

EDSON TADEU MONTEIRO MANOEL

**CODIFICAÇÃO DE VÍDEO H.264 – ESTUDO DE
CODIFICAÇÃO MISTA DE MACROBLOCOS**

**FLORIANÓPOLIS
2007**

UNIVERSIDADE FEDERAL DE SANTA CATARINA

**PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA ELÉTRICA**

**CODIFICAÇÃO DE VÍDEO H.264 – ESTUDO DE
CODIFICAÇÃO MISTA DE MACROBLOCOS**

Dissertação submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a
obtenção do grau de Mestre em Engenharia Elétrica

EDSON TADEU MONTEIRO MANOEL

FLORIANÓPOLIS, MARÇO DE 2007.

CODIFICAÇÃO DE VÍDEO H.264 – ESTUDO DE CODIFICAÇÃO MISTA DE MACROBLOCOS

Edson Tadeu Monteiro Manoel

‘Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Engenharia Elétrica, Área de Concentração em Processamento Digital de Sinais, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.’

Prof. Rui Seara, Dr.
Orientador

Prof^a. Kátia Campos de Almeida, Dr^a.
Coordenadora do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

Prof. Rui Seara, Dr.
Presidente

Prof. Sidnei Noceti Filho, Dr.

Prof. Orlando J. Tobias, Dr.

AGRADECIMENTOS

Ao meu orientador, Prof. Rui Seara, e ao Prof. Marcos Moecke, pela atenção e orientação despendida, pelo conhecimento transmitido e por todas as reuniões, revisões e demais colaborações essenciais ao desenvolvimento desse trabalho.

À minha esposa Maria Luisa, pela total compreensão, apoio, confiança e inspiração, e por todos os momentos de felicidade (incluindo nosso casamento e a gravidez de nosso primeiro filho).

À minha família e especialmente a meus pais, que sempre me motivaram e ajudaram de todas as formas em minha educação, e que me incentivaram nos momentos mais difíceis.

Aos membros da banca, pelo interesse e pelas contribuições com correções e sugestões.

Ao Elton Fontão e a todos os demais colegas do Linse, pelo companheirismo e por todas as vezes que me auxiliaram.

A todos meus professores do Programa de Pós-Graduação em Engenharia Elétrica, por proporcionarem minha formação, fornecendo valiosos conhecimentos.

Aos colegas das empresas em que estive trabalhando durante o desenvolvimento deste trabalho, pela compreensão e pela flexibilização de horário que foi necessária.

A todas as pessoas que, direta ou indiretamente, me ajudaram a seguir esse caminho e a construir todo o conhecimento necessário para a realização desse trabalho, seja através de ensinamentos ou de inspiração.

Resumo da Dissertação apresentada à UFSC como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica

CODIFICAÇÃO DE VÍDEO H.264 – ESTUDO DE CODIFICAÇÃO MISTA DE MACROBLOCOS

Edson Tadeu Monteiro Manoel

Março/2007

Orientador: Prof. Rui Seara, Dr.

Área de Concentração: Comunicações e Processamento de Sinais

Palavras-chave: Codificação e compressão de vídeo, padrão H.264, televisão digital.

Número de Páginas: 97

RESUMO: Esta dissertação trata da codificação e compressão de vídeo digital, particularmente focando o recente padrão de codificação H.264. O interesse em tal padrão tem crescido bastante ultimamente, principalmente para o uso em novos sistemas de armazenamento e de transmissão de vídeo digital. O H.264 é um padrão de vídeo atual que possui muito bom desempenho – taxa de bits 50% menor do que a do seu antecessor MPEG-2, mantendo o mesmo desempenho de qualidade. Nesta dissertação, são abordadas duas extensões (aprimoramentos) ao padrão H.264, visando melhorar ainda mais o desempenho de tal padrão, isto é, aumentar a qualidade do sinal comprimido (mantendo a mesma taxa de bits), ou reduzir a taxa de bits (mantendo a mesma qualidade). As extensões fundamentam-se no fato de que alguns tipos de macroblocos (segmentos do sinal de vídeo) possuem uma pequena região distinta que geralmente tem influência negativa na taxa de bits. Inicialmente, são apresentadas as principais características da codificação de vídeo e do padrão H.264. Em seguida, um detalhamento dos diversos processos que estão relacionados aos aprimoramentos são considerados, principalmente nos processos de codificação de macroblocos, predição e otimização da taxa-distorção Lagrangiana. Para avaliar os novos modos de codificação, o código do modelo de referência (JM) de implementação do padrão H.264 é modificado para incluir tais extensões (tanto na codificação quanto na decodificação). Os resultados obtidos decorrentes das modificações propostas são avaliados através do uso de diversas seqüências-padrão de teste. Tais resultados indicam que os aprimoramentos conseguidos são passíveis de serem incluídos no referido padrão.

Abstract of Dissertation presented to UFSC as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

H.264 VIDEO CODING – STUDY OF MIXED MACROBLOCK CODING

Edson Tadeu Monteiro Manoel

March/2007

Advisor: Prof. Rui Seara, Dr.

Area of Concentration: Communications and Signal Processing

Keywords: Digital television, H.264 standard, video coding and compression.

Number of Pages: 97

ABSTRACT: This work discusses the problem of video coding and digital video compression, focusing particularly on the recent video coding H.264 standard. The interest in such a standard has been substantially growing, mainly for its use in new systems of digital video storage and video transmission. H.264 is a current video standard exhibiting a very good performance – a compression rate twice better than its predecessor MPEG-2, while maintaining the same quality performance. In this dissertation, two extensions (improvements) to the H.264 standard are suggested and assessed, aiming to improve its compression performance even further, i.e., either to increase the video quality maintaining the same bit rate, or to reduce the bit rate maintaining the same quality. These extensions are based on the fact that some macroblock types (segments of the video signal) have a small region with distinct features which may have a negative effect on the bit rate. Initially, the main video coding characteristics as well as the H.264 standard features are discussed. After that, a more detailed analysis about several related processes to the H.264 standard aiming their improvements are considered, mainly regarding macroblock coding, prediction, and Lagrangian rate-distortion optimization. The H.264 test model code (JM) has been modified (including changes in both coding and decoding modules) in order to assess the new coding modes. Several standard test sequences are used and the obtained results with the modified video standard are evaluated. Such results point out that the proposed improvements are feasible of being included in the referred video standard.

Sumário

1. Introdução	1
1.1. Breve Histórico	2
1.2. H.264.....	4
1.3. Organização da Dissertação	6
2. Introdução à Codificação de Vídeo.....	8
2.1. Introdução	8
2.2. Sinal de Vídeo Digital.....	8
2.3. Compressão Com Perdas e Sem Perdas.....	9
2.4. Espaços de Cores	10
2.5. Subamostragem Croma	12
2.6. Particionamento	13
2.7. Compensação de Movimento.....	15
2.8. Transformação	16
2.9. Quantização.....	18
2.10. Reordenação e Codificação de Entropia	19
3. Características Principais do Padrão H.264	21
3.1. Introdução	21
3.2. Transformação	21
3.2.1. Introdução.....	21
3.2.2. Transformada DCT Inteira com Inversa “Exata”	21
3.2.3. Transformada de Tamanho 4×4.....	22
3.2.4. Transformadas “Hierárquicas”	22
3.2.5. Utilização de Aritmética de 16-bits.....	23
3.3. Predição	23
3.3.1. Predição Intra 4×4	23
3.3.2. Precisão de Um Quarto de Amostra na Compensação de Movimento.....	24
3.3.3. Tamanhos de Blocos Variáveis na Compensação de Movimento.....	25
3.3.4. Utilização de Múltiplas Referências Temporais.....	26
3.3.5. Predição de Movimento.....	26
3.3.6. VMs que Apontam para Fora do Quadro	26

3.3.7.	Filtro Antiblocagem Adaptativo.....	26
3.3.8.	Predição com Peso.....	27
3.4.	Codificação	27
3.4.1.	Organização dos Dados	28
3.4.2.	Grupos de <i>Fatias</i>	29
3.4.3.	Codificação de Entropia Adaptativa ao Contexto	30
3.4.4.	Codificação de Entropia Aritmética	31
4.	Codificação dos Macroblocos no H.264.....	33
4.1.	Introdução	33
4.2.	Processos.....	34
4.3.	Codificação de Entropia nos Macroblocos	37
4.3.1.	Codificação Exp-Golomb sem sinal	37
4.3.2.	Codificação Exp-Golomb com sinal.....	39
4.4.	Predição	40
4.4.1.	Predição <i>Inter</i>	40
4.4.2.	Predição de Vetor de Movimento	42
4.4.3.	Predição <i>Intra</i>	45
4.5.	Codificação do Erro Residual	47
4.6.	Otimização Taxa-Distorção	48
4.7.	Estimação de Movimento	51
5.	1ª Proposta de Modificação: Codificação com MB Duplo-Inter	53
5.1.	Introdução	53
5.2.	Codificação	55
5.2.1.	Mudanças nos Elementos de Sintaxe do MB	56
5.2.2.	Estudo de Caso	58
5.3.	Alterações nos Processos de Estimação de Movimento e Otimização Taxa-Distorção.....	59
5.4.	Alteração no Processo de Predição dos VMs	60
5.5.	Testes de Codificação	62
5.5.1.	Condições de Teste.....	62
5.5.2.	Frequência de Utilização do Modo de MB Proposto	63
5.5.3.	Ganho em Taxa-Distorção.....	73

5.6.	Conclusões	80
6.	2ª Proposta de Modificação: Codificação com MB “Híbrido” Inter/Intra	81
6.1.	Introdução	81
6.2.	Alteração dos Processos do Codificador	82
6.3.	Alterações no Fluxo de Bits.....	83
6.3.1.	Codificação Original do H.264.....	84
6.3.2.	Alteração Proposta na Codificação.....	85
6.4.	Resultados.....	86
6.4.1.	Ganho em Taxa-Distorção.....	87
6.5.	Conclusões	89
7.	Conclusão	90
7.1.	Discussão	90
7.2.	Propostas para Trabalhos Futuros.....	90
A.	Sequências de Teste	92

LISTA DE FIGURAS

Fig. 1.1 – Cronologia do desenvolvimento dos diversos padrões de compressão de vídeo, apresentados isoladamente pelos órgãos ITU-T e MPEG, ou em um esforço conjunto de ambos [5].....	3
Fig. 1.2 – Curva taxa-distorção para a seqüência “Foreman”, no formato QCIF, a 10 quadros/s, sendo codificada com vários dos protótipos desenvolvidos durante o processo de padronização do H.264, mostrando a evolução dos mesmos, em comparação com o H.263+ e o MPEG-4 ASP [10].....	5
Fig. 2.1 – Ilustração de amostragem temporal de um vídeo e da amostragem espacial de um quadro.	9
Fig. 2.2 – Diferença entre um sinal linear e um com <i>correção gama</i>	11
Fig. 2.3 – Bases (eixos) dos espaços de cores. (a) Base do R’G’B’. (b) Base do Y’CbCr.	12
Fig. 2.4 - Posição relativa das amostras <i>luma</i> , representadas pelos círculos cinzas, e das amostras <i>croma</i> , representadas pelos círculos azul/amarelo (<i>Cb</i>) e vermelho/verde (<i>Cr</i>), em diferentes formatos de subamostragem. (a) Formato 4:4:4. (b) Formato 4:2:2. (c) Formato 4:2:0 do MPEG-1. (d) Formato 4:2:0 do MPEG-2 e do H.264.	13
Fig. 2.5 – Divisão de um quadro em macroblocos. (a) Divisão do quadro em 99 macroblocos (grade de 11 MBs na horizontal × 9 MBs na vertical). (b) Blocos de amostras <i>Y’</i> , <i>Cb</i> e <i>Cr</i> de um MB.....	14
Fig. 2.6 – Exemplo do resultado de um processo de CM. Neste caso o VM do bloco 4×4 ilustrado é (-3, -2).	15
Fig. 2.7 – Varredura em <i>zig-zag</i>	19
Fig. 3.1 – Ilustração da utilização da transformada Hadamard no padrão H.264.	22
Fig. 3.2 – Sentido dos 8 modos de predição <i>intra</i> direcionais utilizados no H.264. O modo 2, que não está mostrado na figura, é o modo de predição DC, que utiliza a média dos vizinhos.	24

Fig. 3.3 – Exemplo de compensação de movimento utilizando um vetor não inteiro (neste caso: 2 amostras e $\frac{3}{4}$ à direita e $\frac{1}{2}$ amostra acima).	25
Fig. 3.4 – Exemplos de particionamentos possíveis utilizados para a compensação de movimento. (a) Tipo 0 (16×16). (b) Tipo 1 (16×8). (c) Tipo 2 (8×16). (d) Tipo 3 (8×8). (e) Variações do tipo das partições dos SubMBs, quando o tipo é 3.....	25
Fig. 3.5 – Organização hierárquica da camada de codificação de vídeo, subdividindo o vídeo em quadros, quadros em fatias e fatias em macroblocos.	29
Fig. 3.6 – Exemplos dos seis tipos de mapeamento de grupos de fatias pré-definidos no H.264, cada grupo tem uma tonalidade diferente. (a) <i>Interleaved</i> com três grupos. (b) <i>Dispersed</i> com quatro grupos. (c) <i>Foreground and Background</i> com quatro grupos. (d) <i>Box-out</i> com 18 MB no grupo 0. (e) <i>Raster</i> com 29 MB no grupo 0. (f) <i>Wipe</i> com 29 MB no grupo 0.	30
Fig. 4.1 – Esquema simplificado do encadeamento dos processos no decodificador do padrão H.264.....	34
Fig. 4.2 - Esquema do encadeamento dos processos no codificador do padrão H.264.....	36
Fig. 4.3 – Tamanho de código $ue(v)$ e a distribuição de probabilidade ideal. (a) Tamanho de código. (b) Distribuição de probabilidade ideal.....	39
Fig. 4.4 – Tamanho de código $se(v)$ e a distribuição de probabilidade ideal. (a) Tamanho de código. (b) Distribuição de probabilidade ideal.....	40
Fig. 4.5 – Características (resposta em freqüência e resposta ao degrau) do filtro utilizado para a interpolação das meias-amostras. Na resposta em freqüência, o filtro é comparado com o filtro de interpolação ($\frac{1}{2}$, $\frac{1}{2}$).....	41
Fig. 4.6 – Ilustração das amostras utilizadas nas interpolações das amostras em posições não-inteiras. (a) (Quadrados cinza) interpolação em meias-amostras utilizando filtro FIR de tamanho 6. (b) Interpolação em um quarto de amostra com um filtro FIR ($\frac{1}{2}$, $\frac{1}{2}$) para as amostras indicadas em negrito (a, c, d, e, f, g, i, k, n, p, q e r).....	42
Fig. 4.7 – Exemplos de resultados do processo de localização da vizinhança de uma subpartição para a predição de VM. (a) Com todos MBs sem partição (16×16).	

(b) Com MBs vizinhos particionados e subparticionados. (c) Subpartições vizinhas de uma subpartição no mesmo MB.	44
Fig. 4.8 – Exemplos da predição do vetor de movimento para outros particionamentos do MB. (a) Particionamento em 16×8. (b) Particionamento em 8×16.	45
Fig. 4.9 – Oito dos nove tipos de predição Intra 4×4; as descrições estão na Tabela 4.3. Há também o modo (2), que é a predição DC, uma média de todas as amostras vizinhas.	46
Fig. 4.10 – Exemplo de uma curva taxa-distorção com pontos em diferentes estratégias de otimização: (a) Minimização da taxa de bits. (b) Minimização da distorção, sujeita a uma taxa máxima. (c) Minimização da distorção.	48
Fig. 5.1 – Exemplo da codificação “desacoplada” de bloco 4×4 com erro residual maior que o do restante do MB.	53
Fig. 5.2 – Modos de particionamento do MB usados no padrão H.264 comparados com o modo proposto. (a) Modo <i>inter</i> utilizando apenas um VM, do padrão H.264. (b) Modo <i>inter</i> utilizando partições do padrão H.264. (c) Modo <i>duplo-inter</i> proposto.	55
Fig. 5.3 – Particionamento do MB <i>duplo-inter</i>	56
Fig. 5.4 – Códigos e valores para o parâmetro <code>mix_block</code> (posição da partição menor).	57
Fig. 5.5 – Influência do VM da partição maior na predição do VM da partição 4×4. (a) Vizinhos A, B, C e D de um bloco 4×4. (b) Quando a posição da partição 4×4 for uma das posições indicadas em cinza, o VM da partição maior é a predição do VM da partição 4×4.	60
Fig. 5.6 – Predição alternativa dos VMs para a partição menor do MB <i>duplo-inter</i> . Dependendo da posição, a predição do VM do bloco é dada pelo VM de um bloco vizinho diferente, ou pela mediana entre 3 blocos vizinhos. O “F” corresponde ao VM da partição maior do MB <i>duplo-inter</i> atual.	61
Fig. 5.7 – Coluna da esquerda: histograma dos tipos de MBs selecionados, para cada sequência (sem modo <i>duplo-inter</i> em azul e com modo <i>duplo-inter</i> em verde). Coluna da direita: histograma da posição selecionada (<code>mix_block</code>).	64

Fig. 5.8 – Coluna da esquerda: histograma dos tipos de MBs selecionados, para cada seqüência (sem modo <i>duplo-inter</i> em azul e com modo <i>duplo-inter</i> em verde). Coluna da direita: histograma da posição selecionada (<i>mix_block</i>). O QP varia entre 18 e 40.....	70
Fig. 5.9 – Curvas taxa-distorção da codificação no Cenário 0 (padrão H.264) e da codificação no Cenário 1 (com modo <i>duplo-inter</i> , coluna da esquerda). Diferença entre a distorção obtida no Cenário 1 e a obtida no Cenário 0 (coluna da direita). Os “marcadores” do eixo superior indicam o QP, que varia de 40 (mais à esquerda) a 18 (mais à direita).	73
Fig. 5.10 – Curva taxa-distorção da codificação normal e da codificação com modo <i>duplo-inter</i> (coluna da esquerda). Diferença entre a distorção obtida no modo <i>duplo-inter</i> e a obtida no modo normal (coluna da direita - quanto mais, melhor). Os “marcadores” do eixo superior indicam o QP, que varia de 40 (mais à esquerda) e 18 (mais à direita).....	78
Fig. 6.1 – Exemplo de um MB que pode se beneficiar da codificação com MB <i>híbrido</i> , pois possui uma região que pode ser predita através da predição <i>inter</i> (círculo) e outra região que se beneficiaria mais da predição <i>intra</i>	81
Fig. 6.2 – Exemplos de MBs <i>híbridos inter/intra</i> possíveis.....	82
Fig. 6.3 – Esquematisação dos processos do codificador com a inserção do modo <i>híbrido</i> . As setas tracejadas indicam que o processo de codificação do MB <i>híbrido</i> recebe as mesmas entradas que o processo de CM (quadros anteriores e VMs) e de codificação <i>intra</i> (predição <i>intra</i> e amostras reconstituídas).....	83
Fig. 6.4 - MB com 4 partições 8×8 e com 4 particionamentos diferentes para cada subMB: 8x8 para o subMB 0 (ao topo e à esquerda), 8x4 para o subMB 1 (ao topo e à direita), 4x8 para o subMB 2 (abaixo e à esquerda) e 4x4 (abaixo e à direita).....	84
Fig. 6.5 – Comparação das curvas taxa-distorção obtidas para a codificação padrão e a codificação com o modo <i>híbrido inter/intra</i>	87
Fig. A.1 – Primeiro quadro de cada uma das seqüências de vídeo utilizadas para os testes. (a) <i>Hall QCIF</i> . (b) <i>Stefan SIF</i> . (c) <i>Tennis SIF</i> . (d) <i>Foreman CIF</i> . (e) <i>Coastguard CIF</i>	92

LISTA DE TABELAS

Tabela 3.1 – Tipos pré-definidos de particionamento do quadro em grupos de fatias.....	30
Tabela 4.1 – Tipos de MB permitidos em cada tipo de fatia.....	34
Tabela 4.2 – Sequências Exp-Golomb $ue(v)$	38
Tabela 4.3 – Modos de Predição Intra.....	46
Tabela 5.1 – Valores e códigos para os tipos de MB em fatias P.....	57
Tabela 5.2 – Comparação da codificação de um MB pelo padrão H.264 com modo <i>duplo-inter</i> proposto	58
Tabela 5.3 – Descrição da numeração de modos utilizada nos gráficos de histograma.....	63
Tabela 5.4 – Frequência com que o modo <i>duplo-inter</i> foi escolhido para diferentes QPs - Cenário 1	68
Tabela 5.5 – Frequência com que os diferentes modos foram escolhidos, para QP = 27 - Cenário 1.....	69
Tabela 5.6 – Frequência com que os MBs foram escolhidos como tipo <i>duplo-inter</i> , para diferentes QPs, - Cenário 2.....	72
Tabela 5.7 – Frequência com que os MBs foram escolhidos nos diferentes modos, para QP = 27, no Cenário 2	72
Tabela 5.8 – Ganhos médio e máximo de PSNR no Cenário 1 em relação à codificação padrão	77
Tabela 5.9 – Ganhos médio e máximo de PSNR no Cenário 2 em relação à codificação padrão	80
Tabela A.1 – Descrição das sequências utilizadas nos testes	93

ACRÔNIMOS

CABAC	Codificação Aritmética Binária Adaptativa ao Contexto
CAVLC	Codificação Adaptativa de Comprimento de Palavra Variável
CBP	Padrão de Blocos Codificados (<i>Coded Block Pattern</i>)
CM	Compensação de Movimento
EM	Estimação de Movimento
EOB	Fim de Bloco (<i>End of Block</i>)
FRExt	<i>Fidelity Range Extensions</i>
HD	Alta Definição (<i>High-Definition</i>)
ISO	<i>International Standard Organization</i>
ITU	<i>International Telecommunication Union</i>
MAD	Média das Diferenças Absolutas (<i>Mean Absolute Difference</i>)
MB	Macrobloco
mb_type	Tipo de Macrobloco
MSE	Erro Médio Quadrático (<i>Mean Squared Error</i>)
NAL	Camada de Abstração de Rede (<i>Network Abstraction Layer</i>)
OTD	Otimização Taxa-Distorção
PSNR	Razão entre Pico de Sinal e Ruído (<i>Peak Signal to Noise Ratio</i>)
QP	Parâmetro de Quantização (<i>Quantization Parameter</i>)
SAD	Somas das Diferenças Absolutas (<i>Sum of Absolute Differences</i>)
SATD	Soma das Diferenças Transformadas Absolutas
SSD	Soma das Diferenças Quadráticas (<i>Sum of Squared Differences</i>)
sub_mb_type	Tipo do Submacrobloco
SubMB	Submacrobloco
TD	Taxa-Distorção
VCL	Camada de Codificação de Vídeo (<i>Video Coding Layer</i>)
VM	Vetor de Movimento

Bloco $M \times N$

Uma matriz de amostras (ou coeficientes) de M colunas por N linhas.

