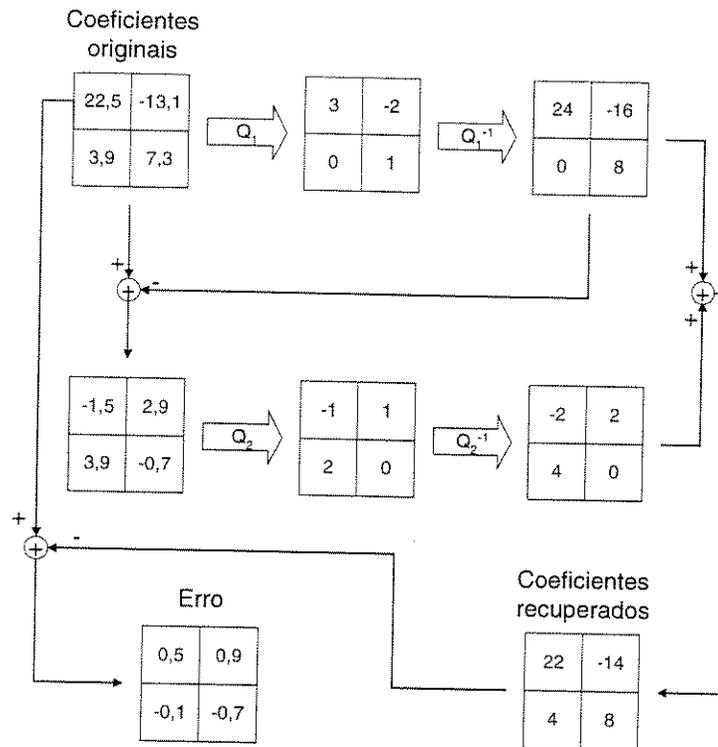


Fig. 5.6 – Exemplo de requantização de coeficientes

5.3 Modelagem do Congestionamento de Tráfego

Como já foi apresentado no Capítulo 4, um dos principais problemas de uma rede ATM é a perda de células que ocorre devido ao congestionamento de um nó desta rede. Isto se deve ao fato que estas redes são baseadas em multiplexação estatística. Modelar perdas de células não é uma tarefa simples, já que, como vem sendo apresentado em vários trabalhos [5,27-29], o tráfego destas redes apresenta características auto-similares. Aproximações através de modelos poissonianos e “on-off” (em surtos) podem ser insatisfatórias.

Para as presentes simulações, deseja-se um modelo que represente as perdas de células ao longo de toda a rede, do transmissor ao receptor. Um modelo complexo necessita de um grande número de parâmetros e é dependente de determinada topologia, o que dificulta a análise dos resultados. Por isso, considerou-se a utilização de dois modelos bem simples: um utilizando um tráfego agregado de seqüências de vídeo e o outro uma cadeia de markov de dois estados. Ambos consideram uma característica importante para a modelagem, que é a correlação entre perdas de células.

O sistema implementado possibilita a utilização de qualquer um destes modelos, entretanto, nas presentes simulações foi utilizado o modelo baseado em tráfego agregado para representar perdas de células. Nestas simulações considerou-se um tráfego agregado representado pela soma de nove tributários de vídeo (MPEG). As taxas geradas pelos tributários foram obtidas nos "sites" das Universidades de Wuerzburg [14] (Alemanha) e da Califórnia [26] (Berkeley, Estados Unidos). Estas taxas foram somadas, e aplicou-se um fator de proporcionalidade pois as dimensões dos quadros codificados, que geraram estas estatísticas, são menores que as dimensões dos quadros das seqüências utilizadas nas simulações. Para representar perdas de células, considera-se um limiar que quando ultrapassado pela soma do tráfego agregado com o tráfego gerado pela seqüência testada, resulta em perdas de células de baixa prioridade. Para se obter as taxas de perdas de células desejadas nas simulações, este limiar é deslocado para cima e para baixo em passos logarítmicos até que se obtenha uma taxa de perdas (para a seqüência em teste) o mais próxima possível da taxa de perdas desejada.

A Fig. 5.7 apresenta um exemplo do processo de simulação de perda de células. Neste exemplo considera-se o tráfego agregado de um período equivalente a 60 quadros. O procedimento para escolha do limiar que é utilizado na simulação de perdas de células é o seguinte:

Passo 1: dois limiares são estabelecidos: um limiar mínimo (mínimo instantâneo da curva de tráfego agregado), e um limiar máximo (máximo instantâneo da curva de tráfego agregado).

Passo 2: calcula-se o limiar médio entre o mínimo e o máximo.

Passo 3: considera-se que, quando o tráfego agregado ultrapassa o limiar médio calculado, a informação do canal de baixa prioridade do sinal de interesse é perdida. Calcula-se a taxa de perda de dados do sinal de interesse. Se esta taxa satisfizer (dentro de uma margem de erro pré-estabelecida) a taxa de perda de dados desejada para o presente caso, este limiar é utilizado para simular perdas de células e encerra-se a etapa de busca pelo limiar desejado.

Passo 4: caso contrário, se esta taxa de perdas for menor que a desejada (como no exemplo da Fig. 5.7) calcula-se um novo limiar tomando-se a média entre o limiar atual e o limiar imediatamente inferior (no exemplo, o limiar mínimo), e volta-se ao Passo 3. Se a taxa de

perdas for maior, o novo limiar é calculado entre o limiar atual e o limiar imediatamente superior. Volta-se ao Passo 3 até que se encontre um limiar satisfatório.

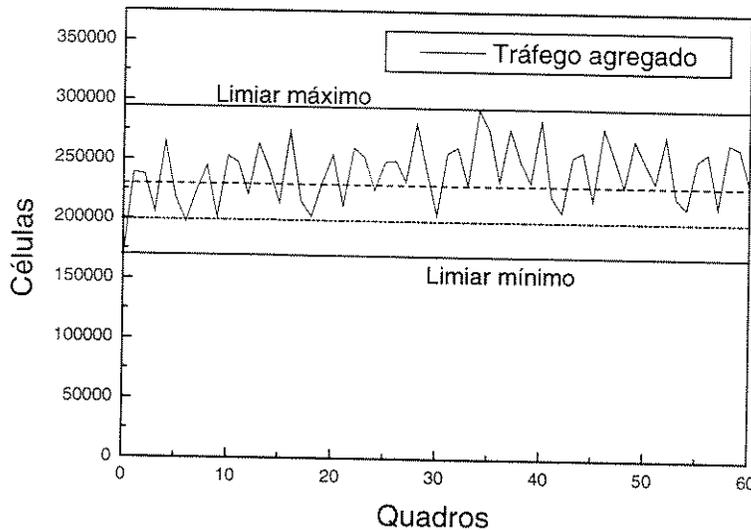


Fig. 5.7 – Exemplo de tráfego agregado para simulação de perdas de células

5.4 Simulações e Análise dos Resultados Obtidos

Nas simulações realizadas foram utilizadas três seqüências de vídeo digital: “Kiel”, “Mobile”, e “Tennis”. O primeiro quadro de cada uma destas seqüências é apresentado nas Figs. 5.8, 5.9, e 5.10. Estas seqüências apresentam características que as possibilitam representar bem o universo de seqüências reais. Dentre estas características, pode-se citar presença de objetos com deslocamentos lineares e circulares, zoom (bastante marcante na seqüência “Kiel”) e movimentos panorâmicos de câmera (“pan”). As seqüências originais foram obtidas junto ao CPQD-Telebrás e tem formato PPM (“Portable Pixmap”) - “raw”. Neste formato cada pixel é representado pelas três componentes de cor: vermelha (R - “red”), verde (G - “green”) e azul (B - “blue”), e para representar cada componente é utilizado um byte, possibilitando $2^8 = 256$ níveis de intensidade para cada um deles.

Estas seqüências foram obtidas através da digitalização de seqüências analógicas entrelaçadas, portanto, nas regiões onde há movimento aparecem deslocamentos entre os campos par e ímpar de um quadro. Como o codificador implementado é seqüencial, para ajustar estas seqüências a esta característica, elas foram submetidas a um processo de conversão temporal [4] no qual foram convertidas em seqüências não entrelaçadas. O quadro

não entrelaçado é gerado copiando-se o campo ímpar e filtrando-se o campo par utilizando-se os seguintes taps: $(-1 \ 4 \ 2 \ 4 \ -1) / 8$. O tamanho dos quadros foi feito igual a 720×480 para que as duas dimensões fossem múltiplo de 16, possibilitando que cada quadro possa ser representado por um número inteiro de macroblocos.

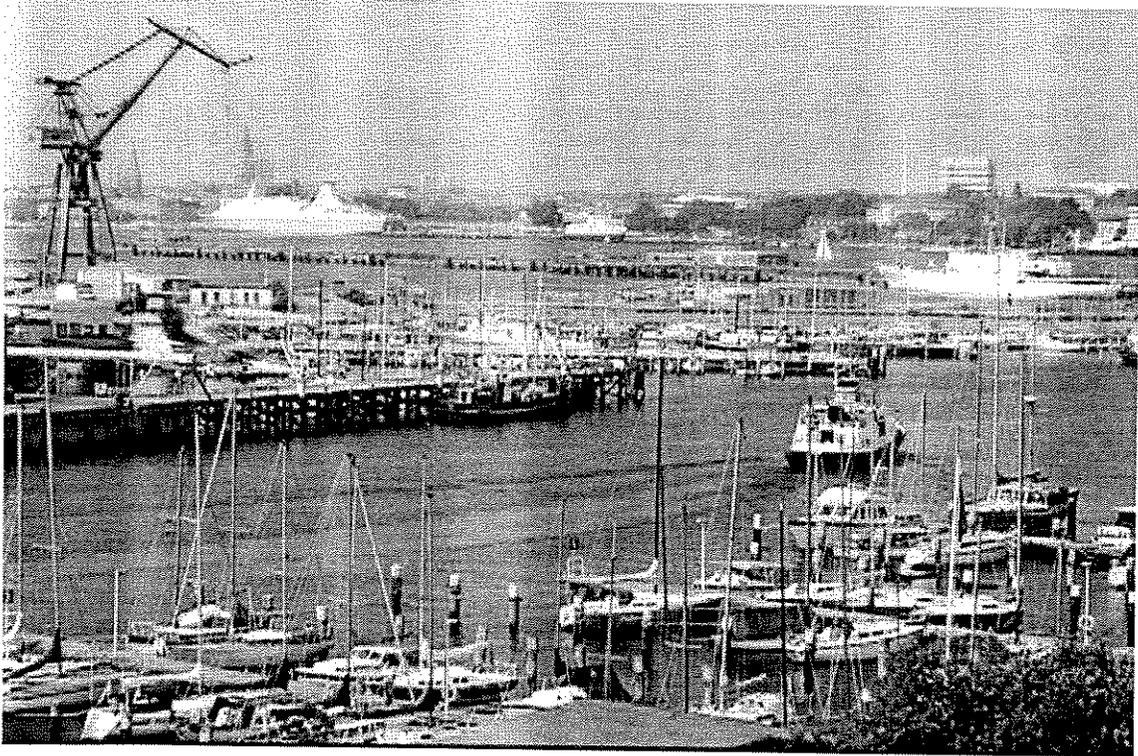


Fig. 5.8 - Primeiro quadro da seqüência "Kiel"

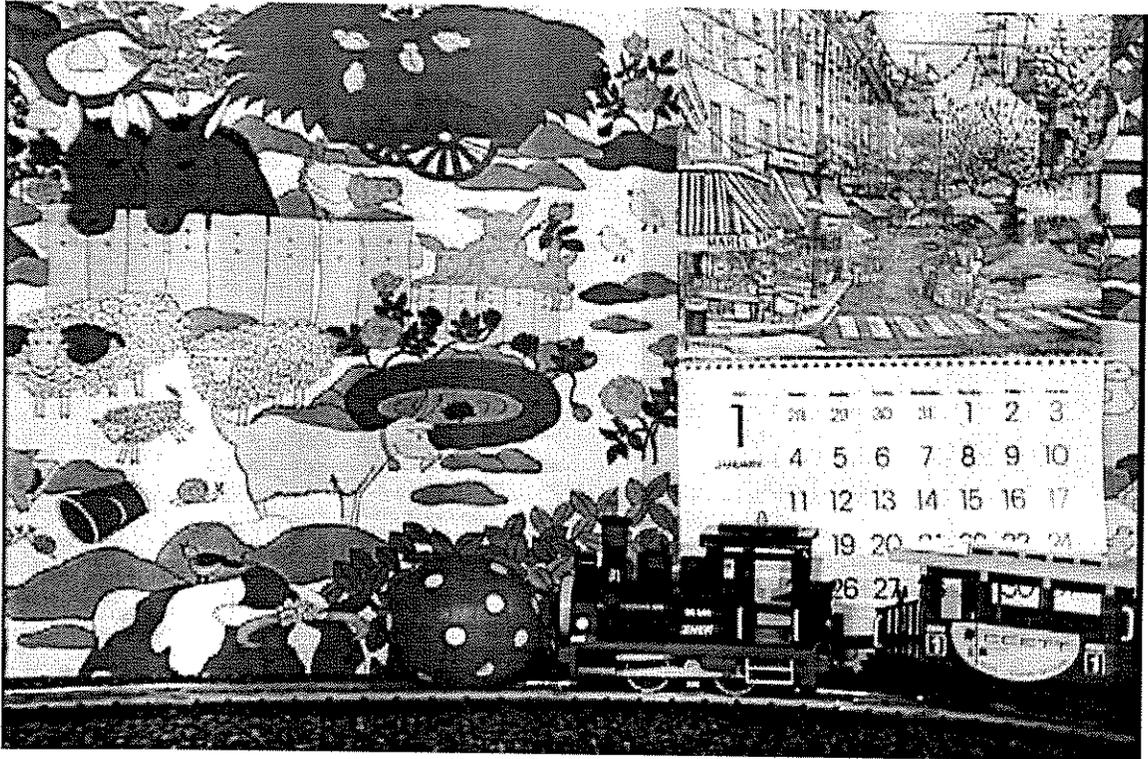


Fig. 5.9 - Primeiro quadro da seqüência "Mobile"

