

Transmissão Robusta de Vídeo em Pacotes

Autor : Renê Togni Del Pietro.

Orientador: Profº Max Henrique Machado Costa.

Este exemplar corresponde a redação final da tese
defendida por RENÊ TOGNI DEL PIETRO
e aprovada pela Comissão
Julgada em 27 / 08 / 1998.
Max Costa
Orientador

Dissertação submetida à Faculdade de
Engenharia Elétrica e de Computação
da Universidade Estadual de Campinas
para a obtenção do título de Mestre em
Engenharia Elétrica

Agosto 1998

2207002898

UNIDADE	80
N.º CHAMADA:	T/UNICAMP
	D385t
V.	Ex.
TOMBO BC	40539
PROC.	278/00
C	☐
D	☒
PREÇO	811,00
DATA	16/03/00
N.º CPD	

CM-00139021-B

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

D385t Del Pietro, Renê Togni
Transmissão robusta de vídeo em pacotes / Renê Togni Del Pietro.--Campinas, SP: [s.n.], 1998.

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

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

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

Aos meus pais

Margarida e Adolpho Del Pietro

"A melhor maneira de prever o futuro é inventá-lo"

(Alan Kay)

Agradecimentos

Nem só de madrugadas e trabalho árduo viveu este trabalho, mas de pessoas que, com seus ensinamentos, sua amizade e espírito de luta, animaram-me nos momentos difíceis. Em especial aos amigos

Profº Max Henrique Machado Costa.

Profº Dalton Soares de Arantes.

Luis Rômulo Mendes.

Fabbryccio Akkazzha Chaves M. Cardoso.

Elenir Belezso.

e aos que indiretamente contribuíram para a realização deste trabalho, os meus mais sinceros agradecimentos.

Agradecimentos são estendidos, em especial, ao CPqD TELEBRÁS e a CAPES, órgãos que me apoiaram e incentivam esta pesquisa, num gesto nobre de valorizar o desenvolvimento científico nacional.

ÍNDICE

RESUMO.....	VI
ABSTRACT.....	VII
LISTA DE FIGURAS.....	VIII
LISTA DE TABELAS.....	XI
CAPÍTULO 1 - Introdução	1
CAPÍTULO 2 - Sistema de Comunicação para Sinais de Vídeo.....	5
2.1 - O Sistema Digital de Comunicação	6
2.2 - O Codificador e Decodificador de Fonte	8
2.2.1 - Codificação Intraquadro	9
2.2.2 - Codificação Interquadro	21
2.3 - Tratamento com Imagens Coloridas	27
2.4 - O Codificador e Decodificador de Canal	29
2.5 - Escalonamento do Sinal de Vídeo	33
2.5.1 - Separação Espectral	34
2.5.2 – Requantização	36
CAPÍTULO 3 - O Ambiente de Simulação	38
3.1 - Módulo 1 : Codificador e Decodificador para Sequências de Vídeo	39
3.2 - Módulo 2 : Compactação e Descompactação dos Dados em Células ATM.....	48
3.3 - Módulo 3 : Modelagem de Perdas de Células no Canal de Transmissão.....	51
CAPÍTULO 4 - Análise dos Resultados	54
4.1 - Análise do Codificador e Decodificador.....	54
4.2 - Análise das Perdas de Células sem Escalonamento	59
4.3 - Análise com Perdas de Células e com Escalonamento	67
4.3.1 – Requantização	68

4.3.2 - Separação Espectral	74
4.4 - Comparação entre as Codificações com e sem Escalonamento	79
4.5 - Comparação entre as Codificações com Escalonamento	83

CAPÍTULO 5 - Conclusões	92
--------------------------------------	-----------

APÊNDICE A

TABELA A.1 - Matrizes de quantização utilizadas pelo codificador de fonte	97
TABELA A.2 - Filtros utilizados na conversão das componentes (R,G,B) para (Y,Cr,Cb).....	98
TABELA A.3 - Tabela com os códigos de categoria para os elementos DC e AC (JPEG).....	99
TABELA A.4 - Códigos de comprimento variável para as componentes de luminância.....	100
TABELA A.5 - Códigos de comprimento variável para as componentes de crominância.....	102
TABELA A.6 - Códigos de comprimento variável para os vetores de movimento.	104
TABELA A.7 - Código de comprimento variável para os macroblocos nulos.....	105

APÊNDICE B

1. Algoritmo para o cálculo da DCT (IDCT).....	106
--	-----

REFERÊNCIAS BIBLIOGRÁFICAS.....	112
--	------------

RESUMO

Este trabalho tem por objetivo o estudo da qualidade de transmissão de sinais de vídeo em redes de faixa larga sujeitas à perda de pacotes. A robustez da transmissão é obtida por escalonamento (ou separação de camadas) do sinal de vídeo. Consideram-se escalonamentos por requantização e separação espectral, assim como diferentes níveis de perda de pacotes e taxas de transmissão constantes e variáveis.

O estudo baseia-se em simulações via *software* de um modelo de codificador-canal-decodificador. Resultados numéricos são analisados e curvas típicas (taxa x distorção) do desempenho do modelo são obtidas para os vários conjuntos de parâmetros. Estes resultados apontam algumas direções quanto à seleção de métodos robustos de transmissão de vídeo e permitem algumas comparações entre os esquemas implementados.

A contribuição deste trabalho concentra-se na implementação do modelo e na análise dos resultados de suas simulações.

ABSTRACT

The objective of this work is to study the quality of video signals transmitted over wideband networks subject to packet losses. Increased transmission robustness is achieved by “scalability” or layered segmentation of the video signal. We consider spectral separation and requantization methods of scalability, as well as different levels of packet losses and bit rates.

The study is based on software simulations of a coder-channel-decoder model. Numerical results are analyzed and typical rate-distortion curves are obtained for different sets of parameters. These results give guidelines in regard to the selection of robust methods of video transmission over lossy networks, and allow some comparisons between implemented schemes.

The main contributions of this work consist of the model implementation and the analysis of simulation results.

LISTA DE FIGURAS

2.1	Diagrama em blocos de um sistema de comunicação digital.....	6
2.2	Diagrama em blocos do codificador e do decodificador JPEG.....	10
2.3	Efeito da transformação na redução do número de <i>bits</i> do sinal.....	11
2.4	Seqüência periódica obtida por extensão da seqüência finita original.....	14
2.5	Seqüência periódica obtida pela reflexão (no eixo vertical) e replicação.....	15
2.6	Curva do quantizador linear.....	16
2.7	Processo de codificação diferencial dos elementos DC.....	18
2.8	Leitura "zig-zag" dos coeficientes de um bloco.....	18
2.9	Processo de codificação	20
2.10	Esquema da codificação e decodificação por predição e compesação de movimento.....	22
2.11	Técnica de casamento de blocos.....	24
2.12	Técnica de casamento de blocos com passos logaritmos.....	25
2.13	Esquemas de codificação e decodificação por interpolação.....	26
2.14	Processo de sub-amostragem das componentes Cr e Cb.....	29
2.15	Gama de serviços atendidos pelas redes BISDN.....	30
2.16	Processo de transmissão utilizando TDM.....	31
2.17	Processo de transmissão utilizando ATDM.....	31
2.18	Estrutura de uma célula ATM.....	32
2.19	Esquema e exemplo de separação espectral.....	35
2.20	Representação da imagem por um cubo mostrando o processo de separação espectral.....	35
2.21	Esquema da requantização.....	36
2.22	Representação da imagem por um cubo mostrando o processo de requantização.....	37
3.1	Diagrama mostrando os diversos módulos do ambiente de simulação.....	38
3.2	Codificador para a extração da redundância espacial e temporal.....	39
3.3	Exemplo de codificação com GOP = 9.....	44
3.4	Fluxograma de codificação dos macroblocos de um quadro.....	45
3.5	Palavra código associada à codificação de alta prioridade.....	47

3.6	Palavra código associada à codificação de baixa prioridade.....	48
3.7	Processo de formação das células a partir dos dados da sequência de vídeo...	49
3.8	Modelo para perda de células.....	52
4.1a	Curvas de taxa x distorção para as sequências FLOWER, KIEL, MOBILE....	58
4.1b	Primeiro quadro das sequências KIEL, FLOWER e MOBILE.....	60
4.1c	Comparação entre a qualidade subjetiva e o RMSE para o primeiro quadro da sequência FLOWER.....	61
4.1d	Comparação entre a qualidade subjetiva e o RMSE para o primeiro quadro da sequência KIEL.....	62
4.1e	Comparação entre a qualidade subjetiva e o RMSE para o primeiro quadro da sequência MOBILE.....	63
4.2	Formação do tráfego de células para o canal 2.....	64
4.3	Exemplo do tráfego com variação da taxa do codificador.....	65
4.4	Curvas de perda de células para as sequências codificadas sem escalonamento.....	66
4.5	Curva do quantizador linear com passos uniformes.....	69
4.6	Análise do erro de quantização no processo de requantização.....	70
4.7a	Curvas de perda de células para a requantização com $Q1/Q2 = 2$	71
4.7b	Curvas de perda de células para a requantização com $Q1/Q2 = 4$	72
4.8	<i>Overhead</i> na taxa de transmissão quando se utiliza requantização.....	73
4.9a	Curvas de perda de células para a separação espectral tipo 1.....	76
4.9b	Curvas de perda de células para a separação espectral tipo 2.....	77
4.10	<i>Overhead</i> na taxa de transmissão com separação espectral.....	78
4.11	Comparação entre as codificações com e sem escalonamento para um dado nível de perda de células (sequência FLOWER).....	80
4.12	Comparação entre as codificações com e sem escalonamento para um dado nível de perda de células (sequência KIEL).....	81
4.13	Comparação entre as codificações com e sem escalonamento para um dado nível de perda de células (sequência MOBILE).....	82
4.14a	Comparação entre a separação espectral tipo 1 e a requantização ($Q1/Q2=2$) para a sequência FLOWER.....	84
4.14b	Comparação entre a separação espectral tipo 2 e a requantização ($Q1/Q2=4$) para a sequência FLOWER.....	85
4.15a	Comparação entre a separação espectral tipo 1 e a requantização ($Q1/Q2=2$) para a sequência KIEL.....	86
4.15b	Comparação entre a separação espectral tipo 2 e a requantização ($Q1/Q2=4$) para a sequência KIEL.....	87
4.16a	Comparação entre a separação espectral tipo 1 e a requantização ($Q1/Q2=2$) para a sequência MOBILE.....	88
4.16b	Comparação entre a separação espectral tipo 2 e a requantização ($Q1/Q2=4$) para a sequência MOBILE.....	89

4.17	Taxas de perda de células em função do carregamento de uma rede ATM multiplexada.....	90
4.18	Determinação das curvas de taxa x distorção.....	91
5.1	Esquema de codificação ótimo para as diversas taxas de perda.....	95

LISTA DE TABELAS

2.1 Matriz de quantização típica do sinal de luminância utilizada pelo JPEG..... 17

4.1 Número de elementos direcionados ao canal de alta prioridade (ns) em função
do nível de quantização..... 75

CAPÍTULO 1

INTRODUÇÃO

Refletindo sobre as mudanças que estão ocorrendo no mundo, não temos dificuldade em perceber que estamos caminhando em direção a uma nova "revolução industrial": a revolução da informação.

Devido ao efeito da globalização, os fatores de produção têm se tornado abundantes e não servem mais como uma vantagem competitiva para as empresas. A vantagem proveniente dos fatores de produção vem sendo substituída pela rapidez com que as empresas antecipam as necessidades dos seus clientes e as satisfaçam mais velozmente que os seus concorrentes. Esta rapidez de ação está ancorada em novas necessidades, como rapidez na obtenção e divulgação da informação.

Estas novas necessidades provocam mudanças tecnológicas importantes nos campos da comunicação e transmissão da informação. Vejamos algumas tendências:

1. A tecnologia está migrando para uma total digitalização dos sistemas. Esta migração deve-se as muitas vantagens do sistema digital de comunicação frente aos seus concorrentes analógicos. Dentre estas, destacamos:
 - Possibilidade de unificação da informação advinda de várias fontes através da sua conversão em *bits* (unidades mínimas de informação) os quais podem ser misturados no transmissor, constituindo um fluxo único de transmissão, e recuperados no receptor, conservando as informações individuais de cada fonte.
 - Redução do ruído do canal de transmissão, o que permite comunicação eficiente em distâncias mais longas.
 - Facilidade no projeto dos sistemas propiciado pelo avanço da tecnologia microeletrônica no desenvolvimento de componentes complexos (que englobam muitas partes dos sistemas de comunicação), programáveis (com facilidade de substituição de *hardware* x *software*) e de baixos custos.
2. Constatamos uma crescente proliferação das redes de comunicação de dados. Esta proliferação se deve, em grande parte, ao aumento expressivo da demanda

encabeçada pelos sistemas de automação de escritórios, fábricas e negócios de modo geral, e ao aparecimento dos computadores pessoais, com aceitação e utilização quase universais. As redes de comunicação surgiram inicialmente como forma de solucionar problemas locais de comunicação. Com o aumento da demanda e a proliferação dos serviços, essas redes se multiplicaram, na sua grande maioria, de forma independente. Tal independência ocasionou uma incompatibilidades entre estas redes, levando a sistemas complexos e caros de gerenciamento. Surgiu, então, uma primeira proposta de integração dos serviços em uma única rede, denominada ISDN (*Integrated Service Digital Network*). Esta rede seria responsável por integrar serviços de voz, dados e vídeo de baixa qualidade (este último devido à sua banda relativamente limitada). Atualmente, esta concepção foi estendida para acomodar sinais com bandas muito mais largas (tais como sinais de vídeo, videoconferência, TV a cabo, etc.) levando ao aparecimento das redes BISDN (*Broadband Integrated Service Digital Network*).

3. Notamos um uso crescente das formas de comunicação visual (formas de comunicação que utilizam imagens, sinais de vídeo, etc.) que não podiam se desenvolver devido ao entrave tecnológico ocasionado pela extensa largura de faixa exigida para a transmissão. Com o advento das redes de faixa larga, estas formas de comunicação ganharam um grande impulso, incorporando-se aos serviços de comunicação.

Percebe-se, então, uma tendência na utilização das redes BISDN (ditas de faixa larga), propiciando um meio de comunicação eficaz para os mais diversos serviços. A utilização destas redes se deve à sua principal característica: fornecer uma largura de faixa diferenciada e adequada a cada serviço. Para este fim, elas utilizam uma tecnologia de transmissão denominada ATM (*Asynchronous Transfer Mode*), a qual agrupa a informação de vídeo em pequenos “pacotes” denominados células (em quantidades variáveis compatíveis com a necessidade de largura de faixa do sinal) e os transmite pelo canal.

Como toda nova tecnologia, a tecnologia ATM apresenta muitos desafios à sua implantação, desafios relacionados aos problemas de atraso e de perda de células no processo de transmissão-recepção.

O problema do atraso de células está relacionado à necessidade de sincronização entre transmissor e receptor. O sinal de sincronismo não é transmitido

em um canal separado, mais é englobado nas células enviadas ao receptor. Os atrasos variáveis sofridos pelas células quando trafegam pelas várias rotas disponibilizadas pela rede dificultam a regeneração do sinal de sincronismo e a decodificação do sinal de vídeo no receptor. Este problema pode ser parcialmente corrigido introduzindo-se atrasos variáveis entre as células no momento da recepção de modo a torná-los constantes, compensando o efeito da rede.

Os problemas relacionados com a perda de células na rede (que também podem ocasionar problemas de sincronização), se apresentam de duas formas:

- a) Perda de células devido à dimensão finita dos *buffers* nos elementos multiplexadores, os quais, estando sujeitos a tráfegos intensos, não conseguem armazenar todas as células recebidas para a posterior transmissão.
- b) Perda de células devido ao ruído de canal que, alterando os seus *bits* constitutivos, levam-na a inteligibilidade.

Estes problemas se acentuam à medida que o tráfego na rede se torna mais intenso (com a entrada de novos usuários), degradando o sistema com um todo.

Neste trabalho analisamos e propomos soluções para o problema de transmissão de sinais de vídeo em rede de faixa larga sujeita à perda de células, atingindo uma transmissão mais robusta de sinais neste tipo de rede. As perdas de células analisadas são aquelas devidas a capacidade finita dos *buffers* dos elementos multiplexadores da rede. Para procedermos a tal análise, foi elaborado um ambiente de simulação, implementado via *software*, composto por programas de codificação e decodificação de sinais de vídeo utilizando os padrões JPEG e MPEG com opções de escalonamento do sinal gerado; programas de geração de células ATM e programas que simulam um canal ATM sujeito a perdas de células. Com estes programas, temos oportunidade de avaliar qualitativamente e quantitativamente o efeito das perdas de células na qualidade no sinal de vídeo recebido e de propor esquemas de codificação que visem atenuar este efeito. Assim, este trabalho foi dividido em:

Capítulo 2: Neste capítulo desenvolvemos a teoria aplicada à transmissão e recepção de sinais de vídeo em redes de faixa larga com perdas de células. Analisa-se o diagrama de blocos de um sistema de comunicação para seqüências de vídeo, onde se destacam os codificadores e decodificadores de fonte e canal. A teoria que envolve o projeto dos codificadores e decodificadores de fonte (baseada nos padrões JPEG e MPEG) será analisada em detalhes visando a implementação prática, bem como a

teoria associada aos métodos de escalonamento (em especial a requantização e a separação espectral), utilizadas como forma de atenuar o efeito das perdas de células. Os codificadores e decodificadores de canal são também analisados em termos da formação das células ATM e de sua transmissão pelo canal digital.

Capítulo 3: Neste capítulo considera-se a implementação prática do ambiente de simulação baseado no desenvolvimento teórico do Capítulo 2. Este capítulo tem duplo objetivo: esclarecer os modelos e simplificações adotadas em todo o processo de codificação - transmissão - decodificação, de modo a se obter uma melhor compreensão dos resultados obtidos e servir de base para futuras implementações que visem melhorar a análise do problema.

Capítulo 4: Neste capítulo são analisados os resultados obtidos com as simulações. São avaliados os desempenhos dos sistemas de codificação (com e sem escalonamento) através da obtenção das curvas operacionais de taxa x distorção para os vários níveis de perda e os resultados são comparados visando o entendimento de como a perda de células afeta os sistemas de codificação e as possíveis soluções que pode atenuar o problema.

Capítulo 5: Este capítulo é dedicado às conclusões do trabalho, analisando os resultados obtidos e sugerindo novas propostas de pesquisa para um aprofundamento da análise do problema.

CAPÍTULO 2

O SISTEMA DE COMUNICAÇÃO PARA SINAIS DE VÍDEO

Neste capítulo, analisamos a teoria envolvida no processo de comunicação visual associada à transmissão e recepção de seqüências de vídeo. Iniciamos com a descrição geral de um sistema de comunicação digital, ressaltando os seus principais elementos constitutivos e como eles interagem. Particularizamos a descrição para o processo de comunicação digital visual (transmissão e recepção de seqüências de vídeo) e analisamos os principais blocos que compõem este sistema, dentre eles os codificadores e decodificadores de fonte e canal. A codificação e decodificação de fonte refere-se ao processo de diminuição da redundância e da entropia do sinal de vídeo original, visando uma transmissão eficiente. Por outro lado, os processos de codificação e decodificação de canal atuam protegendo as informações contra possíveis erros no canal de transmissão. O codificador de fonte, por ser um mapeamento do tipo "muitos-para-um", geralmente apresenta um projeto complexo. No caso de sinais de vídeo, o codificador envolve métodos de codificação espacial (codificação dos quadros individuais da seqüência de vídeo) e codificação temporal (envolvendo a codificação entre os quadros da seqüência). Em particular, analisamos os algoritmos propostos pelo JPEG (codificação espacial) e MPEG (parte referente à codificação temporal) para formularmos nosso codificador. Como trabalhamos com imagens coloridas, considera-se a teoria que trata da sua compressão, antes da eventual codificação de fonte, baseando-se na transformação do espaço de cor para redução da redundância. No processo de codificação de canal, analisamos a teoria envolvida com a codificação dos dados em células ATM e sua transmissão através de redes BISDN. Analisamos, em especial, o problema de transmissão (perda de células por congestionamento) nos multiplexadores da rede e como atenuar este problema através de métodos de escalonamento, que dividem a informação codificada em camadas e atribuem prioridades de perdas para cada uma delas.

2.1 - O SISTEMA DIGITAL DE COMUNICAÇÃO

O processo geral de comunicação utilizando tecnologia digital segue o diagrama da Figura 2.1.

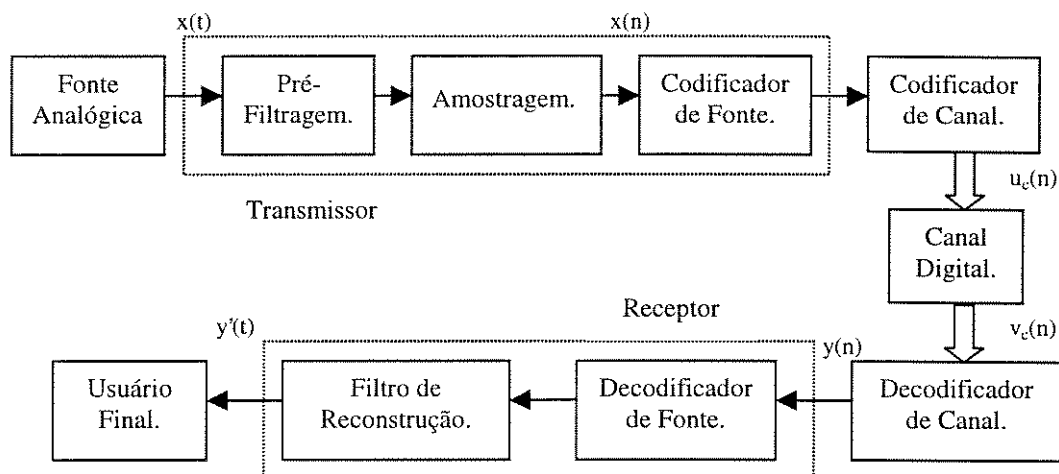


Fig. 2.1- Diagrama em blocos de um sistema de comunicação digital.

O sinal originário de uma fonte analógica (que pode representar os mais diversos sinais, como voz, vídeo, etc.) sofre uma conversão para sinal digital e posterior codificação, para poder ser convenientemente enviado pelo canal. Neste processo de conversão, é necessário transformar o sinal analógico (um sinal que é contínuo tanto no tempo como em níveis de amplitude) em um sinal que é discreto tanto em amplitude como no tempo. O processo de conversão se inicia através da amostragem do sinal analógico, onde são tomados os valores de sua amplitude com determinada frequência. Um dos teoremas fundamentais da teoria de comunicação afirma que, se o sinal for amostrado em uma frequência maior ou igual a duas vezes a banda do sinal (implicando que o sinal deve apresentar um espectro de frequência com banda limitada), conseguimos reconstruir inteiramente o sinal a partir de suas amostras, ou seja, não há perda de informação ao se proceder à amostragem do sinal. De modo a garantir uma banda limitada ao sinal, o mesmo é filtrado antes do processo de amostragem. Este filtro deve apresentar características de amplitude e fase que não levem a uma distorção acentuada do sinal. As amostras do sinal filtrado apresentam amplitudes contínuas e sofrem então um processo de quantização (ou seja, a

amplitude de cada amostra, que necessita de infinitos valores para ser representada, passa ser representada por apenas um conjunto discreto de valores). Por fim, as amostras quantizadas são codificadas em *bits* para transmissão. Todo este processo é realizado pelo que denominamos codificador de fonte, cuja função pode ser separada em duas partes principais: uma parte relacionada com a discretização da amplitude do sinal realizada por um quantizador (diminuição da entropia do sinal) e outra relacionada com a codificação destas amostras em *bits*, permitindo uma transmissão eficiente (codificador de entropia). Este codificador utiliza a estatística do sinal de entrada de modo a associar, eficientemente, seqüências de *bits* aos elementos quantizados, diminuindo a quantidade média de *bits* transmitida. Devido às características do sinal de entrada, o processo de codificação de fonte constitui-se de um mapeamento do tipo "muitos-para-um", em geral mais complexo que os decodificadores correspondentes, que são mapeamentos "um-para-um".

Os sinais produzidos pelo codificador de fonte sofrem novamente um outro tipo de codificação (codificação de canal) antes de serem transmitidos. Isto ser faz necessário, pois o canal não é ideal, apresentando ruídos que causam distorção e comprometem a qualidade da comunicação. A codificação de canal acrescenta redundância ao sinal codificado de modo reduzir o efeito do ruído. O balanceamento entre o codificador de entropia e o codificador de canal (o quanto de redundância retirar e em seguida acrescentar) constitui um dos principais problemas no projeto de um sistema de comunicação eficiente. O projeto do codificador de canal, por ser tipicamente um mapeamento "um-para-um", é geralmente de implementação mais fácil que o decodificador correspondente.

O sinal enviado pelo canal sofre o efeito do ruído. O sinal recebido é submetido à decodificação de canal que tenta regenerá-lo, corrigindo possíveis imperfeições através das redundâncias introduzidas pelo codificador. O processo de decodificação de canal constitui geralmente um mapeamento de "muitos-para-um", devido ao efeito do ruído. Por esta razão, os decodificadores de canal são, em geral, mais complexos que os codificadores correspondentes.

Após a decodificação de canal, o sinal é decodificado por um decodificador de fonte (mapeamento "um-para-um") reconstruindo o sinal amostrado original, o qual, através de um processo de filtragem por um filtro de reconstrução, gera o sinal

analógico $y(t)$ representativo de $x(t)$, o sinal analógico original. Notamos que, invariavelmente, o erro de quantização está sempre presente no sistema, pois a quantização se apresenta como um processo irreversível (i.e., um processo onde há perda de informação), muito embora possa ser controlado através da variação do número de níveis de quantização (o ruído de quantização diminui conforme o número de níveis aumenta).

2.2 - O CODIFICADOR E DECODIFICADOR DE FONTE

Descrevemos, a seguir, a teoria associada ao codificador e decodificador de fonte para sinais de vídeo. Supomos para a análise que o sinal de vídeo já se encontra devidamente amostrado segundo o Teorema de Nyquist, e quantizado, com os valores de suas amostras variando no conjunto $\{0,1,...,255\}$ (sendo representados por $\log_2(256) = 8 \text{ bits}$) para os vários quadros da sequência de vídeo. Este processo de quantização inicial, efetuado no momento da captura das imagem pela câmera de filmagem, não representa o processo de quantização utilizado pelo codificador de canal.

Um sinal de vídeo (composto por uma sequência de quadros apresentados sequencialmente no tempo) basicamente carrega dois tipos de informações: informações associadas a cada quadro em separado (informações espaciais) tais como a forma, a cor, a textura, a luminosidade, etc., dos elementos da cena e informações associadas aos movimentos dos elementos da cena através da sucessão destes quadros (informações temporais). Dividimos, para facilidade de análise, o método de codificação de fonte em duas etapas: uma denominada codificação intraquadro, (intra = interno) que atuará na codificação dos aspectos espaciais de cada quadro e uma etapa denominada interquadro (inter = entre) que atuará na codificação dos aspectos temporais entre os vários quadros da sequência de vídeo.

2.2.1 - CODIFICAÇÃO INTRAQUADRO

O método de compressão intraquadro utiliza somente as informações espaciais contidas em cada quadro da sequência de vídeo. Podemos ter:

a) Técnicas de Compressão Sem Perdas

Incluem-se aqui as técnicas de compressão que não causam perdas de informação na imagem a ser processada (são processos reversíveis). Por apresentarem essa reversibilidade, estas técnicas produzem baixas taxas de compressão frente a outras técnicas que admitem perda de informação. Dentre as principais técnicas de compressão sem perdas, podemos destacar a codificação no plano de *bits* [1] os códigos de Huffman [2-3] e os códigos aritméticos[4-5].

b) Técnicas de Compressão com Perdas

Neste grupo de técnicas, a compressão da imagem é acompanhada de uma perda de informação não recuperável na descompressão (processo irreversível). No entanto, conseguem-se boas taxas de compressão quando comparadas com as técnicas de compressão sem perdas. Algumas das principais técnicas de compressão com perdas são: codificação com truncagem de blocos [6-13], quantização vetorial [14-25], codificação por sub-bandas [26-35] e codificação por transformada [36-42].

Baseado no fato que os quadros das seqüências de vídeo geralmente apresentam muita redundância e que a qualidade é um fator subjetivo, já que varia de observador para observador, conseguimos com as técnicas de compressão com perdas um melhor desempenho de compressão frente as técnicas de compressão sem perdas. Dentre as técnicas de compressão com perdas, destacamos o método de codificação por transformada que apresenta boas de taxas de compressão [36] e simplicidade de implementação, sendo esta técnica a adotada em nosso trabalho. Baseado no método de codificação por transformadas, um grupo de pesquisadores do JPEG¹ [36][40][41] propôs um algoritmo de codificação simples e eficiente. Esse é um dos algoritmos utilizados como padrão para o desenvolvimento do nosso ambiente de simulação e pode ser visto na Figura 2.2.

¹ JPEG é a abreviação de *Joint Photograph Expert Group*.

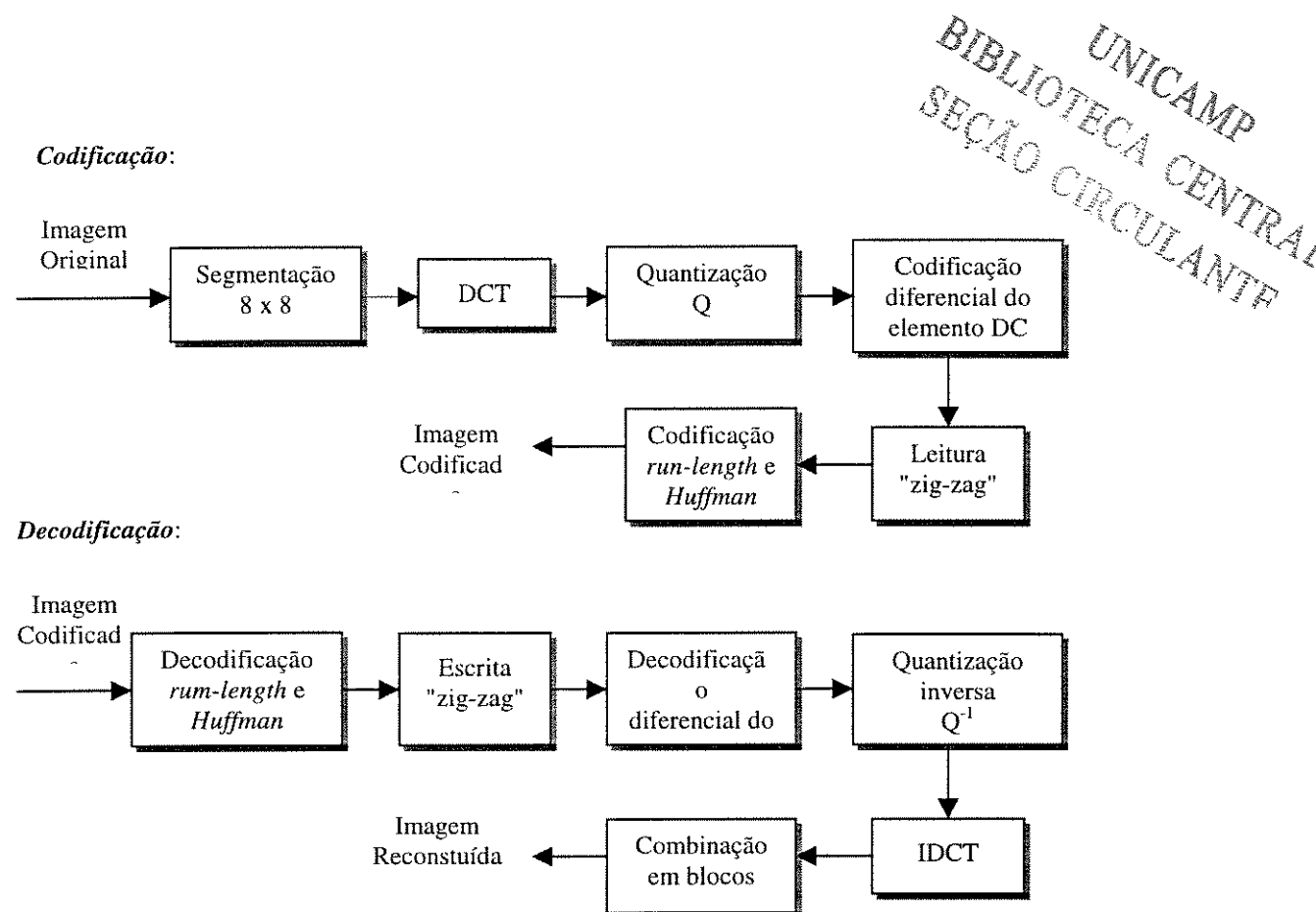


Fig. 2.2 - Diagrama em blocos do codificador e do decodificador JPEG.

A seguir comentamos a atuação de cada bloco do diagrama.

• SEGMENTAÇÃO 8 X 8

Inicialmente, a imagem original de $M \times N$ *pixels* é dividida em $m \times n$ blocos de tamanho 8×8 , os quais são submetidos a uma transformada (DCT). O valor 8×8 foi obtido experimentalmente de modo a se obter uma melhor relação entre compressão x qualidade da imagem reconstruída. Testes práticos demonstraram que um aumento da área utilizada para o cálculo da transformada (DCT) tende a ocasionar uma maior compressão na imagem ao mesmo tempo que aumenta a complexidade do processo.

• DCT

O núcleo do algoritmo do JPEG é a transformação DCT aplicada a cada bloco 8×8 da imagem de modo a descorrelacionar os seus elementos, reduzindo a quantidade de *bits* necessária para codificá-los. Este processo pode ser associado a uma mudança de base no espaço $\mathbf{R}^2$, sendo que os elementos da imagem nessa nova base necessitam de menos *bits* para serem representados. As funções base devem ser ortonormais associadas a uma transformação linear para facilitar os procedimentos de cálculo. A Figura 2.3 nos esclarece como uma mudança de base pode diminuir a quantidade de *bits* necessária para a representação do sinal.

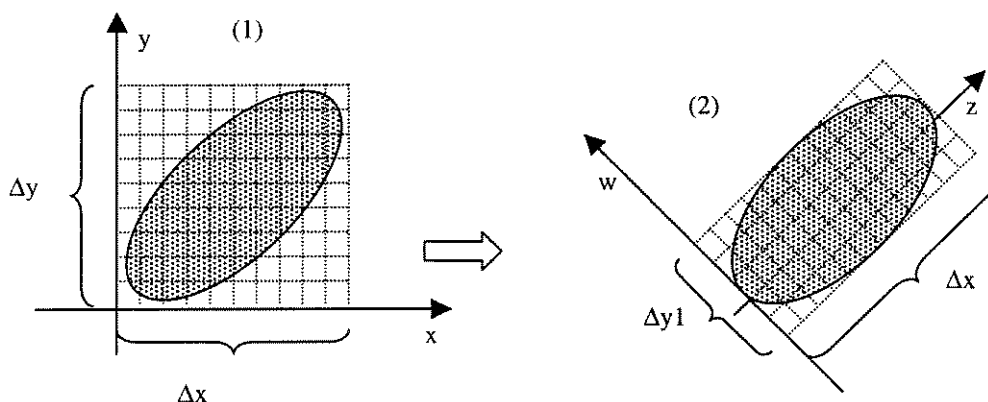


Fig. 2.3 - Efeito da transformação na redução do número de *bits* do sinal

- 1) Necessários $\Delta x \cdot \Delta y$ níveis discretos para se representar cada par (x, y) .
- 2) Necessários $\Delta x_1 \cdot \Delta y_1 < \Delta x \cdot \Delta y$ níveis discretos para se representar o mesmo par (x, y) .

No esquema (1) são necessários Δx níveis para se representar o sinal na sua componente x e Δy níveis para se representar o sinal na sua componente y para um determinado ruído de quantização. Portanto, necessitam-se $\Delta x \cdot \Delta y$ níveis discretos para representar cada par (x, y) . Como estes níveis estão associados ao número de *bits* necessários para a sua representação, não temos uma codificação eficiente, pois para um sinal com amostras correlacionadas, como mostrado na Fig. 2.3, aparecem regiões de baixa probabilidade de ocorrência para as quais são alocados *bits* de codificação. Quando efetuamos uma rotação no sistema de coordenadas, reduzimos estas regiões

(área não hachurada) de modo que temos uma redução no número de níveis discretos e como consequência uma redução na quantidade de *bits* necessária para a representação do sinal com um mesmo ruído de quantização.

A DCT (Transformada Discreta do Coseno) [36][43-47] efetua este tipo de rotação no espaço de sinais e foi a transformada escolhida pelo grupo de pesquisadores do JPEG devido a uma série de características que passamos a descrever sucintamente.

Para aplicações de processamento de imagens que lidam com duas dimensões (linhas e colunas de cada quadro), a DCT para um bloco de $N \times N$ elementos é dada por [36]:

$$F(u, v) = \frac{2C(u)C(v)}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j, k) \cos\left[\frac{(2j+1)u\pi}{2N}\right] \cos\left[\frac{(2k+1)v\pi}{2N}\right] \quad (2.1)$$

e a sua transformada inversa é definida por:

$$f(j, k) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u, v) \cos\left[\frac{(2j+1)u\pi}{2N}\right] \cos\left[\frac{(2k+1)v\pi}{2N}\right] \quad (2.2)$$

$$\text{onde } C(w) = \begin{cases} \frac{1}{\sqrt{2}} & \text{para } w = 0; \\ 1 & \text{para } w = 1, 2, \dots, N-1 \end{cases}$$

As principais características da DCT são as seguintes:

• A DCT é uma transformação com núcleo separável

O cálculo da DCT em duas dimensões pode ser efetuado separadamente para cada uma das suas dimensões, sendo o resultado final a composição dos resultados parciais.

De fato, se definirmos:

$$F(u, \dots) = \sqrt{\frac{2}{N}} C(u) \sum_{j=0}^{N-1} f(j, \dots) \cos \left[\frac{(2j+1)u\pi}{2N} \right] \quad (2.3)$$

como a DCT de N pontos em uma dimensão, temos para duas dimensões (facilmente generalizado para um número maior de dimensões):

$$\begin{aligned} F(u, v) &= \frac{2C(u)C(v)}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j, k) \cos \left[\frac{(2j+1)u\pi}{2N} \right] \cos \left[\frac{(2k+1)v\pi}{2N} \right] = \\ &= \sqrt{\frac{2}{N}} C(u) \sum_{j=0}^{N-1} \underbrace{\left[\sqrt{\frac{2}{N}} C(v) \sum_{k=0}^{N-1} f(j, k) \cos \left[\frac{(2k+1)v\pi}{2N} \right] \right]}_{F(j, v)} \cos \left[\frac{(2j+1)u\pi}{2N} \right] = \\ &= \sqrt{\frac{2}{N}} C(v) \sum_{j=0}^{N-1} F(j, v) \cos \left[\frac{(2j+1)u\pi}{2N} \right] \end{aligned} \quad (2.4)$$

Portanto, a transformada 2D pode ser obtida pela aplicação de duas transformadas 1D em cada uma das dimensões.