Compensação de Movimento (CM)

Principal processo de redução da redundância temporal. Consiste da subtração das amostras atuais por amostras de algum quadro previamente decodificado, deslocadas por um *vetor de movimento* (VM).

Croma (ou Crominância)

Corresponde às amostras dos dois sinais relacionados à colorização (U e V). Geralmente possuem uma frequência de amostragem mais baixa em relação às amostras *luma*.

Custo Lagrangiano

Corresponde à métrica de taxa-distorção $J = D + \lambda T$, onde D é uma medida de *distorção* (por exemplo, MAD) e T é a *taxa de bits*.

Distorção

Medida de qualidade da codificação, geralmente dada através do *PSNR* ou do *MSE*.

Elemento de Sintaxe

Cada parâmetro de codificação que é representado no *fluxo de bits*.

Erro Quadrático Médio (MSE)

Corresponde à média do **quadrado** das diferenças entre cada amostra original e a amostra correspondente após o processo de codificação/decodificação. Quanto maior, pior a qualidade.

$$MSE = \frac{SSD}{M \times N} \quad (1)$$

Estimação de Movimento (EM)

Processo que busca o VM e a *referência temporal* que, quando usados no processo de *compensação de movimento*, minimizam a relação *taxa-distorção* para um dado *bloco*.

Fatia (Slice)

Um conjunto de MB de um mesmo quadro, numerados seqüencialmente, que podem ser codificados isoladamente dos MBs das outras *fatias* do mesmo quadro. Cada *fatia* pode ser do tipo I, P, B, SI ou SP – restringindo os possíveis tipos de seus MBs.

Fluxo de Bits (Bitstream)

Uma seqüência de bits que adota a sintaxe especificada pelo padrão H.264 e que codifica um determinado vídeo.

Intra

Modo de predição/codificação que utiliza apenas as amostras da *fatia* atual (abreviação de “intraquadros”).

Inter

Modo de predição/codificação que utiliza apenas amostras de outros quadros (predição temporal), através da *compensação de movimento*. Beneficia-se do fato de que, entre um quadro e outro, boa parte da imagem sofre apenas uma translação (abreviação de “interquadros”).

Luma

Corresponde às amostras do sinal acromático (componente Y' do espaço de cor Y'CbCr), que representa o brilho. *Luma* é diferente de *luminância*, que é uma medida puramente fotométrica e independente de dispositivo, definida pelo CIE.

Macrobloco (MB)

Unidade básica de codificação de vídeo. Um bloco de 16×16 de amostras *luma* e dois blocos com as amostras correspondentes de *croma*.

Média das Diferenças Absolutas (MAD/MAE)

Média do valor absoluto das diferenças entre cada amostra original e a amostra correspondente após o processo de codificação/decodificação. Quanto maior, pior a qualidade.

$$MAD = \frac{SAD}{M \times N} \quad (2)$$

Otimização Taxa-Distorção (OTD)

Processo que realiza várias codificações com parâmetros diferentes, visando minimizar uma medida de taxa-distorção. A medida mais comum é o custo Lagrangiano.

Parâmetro de Quantização (QP)

Parâmetro que seleciona o nível de truncamento dos coeficientes. Quanto mais alto o QP, maior o truncamento e, portanto, menor a qualidade do vídeo decodificado.

Partição

Um bloco de amostras resultantes do *particionamento* de um *sub-macrobloco* para a predição *inter*.

Pixel

A palavra *pixel* deriva de “*picture element*”. *Pixel* é um ponto indivisível de uma imagem digital, o que equivale a uma amostra *luma* junto com as subamostras *croma* correspondentes.

Quadro

Todas as amostras que formam uma imagem estática. No caso de vídeo entrelaçado, o quadro é o conjunto do *campo superior* e do *campo inferior* (correspondentes às linhas pares e às linhas ímpares).

Razão Sinal-Ruído de Pico (PSNR)

Medida de qualidade objetiva de vídeo mais amplamente usada. Quanto maior o PSNR, melhor a qualidade. O PSNR é geralmente calculado sobre as amostras *luma* e é dado em decibéis através da seguinte expressão (para amostras de n bits):

$$PSNR = 10 \log_{10} \frac{(2^n - 1)^2}{MSE} \quad (3)$$

Referência Temporal

Parâmetro que seleciona qual dos quadros previamente decodificados é utilizado para a *compensação de movimento* de um bloco.

Residual (ou Erro Residual)

A diferença entre os valores *preditos* e os valores *decodificados* das amostras. Equivale também à diferença entre o valores *preditos* e os valores *originais*, encadeados aos seguintes processos: *transformação* (em coeficientes), *quantização* (truncamento), *quantização inversa* e *transformação inversa*.

Soma das Diferenças Absolutas (SAD/SAE)

Medida de distorção que é a soma do valor absoluto das diferenças entre cada amostra original e a amostra correspondente após o processo de codificação/decodificação.

$$SAD = \sum_{i=0}^M \sum_{j=0}^N |Ori_{i,j} - Dec_{i,j}| \quad (4)$$

Soma das Diferenças Quadráticas (SSD/SSE)

Medida de distorção aditiva que corresponde à soma do *quadrado* das diferenças entre cada amostra original e a amostra correspondente após o processo de codificação/decodificação.

$$SSD = \sum_{i=0}^M \sum_{j=0}^N (Ori_{i,j} - Dec_{i,j})^2 \quad (5)$$

Sub-macrobloco

Corresponde a um quarto do MB, ou seja, um bloco de 8×8 de amostras *luma* e dois blocos com as amostras correspondentes de *croma*.

Taxa de Bits

Quantidade de bits necessária para codificar **um segundo de vídeo** (geralmente em *kbps – kilobits por segundo*).

Taxa-Distorção (TD)

Conjunto de medidas de desempenho de codificação que são inversamente proporcionais à *taxa de bits* e à *distorção*. Uma dessas medidas é o **custo lagrangiano**.

INTRODUÇÃO

A compressão de vídeo permite reduzir a quantidade de dados necessária para representar vídeos digitais, diminuindo os custos de transmissão e armazenamento dos mesmos.

As principais aplicações do vídeo digital são:

- Armazenamento e distribuição de filmes.
- Difusão de TV digital aberta e por assinatura.
- Videoconferência e vídeo-mensagem.

Atualmente, estas aplicações utilizam as seguintes tecnologias como meios de armazenamento e comunicação:

- Mídias óticas de acesso aleatório, tais como CD, DVD e os novos formatos HD-DVD e Blu-Ray. Essas mídias são geralmente usadas em equipamentos com *hardware* dedicado, conectados a um sistema de TV.
- Internet e redes IP (fluxo contínuo e *multicasting*) – um meio adequado para videoconferências em tempo real, principalmente com redes que possuem garantia de qualidade de serviço (QoS). Também utilizada para distribuição *on-line* e *off-line* de filmes e programas de TV, cuja demanda tem aumentado muito com a crescente utilização de conexões domésticas de banda larga.
- Radiodifusão (*wireless*) – A maior parte da transmissão de vídeo entre satélites como também as transmissões diretas via satélite atualmente são digitais. Em médio prazo, a radiodifusão terrestre de vídeo analógico também deverá ser suplantada pela radiodifusão digital. O meio de telefonia

celular, que já provê transmissão de dados digitais, também tem possibilitado a utilização de vídeo digital.

Estas tecnologias são de alto custo e limitadas em termos de capacidade, por isso a compressão de vídeo é essencial para a viabilidade das aplicações de vídeo digital. Em um DVD, por exemplo, caberiam apenas 3,7 minutos de vídeo sem compressão¹. Além de reduzir os custos, a compressão de vídeo auxilia também na introdução de novas tecnologias, como a TV de alta definição (HDTV).

Para que os diferentes aparelhos e *softwares* de compressão e descompressão de vídeo sejam compatíveis, é necessário que possuam especificações em comum, i.e., que adotem um determinado padrão. Os DVDs, por exemplo, utilizam o padrão MPEG-2 [1], que possibilita comprimir vídeos para uma taxa de bits em torno de 20 vezes menor do que a taxa original [2]. O padrão de codificação de vídeo mais recente é o H.264 [3], que em geral reduz a taxa de bits entre duas e três vezes mais do que MPEG-2 [4].

Esta dissertação de mestrado aborda duas propostas de modificação para o H.264, visando obter um maior ganho de desempenho, significando uma taxa de bits menor e/ou uma qualidade maior do sinal de vídeo. As modificações consistem em introduzir dois novos tipos de macroblocos, o macrobloco *duplo-inter* e o macrobloco *híbrido*. Cada modificação, separadamente, consegue obter ganhos de até 0,25 dB na PSNR.

1.1. Breve Histórico

A padronização de técnicas de codificação de vídeo digital é essencial para permitir que equipamentos de vídeo digital de diferentes fabricantes se comuniquem. Atualmente, os principais órgãos responsáveis por tal padronização são o ITU-T *Video Coding Experts Group* (VCEG) e o ISO/IEC *Moving Picture Experts Group* (MPEG).

O ITU-T (ITU *Telecommunication Standardization Sector*) é um setor da ITU (*International Telecommunication Union*), localizado em Genebra, Suíça. Antes de 1992, o

¹ Um vídeo digital não comprimido que adota a recomendação ITU-R BT.601 necessita de uma taxa de 166 Mbits/s. Um DVD tem capacidade de armazenar 4.7 Gbytes, ou seja, $(4.7 \times 8 \times 1000) / (166 \times 60) = 3,7$ min de vídeo não comprimido.

ITU era conhecido por CCITT (*Comité consultatif international téléphonique et télégraphique*) – comitê internacional responsável pela telegrafia e telefonia.

O MPEG é um grupo de trabalho conjunto das instituições ISO e IEC (*International Organization for Standardization* e *International Electrotechnical Commission*).

Os primeiros esforços para a padronização da codificação de vídeo digital datam do final da década de 80 e início da década de 90, com a criação do padrão H.261 pelo ITU-T. O ITU-T focou inicialmente na compressão de vídeo digital priorizando aplicações em telefonia, enquanto o MPEG considerava a mais outros aspectos relevantes, tais como, codificação de áudio, sincronização, e multiplexação de áudio e vídeo.

A Fig. 1.1 apresenta a cronologia dos padrões de codificação que foram desenvolvidos por tais órgãos.

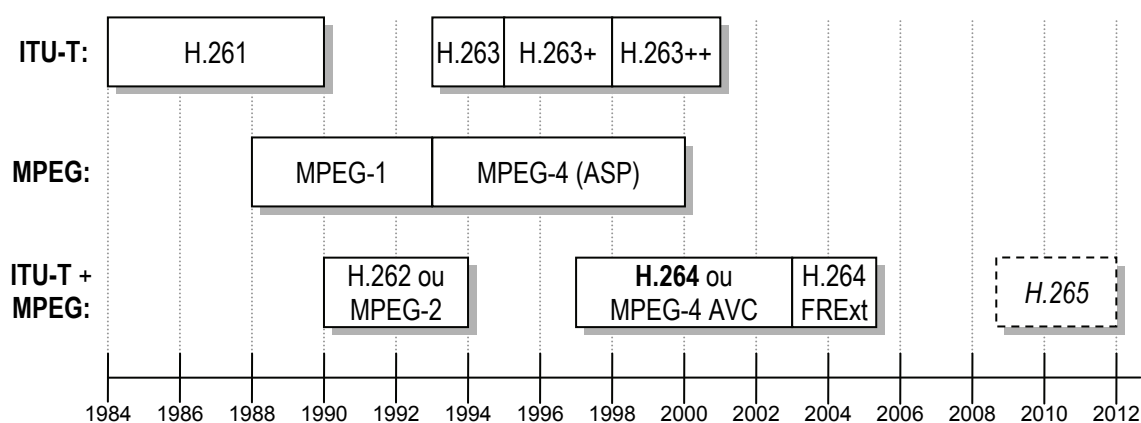


Fig. 1.1 – Cronologia do desenvolvimento dos diversos padrões de compressão de vídeo, apresentados isoladamente pelos órgãos ITU-T e MPEG, ou em um esforço conjunto de ambos [5].

O padrão de codificação de vídeo digital mais recente, o H.264, define vários perfis de aplicação (cada um com um subconjunto de características). Um deles, o *H.264 High Profile*, atualmente faz parte de especificações e padrões importantes, tais como:

- As especificações para difusão terrestre, por satélite e por cabo TS 101 154 e TS 102 005, do DVB;

- As especificações do HD-DVD e BD (Blu-Ray Disc), padrões esses considerados substitutos aos usados atualmente nos DVDs. Essas especificações exigem o *High Profile* do H.264 em todos os decodificadores [6];
- O SBTVD – Sistema Brasileiro de Televisão Digital [7]. Devido ao emprego do H.264 HP nas especificações de compressão de vídeo digital do SBTVD, esse sistema possui uma grande vantagem em relação à maioria dos outros sistemas de difusão terrestre de TV digital, que, por terem sido criados há mais tempo, ainda utilizam o padrão MPEG-2.

1.2. H.264

O H.264 possui uma série de vantagens em relação a seus antecessores, (principalmente em comparação ao MPEG-1, MPEG-2 e H.263 [8]), com respeito às etapas de predição, transformação e codificação. Muitas dessas novas características apareceram primeiro nos anexos do H.263 (resultando no H.263+ e H.263++), bem como no padrão MPEG-4 ASP [9].

Não há uma característica que se destaque isoladamente – todas as novas características trazem pequenas melhorias que, conjuntamente, provêm um grande impacto na taxa-distorção do H.264 relativamente a seus antecessores. Todas essas novas características agregam uma complexidade à codificação como também à decodificação, tanto em termos de complexidade computacional – sendo necessários *hardwares* muito mais eficientes, principalmente para o codificador – quanto em termos de complexidade de desenvolvimento, pois existem muito mais processos envolvidos e muito mais interdependências entre tais processos.

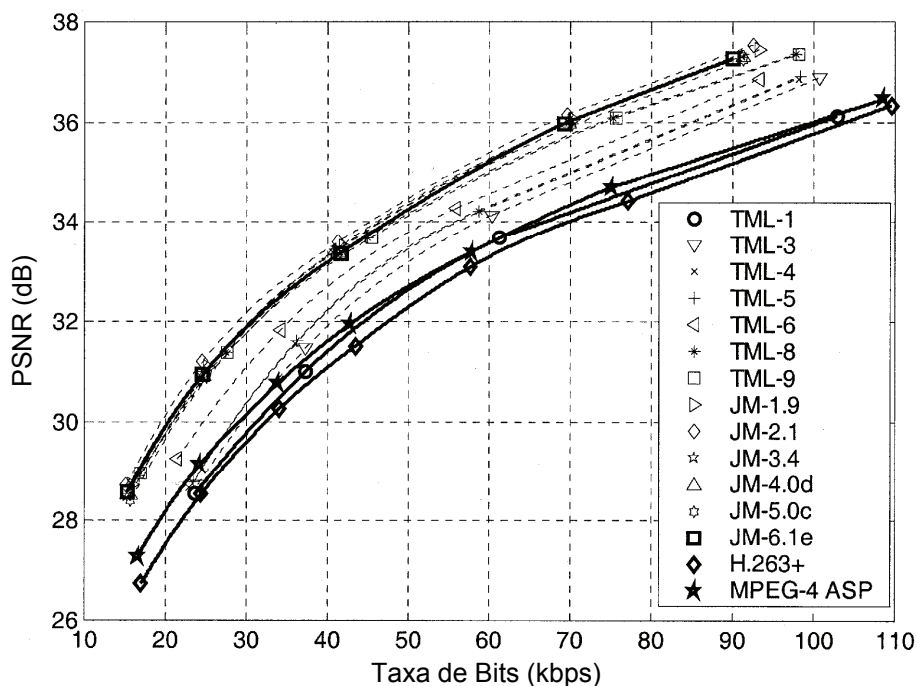


Fig. 1.2 – Curva taxa-distorção para a sequência “Foreman”, no formato QCIF, a 10 quadros/s, sendo codificada com vários dos protótipos desenvolvidos durante o processo de padronização do H.264, mostrando a evolução dos mesmos, em comparação com o H.263+ e o MPEG-4 ASP [10].

O H.264, inicialmente H.26L, começou a ser desenvolvido apenas pelo ITU-T VCEG (*Video Coding Experts Group*), no mesmo período em que se iniciou o desenvolvimento da última versão do H.263.

Em conjunto com o desenvolvimento das especificações do H.264, foi sendo criado o modelo de referência de codificação/decodificação, com o objetivo de avaliar os diferentes processos sendo propostos. Esse modelo consiste de um *software*, escrito em linguagem C, que contém todos os processos necessários para a completa codificação de um sinal de vídeo em um **fluxo de bits**² comprimido e posterior decodificação desse fluxo de bits no sinal de vídeo “original” (com alguma distorção, já que se trata de uma compressão com perdas).

As primeiras versões do modelo de referência eram denominadas TML (*Test Model Long-Term*), sendo que o primeiro TML foi lançado em Agosto de 1999. Quando o VCEG se juntou ao MPEG no *Joint Video Team* (JVT), para continuar o desenvolvimento em conjunto do padrão H.264, as versões passaram a se chamar *Joint Model* (JM). A Fig. 1.2

² O **fluxo de bits** (*bitstream*) é uma sequência de bits que adota a sintaxe especificada pelo padrão H.264 e que codifica um determinado vídeo.

mostra os pequenos ganhos de desempenho do codificador que foram sendo obtidos, desde o desenvolvimento da versão TML-1 até a JM-6.1e, lançada pouco antes da primeira versão oficial do padrão em abril de 2003.

Recentemente, foram adicionadas as primeiras extensões ao padrão H.264, denominadas *fidelity range extensions (FRExt)*, que visam suprir a demanda com vistas à codificação de vídeo em “alta fidelidade” (muito utilizada em TVs de alta definição e para edição e produção de vídeo) [6]. Já existe previsão, provavelmente para depois de 2010, para a criação de um novo padrão, o H.265, visando reduzir a taxa de bits pela metade (mantendo a mesma qualidade do H.264). Para que tal objetivo seja alcançado, será necessário investir em diferentes pontos do codificador, resultando provavelmente em um conjunto de “pequenas melhorias” nos processos de predição, transformação e codificação. [11], [12].

1.3. Organização da Dissertação

Esta dissertação de mestrado propõe duas modificações no padrão de codificação de vídeo H.264. Tais modificações têm como objetivo melhorar a taxa-distorção do codificador em questão. Como as modificações tornam o código resultante não compatível com o padrão atual do H.264, a pesquisa realizada se dá na direção de uma futura extensão ao H.264. Essa extensão se daria da mesma forma que as extensões que foram realizadas para o H.263 (H.263+ e H.263++) – consistindo de conjuntos de propostas que trazem diversas melhorias, porém modificando o padrão original e quebrando a compatibilidade. Possivelmente, as melhorias poderão também ser integradas ao padrão H.265, partindo da mesma filosofia de se obter várias “pequenas melhorias” que serão somadas a outras.

O Capítulo 2 apresenta uma introdução concisa aos principais aspectos da codificação de vídeo: sinais de vídeo digital, espaços de cores, particionamento, predição, transformação, quantização e codificação de entropia.

No Capítulo 3, são apresentadas as principais características do padrão H.264 que o diferenciam dos demais padrões de codificação até então, junto com alguns fundamentos da codificação híbrida de vídeo. As características discutidas se subdividem entre as relacionadas com transformadas, com predição ou com a própria codificação.

No Capítulo 4, o processo de codificação de macroblocos é descrito com uma maior profundidade, pois esse é o principal processo afetado pelas modificações aqui propostas.

O Capítulo 5 apresenta a primeira proposta de modificação do padrão H.264. Essa reside na inclusão de um novo tipo de macrobloco, denominado **macrobloco duplo-inter**. Esse macrobloco visa melhorar a relação taxa/distorção do codificador em determinadas situações, principalmente quando uma pequena parte do macrobloco possui um comportamento bem diferente do restante.

O Capítulo 6 discute uma segunda proposta de modificação, desta vez a inserção de um tipo de macrobloco denominado **macrobloco “híbrido” inter/intra**. Esse novo tipo de macrobloco visa trazer uma flexibilidade maior ao codificador, possibilitando uma granularidade menor na escolha entre as codificações *inter* e *intra*.