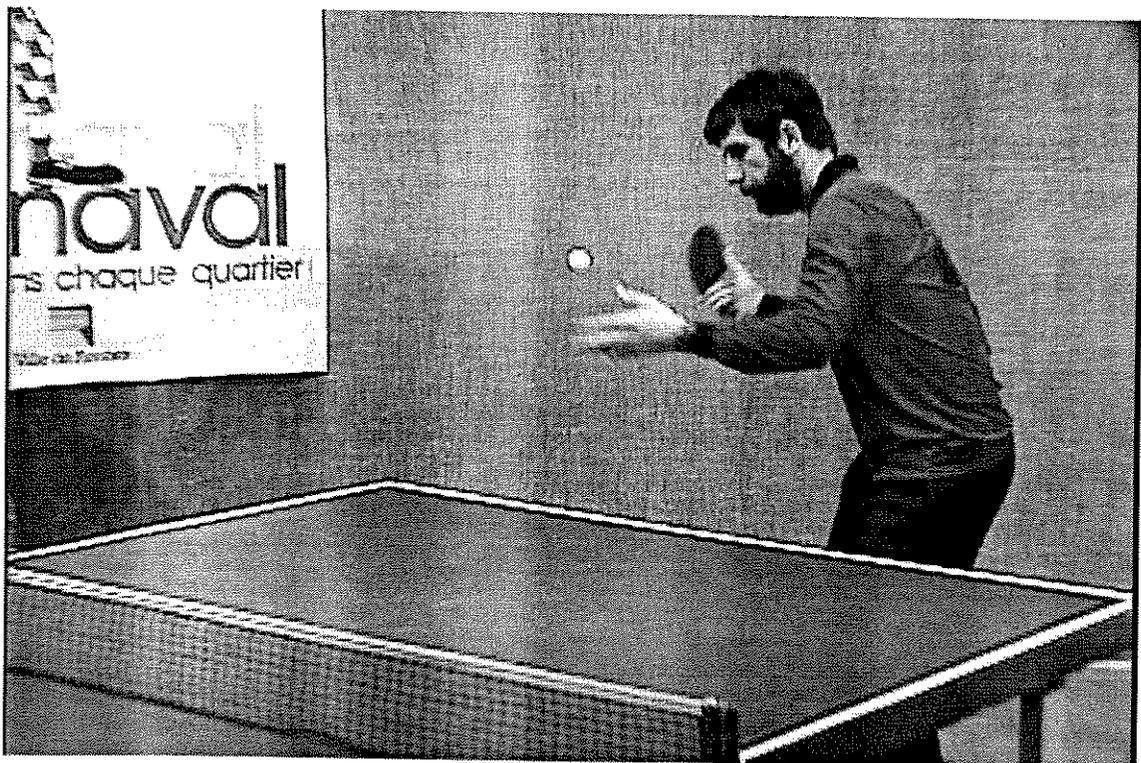


Fig. 5.10 - Primeiro quadro da seqüência "Tennis"

Utilizando-se o codec implementado, três diferentes esquemas de escalonamento foram testados através de simulações:

- separação espectral fixa (SE);
- particionamento de dados (DP - “Data Partitioning”);
- requantização dos coeficientes espectrais (RQ).

Cada um deles foi analisado em dois casos. No primeiro caso, os dados relativos aos dois canais (de alta e baixa prioridade) são utilizados na etapa de predição (PDC). No segundo, somente os dados de alta prioridade são utilizados na etapa de estimação do preditor (PAP).

Na primeira etapa de testes duas situações são testadas:

- na primeira, não há perda de nenhuma célula transmitida
- na segunda, o canal de baixa prioridade é completamente perdido.

Os resultados são comparados com um caso em que não há escalonamento de dados, ou seja, toda a informação é enviada por um único canal.

Em uma segunda etapa considera-se que o canal de baixa prioridade é sujeito a perdas de células de modo que as taxas de perda do tráfego total (soma do tráfego dos dois canais) sejam de: 1%, 5%, 10%, 20%, 30%, 40%. Estas taxas podem parecer altas, entretanto elas se referem a um período de tempo que, em outras aplicações, é considerado curto (e.g., da ordem de minutos, em aplicações que não requerem transmissão em tempo real), no qual ocorre um surto de perda de células. Na maioria das aplicações em que dados são transmitidos estes surtos de perdas não são um problema grave. Entretanto, quando se trata de transmissão de vídeo este surto de perdas é significativo até quando ele dura poucos segundos, já que muitas vezes o vídeo representa a transmissão ao vivo de algum programa².

Os resultados são apresentados na forma de curvas de distorção x taxa, com a taxa dada em bits por pixel, e a distorção representada pelo RMSE médio (considerando as componentes de cor R, G e B), dado pela equação (5.6).

² Por exemplo, a perda do sinal de vídeo durante a exibição de um jogo, no momento em que é marcado um gol ou quando acontece um lance importante, certamente provocará grande frustração nos espectadores, sendo que, geralmente são eles quem pagam (direta ou indiretamente) pelo serviço de transmissão.

$$RMSE_{RGB} = \sqrt{\frac{MSE_R + MSE_G + MSE_B}{3}} \quad (5.6)$$

Os passos de quantização foram escolhidos de forma que, para a seqüência “Tennis”, as taxas variassem aproximadamente entre 0.1 e 2 bits/pixel. Dessa forma, foram utilizados valores de “quantiser_scale_code” [2] iguais a 5, 7, 9, 11, 14, 18, 23, 28, que corresponde a fatores de escala do quantizador (“quantiser_scale” [2]) iguais a: 10, 14, 18, 22, 28, 36, 46, 56.

Como cada quadro das seqüências utilizadas na simulações tem dimensões de 720 x 480 pixels, e a taxa de quadros é 30 quadros/s. Portanto, a taxa de 1 bit/pixel corresponde a 10.368 Mbits/s (ou seja, aproximadamente 10 Mbits/s).

Na requantização, a relação entre as taxas dos canais de alta e baixa prioridade é controlada através da relação entre os passos de quantização Q1/Q2. Por outro lado, na separação espectral fixa e no particionamento de dados, esta relação entre taxas é controlada pelo valor de N, que representa o número de coeficientes (não nulos, no caso do particionamento de dados) que são enviados no canal de alta prioridade.

O primeiro passo, foi a escolha de valores para Q1/Q2 e para N de forma que a relação entre taxas dos dois canais fosse, aproximadamente, constante e igual a 1. Se os valores de Q1/Q2 e N forem constantes para todos os pontos da curva (ou seja, para várias taxas de bits/pixel), a relação de taxas entre os canais pode variar bastante [15]. Dessa forma, foram realizadas simulações para os três casos de escalonamento (para cada seqüência), considerando que os dois canais são utilizados na etapa de predição, a fim de encontrar valores de Q1/Q2 e N que mantivessem a relação entre taxas dos dois canais em torno de 1. Isto equivale a uma relação entre a taxa do canal de alta prioridade e a taxa total (soma das taxas dos dois canais) aproximadamente igual a 0.5, como apresentado nas Figs. 5.12, 5.13, e 5.14. Para estas simulações, foram utilizados valores de Q1/Q2 espaçados de 0.1. No caso da variável N, foram utilizados valores inteiros maiores que 0, já que N representa o número de coeficientes a serem alocados na camada de alta prioridade.

A Fig. 5.11 apresenta um exemplo de resultados destas simulações, onde são testados valores de Q1/Q2 para a seqüência "Mobile". De cada curva é escolhido o ponto cujo valor no eixo das ordenadas mais se aproxima de 0.5 (taxa AP/ taxa total = 0.5).

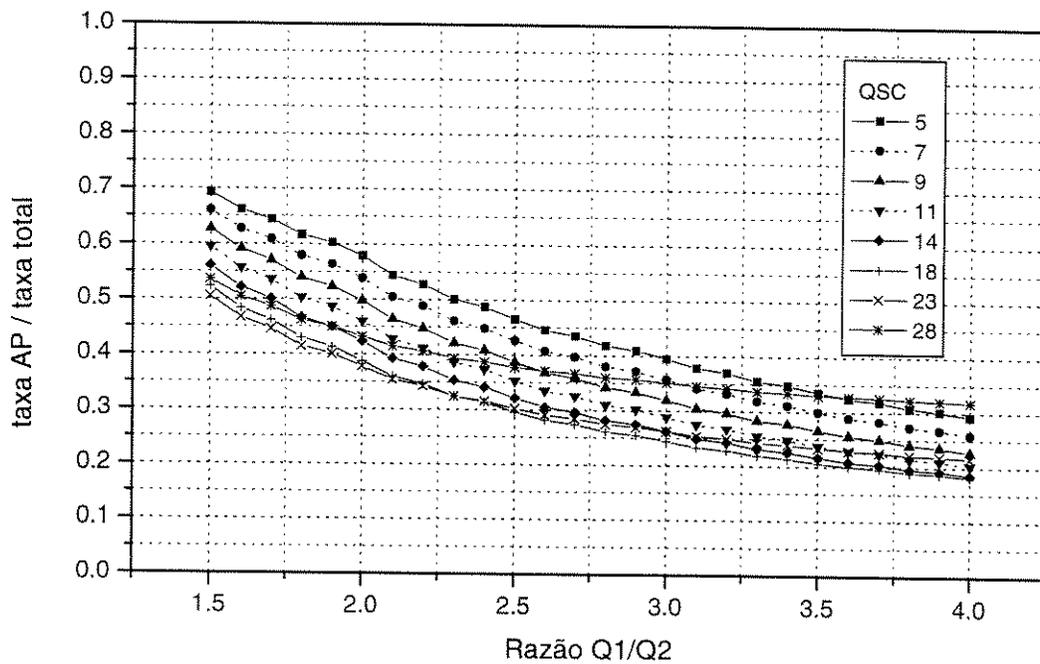


Fig. 5.11 – Relação $\frac{\text{taxa AP}}{\text{taxa total}}$ versus razão $\frac{Q1}{Q2}$, para a seqüência "Mobile"

De maneira similar ao que foi apresentado no exemplo da Fig. 5.11, foram obtidos valores de Q1/Q2 e N que mantivessem a relação entre taxas aproximadamente igual a 0.5. Estes valores estão dispostos na Tabela 5.1. Nestas simulações, considerou-se que a etapa de predição é baseada nos dados de alta e baixa prioridade. Entretanto, estes valores de Q1/Q2 e N também foram utilizados para os casos em que a etapa de predição é baseada somente nos dados do canal AP. Nestes casos, a relação entre taxa ficou, em média, um pouco abaixo de 0.5, que era o valor desejado.

Tabela 5.1 – Valores de Q1/Q2 e de N

QSC	"Mobile", PDC			"Kiel", PDC			"Tennis", PDC		
	Q1/Q2	N (SE)	N (DP)	Q1/Q2	N (SE)	N (DP)	Q1/Q2	N (SE)	N (DP)
5	2.3	17	12	2.5	17	12	2.0	15	8
7	2.2	16	10	2.3	15	10	1.8	13	6
9	2.0	15	9	2.1	14	9	1.8	10	5
11	1.9	13	7	2.0	13	7	1.9	8	4
14	1.7	12	6	1.8	11	5	2.0	6	3
18	1.6	9	4	1.6	8	3	2.1	4	2
23	1.5	6	2	1.6	5	2	2.3	2	1
28	1.7	2	1	2.0	1	1	5.9	1	1

Utilizando-se os valores dados na Tabela 5.1, foram traçados gráficos que relacionam a taxa de bits com a razão entre a taxa do canal AP e a taxa total. Estes gráficos são apresentados nas Figs. 5.12, 5.13, e 5.14, para as seqüências "Kiel", "Mobile" e "Tennis", respectivamente. Pode-se observar que para taxas muito baixas, quando o escalonamento é realizado através de particionamento de dados (DP) não é possível manter a relação entre taxas próxima de 0.5, pois nem o menor N possível, que é N=1, consegue atingir este objetivo. Por exemplo, para a seqüência "Tennis", a mínima relação entre taxas obtida, quando é utilizado o particionamento de dados e a taxa é de 0.108 bits/pixel, foi 0.67. Entretanto, de uma maneira geral, a relação entre taxas ficou próxima de 0.5, como era desejado.