- A DCT unidimensional apresenta implementação computacional rápida, com ordem $O(N \log_2(N))$ operações para transformação de tamanho N .

Este resultado é conseguido graças a periodicidade das funções coseno para os quais se evitam cálculos repetitivos na geração dos coeficientes de transformação.

- A DCT é uma transformada real.

A DCT é uma transformação que, diferentemente de outras (por exemplo, a transformada de Fourier [48-49]), gera coeficientes reais. Este fato facilita os cálculos ao mesmo tempo que reduz a quantidade de memória necessária para a sua implementação.

- A DCT propicia alta decorrelação para os coeficientes transformados.

Para imagens típicas que exibem uma alta correlação entre *pixels*, o desempenho da DCT, em termos de decorrelacionar os elementos de um bloco, apresenta resultados próximos aos que seriam obtidos pela transformação de Karhunen-Loève [36] (que é a transformação ótima). De fato, pode-se mostrar que para uma fonte de sinal que segue o modelo de Markov de primeira ordem, com correlação entre as amostras se aproximando do valor unitário, o núcleo da transformação de Karhunen-Loève, baseada na estatística do sinal se aproxima do núcleo da DCT [38]. Como vantagem, temos que o núcleo da DCT é independente da estatística da imagem, embora atue tão eficientemente quanto a transformação ótima.

- A DCT evita a geração de componente espectrais de alta frequência permitindo uma compressão mais eficiente por quantização.

Podemos entender este fenômeno, analisando a DFT [48] (Transformada Discreta de Fourier) de uma seqüência finita simples (uma linha de blocos da imagem, por exemplo). Para calcularmos a DFT desta seqüência, construímos uma seqüência periódica baseada nesta seqüência e calculamos a Série Discreta de Fourier da mesma. A seqüência gerada é periódica e um período corresponde a DFT da seqüência original.

Devido à replicação da seqüência original na formulação da seqüência periódica, ocorrem descontinuidades nos seus segmentos, ocasionando a geração de componentes de alta frequência significativas, que não podem ser descartadas no processo de quantização, sob pena de ocasionarem erros de reconstrução (Fig. 2.4).

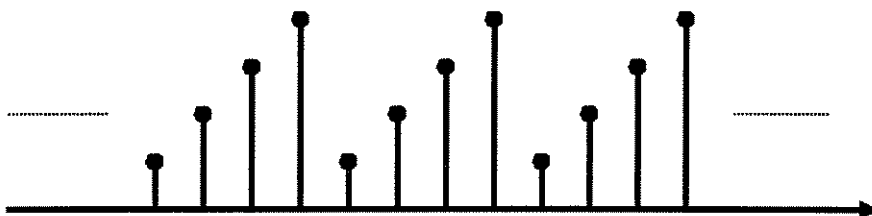


Fig. 2.4 - Seqüência periódica obtida por extensão da seqüência finita or

No nosso caso, com a imagem sendo “quebrada” em blocos de 8×8 *pixels*, formando uma seqüência finita em duas dimensões, estes erros de reconstrução resultam na visualização das fronteiras entre os blocos (efeitos de blocos).

Para reduzir o efeito de blocos, a seqüência finita original pode ser refletida no seu eixo vertical e replicada de modo a formar uma nova seqüência periódica, apresentando o dobro do período original. Novamente, calculando-se a Série Discreta de Fourier desta seqüência e tomando-se um período, teremos a DFT da seqüência finita original. O que notamos agora é que a DFT não apresenta mais componentes de alta freqüência significativas (Fig. 2.5). Pode ser demonstrado [44] que o processo de se estender a seqüência original como sugerido acima, corresponde (com pequenas modificações) a se calcular a DCT da seqüência finita original. Assim, a DCT não apresenta componentes de alta freqüência significativas se comparada com a DFT, permitindo uma maior concentração da energia nas componentes de baixa freqüências.

Pode-se, então, obter uma compressão de alta eficiência descartando as componentes de alta freqüência sem um efeito de borda acentuado.

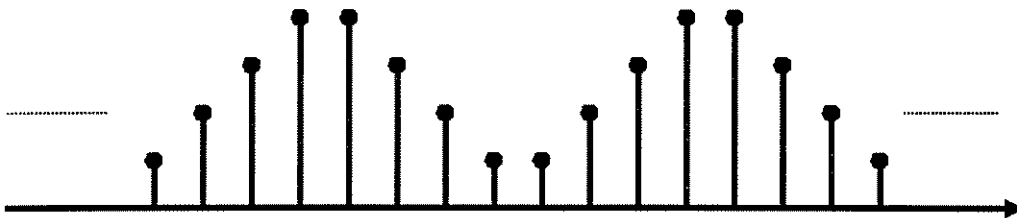


Fig. 2.5 - Seqüência periódica obtida pela reflexão (no eixo vertical) e replicação

• QUANTIZAÇÃO

O processo de quantização limita o conjunto dos possíveis valores assumidos pelos coeficientes de cada bloco transformado, permitindo uma codificação mais eficiente.

A quantização atua mapeando os valores pertencentes a um certo intervalo em um único valor que passa a ser o representante deste intervalo. Na Figura 2.6 podemos verificar a curva típica do quantizador utilizado em nosso trabalho. Este quantizador é dito uniforme por atuar sobre intervalos de largura constante. O passo de quantização

(diferença entre níveis consecutivos de quantização) é constante e igual a Δ , sendo do tipo "midtread" pois mapeia os valores entre $(-\Delta/2$ a $\Delta/2)$ no valor 0. A quantização é realizada dividindo-se o valor do elemento a ser quantizado pelo passo de quantização e arredondando este resultado para o valor inteiro mais próximo. Este valor passa a ser o elemento representativo do dado quantizado.

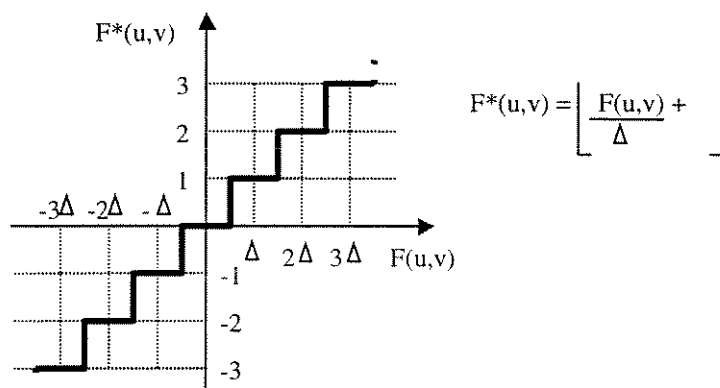


Fig. 2.6 - Curva do quantizador

$F(u,v)$ = elemento de cada bloco na posição (u,v) a ser quantizado.

$F^*(u,v)$ = elemento correspondente quantizado.

Δ = coeficiente de quantização.

O valor de Δ determina quantos *bits* serão necessários para representar o valor de $F(u,v)$ (coeficiente da transformada na posição (u,v)). O valor de Δ pode ser escalonado para se obter vários níveis de compressão. As degradações visuais provocadas pelo processo de quantização podem ser reduzidas adaptando-se os coeficientes às características psicovisuais do olho humano. Isto que pode ser feito adotando-se coeficientes de quantização (Δ) diferentes para os vários coeficientes $F(u,v)$ do bloco. Valores maiores de Δ são utilizados para quantizar os coeficientes da DCT de alta frequência, já que o sistema visual humano é menos sensível a estes estímulos, e valores menores de passo de quantização são usados para os coeficientes da DCT de mais baixas frequências, que são mais importantes para o sistema visual humano. A Tabela 2.1 mostra uma matriz típica utilizada pelo JPEG, com os coeficiente (passos) de quantização obtidos experimentalmente.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Tab. 2.1 - Matriz de quantização típica do sinal de luminância utilizada pelo JPEG.

Outras matrizes de quantização podem ser adotadas para se proceder a quantização dos blocos da imagem. No Apêndice A (Tabela A.1) são mostradas algumas matrizes utilizadas em testes com nosso codificador.

• CODIFICAÇÃO DIFERENCIAL

Após o cálculo da DCT, o coeficiente de maior importância no bloco passa a ser o que ocupa a posição (0,0), pois ele representa a média do bloco em questão. Ele é denominado elemento DC enquanto os demais são chamados elementos AC.

Como o elemento DC geralmente apresenta correlação elevada como o elemento DC do bloco anterior, é realizada uma codificação diferencial entre estes elementos, o que diminui o número de *bits* necessários à sua codificação. A codificação é realizada calculando a diferença entre o valor do elemento DC de um bloco e o valor do elemento DC do bloco anterior, resultado no coeficiente diferencial do bloco em questão. Este procedimento é realizado por todos os blocos que compõem um *slice*, ou seja, uma linha de blocos na dimensão horizontal da imagem, repetido por todas as linhas de blocos até o limite da dimensão vertical da imagem. Para cada novo *slice*, a predição do primeiro elemento DC tem valor nulo (Fig. 2.7).

$$\begin{array}{ccc}
 DC_{00} & DC_{01} & \dots DC_{0N} \\
 DC_{10} & DC_{11} & \dots DC_{1N} \\
 \dots & \dots & \dots \\
 DC_{M0} & DC_{M1} & \dots DC_{MN}
 \end{array}
 \Rightarrow
 \begin{array}{ccc}
 DC'_{00} & DC'_{01} & \dots DC'_{0N} \\
 DC'_{10} & DC'_{11} & \dots DC'_{1N} \\
 \dots & \dots & \dots \\
 DC'_{M0} & DC'_{M1} & \dots DC'_{MN}
 \end{array}
 \quad
 \begin{array}{l}
 DC'_{i,j} = DC_{i,j} - DC_{i,j-1} \\
 j = 0, 1, \dots, N \\
 i = 0, 1, \dots, M \\
 DC_{i,-1} = 0
 \end{array}$$

Fig. 2.7 - Processo de codificação diferencial dos elementos DC.

• LEITURA "ZIG-ZAG" DOS COEFICIENTES

A quantização realizada pela matriz do JPEG tem a peculiaridade de apresentar passos de quantização crescentes à medida que se distancia do elemento DC (posição (0,0)), em direção ao último elemento AC (posição (7,7)). Em vista desta característica, estatisticamente, encontramos mais elementos nulos na parte inferior direita de cada bloco do que na parte superior esquerda (vide Tab. 2.1). Para se conseguir uma codificação eficiente, aproveitando a longa seqüência de zeros gerada pela quantização, procedemos a uma leitura dos coeficientes na ordem "zig-zag" conforme mostrado na Figura 2.8.

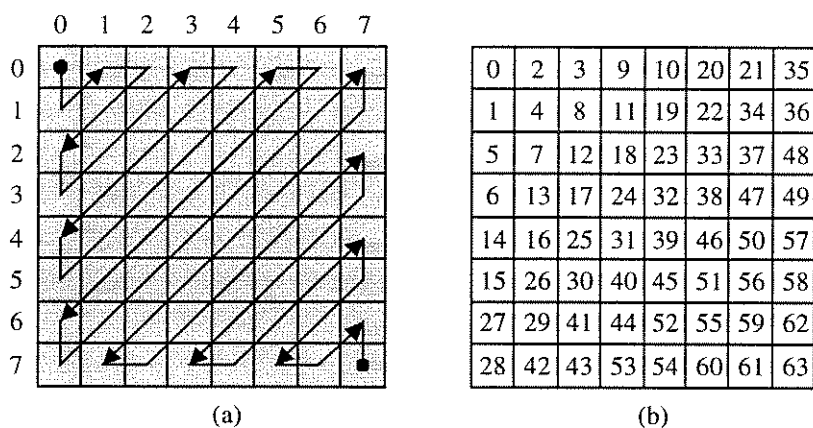


Fig. 2.8 - Leitura "zig-zag" dos coeficientes de um bloco.

(a) ordem de leitura, (b) Posição dos elementos no bloco.

• CODIFICAÇÃO RUN-LENGTH

A codificação *run-length* (codificação por corrida de zeros) atua associando códigos aos dados lidos segundo a ordem "zig-zag", aproveitando a longa seqüência de zeros gerada pelo processo de. Neste modo de codificação é feita uma separação entre as codificações do elemento DC e os elementos AC restantes, já que o elemento DC é tido como sendo o mais importante no bloco.

A codificação se dá associando ao(s) dado(s) lido(s) uma palavra binária composta por dois campos distintos:

- ♦ O primeiro campo de *bits* contém a informação da categoria do dado (se for o elemento DC do bloco) ou do número de zeros anteriormente lidos antes do primeiro

dado não nulo, segundo a leitura “zig-zag”, acrescido de sua categoria (se for um elemento AC).

- ♦ O segundo campo será composto pela codificação do dado segundo a sua categoria.

A categoria do dado representa o número de *bits* necessários para sua codificação sendo definida como um número k , onde $k \in [1, \dots, 10]$, para um dado cujo valor esteja compreendido nas faixas $(2^{k-1}, 2^k - 1)$ ou $(-2^k + 1, -2^{k-1})$. No Apêndice A (Tabela A.3), encontramos uma tabela que nos auxilia a encontrar a categoria do dado em função do seu valor. Assim, por exemplo, se temos um dado de valor 10, observamos na tabela que ele pertence à faixa de 8 a 15, correspondendo a categoria $k = 4$ (tanto para DC como AC). A categoria (dado DC) ou a categoria do primeiro dado não nulo junto ao comprimento da corrida de zeros é associada a um código de Huffman [50-51]. Este código é calculado segundo a estatística de imagens típicas. No Apêndice A (Tabelas A.4 e A.5) apresentam-se duas tabelas de códigos de Huffman, uma para os sinais de luminância e outra para os sinais de cromaticidade. Estes códigos foram produzidos pelo JPEG após várias experiências com a codificação de imagens de diversos tipos, sendo adotados em nosso trabalho. Nessas tabelas, podemos verificar a existência de códigos para dados DC e para dados AC. No caso dos dados AC, notamos a existência de duas palavras de código especiais: uma palavra que nos informa que todos os elementos restantes do bloco são nulos (código EOB - *End of Block*) e uma palavra que informa que uma sequência de mais de quinze zeros consecutivos foi encontrada, devendo o processo de contagem de zeros ser reiniciado. Essa palavra de código forma a primeira parte do processo de codificação. A segunda parte é composta pela representação binária do dado de acordo com a sua categoria :

- ♦ Se o dado é positivo, a codificação é simplesmente a representação binária do mesmo (Ex.: dado = 10; $k=4$; codificação = 1010).
- ♦ Se o dado é negativo, a codificação é o complemento da representação binária do módulo do dado (Ex.: dado = -10; $k=4$; módulo do dado = $|-10| = 10$; representação binária do módulo do dado = 1010; codificação do dado = 0101).

A Figura 2.9 mostra o processo de codificação descrito acima:

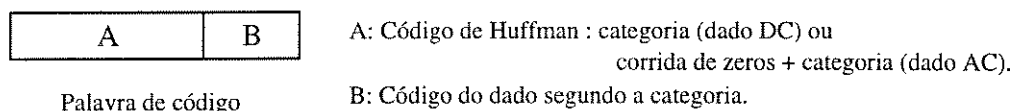


Fig. 2.9 - Processo de codificação.

Para esclarecer melhor este tipo de codificação, suponhamos que após a leitura dos elementos de um bloco (componentes de luminância) segundo a ordem "zig-zag", encontramos a seguinte sequência de valores: 80 0 0 0 0 -1 0 00, ou seja, o elemento DC vale 80, e encontramos, a seguir, uma corrida de 4 zeros consecutivos até obtermos o primeiro elemento não nulo igual a -1. Após este elemento, verificamos a existência de elementos nulos até o final do bloco. A codificação para este bloco será:

Elemento DC:

DC = 80 $\rightarrow$ categoria (Tabela A.3) = 7 $\rightarrow$ código da categoria = 11110 (Tabela A.4a) $\rightarrow$ código do dado: 80 em binário = 1010000 $\rightarrow$ Palavra código = 111101010000 (A+B)

Primeiro elemento AC:

Corrida de 4 zeros e dado de valor -1 $\rightarrow$ categoria do dado = 1 (Tabela A.3) $\rightarrow$ corrida de zeros = 4 ; categoria = 1 $\rightarrow$ código = 111011 (Tabela A.4b). $\rightarrow$ código do dado $|-1| = 1$, complemento = 0 $\rightarrow$ Palavra código: 1110110 (A+B)

Segundo elemento AC:

Restante dos elementos do bloco iguais a 0 $\rightarrow$ código especial EOB (fim de bloco) = 1010 (Tabela A.4b).

Assim, a codificação do bloco fica sendo: 11110101000011101101010.

No caso de imagens coloridas, cada componente (luminância e crominâncias) terá uma codificação (códigos de Huffman diferentes para cada componente). Na Tabela A.5, podemos verificar as tabelas de codificação para as componentes de crominância.

O processo de decodificação corresponde às operações inversas. Assim, a sequência de *bits* correspondente à imagem codificada será decodificada segundo as tabelas do código de Huffman correspondentes, sendo os dados reinseridos nos

respectivos blocos da imagem segundo a ordem de leitura "zig-zag". Após a reinserção, os elementos DC originais são restabelecidos pelo processo inverso da codificação diferencial que os originou; são recuperados pela sua multiplicação com os coeficientes da matriz de quantização e sofrerão a transformação inversa do cosseno (IDCT) restaurando os dados da imagem alocados por blocos. Os blocos, então, são combinados formando a imagem reconstruída (Fig. 2.2). Os erros de reconstrução são determinados pela matriz de quantização, pois todas as outras operações são reversíveis.

2.2.2 - CODIFICAÇÃO INTERQUADRO

Um segundo tipo de redundância que se verifica em seqüências de vídeo é a redundância temporal, advinda da forte correlação entre quadros consecutivos de uma seqüência. Segundo [42], geralmente menos de 5% da área de um quadro sofre modificações acentuadas durante a transmissão de uma seqüência de vídeo (praticamente o fundo da cena permanece imutável) acarretando grande correlação entre quadros. Uma das maneiras de eliminar esta redundância é utilizando a codificação preditiva interquadros que basicamente descreve os elementos de cena de um quadro baseados em elementos de outro quadro. Esta descrição se faz em termos de duas componentes: uma componente é associada ao possível deslocamento que um certo elemento de cena tenha sofrido em relação àquele que se tem como referência (vetor de deslocamento). O componente é associada à diferença entre estes dois elementos (elementos do quadro que se deslocou e o que serviu de referência para este deslocamento). Espera-se que esta diferença resulte em muitos elementos nulos e proporcione uma compressão.

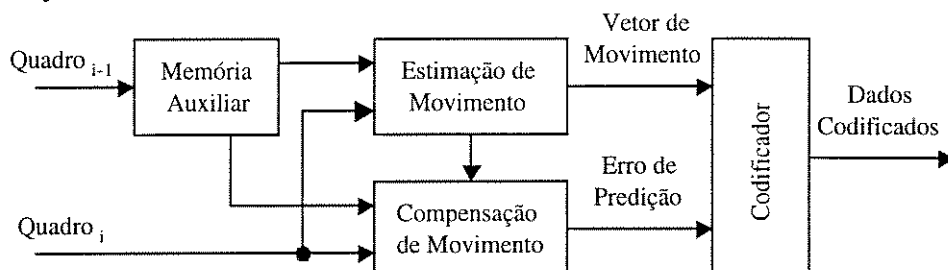
Existem vários métodos para se executar esta operação. Adotaremos os métodos utilizados no padrão MPEG² [42][52-57] que passamos a descrever:

² MPEG é a sigla de *Motion Picture Experts Group*, o grupo de especialistas que desenvolveu este padrão

2.2.2A - CODIFICAÇÃO POR PREDIÇÃO

Um esquema que realiza este tipo de codificação pode ser visto na Figura 2.10.

Codificador



Decodificador

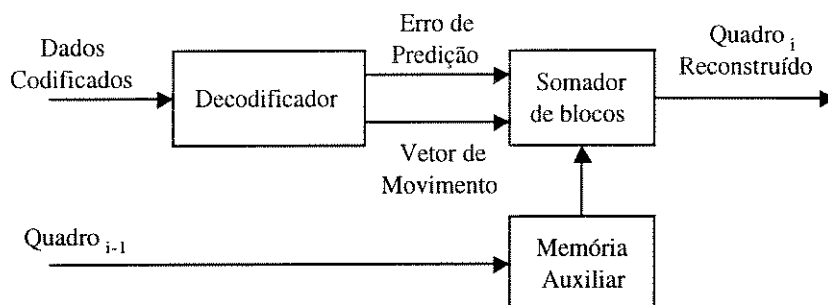


Fig. 2.10 - Esquema da codificação e decodificação por predição e compensação de m

Inicialmente, o quadro anterior³ (quadro_{i-1}) é armazenado em uma memória auxiliar. Quando da chegada do próximo quadro (dito quadro atual – quadro_i), efetua-se uma estimativa de movimento. Sendo o quadro dividido em blocos, tenta-se estimar para cada bloco do quadro atual qual é o bloco do quadro anterior que mais se assemelha a este segundo um dado critério. O resultado desta operação é um par de coordenadas relativas que indicam a posição do bloco mais correlatado localizado no quadro anterior em relação ao quadro atual. Este par de coordenadas representa o vetor de movimento.

De posse do vetor de movimento, efetua-se a compensação de movimento que consiste numa subtração realizada entre os elementos do bloco do quadro atual e do seu correspondente deslocado no quadro anterior, gerando um bloco de erro de predição. Estes blocos de erro, juntamente com os vetores de movimento calculados, são codificados e transmitidos. Notamos que este processo é um processo reversível,

já que qualquer bloco do quadro atual pode ser integralmente reconstruído de posse das informações de erro, dos vetores de movimento e do quadro anterior.

O processo de decodificação é bastante simples. Das informações codificadas recuperamos os vetores de movimento e as informações de erro (blocos de erro). Com estes dados e com os dados do quadro anterior previamente armazenado na memória auxiliar, somamos todos os blocos do quadro atual aos blocos de erro correspondentes aos vetores de movimento, recuperando, assim, a imagem original (Fig. 2.10).

O processo é mais eficiente quando se obtém uma boa estimativa pois, com ela, teremos vários zeros nos blocos de erro. Várias técnicas de estimativa são sugeridas, dentre as quais destacamos:

- Técnica do gradiente [58-62].
- Técnica recursiva *pel* [58][63-65].
- Técnica de casamento de blocos [58][66-68].
- Técnicas no domínio da frequência [58][69-71].

Das técnicas apresentadas, trabalharemos apenas com a técnica de casamento de blocos, por ser uma técnica que associa simplicidade de operações (rapidez em termos computacionais) e resultados satisfatórios.

TÉCNICAS DE CASAMENTO DE BLOCOS

Inicialmente, é atribuído a cada bloco do quadro um vetor de posição (x,y) indicativo da posição que o bloco ocupa no mesmo. Um modelo mais simplificado da estimativa de movimento considera que somente translações são efetuadas [58]. Temos portanto:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix} \quad (2.5)$$

$\uparrow$
 Translação

³ Este quadro é um quadro já processado (quadro reconstruído), como será visto posteriormente.

O modelo dado em (2.5) tem a grande vantagem da simplicidade e, em se tratando de imagens reais, a sua adoção não ocasiona grandes discrepâncias quando comparado com outros modelos mais complexos [58].

Assim, baseado neste modelo de movimento, executamos, para cada bloco do quadro atual de dimensão W , uma procura de um bloco de mesmas dimensões sobre uma área S do quadro anterior, de modo que as disparidades entre os seus elementos, seguindo um determinado critério, sejam minimizadas. Encontrado este bloco, o vetor de movimento consiste na diferença entre os vetores de posição dos blocos nos dois quadros (Fig. 2.11).

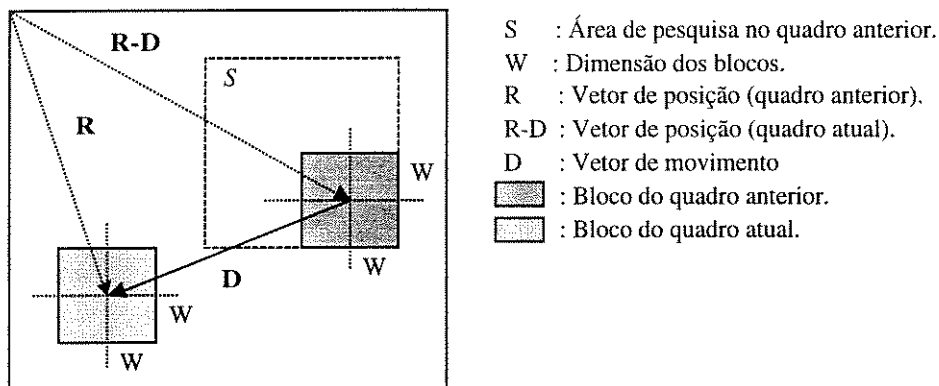


Fig. 2.11 - Técnica de casamento de blocos.

Obtidos esses vetores, o erro de predição pode ser encontrado pela subtração entre os blocos, como já comentamos.

Vários critérios para se minimizar as disparidades entre os blocos podem ser adotados. Os mais comuns tentam minimizar a diferença quadrática média entre os elementos dos blocos, a diferença absoluta média entre os mesmos ou ainda obter os blocos que apresentam a maior correlação.

Para se encontrar o bloco "ótimo", é necessário efetuar a pesquisa por toda a área S , o que demanda grande esforço computacional. De modo a contornar esse problema, foram desenvolvidos vários algoritmos de pesquisa [58][73-74] cuja função é fornecer uma solução aproximada em detrimento a um menor esforço

computacional. Uma destas técnicas utilizada em nosso trabalho pode ser visualizada na Figura 2.12.

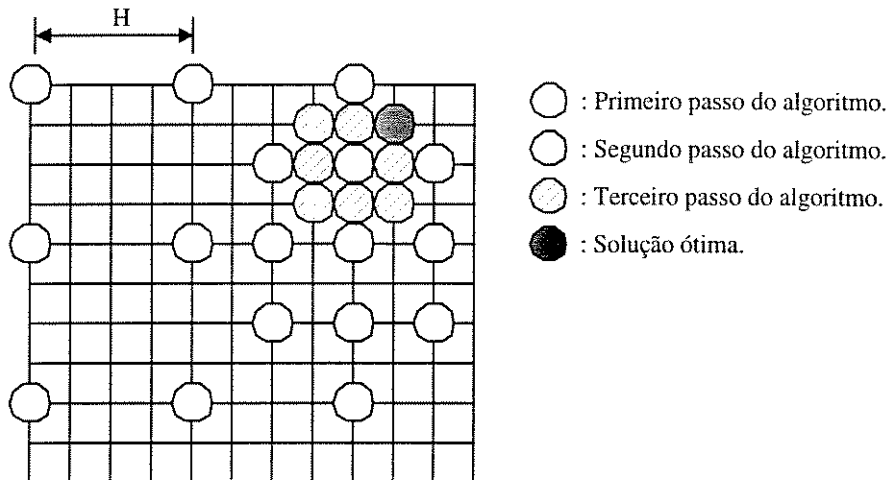


Fig. 2.12 - Técnica de casamento de blocos com passos logarítmicos.

A técnica funciona definindo-se uma área de pesquisa quadrada S de pesquisa, tendo como ponto central o bloco que se deseja estimar. Definindo-se H como a distância do centro da área a uma de suas bordas (vide figura), procedemos o primeiro passo do algoritmo calculando o valor do casamento nas 9 posições obtidas por deslocamento de $-H$, 0 , ou $+H$ nas ordenadas horizontal e vertical em relação ao bloco sendo considerado e selecionando-se o bloco que resulta no melhor casamento. A partir da posição deste bloco, reduzimos a distância por um fator de 2 e procedemos a uma nova pesquisa para as 8 novas posições em torno desta nova área, selecionado novamente o melhor bloco. Reduzimos novamente a distância por um fator de 2 e repetimos o processo até encontrarmos uma distância menor que 1. Neste ponto temos um bloco cujo casamento é, pelo menos, um máximo local⁴.

2.2.2B - CODIFICAÇÃO POR INTERPOLAÇÃO

O esquema de codificação e decodificação por interpolação [42][52] é mostrado na Figura 2.13:

⁴ Se o critério de casamento entre os blocos for uma função monotônica da distância D , a

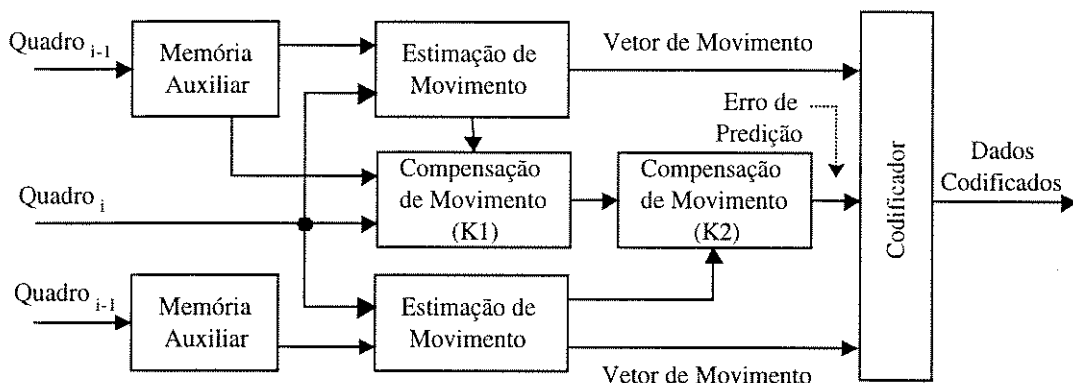
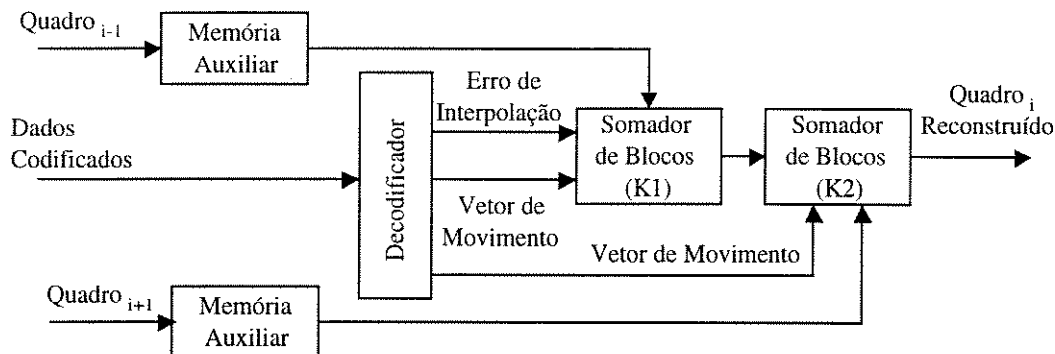
Codificador**Decodificador**

Fig. 2.13 - Esquema da codificação e decodificação por interpolação.

O esquema de codificação por interpolação é uma versão estendida da codificação por predição, pois além de executarmos uma estimativa com relação à imagem anterior (nos moldes que fizemos para a codificação por predição), executamos outra com relação à imagem posterior. Assim, para um dado bloco do quadro atual (quadro_i), procura-se o melhor bloco no quadro anterior (quadro_{i-1}) e o melhor bloco no quadro posterior (quadro_{i+1}). Encontrados tais blocos, são gerados dois vetores de movimento referentes às suas localizações. É também gerado o bloco de erro de interpolação correspondendo à diferença entre o bloco do quadro atual e uma combinação linear dos blocos encontrados ponderados com os coeficientes k_1 e $k_2 = 1 - k_1$. O erro de interpolação, juntamente com os vetores de movimento obtidos, são codificados e enviados pelo canal de transmissão. Por apresentar duas predições

espera-se uma melhor compressão, principalmente quando há mudanças acentuadas de cena, já que na transição de uma cena para outra, tem-se a oportunidade de escolher o melhor quadro para proceder à interpolação através do ajuste do coeficiente k_1 (predição somente com relação ao quadro anterior ($k_1=0$), "predição" somente com o quadro posterior ($k_1=1$) ou um misto de ambos $0 < k_1 < 1$). Esta melhor estimativa resultará, via de regra, em um menor erro, ainda que aumente o número de vetores de movimento associados à codificação e a complexidade dos cálculos.

O processo de codificação é análogo ao da decodificação por predição, sendo o erro de interpolação somado aos blocos deslocados correspondentes nos quadros anterior e posterior segundo os fatores de ponderação k_1 e k_2 . Na escolha do parâmetro k_1 deve ser levar em conta a correlação entre quadros de modo a se maximizar a estimativa.

2.3 - TRATAMENTO COM IMAGENS COLORIDAS

Tipicamente, uma imagem colorida pode ser representada pelas três componentes relativas às cores vermelha(*Red*), verde(*Green*) e azul (*Blue*). Tem-se assim os valores $R(i,j)$, $G(i,j)$ e $B(i,j)$ que compõem um *pixel* da imagem na posição (i,j) do quadro. Podemos pensar em comprimir uma imagem colorida, simplesmente comprimindo cada uma de suas componentes separadamente pelos padrões já estudados. Este processo leva a uma compressão ineficiente devido ao fato que, geralmente, ocorre uma forte correlação entre os planos de cores R,G e B [75].

Para se realizar uma compressão eficiente, deve-se proceder a descorrelação dos planos R,G e B. Dentre os vários métodos empregados [76][77], adotamos um que utiliza uma transformação linear dos planos R,G e B [76][78]. A técnica consiste em se transformar, via uma transformação linear, as componentes R, G e B em uma componente acromática denominada luminância (Y) e duas componentes cromáticas. Devido ao fato que o sistema visual humano é muito mais sensível à componente acromática, as componentes cromáticas podem ser codificadas com menor precisão, sem prejuízo da qualidade subjetiva final após a transformação inversa. Do fato de

codificarmos as componentes cromáticas com menor precisão, provêm a compressão desejada.

As transformações lineares que geram as componentes de luminância (Y) e as componentes de croma (Cr e Cb) adotadas neste trabalho são apresentadas abaixo [79][80]:

• Transformação direta:

$$\begin{bmatrix} Y(i, j) \\ Cr(i, j) \\ Cb(i, j) \end{bmatrix} = \begin{bmatrix} 0,299 & 0,5870 & 0,1440 \\ 0,5000 & -0,4187 & -0,0183 \\ -0,1687 & -0,3313 & 0,5000 \end{bmatrix} \times \begin{bmatrix} R(i, j) \\ G(i, j) \\ B(i, j) \end{bmatrix} \quad (2.6)$$

• Transformação inversa:

$$\begin{bmatrix} R(i, j) \\ G(i, j) \\ B(i, j) \end{bmatrix} = \begin{bmatrix} 1,0000 & 1,4020 & 0 \\ 1,0000 & -0,7141 & -0,3141 \\ 1,0000 & 0 & 1,7726 \end{bmatrix} \times \begin{bmatrix} Y(i, j) \\ Cr(i, j) \\ Cb(i, j) \end{bmatrix} \quad (2.7)$$

Após efetuarmos a transformação, praticamente toda energia dos planos R, G e B se concentrará na componente de luminância (Y) (para imagens naturais com banda limitada este percentual é aproximadamente 93% [76]). Esta concentração de energia nos permite sub-amostrar as componentes por um fator de 4:1, ou seja, para cada quatro componentes de luminância, tomamos uma componente de croma Cr e uma componente Cb (Fig. 2.14).

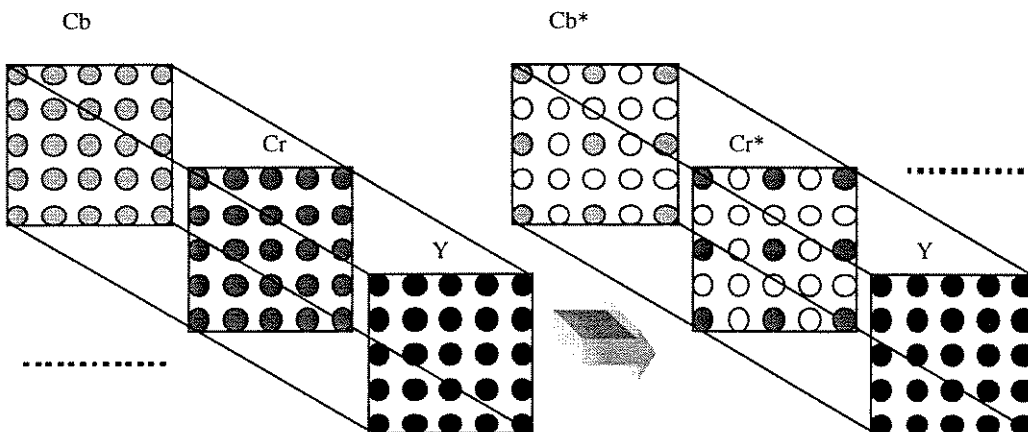


Fig. 2.14 - Processo de sub-amostragem das componentes Cr e Cb (* denota sinal amostrado).

Após a sub-amostragem, faz-se necessária uma filtragem para se evitar distorção por *aliasing*. Os filtros utilizados podem ser obtidos de modo heurístico. Adotaremos o padrão utilizado nos laboratórios de processamento de imagem do CPqD (Centro de Pesquisa e Desenvolvimento) da TELEBRÁS (vide Tabela A.2).

No processo de reconstrução da imagem, as componentes Cr e Cb sofrem uma interpolação, recuperando o número original de amostras. A interpolação é feita acrescentando-se zeros nos locais onde originalmente havia amostras e procedendo-se a uma filtragem de passa baixas de modo a eliminar frequências espúrias. Os filtros utilizados para interpolação são também apresentados no Apêndice A (Tabela A.2). Recuperando-se as componentes Cr e Cb, procede-se a transformação linear inversa para a reconstrução das componentes R, G e B.

2.4 - O CODIFICADOR E DECODIFICADOR DE CANAL

No mundo atual, observa-se uma demanda crescente por serviços de comunicações e, em especial, por serviços que exigem maior capacidade de canal para a transmissão, tais como: serviços multimídia, serviços de TV a cabo, vídeofone, etc. O aumento e a diversidade dos serviços de comunicações, classificados como sendo iterativos (ex.: vídeofone) ou distributivos (ex.: TV a cabo), inviabilizou a construção de redes dedicadas para o seu atendimento, principalmente devido à complexidade e aos altos custos envolvidos.[81-83].

Neste contexto, surgiu o conceito de uma rede capaz de acomodar os mais diversos tipos de serviços e modos de comunicação. Assim, serviços que exigem baixa capacidade do canal (serviços de telefone, telemetria, telex, fax, etc.) e alta capacidade de canal (serviços de vídeofone, TV a cabo, serviços de transmissão de dados em alta velocidade, transmissão de imagens em alta resolução, etc.), deveriam conviver em harmonia neste nova rede. Esta rede denominada Rede Digital de Serviços Integrados de Faixa Larga (RDSI-FL) [81][84-88] deveria ser capaz de transmitir sinais com taxas de alguns *bits* por segundo até sinais com taxas com centenas de megabits por segundo (Fig. 2.15).