Finalizamos com o Capítulo 7, no qual são apresentados os comentários e conclusões, como também algumas propostas para trabalhos futuros.

INTRODUÇÃO À CODIFICAÇÃO DE VÍDEO

2.1. Introdução

O objetivo deste capítulo é apresentar uma breve introdução aos aspectos gerais da codificação de vídeo. Este capítulo apresenta os aspectos de sinais de vídeo digital, espaços de cores, particionamento, predição, transformação, quantização, reordenação e codificação de entropia.

2.2. Sinal de Vídeo Digital

Todo sinal de vídeo consiste de uma seqüência de **quadros** amostrados temporalmente, que, quando reproduzidos, geram a ilusão de movimento. Em um sinal de vídeo digital, esses quadros passam também por uma amostragem **espacial**, na qual cada amostra representa uma determinada intensidade de **brilho** ou de **cor**.

A Fig. 2.1 mostra um típico sinal de vídeo digital, ilustrando as amostragens temporal e espacial.

O sinal de vídeo digital pode ser progressivo ou entrelaçado. No primeiro caso, é realizada uma amostragem espacial de todas as linhas; no segundo caso, alterna-se temporalmente entre amostrar linhas pares e linhas ímpares. Os aparelhos de TV baseados em CRT (analógicos) utilizam varredura entrelaçada, enquanto os aparelhos usando tecnologia de plasma e LCD (digitais) utilizam em sua maioria a varredura progressiva. Utilizaremos neste trabalho apenas a codificação de vídeo progressivo, que está sendo cada vez mais empregada.

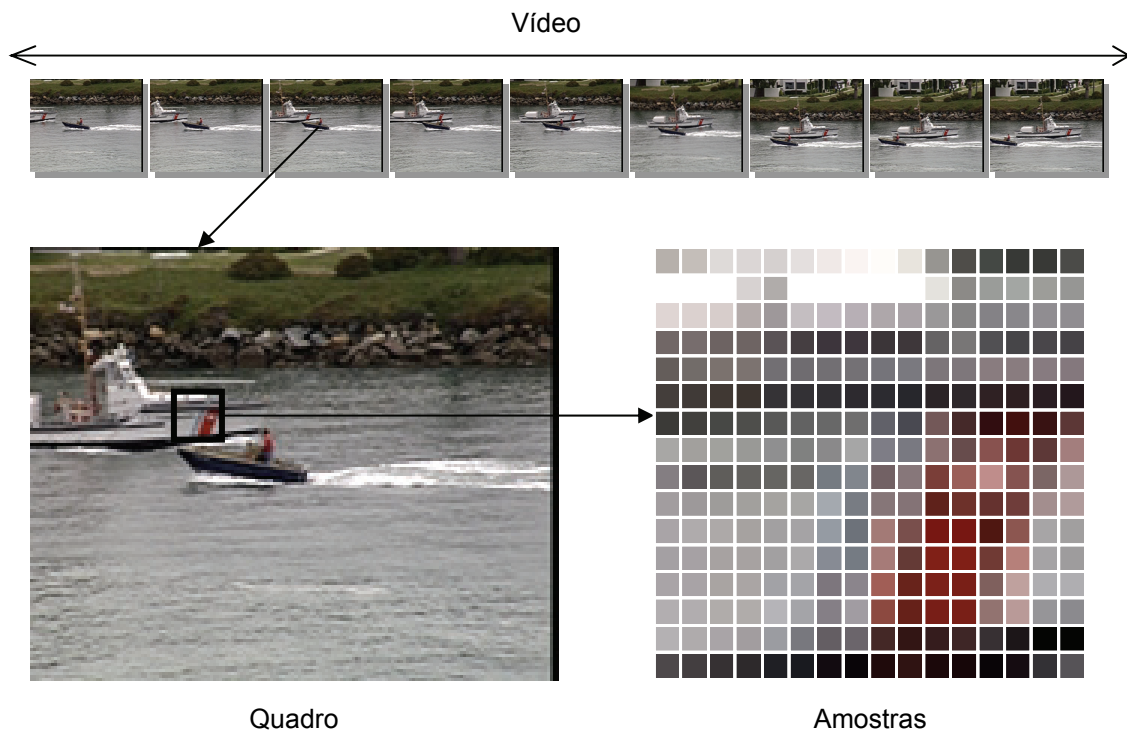


Fig. 2.1 – Ilustração de amostragem temporal de um vídeo e da amostragem espacial de um quadro.

2.3. Compressão Com Perdas e Sem Perdas

A compressão de vídeo é uma técnica que faz parte do grupo mais geral de tecnologias de compressão de dados, que se classificam em **compressão sem perdas** e **compressão com perdas**. No primeiro caso, após o processo de compressão/descompressão, o sinal reconstituído é idêntico ao original, enquanto no segundo caso há uma degradação desse sinal, denominada **distorção**.

A compressão sem perdas consegue, geralmente, compactar dados de duas a três vezes. Já na compressão com perdas, a taxa resultante depende apenas da distorção admitida – as tecnologias de compressão mais recentes podem comprimir para uma taxa até 100 vezes menor do que a taxa original, com uma distorção ainda aceitável.

A compressão sem perdas limita-se a um pequeno nicho de aplicações que não toleram qualquer distorção, como vídeos médicos ou sistemas de arquivamento. Por sua vez, a compressão com perdas é a mais usada e difundida, já que certas distorções podem

ser imperceptíveis ao olho humano, ou mesmo toleradas – é este o tipo de compressão que trata o padrão H.264 aqui apresentado.

2.4. Espaços de Cores

No sistema óptico humano, a percepção das cores se deve principalmente às células fotossensíveis cones, que se dividem em três tipos de fotorreceptores cujas respostas variam para comprimentos de onda diferentes. Por isso, para que seja possível criar todas as cores que podemos visualizar, são sempre necessários três sinais de cor.

Em todos os tipos de TVs e monitores coloridos, sejam baseados em LCD, plasma ou CRT, as cores são formadas a partir da combinação da emissão de três cores “primárias”: vermelho, verde e azul – este sistema denomina-se RGB (*Red-Green-Blue*).

Estas cores primárias compõem a base de um **espaço de cor** RGB de três dimensões, no qual cada ponto constitui uma diferente cor. Apesar de ser um espaço de cor adequado para os dispositivos de visualização, essa representação não é tão propícia para a codificação de vídeo, já que existe uma alta correlação entre os três sinais, resultando em uma certa redundância quando usada tal representação.

Através de transformações lineares ou não-lineares, pode-se representar as cores em outros diferentes espaços de cor, tal como o Y’CbCr, que é mais apropriado para a codificação de vídeo, pois:

- não apresenta tão alta redundância quanto o RGB;
- separa o sinal representativo de intensidade de luz (*luma*) dos dois sinais responsáveis pela colorização (crominância ou *croma*) – a vantagem dessa separação está apresentada na próxima seção denominada Subamostragem Croma.

A conversão de RGB para Y’CbCr é padronizada pelo ITU-R BT.601 [13], sendo realizada da seguinte forma:

- A partir do sinal RGB, deriva-se o sinal R'G'B' aplicando-se uma operação de exponenciação denominada correção gama. Esta correção mapeia o sinal para um domínio que é mais uniforme perceptualmente (Fig. 2.2).

Intensidade linear	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Com correção gama	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0

Fig. 2.2 – Diferença entre um sinal linear e um com correção gama.

- A partir dos componentes R', G' e B' (que estão no intervalo entre zero e um), deriva-se o sinal Y'PbPr (analógico), da seguinte forma:

$$\begin{aligned}
 Y_p' &= Kr \cdot R' + (1 - Kr - Kb) \cdot G' + Kb \cdot B', & Y_p' &\in [0,1] \\
 Pb &= \frac{1}{2} \cdot \frac{B' - Y_p'}{1 - Kb}, & Pb &\in \left[-\frac{1}{2}, \frac{1}{2}\right] \\
 Pr &= \frac{1}{2} \cdot \frac{R' - Y_p'}{1 - Kr}, & Pr &\in \left[-\frac{1}{2}, \frac{1}{2}\right]
 \end{aligned}
 \tag{2.1}$$

Os parâmetros Kb e Kr são especificados pelo padrão ITU-R BT.601, sendo

$$\begin{aligned}
 Kb &= 0,114 \\
 Kr &= 0,299
 \end{aligned}
 \tag{2.2}$$

- Os componentes do sinal Y'PbPr são remapeados para novos intervalos e arredondados, produzindo o sinal digital Y'CbCr. Assim,

$$\begin{aligned}
 Y' &= \lfloor (235 - 16) \cdot Y_p' + 16\frac{1}{2} \rfloor, & Y' &\in [16, 234] \cap \mathbb{N} \\
 Cb &= \lfloor (240 - 16) \cdot Pb + 16\frac{1}{2} \rfloor, & Cb &\in [16, 240] \cap \mathbb{N} \\
 Cr &= \lfloor (240 - 16) \cdot Pr + 16\frac{1}{2} \rfloor, & Cr &\in [16, 240] \cap \mathbb{N}
 \end{aligned}
 \tag{2.3}$$

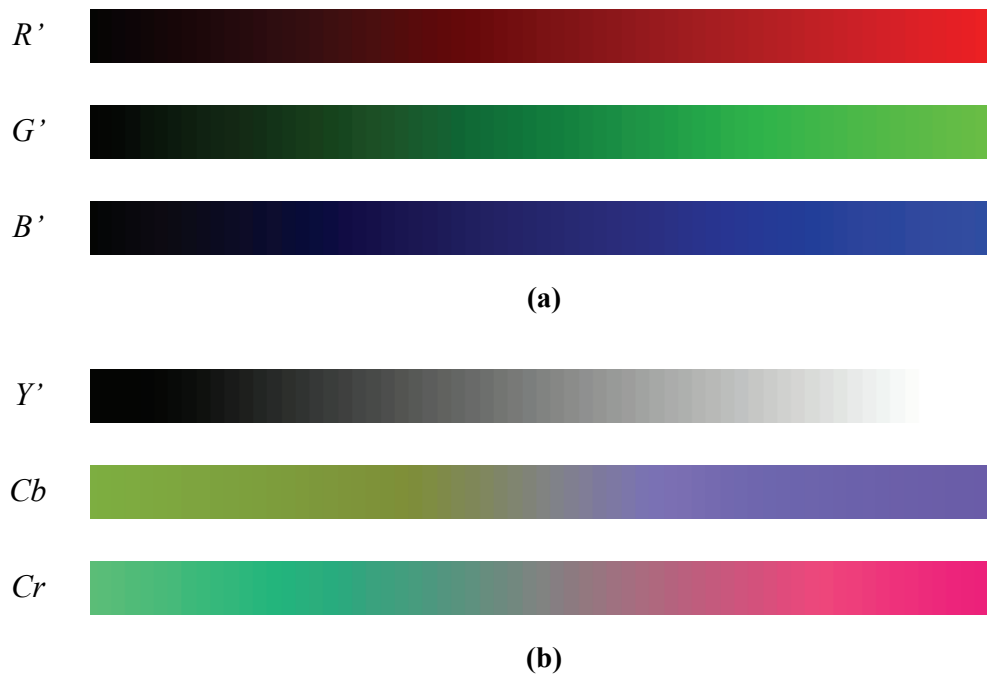


Fig. 2.3 – Bases (eixos) dos espaços de cores. (a) Base do R'G'B'. (b) Base do Y'CbCr.

2.5. Subamostragem Croma

Devido ao sistema óptico humano ser mais sensível à intensidade de luz que às cores, pode-se utilizar uma resolução menor na amostragem dos sinais *croma* quase sem prejuízos à qualidade subjetiva das imagens. Existe uma nomenclatura padrão para as diferentes formas de subamostragem comumente utilizadas em codificação de vídeo:

- No formato 4:4:4, não há subamostragem, ou seja, para cada amostra Y' , existe uma amostra Cb e uma amostra Cr correspondentes;
- No formato 4:2:2, a resolução dos dois sinais *croma* é reduzida pela metade horizontalmente;
- No formato 4:2:0, a resolução dos dois sinais *croma* é reduzida pela metade tanto horizontalmente quanto verticalmente, ou seja, para cada quatro amostras Y' existe uma amostra Cb e uma amostra Cr .

A posição das amostras está ilustrada na Fig. 2.4. Para que uma imagem com subamostragem *croma* seja mostrada corretamente, as amostras *croma* devem ser interpoladas nas posições das amostras *luma*, antes da conversão para o espaço de cores RGB.

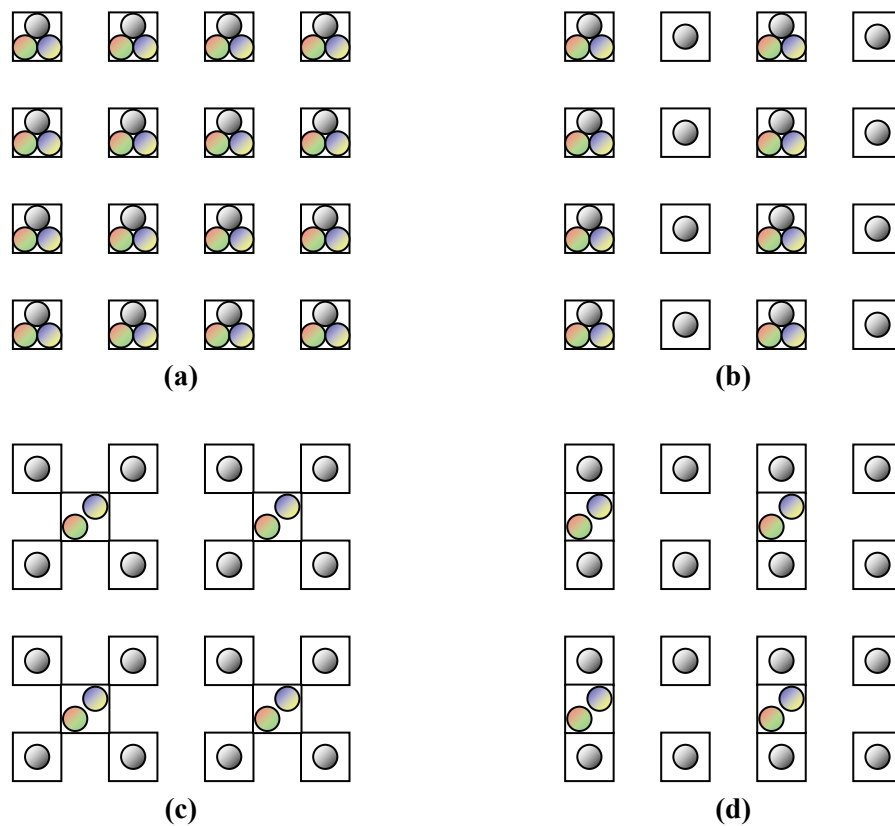


Fig. 2.4 - Posição relativa das amostras *luma*, representadas pelos círculos cinzas, e das amostras *croma*, representadas pelos círculos azul/amarelo (*Cb*) e vermelho/verde (*Cr*), em diferentes formatos de subamostragem. (a) Formato 4:4:4. (b) Formato 4:2:2. (c) Formato 4:2:0 do MPEG-1. (d) Formato 4:2:0 do MPEG-2 e do H.264.

2.6. Particionamento

Em diversos padrões de codificação de vídeo, a unidade básica de codificação é o **macrobloco** (MB). O MB corresponde a uma matriz de 16×16 amostras *luma* mais amostras *croma* correspondentes. Por exemplo, no formato 4:2:0, cada MB consistirá de

uma matriz de 16×16 amostras *luma* (Y') e duas matrizes de 8×8 amostras *croma* (uma para Cb e outra para Cr), como mostra a Fig. 2.5(b).

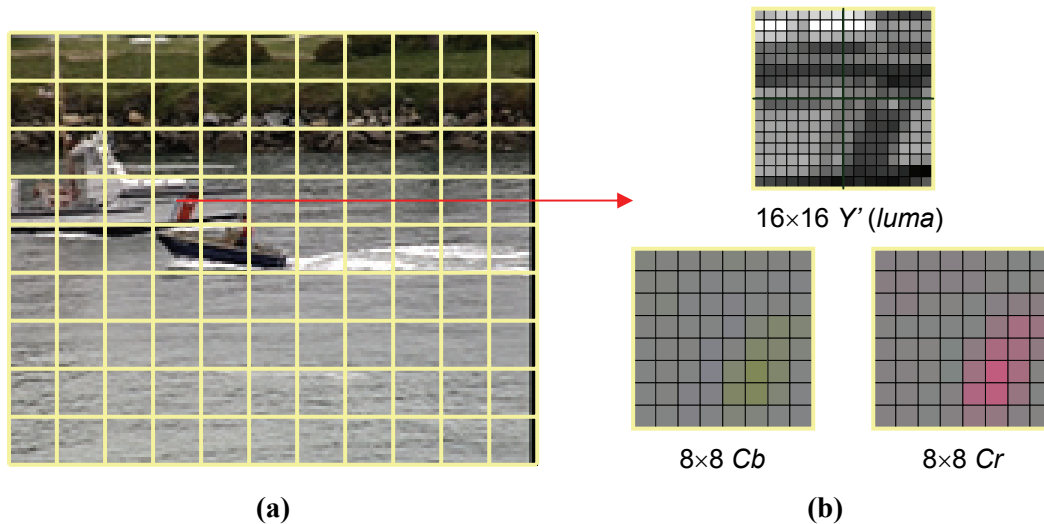


Fig. 2.5 – Divisão de um quadro em macroblocos. (a) Divisão do quadro em 99 macroblocos (grade de 11 MBs na horizontal \times 9 MBs na vertical). (b) Blocos de amostras Y' , Cb e Cr de um MB.

A divisão de cada quadro em pequenos blocos de amostras é fundamental para o bom desempenho da maioria dos processos envolvidos na codificação de vídeo, como a compensação de movimento e a transformação. Como cada bloco possui características distintas, essa divisão permite que cada diferente bloco seja codificado com os parâmetros e coeficientes mais apropriados. Além disso, as amostras de blocos pequenos tendem a ser mais homogêneas, formando imagens relativamente simples, o que auxilia ainda mais no desempenho de tais processos.

Especificamente no padrão H.264, os MBs podem ainda ser subdivididos em blocos menores, tanto para fins de compensação de movimento quanto para fins de transformação. Quando um bloco de 16×16 amostras *luma* é dividido em 4 blocos de 8×8 amostras, esses blocos são denominados submacroblocos (SubMB).

O processo encarregado em determinar o melhor VM para cada bloco durante a codificação denomina-se **estimação de movimento** (EM). No padrão H.264, a CM pode ser realizadas em blocos de diferentes tamanhos.

A CM é um processo de **predição interquadros**. No padrão H.264, existe também um processo de predição **intraquadros** que utiliza as amostras de um mesmo quadro – este processo está descrito em detalhes na Seção 3.3.1. O resultado do processo de predição é a entrada para o processo de transformação.

2.8. Transformação

Esse processo consiste em aplicar uma transformação nos valores das amostras de um bloco, de forma a:

- Transformar o sinal para o domínio da **frequência espacial**;
- Reduzir as redundâncias espaciais, já que amostras próximas possuem valores semelhantes;
- Descorrelacionar o sinal, concentrando a energia (informação) em alguns poucos coeficientes.

A transformada ótima para concentração de energia é a KL (Karhunen-Loeve), que depende de informações estatísticas do sinal. A maioria dos padrões de codificação de vídeo e imagens utiliza a transformada DCT (*discrete cosine transform*) [14], que é mais simples que a KL e possui resultados muito próximos, sem depender de informações adicionais.

O H.264 utiliza uma transformada mais simples, que utiliza apenas números inteiros, cujo resultado se aproxima muito da transformada DCT 4×4, dado um escalamento adequado. Para uma matriz 4×4 de valores de amostras **A**, a matriz de coeficientes **C** é escrita por

$$\mathbf{C} = \mathbf{TAT}^T \quad (2.4)$$

Onde **T** é a matriz da transformada inteira, dada por:

$$\mathbf{T} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \quad (2.5)$$

Após realizada a transformada, a informação (ou energia) do sinal concentra-se nos primeiros elementos da matriz \mathbf{C} ($c(1,1)$, $c(1,2)$ e $c(2,1)$), que correspondem às freqüências espaciais mais baixas.

Exemplificando, vamos supor que o resultado da CM em um bloco 4×4 seja a seguinte matriz:

$$\mathbf{A} = \begin{bmatrix} 111 & 108 & 104 & 123 \\ 166 & 165 & 156 & 158 \\ 56 & 46 & 40 & 36 \\ 4 & -4 & 4 & 13 \end{bmatrix} \quad (2.6)$$

Aplicando a transformada, temos:

$$\mathbf{C} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} 111 & 108 & 104 & 123 \\ 166 & 165 & 156 & 158 \\ 56 & 46 & 40 & 36 \\ 4 & -4 & 4 & 13 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \quad (2.7)$$

$$\mathbf{C} = \begin{bmatrix} 1286 & 25 & 48 & -15 \\ 1325 & -9 & 7 & -72 \\ -360 & -117 & 30 & -11 \\ -505 & 48 & 11 & 9 \end{bmatrix}$$

O coeficiente resultante de mais baixa freqüência, $c(1,1)$, denomina-se coeficiente DC, enquanto os demais coeficientes são denominados coeficientes AC, em analogia aos componentes de corrente direta e de corrente alternada em um sinal elétrico.