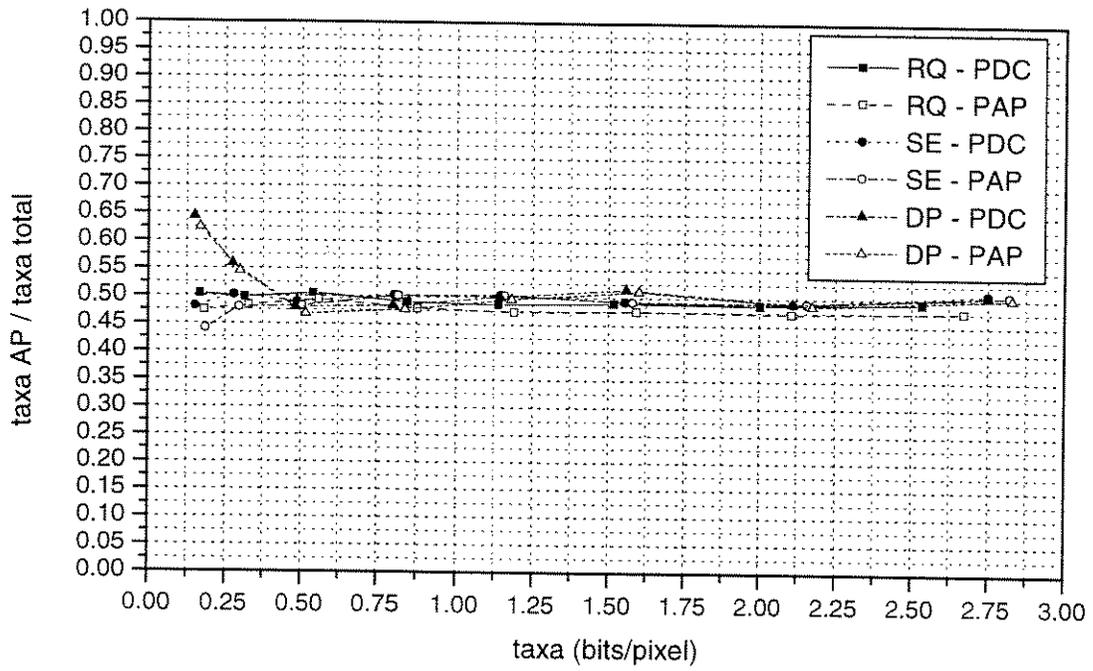


Fig. 5.12 - Razão entre taxas para a seqüência "Kiel"

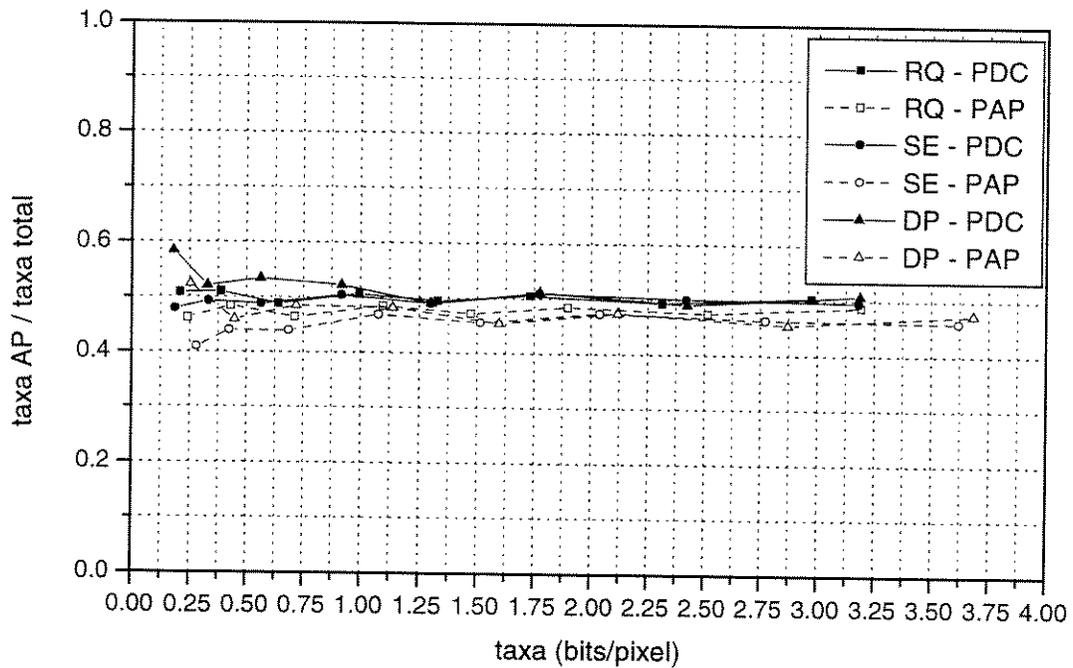


Fig. 5.13 - Razão entre taxas para a seqüência "Mobile"

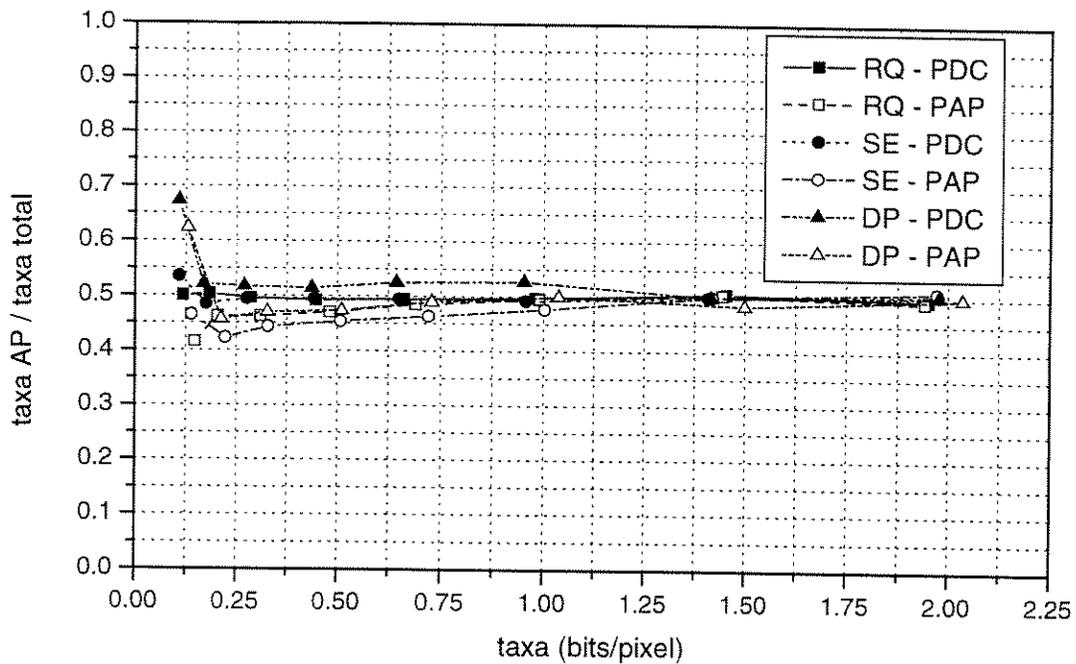


Fig. 5.14 - Razão entre taxas para a seqüência "Tennis"

5.4.1 Simulações considerando perda total do canal de baixa prioridade

Foram realizadas algumas simulações para comparar os três sistemas de codificação que utilizam separação espectral fixa (SE), particionamento de dados (DP) e requantização dos coeficientes(RQ). Quanto aos dados utilizados na etapa de predição, foram testados casos em que somente os dados de alta prioridade são utilizados para estimar o preditor (PAP), e casos em que tanto os dados de alta prioridade quanto os dados de baixa prioridade (PDC) são utilizados para estimar o preditor. Cada caso foi analisado em duas situações. Na primeira, os dois canais são recebidos sem erros (0% de perda). Na segunda, o canal de baixa prioridade é completamente perdido (BPP). Para as simulações, foram utilizados 30 quadros de cada uma das seqüências: "Kiel", "Mobile", e "Tennis". A ordem de tipos de quadros utilizada (GOP = 6) foi: "IBBPBBI...".

Os resultados são apresentados nos gráficos das Figs. 5.15, 5.16, e 5.17, correspondendo às seqüências "Kiel", "Mobile", e "Tennis", respectivamente.

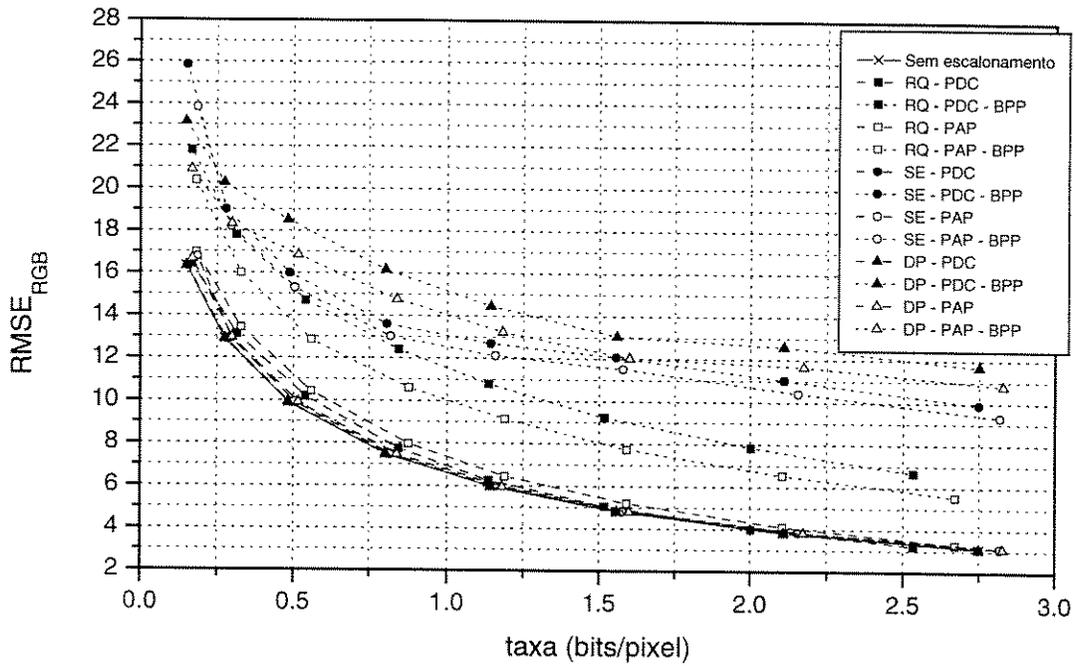


Fig. 5.15 - Curvas de distorção x taxa para casos de 0% de perda, e perda total do canal de baixa prioridade (BPP), para a seqüência "Kiel"

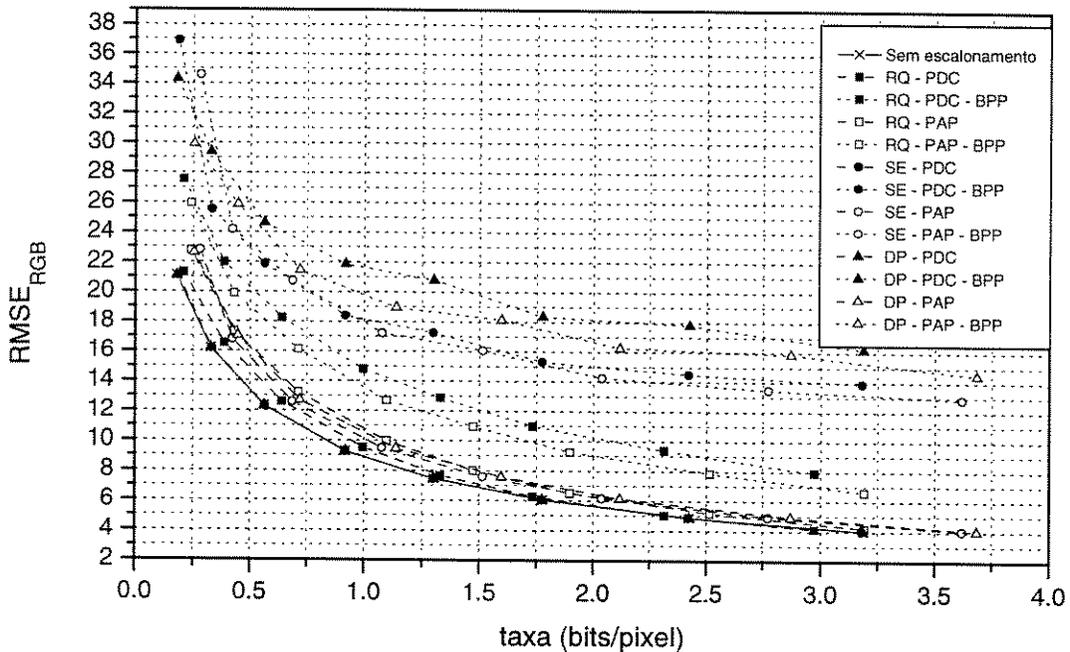


Fig. 5.16 - Curvas distorção x taxa para casos de 0% de perda, e perda total do canal de baixa prioridade (BPP), para a seqüência "Mobile"

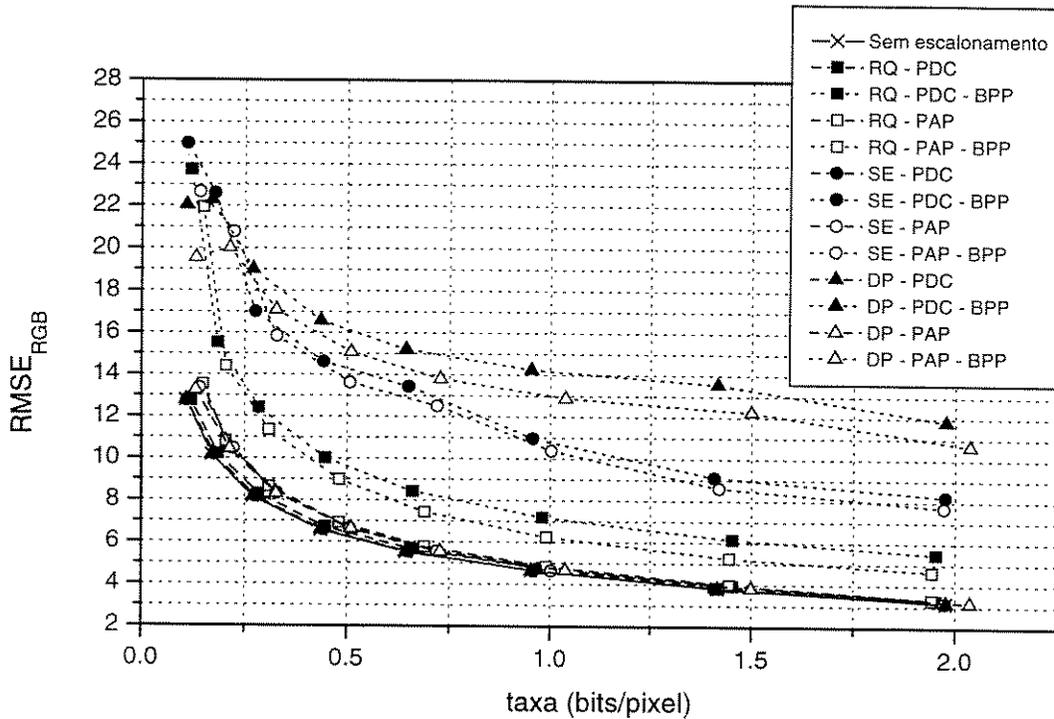


Fig. 5.17 - Curvas de distorção x taxa para casos de 0% de perda, e perda total do canal de baixa prioridade (BPP), para a seqüência “Tennis”

Através da observação destes gráficos fica clara a vantagem da requantização na situação em que ocorre perda total do canal de alta prioridade, especialmente quando só o canal de baixa prioridade é utilizado na etapa de predição (RQ-PAP-BPP). Entretanto também é importante considerar o overhead de taxa quando o sistema opera em condições normais, ou seja, quando o canal BP é bem recebido. Neste caso, a requantização apresenta um pior desempenho comparada ao DP ou a SE. Entretanto, como overhead introduzido pela RQ é pequeno, se a perda do canal de baixa prioridade acontece com frequência, globalmente a RQ apresenta um melhor desempenho. O overhead introduzido pelos esquemas de SE e DP é muito pequeno, da ordem de 1-2%. Isto torna a utilização destes dois sistemas viável quando o canal de baixa prioridade é sujeito a perdas mas estas situações de perdas não ocorrem com muita frequência. De modo geral, a SE apresentou um desempenho melhor que o DP. Esta observação é interessante pois o Padrão MPEG-2 contempla o DP, mas não a separação espectral fixa, da maneira como ela foi implementada.