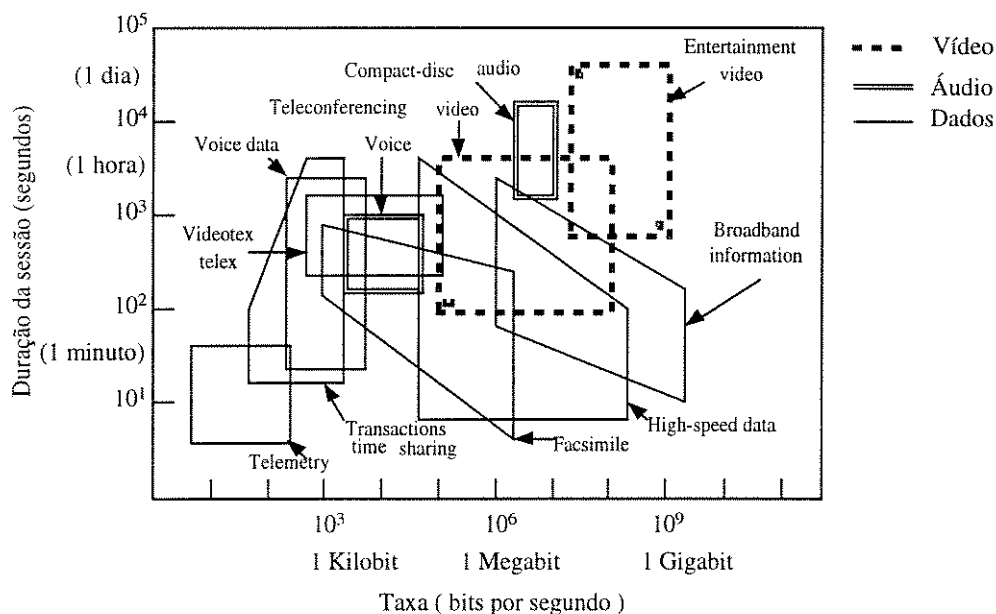


Fig. 2.15 - Gama de serviços atendidos pelas redes BISDN.

Um dos caminhos escolhidos para a implementação das redes RDSI-FL é utilizar a técnica de transmissão *Asynchronous Transfer Mode* (ATM) [81][84][89-90] que atua segmentando os vários sinais representativos dos serviços, mapeando-os em células de tamanho fixo denominadas células ATM [81][84][92-96] e transmitindo-as, em seguida, através de um canal utilizando multiplexação por divisão temporal assíncrona (ATDM - *Asynchronous Time Division Multiplex*). Este modo de transmissão é um pouco diferente do tradicional (TDM) que funciona de maneira síncrona, ou seja, onde as células que chegam ao elemento multiplexador são liberadas para trafegar na rede em consonância com o sinal de relógio do elemento multiplexador. Se em dado instante um ou mais canais que se conectam ao elemento multiplexador não apresentarem células, a multiplexação continuará sendo efetuada e como consequência teremos, na saída do elemento, um fluxo de células vazias (Fig. 2.16).

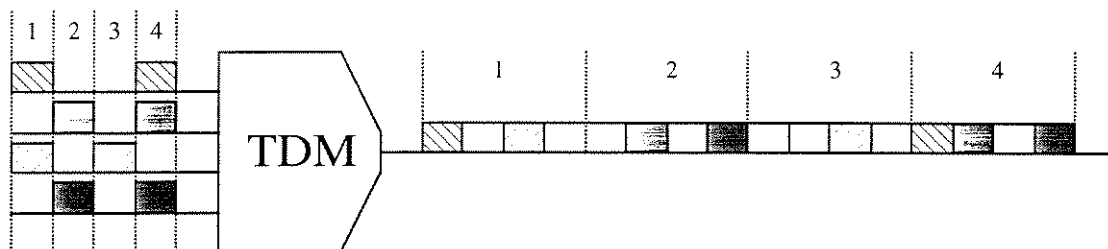


Fig. 2.16 - Processo de transmissão utilizando TDM.

Ao se utilizar a transmissão assíncrona, introduzimos um elemento de *buffer* no multiplexador de modo que ele passe a armazenar todas as células de entrada antes de liberá-las para a rede. A ordem de liberação das células depende do método de atribuição de prioridade, sendo o mais comum o "FIFO" (*First In First Out*), ou seja, a primeira célula que entra no multiplexador é a primeira a sair. O grande mérito deste método de transmissão é melhorar o aproveitamento da rede, pois ele tenta agrupar os vários "buracos" que apareceriam de maneira desordenada na transmissão TDM em um conjunto uniforme, aumentando o aproveitamento destes espaços outros serviços [97] (Fig. 2.17).

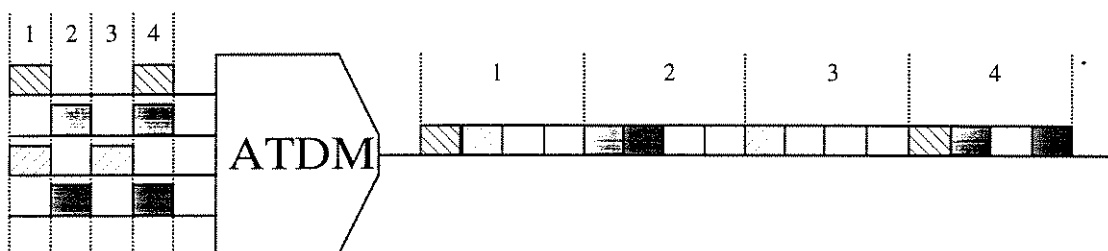


Fig. 2.17 - Processo de transmissão utilizando ATDM.

Este processo, por sua vez, tem o seu preço. Trabalhando desta forma, temos a necessidade de acumular células em um acumulador antes de transmiti-las pela rede. Ocorre que, para tráfego de dados em alta velocidade, os acumuladores podem não conseguir armazenar todas as células que nele chegam, ocorrendo transbordamento. O acumulador descarta algumas células antes de enviar o conjunto pela rede. Este problema, juntamente com o problema de sincronização causado por atrasos não uniformes que as células sofrem ao trafegar pela rede, dificultam o processo de

"montagem" do sinal no receptor e constituem um dos grandes problemas relacionados com a transmissão ATM. O protocolo para transmissão ATM envolve a divisão e controle do fluxo de dados em várias camadas [93-95]. Dentre elas destacamos as camadas de ordem superior responsáveis pelo controle da rede RDSI-FL, a camada AAL (*ATM Adaptation Layer* - camada de adaptação ATM) responsável pela interface entre os sinais de serviços enviados pelos usuários e o respectivo conjunto de células que viajarão pela rede, a camada ATM responsável pela interpretação dos cabeçalhos das células e a camada física (corpo das células ATM). A composição de uma célula ATM engloba um conjunto de 53 *bytes*, sendo os cinco primeiros destinados ao cabeçalho das células e os restantes 48 *bytes* destinados aos dados (sinal útil). O cabeçalho identifica a célula e o seu destino quando incluídas dentro do fluxo total da rede. A Figura 2.18 esclarece melhor o exposto. Nela podemos verificar a estrutura de um cabeçalho de uma célula ATM (cabeçalho tipo UNI - "interface de usuário"). Ele é composto pelos campos: VPI (identificador de caminho virtual) e VCI (identificador de canal virtual), PT (tipo de sinal útil); CLP (prioridade em relação à perda de células) e HEC (controle de erros de cabeçalhos).

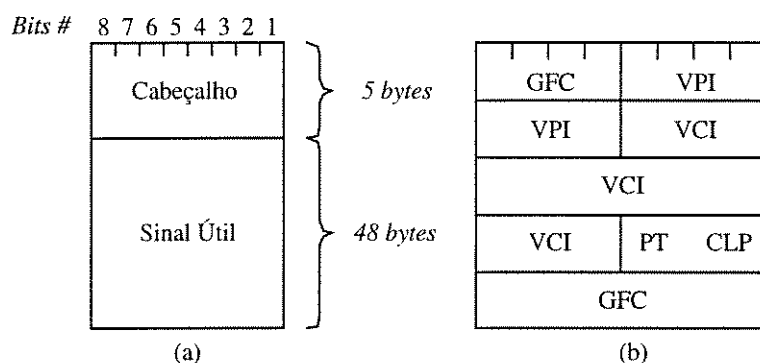


Fig. 2.18 - Estrutura de uma célula ATM.

(a): Estrutura da célula.

(b): Exemplo de um cabeçalho para a célula.

Verificamos, na estrutura do cabeçalho um campo que designa a prioridade em relação à perda de células (CLP). Em condições de descarte das células pelo elemento multiplexador, ele observa este campo, procedendo o descarte apenas daquelas células possuem baixa prioridade. Este campo identifica, também, no caso de transmissão escalonada, quais células levam os dados de alta prioridade e de baixa prioridade na estrutura de escalonamento do sinal de vídeo.

2.5 - ESCALONAMENTO DO SINAL DE VÍDEO

Como mencionamos, um dos principais problemas das redes ATM é a perda de células causada pelos picos de demanda na rede. Quando várias fontes de sinal transmitem suas informações, pode acontecer que os acumuladores localizados nos multiplexadores da rede não consigam armazenar todas as células que nele chegam, ocasionando transbordamento e, conseqüentemente, a perda de células.

Quando utilizamos códigos de comprimento variável (códigos que propiciam um uma melhor codificação do sinal elaborados através das estatísticas do mesmo), a taxa de *bits* enviada se torna variável (VBR – *variable bit rate*) e o problemas se tornam mais críticos, pois um sinal de taxa de *bits* variável necessita de sincronização para que os *bits* possam ser corretamente interpretados no receptor. Possíveis perdas de células poderão acarretar na perda de sincronismo no receptor causando erros nos quadros, os quais se propagam não só no domínio espacial da imagem mas também no domínio temporal, contribuindo para a degradação da seqüência de vídeo.

Um dos métodos que tem sido proposto para se minimizar este efeito quando se utiliza a codificação com taxa variável é segmentar o sinal de vídeo em duas ou mais camadas[98-101]. A informação considerada essencial é transmitida através de uma canal de alta prioridade (HP – *high priority*), que apresenta uma taxa de perda de células muito pequena. O restante da informação de vídeo (dita de realce) é transmitida em um canal de baixa prioridade (LP – *low priority*), onde é mais provável que ocorram perdas de células. As perdas de células afetam somente as informações de realce, o que não compromete a qualidade essencial da seqüência transmitida⁵

As células do canal de alta prioridade carregam, além da informação de vídeo necessária a reconstrução do sinal com uma qualidade “aceitável” no receptor, toda a informação de sincronismo da codificação, minimizando os erros causados pelas perdas de células.

Há vários métodos de escalonamento do sinal de vídeo. Neste trabalho consideremos dois métodos dos mais promissores nesta aplicação: a separação

⁵ Na prática existe apenas um único canal físico de transmissão e a informação de alta prioridade se distingue da informação de baixa prioridade pelo campo CLP do cabeçalho das células ATM.

espectral [98][102-104] e a requantização [98][105-107].

2.5.1 - SEPARAÇÃO ESPECTRAL

Seguindo os passos normais de uma codificação intraquadro, uma imagem, após ser dividida em blocos de 8×8 elementos, tem a sua DCT calculada para cada bloco, seguido de quantização e de codificação de entropia. O que o método da separação espectral realiza é, antes de se proceder a transmissão, separar as componentes espectrais de cada bloco 8×8 em dois conjuntos: os N primeiros elementos, segundo a ordem “zig-zag” do bloco (primeiro conjunto) e os $64 - N$ elementos restantes (segundo conjunto). Depois de separados, os dados do primeiro conjunto são quantizados e juntamente com informações de sincronismo e outras informações relevantes são codificados e enviados para o canal de alta prioridade (HP), enquanto os dados do segundo conjunto passam por um outro quantizador e após serem codificados, são enviados pelo canal de baixa prioridade (LP) (vide Fig. 2.19).

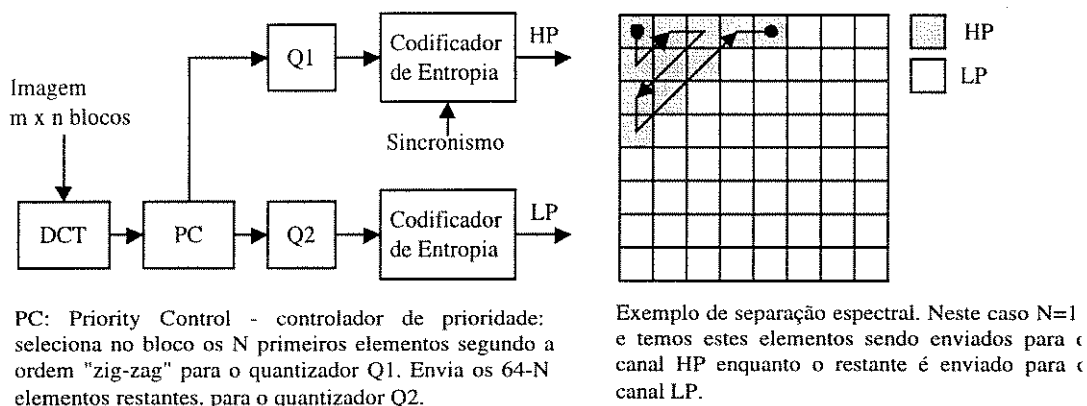


Fig. 2.19 - Esquema e exemplo de separação espectral.

Se representarmos toda a imagem por um cubo, sendo que no eixo x representamos os elementos que compõem os blocos de 0 a 63 segundo a ordem “zig-zag”; no eixo y os *bits* dos valores que estes elementos podem assumir e no eixo z os vários blocos que constituem a imagem, teremos para a separação espectral (Fig. 2.20).

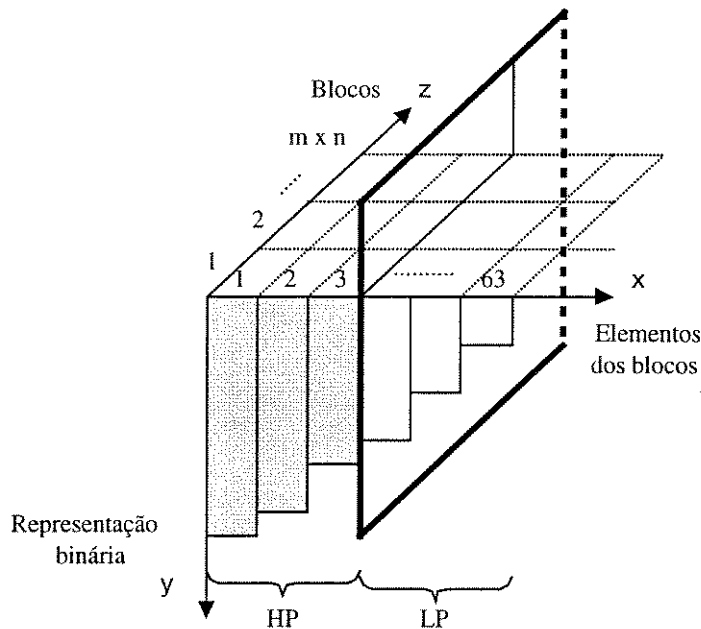


Fig. 2.20 - Representação da imagem por um cubo mostrando o processo separação espectral.

O plano (x,y) tem a forma de uma “escada”, resultado da quantização, pois quanto maior a ordem do elemento segundo o ordenamento “zig-zag”, maior é o coeficiente de quantização e, conseqüentemente, menores são os valores típicos que o elemento pode assumir. Note que a separação espectral corresponde a um corte neste cubo na vertical, onde os elementos à esquerda são codificados para o canal (HP) e os restantes (à direita) para o canal (LP).

2.5.2 - REQUANTIZAÇÃO

No processo de requantização atuamos diretamente sobre os quantizadores para proceder à segmentação do sinal. O esquema da requantização pode ser visto na Figura 2.21.

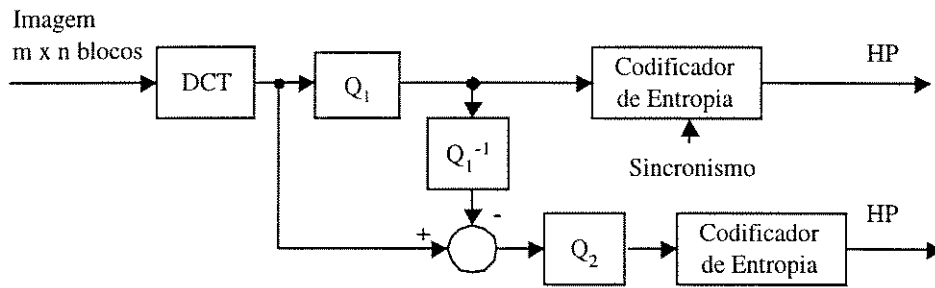


Fig. 2.21 - Esquema da requantização

No processo de requantização, cada um dos blocos da imagem transformados pela DCT são quantizados por Q_1 e juntamente com os sinais de sincronismo passam pelo codificador de entropia e formamos dados do canal de alta prioridade (HP). Os dados quantizados por Q_1 são então "desquantizados" através do processo inverso⁶ de Q_1^{-1} e subtraídos dos dados originais vindos do bloco da DCT. Este resultado, o erro de quantização relativo ao quantizador Q_1 , é então quantizado por Q_2 e codificado pelo codificador de entropia, formando o sinal do canal de baixa prioridade (LP). Atuando sobre os passos de Q_1 e Q_2 , conseguimos definir que parcela de dados é codificada em alta prioridade. Geralmente Q_1 é definido de modo a garantir um qualidade mínima "aceitável" no receptor para uma certa taxa de transmissão. O valor de Q_2 geralmente é tomado de baixo valor, de modo que a informação de realce advinda da diferença entre os dados da DCT e de Q_1^{-1} não se perca em demasia, garantindo melhor qualidade à imagem. Podemos ilustrar o que foi dito através de um exemplo. Suponhamos que se deseja enviar um dado de valor 42,3 utilizando requantização com $Q_1 = 10$ e $Q_2 = 1$. Temos então:

- Valor enviado pelo canal HP: 42,3 quantizado por 10 = 4
- Valor enviado pelo canal LP: $(42,3 - 4 \times 10)$ quantizado por 1 = 2

Assim, o canal de alta prioridade será responsável pela transmissão da maior parcela da informação e o canal de baixa prioridade carregará a informação restante (realce).

Em termos de nossa representação da imagem por um cubo, teríamos para o caso da requantização o esquema da Figura 2.22. Nesta representação, notamos os *bits* mais significativos (parte escurecida) e os *bits* menos significativos (parte não

⁶ Q^{-1} simboliza o processo inverso da requantização e não a função inversa de Q (pois ela não existe).

escurecida). O corte de separação é feito na diagonal e não na vertical como no processo de separação espectral.

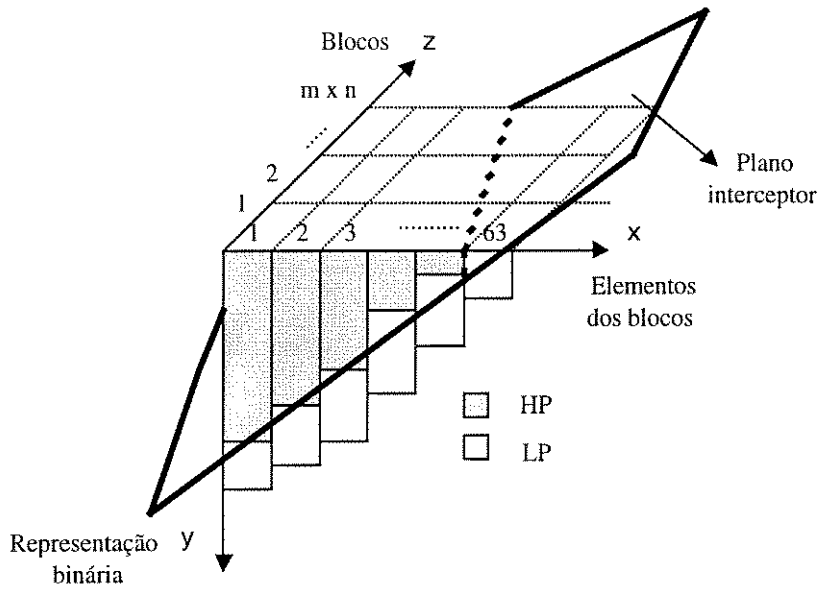


Fig. 2.22 - Representação da imagem por um cubo mostrando o processo requantização.

CAPÍTULO 3

O AMBIENTE DE SIMULAÇÃO

Apresentamos, neste capítulo, o ambiente de simulação utilizado na obtenção da parte prática da pesquisa. Entendemos este ambiente de simulação como um conjunto de programas elaborados em linguagem C composto por três módulos principais:

- (a) Módulo de codificação (decodificação) de seqüências de vídeo,
- (b) Módulo de compactação (descompactação) de dados em células ATM,
- (c) Módulo de modelamento de perda de células no canal de transmissão.

O primeiro módulo inclui um codificador (decodificador) de seqüências de vídeo que utiliza a teoria desenvolvida no Capítulo 2. Ele atua na redução das redundâncias temporal e espacial da seqüência de vídeo. Uma vez reduzidas estas redundâncias, a seqüência de vídeo se encontra comprimida e os dados serão “empacotados” em células ATM para serem enviados pelo canal. Esta tarefa é executada pelo segundo módulo. Neste canal, poderão ocorrer perdas de células, principalmente devido a um excesso de demanda em certos instantes de tempo. O terceiro módulo é responsável por gerar esta perda de células como função das características do canal. Uma vez recebidas, as células são “desempacotadas” e decodificadas, e sua qualidade e outros dados relevantes são levantados para análise. Notamos que este ambiente tende a simular, ainda que de forma simplificada, todo o processo de transmissão e recepção de uma seqüência de vídeo através de uma canal ATM, permitindo-nos analisar o problema inicialmente proposto e estudar possíveis soluções para o mesmo. O conjunto de operações é representado na Figura 3.1.

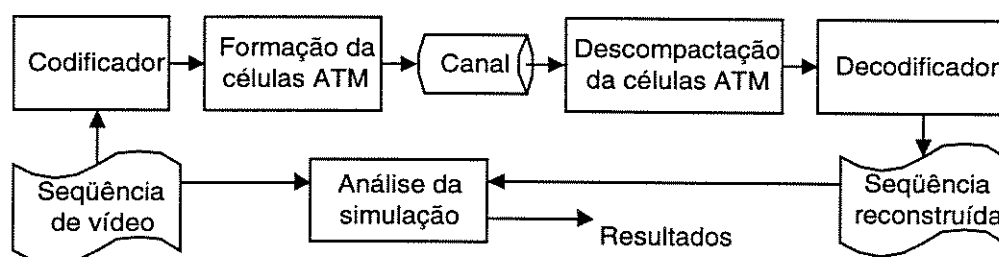


Fig. 3.1 - Diagrama mostrando os diversos módulos do ambiente de simulação.

A seguir, descrevemos sucintamente cada módulo e suas principais características.

3.1 - MÓDULO 1: CODIFICADOR E DECODIFICADOR PARA SEQUÊNCIAS DE VÍDEO.

Um esquema do módulo codificador pode ser visualizado na Figura 3.2:

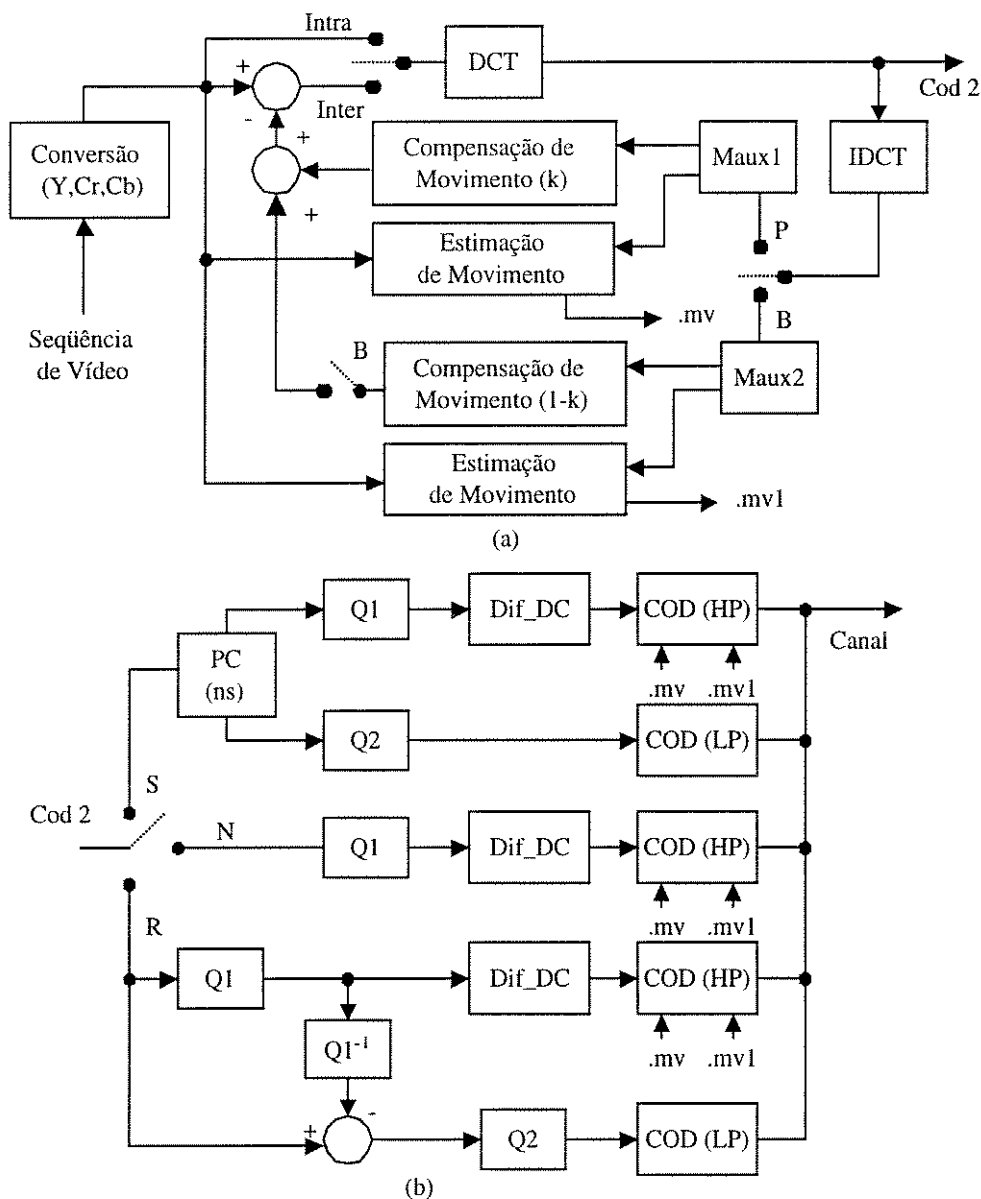


Fig. 3.2 -Codificador para a extração da redundância espacial e temporal
(a): Codificação Preditiva + DCT (Cod1).
(b): Quantizador e Codificação de Entropia (Cod2).

Para proceder a codificação do sinal de vídeo, dividimos a informação em quatro níveis, seguindo a metodologia empregada no padrão MPEG [52][53][101].

- (a) Bloco: É a menor unidade de codificação utilizada para a redução da redundância espacial do quadro. É feito de um conjunto de 8×8 *pixels*, podendo ser composto por informações de: luminância (Y) e crominâncias (Cr e Cb).
- (b) Macrobloco: É a unidade utilizada para a codificação MPEG, visando a redução da redundância temporal. Um macrobloco é composto por um conjunto de 16×16 *pixels* do sinal de luminância (4 blocos de 8×8 *pixels*) e dois blocos de 8×8 *pixels* dos sinais de crominância Cr e Cb, já que estes são sub-amostrados por um fator de dois em cada direção.
- (c) Fatia: Uma fatia é uma linha de macroblocos da imagem na direção horizontal utilizada no esquema de codificação MPEG. As operações com blocos e macroblocos são efetuadas ao longo de uma fatia, fatia a fatia por todos os quadros da seqüência de vídeo. Um exemplo é a codificação diferencial do elemento DC dos blocos.. No começo de cada nova fatia, o preditor é zerado, de modo que as fatias são tratadas de modo independente.
- (d) Quadro: É a unidade básica que corresponde a uma imagem da seqüência de vídeo. É definida em termos do número de *pixels* nas direções horizontal e vertical.

Comentamos agora o esquema visto na Figura 32a (codificador para extrair a redundância temporal). Notamos duas maneira de se proceder a codificação: codificação intraquadros e codificação interquadros.

3.1.1 - CODIFICAÇÃO INTRAQUADRO (I)

Neste modo de codificação, o quadro colorido, após sofrer a decomposição (R,G,B) para (Y,Cr,Cb), é dividido em blocos donde e transformado pela DCT. As informações seguem para o segundo codificador (Cod 2), onde se processará a quantização e a codificação de entropia. O bloco que executa a DCT foi implementado com um algoritmo de execução rápida de forma a agilizar o programa de codificação. Este algoritmo e a sua formulação são apresentados no Apêndice 7.

3.1.2 - CODIFICAÇÃO INTERQUADRO

Há basicamente dois tipos: a codificação interquadros com predição e a com interpolação.

a) Codificação com Predição (P):

Neste modo, é efetuada uma estimação de movimento com a obtenção do vetor de movimento associado, e uma compensação de movimento, sendo gerado um sinal de erro que, após codificado, é transmitido. O registrador Maux representa uma memória utilizada para armazenar o quadro anterior, necessário para o processamento.

Convém ressaltar algumas peculiaridades deste método, utilizadas pelo programa de codificação:

1) Notamos pela Fig. 3.2a que o quadro de referência é equivalente ao erro de quantização produzido pela diferença $Q - Q^{-1}$. Esta operação se faz necessária pelo casamento que deve existir entre transmissor e receptor. Como no receptor temos acesso apenas à imagem quantizada, é necessário, no transmissor, proceder a predição com esta referência, cuja qualidade é definida pelo bloco quantizador. Casando a operação de predição entre transmissor e receptor, melhoramos a qualidade da sequência reproduzida

2) Tendo obtido a imagem de referência, ocorre o processo de estimação de movimento, através da divisão do quadro em macroblocos, compostos por regiões de 16×16 *pixels* para as componentes Y e 2 blocos de 8×8 *pixels* para as componentes Cr e Cb. Efetuamos o cálculo do vetor de movimento (vide Cap2) utilizando somente a componente Y, por ser esta componente a mais importante para a representação da imagem. Utilizamos o mesmo vetor de movimento para a compensação de movimento das componentes Cr e Cb, reduzindo assim a quantidade de informação enviada.

O vetor de movimento apresenta duas componentes necessárias para a localização do macrobloco dentro do quadro: x (na direção horizontal) e y (na vertical). Os macroblocos são lidos sequencialmente, fatia a fatia por todo o quadro, e os vetores de movimento calculados passam a ser referenciados às posições dos macroblocos lidos. Como o vetor de movimento é calculado em torno de uma área S

na posição do macrobloco (área limitada com deslocamentos de até 16 na horizontal e na vertical em relação ao contorno do macrobloco), esta faixa de valores passa a ser única necessária para a especificação do vetor de movimento. No Apêndice 5, apresenta-se a tabela de codificação para os vetores de movimento.

O cálculo do vetor de movimento também depende de alguns parâmetros, que podem ser alterados no programa:

- a) A área de pesquisa S pode ser escolhida entre os valores de 0 a 16. Valores elevados resultam em erros de predição/interpolação menores, mas aumentam o tempo de computação.
 - b) O método de busca pode ser escolhido como exaustivo (busca por todos os *pixels* da área S) ou por passos logarítmicos, permitindo um compromisso entre o erro de estimativa e esforço computacional exigido.
 - c) A função de custo utilizada define o critério de obtenção do macrobloco "ótimo", podendo-se optar pela correlação, pelo erro quadrático médio, ou pelo erro absoluto médio entre os macroblocos.
- 3) Por último, efetuamos a compensação de movimento através da diferença entre os macroblocos (referência e o obtido pelo vetor de movimento). O parâmetro k (vide Fig. 3.2a), que se situa na faixa de $k \in [0,1]$, atua multiplicando todo o macrobloco obtido pelo vetor de movimento antes de proceder a diferença com o macrobloco de referência. O valor de k , no caso da interpolação, permite a escolha de qual parcela deve contribuir mais para a formação do macrobloco de erro. No caso da predição o valor de k é igual a 1.

b) Codificação com Interpolação(B):

Este modo de codificação utiliza tanto as informações do quadro anterior como posterior (armazenados nas memórias auxiliares $Maux1$ e $Maux2$) além do quadro que se deseja codificar para a geração dos macroblocos de erro. É efetuada uma estimação de movimento com relação a cada quadro e a compensação é realizada, gerando-se os macroblocos de referência através da combinação linear dos macroblocos deslocados, ponderados pelos parâmetros k e $1-k$ ($k \in [0,1]$).

Esta característica de se utilizar tanto os macroblocos do quadro anterior como posterior para o cálculo dos macroblocos de erro confere a interpolação (também chamada de predição bidirecional) algumas vantagens em relação à predição. Podemos destacar os seguintes aspectos:

- a) Diminuição dos erros gerados em virtude da maior quantidade de informação sobre a seqüência de vídeo.
- b) Redução dos erros no pontos de mudança de cenas das seqüências de vídeo (como, por exemplo, o primeiro quadro de um comercial), já que a baixa correlação entre este quadro e o anterior é compensada pela alta correlação com o quadro posterior.

Dado uma seqüência de vídeo de tamanho N quadros, utilizamos um parâmetro para a sua codificação denominado GOP (*Group of Pictures*) que define o tipo de codificação (I, P ou B) para cada quadro da seqüência. Como só podemos efetuar um predição se já possuímos um quadro anterior já codificado e uma interpolação se já possuímos, além do quadro anterior, o quadro posterior também já codificado, um GOP deve sempre começar por um quadro do tipo I. Também, quadros do tipo P só podem vir depois de quadros do tipo I ou P e quadros do tipo B estarão necessariamente contidos entre quadros do tipo I ou P. Um exemplo com GOP igual a 9 pode ser visto na Figura 3.3.

A seqüência de processamento segue a ordem:

- 1º Quadro (1): Tipo **I**.
- 2º Quadro (5): Tipo **P** (predição com o quadro 1).
- 3º Quadro (2): Tipo **B** (interpolação com os quadros 1 e 4).
- 4º Quadro (3): Tipo **B** (interpolação com os quadros 1 e 4).
- 5º Quadro (4): Tipo **B** (interpolação com os quadros 1 e 4).
- 6º Quadro (9): Tipo **P** (predição com o quadros 5).
- 7º Quadro (6): Tipo **B** (interpolação com os quadros 5 e 9).

8º Quadro (7): Tipo **B** (interpolação com os quadros 5 e 9).

9º Quadro (8): Tipo **B** (interpolação com os quadros 5 e 9).

10º Quadro (10): Tipo **I** (novo GOP).

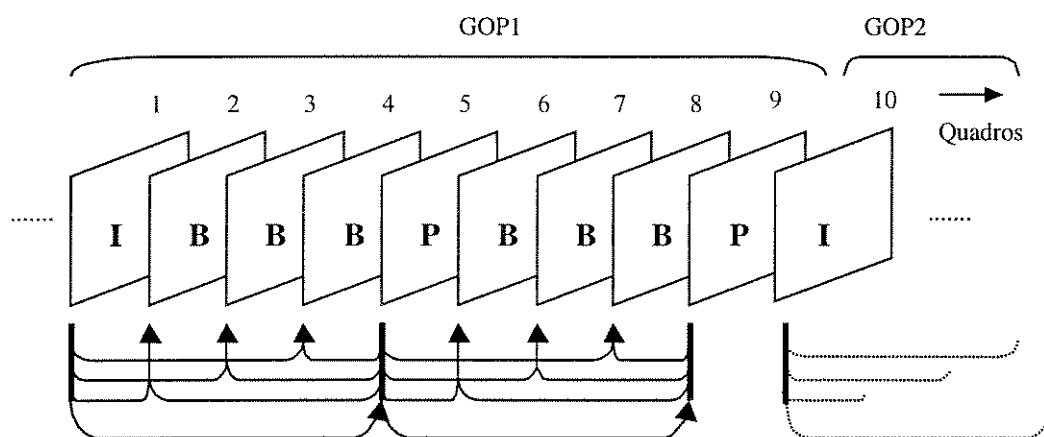


Fig. 3.3 - Exemplo de codificação com GOP = 9.

Para grandes valores de GOP (muitos quadros do tipo B e P), temos uma maior compressão da sequência da vídeo, mas, em contrapartida, tornamos a sequência transmitida mais vulnerável a erros, pois estes, se ocorrerem, se propagarão por todo GOP até um quadro do tipo I. Sendo este um quadro de referência, ele bloquearia a propagação do erro. Assim, o tamanho do GOP deve levar em conta estas considerações. No programa desenvolvido, o tamanho do GOP pode ser variado para se obter o seu melhor valor.

Ocorre muitas vezes que, definindo um GOP, tentamos executar uma predição ou interpolação para todos os macroblocos do quadro e, para alguns destes, obtemos erros muito grandes (suponha, por exemplo, uma sequência de vídeo onde os quadros sofrem mudanças muito rápidas). Assim, não há vantagem em se proceder estas codificações, pois não obteremos uma taxa de compressão significativa e teremos o problema da propagação de erro devido à possível perda de células no canal. Nestes casos, procede-se a uma codificação do tipo I (intraquadro) para os macroblocos cujo erro supere um certo limiar definido. A taxa de transmissão não muda significativamente e reduz-se o potencial de propagação de erro dentro do GOP.