2.9. Quantização

Para tornar menores os valores dos coeficientes resultantes da transformação, aplica-se o processo de **quantização**, que é equivalente à uma divisão seguida por um arredondamento.

$$\mathbf{C}_Q = \left\lfloor \mathbf{C} \div \mathbf{Q} + \frac{1}{2} \right\rfloor \quad (2.8)$$

A Eq. 2.8 mostra como se obtém os coeficientes quantizados \mathbf{C}_Q a partir da matriz de quantização \mathbf{Q} . A operação \div denota a divisão elemento-a-elemento, e a operação $\lfloor \cdot \rfloor$ denota a função piso (maior inteiro menor ou igual a). No H.264, o processo de quantização é mais complexo, pois utiliza apenas aritmética inteira (multiplicação e deslocamento de bits), entretanto o efeito é o mesmo.

Os valores da matriz \mathbf{Q} são tabelados, e variam de acordo com o **parâmetro de quantização** (QP). Quanto maior o QP, maiores os valores da matriz \mathbf{Q} . Por exemplo, para um QP = 30, a matriz \mathbf{Q} do H.264 é aproximadamente igual a

$$\mathbf{Q} = \begin{bmatrix} 80 & 130 & 80 & 130 \\ 130 & 200 & 130 & 200 \\ 80 & 130 & 80 & 130 \\ 130 & 200 & 130 & 200 \end{bmatrix} \quad (2.9)$$

Portanto, para a matriz \mathbf{C} obtida de (2.7), temos a seguinte matriz de coeficientes quantizados:

$$\mathbf{C}_Q = \left\lfloor \begin{bmatrix} 1286 & 25 & 48 & -15 \\ 1325 & -9 & 7 & -72 \\ -360 & -117 & 30 & -11 \\ -505 & 48 & 11 & 9 \end{bmatrix} \div \begin{bmatrix} 80 & 130 & 80 & 130 \\ 130 & 200 & 130 & 200 \\ 80 & 130 & 80 & 130 \\ 130 & 200 & 130 & 200 \end{bmatrix} + \frac{1}{2} \right\rfloor \quad (2.10)$$

$$\mathbf{C}_Q = \begin{bmatrix} 16 & 0 & 1 & 0 \\ 10 & 0 & 0 & 0 \\ -4 & -1 & 0 & 0 \\ -4 & 0 & 0 & 0 \end{bmatrix}$$

Os valores resultantes são menores necessitando menos bits na codificação; contudo existe uma perda na precisão. No processo de decodificação, quando esses valores forem reconstituídos, serão obtidos valores que diferem dos valores originais (já que a operação de arredondamento pode causar uma “perda” de informação). A maior parte da distorção é resultante do processo de quantização e, conseqüentemente, controlada pelo valor de QP.

2.10. Reordenação e Codificação de Entropia

De forma a utilizar o menor número possível de bits na codificação final, os coeficientes quantizados são primeiramente reordenados em uma lista de 16 elementos através da varredura em *zig-zag* (Fig. 2.7). Os primeiros elementos da lista são os que geralmente têm valores maiores e os últimos aqueles que provavelmente têm valor zero.

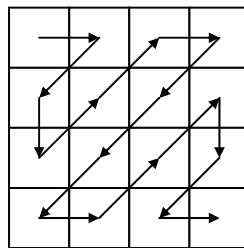


Fig. 2.7 – Varredura em *zig-zag*.

Para os elementos da matriz C_0 de (2.10), o resultado da varredura origina a seguinte lista: 16; 0; 10; -4; 0; 1; 0; 0; -1; -4; 0; 0; 0; 0; 0; 0. Como nas listas resultantes costuma haver muitos zeros em seqüência, essa lista é representada de forma especial no fluxo de bits, com a intenção de:

- Representar as seqüências de zeros de forma mais “compacta” (apenas através da quantidade de zeros de cada seqüência);
- Representar a seqüência de coeficientes que são ± 1 (e que precedem a última seqüência de zeros) de forma compacta, apenas através da quantidade dos mesmos, e dos respectivos sinais;

Os símbolos resultados são codificados utilizando **palavras de bits de comprimento variável**. Isso implica que símbolos mais freqüentes considerem menos bits do que os símbolos menos freqüentes.

CARACTERÍSTICAS PRINCIPAIS DO PADRÃO H.264

3.1. Introdução

Neste capítulo, são tratadas as principais características do padrão H.264 que o diferenciam dos padrões anteriores e que possibilitam a obtenção de um melhor desempenho de compressão. Essas características estão classificadas em três grupos: transformação, predição e codificação.

3.2. Transformação

3.2.1. Introdução

Essa etapa consiste em transformar o valor das amostras do domínio espacial para o domínio da frequência de forma a descorrelacionar as amostras, fazendo com que a maior parte da energia se concentre em poucos coeficientes de mais baixa frequência.

As vantagens do H.264 no que diz respeito às transformadas utilizadas são:

3.2.2. Transformada DCT Inteira com Inversa “Exata”

Esta é uma das grandes inovações do padrão H.264. Anteriormente, todos os padrões trabalhavam com transformada não-inteira. Devido a diferenças de precisão numérica nas diversas implementações da transformada inversa, as amostras resultantes da decodificação poderiam variar ligeiramente de decodificador para decodificador. Isso degradava o sinal que seria posteriormente utilizado para a predição, impossibilitando a utilização de métodos de predição mais sofisticados, como, por exemplo, a nova *predição intra*, assunto desta dissertação.

3.2.3. Transformada de Tamanho 4×4

Esta é outra evolução no processo de transformação, introduzida para tornar a transformada consistente com o tamanho do bloco em que se realizam as novas predições (como a predição *intra*). Com o bloco menor, também se reduz o efeito visual não desejado de “anelamento”, causado pelo recobrimento na frequência, que acontece nos padrões anteriores por utilizarem uma transformada de tamanho 8×8. Mais detalhes sobre a transformada utilizada no padrão H.264 podem ser encontrados em [15].

3.2.4. Transformadas “Hierárquicas”

Além da transformada DCT 4×4 (apresentada na Seção 2.8), o padrão H.264 também possibilita, em casos especiais, aplicar uma transformada Hadamard 2×2, 2×4 ou 4×4 nos coeficientes DC (de mais baixa frequência) dos blocos 4×4. Essa transformação extra estende a transformada 4×4 para os tamanhos 8×8, 8×16 (utilizadas para as amostras *chroma* de um MB) ou 16×16 (utilizada para as amostras de um MB *intra* especial denominado *Intra_16×16*).

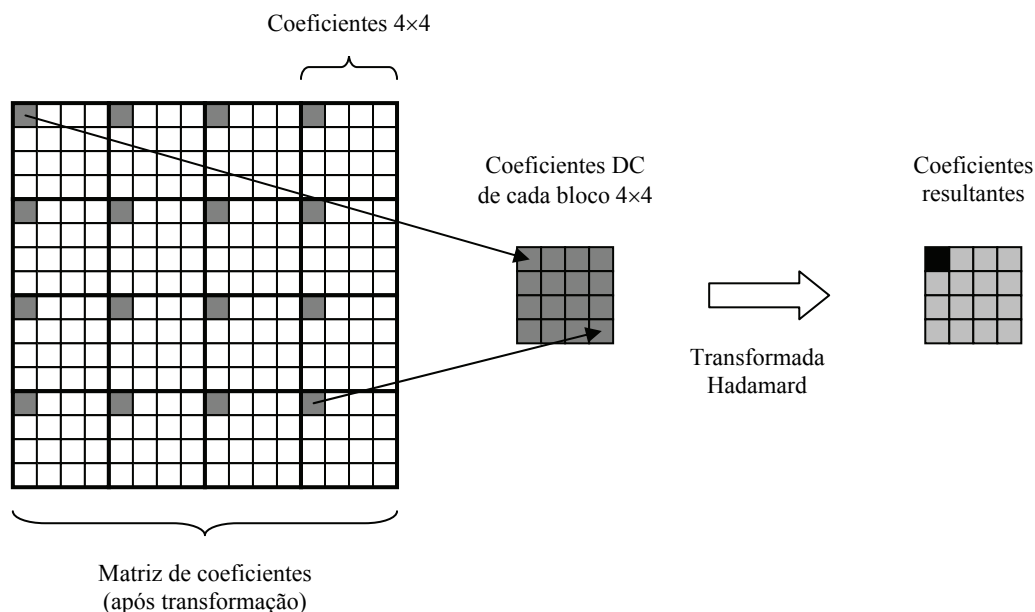


Fig. 3.1 – Ilustração da utilização da transformada Hadamard no padrão H.264.

Além disso, a extensão FExt do padrão também possibilita a codificação de um bloco 8×8 *luma* utilizando uma transformada DCT 8×8 inteira (intercambiável em nível de MB,

i.e., o codificador pode optar por usar a transformada 4×4 padrão ou a transformada 8×8 a cada MB).

3.2.5. Utilização de Aritmética de 16-bits

Anteriormente ao padrão H.264 era necessário utilizar operações com palavras de no mínimo 32-bits, além de uma aritmética de ponto flutuante. O H.264 especifica as transformadas utilizando apenas operações simples de adição, subtração e deslocamento, necessitando apenas de uma aritmética de 16-bits de ponto fixo. Isso reduz a complexidade do *hardware* necessário para computar a transformada, reduz a carga computacional, bem como permite alocar o processamento para outras etapas, por exemplo, predição ou decodificação.

3.3. Predição

Esta etapa consiste em utilizar valores de amostras previamente decodificadas para prever os valores das amostras correntes. Esses valores são subtraídos das amostras antes de passar pela transformação de forma a reduzir ainda mais os valores a serem codificados. A predição remove redundâncias temporais através de amostras de outros quadros (nesse caso, denominada **predição *inter***), ou redundâncias espaciais através de amostras do mesmo quadro (denominando-se predição *intra*).

Dentre as vantagens que o H.264 traz na etapa de predição, destacam-se:

3.3.1. Predição Intra 4×4

Uma das grandes novidades do H.264 é a utilização de amostras das bordas de blocos vizinhos, previamente decodificados, para realizar uma predição no domínio espacial. A predição mais comum é feita através de uma extrapolação direcional (isto é, repetição dos padrões em uma determinada direção), realizada a cada bloco *luma* 4×4, permitindo uma granularidade adequada na seleção do modo (direção) de predição.

A predição *intra*, junto com outras características relevantes a fatias *intra*, como também a nova transformada 4×4 garantem ao padrão H.264 uma maior taxa de

compressão, mesmo para comprimir imagens estáticas (i.e., um único quadro). Alguns estudos indicam que o modo *intra* do padrão H.264 tem desempenho melhor do que o padrão JPEG2000 para a compressão de imagens estáticas, e *muito* melhor ainda do que o padrão JPEG. [16]-[18]

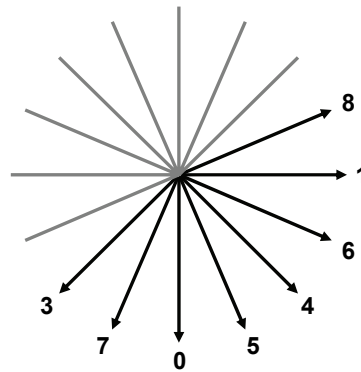


Fig. 3.2 – Sentido dos 8 modos de predição *intra* direcionais utilizados no H.264. O modo 2, que não está mostrado na figura, é o modo de predição DC, que utiliza a média dos vizinhos.

3.3.2. Precisão de Um Quarto de Amostra na Compensação de Movimento

O padrão H.264 permite, na compensação de movimento da predição *inter*, utilizar subamostras interpoladas em pontos intermediários com precisão de $\frac{1}{4}$ do tamanho da amostra. Por exemplo, é possível ter um vetor de movimento com o valor $(2\frac{3}{4}, -\frac{1}{2})$, como mostrado na Fig. 3.3. A maioria dos padrões anteriores permitia apenas uma precisão de meia amostra. Além de melhor prever os movimentos, essa interpolação também traz a possibilidade de se “escolher” uma filtragem passa-baixas para a predição, pois, dependendo da posição (inteira, meia amostra ou $\frac{1}{4}$ de amostra), tem-se inerentemente uma filtragem passa-baixas associada ao método de interpolação, o que muitas vezes ajuda na redução do erro residual.

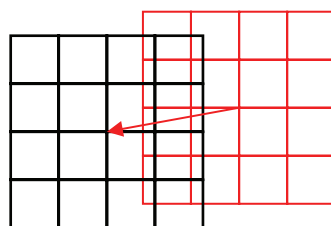


Fig. 3.3 – Exemplo de compensação de movimento utilizando um vetor não inteiro (neste caso: 2 amostras e $\frac{3}{4}$ à direita e $\frac{1}{2}$ amostra acima).

Uma desvantagem de se utilizar $\frac{1}{4}$ de amostra é que são necessários mais bits para codificar os VM, porém esse custo adicional é, na maioria das vezes, justificado pelos ganhos de predição.

3.3.3. Tamanhos de Blocos Variáveis na Compensação de Movimento

Além de se poder utilizar apenas um VM para o MB inteiro (16×16), o padrão H.264 também provê a flexibilidade de se particionar o MB e realizar a compensação de movimento em blocos menores. O MB pode ser particionado em dois blocos de 16×8, dois blocos de 8×16 ou quatro blocos 8×8 (denominados **partições**), sendo que, nesse último caso, cada bloco 8×8 pode ser novamente particionado em dois blocos 8×4, dois blocos 4×8 ou quatro blocos 4×4 (denominados **subpartições**), totalizando $3 + 4^4 = 259$ particionamentos possíveis.

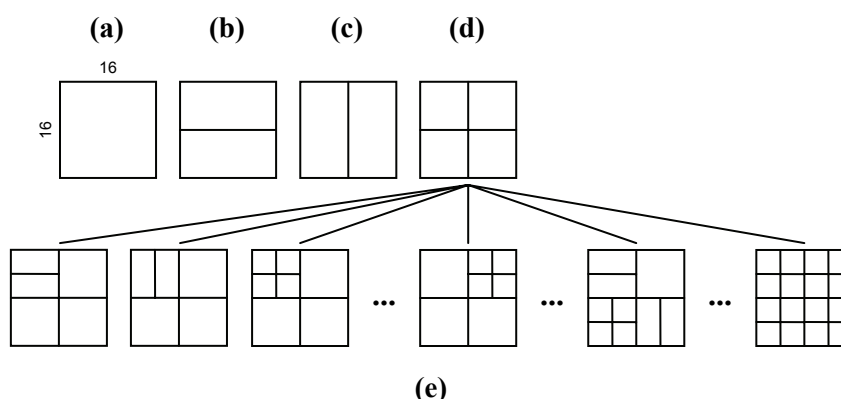


Fig. 3.4 – Exemplos de particionamentos possíveis utilizados para a compensação de movimento. (a) Tipo 0 (16×16). (b) Tipo 1 (16×8). (c) Tipo 2 (8×16). (d) Tipo 3 (8×8). (e) Variações do tipo das partições dos SubMBs, quando o tipo é 3.

3.3.4. Utilização de Múltiplas Referências Temporais

Em vez de utilizar apenas uma referência temporal para fazer a compensação de movimento – os padrões mais antigos permitiam utilizar apenas o quadro anterior – o H.264 permite que cada quadro mantenha uma lista de quadros de referência, sendo que cada vetor de movimento pode “selecionar” livremente qual desses quadros ele faz referência. Isso vale também para quadros tipo B (bipreditivos), sendo que, nesse caso, há duas listas de quadros referência. Essa “expansão” do espaço de busca temporal possibilita uma melhor predição *inter*, por exemplo, em casos em que ocorre a oclusão ou deformação de um determinado objeto na cena anterior; porém, o mesmo objeto aparece com uma representação idêntica a uns três ou quatro quadros anteriores. Uma outra vantagem introduzida pelo H.264 é que a ordem utilizada para referências temporais pode ser diferente da ordem de exibição dos quadros.

3.3.5. Predição de Movimento

A predição utilizada para os vetores de movimento já existia nos padrões anteriores; porém, o H.264 introduz novidades nesse processo – possibilita a predição de movimento para blocos *skip* (blocos “pulados”, sem qualquer informação extra de movimento ou coeficiente). Isso é útil para casos de movimento global da câmera; possibilita a predição de movimento nas partições internas do MB e possibilita uma inferência especial nos blocos tipo B (bipreditivos), que utiliza os VMs de dois quadros, denominada inferência **direta**.

3.3.6. VMs que Apontam para Fora do Quadro

Através de uma simples extrapolação (nesse caso, uma repetição) dos valores da borda do quadro, é possível utilizar VMs apontando parcialmente para fora do mesmo.

3.3.7. Filtro Antiblocagem Adaptativo

O H.264 define um filtro adaptativo utilizado para diminuir os artefatos de “blocagem”. Esses artefatos são causados pela descontinuidade entre um bloco e outro, que se deve a diferenças de codificações dos blocos e à quantização. Anteriormente, o filtro

antiblocagem era opcional, usado apenas como um pós-processamento do decodificador – já o padrão H.264 o insere diretamente no processo de reconstrução (que gera as imagens de referência a serem usadas para compensação de movimento), melhorando a qualidade da predição [19].

3.3.8. Predição com Peso

Nos padrões anteriores, os blocos bipreditivos (tipo B) produziam a amostra predita através da soma ponderada das amostras de dois quadros diferentes, ambas com peso $\frac{1}{2}$; os blocos predictivos (tipo P) não utilizavam pesos. O H.264 possibilita que os valores das amostras a sejam escalados por um peso p especificado, além de adicionados a um segundo valor b , como em

$$a' = ap + b \quad (3.1)$$

Os pesos podem variar dentro de uma mesma fatia e também no caso de fatia B, podem ser inferidos a partir dos pesos dos quadros de referência anterior e posterior.

Esse processo gera um ganho adequado de compressão para seqüências de imagens com efeitos de transição (*fade in*, *fade out* ou *crossfade*).

3.4. Codificação

A etapa de codificação consiste em organizar em um **fluxo de bits** os coeficientes resultantes das etapas de transformação e quantização. Utilizando uma **codificação de entropia** (por exemplo, através de códigos de tamanho variado), pode-se remover mais redundância e comprimir ainda mais o sinal de vídeo. Além disso, a codificação também se encarrega de organizar os dados de forma que sua transmissão em fluxo contínuo (*streaming*) seja realizada de forma segura, interoperável e com o mínimo de problemas e interrupções.

Para a etapa de codificação, o H.264 traz as seguintes novidades:

3.4.1. Organização dos Dados

A organização dos dados no padrão H.264 traz muita flexibilidade em relação aos padrões anteriores. Todas as estruturas de sintaxe são encapsuladas em “pacotes” de dados denominados NAL (*Network Abstraction Layer*), que podem então ser utilizados diretamente em sistemas de transporte ou armazenamento [20], tais como:

- H.222.0 (ou MPEG-2 systems) [21] – utilizado em transmissões por satélite, cabo ou terrestres;
- H.320 [22], H.323 [23], H.324 [24] ou SIP [25] – utilizados em telefonia e videoconferências;
- Sistemas RTP/Redes IP [26] – utilizados para transmissão em fluxo contínuo (*streaming*) via internet/intranet.

As unidades NAL que carregam os elementos de sintaxe da codificação de vídeo denominam-se VCL (Camada de Codificação de Vídeo - *Video Coding Layer*), enquanto as unidades que carregam informações “extras” ou conjuntos de parâmetros mais gerais (por exemplo, dados como o formato do vídeo ou o tipo de codificação de entropia) denominam-se não-VCL. As unidades não-VCL que carregam conjuntos de parâmetros têm grande importância pois afetam um grande número de unidades NAL; essa separação permite que tais unidades possam ser transportadas por um canal mais robusto, com menor probabilidade de perda de dados.

A VCL organiza o vídeo hierarquicamente em objetos encapsulados, como ilustrado pela Fig. 3.5. O MB, de tamanho fixo 16×16, é a unidade básica de codificação da VCL – ele pode ser subdividido em blocos menores para fins de predição ou transformação – mas a codificação é feita MB a MB.

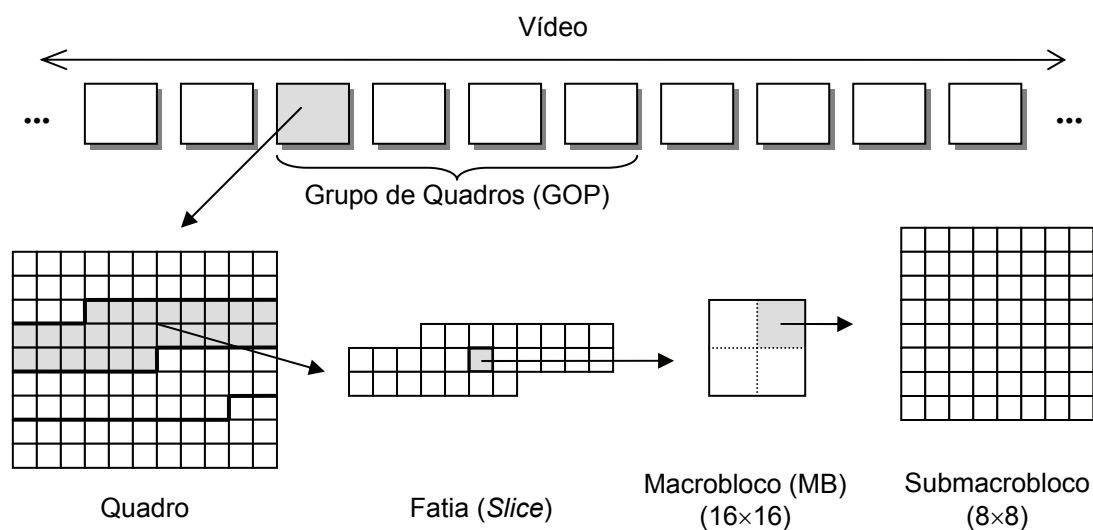


Fig. 3.5 – Organização hierárquica da camada de codificação de vídeo, subdividindo o vídeo em quadros, quadros em fatias e fatias em macroblocos.