5.4.2 Simulações considerando perda de células do canal de baixa prioridade

Nesta segunda fase de simulações, o canal de baixa prioridade é sujeito a perdas parciais de informação. Considera-se que a transmissão dos dois canais é feita em uma rede ATM onde os dados de cada canal são enviados em células de diferentes níveis de prioridades. Desta forma, estas perdas parciais de informação do canal de baixa prioridade correspondem a perdas de células. Estas perdas são simuladas de acordo com o modelo baseado em tráfego agregado, descrito na Seção 5.3.

Para análises comparativas foram também incluídos casos em que o sistema sem escalonamento é submetido a perdas. Neste caso, foi utilizado um método simples de correção de erros que consiste em substituir a região espacial do quadro atual relacionada com os dados perdidos, copiando-se a mesma região espacial do quadro de referência anterior, ou seja do último quadro de referência codificado. Quando ocorrem erros na recepção, estes se propagam, pelo menos, até o final do slice atual, pois o “slice” representa a menor unidade de sincronismo. Para simplificar as simulações considera-se que o sincronismo principal não é perdido, ou seja é, garantido que os principais cabeçalhos (e.g., cabeçalhos de seqüência, cabeçalhos de quadro, etc.) são recebidos sem erros pelo decodificador. Em situações reais, a perda destes cabeçalhos acarreta uma queda de desempenho bem maior que a observada nos resultados obtidos nestas simulações. Pode-se considerar que os resultados aqui obtidos são uma espécie de limitante inferior de desempenho de sistemas sem escalonamento e com perdas.

Nesta seção são apresentados os resultados obtidos para a seqüência “Tennis”. As taxas de perdas utilizadas foram: 1%, 5%, 10%, 20%, 30% e 40%, apresentadas nas Figs. 5.18, 5.19, 5.20, 5.21, 5.22, e 5.23, respectivamente. Simulações semelhantes foram realizadas para as outras duas seqüências (“Kiel” e “Mobile”), e estes resultados complementares são apresentados no Apêndice A.

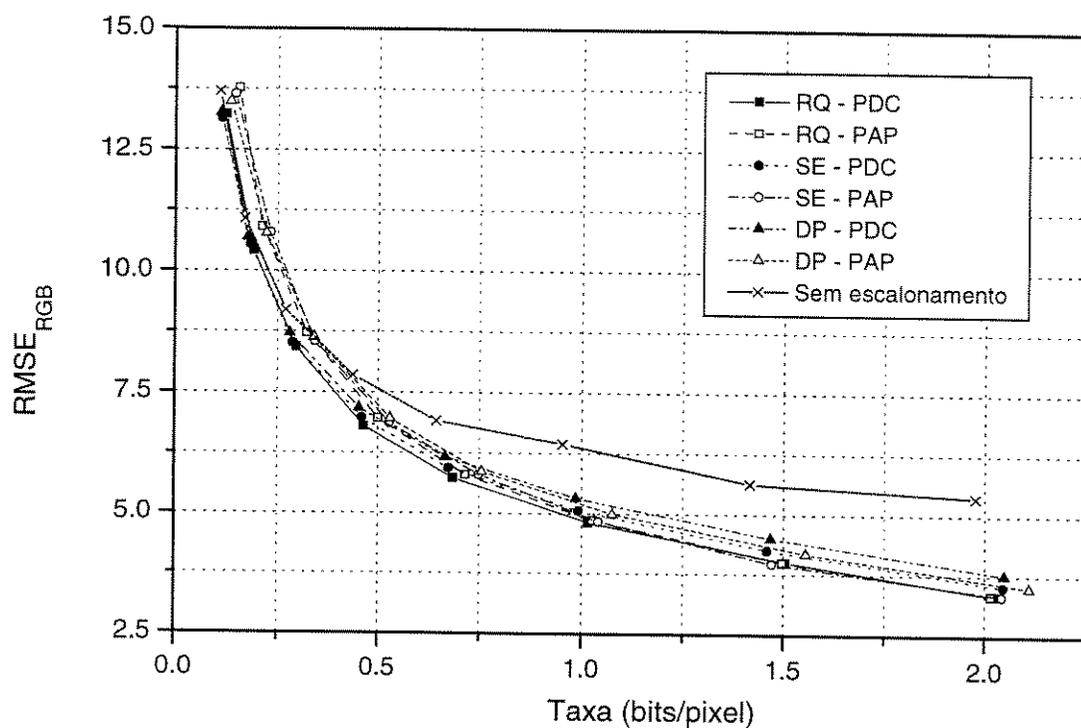


Fig. 5.18 - Curvas de distorção x taxa para a seqüência "Tennis", taxa de perda: 1%

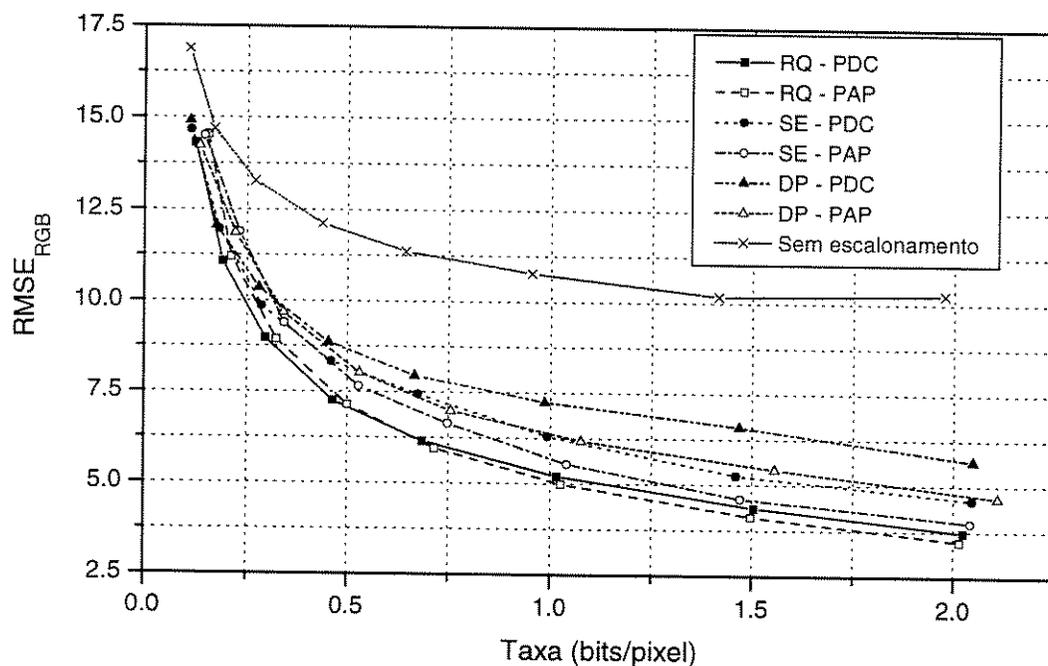


Fig. 5.19 - Curvas de distorção x taxa para a seqüência "Tennis", taxa de perda: 5%

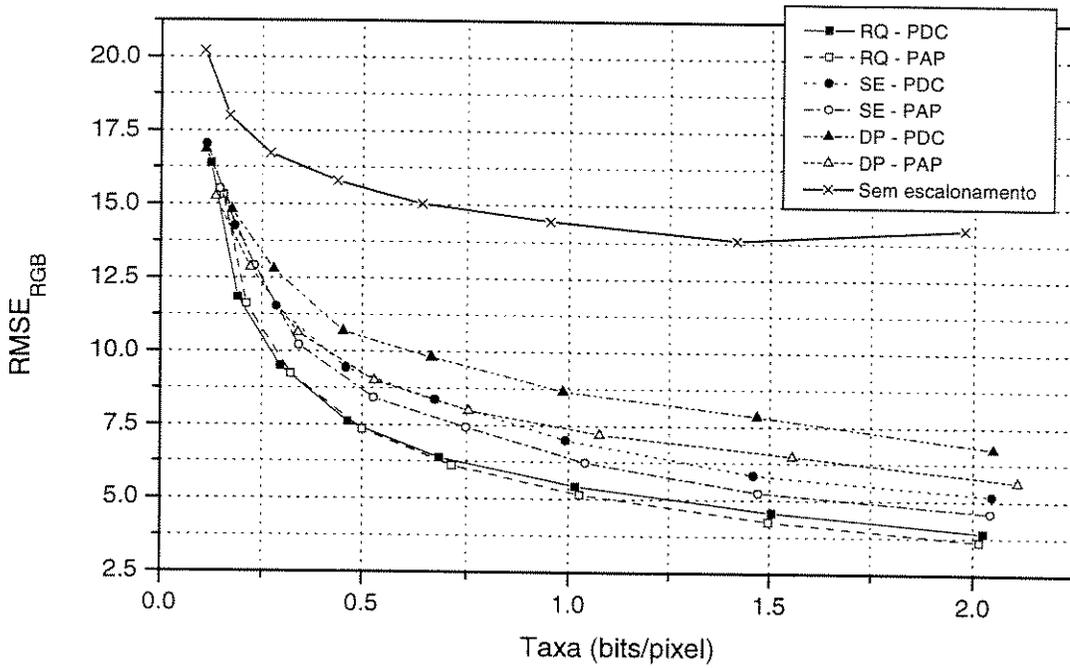


Fig. 5.20 - Curvas de distorção x taxa para a seqüência “Tennis”, taxa de perda: 10%

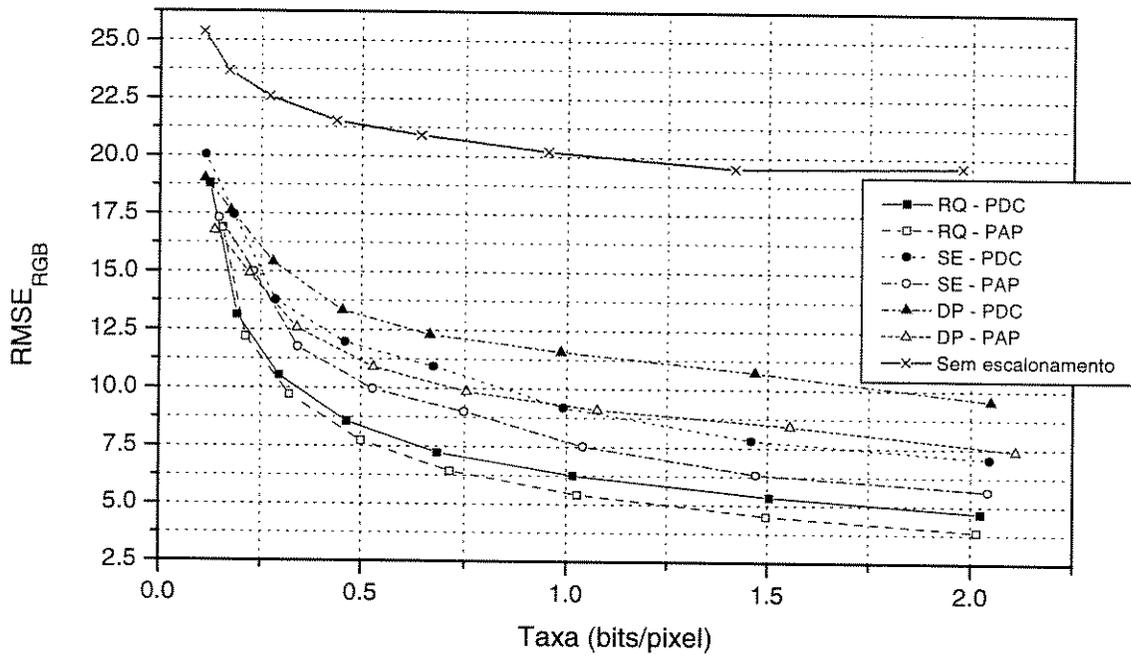


Fig. 5.21 - Curvas de distorção x taxa para a seqüência “Tennis”, taxa de perda: 20%