Para quadros do tipo B, pode ocorrer em alguns casos que o erro gerado seja maior que o gerado utilizando-se predição (tanto em relação ao quadro anterior como o

posterior). Logo, quando se efetua uma interpolação, o erro gerado é comparado com os erros de predição obtidos utilizando-se a imagem posterior e anterior antes de se decidir qual esquema deve ser adotado¹. O fluxograma da Figura 3.4 esclarece o procedimento:

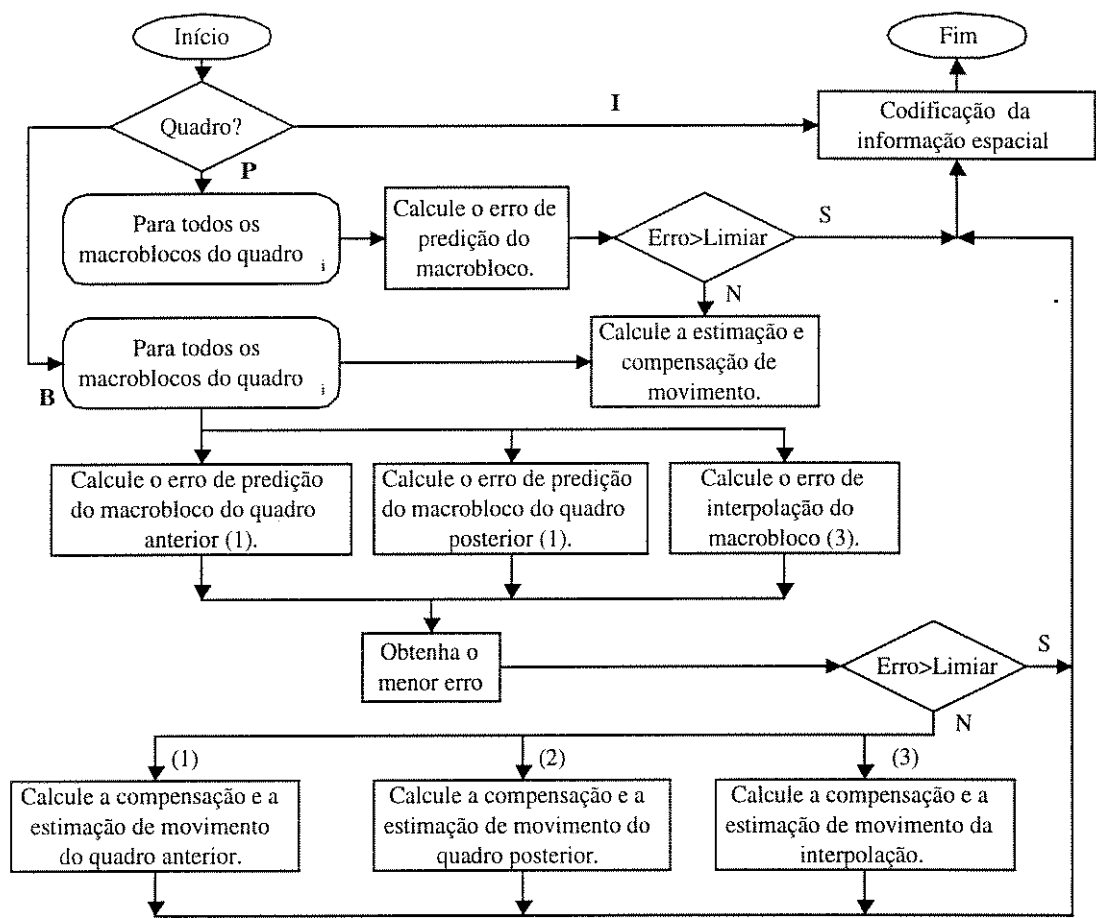


Fig. 3.4 - Fluxograma de codificação dos macroblocos de um quadro.

No esquema da Fig. 3.2b podemos visualizar a segunda etapa do codificador com os blocos destinados a separação em camadas. A seleção é obtida através da chave seletora que permite os seguintes modos:

S: Separação espectral: Consiste num bloco PC (*priority control*) com parâmetro ns, o qual permite separar em dois conjuntos complementares os primeiros ns elementos de cada bloco segundo a ordem "zig-zag" (ou uma outra ordem definida pelo usuário do programa) dos 64 - ns elementos restantes. Esta separação se dá logo após ser efetuada

¹ Falar em predição com base no quadro posterior constitui-se em um abuso de linguagem, usado aqui por conveniência de notação.

a DCT (componentes espectrais) e os dados são levados por caminhos diferentes para serem codificados formando os dados de alta e baixa prioridade. Os dados de alta prioridade sofrem uma codificação diferencial com quantização definida por Q1, seguida de uma codificação de entropia realizada pelo codificador de alta prioridade (HP). Os dados de baixa prioridade sofrem uma quantização por outra matriz (Q2) e seguem para o codificador de entropia de baixa prioridade (LP). No processo, as componentes de luminância (Y) e cromaticidade (Cr e Cb) são repartidas pelo mesmo parâmetro ns.

N: Modo normal: Constitui-se na extração da redundância espacial seguindo o esquema de codificação JPEG: DCT → Quantização (Q1) → Codificação diferencial (DIF_DC) → Codificação de entropia (utilizando o codificador de alta prioridade (HP)).

R: Requantização: Consiste na separação dos valores dos elementos dos blocos através de níveis de quantização. Um bloco, após sofrer a DCT, tem os seus elementos codificados diferencialmente e quantizados por um quantizador Q1, sofrem a codificação de entropia pelo codificador de alta prioridade (HP). Os elementos quantizados por Q1 sofrem um processo de "quantização inversa" e da sua diferença com os dados originais são gerados elementos que, quantizados pelo quantizador Q2, são codificados pelo codificador de entropia de baixa prioridade (LP) (Fig. 3.2a). O passo do quantizador Q1 é geralmente feito maior que o do quantizador Q2 de modo que os dados de alta prioridade carregam a informação grosseira dos elementos. Os dados de baixa prioridade constituem a informação de detalhe ou realce. Sendo o elemento DC o mais importante do bloco, o mesmo não sofre o processo de requantização e é codificado em alta prioridade.

Passemos à descrição dos codificadores de entropia para entendermos melhor os resultados das simulações.

COD(HP):

A codificação dos dados para o canal de alta prioridade se faz percorrendo os macroblocos que compõem uma fatia, fatia a fatia, por todo o quadro. No processo de codificação, associamos aos dados a seguinte palavra código (Fig. 3.5):

Nmb	Vmb	Ibo	Bo	Ib1	B1	Ib2	B2	Ib3	B3	Icr	Cr	Icb	Cb
-----	-----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----

Fig. 3.5 - Palavra código associada à codificação de alta prioridade.

O primeiro campo (Nmb) indica o número de macroblocos nulos lidos durante o processo de codificação. O teste de verificação de um macrobloco é nulo abrange as três componentes da imagem (Y, Cr e Cb) e pode indicar um número máximo de 45 macroblocos, que corresponde a uma fatia da imagem de tamanho igual a 720 *pixels*, dimensão horizontal máxima da imagem que pode ser processada pelo programa. No Apêndice 6 temos o código para este campo.

O segundo campo (Vmb) é composto pelos vetores de movimento associados aos macroblocos lidos. A cada macrobloco lido associamos uma das seguintes opções: nenhum vetor de movimento para codificação do tipo I (interquadro), um vetor de movimento para a codificação do tipo P (predita) e dois vetores de movimento para codificação do tipo B (bidirecional).

Quando um macrobloco não é nulo, procedemos sua codificação interna, analisando cada uma das suas componentes, quais sejam: quatro blocos de luminância (Y) (associados aos campos da palavra código Ib(0..3) e B(0..3)) e dois blocos de crominância, Cr e Cb, (com campos associados Icr, Cr e Icb e Cb). Os campos Ib(0..3), Icr ou Icb da palavra código representam *flags*, que indicam se o bloco em questão é nulo ou não. Não sendo nulo (*flag* = 0), ele será codificado (leitura "zig-zag" seguida de codificação de Huffman) e sua codificação armazenada no respectivo campo B(0..3), Cr ou Cb. Sendo o bloco nulo, a *flag* assume valor 1, não ocorrendo a codificação do bloco. Neste caso, o respectivo campo (B(0..3), Cr ou Cb) não existirá.

A codificação de alta prioridade (HP) será formada por um conjunto destas palavras código aplicadas a todos os quadros da sequência processada. Devemos notar que o sincronismo requerido na leitura dos macroblocos e na codificação de Huffman utilizados, etc.) tende a tornar o sistema mais vulnerável a perdas de células que possam ocorrer no canal de transmissão.

COD(LP):

Como os dados de baixa prioridade podem sofrer uma taxa de perdas maior ao trafegarem pelo canal de transmissão, é necessário aumentar a robustez na sua codificação.

A palavra código associada a codificação é dada por (Fig. 3.6):

(Xm,Yn)	Ibo	Bo	Ib1	B1	Ib2	B2	Ib3	B3	Icr	Cr	Icb	Cb
---------	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----

Fig. 3.6 - Palavra código associada à codificação de baixa prioridade.

O primeiro campo (Xm,Yn) indica a posição de um macrobloco não nulo dentro do quadro com relação ao início do mesmo (posição (0,0)). Esta posição é codificada de modo a destruir o sincronismo entre os macroblocos no processo de leitura de cada fatia do quadro. Logo, possíveis perdas de macroblocos na transmissão não afetarão a decodificação dos macroblocos recebidos dentro do quadro, aumentando a robustez da codificação. A robustez poderá também ser aumentada no momento de formação das células ATM, com um "empacotamento" eficiente dos dados, como será visto mais a frente.

Os outros campos seguem a mesma metodologia aplicada à codificação de dados de alta prioridade.

O projeto do decodificador para as seqüências de vídeo segue os passos inversos aos do codificador aqui exposto, sendo facilmente deduzido a partir deste.

3.2 - MÓDULO 2: COMPACTAÇÃO E DESCOMPACTAÇÃO DOS DADOS EM CÉLULAS ATM

Após codificados, os dados necessitam ser "empacotados", ou seja, repartidos em células para serem transmitidos através do canal. As células utilizadas apresentam 48 *bytes* de tamanho útil para alocação de dados e sinais de sincronismo. Estas células tendem a se assemelhar das células ATM [92-95] que contém 53 *bytes* de comprimento, onde os 5 primeiros *bytes* formam o cabeçalho da células (com dados para a sua identificação) os 48 *bytes* restantes carregam a informação de codificação da imagem. Adotamos um modelo simplificado e ignoramos os cinco primeiros *bytes*,

trabalhando com os 48 restantes. Células ATM podem apresentar também dentro dos seus 48 *bytes* uma codificação de *Reed-Solomon* (RS) [42], que geralmente ocupa de 5 a 15% do seu tamanho. Esta codificação tem como objetivo, proteger o conteúdo das células contra possíveis erros de canal. Utilizamos este possível espaço reservado para a alocação da informação codificada da imagem, já que admitimos um modelo de perda total de células. As células são formadas por dados codificados através de códigos de Huffman, sendo então sincronizadas, e um dos nossos problemas é tentar desfazer esta sincronização, de modo que os erros devido às perdas de células não se propaguem. De modo a minimizar estes efeitos, adotaremos o método de codificação mostrado na Figura 3.7.

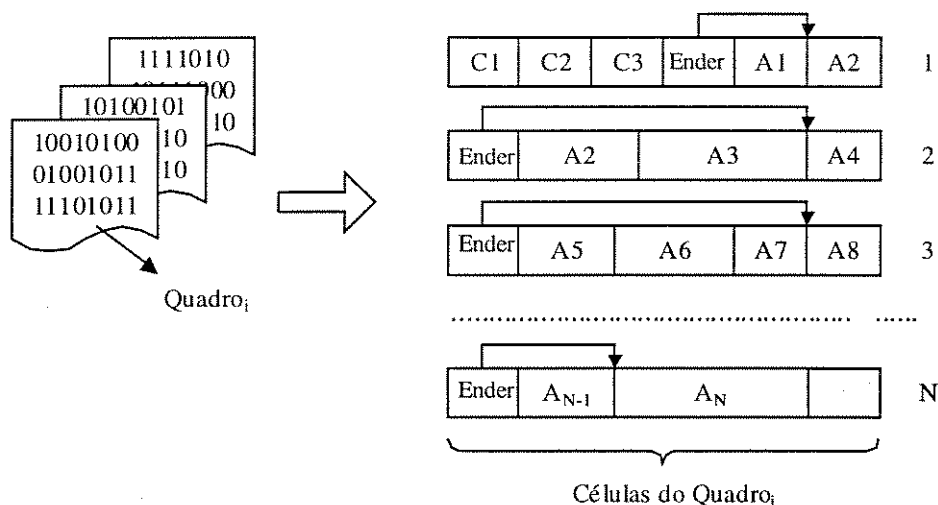


Fig. 3.7 - Processo de formação das células a partir dos dados da seqüência de vídeo.

Cada quadro da imagem será codificado em um conjunto de células. A primeira célula deste conjunto representará a célula de cabeçalho, composta por três campos C_1 , C_2 e C_3 , com as seguintes características:

C_1 : Identifica uma dada célula como sendo de cabeçalho ou uma célula comum de dados. Este campo indica o início de cada novo quadro da seqüência, sendo composto por três *bits*.

C_2 : Identifica o número de células utilizadas para a codificação do quadro em questão. Utiliza-se um campo fixo de 20 *bits*, permitindo um número máximo de 2^{20} células por quadro.

C_3 : identifica o número do quadro codificado segundo a ordem de exposição da seqüência. É formado por um código DCPM de 6 *bits* com capacidade de codificação de 64 quadros.

Seguindo estes campos, encontra-se um campo denominado Ender (campo de endereço) e os campos de dados $A_1, A_2, \dots, A_N$, onde os dados da seqüência de vídeo são alocados. Estes campos são comuns ao restante das células, diferindo no posicionamento do campo de endereço da célula de cabeçalho. Nas células de dados ele sempre ocorrerá no início das mesmas (Fig. 3.5). A função do campo de endereço é indicar, dentro de cada célula, o início da codificação do último macrobloco que tem seu código ou parte dele armazenado na célula. Assim, no caso de perdas de células, conseguimos ressincronizar o processo de decodificação, lendo na célula recebida o início do próximo conjunto de dados válidos através do campo de endereço da mesma, aumentando a robustez dos dados a perdas de células. Este campo indicará um dos $48 \times 8 = 384$ possíveis *bits* dentro da célula onde será iniciado o código do último macrobloco, necessitando o campo de endereço de $\lceil \log_2(384) \rceil = 9$ *bits*.

Os campos C_1, C_2, C_3 e *End* representam uma certa sobrecarga na taxa de transmissão, já que redundâncias são acrescentadas para proteger os dados das perdas de células. Esta sobrecarga vale

$$\text{Sobrecarga} = \frac{\text{Número de bits necessários para codificar a seqüência de vídeo em células}}{\text{Número de bits necessários para codificar a seqüência de vídeo}}$$

Portanto

$$\text{Sobrecarga} = \frac{C_1 + C_2 + C_3 + \frac{n \times 384}{384 - \text{End}}}{n} \quad (3.1)$$

onde n é o número de *bits* necessários para codificar a seqüência de vídeo e C_1, C_2, C_3 e *End* são os números de *bits* destes campos.

Para n grande, (3.1) se transforma em $\text{Sobrecarga} \cong 384/375 = 1,024$, ou seja, temos uma sobrecarga de 2,4% para a codificação dos dados em células.

3.3 - MÓDULO 3: MODELAGEM DAS PERDAS DE CÉLULAS NO CANAL DE TRANSMISSÃO

A quantidade de células perdidas no canal de transmissão é uma função do número total de células que trafegam no mesmo em dado instante. Isto decorre, principalmente, devido à capacidade finita de armazenamento dos acumuladores (*buffers*) presentes nos multiplexadores da rede. O multiplexador recebe, em sua entrada, vários canais com uma taxa de células variáveis e fornece uma única saída, limitada a uma taxa de células máxima. Se a velocidade de chegada das células for superior a velocidade de saída, as mesmas são armazenadas nos acumuladores internos do multiplexador, que absorvem os picos de sobrecarga. Se os picos forem de longa duração, os acumuladores transbordarão, ocorrendo o descarte de células.

Para a modelagem do processo de perdas, suporemos o multiplexador recebendo continuamente células advindas no nosso codificador (Canal 1) superpostas a células advindas de outros canais (agrupadas por motivo de simplicidade no Canal 2). De modo a simplificarmos o nosso modelo, admitiremos que já estamos ocupando plenamente o acumulador, de modo que a perda de células se dará toda vez que o tráfego agregado (Canal 1 + Canal 2) superar um certo limiar, como nos mostra a Figura 3.8. Este modelo constitui-se numa simplificação, visto que não representa o efeito da filtragem de oscilações rápidas de tráfego produzido pelos acumuladores. Os resultados são, portanto, uma aproximação da situação real.

No eixo horizontal temos os vários quadros que compõem a sequência e no eixo vertical o número de células por quadro. Quando o número de células totais for superior ao limiar, ocorrerá um descarte de células da quantidade B atingindo os quadros compreendidos na região A, como mostrado na Figura 3.8. Esta perda de células atingirá somente as células do canal de baixa prioridade, já que, por hipótese, as células do canal de alta prioridade estão isentas de descartes (para as células de baixa prioridade, ajusta-se o *bit* CLP no cabeçalho das células ATM). Ambos os canais sofrerão com descarte de células, sendo mais penalizado o canal que mais contribuiu para que o limiar fosse atingido. Consideremos o seguinte exemplo:

Canal 1: 100 células sendo 50 de alta prioridade e 50 de baixa prioridade.

Canal 2: 400 células sendo 100 de alta prioridade e 300 de baixa prioridade.

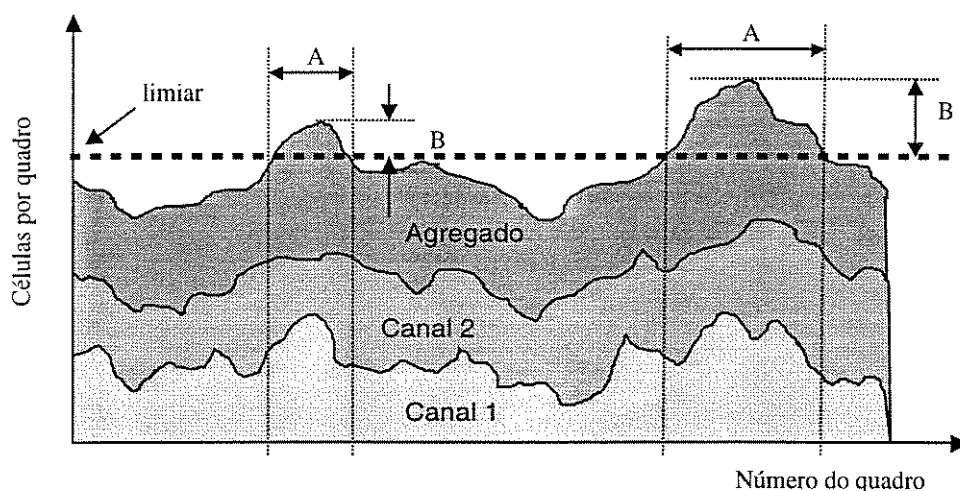


Fig. 3.8 - Modelo para perda de células

A: Região onde as células são perdidas.

B: Quantidade de células perdidas.

Canal 1: Canal com células geradas pelo nosso codificador.

Canal 2: Canal com células agrupadas dos outros codificadores

Agregado: Soma das células do Canal 1 e Canal 2.

Para um limiar de 400 perdemos um total de $(500 - 400) = 100$ células nas proporções

$100 \times (100/500) = 20$ células do Canal 1 e

$100 \times (400/500) = 80$ células do Canal 2.

Portanto, os descartes são feitos proporcionalmente ao número de células de cada canal. Das células totais perdidas do Canal 1 procederemos apenas ao descarte das células codificadas em baixa prioridade, como já comentamos. Portanto, temos

Canal 1: Perda de 20 células de baixa prioridade, restando 50 células de alta prioridade e 30 células de alta prioridade

Canal 2: Perda de 80 células de baixa prioridade, restando 100 células de alta prioridade e 220 células de baixa prioridade

Caso o nível de perdas supere o número de células disponíveis codificadas em baixa prioridade no Canal 1, procedemos ao seu descarte total e a parcela restante será retirada dos dados de baixa prioridade do Canal 2, suposto conter um número suficiente de células para esta operação. Determinada a quantidade de células perdidas e a que quadro elas pertencem, um outro problema associado ao processo é determinar quais células do quadro serão extraídas no processo (por exemplo, se são as primeiras que entrarão no *acumulador* ou as últimas). Adotamos aqui a retirada de células de forma aleatória com distribuição de probabilidade uniforme de modo a "diluir" o erro dentro do quadro, atenuando sua percepção. Definido o método e o modo de descarte,

resta-nos determinar o limiar que define a capacidade do acumulador. O valor do limiar é obtido indiretamente após estipulado um certo nível de perdas para o Canal 1. Assim, de posse do valor total agregado do tráfego, o limiar é calculado de modo que uma certa taxa de perda de células no Canal 1 seja atingida. Deste modo podemos analisar o comportamento do processo de codificação - transmissão - decodificação sujeito aos vários níveis de perda no canal de transmissão. Os dados do Canal 2 utilizados no modelo de perdas são obtidos através da combinação aleatória de vários trechos de seqüências codificadas segundo o padrão MPEG-II, como será visto adiante

Capítulo 4

ANÁLISE DOS RESULTADOS

Neste capítulo, analisamos os resultados das simulações de perda de células a sua influência na qualidade das seqüências de vídeo. Adotamos a suposição que a quantidade de células gerada por nosso codificador não afeta significativamente o tráfego da rede de modo que o nível de perda de células seja independente do tipo de codificação utilizada. O problema de perda de células é analisado através da comparação entre as curvas de taxa x distorção que definem a eficiência do sistema de comunicação. Obtendo-se estas curvas para os vários modos de codificação e níveis de perda de células, podemos além de estudar o desempenho e a robustez dos sistemas de codificação, obter o esquema mais apropriado para a transmissão

4.1 - ANÁLISE DO CODIFICADOR E DECODIFICADOR

Antes de prosseguirmos com a análise dos resultados esclarecemos alguns termos utilizados:

1. **RMSE (Root Mean Square Error):** Este termo se refere à medida da raiz quadrada do erro quadrático médio entre as seqüências de vídeo original e a processada. Seu valor é obtido extraíndo-se a raiz quadrada da média dos erros quadráticos médios entres quadros equivalentes. O erro quadrático médio entre os quadros é calculado pela média entre as diferenças quadráticas das componentes (R,G,B) da seqüência original e processada. As equações (4.1), (4.2) e (4.3)

$$RMSE = \sqrt{\frac{\sum_{k=1}^{Nq} Er_k^2}{Nq}}, \quad (4.1)$$

$$Er^2 = \frac{\sum_{i=1}^{Nh} \sum_{j=1}^{Nv} D^2 p_{(i,j)}}{NhNv}, \quad (4.2)$$

$$D^2 p_{(i,j)} = \frac{(R_{(i,j)}^1 - R_{(i,j)}^0)^2 + (G_{(i,j)}^1 - G_{(i,j)}^0)^2 + (B_{(i,j)}^1 - B_{(i,j)}^0)^2}{3}. \quad (4.3)$$

esclarecem a definição.

Nestas equações, tem-se:

Er_k : Erro quadrático médio entre quadros equivalentes.

Nq : Número de quadros da seqüência.

$Dp_{(i,j)}$: Diferença quadrática entre componentes (R,G e B) do quadros nas posições (i,j).

Nh : Número de *pixels* na direção horizontal do quadro.

Nv : Número de *pixels* na direção vertical do quadro.

O problema de avaliação da qualidade de uma seqüência de vídeo passa por aspectos subjetivos, pois é dependente das pessoas que a avaliam [109]. Neste trabalho adotamos uma avaliação objetiva da qualidade, baseada no valor do RMSE, para maior simplicidade

2. $R^1_{(i,j)}$, $R^0_{(i,j)}$: valores da componente (i,j) vermelha nos quadros original (0) e processado (1) (notação análoga para G^1 , G^0 e B^1 e B^0).
3. **Taxa**: Informa a quantidade média de *bits* necessários para se representar um dado *pixel* de um quadro. Seu valor é dado em *bits/pixel* (b/p), tendo um valor de 24 b/p para quadros não comprimidos (oito *bits* para cada componente R,G e B).

As simulações foram realizadas com três seqüências de vídeo intituladas "FLOWER GARDEN", "KIEL" e "MOBILE CALENDAR". Estas seqüências são formadas por imagens naturais coloridas com resolução de 720 x 480 *pixels*. Foram escolhidas pelo nível de detalhamento de cores e texturas e pelos vários tipos de movimento dos elementos das cenas; indo desde um simples movimento translacional horizontal ("FLOWER"), passando por movimentos translacionais verticais e horizontais ("MOBILE") até movimentos do tipo "zoom", ou seja, distanciamento e aproximação da câmara de filmagem ("KIEL"). As simulações foram realizadas para um conjunto de 33 quadros de cada seqüência e as variáveis do codificador e decodificador foram ajustadas de modo a se obter uma taxa de transmissão compatível com o padrão MPEG II, que se situa na faixa de 5 a 10 *Mbit/s*. Para este ajuste, adotamos os seguintes parâmetros no codificador:

- (a) Um GOP (*Group of Picture*) tipo IBPB, repetido para os 28 quadros restante (terminando com um quadro do tipo I). O comprimento do GOP foi escolhido de modo se minimizar uma possível propagação de erros. Sua composição foi obtida

de modo empírico, através de várias simulações, obtendo-se a menor distorção para uma dada taxa.

- (b) Filtros de conversão $(R,G,B) \mapsto (Y,Cr,Cb)$ adotados pelo CPqD nos trabalhos de codificação de imagens.
- (c) Matrizes de quantização do tipo mpeg-intra (componentes de luminância) e mpeg-inter (componentes de croma) no Apêndice 1. A escolha destas matrizes foi resultado da comparação das curvas de taxa x distorção para várias simulações. A matriz de quantização caracteriza os passos de quantização usados para cada componente da DCT do bloco considerado. Para uma alocação de *bits* eficiente [110], esta matriz deve equilibrar o erro de quantização observado nas várias componentes da DCT. Para o sinal de luminância, isto equivale a associar passos de quantização menores para as componentes de frequência mais baixas, para as quais é maior a sensibilidade do sistema visual humano (como na matriz mpeg-intra). Para os sinais de croma, as componentes da DCT contribuem de modo aproximadamente uniforme para o erro de quantização observado. Assim, utilizam-se matrizes de quantização uniformes para as componentes Cr e Cb (matrizes mpeg-inter).
- (d) Tabelas de codificação de entropia adotadas no padrão JPEG, com distinção para as componentes de luminância e croma (Apêndice 4).
- (e) Valor da constante K (que define a parcela de contribuição do macrobloco estimado no processo de compensação de movimento) igual a 0,5 de modo que o quadro anterior e o quadro posterior possam contribuir igualmente para o processo de compensação de movimento.
- (f) Para a estimação de movimento, adotaram-se os seguintes parâmetros: uma área de pesquisa de ± 16 *pixels* em cada direção (valor máximo); limiar de aceitação do macrobloco igual a 40; função erro absoluto médio utilizada como função custo para o cálculo das discrepâncias entre macroblocos e método do salto logarítmico para a pesquisa do macrobloco. Este método foi comparado como o método de busca exaustiva. Para a FLOWER (que apresenta movimento translacional lento) obteve-se para o método de salto logarítmico um erro absoluto médio 5% superior ao valor obtido com a busca exaustiva. Para as outras seqüências (KIEL e MOBILE), o método de salto logarítmico propiciou erros absolutos em torno de 10% maiores que as buscas exaustivas. Em termos de rapidez computacional,

verificamos que o salto logarítmico consome um tempo quatro vezes menor que a busca exaustiva, justificando a sua adoção. Observa-se que há outros métodos de salto logarítmico que atuam procurando a melhor estimativa em outras regiões do quadro, reduzindo o erro às custas de um maior tempo computacional [113].

No processo de geração das curvas de taxa x distorção, amplificaram-se linearmente as matrizes de quantização. Esta amplificação (indexada de 1 a 10) deu origem aos vários pontos das curvas de taxa x distorção obtidas. Na Figura 4.1a podemos visualizar estas curvas. No eixo horizontal apresenta-se a taxa em valores de *bit/pixel* e no eixo vertical os valores do RMSE. Por exemplo, tomando-se o ponto A da sequência FLOWER, verificamos que podemos transmitir todos os 33 quadros com uma taxa de $\cong 0,86$ b/p e obter um erro quadrático médio de $\cong 6.66$. O valor da taxa pode ser transformado para expressar a taxa de transmissão em *Mbits/s* ou se referir ao fator de compressão conseguido. Para tal, temos as seguintes fórmulas de transformação.

$$T = T_c \times (M \times N) \times F, \quad (4.3)$$

$$C = P/T_c, \quad (4.4)$$

onde

T_c = Taxa em *bits/pixel*;

T = Taxa de transmissão em *bits/s*;

C = Fator de compressão;

M = Dimensão horizontal da imagem (em pixels);

N = Dimensão vertical da imagem (em pixels);

F = Número de quadros exibidos por segundo;

P = Número de *bits* necessários para se representar o *pixel* no original não comprimido.

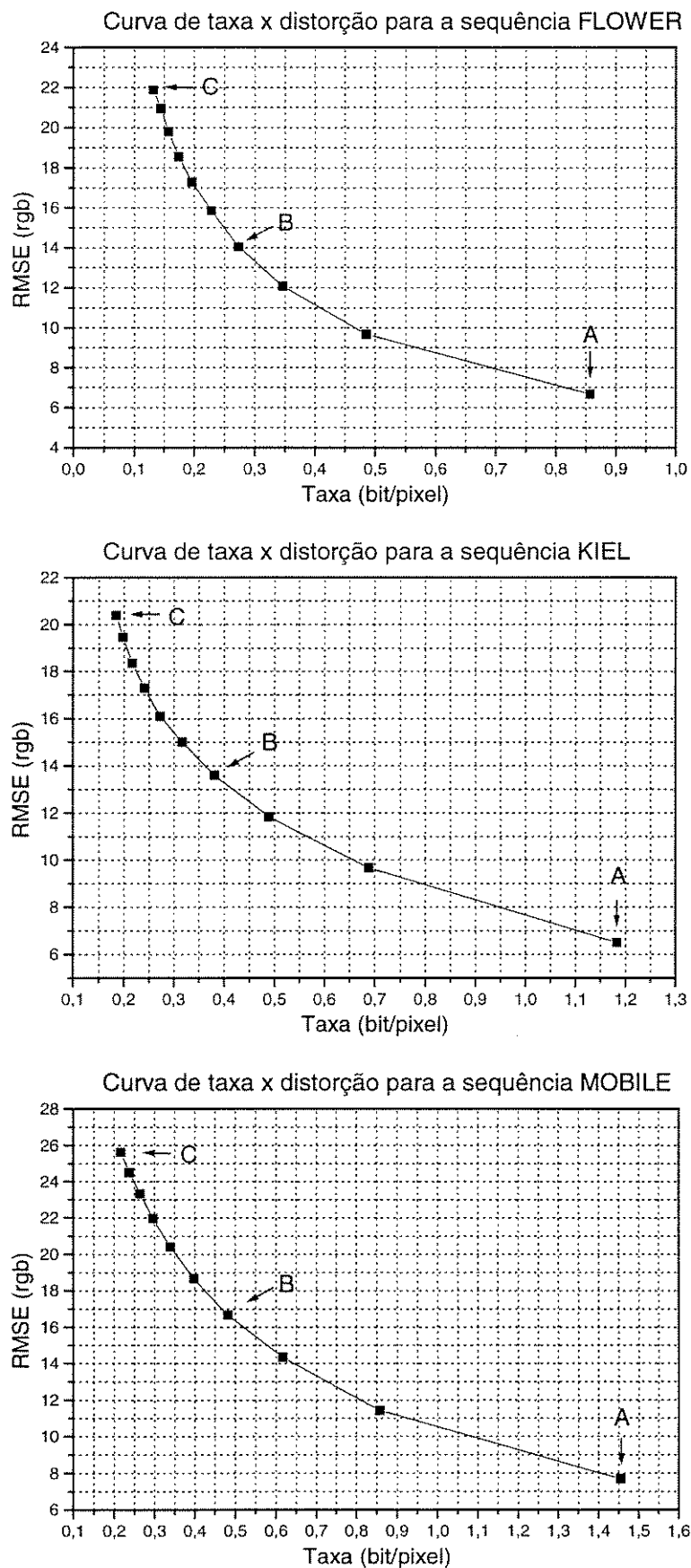


Fig. 4.1a - Curvas de taxa x distorção para as sequências FLOWER, KIEL e MOBILE.

Por exemplo, para quadros com $M = 720 \text{ pixels}$, $N = 480 \text{ pixels}$, $P = 24 \text{ bits}$ (imagens coloridas com 8 bits para cada componente de cor (R,G,B)) e $F = 30 \text{ quadros/s}$, obtemos para uma taxa padrão de $T_c = 1 \text{ b/p}$,

$$T = 1 \times (720 \times 480) \times 30 = 10,36 \text{ Mbits/s}$$

$$C = 24/1 = 24.$$

O padrão MPEG-II recomenda taxas de transmissão inferiores a 10Mbits/s. Assim, trabalharemos como nosso codificador a taxas menores que 1 b/p.

Na Figura 4.1b apresentamos o primeiro quadro original de cada seqüência utilizada. Nas Figuras 4.1c, 4.1d e 4.1e estes mesmos quadros são mostrados após codificação-decodificação com taxas e níveis de erro indicados pelos pontos A,B e C nas respectivas curvas de taxa x distorção (Fig. 4.1a). Assim, passamos a ter um sentimento de como se relacionam os valores de RMSE com a qualidade visual da seqüência de vídeo decodificada.

4.2 - ANÁLISE DAS PERDAS DE CÉLULAS SEM ESCALONAMENTO

Analizamos o caso da transmissão sem escalonamento por um canal sujeito à perda de células. Nessa análise consideramos que somente células que carregam dados da seqüência (e não sinais de sincronismos) são perdidas no processo.

As perdas de células são geralmente provocadas por sobrecargas dos acumuladores localizados nos multiplexadores da rede. A perda de células ocorre quando o fluxo das células acumuladas no armazenador geradas pelo codificador (canal 1) e pelos outros codificadores (canal 2) ultrapassam um dado limiar. Variando-se este limiar, pode-se simular diferentes taxas de perdas de células.

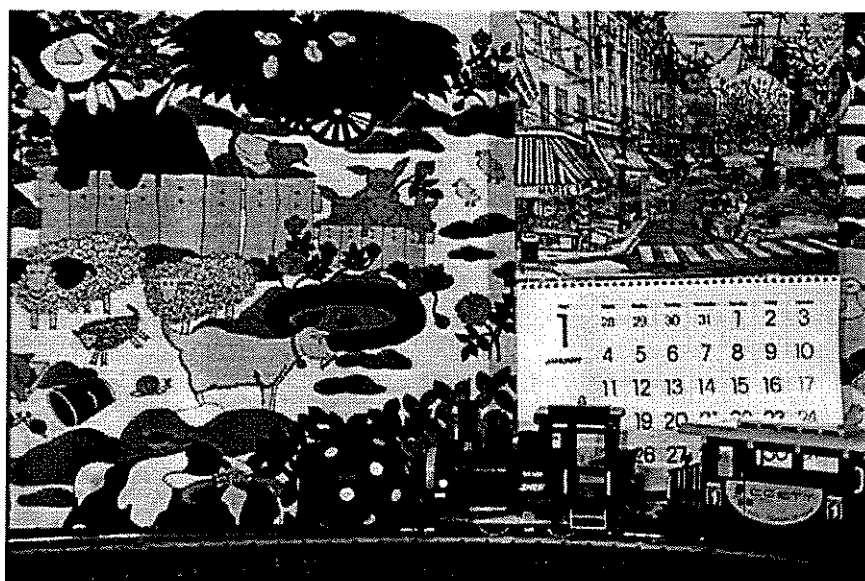
O Canal 2 é formado com dados de nove seqüências de vídeo codificadas pelo padrão MPEG, com GOP de tamanho 3 (BPI) representativos de diversos sinais de vídeo (filmes, jornais, desenhos, programas musicais, etc). O total de células para cada quadro individual destas nove seqüências são agrupadas, formando um conjunto de quadros fictícios com um modelo de células por quadro semelhantes à representação adotada no nosso codificador. As seqüências de vídeo foram extraídas do sítio na Internet da universidade de Wuerzburg (Alemanha), cujo endereço é <ftp://ftp-info3.informatik.uni-wuerzburg.de/pub/MPEG>. As seqüências utilizadas foram



FLOWER



KIEL



MOBILE

Fig. 4.1b - Primeiro quadro das seqüências FLOWER, KIEL e MOBILE.



Ponto A

RMSE $\cong$ 6.66Taxa $\cong$ 0.86 b/p

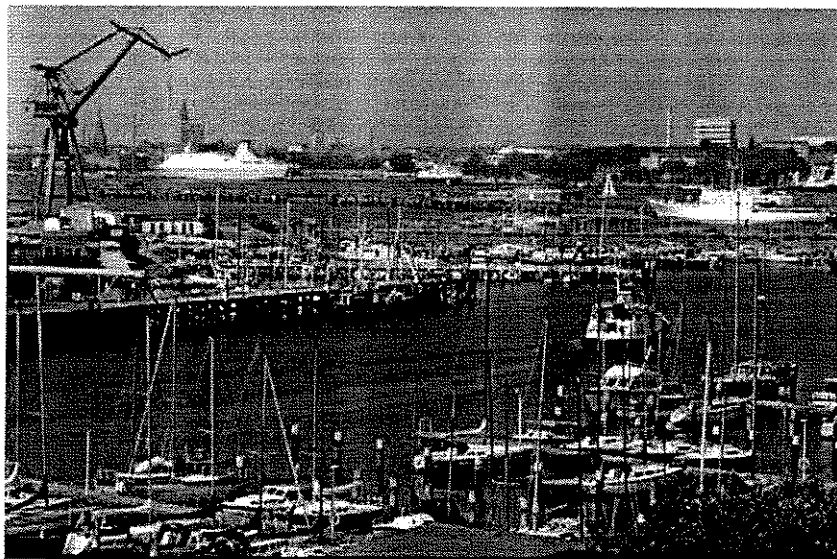
Ponto B

RMSE $\cong$ 14.03Taxa $\cong$ 0.27 b/p

Ponto C

RMSE $\cong$ 21.86Taxa $\cong$ 0.13 b/p

Fig. 4.1c - Comparação entre a qualidade subjetiva e o RMSE para o primeiro quadro da sequência FLOWER.