3.4.2. Grupos de *Fatias*

O H.264 inova ao permitir particionar o quadro em **grupos de fatias** [27], que são subconjuntos de macroblocos não necessariamente seqüenciais no quadro. Um **grupo de fatias** pode conter uma ou mais **fatias**. Para reduzir a quantidade de parâmetros a ser transmitida, existem no padrão alguns tipos pré-definidos de particionamento do quadro em grupos, porém o usuário pode definir seu próprio mapeamento explícito. Os tipos de particionamento são mostrados na Tabela 3.1 e alguns exemplos estão ilustrados na Fig. 3.6. Os grupos de fatias podem melhorar significativamente a **robustez à perda de dados** e podem também ser utilizados para codificação de **regiões de interesse** com parâmetros distintos.

Uma outra inovação é a possibilidade de enviar e decodificar as fatias de um quadro em qualquer ordem, independente da ordem seqüencial das mesmas. Essa característica é denominada ASO (*Arbitrary Slice Order*).

Tabela 3.1 – Tipos pré-definidos de particionamento do quadro em grupos de fatias

Nº.	Direção = 0	Direção = 1	Máx. de Grupos	Parâmetros
0	<i>Interleaved</i>		-	Núm. de grupos
1	<i>Dispersed</i>		-	Núm. de grupos
2	<i>Foreground and background</i>		-	Núm de grupos, MB superior-esquerdo e inferior-direito de cada grupo
3	<i>Box-out clockwise</i>	<i>Box-out counter-clockwise</i>	2	Núm. de MB do grupo 0
4	<i>Raster scan</i>	<i>Reverse raster scan</i>	2	Núm. de MB do grupo 0
5	<i>Wipe right</i>	<i>Wipe left</i>	2	Núm. de MB do grupo 0
6	Explícito		-	Núm. do grupo de cada MB

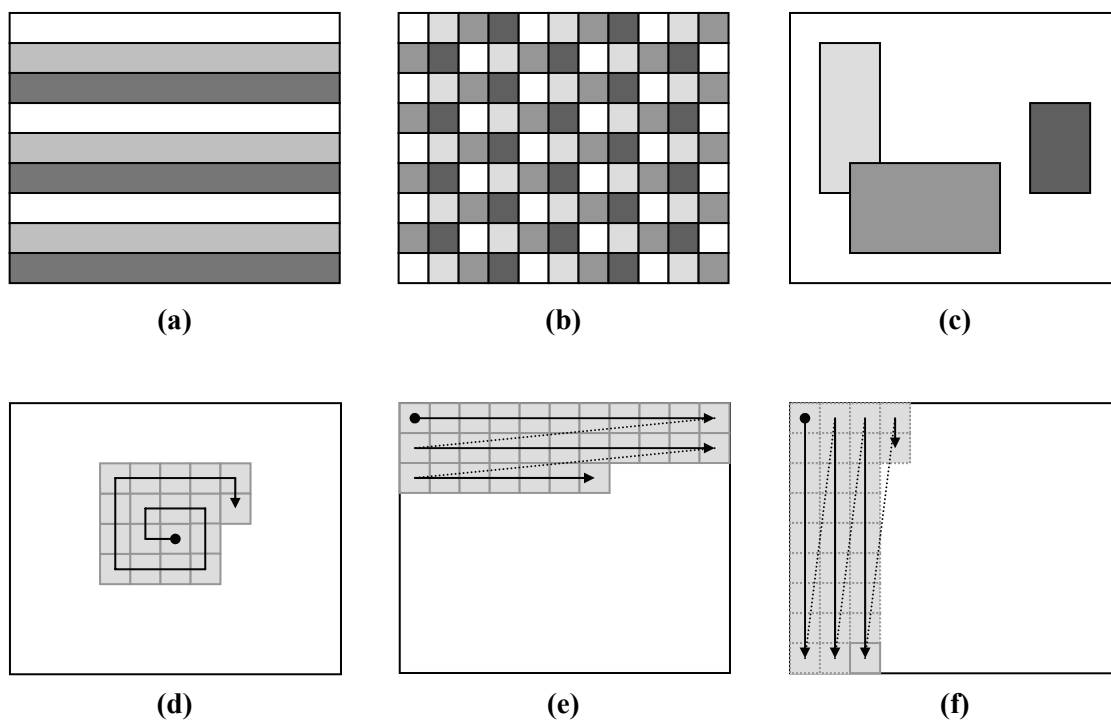


Fig. 3.6 – Exemplos dos seis tipos de mapeamento de grupos de fatias pré-definidos no H.264, cada grupo tem uma tonalidade diferente. (a) *Interleaved* com três grupos. (b) *Dispersed* com quatro grupos. (c) *Foreground and Background* com quatro grupos. (d) *Box-out* com 18 MB no grupo 0. (e) *Raster* com 29 MB no grupo 0. (f) *Wipe* com 29 MB no grupo 0.

3.4.3. Codificação de Entropia Adaptativa ao Contexto

A codificação de entropia consiste em atribuir seqüências binárias de tamanho diferentes para símbolos (coeficientes, VMs, parâmetros, etc.) com probabilidades diferentes de aparecerem, de forma a reduzir a quantidade total de bits a ser codificado.

Do ponto de vista de teoria da informação [28], para que a quantidade de bits total produzida seja a menor possível, cada símbolo deve ser representado por uma palavra binária com comprimento correspondente à sua entropia (ou seja, a informação que ele carrega). A informação $I(x)$ é diretamente relacionada à probabilidade $P(x)$ de um dado símbolo x , definida como

$$I(x) = -\log_2 P(x) \quad (3.2)$$

Por exemplo, um símbolo com probabilidade $\frac{1}{2}$ deverá ser codificado com 1 bit, enquanto um símbolo com probabilidade $\frac{1}{4}$ deverá ser codificado com 2 bits.

Na codificação adaptativa ao contexto, dado um símbolo s em um contexto c , em vez de se utilizar as probabilidades totais $P(s)$ para determinar as seqüências de bits, utilizam-se as probabilidades condicionais $P(s | c)$.

Existem dois modos de codificação no H.264, o CAVLC (*Context Adaptive Variable Length Coding*) e o CABAC (*Context Adaptive Binary Arithmetic Coding*), sendo que ambos são adaptativos ao contexto, isso é, consideram as probabilidades condicionais dos símbolos no contexto dos símbolos previamente decodificados. Dessa forma, podem ser ainda mais eficientes.

No caso do CAVLC, são utilizados códigos de tamanho variável para representar os símbolos, sendo que tais códigos são previamente definidos e tabelados pelo padrão. Por exemplo, ao se codificar o número de coeficientes iniciais não-zero do bloco atual³, como também o número de coeficientes com valor ± 1 que os seguem, o contexto é dado por algumas regras que dependem se os coeficientes codificados referem-se a um bloco *croma* ou *luma*. Nesse último caso, depende também dos números de coeficientes não-zero dos blocos vizinhos. Para cada contexto diferente, uma tabela diferente é utilizada.

O CAVLC será visto com mais detalhes na Seção 4.3.

3.4.4. Codificação de Entropia Aritmética

Como a utilização de palavras de comprimento variável permite apenas a utilização de códigos com uma quantidade inteira de bits, ela não é eficiente quando as

³ A codificação dos coeficientes é tratada na Seção 2.10.

probabilidades não são potências exatas de $\frac{1}{2}$ – principalmente para representar símbolos com probabilidade maior do que $\frac{1}{2}$, pois esses teriam que ser representados com menos do que 1 bit. Tal limitação pode ser suplantada utilizando codificação aritmética [29].

A codificação aritmética baseia-se na subdivisão recursiva de intervalos de números, sendo que a distribuição dos intervalos corresponde à distribuição de probabilidade dos símbolos.

Geralmente a codificação aritmética possui um custo computacional alto por utilizar muitas operações de multiplicação e divisão com palavras longas. O mecanismo utilizado no CABAC permite, através de um esquema de **binarização** dos símbolos, reduzir o alfabeto a ser codificado, levando a uma codificação com um custo computacional mais baixo, não muito maior do que o do CAVLC. Com uma modelagem de contexto avançado, praticamente não há desvantagens em se utilizar esse esquema de binarização.

Em média, o CABAC consegue reduzir a taxa de bits entre 9% e 14% em relação ao CAVLC [30] (sem qualquer perda de qualidade, pois a codificação de entropia é sem perdas). Apesar deste ganho, ele ainda é um processo muito mais complexo do que o CAVLC.

CODIFICAÇÃO DOS MACROBLOCOS NO H.264

4.1. Introdução

A principal unidade de codificação é o macrobloco (MB), que é um bloco de 16 amostras de largura e 16 amostras de altura (como o formato de vídeo é 4:2:0, isso corresponde a 16×16 amostras do canal *luma* e 8×8 amostras dos dois canais *chroma*, Cb e Cr).

Os MBs são numerados seqüencialmente em cada quadro e são organizados em estruturas denominadas **fatias** (*slices*). Um quadro pode conter uma ou mais fatias e o número de fatias pode variar de um quadro para outro, assim como o número de MB por fatia é variável (no caso geral, o quadro contém apenas uma fatia, que corresponde ao próprio quadro). As fatias são codificadas “desacopladas” umas das outras (i.e., sem a necessidade de valores codificados em outras fatias).

Um MB pode ser codificado nos seguintes modos:

I (Intra) – Intra-codificado, i.e., sem utilizar compensação de movimento (CM) de quadros de referência. Pode utilizar predição baseada em MB vizinhos do mesmo quadro, previamente decodificados.

P (Predito) – Pode ser predito a partir da compensação de movimento (CM) de quadros de referência.

B (Bi-predito) – Pode ser predito a partir de um ou dois quadros de referência.

SI – Utilizado para fazer a transição entre duas seqüências diferentes (por exemplo, uma com menos taxa de bits do que a outra), para aplicações de fluxo contínuo (*streaming*) adaptativo ou para a realização de avanço rápido de cenas (*fast-forward*).

As fatias também se dividem nos tipos I, P, B, SP e SI.

Tabela 4.1 – Tipos de MB permitidos em cada tipo de fatia

Fatia tipo	Pode conter MB tipo
I	I
P	P ou I
B	B ou I
SP	P ou I
SI	SI

4.2. Processos

O padrão recomendado pelo ITU-T em conjunto com o MPEG para o H.264, seguindo a tradição dos padrões anteriores, apenas especifica a **decodificação** de uma seqüência de bits. A vantagem é que isso oferece mais liberdade no projeto dos codificadores, cuja única restrição é que gerem seqüências de bits **decodificáveis**.

Para ser bem aceito, o **codificador** deve produzir uma seqüência com uma relação taxa/distorção adequada, i.e., procurando minimizar a taxa de bits para um fator de qualidade especificado ou buscando minimizar a distorção para uma dada taxa de bits. O padrão provê um **decodificador** e um **codificador de referência**, cujos principais processos estão esquematizados de forma simplificada nas Figs. 4.1 e 4.2, respectivamente.

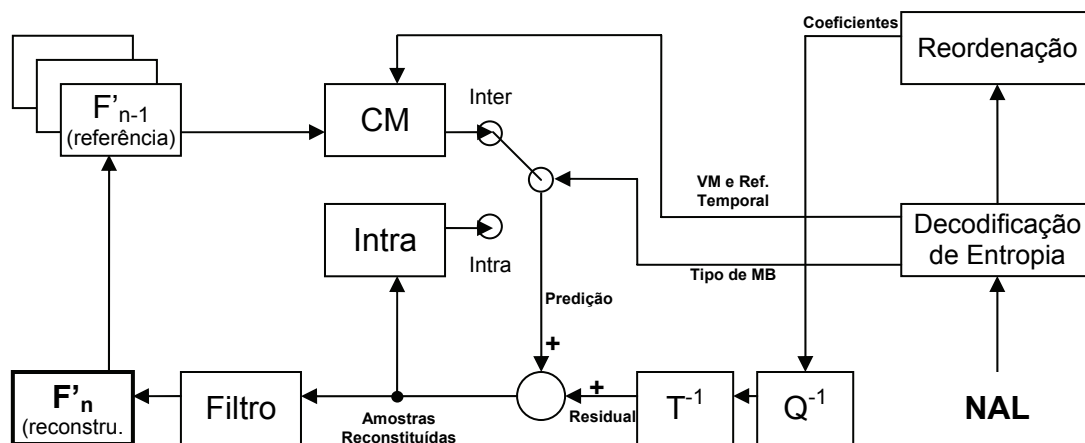


Fig. 4.1 – Esquema simplificado do encadeamento dos processos no decodificador do padrão H.264.

Os processos de decodificação são descritos a seguir:

- **NAL** – A seqüência de bits que provém da *Network Abstraction Layer*, e será decodificada.

- **Decodificação de Entropia** – Processo que, dados os bits codificados em CAVLC ou CABAC, produz os **elementos de sintaxe** (tais como, os coeficientes e vetores de movimento).
- **Reordenação** – Como o padrão H.264 permite a transmissão de elementos fora de ordem, esse processo se ocupa em ordená-los antes da decodificação.
- Q^{-1} – Quantização inversa dos coeficientes (equivalente a uma multiplicação).
- T^{-1} – Transformada inversa dos coeficientes. Realiza a transformada DCT inteira 4×4 inversa, transformando os coeficientes do domínio da frequência em amostras do domínio espacial. Produz o **erro residual**.
- $F'_{n-1}, F'_{n-2}, \dots$ – Quadros de referência (quadros previamente reconstruídos), sendo que só é armazenado um número limitado de quadros, geralmente entre um e dez.
- **CM** (Compensação de Movimento) – Através dos vetores de movimento do MB atual e dos quadros de referência, compõe a **predição *inter***.
- **Intra** – Processo que produz as amostras da **predição *intra***, dadas as seguintes informações: tipo de predição e MBs previamente decodificados do quadro atual, vizinhos ao MB atual.
- A **predição (*intra* ou *inter*)**, somada ao **erro residual**, gera as amostras reconstituídas. Essas últimas ainda passam por um **filtro** “antiblocagem”, que suaviza as bordas entre blocos, eliminando artefatos de codificação que aparecem devido à codificação separada de cada bloco;
- O resultado é F'_n , o quadro reconstruído (decodificado) atual.

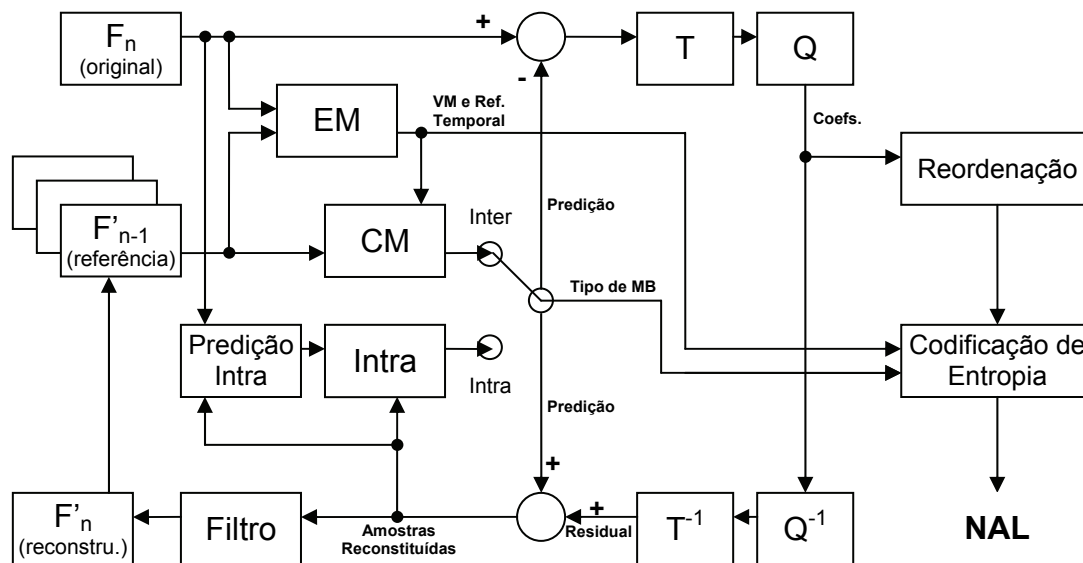


Fig. 4.2 - Esquema do encadeamento dos processos no codificador do padrão H.264.

O codificador reutiliza vários dos processos do decodificador, pois ele não pode utilizar os quadros originais como referência, já que eles não estarão disponíveis no decodificador. Por isso, o codificador deve gerar as **referências temporais** através de uma reconstrução idêntica à do decodificador. A seguir, são descritos os processos exclusivos da codificação:

- **EM** – Estimação de Movimento. Um dos processos mais custosos em termos de processamento – consiste em fazer uma busca na região em torno de um MB, em todos os quadros de referência, para determinar o **vetor de movimento** que minimiza o erro residual (dada uma determinada métrica). Quanto maior a região de busca e em quanto mais quadros de referência se procurar, melhor será o resultado, porém com um custo computacional maior. Existem diversos algoritmos que buscam “atalhos” para reduzir o tempo de EM sem muito prejuízo à relação taxa/distorção.
- **Predição Intra** – Consiste em testar os diversos tipos de predição *intra*, i.e., utilizando os valores dos MB vizinhos previamente decodificados do quadro atual.
- **T** – Transformada do bloco. Realiza a transformada DCT inteira 4×4 , transformando as amostras da diferença entre os valores originais e preditos do bloco, no domínio espacial, em coeficientes no domínio da frequência. Além da

transformada DCT inteira 4×4, também pode ser considerada uma transformada DCT inteira 8×8.

- **Q** – Quantização dos coeficientes (equivalente a uma divisão). Na quantização é que efetivamente se troca qualidade por taxa. Quanto maior a quantização, mais bits se perdem, portanto, maior é a degradação da imagem. Porém, como os coeficientes tornam-se menores, eles podem ser codificados a uma taxa menor.

4.3. Codificação de Entropia nos Macroblocos

Nesta seção revisaremos com um maior aprofundamento o processo de codificação de entropia CAVLC. Tal revisão é necessária para o melhor entendimento das seções subseqüentes.

4.3.1. Codificação Exp-Golomb sem sinal

O processo de codificação **ue(v)** (*unsigned Exp-Golomb*) é um dos processos mais utilizados na codificação de tamanho de palavra variável no H.264, sendo simples e de baixa complexidade computacional, sem a necessidade de tabelas ou árvores de código.

A *Exp-Golomb* é uma codificação na qual o tamanho do código é proporcional ao valor a ser codificado, sendo ideal para comprimir dados cujos valores pequenos têm uma frequência muito grande. Esse código consiste de duas partes: uma codificação **unária** de tamanho variável e uma codificação **binária** de tamanho fixo (dado pelo valor da codificação unária). A codificação unária é simplesmente uma seqüência de N bits de valor zero, seguidos por um bit de valor 1 ($N \geq 0$). A codificação binária é uma seqüência binária comum de N bits (onde cada bit pode assumir os valores zero ou um). A Tabela 4.2 mostra os primeiros códigos **ue(v)**, indicando o valor de N e destacando em negrito a seqüência unária:

Tabela 4.2 – Seqüências Exp-Golomb ue(v)

Seqüência	Valor	N (Seq. Unária)
1	0	0
010	1	1
011	2	1
00100	3	2
00101	4	2
00110	5	2
00111	6	2
0001000	7	3
0001001	8	3
0001010	9	3
0001011	10	3
...	...	
0001111	14	3
000010000	15	4

Uma desvantagem de tal codificação é que o tamanho das seqüências de bits torna-se muito grande quando os valores vão se tornando maiores. Em geral, do valor $2^{N-1} - 1$ até $2^N - 2$, necessitam-se $2N - 1$ bits, portanto, o comprimento do código para um definido valor n é dado em bits por

$$c(n) = 1 + 2 \lfloor \log_2(n+1) \rfloor \text{ bits} \quad (4.1)$$

De acordo com a teoria da informação, $c(n)$ é o comprimento de código ideal para um elemento com função de distribuição de probabilidade dada por

$$f(n) = 2^{-c(n)} = 2^{-1-2 \lfloor \log_2(n+1) \rfloor} \quad (4.2)$$

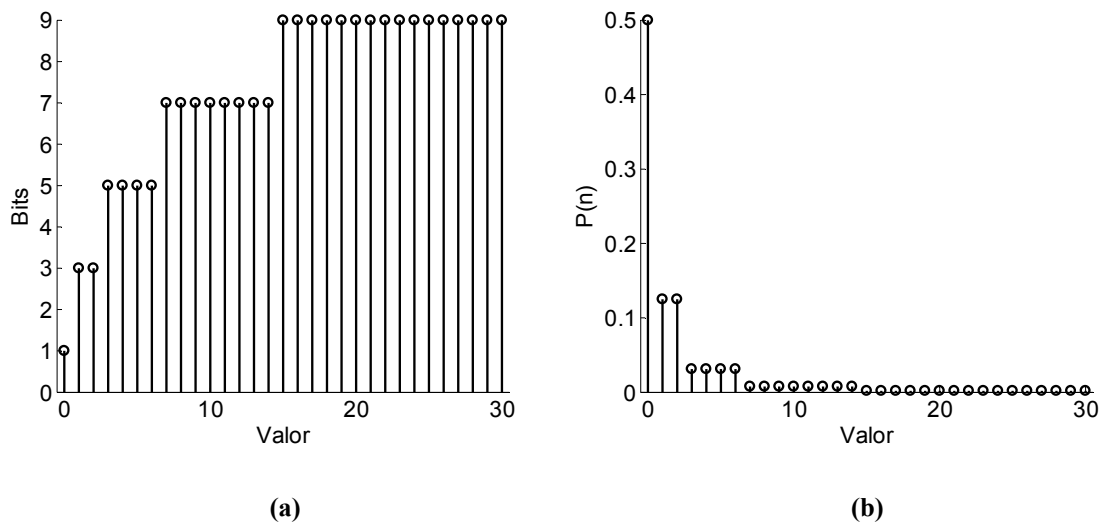


Fig. 4.3 – Tamanho de código $ue(v)$ e a distribuição de probabilidade ideal. (a) Tamanho de código. (b) Distribuição de probabilidade ideal.