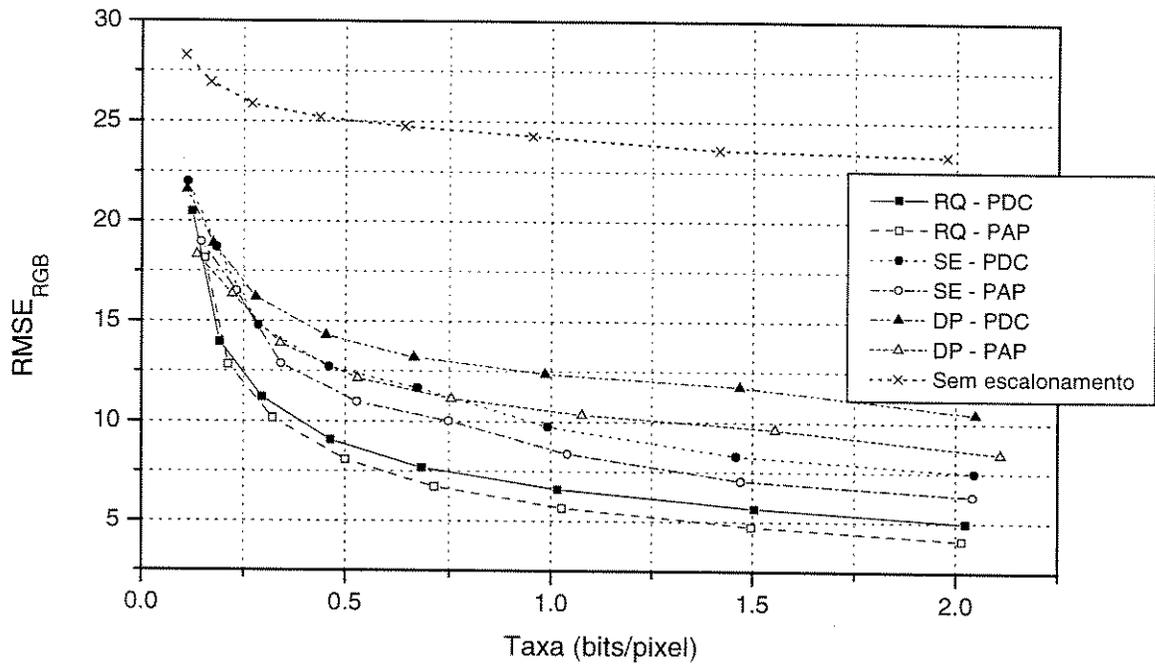


Fig. 5.22 - Curvas de distorção x taxa para a seqüência “Tennis”, taxa de perda: 30%

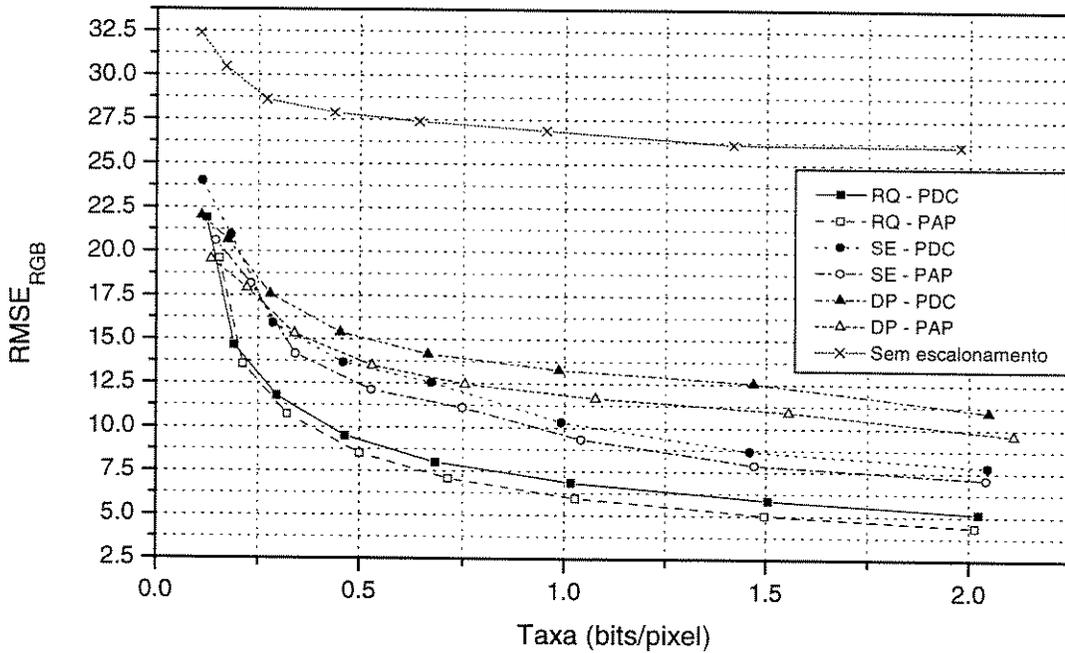


Fig. 5.23 - Curvas de distorção x taxa para a seqüência “Tennis”, taxa de perda: 40%

Analisando-se os gráficos obtidos fica clara a vantagem de se utilizar alguma técnica de escalonamento pois, nestes gráficos, a única situação em que o sistema sem escalonamento apresenta melhor desempenho é para taxas abaixo de 0.5 bits/pixel com taxa de perda de 1%. Estes gráficos reafirmam a vantagem da requantização sobre as demais formas de escalonamento, especialmente para taxas de perdas de células maiores que 5%. Para taxas de perdas de células menores que 5%, a SE pode se tornar o sistema de melhor desempenho global, já que o overhead de dados transmitidos (considerando-se situação em que não há perdas) é bem menor que o overhead introduzido pela RQ na mesma situação.

Capítulo 6

Conclusões e Trabalhos Futuros

Neste trabalho foram investigados métodos de escalonamento de seqüências de vídeo digital codificado para transmissão, a taxas variáveis (VBR), através de redes RDSI-FL baseadas no protocolo ATM.

Inicialmente foram apresentadas técnicas de compressão de imagens e vídeo que são utilizadas nos padrões de compressão de vídeo digital. Em seguida foi apresentada uma visão geral dos padrões de compressão de imagens e vídeo digital, em especial o Padrão MPEG-2. No Capítulo 4 foram apresentados aspectos a serem considerados quando se transmite vídeo em redes digitais de serviços integrados de faixa larga (RDSI-FL) baseadas no protocolo ATM. Em particular, soluções para reduzir os efeitos das perdas de células quando o Padrão MPEG-2 é utilizado na codificação de vídeo foram apresentadas.

Como foi visto, uma técnica que garante boa robustez para altas taxas de perdas (ainda que estas perdas ocorram em um período curto de tempo) é a codificação em camadas, ou seja o escalonamento da seqüência de vídeo codificada. Dois métodos de escalonamento propostos pelo Padrão MPEG-2 foram testados: escalonamento por requantização dos coeficientes (“SNR scalability”) e por particionamento de dados (“data partitioning”). Além destes, foi proposto e testado um terceiro método de escalonamento, através de separação espectral fixa dos coeficientes. Os três métodos de escalonamento foram analisados em duas situações: utilizando-se os dois canais na etapa de estimação e compensação de movimento, como recomendado pelo Padrão MPEG-2, ou utilizando-se somente o canal de alta prioridade nesta etapa.

Entre os métodos testados, a requantização utilizando somente os dados de alta prioridade na etapa de predição apresentou melhor desempenho, especialmente para taxas de perdas acima de 5%, apesar de apresentar maior “overhead” em situações sem perdas. É interessante notar que o GOP utilizado nas simulações foi de tamanho 6 ($N=6$, $M=3$). Caso fossem utilizados GOP's de tamanhos maiores (e.g., $N=12$, $M=3$), os resultados seriam ainda mais favoráveis à utilização de somente dados de alta prioridade na etapa de predição. Isto

ocorre pois o erro de um quadro de referência se propaga até que outro quadro de referência substitua o quadro de referência com erros. Tal resultado é esperado pois, neste caso, a utilização exclusiva do canal de alta prioridade na predição garante o casamento dos preditores, caso ocorra a perda do canal de baixa prioridade. Além disso, a requantização, da maneira como foi utilizada, produz uma equalização do espectro do ruído de quantização, ponderado pela sensibilidade do olho humano [9], como apresentado na Seção 2.3.5. Esta equalização é uma indicação de que, se fossem considerados aspectos subjetivos para determinação da qualidade de reprodução (ao invés do $RMSE_{RGB}$), os resultados provavelmente seriam também favoráveis à requantização.

Observa-se também que o Padrão MPEG-2 não prevê a utilização exclusiva de dados de alta prioridade na predição ou na interpolação de quadros (nos modos de separação espectral e de requantização) como, em vista dos presentes resultados, seria prudente.

Outra observação é que, de maneira geral, a separação espectral fixa proposta apresentou um desempenho superior àquele obtido pelo particionamento de dados, que é recomendado pelo Padrão MPEG-2. Tanto a separação espectral fixa quanto o particionamento de dados são formas de se realizar escalonamento por separação espectral. Para taxas de perdas abaixo de 5%, estas técnicas podem apresentar desempenho melhor que a requantização, devido ao maior "overhead" desta última técnica.

Uma possibilidade para trabalhos futuros, na direção do horizonte aberto por este trabalho, é analisar o desempenho de sistemas de codificação, transmissão, e decodificação de vídeo digital (com ou sem escalonamento da seqüência codificada) que utilizem outras técnicas para a diminuição da sensibilidade a erros de transmissão. Estas técnicas podem ser, por exemplo, aquelas apresentadas na Seção 4.3, e.g., envio de vetores de deslocamento para quadros I, tamanho de "slices" adaptativos, etc. Sistemas que considerem outros aspectos envolvidos na transmissão de vídeo em redes ATM, tais como atrasos de transmissão e "jitter", também devem ser investigados. É importante observar que uma questão em aberto é o estabelecimento de uma camada de adaptação da rede ATM (AAL - "ATM Adaptation Layer") para transmissão de vídeo digital a taxas variáveis (VBR). Esta questão tem sido objeto de estudo do ITU [24] e do "ATM Forum" [23].

Como contribuição deste trabalho pode-se citar o desenvolvimento do pacote de programas de simulação do sistema de codificação e transmissão de vídeo digital, que realiza

escalonamento da seqüência codificada. Outras contribuições são os resultados obtidos das simulações realizadas, que apontam o escalonamento como técnica para aumentar a robustez de sistemas de transmissão de vídeo digital em ambiente de perdas de sinal. Além disso foram obtidos resultados comparativos entre as técnicas de escalonamento em estudo, em diferentes situações de perdas de sinal.

Apêndice A

Resultados Complementares

Neste apêndice, são apresentados resultados complementares, na forma de curvas de distorção x taxa, de simulações realizadas utilizando-se as seqüências “Kiel” e “Mobile”. A inclusão destes resultados na forma de apêndice tem por objetivo reforçar as conclusões delineadas no Capítulo 5 (a partir da observação dos resultados obtidos nas simulações utilizando a seqüência “Tennis”) sem sobrecarregá-lo com gráficos, de forma a facilitar a compreensão do leitor.

As curvas das Figs. A.1, A.2, A.3, A.4, A.5 e A.6 apresentam o compromisso entre distorção e taxa de bits para a seqüência “Kiel” com taxas de perdas crescentes, variando de 1% a 40%. As curvas das Figs. A.7, A.8, A.9, A.10, A.11 e A.12 apresentam resultados, também relacionando distorção e taxas de bits, obtidos para a seqüência “Mobile”, com taxas de perdas entre 1% a 40%.

Legenda:

SE - Separação espectral fixa

DP - Particionamento de dados (“data partitionig”)