Ponto A

RMSE $\cong$ 6.49
Taxa $\cong$ 1.18 b/p



Ponto B

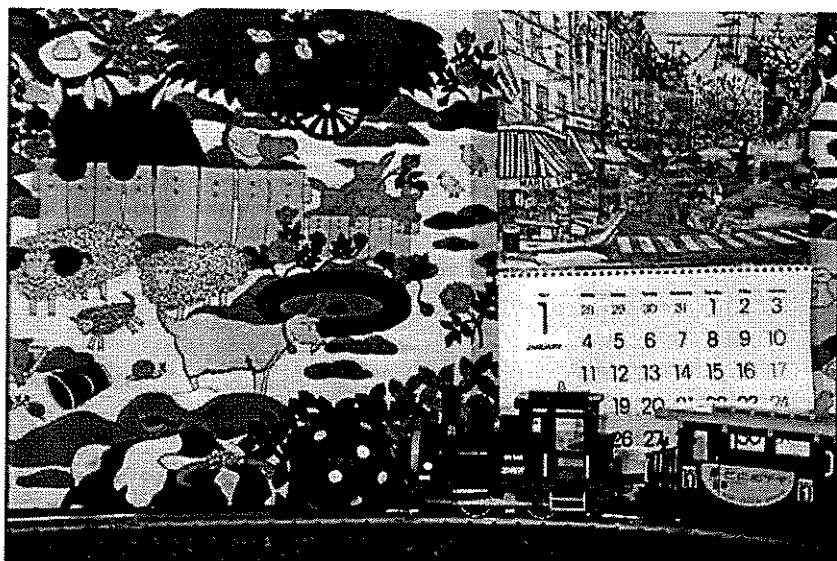
RMSE $\cong$ 13.60
Taxa $\cong$ 0.38 b/p



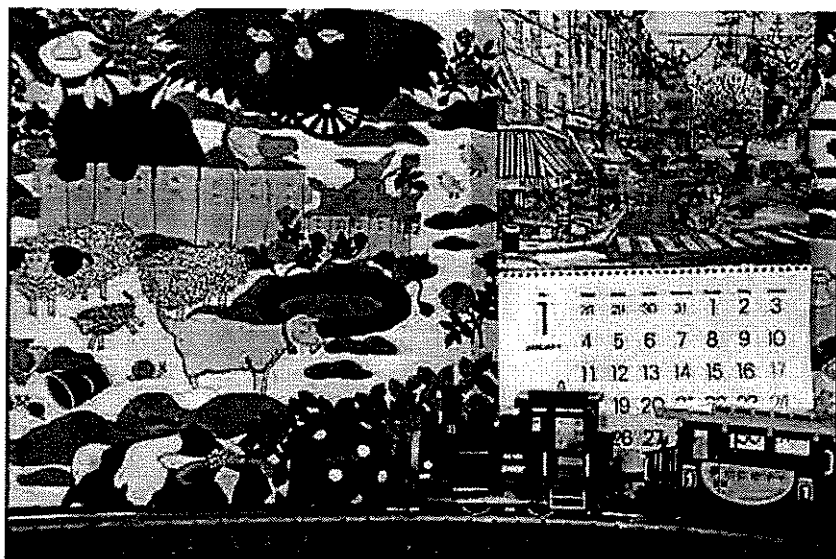
Ponto C

RMSE $\cong$ 20.38
Taxa $\cong$ 0.18 b/p

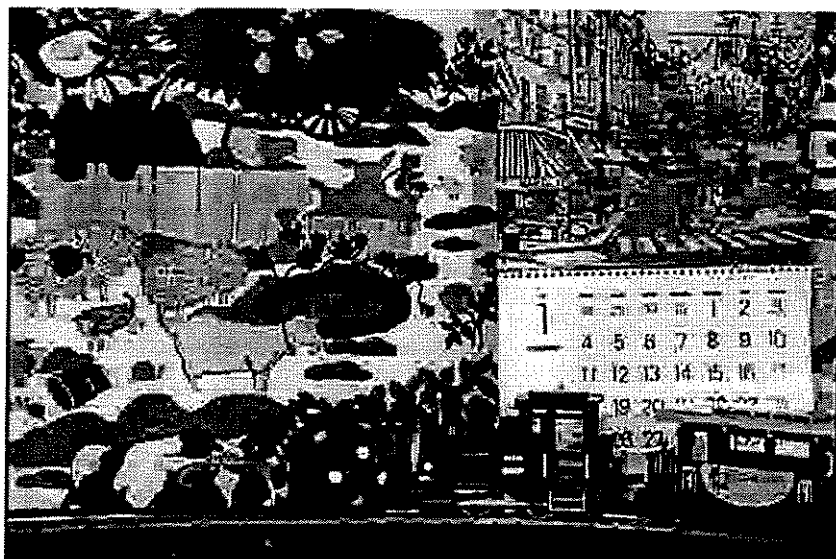
Fig. 4.1d - Comparação entre a qualidade subjetiva e o RMSE para o primeiro quadro da sequência KIEL.



Ponto A

RMSE $\cong$ 7.69Taxa $\cong$ 1.45 b/p

Ponto B

RMSE $\cong$ 16.66Taxa $\cong$ 0.48 b/p

Ponto C

RMSE $\cong$ 25.62Taxa $\cong$ 0.22 b/p

Fig. 4.1e - Comparação entre a qualidade subjetiva o RMSE para o primeiro quadro da sequência MOBILE.

- Filmes: MrBean_.IPB, star2_.IPB
- Desenhos: asterix_.IPB, dino_.IPB,
- Programas de auditório: talk_.IPB, talk2_.IPB,
- Telejornais: news_.IPB,
- Vídeo Clips: mtv_.IPB,
- Programas esportivos: race_.IPB.

A Figura 4.2 mostra o resultado obtido com esta extração.

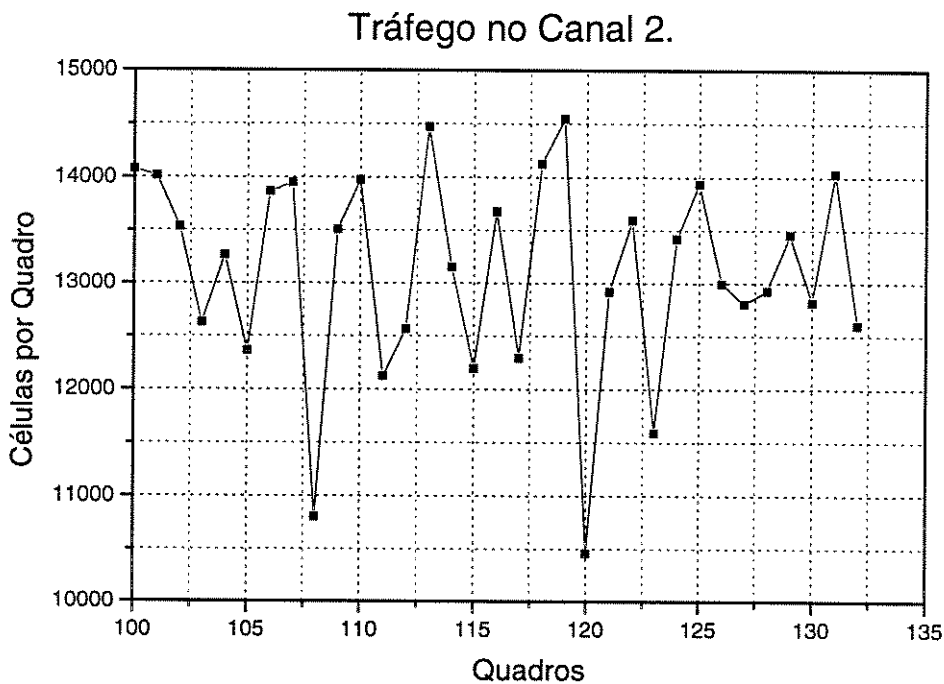


Fig. 4.2 - Formação do tráfego de células para o Canal 2.

Naturalmente, o tráfego total (Canal 1 + Canal 2) sofre a influência da quantidade de células geradas pelo nosso codificador. Como a quantidade de células codificadas por quadro depende da taxa (função dos níveis de quantização), não se pode adotar um mesmo limiar para todos os níveis de taxa e um valor fixo de taxa de perda de células. Adota-se, então, um limiar variável, calculado iterativamente pelo programa, para cada valor de taxa de modo que resulte em um nível de perdas constante por toda a curva de taxa x distorção. Na Figura 4.3, podemos verificar o efeito da quantidade de células gerada pelo nosso codificador no tráfego total de rede (supondo taxas de $\cong 1$ b/p e de $\cong 0,2$ b/p respectivamente).

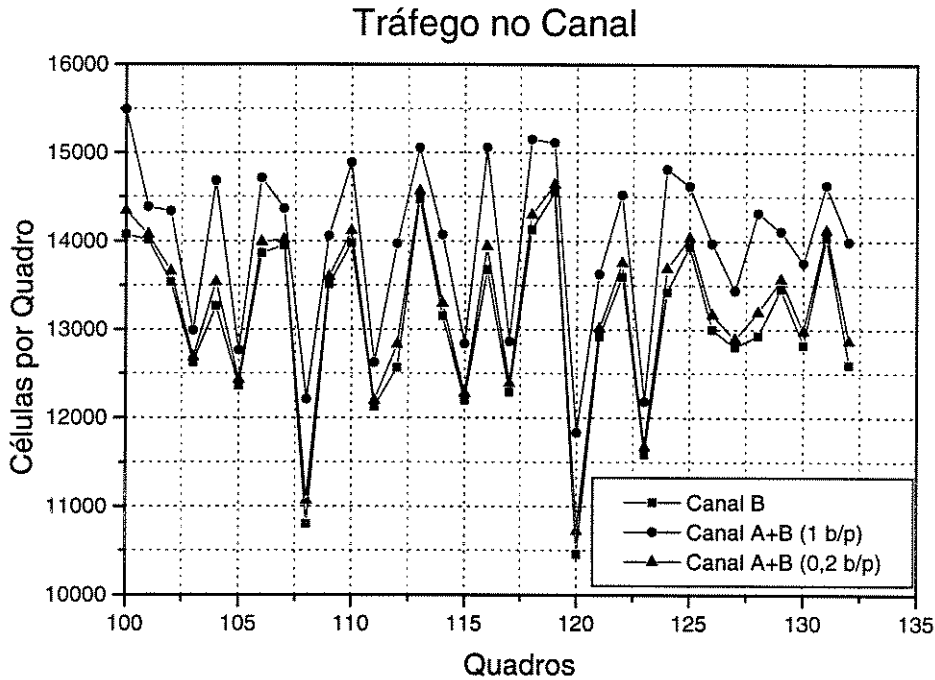


Fig. 4.3 - Exemplo do tráfego com variação da taxa do codificador.

Utilizando este modelo de tráfego, analisamos o efeito da perda de células na distorção do sinal recebido (Fig. 4.4). Convém lembrar que estas curvas referem-se somente à perda de dados de codificação do quadro e não a perda de sincronismos. Os resultados obtidos apresentam uma distorção menor do que ocorreria no caso real, caso ocorressem perdas no sincronismo e esquemas de proteção contra estas perdas não fossem introduzidos. Estas curvas foram obtidas variando-se a taxa de perda de células na faixa de 0,25% a 10%. Destacam-se as seguintes observações:

1. Para níveis de perda muito baixos (até $\approx 0,25\%$ para a sequência FLOWER e até $\approx 0,5\%$ para as sequências KIEL e MOBILE), a perda de células não afeta em muito a qualidade das sequências recebidas, pois as curvas de taxa x distorção se encontram bem próximas das curvas obtidas para o caso sem perdas (0%), que são utilizadas como curvas de referência.
2. Para níveis maiores de perda, ocorre um aumento acentuado da distorção e as curvas de taxa x distorção passam a ter um comportamento não monotônico e aleatório, ora aumentando ora diminuindo quando o nível de perdas de células aumentam (veja que as curvas muitas vezes se cruzam). Estes fatos estão ligados a dois fatores principais: a propagação de erros, função do tipo do quadro onde as células são perdidas e da região nos quadros afetados por esta perda.

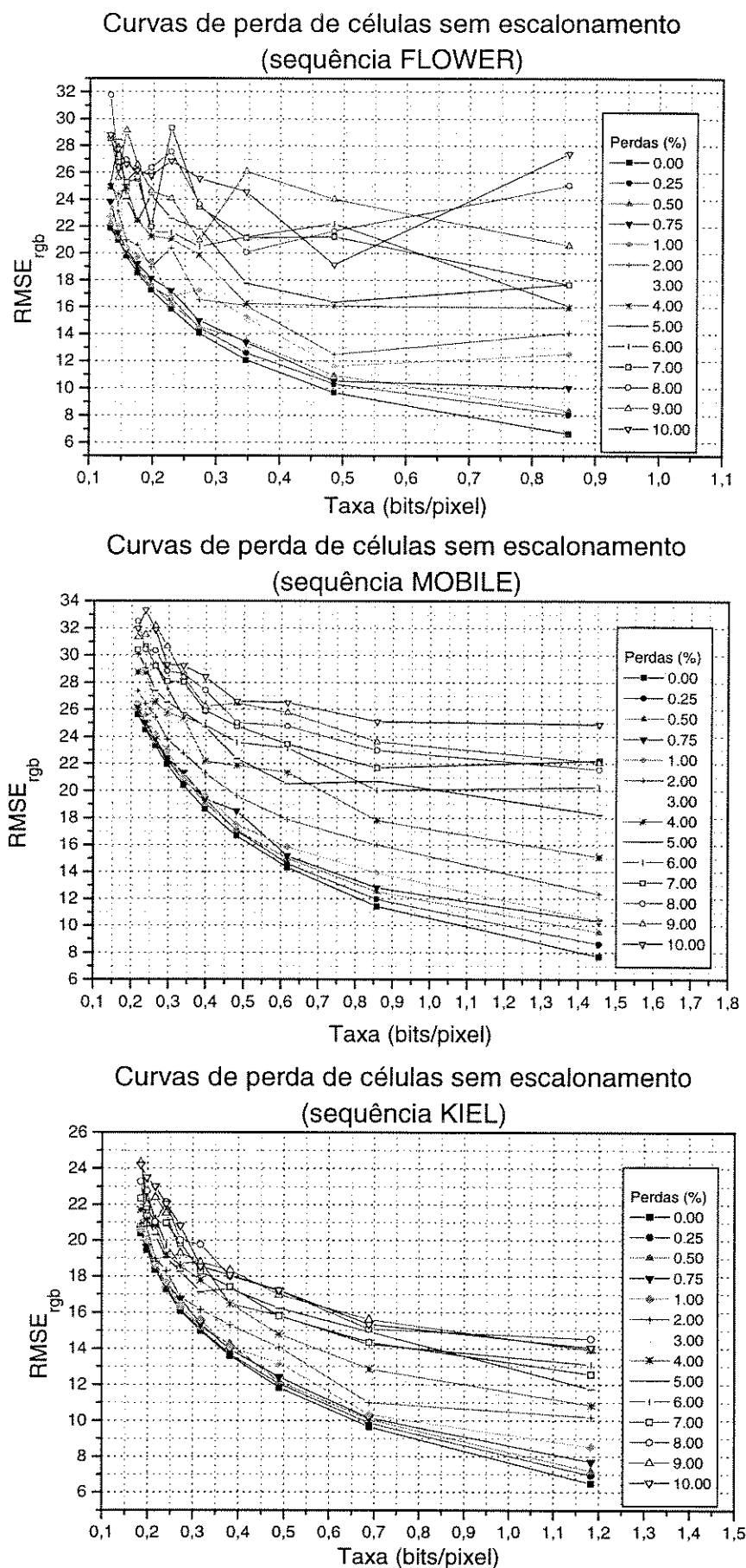


Fig. 4.4 - Curvas de perda de células para as seqüências codificadas sem escalonamento.

Quando perdemos células de um quadro tipo I, não causamos erros apenas neste quadro, mas em todo GOP, pois os quadros tipo I servem de referência para a estimação de movimento nos quadros B e P. Perdas de células em quadros tipo P, podem afetar quadros tipo B se estes usarem aqueles quadros como referência. As perdas de células em quadros do tipo I tem um agravante a mais, já que estes quadros não se referenciam a nenhum outro, deixando a região do quadro onde os dados foram perdidos sem definição. Para minimizar este problema, copiamos no quadro I anterior a região que foi perdida. Para quadros do tipo B e P perdemos apenas a informação diferencial devido à codificação preditiva com compensação de movimento, restando a informação da região do quadro de referência para preencher esta região.

Convém ressaltar que estamos considerando apenas 33 quadros no experimento. As curvas de taxa x distorção consideradas são curvas operacionais, obtidas dos pontos observados de taxas x distorção medidas nas reproduções dos 33 quadros da sequência. Portanto, estas curvas taxas x distorção não precisam obedecer os critérios de convexidade e monotonicidade impostos pela Teoria de Informação. Quando o número de quadros aumenta, espera-se que os efeitos que originaram a não monotonicidade das curvas se aproximem da média ao longo prazo, gerando curvas de taxa x distorção mais suaves, embora ainda em patamares de distorção inaceitáveis do ponto de vista teórico.

Concluímos que a transmissão dos dados sem proteção só é viável quando a taxa de perdas é muito pequena (valores menores que 0,5%). Com taxas de perda maiores (frequentes nos casos práticos) a transmissão sem proteção se torna inviável devido ao aumento acentuado da distorção e degradação da informação recebida.

4.3 - ANÁLISE COM PERDAS DE CÉLULAS E COM ESCALONAMENTO

Para atenuar os efeitos das perdas de células, efetuamos um escalonamento na informação antes de enviá-la. O escalonamento atua basicamente dividindo a informação de vídeo em camadas e transmitindo-as em canais com diferentes níveis de segurança. Trabalhamos com escalonamentos em duas camadas; uma primeira camada

sendo transmitida com alta prioridade¹ contendo a informação essencial para a recuperação do sinal com uma qualidade definida no momento da codificação, e uma segunda camada, de menor prioridade, que contém a informação de enriquecimento do sinal original. Quando se faz necessário o descarte de células, as células de baixa prioridade são as primeiras a serem descartadas garantindo assim, uma qualidade mínima ao sinal recebido.

Consideramos dois métodos de escalonamento. O primeiro denominado de requantização atua nos elementos dos macroblocos separando-os em planos de *bits*, isto é, através de operações de quantização, quantização inversa e subtração, os *bits* representativos dos valores dos elementos dos macroblocos são particionados em dois conjuntos. Os planos de *bits* mais significativos são transmitidos em alta prioridade e os *bits* menos significativos (que "enriquecem" a informação transmitida) são enviados em baixa prioridade. O segundo método, denominado separação espectral, atua no domínio da DCT particionando os elementos dos macroblocos em dois conjuntos. O primeiro conjunto envolve o elemento DC e alguns elementos AC, constituindo os dados transmitidos em alta prioridade. O segundo conjunto, formado pelos elementos AC restantes, são transmitidos em baixa prioridade.

Analisemos a eficiência destes dois modos de escalonamento na melhoria do sinal recebido em ambiente de perda de células, iniciando pela requantização.

4.3.1 - REQUANTIZAÇÃO

Existem muitos critérios que podem ser adotados quando se realiza uma requantização, critérios estes relacionados com as escolhas arbitrárias das matrizes de quantização (Q_1 e Q_2 no nosso caso). Adotamos um critério que estabelece uma relação $Q_1/Q_2 = k$ (onde k é uma potência de 2). Fixando a razão Q_1/Q_2 como um número inteiro, garantimos no caso de não haver perda de células, um mesmo nível de erro de quantização caso a sequência não sofra escalonamento e seja quantizada por Q_2 . Assim, ao se utilizar Q_1/Q_2 inteiro a requantização não eleva o erro de quantização natural do processo de codificação. Verificamos este fato, comparando o erro de quantização gerado no processo de quando codificamos os dados sem

¹ Podemos ajustar a prioridade de uma célula ATM ajustando o valor do *bit* CLP que se encontra no cabeçalho da célula (vide Capítulo 2).

escalonamento e o erro gerado quando utilizamos requantização. Para o caso sem escalonamento, utilizando-se um quantizador linear com passo de quantização Δ (Fig. 4.5a), obtemos um erro conforme mostrado na Figura 4.5b.

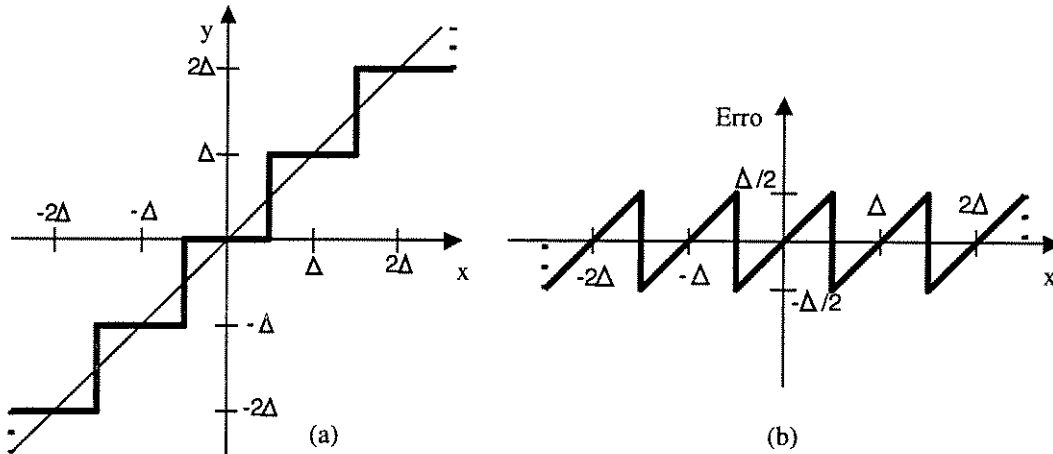


Fig. 4.5 - Curvas do quantizador com passos uniformes

(a) Curva de quantização (passo = Δ).

(b) Erro de quantização.

Verificamos que o erro de quantização está limitado a faixa de $-\Delta/2$ a $\Delta/2$, sendo simétrico em relação à origem. Analisando o erro com requantização, onde Q_2 é igual ao quantizador utilizado na codificação sem escalonamento e $Q_1 = k \cdot Q_2$, verificamos que o erro do processo é dado pelo erro de requantização de Q_2 quando quantiza os erros de quantização devidos a Q_1 (Fig. 4.6).

Como os erros de quantização devido a Q_1 estão limitados à faixa de $x_{\min} = -k\Delta/2$ a $x_{\max} = k\Delta/2$ e estes valores são múltiplos de $\Delta/2$, quando efetuamos a quantização por Q_2 , geramos um curva de erro que apresenta a mesma forma da curva de erro dada na Fig. 4.5b. Sendo estes erros iguais, a distorção será a mesma para os casos sem escalonamento e com requantização. Na Figura 4.6b, formulamos um exemplo com $k=4$. Notamos que o valor quantizado por Q_2 encontra-se limitado a faixa de $-4\Delta/2$ a $+4\Delta/2$, que, após quantizado, gera um erro expresso pela Figura 4.6c, semelhante à curva da Figura 4.5b.

Fixando o valor de k como potência de 2, estamos definindo número de *bits* médios necessários para a representação dos dados de baixa prioridade; número este dado por $\log_2 k$. Este fato pode ser verificado notando-se inicialmente que o erro de quantização gerado por Q_1 pertence a faixa de $[-k\Delta/2, k\Delta/2]$. Quando quantizamos

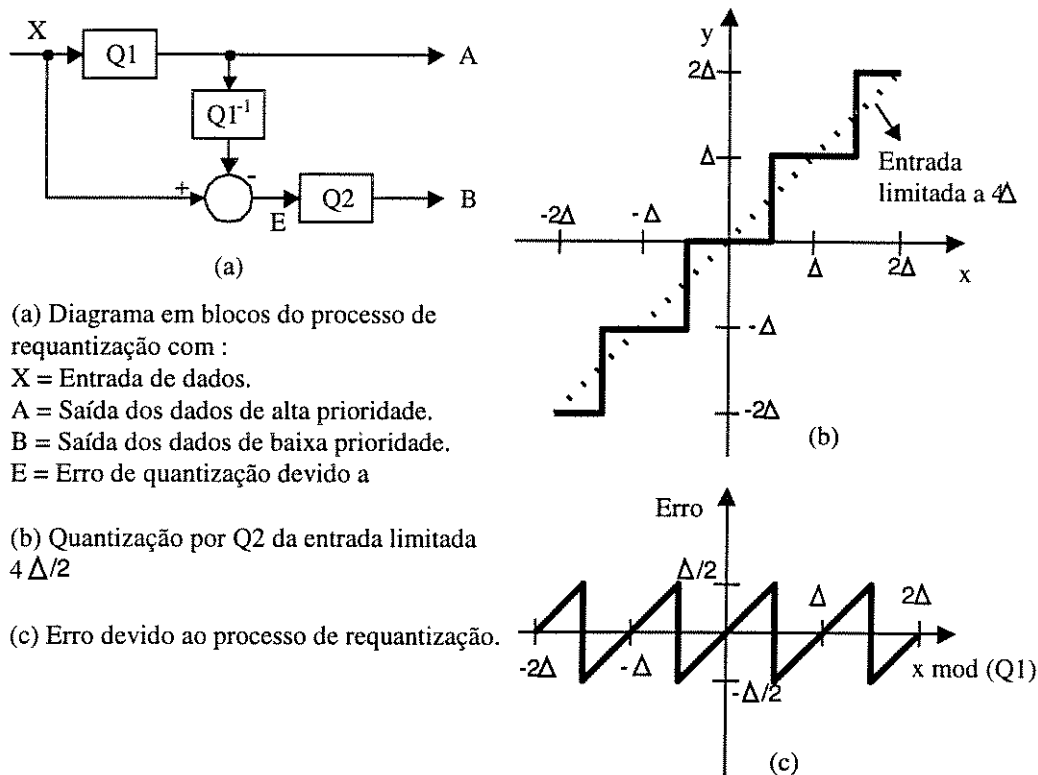


Fig. 4.6 - Análise do erro de quantização no processo de requantização.

este erro por $Q2$ com passo Δ , geramos erros na faixa de $[-k/2, k/2]$. Sendo k potência de 2 e inteiro, teremos $2k/2$ possíveis valores, necessitando de $\log_2(k)$ bits para a sua representação.

Realizamos as simulações referentes à requantização com k assumindo valores de 2 e 4 e níveis de perda de células dados por: 0% (sem perdas), 1%, 5%, 10%, 15%, 20 % e perda de todos os dados transmitidos em baixa prioridade. Os resultados podem ser vistos nas Figuras 4.7a (para $Q1/Q2 = 2$) e 4.7b para ($Q1/Q2 = 4$), juntamente com a curva de referência para 0% de perda sem escalonamento.

Notamos que o processo de requantização apresenta um certo *overhead* com relação à codificação sem escalonamento, ou seja, usando requantização gastamos mais bits para enviar a mesma sequência com um dado nível de distorção. Verificamos este fato observando a posição relativa entre as curvas sem escalonamento e com requantização para o nível de 0% de perdas para as três sequências de vídeo e para os vários valores de k (Fig. 4.7 a e b). As curvas de requantização para 0% de perda

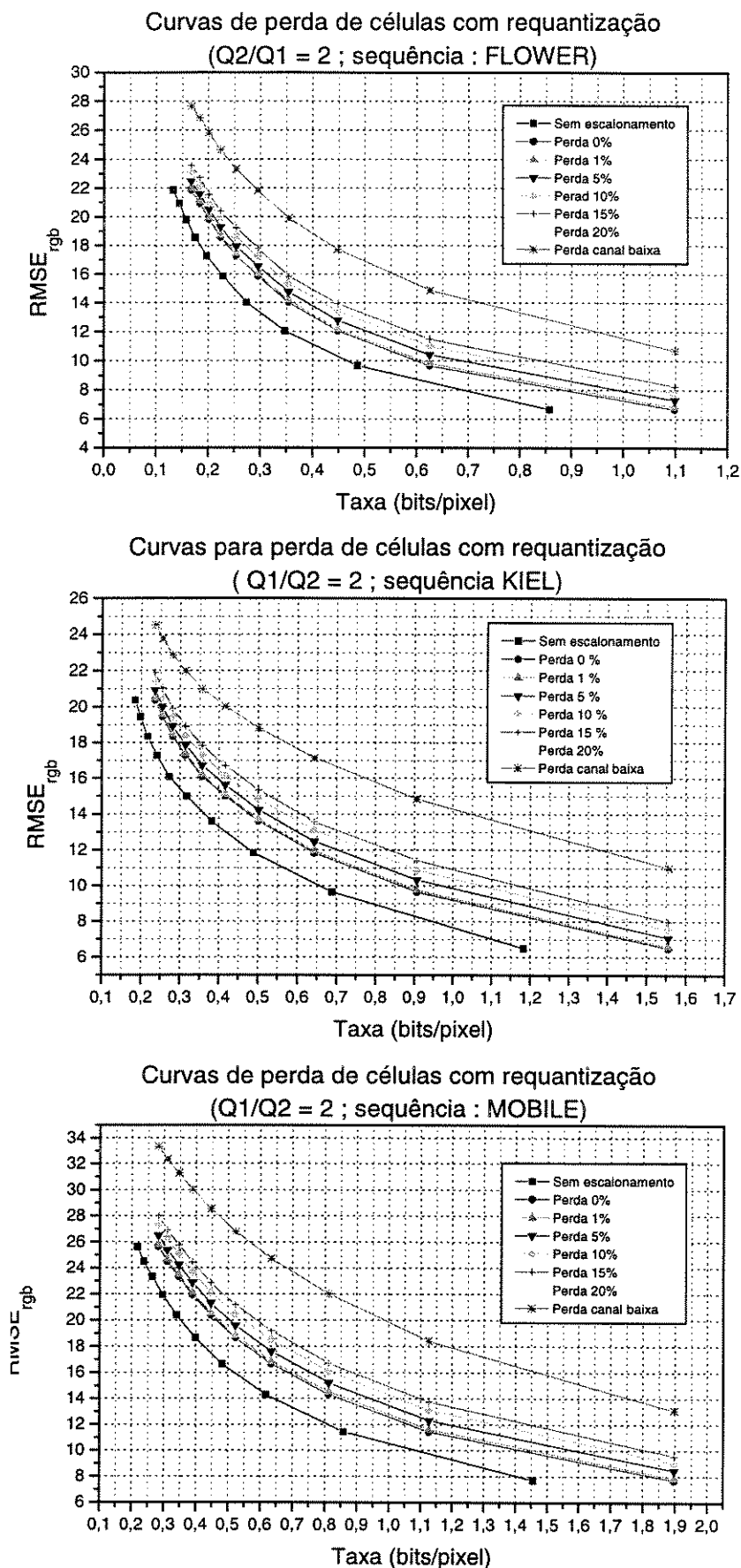


Fig. 4.7a - Curvas de perda de células para a requantização com $Q1/Q2 = 2$.

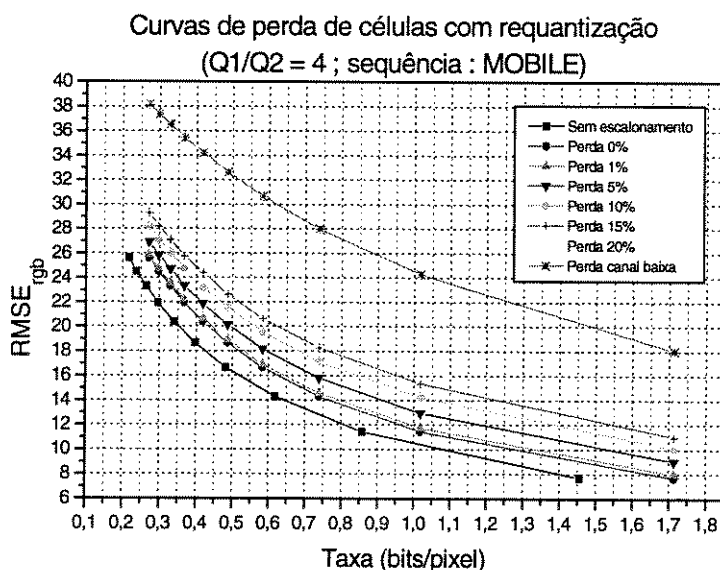
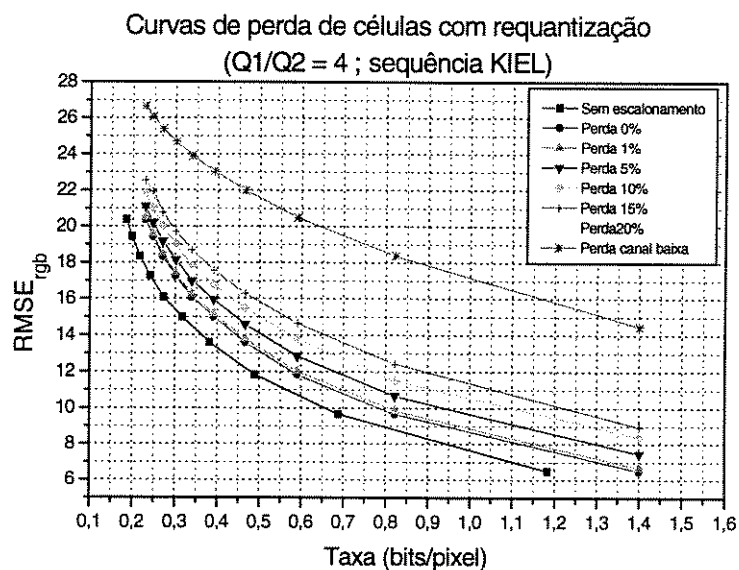
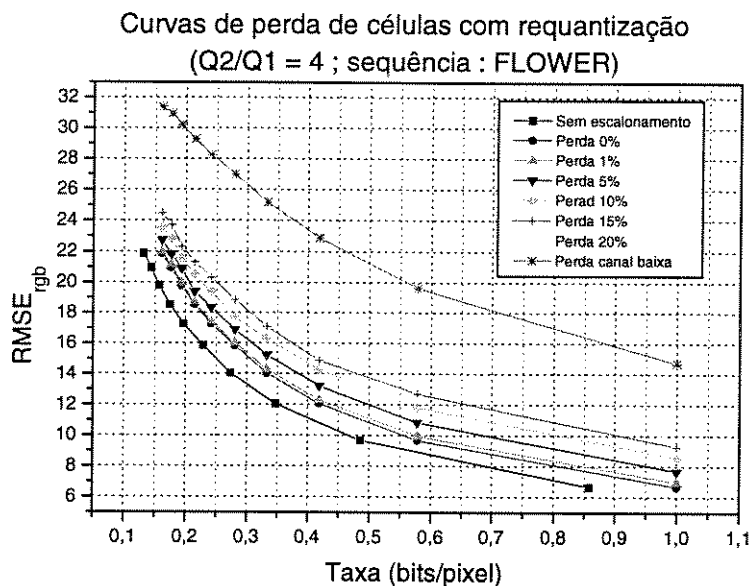


Fig 4.7b - Curvas de perda de células para a requantização com $Q1/Q2 = 4$

sofrem um deslocamento para à direita, indicativo de que a distorção foi mantida e tivemos um aumento da taxa de transmissão.

Definindo o *overhead* entre os esquemas de codificação como

$$\text{Overhead} = (\text{Nce}/\text{Nse}) \times 100\% , \text{ onde} \quad (4.5)$$

Nce denota o número de bits na codificação com escalonamento e Nse é o número de bits na codificação sem escalonamento, podemos desenhar as curvas de *overhead* comparando a requantização com a codificação sem escalonamento para as três seqüências e para os dois valores de k, como se observa na Figura 4.8. Nota-se que o *overhead* da requantização é função do valor k e do nível de distorção.

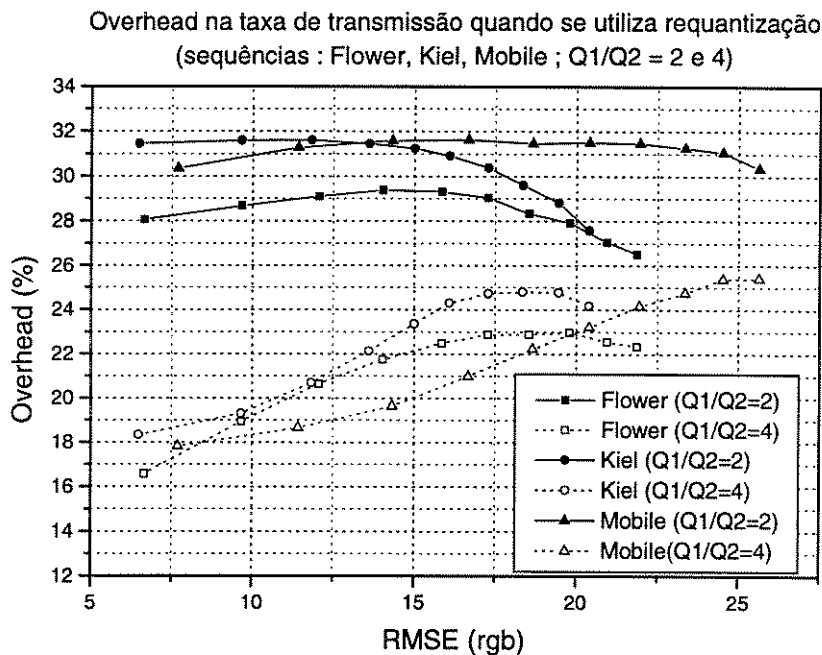


Fig. 4.8 - *Overhead* na taxa de transmissão quando se utiliza requantização.

O *overhead* deve-se as ineficiências de codificação inerentes à requantização e a introdução de dados necessários à recuperação de sincronismo na eventualidade de perda de células. Este *overhead* ou sobrecarga situa-se na faixa de 28 a 32% para a requantização com $Q1/Q2 = 2$ e na faixa de 17 a 25% com $Q1/Q2 = 4$. Esta diferença de eficiência pode estar associada a pequena proporção dos dados transmitidos em

baixa prioridade no caso $Q1/Q2 = 2$, que implica em um maior custo relativo dos dados de controle e sincronismo neste caso.

As curvas com requantização para os casos de perda de células apresentam um comportamento muito parecido com a curva do caso sem perdas. Notamos um deslocamento destas curvas resultando numa maior distorção, como se poderia esperar. As distorções aumentam com o nível de perda de células, deslocando as curvas aproximadamente de uma mesma distância a cada aumento de 5% nas perdas (de 0% a 20%). O pequeno aumento da distorção com o aumento das perdas de células contrasta em muito com o caso sem escalonamento. Com perdas, por exemplo da ordem de 10 % para o caso sem escalonamento o processo de transmissão se torna quase inviável pois a distorção se apresenta elevada para todos os níveis de taxas de transmissão. Utilizando requantização, conseguimos reduzir o nível da distorção tornando a transmissão viável. Notamos que, mesmo perdendo toda a informação de baixa prioridade, garantimos ainda um certo nível de qualidade ao sinal que chega ao receptor (nível de distorção dado pela curva de perda total do canal de baixa prioridade). Analisando as curvas de perda em função do parâmetro k , podemos dizer que as curvas geradas com $k = 2$, apresentam um maior *overhead* de transmissão se comparada com o caso de $k=4$, embora este último caso seja mais sensível as perdas de células, já que sua curva de taxa x distorção sofre um deslocamento mais acentuado para a direita quando o nível de perdas

4.3.2 - SEPARAÇÃO ESPECTRAL

Um segundo modo de escalonar os dados é empregar a separação espectral. Como no caso da requantização, temos uma certa arbitrariedade na escolha dos seus parâmetros. A fim de tornar os escalonamentos comparáveis para um determinado nível de perda de células, impomos que sejam perdidas quantidades iguais de células de baixa prioridade em ambos os esquemas. Isto é conseguido controlando-se o parâmetro n_s da separação espectral (número de elementos dirigidos para o canal de alta prioridade) de modo que a porcentagem de células geradas pelo canal de baixa prioridade em relação ao total de células seja igual ao conseguido com requantização em cada nível de quantização que está relacionada a taxa de transmissão). Logo, através da variação do parâmetro q (multiplicador da matriz de quantização) e do parâmetro n_s , mapeamos as possíveis relações entre o número de células de baixa

prioridade e o número total de células geradas. Assim, podemos escolher o valor de ns que proporcione a mesma relação para a requantização com cada nível de quantização. A requantização foi efetuada para duas relações entre os quantizadores ($k = 2$ e $k = 4$). Assim, teremos dois tipos de separação espectral: tipo 1 e tipo 2, correspondentes aos valores de $k = 2$ e $k = 4$ respectivamente. A Tabela 4.1 mostra os resultados encontrados, com o parâmetro q variando de 1 a 10 e ns variando de 1 a 15 para as três seqüências analisadas.

FLOWER	q	1	2	3	4	5	6	7	8	9	10
Tipo 1	ns	8	5	4	3	3	2	2	2	2	2
Tipo 2	ns	4	2	2	2	1	1	1	1	1	1
KIEL	Q	1	2	3	4	5	6	7	8	9	10
Tipo 1	ns	8	5	4	3	3	2	2	2	2	2
Tipo 2	ns	4	2	2	2	1	1	1	1	1	1
MOBILE	Q	1	2	3	4	5	6	7	8	9	10
Tipo 1	ns	9	6	5	4	3	3	2	2	2	2
Tipo 2	ns	5	3	2	2	2	2	1	1	1	1

Tab. 4.1 - Número de elementos direcionados ao canal de alta prioridade (ns) em função do nível de quantização.