A Fig. 4.3 ilustra tal função de distribuição de probabilidade. Como podemos ver, as probabilidades decaem rapidamente, por exemplo, o ideal é que os valores entre 7 e 14 possuam uma probabilidade de aproximadamente 0,00781, enquanto valores de 15 a 30, uma probabilidade de aproximadamente 0,00195.

Esta codificação é ideal quando apenas valores baixos serão codificados ou quando os valores baixos são realmente muito mais frequentes.

4.3.2. Codificação Exp-Golomb com sinal

A $se(v)$ é uma variação da $ue(v)$ (Exp-Golomb sem sinal), que permite a codificação de números negativos. Antes de serem codificados, os números positivos n são mapeados para $2n - 1$, enquanto os números negativos $-n$ são mapeados para $2n$. O número positivo resultante do mapeamento é então codificado como $ue(v)$ normalmente.

Neste caso, temos as seguintes funções de comprimento de código e de distribuição de probabilidade ideal. Assim,

$$c(n) = 1 + 2 \lfloor \log_2 (2|n| + 1) \rfloor \tag{4.3}$$

$$f(n) = 2^{-c(n)} = 2^{-1-2 \lfloor \log_2 (2|n| + 1) \rfloor} \tag{4.4}$$

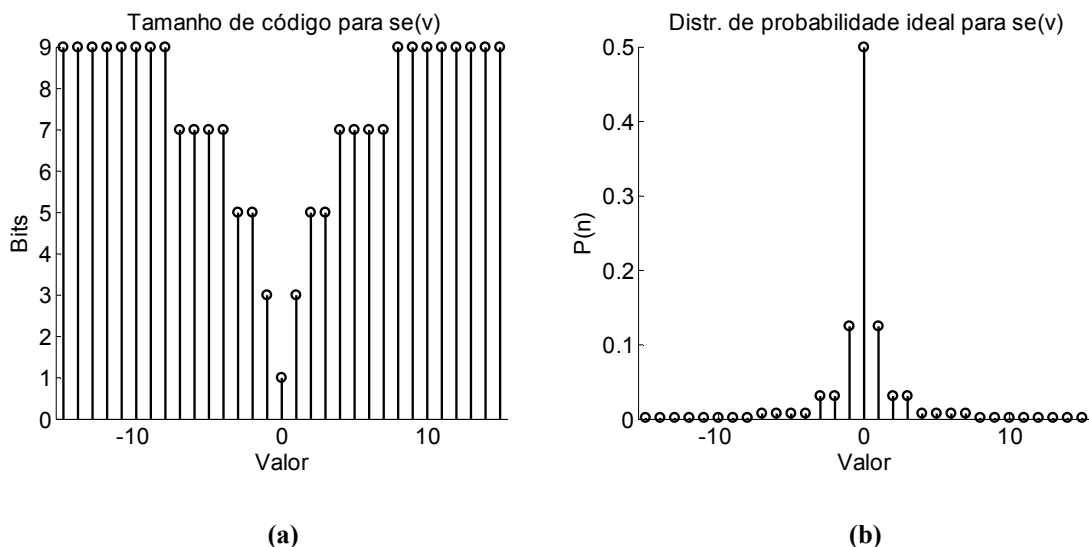


Fig. 4.4 – Tamanho de código $se(v)$ e a distribuição de probabilidade ideal. (a) Tamanho de código. (b) Distribuição de probabilidade ideal.

4.4. Predição

De forma a reduzir os valores das amostras a serem transformadas, é realizada uma predição no MB, tirando proveito das redundâncias temporais e espaciais. Uma introdução à predição é apresentada na Seção 2.7. A predição tem como objetivo diminuir os valores a serem transformados, o que implica em coeficientes menores e, portanto, uma taxa de bits menor.

Para que o decodificador possa refazer o mesmo processo de predição (para então somar os valores resultantes da transformada inversa), os parâmetros usados nessa predição (tipo de predição, vetores de movimento, etc.) também devem ser codificados.

A predição no padrão H.264 acontece em todos os MBs (com exceção de MBs do tipo I_PCM) e se divide em dois tipos: predição *intra* e *inter*. O codificador é livre para fazer qualquer tipo de predição, porém, geralmente, vários tipos de predição possíveis são testadas, escolhendo-se aquela que produz o melhor resultado com vistas à taxa-distorção.

4.4.1. Predição *Inter*

Como citado na Seção 3.3.3, a predição *Inter* possibilita particionar o MB em diversas formas diferentes. O MB pode ser considerado uma única partição 16×16 , duas

partições 16×8 , duas partições 8×16 ou quatro partições 8×8 . A cada uma delas é atribuído uma referência temporal (um quadro, da lista de quadros possíveis) e um vetor de movimento. Se a partição for de 8×8 , ela poderá ser subdividida novamente em duas subpartições de 8×4 , duas subpartições de 4×8 ou quatro subpartições de 4×4 – nesse caso, a cada subpartição é atribuído um vetor de movimento diferente, mas todas as subpartições do mesmo bloco 8×8 devem possuir a mesma referência temporal.

No caso do VM resultar em posições não-inteiras (meia-amostra ou um quarto de amostra), uma interpolação é realizada. Os valores nas posições de meia-amostra são calculados com um filtro FIR com coeficientes $(\frac{1}{32}, -\frac{5}{32}, \frac{20}{32}, \frac{20}{32}, -\frac{5}{32}, \frac{1}{32})$.

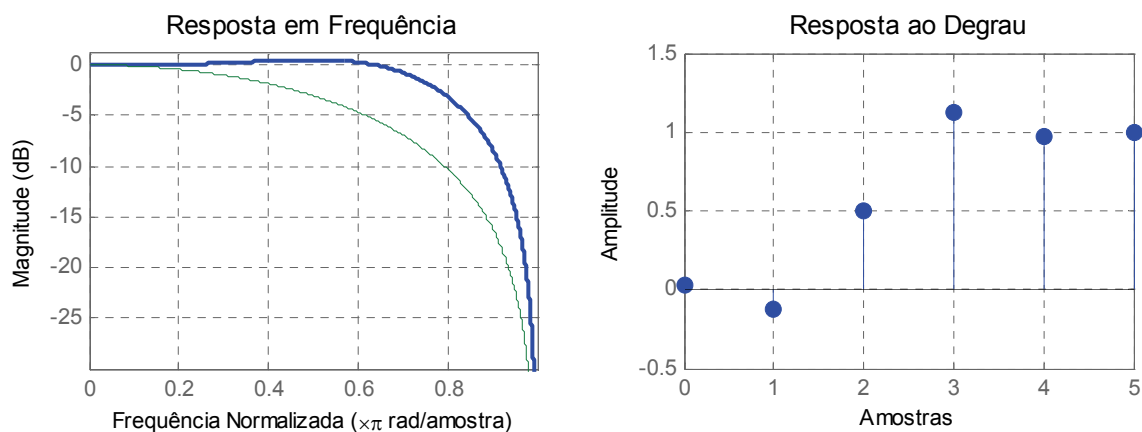


Fig. 4.5 – Características (resposta em frequência e resposta ao degrau) do filtro utilizado para a interpolação das meias-amostras. Na resposta em frequência, o filtro é comparado com o filtro de interpolação $(\frac{1}{2}, \frac{1}{2})$.

Tal filtro é relativamente complexo comparado com aqueles comumente utilizados na interpolação de sinais de vídeo; entretanto, é um filtro qualitativamente superior, sendo eficaz para obter um adequado desempenho na compensação de movimento. A Fig. 4.5 mostra a resposta em frequência, com apenas um pequeno *overshoot* de 0,5 dB e uma atenuação maior do que 6 dB apenas nas frequências normalizadas maiores do que $0,866\pi$ (a atenuação é calculada apenas no sinal interpolado).

Dependendo da posição, a amostra é interpolada utilizando amostras inteiras da mesma linha ou da mesma coluna, como mostra a Fig. 4.6(a). Note que, para calcular a amostra j , deslocada em $\frac{1}{2}$ amostra tanto na horizontal quanto na vertical, é irrelevante utilizar amostras da linha (cc, dd, h, m, ee, ff) ou da coluna (aa, bb, g, s, gg, hh), desde que

se utilizem os valores antes de arredondá-los. Para interpolar os valores nas posições de um quarto de amostra, utiliza-se um filtro mais simples ($\frac{1}{2}, \frac{1}{2}$) entre as amostras inteiras e meia-amostras, como indicado na Fig. 4.6(b). Essa última interpolação tem um leve efeito passa-baixas, o que muitas vezes também auxilia a predição.

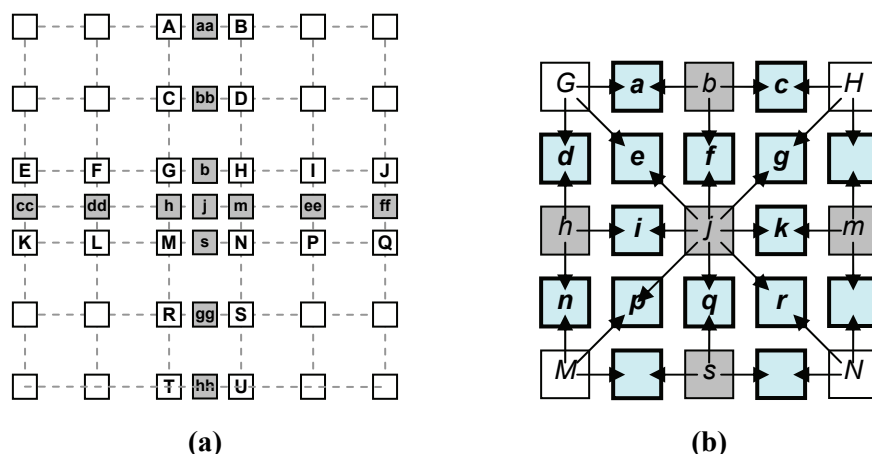


Fig. 4.6 – Ilustração das amostras utilizadas nas interpolações das amostras em posições não-inteiras. (a) (Quadrados cinza) interpolação em meias-amostras utilizando filtro FIR de tamanho 6. (b) Interpolação em um quarto de amostra com um filtro FIR ($\frac{1}{2}, \frac{1}{2}$) para as amostras indicadas em negrito (a, c, d, e, f, g, i, k, n, p, q e r).

4.4.2. Predição de Vetor de Movimento

Na codificação *inter*, os vetores de movimento podem ser responsáveis por uma parcela significativa da informação, principalmente em codificações com taxas de bits mais baixas. Com a introdução de VMs com resolução de $\frac{1}{4}$ de amostra, os valores de tais vetores são multiplicados por 4 antes de serem codificados – um VM com valor $(-1\frac{1}{4}, 2)$ será codificado como $(-5, 8)$ – e isso pode aumentar consideravelmente a taxa de bits, pois a codificação utilizada para os VMs é a $se(v)$, a Exp-Golomb com sinal (ver Seção 4.3.2). Então, para o VM do exemplo, -5 é mapeado em $se(v)$ da mesma forma que 10, em $ue(v)$, com o código **0001011**, enquanto o 8 é mapeado em $se(v)$ da mesma forma que 15 em $ue(v)$, com o código **000010000**. Note que um simples VM com menos de três amostras de distância já usou 16 bits!

Por exemplo, se o VM for $(-27, 30\frac{1}{2})$, algo que não é difícil de acontecer em seqüências com movimento amplo, ele será codificado despendendo 30 bits! Logo, multiplicando-se essa quantidade de bits por cada **partição** que exige um VM,

inviabiliza-se a codificação utilizando predição *inter*. Felizmente, o padrão H.264 possui um processo de predição do próprio **vetor de movimento**, inferindo-o a partir dos VMs vizinhos já decodificados, que reduz substancialmente a quantidade de bits necessária para codificar os VMs. A predição funciona da seguinte forma:

Para os MBs particionados em 8×8 (i.e., com `mb_type P_8×8`) ou para os MBs com uma única partição 16×16, independentemente da forma com que os vizinhos estão particionados, as subpartições adjacentes são determinadas a partir do seguinte esquema.

- 1) Localizam-se os vizinhos **A** (esquerda), **B** (acima), **C** (acima e à direita) e **D** (acima e à esquerda) da **subpartição** atual, através do seguinte procedimento:
 - a. Encontra-se a primeira amostra na parte **superior-esquerda** da subpartição **corrente** (a subpartição atual denomina-se **E**).
 - b. A subpartição que contém a amostra imediatamente a esquerda dessa primeira amostra é a subpartição **A**.
 - c. A subpartição que contém a amostra imediatamente acima dessa primeira amostra é a subpartição **B**.
 - d. Localiza-se a última amostra na parte **superior-direita** da subpartição **corrente**. A subpartição que contém a amostra imediatamente acima e a direita dessa última amostra é a subpartição **C**.

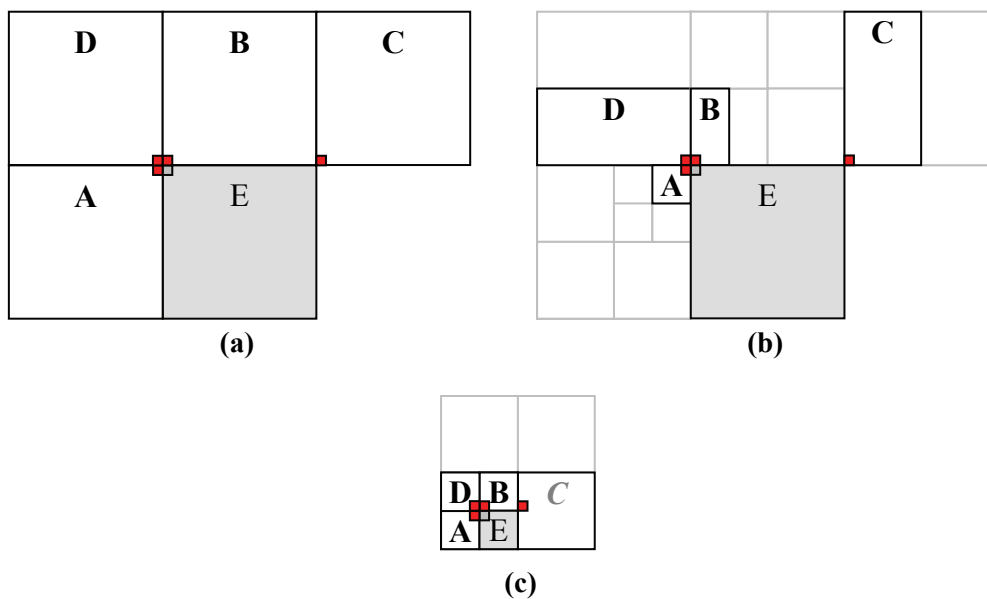


Fig. 4.7 – Exemplos de resultados do processo de localização da vizinhança de uma subpartição para a predição de VM. (a) Com todos MBs sem partição (16x16). (b) Com MBs vizinhos particionados e subparticionados. (c) Subpartições vizinhas de uma subpartição no mesmo MB.

- e. Se a subpartição **C** não estiver disponível (pelo fato de o MB estar na borda da fatia ou da subpartição **C** ainda não ter sido decodificada), utiliza-se a subpartição **D** em seu lugar. Dada a primeira amostra superior-esquerda da subpartição atual, a subpartição **D** é aquela que contém a amostra imediatamente acima e a esquerda dessa amostra.

A Fig. 4.7 mostra alguns exemplos do resultado deste processo. Note que nem todos os vizinhos podem estar disponíveis (por exemplo, se estiverem além da margem ou em outra fatia). No caso (c) desta figura, o vizinho **C** é considerado não-disponível, pois esta partição ainda não foi decodificada.

O VM da subpartição corrente é então predito a partir da **mediana** dos VMs das subpartições **A**, **B**, e **C**, realizada independentemente para os componentes X e Y dos VMs.

Para o caso do MB corrente estar particionado em 16x8 ou em 8x16, não se utiliza a mediana. Nesse caso, a predição é direta – se o MB estiver particionado em 16x8, o VM da partição superior é predito a partir do VM da subpartição **B**, e o da partição inferior a partir do VM da subpartição **A**. Se o MB estiver particionado em 8x16, o VM da partição da esquerda é predito a partir do vizinho **A** e o da direita, a partir do vizinho **C**.

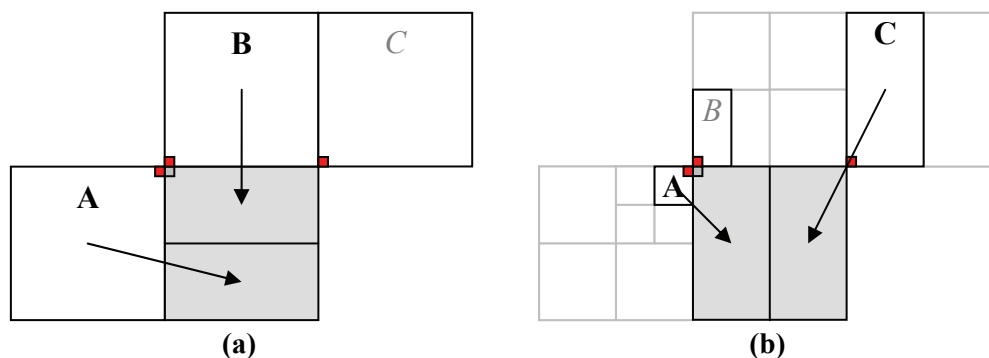


Fig. 4.8 – Exemplos da predição do vetor de movimento para outros particionamentos do MB. (a) Particionamento em 16x8. (b) Particionamento em 8x16.

Ainda há variações neste processo de predição para o caso de uma ou mais das subpartições vizinhas não estarem presentes e também para o caso de possuírem quadros de referência diferentes. Em geral, quando uma subpartição vizinha não está disponível, considera-se seu VM como (0, 0). Quando apenas uma das subpartições vizinhas possuir a mesma referência temporal do que a subpartição atual, considera-se o VM desta subpartição vizinha como o VM predito (sem empregar a mediana).

O processo completo está descrito na Seção 8.4.1.3 do padrão H.264, e o processo de predição do vetor de movimento para os blocos *chroma*, está descrito na Seção 8.4.1.4 [3].

4.4.3. Predição *Intra*

Além da predição *inter*, que utiliza amostras de outros quadros de referência, também pode ser utilizada a predição *intra*, que utiliza as amostras de macroblocos vizinhos previamente decodificados (do próprio quadro atual).

Existem três tipos de predição *intra* possíveis, no que se refere ao tamanho do bloco: a predição *intra* 4x4, que é a mais comum, a predição *intra* 16x16, que age sobre o MB inteiro, e a predição *intra* 8x8, que foi introduzida com as extensões *FRExt*.

Qualquer que seja a predição *intra*, as amostras utilizadas não devem ter sido modificadas pelo filtro antiblocagem.

A Fig. 4.9 mostra quais amostras são utilizadas e de que forma, em oito dos nove tipos de predição *intra* 4×4. Para suavizar as amostras utilizadas na extrapolação, os modos de predição do 3 ao 8 utilizam também filtros FIR com pesos ($\frac{1}{4}, \frac{1}{2}, \frac{1}{4}$) para as amostras inteiras, e com pesos ($\frac{1}{2}, \frac{1}{2}$) para interpolar os pontos entre duas amostras da borda.

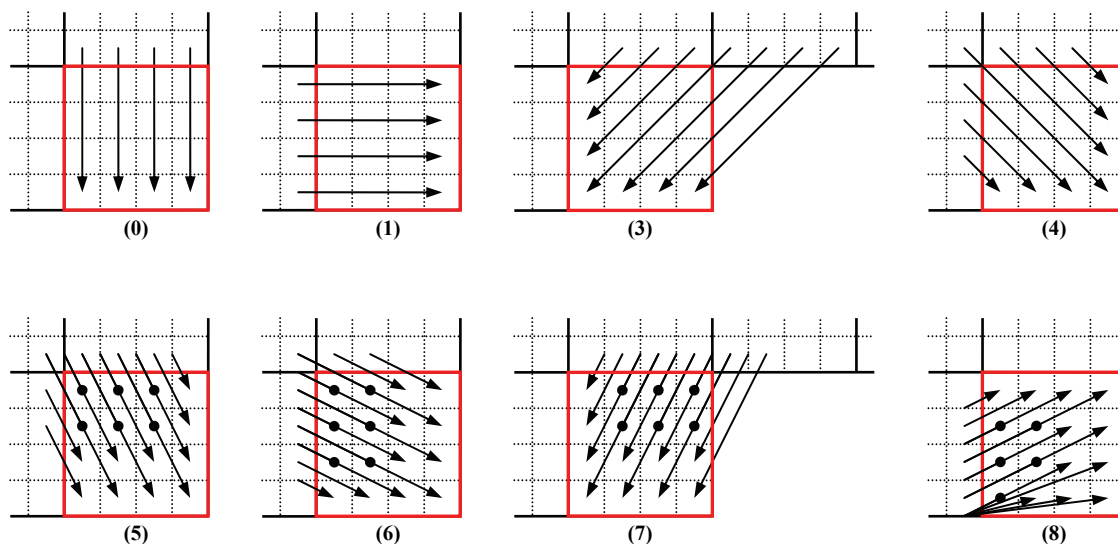


Fig. 4.9 – Oito dos nove tipos de predição Intra 4×4; as descrições estão na Tabela 4.3. Há também o modo (2), que é a predição DC, uma média de todas as amostras vizinhas.