RQ - Requantização dos coeficientes

PAP - Predição baseada nos dados de alta prioridade

PDC - Predição baseada nos dados de alta e baixa prioridade

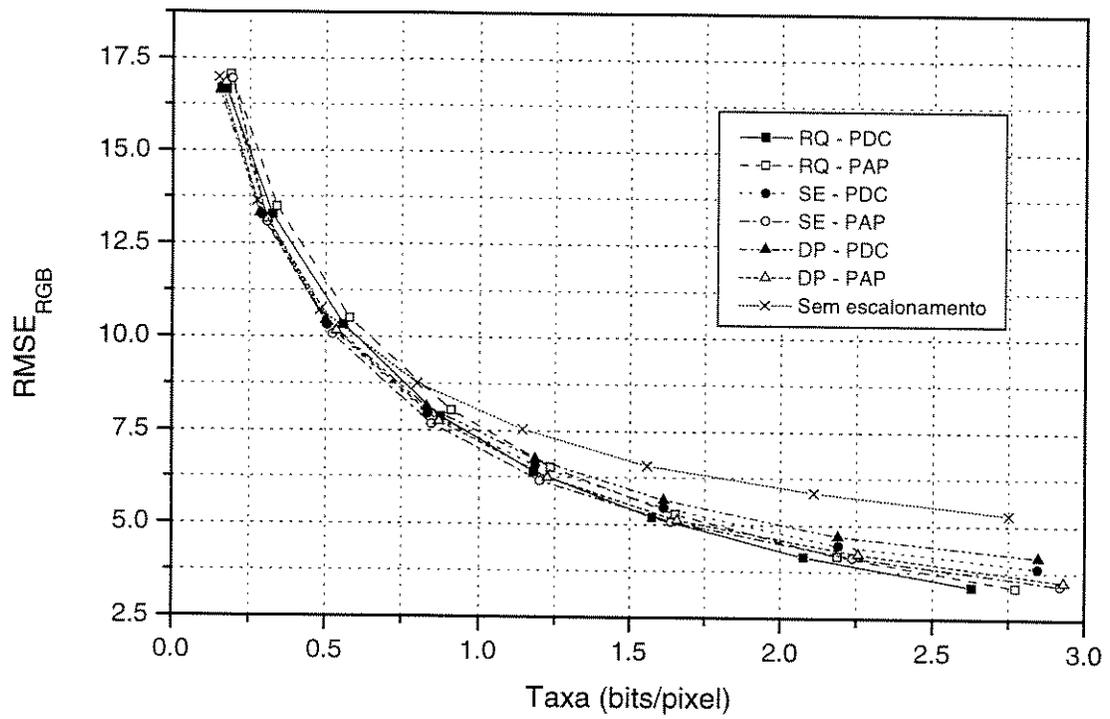


Fig. A.1 - Curvas de distorção x taxa para seqüência “Kiel”, taxa de perda: 1%

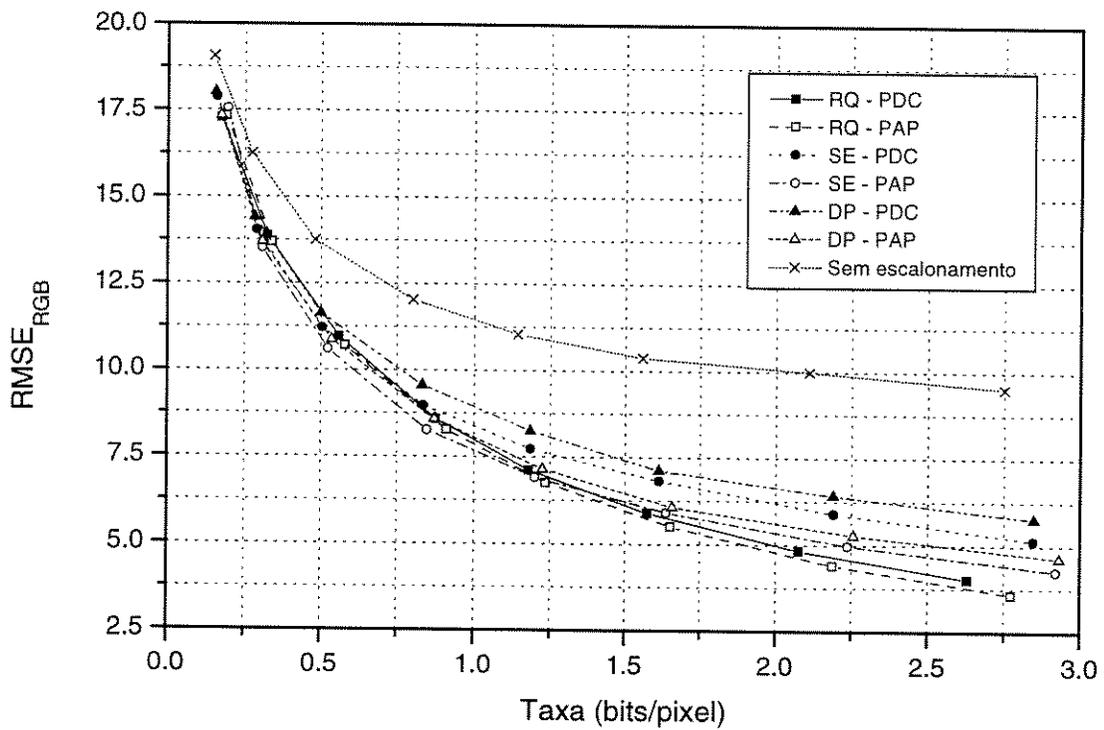


Fig. A.2 - Curvas de distorção x taxa para seqüência “Kiel”, taxa de perda: 5%

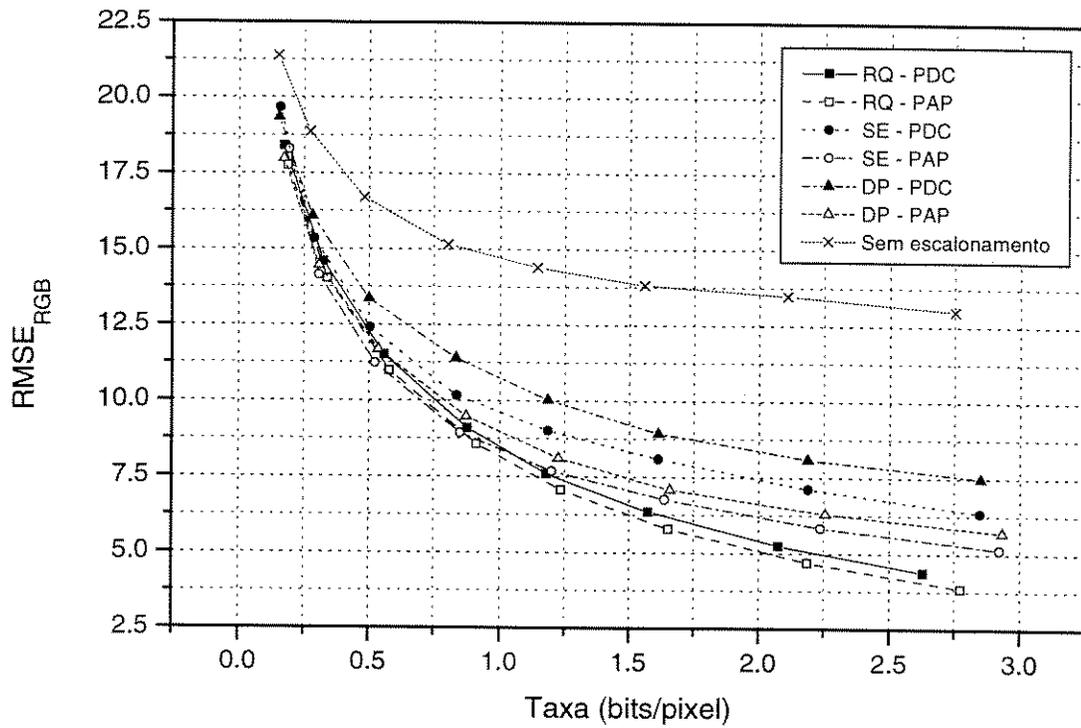


Fig. A.3 - Curvas de distorção x taxa para seqüência “Kiel”, taxa de perda: 10%

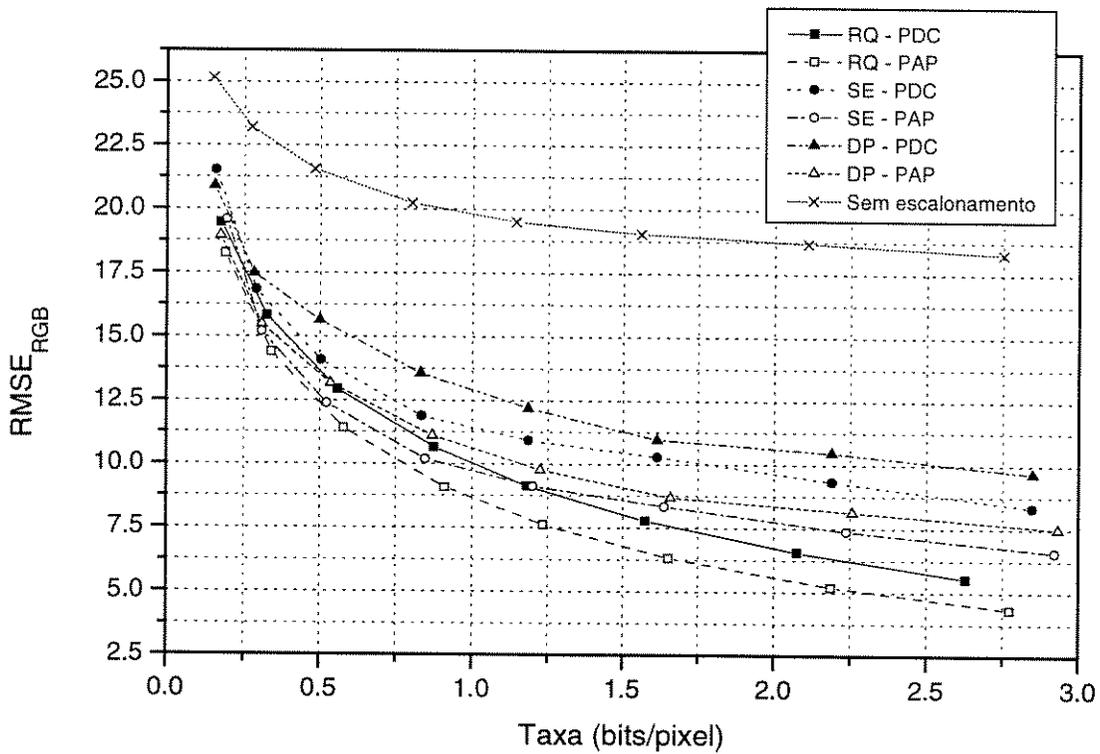


Fig. A.4 - Curvas de distorção x taxa para seqüência “Kiel”, taxa de perda: 20%

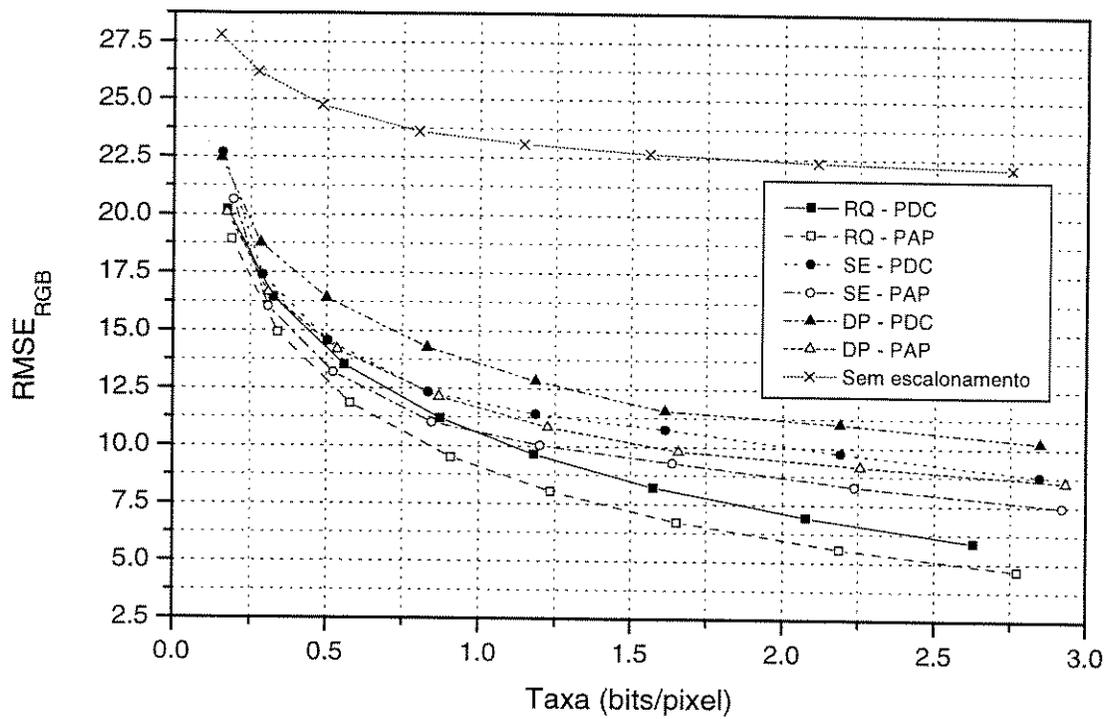


Fig. A.5 - Curvas de distorção x taxa para seqüência “Kiel”, taxa de perda: 30%