Para os valores de ns encontrados, obtemos as curvas de taxa x distorção para a separação espectral (Figuras 4.9a e 4.9b).

Assim como no processo de requantização, observamos um *overhead* quando procedemos a separação espectral. A Figura 4.10 apresenta as curvas de *overhead* em função do nível de distorção RMSE para as três seqüências de vídeo. O *overhead* obtido tende a ser menor que o da requantização devido ao fato que a separação espectral apenas direciona quais elementos dos macroblocos são codificados em alta ou em baixa prioridade constituindo-se em um esquema simples, eficiente e de fácil resincronização.

Como no caso da requantização, notamos uma melhora significativa na robustez do sistema para o caso de perda de células. As curvas de taxa x distorção sofrem pequenos deslocamentos ascendentes relacionados com o aumento do nível de

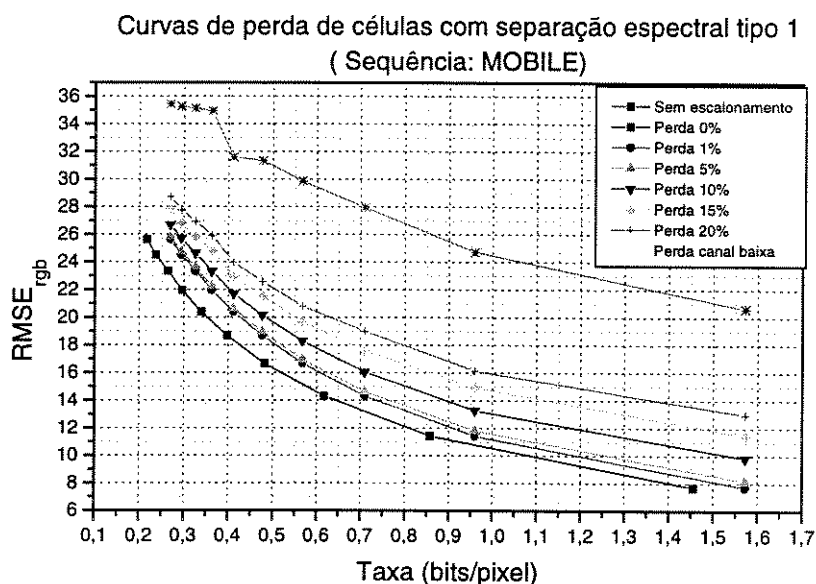
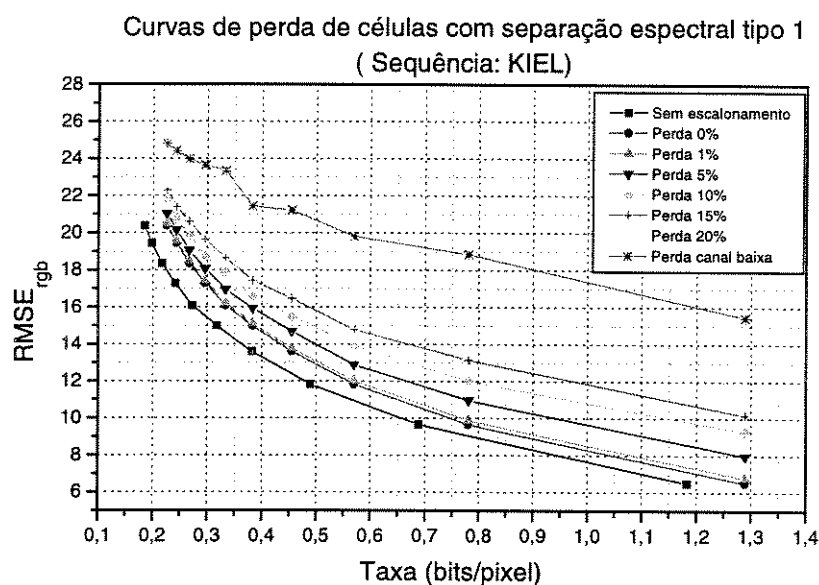
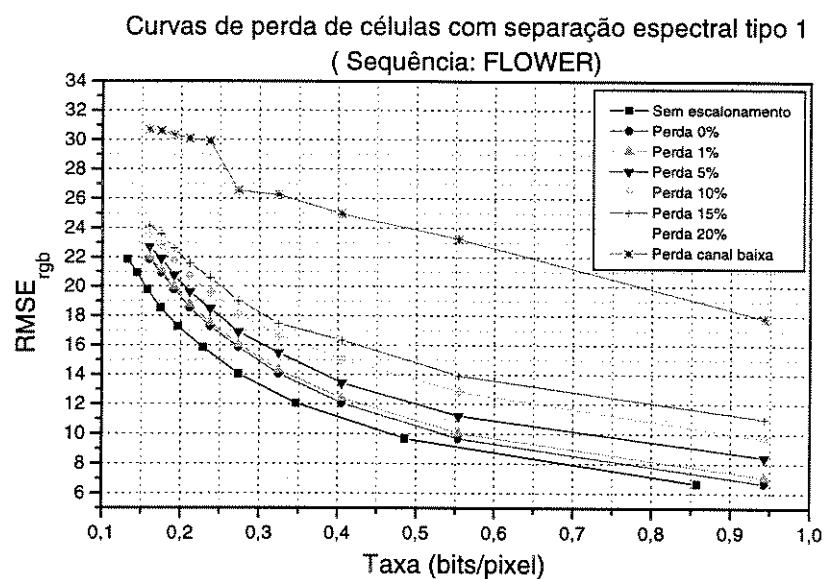
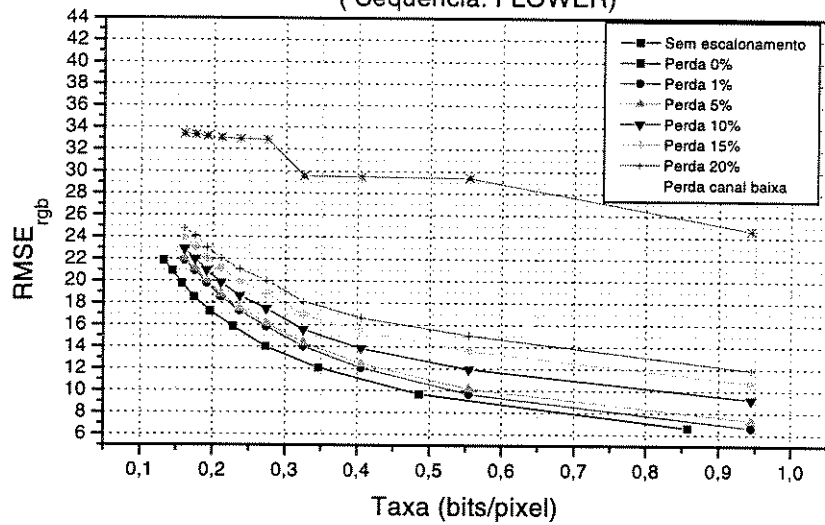
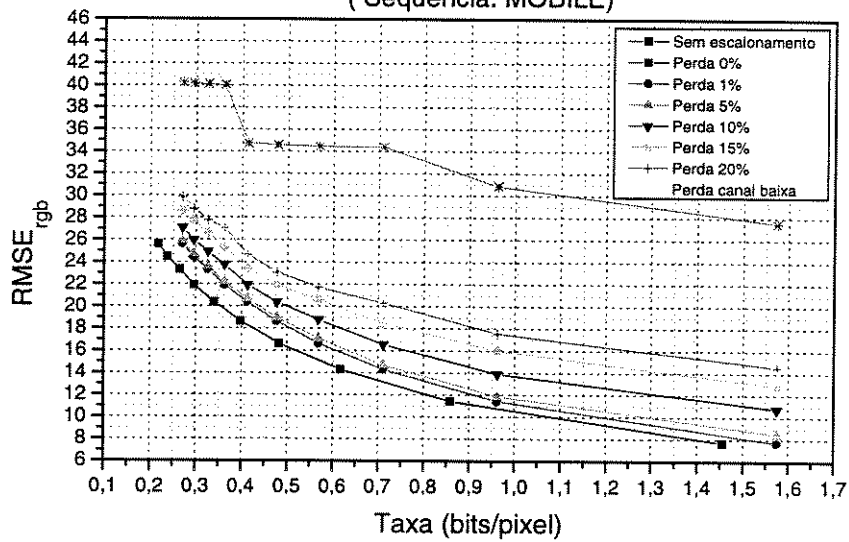


Fig. 4.9a - Curvas de perda de células para a separação espectral tipo 1.

Curvas de perda de células com separação espectral tipo 2
(Sequência: FLOWER)



Curvas de perda de células com separação espectral tipo 2
(Sequência: MOBILE)



Curvas de perda de células com separação espectral tipo 2
(Sequência: KIEL)

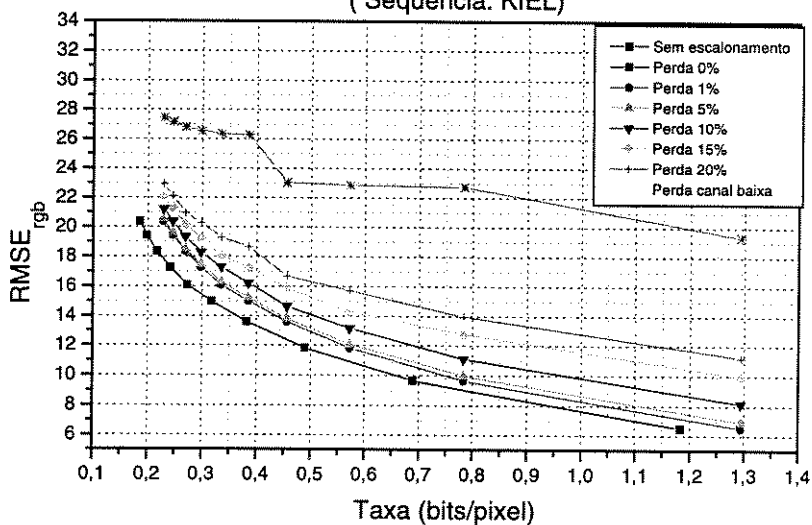


Fig. 4.9b - Curvas de perda de células para a separação espectral tipo 2.

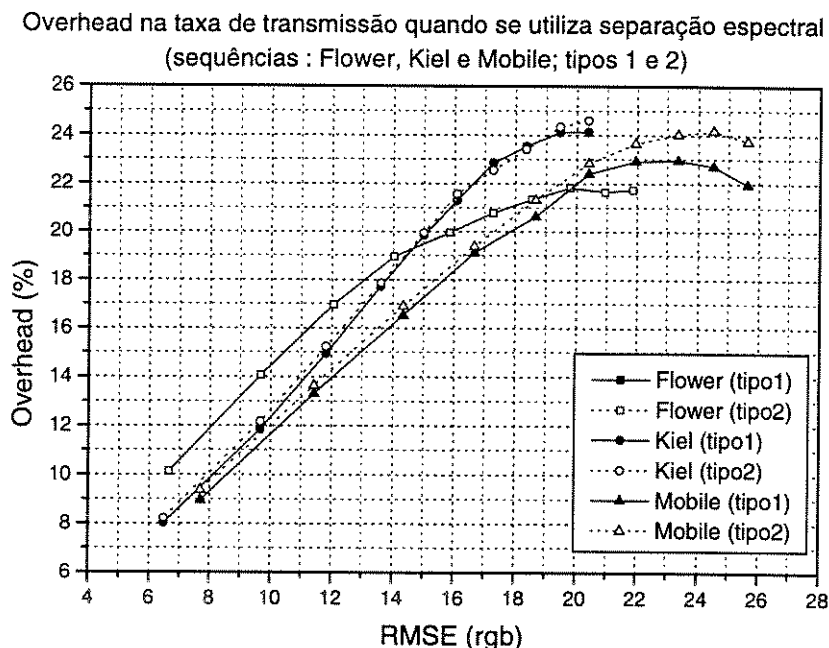


Figura 4.10 - *Overhead* na taxa de transmissão com separação espectral.

perdas. Notamos que o aumento brusco e desordenado da distorção com o aumento do nível de perdas não ocorre neste caso. Mesmo para a perda total do canal de baixa prioridade, garante-se uma certa qualidade para a sequência recebida (vide curvas das Figuras 4.9a e 4.9 b para perda do canal de baixa prioridade).

De um modo geral, verifica-se que utilizando requantização ou separação espectral, consegue-se reduzir o nível de distorção do sinal transmitido em relação à transmissão sem escalonamento. Esta melhoria tem o preço de exigir uma taxa de transmissão extra (*overhead*) para o envio dos dados. Este aumento de taxa ocasiona dois efeitos:

- (1) O deslocamento das curvas de taxa x distorção. A comparação entre os esquemas de codificação deve levar em conta o ponto das curvas em que se operam os esquemas.
- (2) Um aumento do *overhead* geralmente leva a um aumento da taxa de perda de células num canal real, já que o nível de perda de células está relacionado com o mínimo de células que trafegam na rede. Logo, devemos ajustar os níveis de perda para comparar os esquemas de transmissão.

Simplificamos nossa análise, supondo que o codificador não altera significativamente o tráfego na rede, de modo que, numa primeira aproximação comparamos as curvas de taxa x distorção com um mesmo nível de perda de células.

De modo a se determinar o esquema de transmissão mais adequado para um dado nível de perda de células e uma dada taxa de transmissão, compararemos os vários esquemas de codificação. A comparação se dá através da análise das curvas de taxa x distorção. Superpondo-se as curvas, podemos determinar o esquema de codificação que resulta na menor distorção para cada nível de perda de célula e cada valor de taxa de transmissão.

4.4 - COMPARAÇÃO ENTRE AS CODIFICAÇÕES COM E SEM ESCALONAMENTO

Nas Figuras 4.11, 4.12, e 4.13 temos as comparações entre os esquemas de codificação com e sem escalonamento. As técnicas de escalonamento incluem requantização com $Q1/Q2 = 2$ ou 4 e separação espectral dos tipos 1 e 2. Desta comparação vemos que nem sempre é preferível escalonar os sinais, pois as curvas de taxa x distorção podem se interceptar. Para o caso trivial sem perda de células, é sempre preferível transmitir o sinal sem escalonamento. À medida que o nível de perda de células aumenta, as curvas sem escalonamento sofrem um deslocamento mais acentuado que as curvas com escalonamento. Os pontos de possíveis cruzamentos das curvas delimitam as regiões onde é preferível utilizar um esquema de codificação em relação ao outro. Assim, por exemplo, para a sequência MOBILE e requantização com $Q1/Q2 = 2$, encontramos, para um nível de perda de células de 1%, um ponto de intersecção das curvas na taxa de aproximadamente 0,85 *bits/pixel*. Assim, quando o nível de perdas é de 1%, passa a ser mais vantajoso transmitir com requantização para taxas menores que este valor. Observando estas curvas, verifica-se que, para níveis de perda superiores a 1% (FLOWER e MOBILE) ou 5% (KIEL) as curvas não mais se interceptam, de modo que a transmissão utilizando escalonamento sempre é superior (i.e., para qualquer taxa) ao caso sem escalonamento. Na sequência KIEL, notamos que 1% de perda de células, a intersecção entre as curvas ocorre em regiões de baixas taxas, onde possíveis perda de células não afetam acentuadamente a distorção, já elevada devido à quantização grosseira. Portanto, podemos adotar como regra prática que é desejável utilizar escalonamento para níveis superiores a 1% .

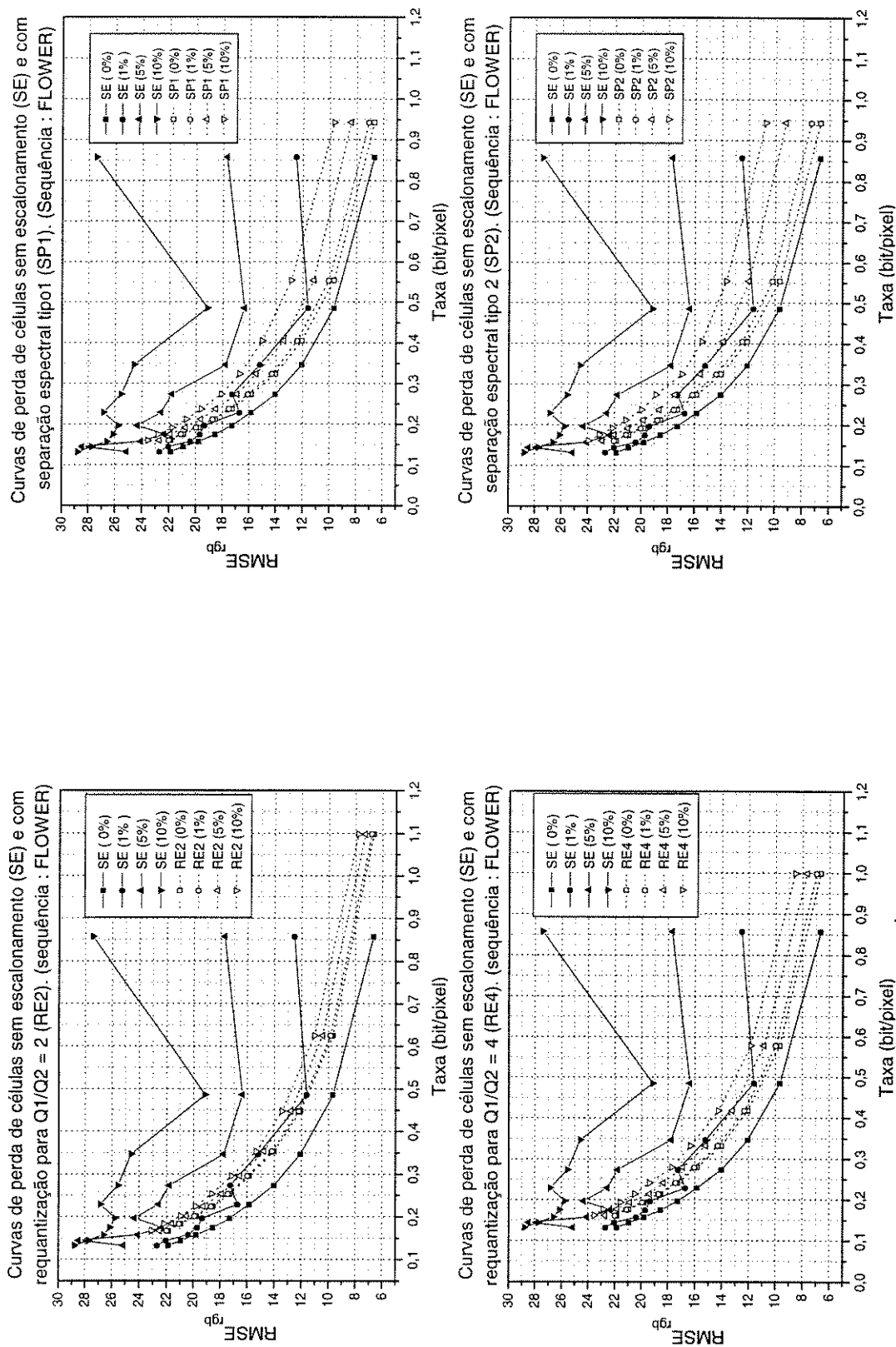


Fig. 4.11 - Comparação entre codificações com e sem escalonamento para um dado nível de perda de células (sequência FLOWER).

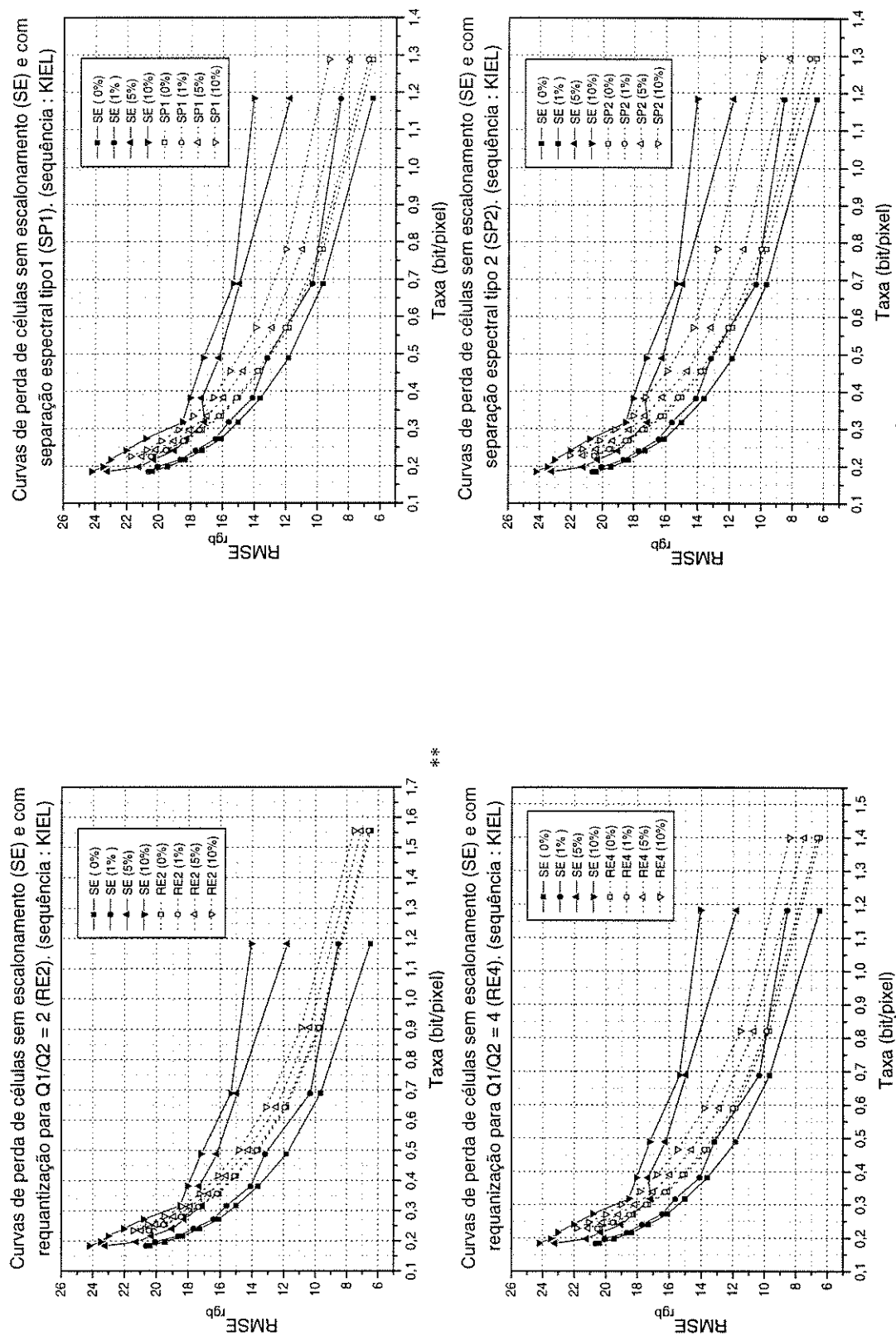


Fig. 4.12 - Comparação entre codificações com e sem escalonamento para um dado nível de perda de células (sequência KIEL).

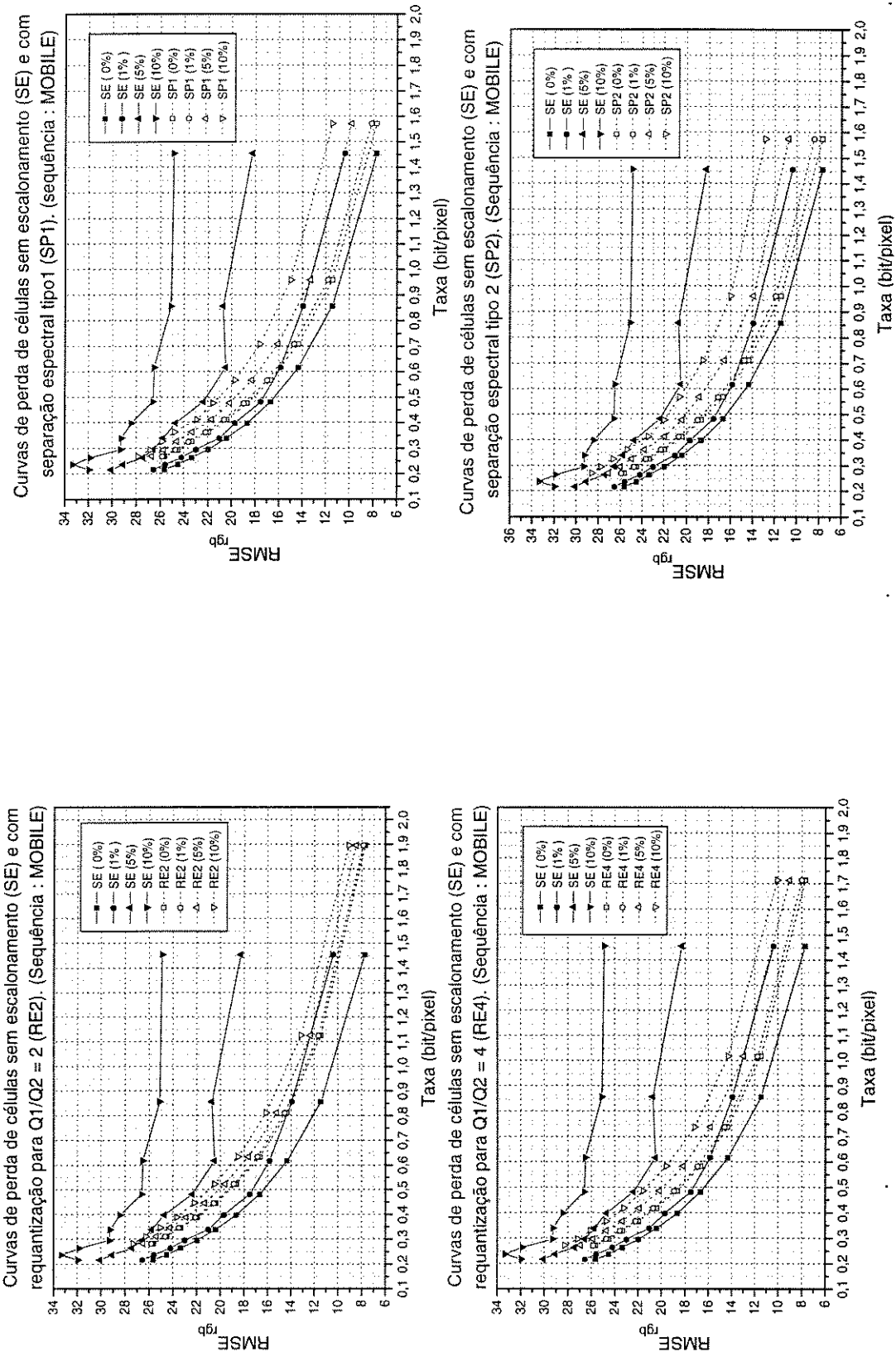


Fig. 4.13 - Comparação entre codificações com e sem escalonamento para um dado nível de perda de células (sequência MOBILE).

4.5 - COMPARAÇÃO ENTRE AS CODIFICAÇÕES COM ESCALONAMENTO

As Figuras 4.14, 4.15 e 4.16, permitem uma comparação entre os dois esquemas de escalonamento (requantização e separação espectral) para as três seqüências utilizadas. Para níveis pequenos de perda, a separação espectral se destaca, já que implica um menor *overhead* se comparado à requantização. À medida que o nível de perda de pacotes aumenta, as curvas de distorção x taxa da separação espectral sofrem um deslocamento mais acentuado que as curvas para requantização, demonstrando uma menor robustez às perdas de pacotes. Neste processo de deslocamento das curvas, elas podem se interceptar. A intersecção ocorre inicialmente em taxas de *bits* mais altas, se deslocando para taxas menores à medida que o nível de perdas de pacotes aumenta. Esta intersecção ocorre para níveis de perda entre 1 e 5% (FLOWER e MOBILE) ou entre 5 e 10% (KIEL). Para níveis superiores de perda de pacotes, a requantização é superior a separação espectral. Para a seqüência KIEL, para níveis de perda entre 5% e 10%, notamos uma grande proximidade entre as curvas de requantização e separação espectral. Se adotarmos esta seqüência como referência, podemos considerar que, para níveis de perda de pacotes superiores a 5%, é preferível transmitir o sinal utilizando requantização, já que apresenta um melhor robustez às perdas. Analogamente, deve-se preferir a separação espectral para níveis de perda de pacotes inferiores a aproximadamente a 5%, onde se pode tirar vantagem do menor *overhead* associado ao escalonamento.

Como comentamos, supomos nessa análise que a quantidade de células geradas pelo codificador não afeta o tráfego total na rede, de modo que supor que o nível de perda de células era constante, independente da taxa de *bits* utilizada. Quando esta suposição não é válida, temos que determinar como o tráfego na rede influencia no nível de perda de células. A Figura 4.17, obtida de [112] mostra este tipo de relação.

No eixo horizontal, tem-se o "carregamento", ou seja, a porcentagem de células em trânsito em relação à capacidade máxima do canal e no eixo vertical tem-se a taxa de perda de células. Apresentam-se gráficos para sinais de vídeo e vídeo e voz

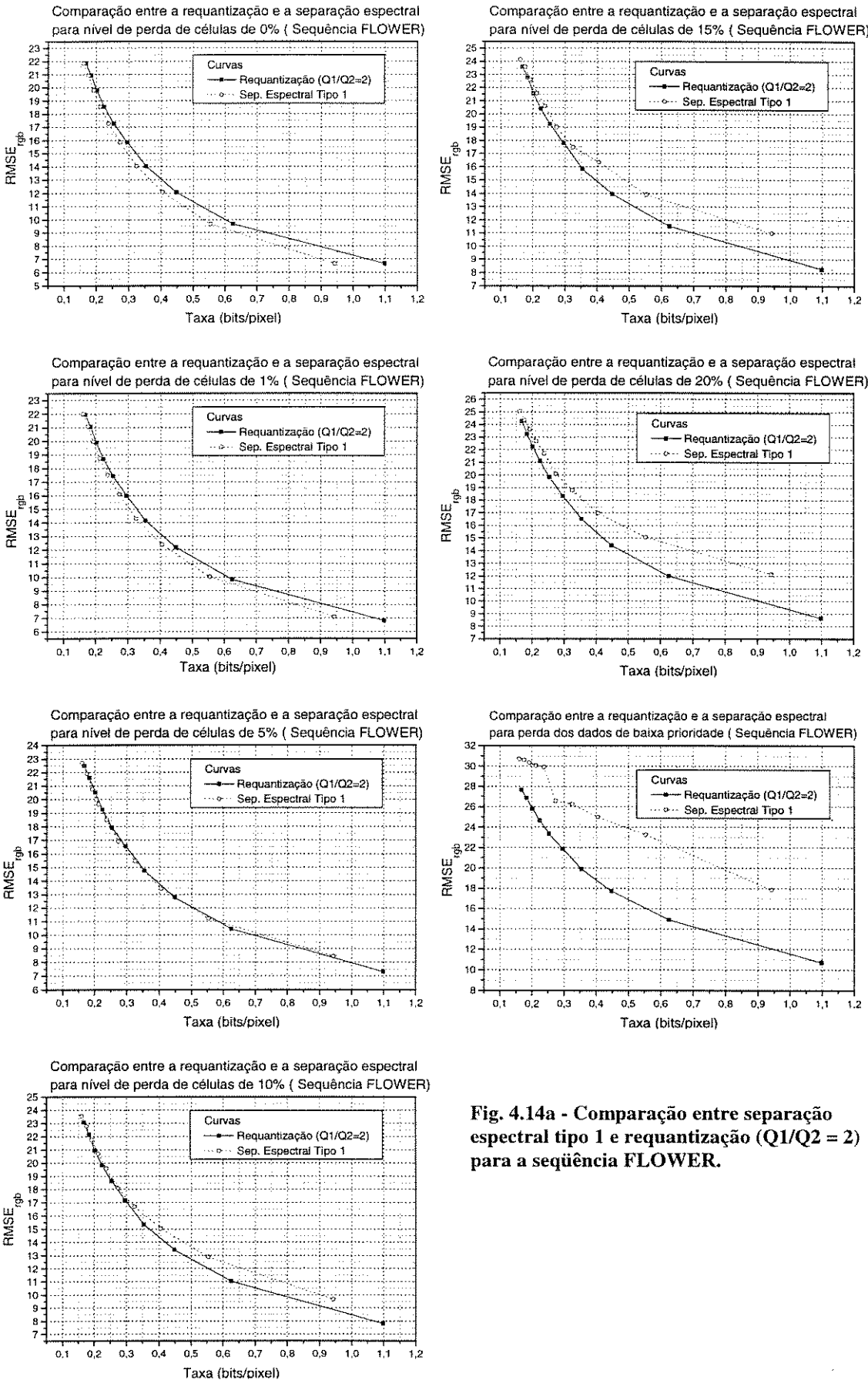


Fig. 4.14a - Compara o entre separa o espectral tipo 1 e requantiza o (Q1/Q2 = 2) para a sequ ncia FLOWER.

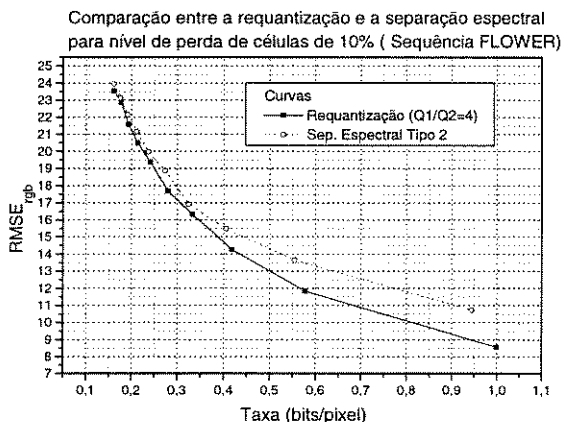
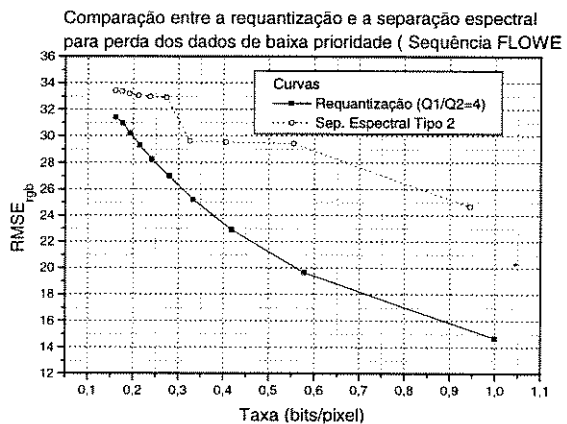
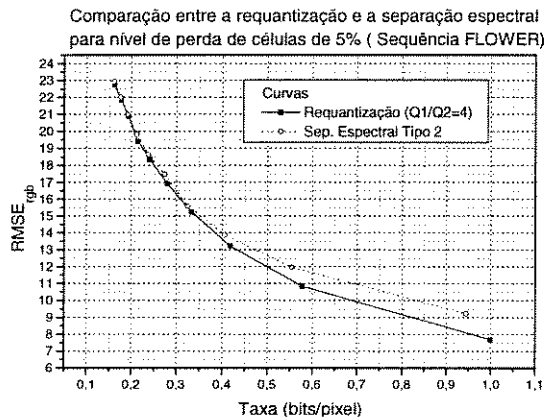
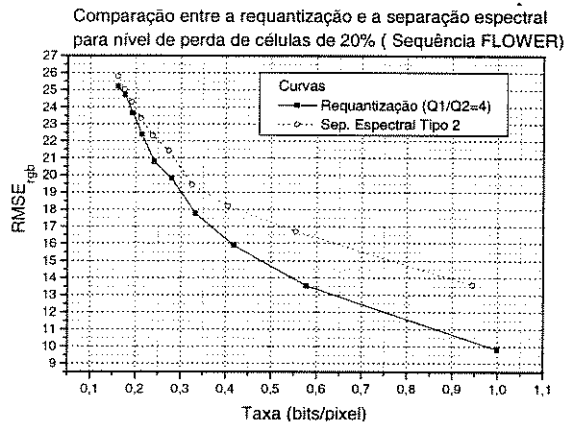
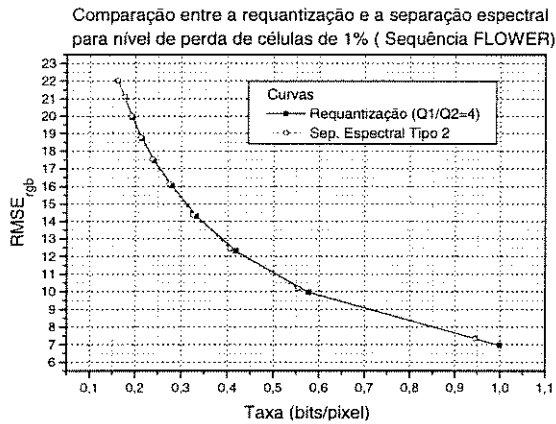
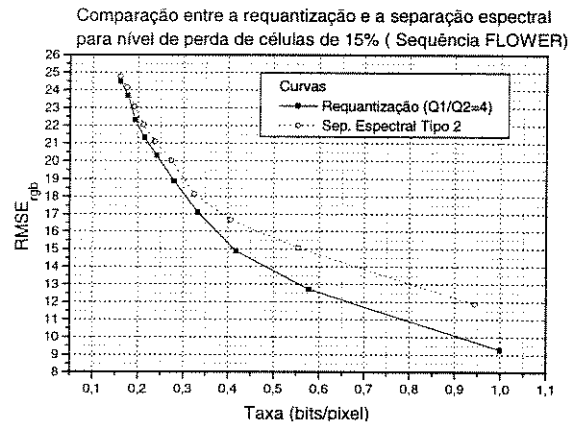
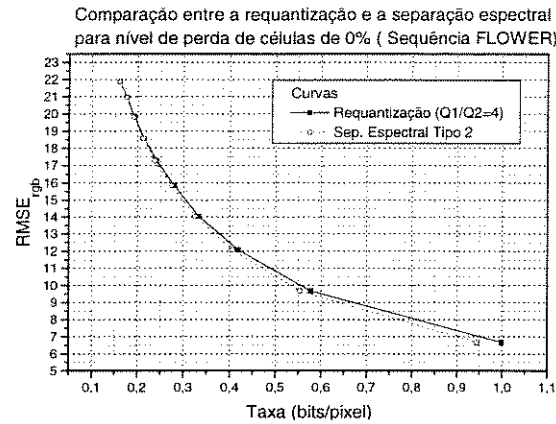
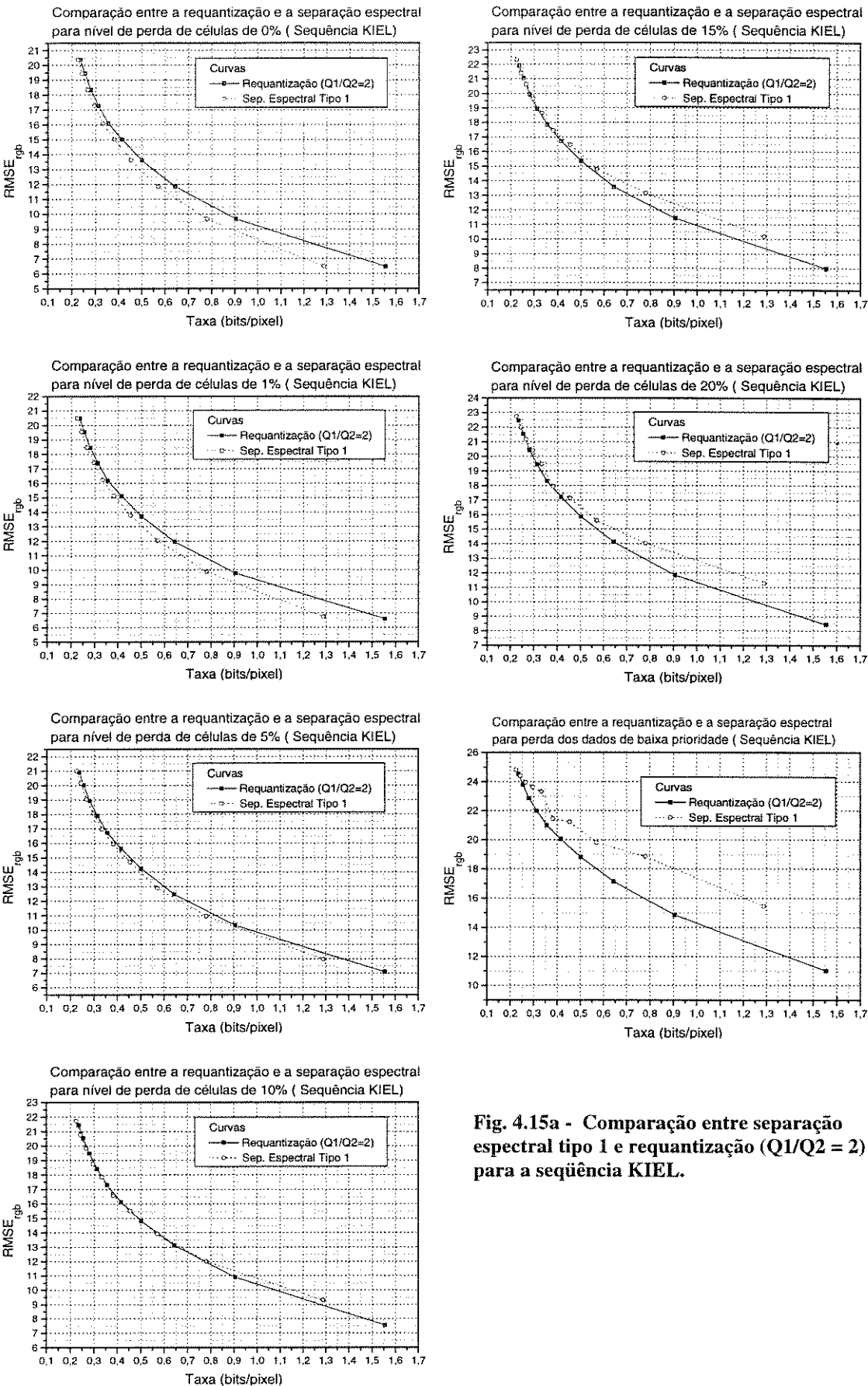


Fig. 4.13b - Comparação entre separação espectral tipo 2 e requantização (Q1/Q2 = 4) para a sequência FLOWER.