Tabela 4.3 – Modos de Predição Intra

Modo	Valor
Vertical	0
Horizontal	1
DC	2
Diagonal Inferior Esquerda	3
Diagonal Inferior Direita	4
Vertical à Direita	5
Horizontal Abaixo	6
Vertical à Esquerda	7
Horizontal Acima	8

Para codificar o modo de predição utilizado (dentre os 9 possíveis), se fosse utilizado apenas um código de tamanho fixo, utilizaria-se 4 bits por bloco 4×4. Como há 16 blocos 4×4 no MB, seriam utilizados 64 bits para codificar os modos de predição intra. Para diminuir essa quantidade de bits, existe também uma predição do próprio modo de predição intra, que é descrita a seguir:

- Utilizam-se para este processo os modos de predição *intra* empregados nos vizinhos A (bloco 4×4 imediatamente à esquerda ao atual) e B (bloco 4×4 imediatamente acima ao atual).
- Caso um destes vizinhos não esteja disponível, o modo predito é o modo 2 (predição DC).
- Caso um dos vizinhos esteja disponível, porém utilizando a predição *inter* no lugar da *intra*, e caso o parâmetro `constrained_intra_prediction_flag` esteja ativado, o modo predito também é o modo 2 (predição DC).
- Caso contrário, o modo predito é o modo de **menor número** entre o modo utilizado no vizinho A e o modo utilizado no vizinho B. Se algum desses vizinhos não for *intra* 4×4 nem *intra* 8×8, considera-se o modo de predição *intra* desse vizinho como o modo 2 (predição DC).

Se o modo de predição *intra* do bloco atual coincidir com o modo predito pelo processo descrito anteriormente, ele é codificado com apenas 1 bit, caso contrário, ele é codificado com 4 bits (bit 0 mais um código de tamanho fixo de 3 bits, representando um dos outros 8 modos possíveis).

4.5. Codificação do Erro Residual

Realizadas as predições, o que é realmente codificado é o “erro residual”, que é a diferença entre as amostras atuais e as amostras preditas. A codificação do erro residual em um MB no padrão H.264 é feita da seguinte forma:

- Cada um dos 16 blocos 4×4 *luma* (e 4 blocos 4×4 *croma*) são transformados por uma transformada inteira baseada no DCT.
- Caso em que o MB seja do tipo `Intra_16×16`, o bloco 4×4 de coeficientes DC *luma* considera a transformada de Hadamard 4×4 (“transformada da transformada”).
- No bloco 2×2 de coeficientes DC *croma* é utilizada a transformada de Hadamard 2×2.

Após a realização da transformada inteira baseada em DCT, nos blocos 4×4, os coeficientes são escalados e quantizados. A quantização dos blocos *luma* é controlada pelo codificador através do parâmetro QP_Y , que controla a quantização **de todos os blocos** do MB.

4.6. Otimização Taxa-Distorção

Apesar de o processo de otimização taxa-distorção (OTD) não estar diretamente especificado no padrão, tal processo é essencial para que o codificador possa obter uma adequada relação entre a qualidade do vídeo resultante e a taxa de compressão. O padrão deixa o codificador livre para negociar os parâmetros de codificação, tais como modos de MB, vetores de movimento, modos de predição *intra* e parâmetros de quantização. Cada possível conjunto desses parâmetros leva a um código com uma determinada taxa de bits e distorção.

Há uma relação intrínseca entre estas medidas – quanto menor a taxa de bits, maior será a distorção, e vice-versa. Se o codificador buscar minimizar apenas a taxa de bits, resultará em um vídeo muito degradado (com muita distorção); se ele buscar minimizar o erro, resultará em um código com uma alta taxa de bits.

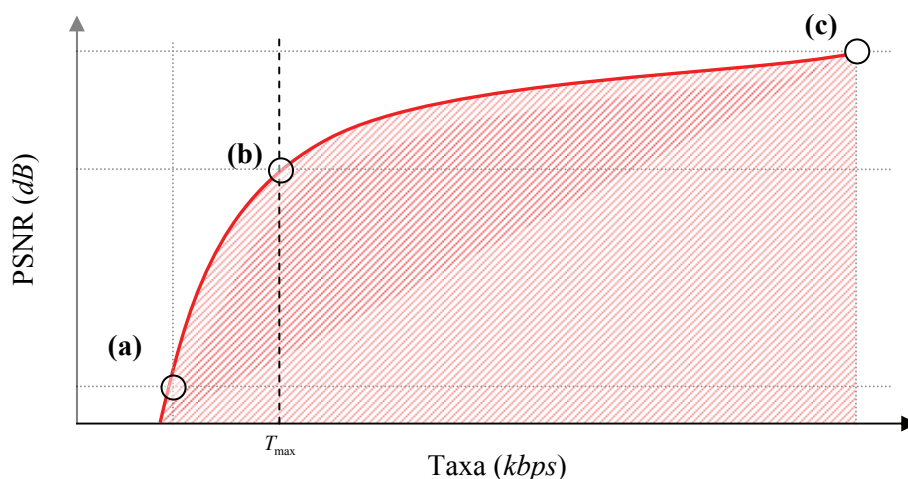


Fig. 4.10 – Exemplo de uma curva taxa-distorção com pontos em diferentes estratégias de otimização:
(a) Minimização da taxa de bits. (b) Minimização da distorção, sujeita a uma taxa máxima.
(c) Minimização da distorção.

Nota-se na Fig. 4.10 que, do ponto (b) ao ponto (a), se ganha pouco em taxa para perder muito em qualidade, enquanto do ponto (b) ao ponto (c), se ganha pouco em qualidade para perder muito em taxa. O objetivo da OTD é encontrar esse ponto (b), ou seja, o melhor compromisso dada a restrição desejada de taxa [31].

Apesar dos parâmetros escolhidos para um MB afetarem também os próximos MBs (e os próximos quadros), supõe-se uma relativa independência entre esses para tornar prático o processo de OTD. Já que as medidas de distorção e de taxa são aditivas, a OTD pode ser realizada isoladamente para cada MB, sendo realizada uma busca dentre todos os parâmetros possíveis, de forma a determinar o conjunto de parâmetros que resulte na melhor taxa-distorção para o MB.

A técnica de otimização mais utilizada é a otimização Lagrangiana (descrita em [32]). Especificamente, dados:

- Um conjunto de amostras \mathbf{A} .
- Vários conjuntos de parâmetros de codificação P_i , que codificam um conjunto de amostras \mathbf{A} em um código $C_{\mathbf{A},P_i}$ - este código, ao ser decodificado, resulta nas amostras reconstituídas \mathbf{A}' .
- Uma função de taxa $T(\mathbf{A}, P_i)$ que especifica a quantidade de bits do código $C_{\mathbf{A},P_i}$.
- Uma medida de **distorção** $D(\mathbf{A}, P_i)$, que representa a diferença (erro) entre as amostras originais \mathbf{A} e as amostras reconstituídas \mathbf{A}' .
- Uma taxa máxima desejada T_{\max} .

A otimização Lagrangiana consiste em **minimizar** a função distorção $D(\mathbf{A}, P_i)$, variando-se os parâmetros P_i e sujeita à restrição de taxa $T(\mathbf{A}, P_i) \leq T_{\max}$. Podemos transformar tal problema restrito em um problema irrestrito, ou seja, minimizar uma função $J(\mathbf{A}, P_i)$ onde:

$$J(\mathbf{A}, P_i) = D(\mathbf{A}, P_i) + \lambda \cdot T(\mathbf{A}, P_i) \quad (4.5)$$

O parâmetro Lagrangiano λ (que deve ser maior ou igual a zero) tem uma relação implícita com a taxa máxima T_{\max} - quando λ é próximo de zero, a minimização da distorção tem prioridade, portanto, o objetivo fica próximo do ponto (a) na Fig. 4.10; quando λ é muito grande, é a minimização da taxa que tem prioridade, ficando o objetivo próximo do ponto (c).

Supondo uma função $D(T)$ como na Fig. 4.10, representando a **menor** distorção possível para uma determinada taxa T , a função a ser minimizada resulta em

$$J(T) = D(T) + \lambda \cdot T \quad (4.6)$$

Em seu ponto mínimo, que queremos que seja na taxa T_{\max} , sua derivada deve ser zero, portanto

$$\frac{dJ(T_{\max})}{dT} = \frac{dD(T_{\max})}{dT} + \lambda = 0 \quad (4.7)$$

logo

$$\lambda = -\frac{dD(T_{\max})}{dT} \quad (4.8)$$

Mais precisamente, a taxa T_{\max} é aquela em que o parâmetro λ é o **custo** de cada bit em termos de distorção - i.e., cada bit a menos na taxa T deve aumentar a medida de distorção D em λ .

Como o principal parâmetro que afeta a curva taxa-distorção é o parâmetro de quantização (QP), utiliza-se um parâmetro λ dependente do QP. Foi determinado experimentalmente [33] que, quando utiliza-se a medida de distorção SSD (soma das diferenças quadráticas), a seguinte relação produz valores satisfatórios de λ para fatias P do H.264:

$$\lambda = 0,85 \cdot 2^{(QP-12)/3} \quad (4.9)$$

Já para processos que empregam a medida de distorção SAD (como o processo de estimação de movimento) utiliza-se $\lambda_m = \sqrt{\lambda}$.

Por exemplo, para um $QP = 22$ (alta qualidade/alta taxa), temos um $\lambda = 8,5675$. Isto significa que, a cada bit a menos, teremos uma variação de 8,5675 no SSD (que, por exemplo, pode significar aproximadamente uma diferença de um nos valores de 8 amostras, ou uma diferença de 3 no valor de uma única amostra – lembrando que a medida de distorção é quadrática). Já para um $QP = 48$ (baixa qualidade/baixa taxa), temos um $\lambda \cong 3481,6$, ou seja, o bit é muito mais caro – um bit a menos no MB poderia significar uma diferença de 3 ou 4 em todas as amostras de um MB, ou uma diferença de 59 concentrada em uma única amostra.

Determinar os parâmetros que minimizam a taxa-distorção em um MB pode ser uma tarefa computacionalmente custosa, pois envolve uma busca de todos os modos de codificação de um MB, de todos os modos de predição *intra* e de predição *inter* (estimação de movimento). Isso envolve codificar o MB várias vezes, uma para cada conjunto de parâmetros, para que sua taxa e distorção seja computada. Esse custo computacional pode ser reduzido utilizando-se algumas regras de parada que eliminam buscas “infrutíferas”, como descrito em [34].

4.7. Estimação de Movimento

Na OTD do codificador de referência do H.264, o processo de estimação de movimento (EM) é um pouco diferente por motivos de complexidade computacional. Como o espaço de busca pode ser muito grande (muitos vetores a se procurar), não se faz a codificação por completo do MB – utiliza-se como taxa apenas os bits necessários para codificar os vetores (aqui entra a predição de VM) e como medida de distorção a SAD ou a SATD. Então, é definida uma janela de busca onde se realizará a estimação de movimento – M amostras em cada direção a partir do VM predito, e em N quadros de referência. Primeiro busca-se dentre todos os VM inteiros possíveis nessa janela aquele que minimiza a medida de distorção SAD (a denominada **busca completa**); em seguida, busca-se (apenas na vizinhança deste vetor) um vetor com precisão de $\frac{1}{4}$ de amostra, utilizando a SATD como medida de distorção. A SATD é mais “realista” do que a SAD, pois realiza uma transformada Hadamard nas diferenças antes de somar os valores absolutos e, ao

mesmo tempo, possui um custo computacional mais baixo em relação ao processo completo de transformação DCT/quantização/quantização inversa/transformação inversa.

A complexidade computacional desta estimação de movimento pode ser reduzida se em vez da **busca completa**, for adotado um procedimento de busca não-exaustiva que restrinja o número de VMs a serem procurados. Existem vários algoritmos propostos na literatura com tal objetivo e se diferenciam em função da redução de complexidade computacional obtida e da diminuição de desempenho taxa-distorção que exibem (o que é indesejado).

1ª PROPOSTA DE MODIFICAÇÃO: CODIFICAÇÃO COM MB DUPLO-INTER

5.1. Introdução

A codificação de MBs que apresentem erro residual baixo em quase toda a sua extensão, mas que possuem uma pequena região com alto erro, implica na necessidade de se utilizar muitos bits para a codificação desse erro (Fig. 5.1). Focando na codificação diferenciada dessa região, apresentamos duas propostas que visam reduzir a taxa de bits. O modelo de codificação proposto consiste em modificações do padrão H.264 que, de acordo com os testes de validação, fornecem um ganho de até 0,2 dB.

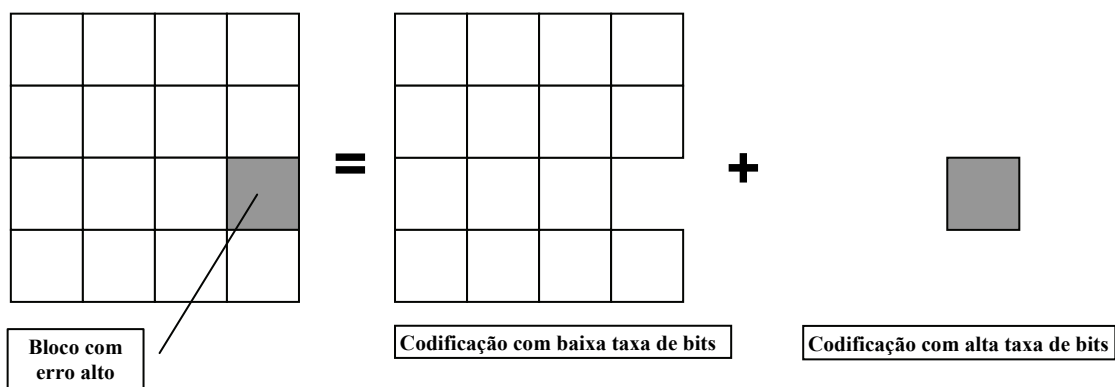


Fig. 5.1 – Exemplo da codificação “desacoplada” de bloco 4x4 com erro residual maior que o do restante do MB.

As duas propostas de codificação consideram as seguintes situações distintas:

- Codificação do MB através de 16 blocos 4x4 usando predição do tipo *inter*, realizando a CM para a maioria desses blocos com um único VM e utilizando um VM distinto para alguns blocos restantes.

- Codificação do MB através de 4 blocos 8×8, codificando alguns dos blocos 8×8 no modo *intra* e outros no modo *inter*.

As situações nas quais tais abordagens apresentam maior eficiência são aquelas em que o erro residual de um bloco pode ser reduzido utilizando uma outra forma de predição *inter*, ou quando não houver qualquer predição *inter* que reduza o erro, caso em que a predição *intra* é usada.

Neste capítulo estudaremos a primeira situação, que ocorre com maior frequência, por exemplo, quando um pequeno objeto em movimento (ou sua borda) adentra em um MB, que, por sua vez, está representando um outro objeto de tamanho maior com movimento em outra direção. Nessa situação o codificador H.264 padrão normalmente realiza a codificação do MB através de uma das seguintes alternativas:

- i) Uso de apenas um VM, resultando em um alto erro residual no bloco 4×4 que representa o novo objeto, uma vez que a predição *inter* usada não é eficiente para esse bloco. O custo de codificação para este bloco dependerá dos coeficientes do bloco 4×4 e do parâmetro de quantização usado. Este esquema de codificação é ilustrado na Fig. 5.2(a).
- ii) Particionamento do MB em blocos 8×8 e subdivisão de uma dessas partições em blocos 4×4, sendo o bloco correspondente ao novo objeto codificado com um VM diferente dos outros. Neste caso reduz-se o erro residual, mas aumenta a quantidade de bits necessária para a codificação dos VMs. Esta codificação é mostrada na Fig. 5.2(b).

Para melhorar a relação taxa-distorção nesta situação, propõe-se introduzir um novo modo de codificação dos MBs, mostrado na Fig. 5.2(c), no qual o MB é subdividido em duas partições com VMs distintos.

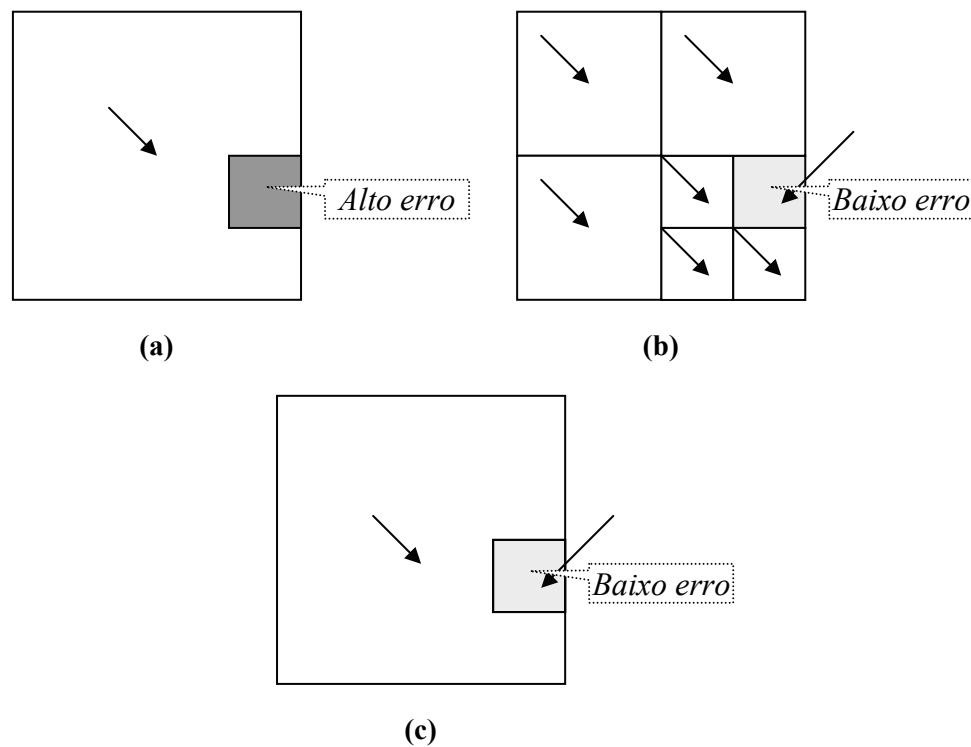


Fig. 5.2 – Modos de particionamento do MB usados no padrão H.264 comparados com o modo proposto. (a) Modo *inter* utilizando apenas um VM, do padrão H.264. (b) Modo *inter* utilizando partições do padrão H.264. (c) Modo *duplo-inter* proposto.

5.2. Codificação

No modo *duplo-inter* proposto são necessárias, além das informações que já são codificadas em um MB (modo, quadros de referência, VMs, etc.), as seguintes informações adicionais:

- Posição do bloco com VM distinto.
- Tamanho do bloco com VM distinto.

Para minimizar estas informações, permitimos que apenas um bloco 4×4 com VM distinto seja usado por MB, sendo sua posição alinhada em múltiplos de 4 amostras, resultando em um acréscimo mínimo de 4 bits para a transmissão da posição do bloco. Assim, teremos o MB particionado da seguinte forma: uma **partição menor** com um bloco de 4×4 amostras, e uma **partição maior**, com um bloco de 16×16 amostras, excluídas as amostras da **partição menor**, conforme mostra a Fig. 5.3.

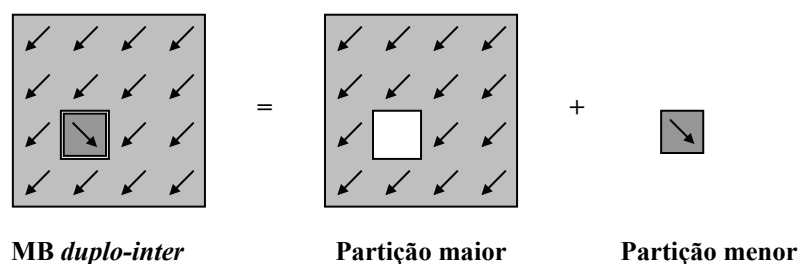


Fig. 5.3 – Particionamento do MB *duplo-inter*.

5.2.1. Mudanças nos Elementos de Sintaxe do MB

Cada parâmetro de codificação representado no fluxo de bits é denominado “elemento de sintaxe”. No padrão H.264 o elemento de sintaxe que codifica o tipo de MB é denominado `mb_type`, sendo sua semântica dependente do tipo de fatia (I, P, B, SI e SP) em que o MB está inserido. Os códigos usados pelo padrão H.264 para as fatias do tipo P são mostrados na Tabela 5.1 na coluna “Padrão”.

Para indicar o tipo de MB *duplo-inter* é necessário especificar um novo valor para o elemento `mb_type`, que será denominado `P_MIX`. Para avaliar a influência da quantidade de bits usada no código, realizamos testes com dois possíveis valores para esse elemento de sintaxe.