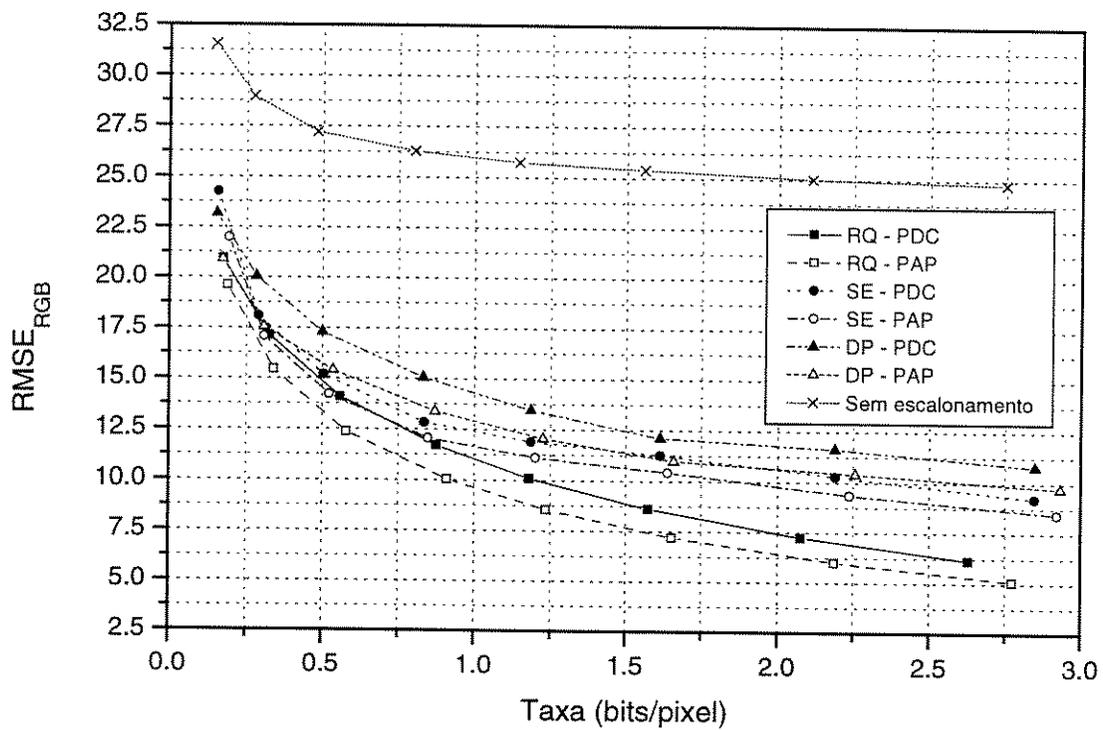


Fig. A.6 - Curvas de distorção x taxa para seqüência “Kiel”, taxa de perda: 40%

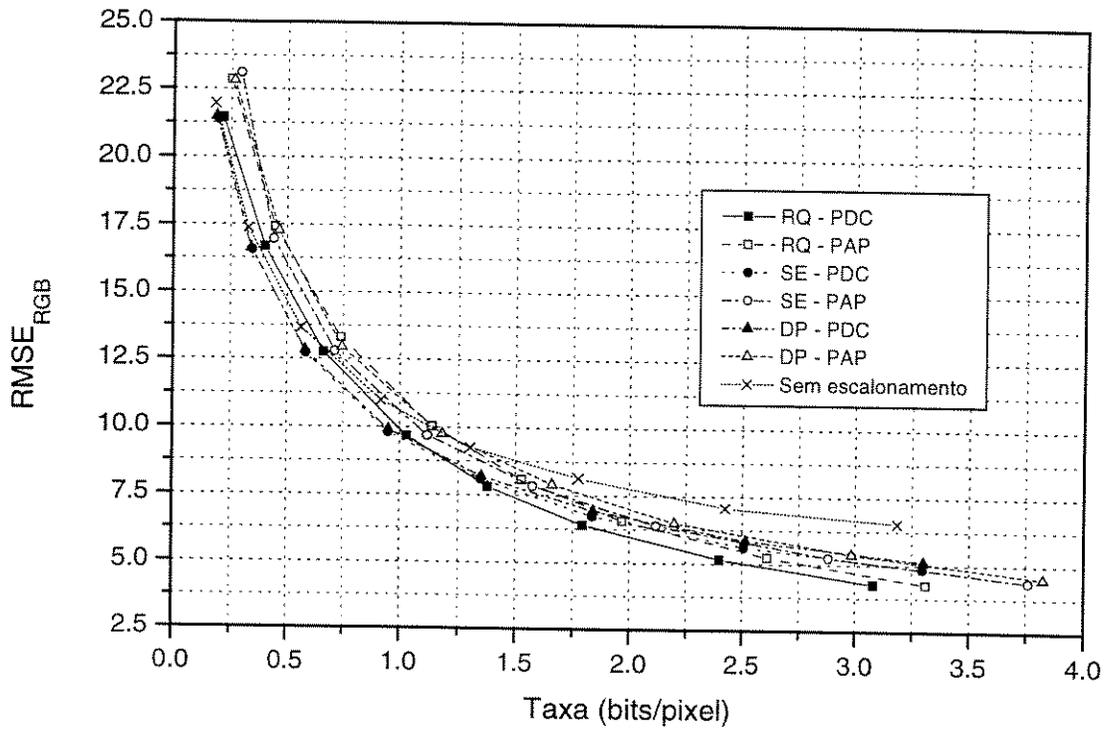


Fig. A.7 - Curvas de distorção x taxa para seqüência “Mobile”, taxa de perda: 1%

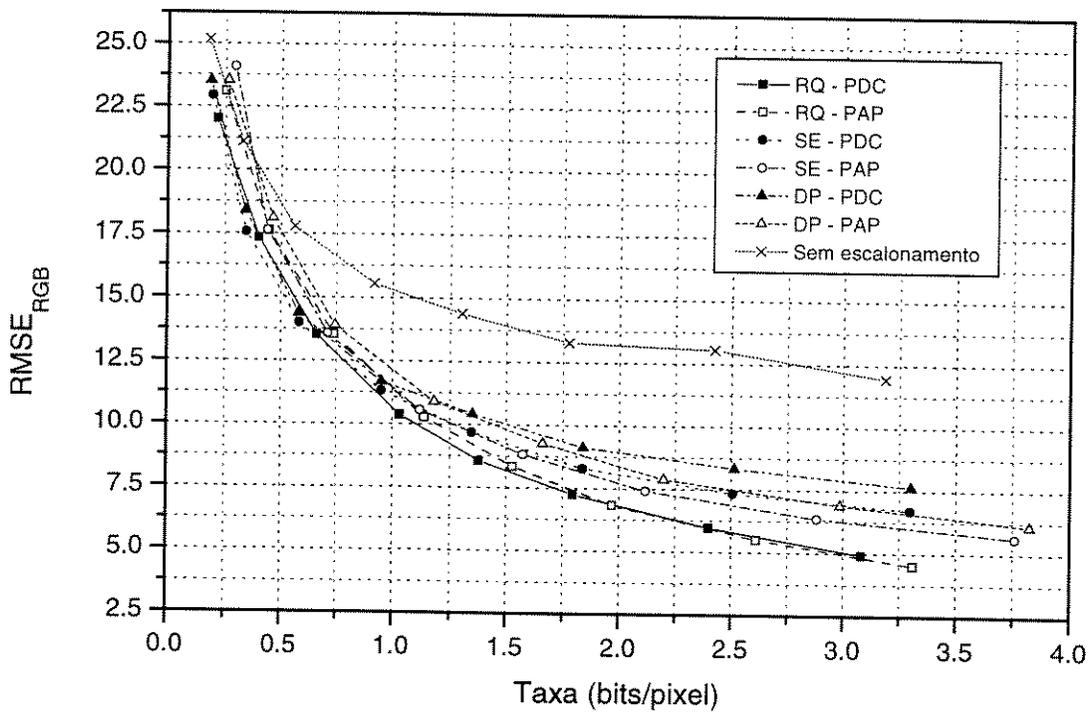


Fig. A.8 - Curvas de distorção x taxa para seqüência “Mobile”, taxa de perda: 5%

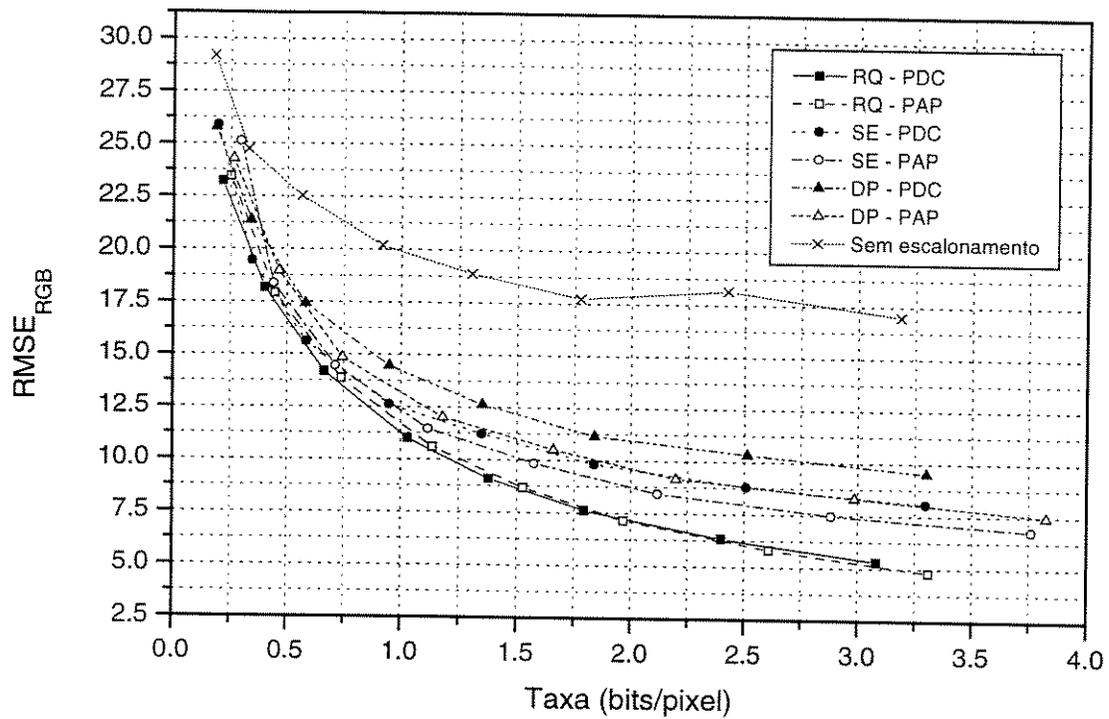


Fig. A.9 - Curvas de distorção x taxa para seqüência “Mobile”, taxa de perda: 10%

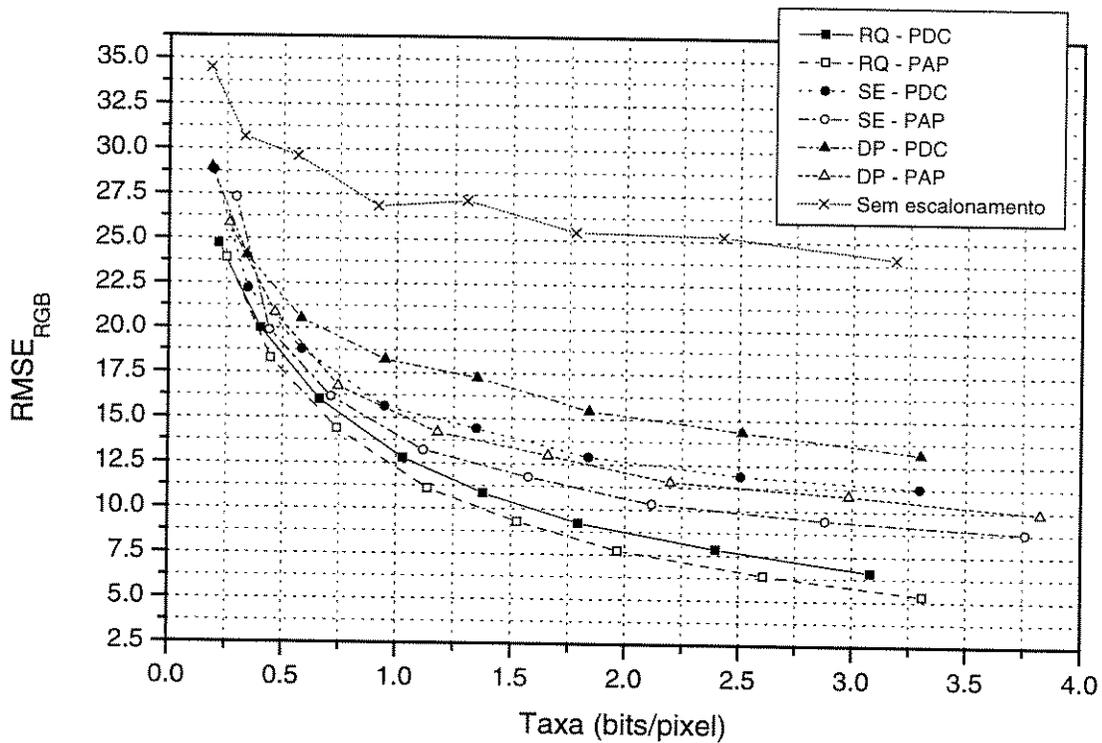


Fig. A.10 - Curvas de distorção x taxa para seqüência “Mobile”, taxa de perda: 20%

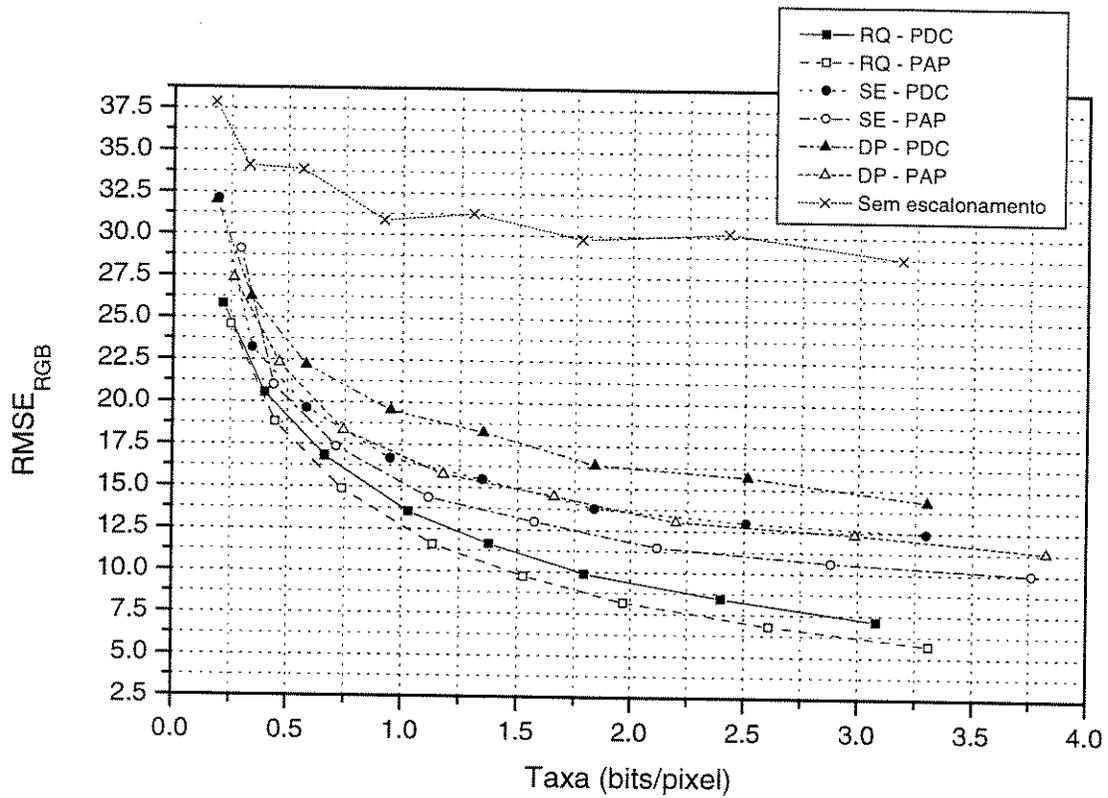


Fig. A.11 - Curvas de distorção x taxa para seqüência “Mobile”, taxa de perda: 30%