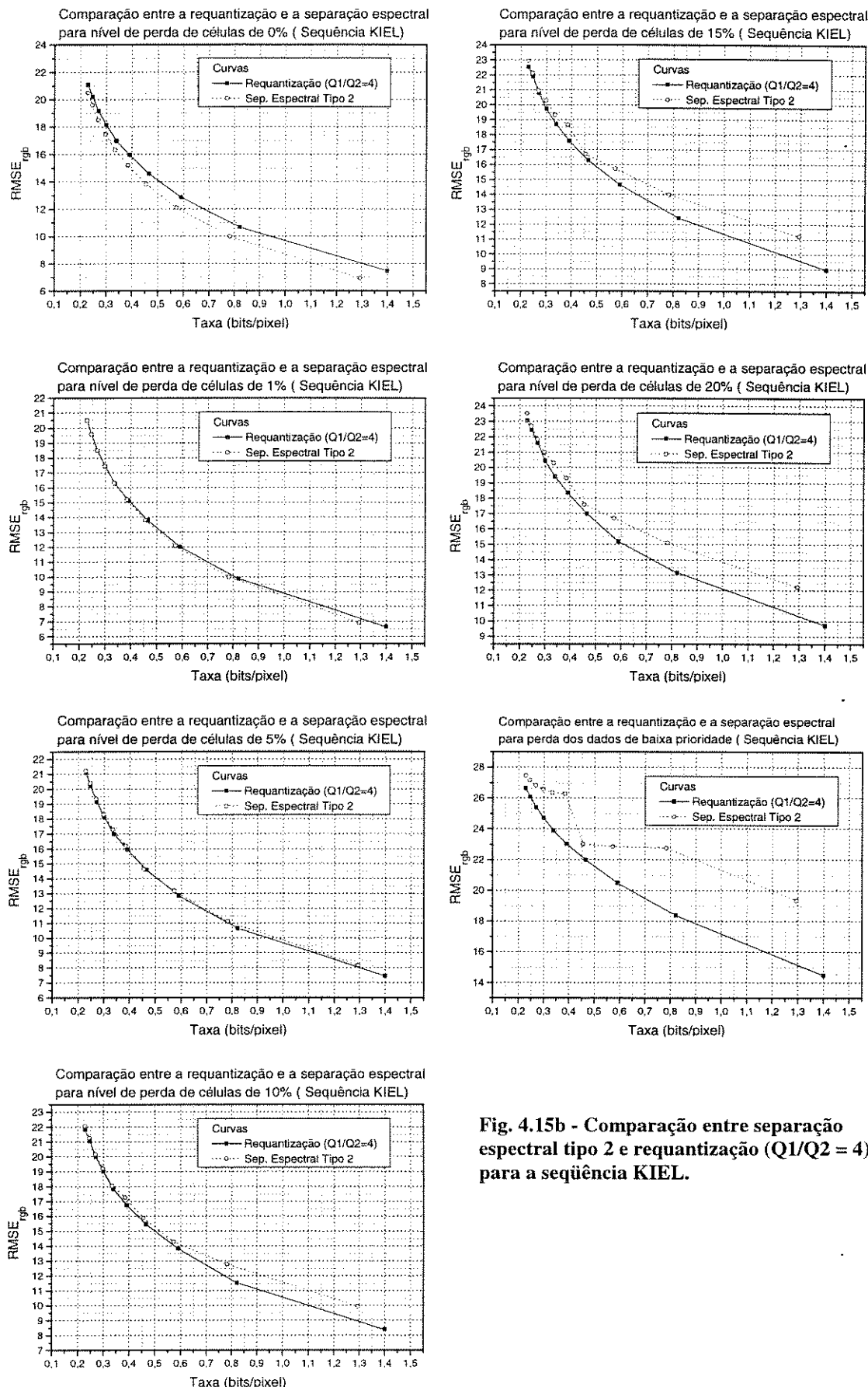


Fig. 4.15b - Comparação entre separação espectral tipo 2 e requantização (Q1/Q2 = 4) para a sequência KIEL.

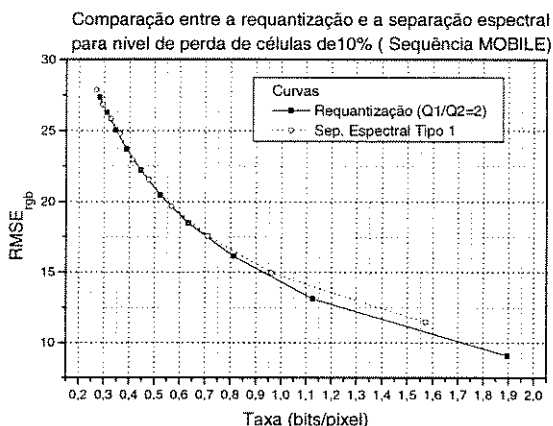
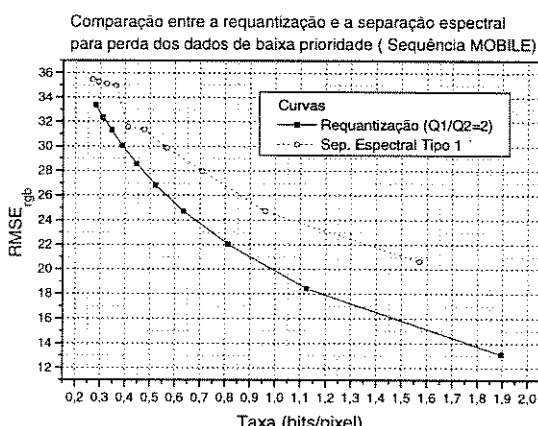
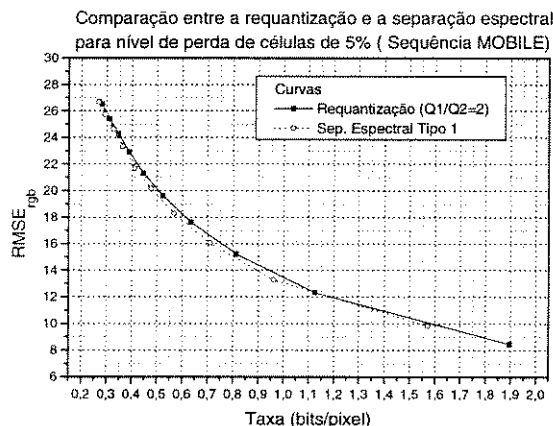
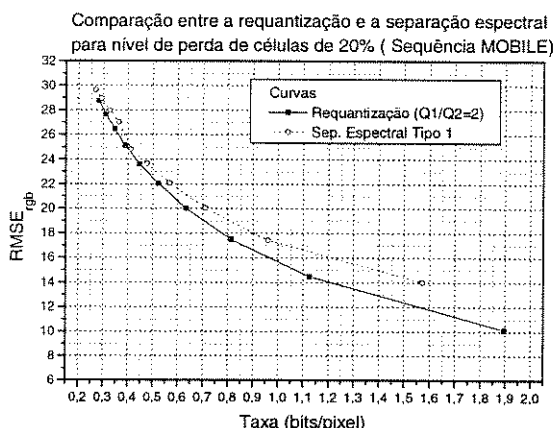
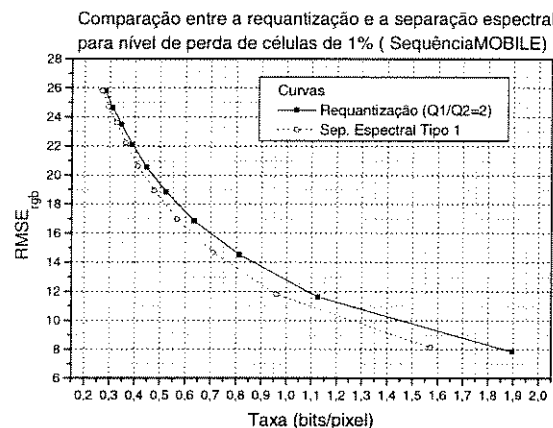
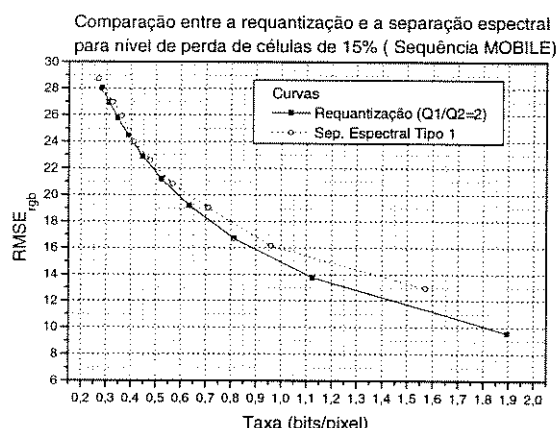
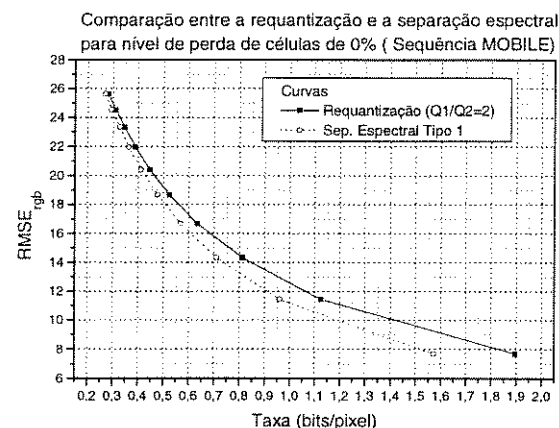


Fig. 4.16a - Comparação entre separação espectral tipo 1 e requantização (Q1/Q2=2) para a sequência MOBILE.

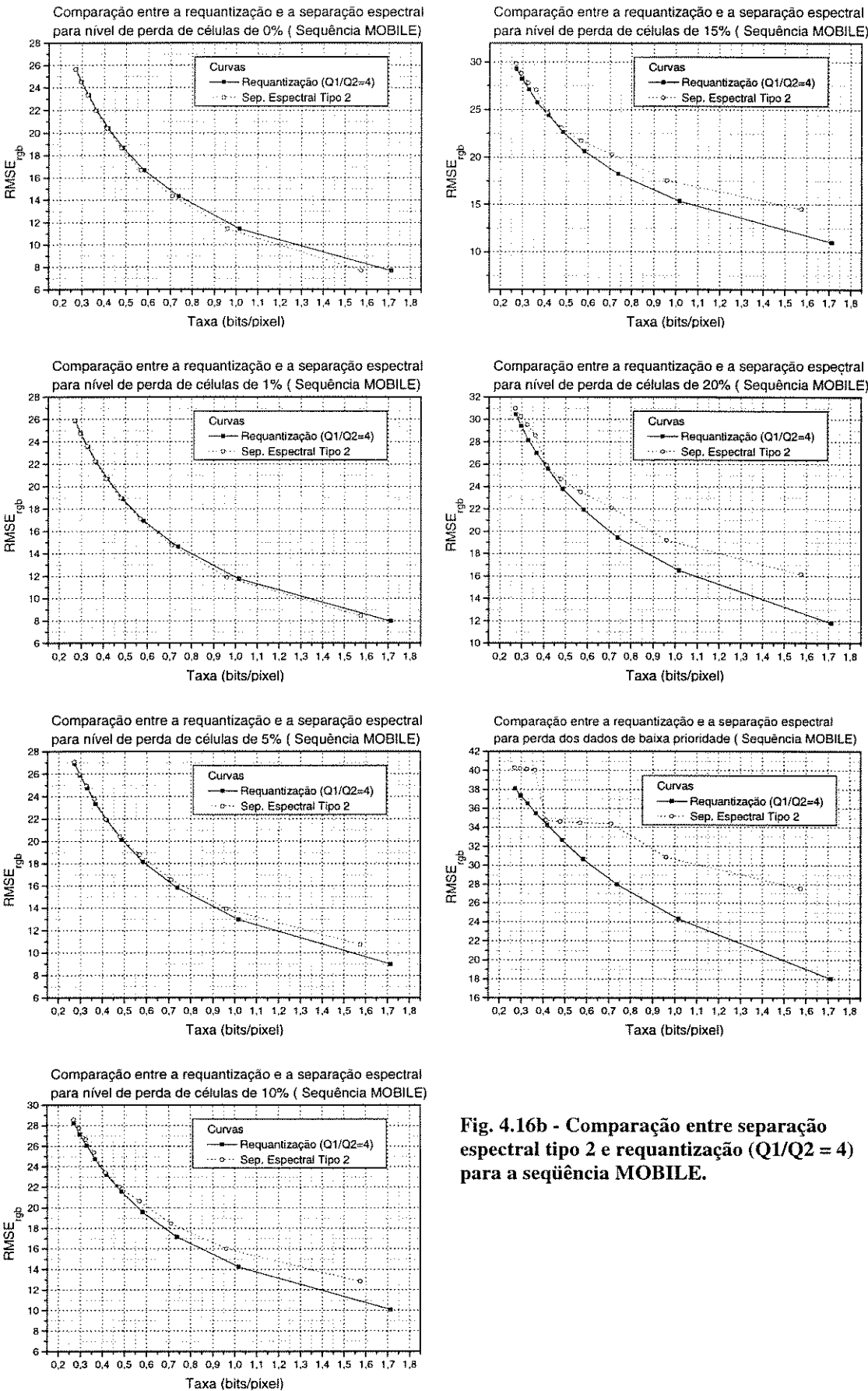


Fig. 4.16b - Comparação entre separação espectral tipo 2 e requantização (Q1/Q2 = 4) para a sequência MOBILE.

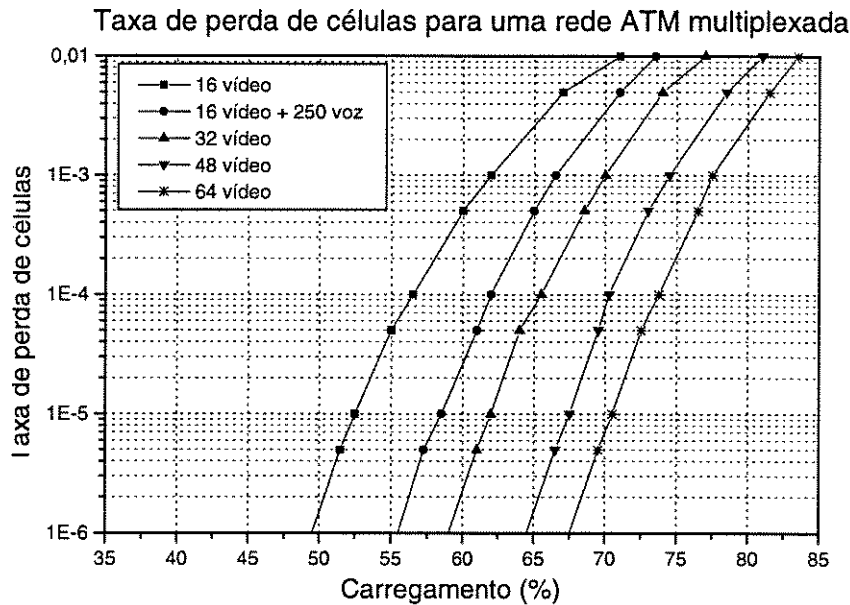


Figura 4.17 - Taxa de perda de células em função do carregamento de uma rede ATM multiplexada.

combinados. Observa-se que o aumento do carregamento é acompanhado por um aumento expressivo da taxa de perdas de células. Suponhamos, por exemplo, que se opera com 16 canais de vídeo e com um carregamento de 50%. Se o carregamento aumentar de 10%, passando a 55%, a taxa de perda de células, segundo o gráfico em [112], sobe de 10^{-6} para 5×10^{-5} , ou seja, um aumento de 4900%.

Supor que o nível de perda de células não é afetado pelo incremento da taxa de *bits* equívale a fazer um corte na superfície do gráfico (em 3D) taxa x distorção x probabilidade de perda de células como nos mostra a Figura 4.18a, isto é, ao longo do plano de perda de células constante. Se considerarmos que o nível de perda é influenciado pelo *overhead* do escalonamento, o corte da superfície em 3D deverá ser efetuado por uma superfície curva determinada pela curva de carregamento da rede (Fir. 4.18b). Deste modo, o gráfico tridimensional que relaciona a distorção (RMSE) e a taxa de *bits* e a probabilidade de perda de células pode ser levantada a partir das curvas de taxa x distorção correspondentes aos níveis constantes de perda de células. Por outro lado, este gráfico deve ser a curva de carregamento característica da rede em uso.

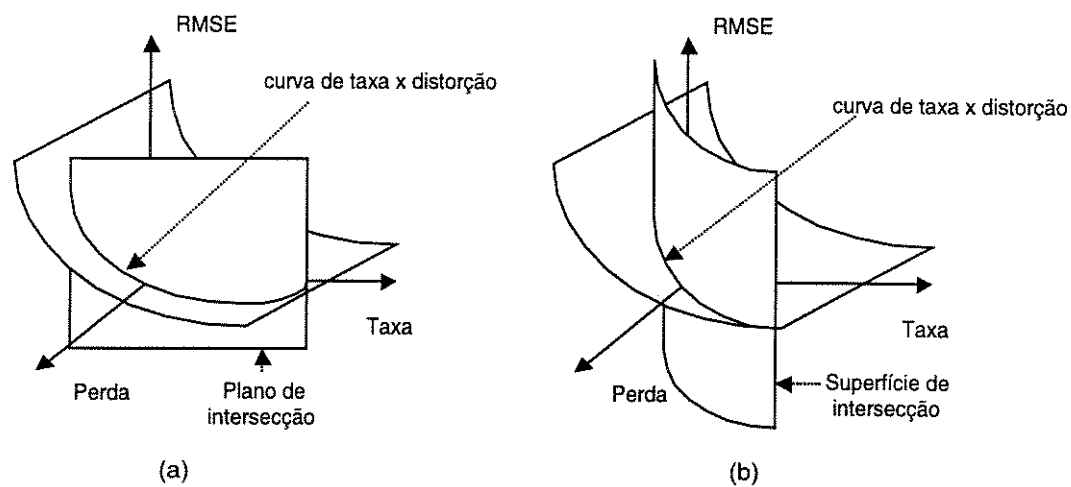


Fig. 4.18 - Determinação das curvas de taxa x distorção

(a). Nível de perda de células constante.

(b) Nível de perda de células variando conforme o 7 carregamento da rede.

Capítulo 5

CONCLUSÕES

Este trabalho investiga a transmissão de sinais de vídeo através de redes de faixa larga com tecnologia de pacotes em ambientes sujeitos a perdas de pacotes. Tais perdas são ocasionadas pelo excesso de tráfego na rede e decorrem da eliminação de células determinada pelo sistema de gerenciamento. De início, revisa-se a teoria da codificação dos sinais de vídeo, abordando-se os aspectos da compressão temporal e espacial e modos de torná-la mais robusta a perdas de células. Argumenta-se que este aumento de robustez pode ser conseguido por meio de escalonamento, técnica que consiste em separar as informações a serem transmitidas em diferentes categorias ou escalões e enviá-las através de canais com níveis diferenciados de proteção a perdas de pacotes. Consideram-se duas técnicas de escalonamento do sinal de vídeo, a saber, separação espectral e requantização. Descreve-se, também, um modelo simples de perda de células para simular o comportamento de uma rede de pacotes submetida a sobrecargas. A partir destes modelos, desenvolve-se um ambiente de simulação do sistema codificador-canál-decodificador, implementado via *software*. Neste ambiente, foram ajustados os parâmetros do codificador e do decodificador para gerar curvas de taxa x distorção similares às obtidas com o padrão MPEG 2. No modelamento das perdas de células, utilizam-se dados reais de seqüências codificadas de vídeo como forma de constituir um modelo simples e realista do efeito de sobrecarga nos nós da rede. As curvas de taxa x distorção obtidas para várias condições de escalonamento são úteis para se estudar o efeito de perdas de células na distorção do sinal recebido e a robustez destes métodos

Inicialmente consideram-se as curvas de taxa x distorção para o caso mais simples, isto é, codificação sem escalonamento e canal sem perda de células, para três seqüências de vídeo (FLOWER, MOBILE e KIEL). Como critério para medida do nível de distorção, utilizou-se a raiz quadrada do erro quadrático médio (RMSE) devido, principalmente, a sua simplicidade quando comparada com a de critérios subjetivos.

Num segundo estágio, introduzem-se níveis de perda de células no canal, onde observamos o efeito destrutivo na qualidade da recepção. Os efeitos mais importantes destas perdas se refletem na propagação dos erros pelos GOP e pela perda de regiões nos quadros com grande quantidade de informação. Na prática, o efeito da perda de pacotes pode ser mais destrutivo, já que supomos nas simulações que não se perdem os sinais de sincronismo. Das simulações realizadas, pode-se deduzir que, salvo em níveis de perda de pacotes muito pequenos (inferiores a 0,5%) não é aconselhável transmitir os sinais de vídeo sem alguma proteção em redes sujeitas a perdas.

Na etapa seguinte, considera-se a técnica de escalonamento do sinal como meio de atenuar o problema das perdas de pacotes. O sinal de vídeo passa a ser dividido em duas camadas: uma camada de alta prioridade, que contém a informação "essencial" (que garante uma qualidade mínima aceitável de recepção), e uma camada de baixa prioridade, com informação de "enriquecimento" (que complementa a informação essencial da primeira camada). O processo de transmissão destas informações se dá por meio de células com prioridade de perdas distintas. Através da informação presente nos cabeçalhos das células (e.g., o *bit* CLP do cabeçalho de células ATM), pode-se distinguir as camadas a que pertencem as células, e determinar as células que serão descartadas quando os acumuladores da rede ficarem saturados.

Implementaram-se dois tipos de escalonamento: requantização e separação espectral. Na requantização dividem-se os *bits* utilizados na representação dos coeficientes de cada bloco da imagem transformado pela DCT em dois conjuntos, os *bits* mais significativos e os menos. Esta divisão dos bits em camadas é efetuada por meio de dois quantizadores, um com coeficientes de quantização maiores para o sinal destinado à camada de alta prioridade e outro com coeficientes de quantização menores para o sinal destinado à camada de baixa prioridade. As matrizes de coeficientes de quantização destes quantizadores são denotados por Q_1 e Q_2 , respectivamente. Se denotarmos $Q_1 = kQ_2$, onde k é uma potência de dois, temos que o número de *bits* por coeficiente DCT enviado pelo canal de baixa prioridade é, em média, $\log_2 k$. Nas simulações realizadas foram adotados os valores de $k=2$ e $k=4$, que correspondem às taxas médias no canal desprotegido de 1 e 2 *bits* por coeficiente, respectivamente.

Na separação espectral, os coeficientes de cada bloco da imagem transformado pela DCT são agrupados em dois conjuntos, segundo um parâmetro de controle que

define a quantidade de coeficientes alocados em cada conjunto. O conjunto dos coeficientes DCT de menor ordem, que caracterizam as componentes de frequência mais baixas, constituem a camada de alta prioridade. Os coeficientes DCT restantes relativos às componentes de frequências mais altas formam a camada de baixa prioridade.

O parâmetro de controle da separação espectral é ajustado de modo a igualar a porcentagem de células geradas na camada de baixa prioridade (em relação ao total de células) à porcentagem correspondente gerada pelo método de requantização. Este ajuste faz com que a mesma proporção de taxas nas camadas de alta e baixa prioridade seja observada nos dois métodos de escalonamento, tornando possível a comparação entre os mesmos. Assim sendo, são produzidos dois tipos de separação espectral: tipo 1 (correspondente a requantização com $k=2$) e tipo 2 (correspondente a requantização com $k=4$).

Submetendo estes esquemas de codificação com escalonamento a perdas de pacotes e analisando as curvas taxa x distorção resultantes, observa-se que os esquemas passam a apresentar maior robustez a perda de células, já que as curvas de desempenho sofrem pequenos deslocamentos quando o nível de perdas aumenta. Observa-se também o aparecimento de um custo de *overhead* devido ao esquema de escalonamento, que eleva a taxa de transmissão em relação à taxa do caso sem escalonamento. Portanto, os esquemas que utilizam escalonamento aumentam a robustez a perdas e associam um *overhead* às taxas do sinal transmitido. Este *overhead* provoca um deslocamento para a direita das curvas de taxa x distorção. Dependendo da probabilidade de perdas de células, este deslocamento pode levar os esquemas de escalonamento a apresentarem um desempenho inferior ao obtido sem escalonamento.

Comparando as curvas de taxa x distorção, verifica-se que para níveis de perda de células superiores a 1%, é aconselhável a utilização de escalonamento, pois tem-se uma menor distorção para cada taxa de transmissão. Para níveis inferiores de perdas de células, as curvas se interceptam, demonstrando a vantagem do uso de escalonamento apenas nas regiões de altas taxas de transmissão.

Considerando a utilidade do escalonamento quando os níveis de perda de células são superiores a 1%, comparamos os desempenhos dos esquemas com requantização e separação espectral. Observa-se que para níveis de perda de células entre 1% e 5% é preferível utilizar o escalonamento por separação espectral que está

associado a menores custos de *overhead*. Quando os níveis de perda de células excedem a marca de 5%, a maior robustez proporcionada pela requantização faz com que este esquema apresente o melhor desempenho, a despeito de apresentar um maior de *overhead*. A Figura 5.1 mostra um resumo dessas observações.

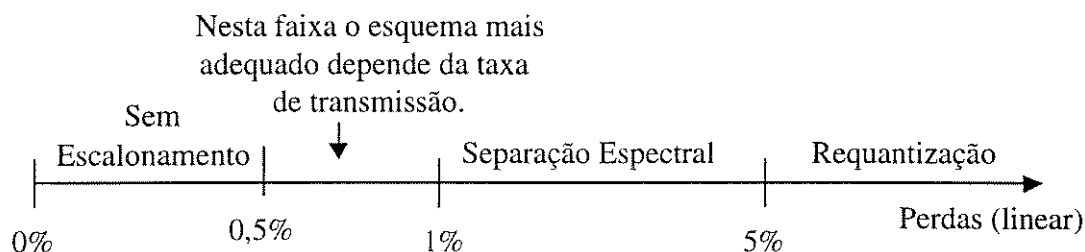


Fig. 5.1 - Esquemas de codificação mais apropriados para os diversos níveis de perdas de células.

Na requantização, o aumento da relação $k = Q1/Q2$ de 2 para 4 tem o efeito de elevar o número médio de *bits* transmitidos pelo canal de baixa prioridade. Isto ocasiona uma maior sensibilidade à perda de células (refletindo-se em um deslocamento mais acentuado das curvas taxa x distorção com o aumento do nível de perda de células) pois temos menos informação “protegida” enviada pelo canal de alta prioridade. Nas simulações efetuadas, para níveis de perdas de células inferiores a 20%, verifica-se um melhor desempenho do método de requantização com $k = Q1/Q2 = 4$ em comparação com $k = 2$. Quando se utilizam as mesmas matrizes de quantização fina ($Q2$), os dois esquemas (para $k = 2$ e $k = 4$) apresentam distorções idênticas em canais sem perdas. Portanto, a diferença observada nas curvas taxa x distorção demonstra a existência de um menor *overhead* de taxa no método com $k = 4$. Por outro lado, um maior valor de k permite a operação em ambientes com maiores probabilidades de perdas de células sem que células de alta prioridade precisem ser sacrificadas.

Na separação espectral, não observamos diferenças significativas nos casos analisados (tipos 1 e 2).

Nas considerações acima, supomos que o efeito da perda de células não afeta a transmissão dos sinais de sincronismo. Caso se verifique a perda deste sincronismo, o aumento da distorção pode ser atenuado tornando ainda mais recomendável o uso de escalonamento.

Quando adotamos a hipótese de que o número de células gerado por nosso codificador não altera o tráfego da rede, simplificamos implicitamente a análise do problema transformando-o em um problema de duas dimensões, i.e., na determinação das curvas de taxa x distorção através do corte por um plano definido por um nível de perdas de células constante no espaço tridimensional taxa x distorção x nível de perdas. Quando o nível de perdas depende da taxa, a determinação das curvas de taxa x distorção não pode mais ser feita por um plano, mais sim por uma superfície curva, que depende das características do canal (probabilidade de perda de células x carregamento na rede), que deve estar disponível para a escolha do esquema de codificação mais apropriado.

Para trabalhos futuros, sugerimos um estudo mais detalhado de métodos de escalonamento de vídeo que apresentem melhor desempenho em termos de taxa x distorção. Como vimos, o custo do escalonamento, ou *overhead*, tem um impacto no desempenho do codificador. Em especial, pode-se reduzir este custo e melhorar o codificador através do aperfeiçoamento dos cabeçalhos de identificação dos dados de alta e baixa prioridade. Além disto, um estudo sobre a relação entre o RMSE e a qualidade subjetiva poderia levar a uma melhor compreensão do processo de distorção, sugerindo um melhor esquema de escalonamento. O modelamento do canal sujeito a perdas de células também pode ser aprimorado, por exemplo, para melhor caracterizar os efeitos dos acumuladores (*buffers*) nos nós da rede.

APÊNDICE A

Tabela A.1 - Matrizes de quantização utilizadas pelo codificador de fonte.

- **constante**

```

1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1

```

- **mpeg_intra**

```

8 16 19 22 26 27 29 34
16 16 22 24 27 29 34 37
19 22 26 27 29 34 34 38
22 22 26 27 29 34 37 40
22 26 27 29 32 35 40 48
26 27 29 32 35 40 48 58
26 27 29 34 38 46 56 69
27 29 35 38 46 56 69 83

```

- **mpeg_inter**

```

16 16 16 16 16 16 16 16
16 16 16 16 16 16 16 16
16 16 16 16 16 16 16 16
16 16 16 16 16 16 16 16
16 16 16 16 16 16 16 16
16 16 16 16 16 16 16 16
16 16 16 16 16 16 16 16
16 16 16 16 16 16 16 16

```

- **jpeg_y**

```

16 11 10 16 24 40 51 61
12 12 14 19 26 58 60 55
14 13 16 24 40 57 69 56
14 17 22 29 51 87 80 62
18 22 37 56 68 109 103 77
24 35 55 64 81 104 113 92
49 64 78 87 103 121 120 101
72 92 95 98 112 100 103 99

```

- **jpeg_c**

```

17 18 24 47 99 99 99 99
18 21 26 66 99 99 99 99
24 26 56 99 99 99 99 99
47 66 99 99 99 99 99 99
99 99 99 99 99 99 99 99
99 99 99 99 99 99 99 99
99 99 99 99 99 99 99 99
99 99 99 99 99 99 99 99

```

Tabela A.2 - Filtros utilizados na conversão das componentes (R,G,B) para (Y,Cr,Cb).

- **Filtro Horizontal**

[1 2 1]

- **Filtro de Reconstrução Horizontal**

[1 2 1]

- **Filtro Vertical**

[5 11 -21 -37 70 228 228 70 -37 -21 11 5]

- **Filtro de Reconstrução Vertical**

[3 7 -16 -32 67 227 227 67 -32 -16 7 3]

Tabela A.3 - Coeficientes de agrupamento para os elementos DC e AC (JPEG).

Faixa	Categoria (Diferença DC)	Categoria AC
0	0	N/A
-1,1	1	1
-3,-2,2,3	2	2
-7,...,-4,4,...,7	3	3
-15,...,-8,8,...,15	4	4
-31,...,-16,16,...,31	5	5
-63,...,-32,32,...,63	6	6
-127,...,-64,64,...,127	7	7
-255,...,-128,128,...,255	8	8
-511,...,-256,256,...,511	9	9
-1023,...,-512,512,...,1023	10	10

Tabela A.4 - Códigos de comprimento variável para as componentes de luminância.

a) Diferença DC de luminância.

Categoria	Palavra Código
0	00
1	010
2	011
3	100
4	101
5	110
6	1110
7	11110
8	111110
9	1111110
A	11111110
B	111111110

b) Coeficientes AC de luminância.

Corrida	Categ.	Palavra Código
0	0	1010
0	1	00
0	2	01
0	3	100
0	4	1011
0	5	11010
0	6	111000
0	7	1111000
0	8	1111110110
0	9	111111110000010
0	A	111111110000011
1	1	1100
1	2	111001
1	3	1111001
1	4	111110110
1	5	11111110110
1	6	111111110000100
1	7	111111110000101
1	8	111111110000110
1	9	111111110000111
1	A	111111110001000
2	1	11011
2	2	11111000
2	3	1111110111
2	4	111111110001001
2	5	111111110001010
2	6	111111110001011
2	7	111111110001100
2	8	111111110001101
2	9	111111110001110
2	A	111111110001111
3	1	111010
3	2	111110111
3	3	11111110111

3	4	1111111110010000
3	5	1111111110010001
3	6	1111111110010010
3	7	1111111110010011
3	8	1111111110010100
3	9	1111111110010101
3	A	1111111110010110
4	1	111011
4	2	1111111000
4	3	1111111110010111
4	4	1111111110011000
4	5	1111111110011001
4	6	1111111110011010
4	7	1111111110011011
4	8	1111111110011100
4	9	1111111110011101
4	A	1111111110011110
5	1	1111010
5	2	1111111001
5	3	1111111110011111
5	4	1111111110100000
5	5	1111111110100001
5	6	1111111110100010
5	7	1111111110100011
5	8	1111111110100100
5	9	1111111110100101
5	A	1111111110100110
6	1	1111011
6	2	11111111000
6	3	1111111110100111
6	4	1111111110101000
6	5	1111111110101001
6	6	1111111110101010
6	7	1111111110101011
6	8	1111111110101100
6	9	1111111110101101
6	A	1111111110101110
7	1	11111001
7	2	11111111001
7	3	1111111110101111
7	4	1111111110110000
7	5	1111111110110001
7	6	1111111110110010
7	7	1111111110110011
7	8	1111111110110100
7	9	1111111110110101
7	A	1111111110110110
8	1	11111010
8	2	1111111110000000
8	3	1111111110110111
8	4	1111111110111000
8	5	1111111110111001
8	6	1111111110111010
8	7	1111111110111011

8	8	1111111110111100
8	9	1111111110111101
8	A	1111111110111110
9	1	111111000
9	2	1111111110111111
9	3	1111111111000000
9	4	1111111111000001
9	5	1111111111000010
9	6	1111111111000011
9	7	1111111111000100
9	8	1111111111000101
9	9	1111111111000110
9	A	1111111111000111
A	1	111111001
A	2	1111111111001000
A	3	1111111111001001
A	4	1111111111001010
A	5	1111111111001011
A	6	1111111111001100
A	7	1111111111001101
A	8	1111111111001110
A	9	1111111111001111
A	A	1111111111010000
B	1	111111010
B	2	11111111111010001
B	3	11111111111010010
B	4	11111111111010011
B	5	11111111111010100
B	6	11111111111010101
B	7	11111111111010110
B	8	11111111111010111
B	9	11111111111011000
B	A	11111111111011001
C	1	1111111010
C	2	11111111111011010
C	3	11111111111011011
C	4	11111111111011100
C	5	11111111111011101
C	6	11111111111011110
C	7	11111111111011111
C	8	1111111111100000
C	9	1111111111100001
C	A	1111111111100010
D	1	1111111010
D	2	1111111111100011
D	3	1111111111100100
D	4	1111111111100101
D	5	1111111111100110
D	6	1111111111100111
D	7	1111111111101000
D	8	1111111111101001
D	9	1111111111101010
D	A	1111111111101011
E	1	11111110110
E	2	1111111111101100
E	3	1111111111101101
E	4	1111111111101110
E	5	1111111111101111

E	6	1111111111110000
E	7	1111111111110001
E	8	1111111111110010
E	9	1111111111110011
E	A	1111111111110100
F	0	111111110111
F	1	1111111111110101
F	2	1111111111110110
F	3	1111111111110111
F	4	1111111111111000
F	5	1111111111111001
F	6	1111111111111010
F	7	1111111111111011
F	8	1111111111111100
F	9	1111111111111101
F	A	1111111111111110

Tabela A.5 - Códigos de comprimento variável para as componentes de cromaticidade.

a) Diferença DC de luminância.