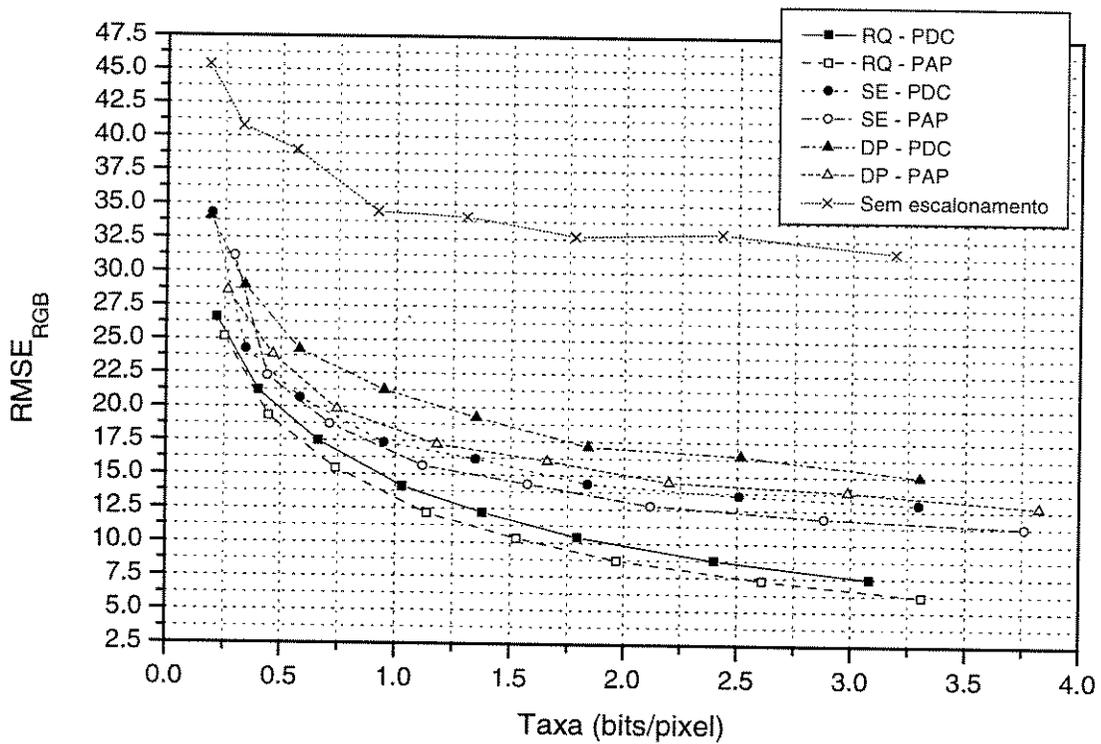


Fig. A.12 - Curvas de distorção x taxa para seqüência “Mobile”, taxa de perda: 40%

Apêndice B

Neste apêndice é apresentada a Tabela B.1, que é recomendada pelo Padrão MPEG-2 para codificação dos coeficientes da DCT (exceto os coeficientes DC de macroblocos intracodificados) quando faz-se a variável. “intra_vlc_format” = ‘0’. Também é apresentada a Tabela B.2 (Tabela B.16 do Padrão MPEG-2 [2]), que é utilizada juntamente com o código de escape para representar pares *corrida de zeros / nível do coeficiente* que têm baixa frequência de ocorrência. Neste caso, utiliza-se o código de escape seguido do número de coeficientes nulos que precedem o coeficiente (corrida de zeros), e do valor deste coeficiente.

Tabela B.1 - Tabela de Huffman para codificação dos coeficientes da DCT
(Tabela B.14, extraída do Padrão MPEG-2 [2])

Palavra-código (Nota 1)	Corrida de zeros	Nível do coeficiente
10 (Nota 2)	Fim de bloco	
1 s (Nota 3)	0	1
11 s (Nota 4)	0	1
011 s	1	1
0100 s	0	2
0101 s	2	1
0010 1 s	0	3
0011 1 s	3	1
0011 0 s	4	1
0001 10 s	1	2
0001 11 s	5	1
0001 01 s	6	1
0001 00 s	7	1
0000 110 s	0	4
0000 100 s	2	2
0000 111 s	8	1
0000 101 s	9	1
0000 01	Escape	
0010 0110 s	0	5
0010 0001 s	0	6
0010 0101 s	1	3
0010 0100 s	3	2
0010 0111 s	10	1
0010 0011 s	11	1
0010 0010 s	12	1

Tabela B.1 - Tabela de Huffman para codificação dos coeficientes da DCT
(Continuação)

Palavra-código	Corrida de zeros	Nível do coeficiente
0010 0000 s	13	1
0000 0010 10 s	0	7
0000 0011 00 s	1	4
0000 0010 11 s	2	3
0000 0011 11 s	4	2
0000 0010 01 s	5	2
0000 0011 10 s	14	1
0000 0011 01 s	15	1
0000 0010 00 s	16	1
0000 0001 1101 s	0	8
0000 0001 1000 s	0	9
0000 0001 0011 s	0	10
0000 0001 0000 s	0	11
0000 0001 1011 s	1	5
0000 0001 0100 s	2	4
0000 0001 1100 s	3	3
0000 0001 0010 s	4	3
0000 0001 1110 s	6	2
0000 0001 0101 s	7	2
0000 0001 0001 s	8	2
0000 0001 1111 s	17	1
0000 0001 1010 s	18	1
0000 0001 1001 s	19	1
0000 0001 0111 s	20	1
0000 0001 0110 s	21	1
0000 0000 1101 0 s	0	12
0000 0000 1100 1 s	0	13
0000 0000 1100 0 s	0	14
0000 0000 1011 1 s	0	15
0000 0000 1011 0 s	1	6
0000 0000 1010 1 s	1	7
0000 0000 1010 0 s	2	5
0000 0000 1001 1 s	3	4
0000 0000 1001 0 s	5	3
0000 0000 1000 1 s	9	2
0000 0000 1000 0 s	10	2
0000 0000 1111 1 s	22	1

**Tabela B.1 - Tabela de Huffman para codificação dos coeficientes da DCT
(Continuação)**

Palavra-código	Corrida de zeros	Nível do coeficiente
0000 0000 1111 0 s	23	1
0000 0000 1110 1 s	24	1
0000 0000 1110 0 s	25	1
0000 0000 1101 1 s	26	1
0000 0000 0111 11 s	0	16
0000 0000 0111 10 s	0	17
0000 0000 0111 01 s	0	18
0000 0000 0111 00 s	0	19
0000 0000 0110 11 s	0	20
0000 0000 0110 10 s	0	21
0000 0000 0110 01 s	0	22
0000 0000 0110 00 s	0	23
0000 0000 0101 11 s	0	24
0000 0000 0101 10 s	0	25
0000 0000 0101 01 s	0	26
0000 0000 0101 00 s	0	27
0000 0000 0100 11 s	0	28
0000 0000 0100 10 s	0	29
0000 0000 0100 01 s	0	30
0000 0000 0100 00 s	0	31
0000 0000 0011 000 s	0	32
0000 0000 0010 111 s	0	33
0000 0000 0010 110 s	0	34
0000 0000 0010 101 s	0	35
0000 0000 0010 100 s	0	36
0000 0000 0010 011 s	0	37
0000 0000 0010 010 s	0	38
0000 0000 0010 001 s	0	39
0000 0000 0010 000 s	0	40
0000 0000 0011 111 s	1	8
0000 0000 0011 110 s	1	9
0000 0000 0011 101 s	1	10
0000 0000 0011 100 s	1	11
0000 0000 0011 011 s	1	12
0000 0000 0011 010 s	1	13
0000 0000 0011 001 s	1	14

**Tabela B.1 - Tabela de Huffman para codificação dos coeficientes da DCT
(Continuação)**

Palavra-código	Corrida de zeros	Nível do coeficiente
0000 0000 0001 0011 s	1	15
0000 0000 0001 0010 s	1	16
0000 0000 0001 0001 s	1	17
0000 0000 0001 0000 s	1	18
0000 0000 0001 0100 s	6	3
0000 0000 0001 1010 s	11	2
0000 0000 0001 1001 s	12	2
0000 0000 0001 1000 s	13	2
0000 0000 0001 0111 s	14	2
0000 0000 0001 0110 s	15	2
0000 0000 0001 0101 s	16	2
0000 0000 0001 1111 s	27	1
0000 0000 0001 1110 s	28	1
0000 0000 0001 1101 s	29	1
0000 0000 0001 1100 s	30	1
0000 0000 0001 1011 s	31	1
<p>NOTAS</p> <p>1- O último bit, 's', representa o sinal do coeficiente: '0' para coeficiente positivo, e '1' para coeficiente negativo</p> <p>2- O fim de bloco não pode ser a única palavra-código de um bloco.</p> <p>3- Este código é utilizado para o primeiro coeficientes (DC) de um bloco.</p> <p>4- Este código é utilizado para todos os outros coeficientes</p>		

Tabela B.2 - Tabela para codificação do par "corrida de zeros / nível do coeficiente", quando utiliza-se o código de Escape (Tabela B.16, extraída do Padrão MPEG-2 [2])

Palavra-código de comprimento fixo	Corrida de zeros
0000 00	0
0000 01	1
0000 10	2
...	...
...	...
...	...
...	...
1111 11	63

Palavra-código de comprimento fixo	Valor do coeficiente
1000 0000 0001	-2047
1000 0000 0010	-2046
...	...
1111 1111 1111	-1
0000 0000 0000	Uso proibido
0000 0000 0001	+1
...	...
0111 1111 1111	+2047

Referências Bibliográficas

- [1] D. Wilson e M Ghanbari, "Optimisation of two-layer SNR scalability for MPEG-2 video," Proc. Int. Conf. Acoust. Speech Signal Processing - ICASSP, pp. 2637-2640, Munich, Alemanha, 1997.
- [2] ITU-T Recommendation H.262, "Information technology - Generic coding of moving pictures and associated audio information: video," agosto 1995.
- [3] ITU-R Recommendation 601-3, "Encoding parameters of digital television for studios," 1992.
- [4] B. G. Haskell, A. Puri, e A. N. Netrevali, Digital Video: An Introduction to MPEG-2. New York: Chapman & Hall, 1997.
- [5] H. Waldman, D. S. Arantes e Max H. M. Costa, Conectividade e Qualidades-de-Serviço em Redes de Faixa Larga, Pub. FEE 015/95, outubro 1995.
- [6] D. E. McDysan e D. L. Spohn, ATM Theory and Application. McGraw-Hill, 1995.
- [7] M. de Prycker, Asynchronous Transfer Mode Solution for Broadband ISDN. Prentice Hall, 1995.
- [8] M. Rabbani, e P. W. Jones, Digital Image Compression Techniques. Bellingham: SPIE Press, 1991.
- [9] N. S. Jayant, e P. Noll, Digital Coding of Waveforms - Principles and Applications to Speech and Video. Englewood Cliffs: Prentice-Hall, 1984.
- [10] F. Dufaux, e F. Moscheni, "Motion estimation techniques for digital TV: a review and a new contribution," Proc. of IEEE, vol. 83, no. 6, pp. 858-876, junho 1995.
- [11] J. L. Mitchel et al., MPEG Video Compression Standard. New York: Chapman & Hall, 1997.
- [12] J. Lu, e M. L. Liou, "A simple and efficient search algorithm for block-matching motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 7, no. 2, abril 1997.
- [13] L. Chiariglione, "MPEG and multimedia communications," IEEE Trans. Circuits Syst. Video Technol., vol. 7, no. 1, fevereiro 1997.
- [14] <ftp://ftp-info3.informatik.uni-wuerzburg.de/pub/MPEG>

- [15] L. R. Mendes, e Max H. M. Costa, "Transmissão de vídeo utilizando o padrão MPEG-2 em duas camadas," XV Simpósio Brasileiro de Telecomunicações, pp. 585-589, setembro 1997, Recife PE.
- [16] M. Ghanbari, e V. Seferidis, "Efficient H.261-based two-layer video codecs for ATM networks," IEEE Trans. Circuits Syst. Video Technol., vol. 5, no. 2, abril 1995.
- [17] M. Khademi, e F. J. Paoloni, "Performance and statistical analysis of two-layer MPEG coding algorithms for ATM Networks", IEE Proc. Commun., vol. 143, no. 3, pp. 155-161, junho 1996.
- [18] P. Pancha, e M. El Zarki, "MPEG coding for variable bit rate video transmission," IEEE Commun. Mag., pp. 54-66, maio 1994.
- [19] W. Verbiest, L. Pinnoo, e B. Voeten, "The impact of the ATM concept on video coding," IEEE Journal on SAC, vol. 6, no. 9, pp. 1623-1632, dezembro 1988.
- [20] G. Karlsson, "Asynchronous transfer of video," IEEE Commun. Mag., pp. 118-126, agosto 1996.
- [21] V. Parthasarathy, e J. W. Modestino, "Design of a transport coding scheme for high-quality video over ATM networks," IEEE Trans. Circuits Syst. Video Technol., vol. 7, no. 2, pp. 358-376, abril 1997.
- [22] W. Ding, "Joint encoder and channel rate control of VBR video over ATM networks," IEEE Trans. Circuits Syst. Video Technol., vol. 7, no. 2, pp. 266-278, abril 1997.
- [23] <http://www.atmforum.com>
- [24] <http://www.itu.int>
- [25] <http://www.mpeg.org>
- [26] <ftp://tenet.berkeley.edu/pub/dbind/traces>
- [27] J. Beran, et al., "Long-range dependence in variable-bit-rate video traffic," IEEE Trans. Commun., vol. 43, nos. 2-4, pp. 1566-1579, fevereiro, março e abril 1995.
- [28] W. E. Leland, et al., "On the self-similar nature of ethernet traffic," IEEE/ACM Trans. on Networking, vol. 2, no. 1, pp. 1-15, fevereiro 1994.
- [29] I. Norros, et al., "The superposition of variable bit rate sources in an ATM multiplexer," IEEE J. Select. Areas Commun., vol. 9, no. 3, abril 1991.
- [30] ITU-T Recommendation H.222.0, "Information technology - Generic coding of moving pictures and associated audio information: systems," agosto 1995.

- [31] ITU-T Draft H.263, "Video coding for low bitrate communications," maio 1996.
- [32] ITU Recommendation H.261, "Video codec for audiovisual services at p x 64 kbits/s," julho 1990.
- [33] ISO/IEC 11172-2, "Coding of moving pictures and associated audio for digital storage media at up 1.5 Mbits/s - Part 2: video," 1993.
- [34] L Chiariglione, "The development of an integrated audiovisual coding standard: MPEG," Proc. of IEEE, vol. 83, no. 2, pp. 151-157, fevereiro 1995.
- [35] Apud J. R. Jain, e A. K. Jain, "Displacement measurement and its application in interframe image coding," IEEE Trans. Comm., COM-29(12):1799-808, dezembro 1981.
- [36] Apud T. Koga, et al., "Motion-compensated interframe coding for video conferencing," NTC 81 Proc., pp. G5.3.1-5, New Orleans, LA, dezembro 1981.
- [37] A. K. Jain, Fundamentals of Digital Image Processing. Englewood Cliffs: Prentice Hall, 1989.