Categoria	Palavra Código
0	00
1	01
2	10
3	110
4	1110
5	11110
6	111110
7	1111110
8	11111110
9	111111110
A	1111111110
B	11111111110

b) Coeficiente AC de cromaticidade.

Corrida	Categ.	Palavra Código
0	0	00
0	1	01
0	2	100
0	3	1010
0	4	11000
0	5	11001
0	6	111000
0	7	1111000
0	8	111110100
0	9	1111110110
0	A	111111110100
1	1	1011
1	2	111001
1	3	11110110
1	4	111110101
1	5	11111110110
1	6	111111110101
1	7	1111111110001000
1	8	1111111110001001
1	9	1111111110001010
1	A	1111111110001011
2	1	11010
2	2	11110111
2	3	1111110111
2	4	111111110110
2	5	111111111000010
2	6	1111111110001100
2	7	1111111110001101
2	8	1111111110001110
2	9	1111111110001111
2	A	1111111110010000

3	1	11011
3	2	11111000
3	3	1111111000
3	4	111111110111
3	5	1111111110010001
3	6	1111111110010010
3	7	1111111110010011
3	8	1111111110010100
3	9	1111111110010101
3	A	1111111110010110
4	1	111010
4	2	111110110
4	3	1111111110010111
4	4	1111111110011000
4	5	1111111110011001
4	6	1111111110011010
4	7	1111111110011011
4	8	1111111110011100
4	9	1111111110011101
4	A	1111111110011110
5	1	111011
5	2	1111111001
5	3	1111111110011111
5	4	1111111110100000
5	5	1111111110100001
5	6	1111111110100010
5	7	1111111110100011
5	8	1111111110100100
5	9	1111111110100101
5	A	1111111110100110
6	1	1111001
6	2	11111110111
6	3	1111111110100111
6	4	1111111110101000
6	5	1111111110101001
6	6	1111111110101010
6	7	1111111110101011
6	8	1111111110101100
6	9	1111111110101101
6	A	1111111110101110
7	1	1111010
7	2	11111111000
7	3	1111111110101111
7	4	1111111110110000
7	5	1111111110110001
7	6	1111111110110010
7	7	1111111110110011
7	8	1111111110110100
7	9	1111111110110101
7	A	1111111110110110
8	1	11111001
8	2	1111111110110111
8	3	1111111110111000

8	4	1111111110111001
8	5	1111111110111010
8	6	1111111110111011
8	7	1111111110111100
8	8	1111111110111101
8	9	1111111110111110
8	A	1111111110111111
9	1	111110111
9	2	1111111111000000
9	3	1111111111000001
9	4	1111111111000010
9	5	1111111111000011
9	6	1111111111000100
9	7	1111111111000101
9	8	1111111111000110
9	9	1111111111000111
9	A	1111111111001000
A	1	111111000
A	2	1111111111001001
A	3	1111111111001010
A	4	1111111111001011
A	5	1111111111001100
A	6	1111111111001101
A	7	1111111111001110
A	8	1111111111001111
A	9	1111111111010000
A	A	1111111111010001
B	1	111111001
B	2	1111111111010010
B	3	1111111111010011
B	4	1111111111010100
B	5	1111111111010101
B	6	1111111111010110
B	7	1111111111010111
B	8	1111111111011000
B	9	1111111111011001
B	A	1111111111011010
C	1	111111010
C	2	1111111111011011
C	3	1111111111011100
C	4	1111111111011101
C	5	1111111111011110
C	6	1111111111011111
C	7	1111111111000000
C	8	1111111111000001
C	9	1111111111000010
C	A	1111111111000011
D	1	11111111001
D	2	1111111111100100
D	3	1111111111100101
D	4	1111111111100110
D	5	1111111111100111
D	6	1111111111101000
D	7	1111111111101001
D	8	1111111111101010
D	9	1111111111101011
D	A	1111111111101100
E	1	11111111100000

E	2	1111111111101101
E	3	1111111111101110
E	4	1111111111101111
E	5	1111111111100000
E	6	1111111111100001
E	7	1111111111100010
E	8	1111111111100011
E	9	1111111111101000
E	A	1111111111101001
F	0	1111111010
F	1	1111111110000011
F	2	1111111111110110
F	3	1111111111110111
F	4	1111111111111000
F	5	1111111111111001
F	6	1111111111111010
F	7	1111111111111011
F	8	1111111111111100
F	9	1111111111111101
F	A	1111111111111110

UNICAMP

BIBLIOTECA CENTRAL

rua João Goulart, 100 - 13081-907 - Campinas, SP

Tabela A.6 - Códigos de comprimento variável para os vetores de movimento.

Código de Comprimento Variável	Índice do vetores de movimento.
0000 0011 001	-16
0000 0011 011	-15
0000 0011 101	-14
0000 0011 111	-13
0000 0100 001	-12
0000 0100 011	-11
0000 0100 11	-10
0000 0101 01	-9
0000 0101 11	-8
0000 0111	-7
0000 1001	-6
0000 1011	-5
0000 111	-4
0000 1	-3
0011	-2
011	-1
1	0
010	1
0010	2
0001 0	3
0000 110	4
0000 1010	5
0000 1000	6
0000 0110	7
0000 0101 10	8
0000 0101 00	9
0000 0100 10	10
0000 0100 010	11
0000 0100 000	12
0000 0011 110	13
0000 0011 100	14
0000 0011 010	15
0000 0011 000	16

Tabela A.7 - Código de comprimento variável para os macroblocos nulos.

Número de macroblocos nulos	Código
0	1
1	00000000
2	00000001
3	00000010
4	00000011
5	00000100
6	00000101
7	00000110
8	00000111
9	00001000
10	00001001
11	00001010
12	00001011
13	00001100
14	00001101
15	00001110
16	00001111
17	00010000
18	00010001
19	00010010
20	00010011
21	00010100
22	00010101
23	00010110
24	00010111
25	00011000
26	00011001
27	00011010
28	00011011
29	00011100
30	00011101
31	00011110
32	00011111
33	00100000
34	00100001
35	00100010
36	00100011
37	00100100
38	00100101
39	00100110

40	00100111
41	00101000
42	00101001
43	00101010
44	00101011
45	01

Algoritmo para o cálculo da DCT (IDCT)

Se implementarmos a DCT ou a IDCT segundo suas definições [2], encontramos as seguintes dificuldades:

- 1) O número de operações necessárias é da ordem de N^2 , com muitos cálculos repetitivos, devido principalmente a periodicidade das funções co-seno para uma dada combinação dos argumentos (j,u) ou (k,v).
- 2) A DCT apresenta cálculos que demandam muito tempo (funções co-seno e multiplicações).

Visando resolver estes problemas, vários algoritmos que implementam a DCT ou a IDCT de forma rápida têm sido sugeridos [43-47]. Neste trabalho, adotamos o algoritmo formulado em [47], que apresenta desempenho superior quando comparado com algoritmos similares [47]. A idéia básica do algoritmo é efetuar o mínimo de cálculos possível com as funções co-seno, aproveitando sua periodicidade, ao mesmo tempo que se evitam as multiplicações, substituindo-as por operações de soma e subtração. A seguir descreves-se o algoritmo.

O Algoritmo:

Denotando por $D(k)$ a DCT de uma seqüência $x(n)$, $n = 0, 1, 2, \dots, N-1$ (N par) tem-se, em uma dimensão

$$D(k) = \sqrt{\frac{2}{N}} e(k) \sum_{n=0}^{N-1} x(n) \cos \left[\frac{(2n+1)}{2N} \right] \quad (1)$$

e

$$d(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} e(n) D(k) \cos \left[\frac{(2n+1)\pi k}{2N} \right], \quad (2)$$

onde

$$e(k) = \begin{cases} \frac{1}{\sqrt{2}} & k = 0 \\ 1 & k \neq 0 \end{cases} \quad (3)$$

Consideremos o algoritmo para a DCT (o algoritmo para a IDCT é facilmente dedutível a partir deste).

Sejam

$$C_{2N}^{(2n+1)k} = \cos\left[\frac{(2n+1)\pi k}{2N}\right] \quad (4)$$

e

$$X(k) = \sqrt{\frac{N}{2}} \frac{1}{e(k)} D(k) \quad (5)$$

Assim, obtemos a equação utilizada para a formulação do algoritmo, qual seja,

$$X(k) = \sum_{n=0}^{N-1} x(n) C_{2N}^{(2n+1)k} \quad (6)$$

Invertendo a ordem do somatório de (6), chega-se a

$$X(k) = \sum_{n=0}^{N/2-1} \left[x(n) C_{2N}^{(2n+1)k} + x(N-n-1) C_{2N}^{[2(N-n-1)+1]k} \right]. \quad (7)$$

Como

$$\begin{aligned} C_{2N}^{[2(N-n-1)+1]k} &= \cos\left\{\frac{[2(N-n-1)+1]k}{2N}\right\} = \cos\left[k - \frac{(2n+1)k}{2N}\right] = \\ &= (-1)^k \cos\left[\frac{(2n+1)k}{2N}\right] = (-1)^k C_{2N}^{(2n+1)k} \end{aligned} \quad (8)$$

Tem-se que $X(k)$ é dado por

$$X(k) = \sum_{n=0}^{N-1} [x(n) + x(N-n-1)(-1)^k] C_{2N}^{(2n+1)k}. \quad (9)$$

Separemos agora $X(k)$ em suas partes par e ímpar:

Parte par:

Fazendo $k = 2m$, tem-se

$$X(2m) = \sum_{n=0}^{N/2-1} [x(n) + x(N-n-1)(-1)^m] C_N^{(2n+1)m} \quad (10)$$

Como $(-1)^{2m} = 1$ e

$$C_{2N}^{(2n+1)2m} = \cos\left[\frac{(2n+1)\pi 2m}{2N}\right] = \cos\left[\frac{(2n+1)\pi m}{N}\right] = C_N^{(2n+1)m}, \quad (11)$$

Chega-se a

$$X(2m) = \sum_{n=0}^{N/2-1} [x(n) + x(N-n-1)] C_N^{(2n+1)m} \quad (12)$$

Denotando $x(n) + x(N-n-1)$ por $g(n)$ obtêm-se

$$X(2m) = \sum_{n=0}^{N/2-1} g(n) C_N^{(2n+1)m} = G(k) \quad (13)$$

para $m \in [0..N/2-1]$.

Observando a Equação (13) notamos que ela é idêntica à Equação (6), sendo calculada em apenas $N/2$ pontos.

Calculemos agora a parte ímpar de $X(k)$.

Parte ímpar:

Considerando $k = 2m+1$, tem-se

$$X(2m+1) = \sum_{n=0}^{N/2-1} [x(n) + x(N-n-1)(-1)^{2m+1}] \cdot C_{2N}^{(2n+1)(2m+1)}. \quad (14)$$

como $(-1)^{2m+1} = -1$, pode-se escrever

$$X(2m+1) = \sum_{n=0}^{N/2-1} [x(n) - x(N-n-1)] C_{2N}^{(2n+1)(2m+1)}. \quad (15)$$

Multiplicando (15) por $2C_{2N}^{(2m+1)}$, obtêm-se

$$2 \cdot C_{2N}^{(2m+1)} \cdot X(2m+1) = \sum_{n=0}^{N/2-1} [x(n) - x(N-n-1)] \cdot C_{2N}^{(2n+1)(2m+1)} \cdot 2 \cdot C_{2N}^{(2m+1)} \quad (16)$$

$$\text{Observando que } 2C_{2N}^A \cdot C_{2N}^B = C_{2N}^{(A+B)} \cdot C_{2N}^{(A-B)}, \quad (17)$$

$$\text{e chamando } u(n) = x(n) - x(N-n-1), \quad (18)$$

a Equação (16) se transforma em

$$2C_{2N}^{(2m+1)} X(2m+1) = \sum_{n=0}^{N/2-1} u(n) C_{2N}^{(2n+1)2m} + \sum_{n=0}^{N/2-1} u(n) C_{2N}^{(2n+1)2(m+1)} \quad (19)$$

Isolando $X(2m+1)$ em (19) chega-se a

$$X(2m+1) = \underbrace{\sum_{n=0}^{N/2-1} \frac{u(n)}{2C_{2N}^{(2m+1)}} C_{2N}^{(2n+1)2m}}_A + \underbrace{\sum_{n=0}^{N/2-1} \frac{u(n)}{2C_{2N}^{(2m+1)}} C_{2N}^{(2n+1)2(m+1)}}_B \quad (20)$$

Da Equação (20) nota-se que, se definirmos

$$h(n) = \frac{u(n)}{C_{2N}^{(2m+1)}} \quad (21)$$

a parte (A) desta equação tem a forma similar a da Equação (13). Para a parte (B) vale a mesma observação, com $m+1$ no lugar de m . Logo, pode-se reescrever (20) da forma

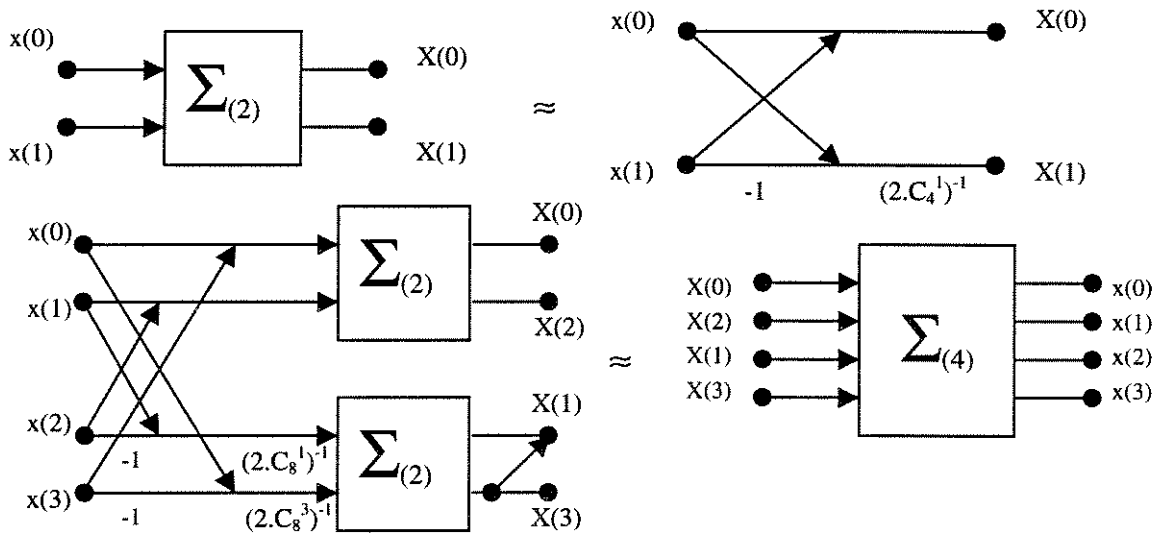
$$X(2m+1) = H(k) + H(k+1) \quad \text{com } H(m) = \sum_{n=0}^{N/2-1} h(n) C_N^{(2m+1)m}. \quad (22)$$

Verificando a equação original de $X(k)$ com N elementos (Eq. 6), nota-se que ela pode ser substituída pelas Equações (13) e (22). Isto equivale a substituir a DCT de N elementos por duas DCT's de $N/2$ elementos. Continuando este processo iterativo, consegue-se calcular toda a DCT com um número de operações reduzido da ordem de $N \log_2 N$.

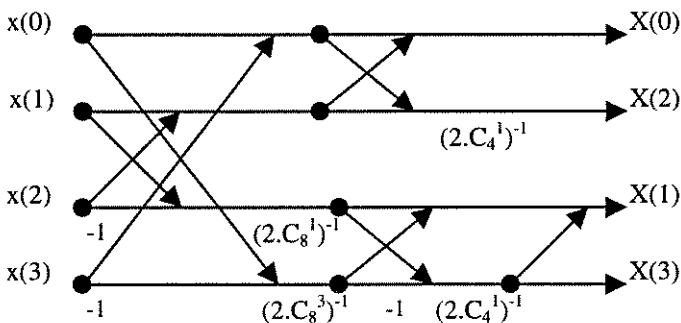
Vejamos alguns exemplos da aplicação do algoritmo:

$$N = 2 \Rightarrow \begin{cases} k = 0 \Rightarrow K = \text{par} \Rightarrow X(0) = g(0) = x(0) + x(2-0-1) = x(0) + x(1) \\ k = 1 \Rightarrow K = \text{impar} \Rightarrow X(1) = h(0) = \frac{[x(0) - x(2-0-1)]}{2 \cdot C_{21}^{(2-0+1)}} = \frac{x(0) - x(1)}{2 \cdot C_2^1} \end{cases} \quad (23)$$

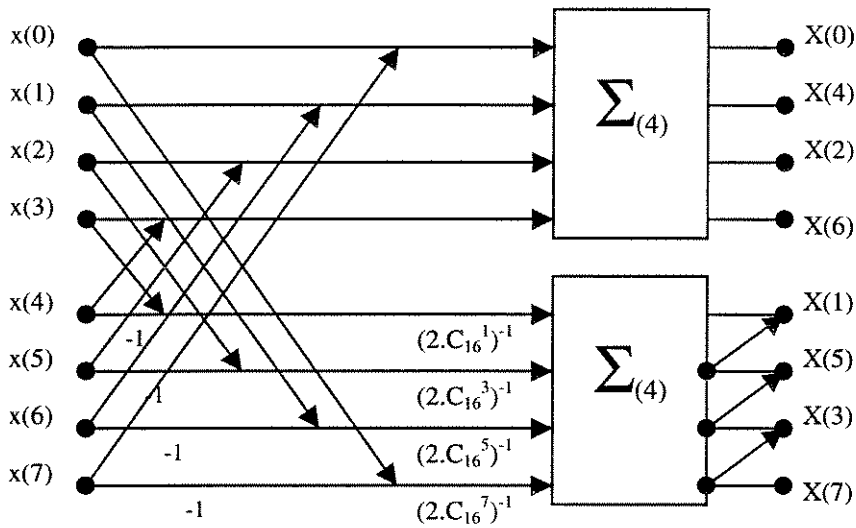
Podemos representarmos (23) pelo diagrama de fluxo abaixo



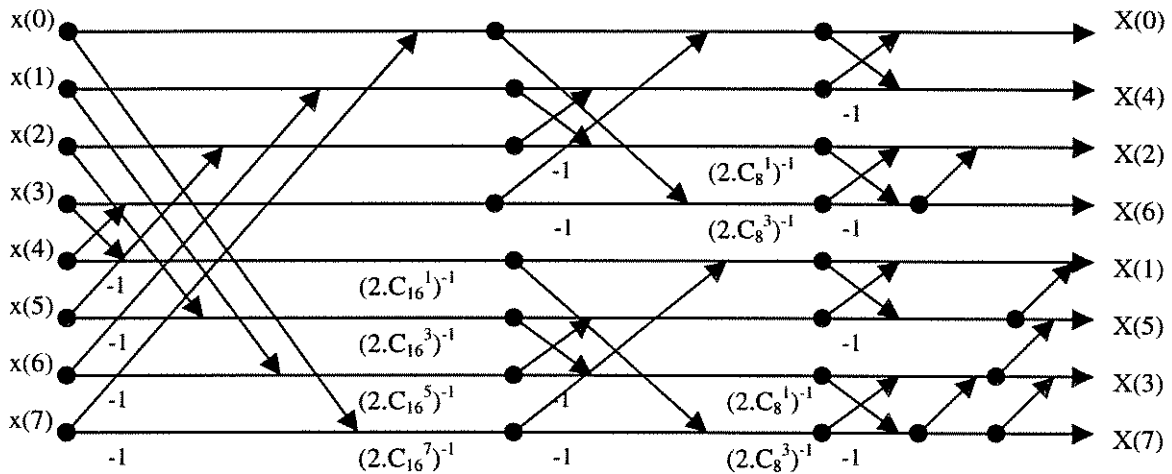
Para $N = 4$



Para $N=8$ (diagrama utilizado na implementação da DCT neste trabalho).



Equivalentemente temos



Nos diagramas acima, a DCT é obtida a partir de $X(k)$ pelo produto indicado na equação (5). Observamos que os valores de entrada se encontram na ordem natural, isto é, $x(0)$, $x(1)$, .. $x(n)$, e os valores de saída encontram-se na ordem *bit-reversa*.

O algoritmo apresentado calcula a DCT em uma dimensão. Como a DCT é uma transformada de núcleo separável, a DCT em duas dimensões é obtida efetuando-se a DCT unidimensional em cada uma das duas dimensões separadamente. No caso de imagens, isto corresponde a efetuar a DCT nas linhas e depois nas colunas (ou vice-versa) do bloco da imagem.

Podemos decompor de maneira similar a IDCT e obter o seu diagrama de fluxo de sinal. Em vez disto, utilizamos as propriedades das transformações ortogonais. Deste

modo, a IDTC é obtida simplesmente invertendo-se todas as setas do diagrama de fluxo de sinal usado no cálculo da DCT.

REFERÊNCIAS

- [1] M. Rabbani, P. W. Jones, "Digital Image Compression Techniques", SPIE Optical Engineering Press, 1991, Capítulos 3 e 4.
- [2] M. Rabbani, P. W. Jones, "Digital Image Compression Techniques", SPIE Optical Engineering Press, 1991, Capítulo 6.
- [3] K. Takahashi e M. Ohta, "Data compression coding of gray-scale images using bit planes", in Proc. ICC, 2.3.1, pp. 34-41, 1985.
- [4] T. V. Ramabadran and K. Chen, "Efficient compression of medical images though arithmetic coding", in Proc. SPIE Medical Imaging IV, 1234, 761-776, 1990.
- [5] García, C. Munoz, and A. Sanz, "Universal compression lossless code statistically built", in Proc. ICASSP, 48.5.1- 48.5.4, 1994.
- [6] M. Rabbani, P.W. Jones, "Digital Image Compression Techniques" SPIE Optical Engineering Press , 1991, Capítulo 11.
- [7]] E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding", IEEE Trans. Commun., COM-27(9), 1135-1142, 1979.
- [8] D. R. Halverson, "On the implementation of a block truncation coding algorithm", IEEE Trans. Commun., COM-30(11), 2482- 2484.
- [9] D. R. Halverson, N. C. Griswold, and G. L. Wise, "A generalized block truncation coding algorithm for image compression", IEEE Trans. Acous. Speech Signal Proc. , ASSP-32(3), 664-668, 1984.
- [10] N. C. Griswold, D. R. Halverson, and G. L. Wise, "A note on adaptive block truncation coding for image processing", IEEE Trans. Acous. Speech Signal Proc. , ASSP-35(8), 1201-1203(1987).
- [11] O. R. Mitchel and E. J. Delp, "Multilevel graphics representation using block truncation coding", Proc. IEEE, 68(7), 868-873.
- [12] J. D. J. Healy and O. R. Mitchel "Digital video bandwidth compression using block truncation coding", IEEE Trans. Commun. COM-29(12), 1809-181, 1981.
- [13] M. D. Lema and O. R. Mitchel, "Absolute moment block truncation coding and its application to color images", IEEE Trans. Commun, COM-32(10), 1148-1157, 1984.
- [14] M. Rabbani, P.W. Jones, "Digital Image Compression Techniques", SPIE Optical Engineering Press , 1991, Capítulo 12.
- [15] V. R. Udpikar and J. P. Raina, "BTC image coding using vector quantization", IEE Trans. Commun., COM-35(3), 352-356, 1987.
- [16] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design", IEEE Trans. Commun., COM-28(1) 84-95, 1980.

- [17] R. M. Gray, "Vector quantization", IEEE ASSP magazine, 1(2), 4-29, 1984.
- [18] W. H. Equitz, "A new vector quantization clustering algorithm", IEEE Trans. Acous. Speech Signal Proc., ASSP-37(10), 1568-1575, 1989.
- [19] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding", Proc. IEEE, 73(11), 1551, 1558, 1985.
- [20] R. L. Baker, "Vector quantization of digital images", Ph.D. thesis, Department of Electrical Engineering, Stanford University, 170-175, June 1984.
- [21] R. L. Baker and R. M. Gray, "Differential vector quantization of achromatic imagery", in Proc. Int. Picture Symposium, 105-106, 1983.
- [22] H. M. Hang and B. Haskell, "Interpolative vector quantization of images", IEEE Trans. Commun., COM-36(4), 465-470, 1988.
- [23] B. Ramamurthi and A. Gersho, "Classified vector quantization of images", IEEE Trans. Commun, COM-34(11), 1105-1115(1986).
- [24] J. Foster, R. M. Gray, and M. O. Dunham, "Finite-state vector quantization for waveform coding", IEEE Trans. Info, Theory, IT-31(3), 348-359, 1985.
- [25] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling", IEEE Trans. Info. Theory, 35(2), 299-315, 1989.
- [26] J. M. Rabbani, P.W. Jones, "Digital Image Compression Techniques" SPIE Optical Engineering Press , 1991, Capítulo 13.
- [27] P. P. Vaidyanathan, "Quadrature mirror filter, M-band extensions and perfect-reconstruction techniques", IEEE ASSP Magazine, 4(3), 4-20, 1987.
- [28] J. D. Johnston, "An filter family designed for use in quadrature mirror filter banks", in Proc. ICASSP, 291-294, 1980.
- [29] M. J. Smith and T. P. Barnwell, "Exact reconstruction techniques for a tree structure subband coders". IEEE Trans. Acous. Speech Signal Proc. , ASSP-34(3), 434-441, 1986.
- [30] D. Le Galland and A. Tabatabai, "Sub-band coding of digital images using symmetric kernel filters and arithmetic coding techniques", in Proc. ICASSP, 761-764, 1988.
- [31] J. W. Woods and S. D. O'Neil , Subband coding of images, "IEEE Trans. Acous. Speech Signal Processing. , ASSP-35(8), 1278-1288, 1986.
- [32] H. Gharavi and A. Tabatabai , "Sub-band coding of monochrome and color images", IEEE Trans. Circuits Systems, 35(2) 207-214, 1988.
- [33] P. H. Westernk, D. E. Boeckee, J. Biemond, and J. W. Woods, "Subband coding of images using vector quantization ", IEEE Trans. Commun., COM-36(6), 713-719, 1988.
- [34] D. M. Baylon and J. S. Lim, "Transform/subband analysis and synthesis of signal", MIT Research Technical Report, June 1990.
- [35] N. S. Jayant and Peter Noll, "Digital Coding of Waveforms - Principles and

- Applications to Speech and Video", Prentice Hall. INC, Capítulo 11.
- [36] M. Rabbani, P.W. Jones, "Digital Image Compression Techniques" SPIE Optical Engineering Press , 1991, Capítulo 10.
 - [37] E. Dubois and J. L. Moncet, "Encoding and progressive transmission of still pictures in NTSC composite format using transform domain methods", IEEE Trans. Commun., COM-34(3), 310-319, 1986.
 - [38] R. J. Clarke, "Transform Coding of Images", Academic Press, London, 1985
 - [39] A. Habibi and P.A. Wintz, "Image coding by linear transformation and block quantization", IEE Trans. Commun. Tech., COM-19(1), 50-63, 1971.
 - [40] W. Pennebaker, "JPEG Technical Specification Revision 8", Whorking Document No. JTC1/SC2/WG1/JPEG-8-R8, Aug. 1990.
 - [41] W. B. Pennebacker and J. L. Mitchell, "JPEG Still Image Data Compression Standard", Van Nostrand Reinhold – New York.
 - [42] B. G. Lee, M. Kang and J. Lee, "Broadband Telecommunications Technology", Artech House, Inc, 1993, Capítulo 6
 - [43] C. A. Gonzales and J. L. Mitchell, "Note on DCT algorithms with low multiplicative complexity", JPEG-150, 1988.
 - [44] J. Makhoul, "A Fast Cosine Transform in One and Two Dimensions", IEEE Trans. Acous., Speech, Signal Proc., ASSP-28(1), 27-34.
 - [45] Z. Wang , "On Computing the Discrete Fourier and Cosine Transform ", IEEE Trans. Acous. Speech Signal Proc., ASSP-33(4), 1341-1344(1985.
 - [46] H. S. Hou, "A Fast Recursive Algorithm for Computing the Discrete Cosine Transform", IEEE Trans. Acous. Speech Signal Proc., ASSP-35(10), 1455-1461, 1987.
 - [47] B. G. Lee " A New Algorithm to Computer the Discrete Cosine Transform ", IEEE Trans. Acous. Speech Signal Proc., ASSP-32(6), 1243-1245, 1984.
 - [48] A. V. Oppenheim and R. W. Shafer, "Digital Signal Processing", Englewood Cliffs, NJ : Prentice Hall, 1975
 - [49] E. O. Brigham, "The Fast Fourier Transmform", Englewood Cliffs, NJ: Prentice Hall, 1974, pp. 166-169.
 - [50] D. A. Huffman, "A method for the construction of minimum redundancy codes", Proc. IRE, 40, 1098-1101, 1952.
 - [51] M. Rabbani, P. W. Jones, "Digital Image Compression Techniques" SPIE Optical Engineering Press , 1991, Capítulo 1, pp. 24-28.
 - [52] Didier Le Gall, " MPEG : A Video Compression Standard for Multimedia Applications", Communications of the ACM, April 1991, Vol. 34, No.4.
 - [53] "Coding of Moving Pictures and Associated Audio", Committee Draft of Standard ISO11172 : ISO/MPEG/90/176, Dec. 1990.
 - [54] ISO-IEC/JTC1/SC29/WG11 "Coded Presentation of Picture and Audio Information", 1992.

-
- [55] Andreson. M., "VCR quality video at 1,5Mb/s", National Communication Forum (Chicago, Oct. 1990).
 - [56] Chen, C. T. and Le Gall D. J., "A Kth Order Adaptive Transform Coding Algorithm for High-Fidelity Reconstruction of Still Images", In proceedings of the SPIE (San Diego, Aug. 1989) .
 - [57] P. Prancha and M. El Zarki, "MPEG coding for Variable Bit Rate Video Transmission", IEEE Communication Magazine, May. 1994.
 - [58] F. Dufaux and F. Moscheni "Motion Estimation Techniques for Digital TV : A Review and a New Contribution", Proceedings of the IEEE, Vol. 83, No. 6, June 1995.
 - [59] B. K. P. Horn and B. G. Shunck , "Determining Optical Flow", Artif. Intell., vol. 17, pp. 185-203, 1981.
 - [60] B. Lucas and T. Kanade, "An Iterative Image Registration Techniques with an Application to Stereo Vision", in Proc. Image Understanding Workshop, pp. 121-130, 1981.
 - [61] H. H. Nagel, "Displacement Vectors Derived from Second-Order Intensity Variation in Image Sequences", Computer Graphics and Image Process., vol. 21, pp. 85-117, 1983.
 - [62] "Constraints for the Estimation of Displacement Vector Fields from Image Sequences", in Proc. Int. Joint Conf. on Artif. Intell., Karlsruhe, Germany, Aug, 1983, pp. 945-951.
 - [63] A. Netravali and J. D. Robbins, "Motion Compensated Television Coding, Part I", Bell Syst. Tech. J., vol 58, No. 3, pp. 629-668, Apr. 1979.
 - [64] C. Cafforio and F. Rocca "The Differential Method for Motion Estimation", in Image Sequence Processing and Dynamic Scene Analysis, T. S. Hung, Ed. New York : Springer-Verlag, pp. 104-124.
 - [65] D. R. Walker and K. R. Rao, "Improved Pel-Recursive Motion Compensation", IEEE Trans. Commun. vol. COM-32, pp. 1280-1134, Oct., 1984.
 - [66] J. R. Jain and A. K. Jain, "Displacement and Measurement and its Application in Interframe Image Coding", IEEE Trans. Commun. Vol. COM-29, pp. 1799-1808, Dec. 1981.
 - [67] T. Koga et al. "Motion compensated Interframe coding of video conferencing", in Proc. Nat. Telecommun. Conf. New Orleans, LA, Dec. 1981, pp. G5.3.1-G5.3.5.
 - [68] H. G. Musmann, P. Pirsh, and H. J. Grallert, "Advances in Picture Coding", Proc. IEEE, vol. PROC-73, pp. 549-575, Apr. 1985.
 - [69] B. G. Haskell, "Frame-to- Frame Coding of Television Pictures Using Two-Dimensional Fourier Transforms", IEEE Trans. Inform. Theory, vol. IT-20, pp. 119-120, Jan. 1974.
 - [70] D. J. Heeger, "Optical flow using spatio-temporal filters", Int. J. Computer

- Vision, vol. 1, No 4, pp. 279-302, Jan. 1988.
- [71] D. J. Fleet and A. D. Jepson, "Computation of component image velocity from local phase information", *Int. J. Computer Vision*, vol. 5 pp. 77-104, 1990.
 - [72] B. K. P. Horn, "Robot Vision", Cambridge, MA : MIT Press, 1986.
 - [73] R. Srinivasan and K. R. Rao, "Predictive Coding Based on Efficient Motion Estimation", *IEEE Trans. Commun.* vol. COM-33, pp. 888-896, Aug. 1985.
 - [74] B. Liu and A. Xscarin, "New Fast Algorithms for the Estimation of Block Motion vectors", *IEEE Trans. Circ. and Syst. for Video Tech.*, vol. CSVt-3, No. 2, pp. 148-157, Apr. 1993.
 - [75] A. de O. Alonso, J. B. T. Yabu-Uti, N. Alens, Y. Yano "Digitalização de Sinais de TV" Relatório Técnico RT-80, Publicação Faculdade de Engenharia Elétrica da Unicamp FEE 44/80, Dezembro 1980., pp. 24-38.
 - [76] M. Rabbani, P. W. Jones, "Digital Image Compression Techniques" SPIE Optical Engineering Press , 1991, pp. 217- 220.
 - [77] N. M. Nasrabadi and R. A. King, "Image Coding Using Vector Quantization : a Review", *IEE Trans. Commun.*, 36(8, 975-971, 1988.
 - [78] L. M. Hurvich, "Color Vision", Sinauer Associates, Sunderland, MA, 1981.
 - [79] R. W. G. hunt, "The Reproduction of Colour", Fountain Press, England, Fouth Edition, 1987.
 - [80] W. K. Pratt, "Digital Image Processing", 2ª Edition, New York, John Wiley, 1991.
 - [81] B. G. Lee, M. Kang and J. Lee, "Broadband Telecommunications Technology", Artech House, Inc, 1993, Capítulo 1.
 - [82] CCITT Rec. 1.120, "Integrated Services Digital Networks (ISDN", 1992 (Rev).
 - [83] CCITT Rec. 1.121, "Broadband Aspects of ISDN", 1990.
 - [84] [B. G. Lee, M. Kang and J. Lee, "Broadband Telecommunications Technology", Artech House, Inc, 1993, Capítulo 4.
 - [85] CCITT Rec. 1.211. "BISDN Services Aspects", 1992 (Rev).
 - [86] CCITT Rec. 1.311, "BISDN General Network Aspects", 1992 (Rev).
 - [87] CCITT Draft Rec. 1.374, "Network Capacity for the Support of Multimedia Services", 1992
 - [88] CCITT Rec. 1.413, "BISDN User-Network Interface", 1992 (Rev).
 - [89] CCITT Rec. 1.327, "BISDN Functional Architecture Aspects", 1992 (Rev).
 - [90] [CCITT Rec. 1.321, "BISDN Protocol Reference Model and Its Application", 1990
 - [91] CCITT Rec. 1.364, "Support of Broadband Connectionless Data Service on BISDN", 1992.
 - [92] CCITT Rec. 1.150, "BISDN ATM Functional Characteristics", 1992 (Rev).
 - [93] CCITT Rec. 1.361, "BISDN ATM Layer Specification", 1992 (Rev).
 - [94] CCITT Rec. 1.363, "BISDN ATM Adaptation Layer (AAL Specification",

- 1992 (Rev).
- [95] CCITT Rec. 1.3632, "BISDN ATM Adaptation Layer (AAL Functional Description", 1992 (Rev).
 - [96] ATM Forum, "ATM User-Network Interface Specification", Version 2.0, 1992
 - [97] CCITT Rec. 1.371, "Traffic Control and Congestion Control in BISDN", 1992.
 - [98] M. Khademi and F. J. Paoloni "Performance and Statistical Analysis of Two-Layer MPEG Coding Algorithm for ATM Networks", IEEE Proc. Commun., vol. 143, No 2, June 1996, pp. 155-161.
 - [99] B. Declene, P. Pancha, M. El Zarki and H. Sorensen, " Comparison of Priority Partition Methods for VBR MPEG', Proceedings of IEEE INFOCOM 94, Canada, 1994, pp. 689-696.
 - [100] C. Guillemot , "Layered Coding Schemes for video Transmission on ATM networks", J. Vis. Commun. Image Represent., 1994, 5, pp. 62-74.
 - [101] International Organization for Standardization ISO MPEG, "Coding of Moving Picture and associated audio for storage media at up about 1,5Mb/s", 1994.
 - [102] M. Ghanbri, "Two-Layer Coding of Video Signal for VBR Networks", IEEE, J. 1989, SAC-7(5).
 - [103] M. Ghanbri, "An Adapted H.261 Two-Layer Video Codec for ATM networks", IEEE, J., 1992, COM-40, pp. 1481-1490.
 - [104] M. Ghanbari and V. Seferidis, "Cell-loss concealment in ATM video coders", IEEE J. 1993, CSVT-3, pp. 238-247.
 - [105] J. Kin and g. Park, "Two-Layered DCT Based Coding Scheme for Recording digital HDTV signals", Proceedings of IEEE ICASSP 94, Australia, 1994, pp. V425-428.
 - [106] R. Siracusa, K. Joseph, J. Zdepski and D. Raychaudhuri, " Flexive and Robust Packet Transport for Digital HDTV", IEEE J. , 1993, SAC-11(1).
 - [107] P. Pancha and M. El. Zarki, "MPEG Coding for Variable Bit Rate Transmission", IEEE Commun. Mag. 1994, pp. 54-66.
 - [108]] M. Ghanbari and J. Azari, "Effect of Bit Rate Variation of the Base Layer on the Performance of Two-Layer Video Codecs", IEEE. J. 1994, CSVT-4 (1, pp. 8-16.
 - [109] A. E. F. e Silva "Avaliação de Sistemas de Redução de Taxa de Bits de Vídeo com Enfoque na Percepção do Usuário", Teste de mestrado, Faculdade de Engenharia Elétrica e de Computação – Universidade Estadual de Campinas, Campinas, SP, fevereiro 1998.
 - [110] N. S. Jayant and Peter Noll, "Digital Coding of Waveforms - Principles and Applications to Speech and Video", Prentice Hall. INC, Capítulo 11.
 - [111] C. J. Hughes, M. Ghanbari, D. E. Pearson, V. Seferidis, and J. Xiong, "Modeling and Subjective Assessment of Cell Discard in ATM Video", IEEE Trans. Image Process., vol. 2 No 2, pp. 212-222, Apr 1993.

- [112] M. Ghanbari and V. Seferidis, "Efficient H.261- Based Two-Layer Video Coders for ATM Networks", IEEE Trans. On Circ. and Syst. Video Tech., vol. 5, No2, pp. 171-174, Apr 1995.