No primeiro teste é adotado o valor 2 (código 011); dessa forma todos os valores superiores a 2 são “deslocados” para um valor acima, como mostra a coluna “Modo *duplo-inter*₁” da Tabela 5.1. Devido a essa mudança, alguns códigos ficarão mais longos. No segundo teste, foi utilizado o valor 6 para o modo `P_MIX`, ficando este com 5 bits, como mostrado na coluna “Modo *duplo-inter*₂”.

Para o primeiro valor adotado, o único tipo de MB com uma mudança importante, em termos de taxa de bits, é o *intra* 8×16, pois seu código passa de 3 para 5 bits (de 011 para 00100); alguns tipos Intra 16×16 e o tipo PCM também possam a ter 2 bits a mais na codificação do `mb_type`, mas nesses casos a diferença é praticamente desprezível, pois esses modos raramente são usados, além de já serem codificados com uma grande quantidade de bits. Para o segundo valor adotado, a diferença é que o próprio modo *duplo-*

inter passa a ter 2 bits a mais em sua codificação, enquanto a codificação do *mb_type* do modo *intra* 8×16 continua com apenas 3 bits.

Tabela 5.1 – Valores e códigos para os tipos de MB em fatias P

Nome do <i>mb_type</i>	Padrão		Modo <i>duplo-inter</i> ₁		Modo <i>duplo-inter</i> ₂	
	Valor	Código	Valor	Código	Valor	Código
P_L0_16×16	0	1	0	1	0	1
P_L0_L0_16×8	1	010	1	010	1	010
P_MIX ₁			2	011		
P_L0_L0_8×16	2	011	3	00100	2	011
P_8×8	3	00100	4	00101	3	00100
P_8×8ref0	4	00101	5	00110	4	00101
I_4×4	5	00110	6	00111	5	00110
P_MIX ₂					6	00111
I_16×16_0_0_0	6	00111	7	0001000	7	0001000
I_16×16_0_0_1	7	0001000	8	0001001	8	0001001
...
I_16×16_3_2_1	29	000011110	30	000011111	30	000011111
I_PCM	30	000011111	31	00000100000	31	00000100000

No modo *duplo-inter* proposto, além do *mb_type*, é necessário codificar a posição da **partição menor**. Para indicar esta posição é utilizado um código de tamanho fixo de 4 bits (denominado *mix_block*), como mostra a Fig. 5.4.

0000	0001	0010	0011
(0)	(1)	(2)	(3)
0100	0101	0110	0111
(4)	(5)	(6)	(7)
1000	1001	1010	1011
(8)	(9)	(10)	(11)
1100	1101	1110	1111
(12)	(13)	(14)	(15)

Fig. 5.4 – Códigos e valores para o parâmetro *mix_block* (posição da partição menor).

5.2.2. Estudo de Caso

Para ilustrar a economia de bits que pode resultar do modo de codificação proposto, avaliamos o caso de um MB no qual todos os vetores estão na mesma direção (inclusive dos MBs vizinhos), com exceção de um vetor de um bloco 4×4, conforme mostrado na Fig. 5.2. Cada VM é predito a partir dos vetores adjacentes já codificados. A diferença (MVD – *Motion Vector Difference*) entre o vetor predito e o vetor atual é codificada. Mesmo que o vetor atual seja idêntico ao vetor predito, são necessários 2 bits para codificar o valor da diferença (0,0).

A Tabela 5.2 mostra a diferença de bits para a codificação de um mesmo MB no modo previsto pelo H.264 (supondo a partição em 8×8) e usando modo *duplo-inter* proposto com P_MIX de 3 bits. No exemplo todos os coeficientes do erro residual do MB são nulos, o que ocorre com muita frequência na codificação com taxas baixas.

Tabela 5.2 – Comparação da codificação de um MB pelo padrão H.264 com modo *duplo-inter* proposto

Codificação Normal			Codificação com MB <i>duplo-inter</i>		
Descrição	Valor	Código	Descrição	Valor	Código
<i>Tipo do MB</i>			<i>Tipo do MB</i>		
mb_type	3 (P_8x8)	00100	mb_type_new	2 (P_MIX)	011
<i>Tipo dos SubMBs (particionamento)</i>			<i>Posição da partição menor (VM diferenciado)</i>		
sub_mb_type [0]	0 (P_L0_8x8)	1	mix_block	11 (3, 2)	1011
sub_mb_type [1]	0 (P_L0_8x8)	1	<i>Referências temporais das partições</i>		
sub_mb_type [2]	0 (P_L0_8x8)	1	ref_idx_L0 [0]	1	010
sub_mb_type [3]	3 (P_L0_4x4)	00100	ref_idx_L0 [1]	0	1
<i>Referências temporais dos SubMBs</i>			<i>VMs</i>		
ref_idx_L0 [0]	1	010	mvd_L0 [0][0][X]	0	1
ref_idx_L0 [1]	1	010	mvd_L0 [0][0][Y]	0	1
ref_idx_L0 [2]	1	010	mvd_L0 [1][0][X]	11	000010110
ref_idx_L0 [3]	0	1	mvd_L0 [1][0][Y]	19	00000100110
<i>VMs</i>			<i>Coefficientes</i>		
mvd_L0 [0][0][X]	0	1	CBP	0	1
mvd_L0 [0][0][Y]	0	1			
mvd_L0 [1][0][X]	0	1			
mvd_L0 [1][0][Y]	0	1			
mvd_L0 [2][0][X]	0	1			
mvd_L0 [2][0][Y]	0	1			
mvd_L0 [3][0][X]	0	1			

mvd_L0 [3][0][Y]	0	1			
mvd_L0 [3][1][X]	11	000010110			
mvd_L0 [3][1][Y]	19	00000100110			
mvd_L0 [3][2][X]	0	1			
mvd_L0 [3][2][Y]	0	1			
mvd_L0 [3][3][X]	0	1			
mvd_L0 [3][3][Y]	0	1			
<i>Coeficientes</i>					
CBP	0	1			
Total de bits:		56 bits			34 bits

No exemplo considerado a codificação do MB no modo *duplo-inter* resultou na economia de **22 bits**, conseguidos com um menor número de VMs, quadros de referência e subdivisões das partições. Na Tabela 5.2 verifica-se que foram economizados 10 bits nos VMs (*mvd_L0*), 6 bits nos quadros de referência (*ref_idx_L0*) e 8 bits nas subdivisões das partições (*sub_mb_type*), enquanto que se gastou 2 bits a mais na codificação do tipo de MB e posição da partição menor.

5.3. Alterações nos Processos de Estimação de Movimento e Otimização Taxa-Distorção

Para acomodar o novo tipo de MB no processo de EM, são realizadas diversas modificações no codificador. Essas alterações podem ser resumidas em:

- Realização do processo completo de EM para cada possível posição da partição 4×4 (16 vezes), sendo feitas duas buscas: primeiro para a partição maior e em seguida para a partição menor. Essa ordem é necessária, pois o VM da partição menor é predito a partir do VM da partição maior;
- Alteração do cálculo da SAD e da SATD⁴ para ignorar as amostras correspondentes à partição menor no processo de busca do VM da partição maior;
- Adaptação das estruturas necessárias para armazenar temporariamente os parâmetros, VMs e códigos gerados para cada tipo de MB e particionamento no processo de OTD;

⁴ SAD e SATD são medidas de erro. SAD = *Sum of Absolute Differences*, SATD = *Sum of Absolute Hadamard Transformed Differences*.

- Modificação da forma de utilização de referências temporais dentro de um mesmo bloco 8×8, permitindo que blocos 4×4 tenham referências temporais distintas.

5.4. Alteração no Processo de Predição dos VMs

O processo de predição do VM a partir dos VMs vizinhos também é modificado de forma a acomodar o modo de MB *duplo-inter* proposto, sendo essa predição realizada da seguinte forma:

- A predição do VM da partição maior é realizada da mesma forma que a predição utilizada para um MB do tipo *inter* 16×16, utilizando a mediana dos VMs dos blocos A, B e C vizinhos – ver Fig. 4.7(a) e (b). Essa etapa deve anteceder o processo de predição do VM da partição menor;
- Determinam-se os blocos vizinhos A, B e C da partição menor e realiza-se a predição do VM dessa partição utilizando a mediana desses VMs vizinhos.

Se for adotada a forma de predição do VM do padrão H.264 (ver seção 4.4.2), na maioria das vezes (quando pelo menos **dois blocos vizinhos** estiverem na partição maior) o VM predito da partição menor é o próprio VM da partição maior. Isso é ilustrado na Fig. 5.5(b) para as posições 4 à 15.

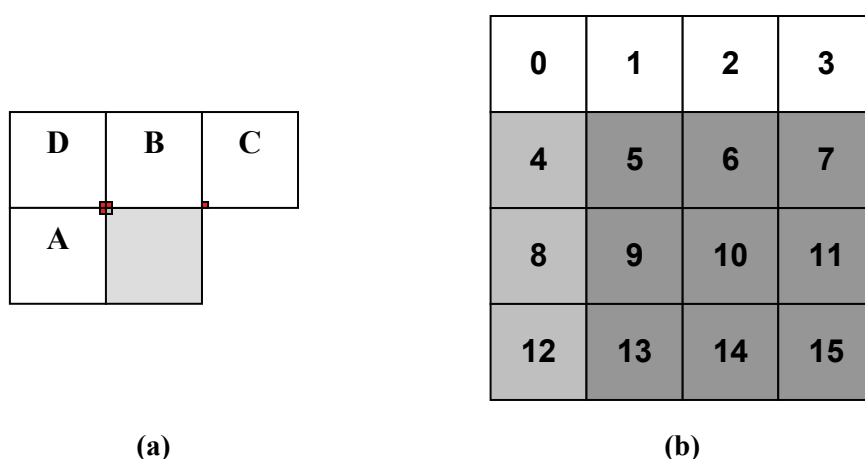


Fig. 5.5 – Influência do VM da partição maior na predição do VM da partição 4×4. (a) Vizinhos A, B, C e D de um bloco 4×4. (b) Quando a posição da partição 4×4 for uma das posições indicadas em cinza, o VM da partição maior é a predição do VM da partição 4×4.

Apenas quando as duas partições do MB *duplo-inter* possuem referências temporais distintas ocorre uma exceção, pois neste caso, as regras de predição mudam, e a partição menor pode ter um VM predito diferente do VM da partição maior se a posição da partição menor for a posição 4, 8 ou 12 [Fig. 5.5(b)].

O fato de o VM predito da partição menor ser na maioria das vezes o próprio VM da partição maior é uma desvantagem, uma vez que este modo de codificação é justamente proposto para situações onde um objeto menor tem movimento diferenciado do restante do MB. Para melhorar a predição para o modo proposto, avaliamos uma predição alternativa para o VM da partição menor, ilustrada na Fig. 5.6. Nessa predição, teremos, por exemplo, as seguintes situações:

- A predição do bloco de posição 2 é o VM do bloco imediatamente acima (B_2).
- Os VMs dos blocos 3, 7 e 11 são todos preditos a partir do VM do bloco C. Faria mais sentido se fossem preditos a partir dos VMs das partições à direita destes blocos, mas estas partições não estão disponíveis no momento da predição.
- Os VMs dos blocos 12, 13 e 14 são preditos a partir do VM do bloco A_3 .
- Os VMs dos blocos 5, 6, 9 e 10 são preditos a partir do VM da partição maior.

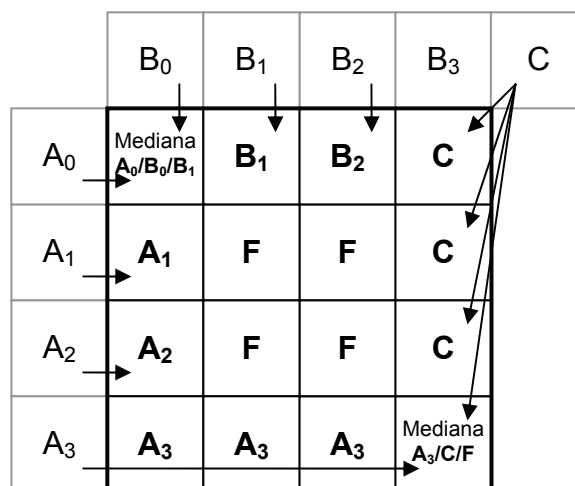


Fig. 5.6 – Predição alternativa dos VMs para a partição menor do MB *duplo-inter*. Dependendo da posição, a predição do VM do bloco é dada pelo VM de um bloco vizinho diferente, ou pela mediana entre 3 blocos vizinhos. O “F” corresponde ao VM da partição maior do MB *duplo-inter* atual.

5.5. Testes de Codificação

Para testar e comparar o modelo de codificação utilizando o modo de MB *duplo-inter*, diversas modificações são realizadas no codificador de referência JM 7.3 [35], incluindo processos relacionados à predição, estimação de movimento, otimização taxa-distorção e codificação de entropia, descritos nas seções 5.3 e 5.4.

5.5.1. Condições de Teste

Nos testes são utilizadas sete seqüências de imagens: *Hall*, *Stefan*, *Tennis*, *Foreman*, *Coastguard*, *Mother & Daughter* e *Mobile & Calendar*, ilustradas na Fig. A.1 (Apêndice A). Para cada seqüência são codificados os 150 quadros iniciais com uma taxa de 15 quadros por segundo (para a seqüência *Tennis*, apenas 75 quadros são codificados). As seqüências *Foreman* e *Coastguard* são codificadas em resolução CIF e QCIF.

O parâmetro de quantização QP é variado entre 18 e 40, para verificar a influência da proposta na qualidade. Em todas as seqüências, apenas o primeiro quadro é codificado como fatia I, sendo os demais codificados como fatias P. No primeiro quadro o QP usado é igual ao dos demais quadros quando QP é menor ou igual a 28, caso contrário o valor do QP é mantido em 28 de modo a não comprometer a qualidade de toda a seqüência (principalmente em seqüências estáticas como a *Hall*, na qual o modo *skip* é muito usado).

No processo de EM, são utilizados 5 quadros de referência temporal e uma janela de busca de tamanho 17×17 em torno do VM predito.

Nos testes cada seqüência foi codificada considerando três cenários diferentes:

- i) Cenário 0 - Codificador padrão.
- ii) Cenário 1 - Codificador modificado, com P_MIX de 3 bits e com a predição de VM **padrão**.
- iii) Cenário 2 - Codificador modificado, com P_MIX de 5 bits e com a predição de VM **alternativa**.

5.5.2. Frequência de Utilização do Modo de MB Proposto

Resultados do teste descrito na seção anterior são mostrados nas Figs. 5.7 e 5.8. Nessas figuras, a coluna da esquerda mostra os histogramas indicando a frequência com que os diferentes tipos de MB são selecionados para cada QP. Essa frequência é indicativa da “utilidade” de cada tipo de MB, pois devido ao processo de OTD o modo escolhido é sempre o que resulta na melhor relação taxa-distorção. Os índices de 0 a 7, utilizados nas Fig. 5.7 a Fig. 5.8 (coluna da esquerda) correspondem aos modos descritos na Tabela 5.3. Os histogramas em azul correspondem aos resultados da codificação padrão H.264, enquanto os histogramas em verde, aos resultados da codificação quando adicionado o modo *duplo-inter*.

Tabela 5.3 – Descrição da numeração de modos utilizada nos gráficos de histograma

Nº. do Modo	Modo
0	<i>skip</i>
1	<i>inter 16×16</i>
2	<i>inter 16×8</i>
3	<i>inter 8×16</i>
4	<i>inter 8×8</i>
5	<i>intra 4×4</i>
6	<i>intra 16×16</i>
7	<i>duplo-inter</i>

Em todas as seqüências de imagens pode-se perceber que o modo *duplo-inter* (índice 7) é utilizado com frequência quando a qualidade da imagem é alta (QP baixo). Em algumas seqüências, esse modo é também muito utilizado mesmo em qualidades mais baixas (QPs mais altos). Em geral o modo proposto é usado em 10% a 30% dos MBs quando a qualidade é maior. Quando a qualidade diminui, o número de blocos codificados no modo *duplo-inter* tende a diminuir, sendo preferidos outros tipos de codificação de MB, tal como o modo *skip* (índice 0).

Nas codificações em que o modo *duplo-inter* é permitido, a frequência de utilização dos outros modos se reduz em comparação com a codificação padrão. Os tipos de MB que tiveram a maior redução são os tipos *inter 16×16*, *8×16* e *8×8*. A diminuição da utilização do *8×16* deve-se, provavelmente, ao maior número de bits usados para codificar seu `mb_type`.

A frequência das posições escolhidas para os blocos de tipo *duplo-inter* é mostrada nos histogramas na coluna da direita da Fig. 5.7. São 16 posições possíveis para a partição menor, numeradas conforme a Fig. 5.4.

a) Cenário 1

Neste cenário, as seqüências de imagens são codificadas utilizando o código P_MIX de 3 bits para indicar o modo *duplo-inter* e a **predição padrão H.264** para a predição do VM da **partição menor**.

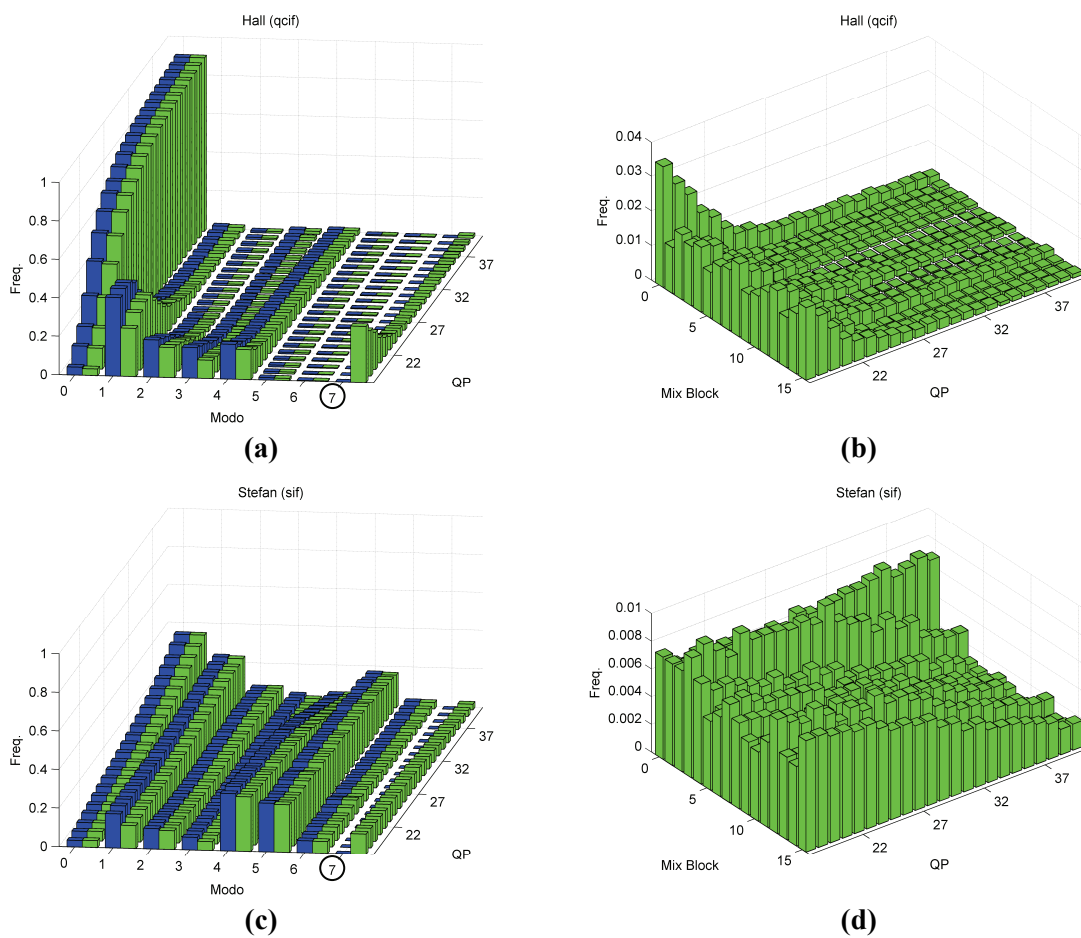
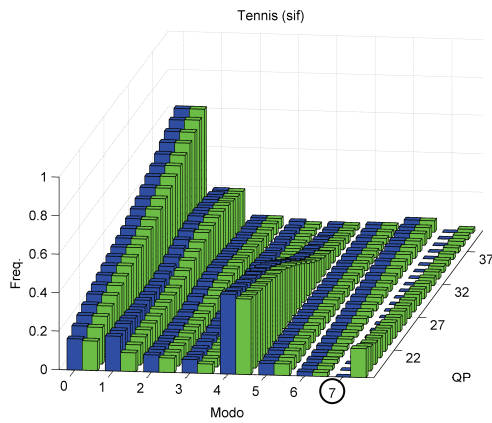
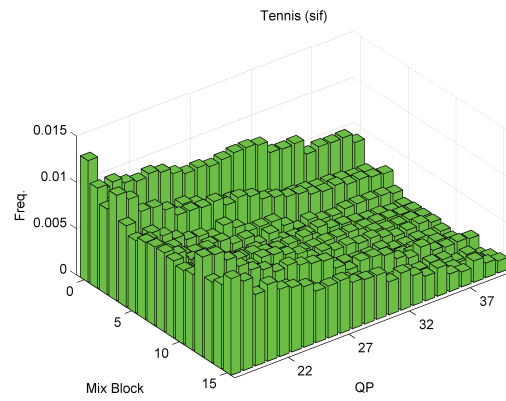


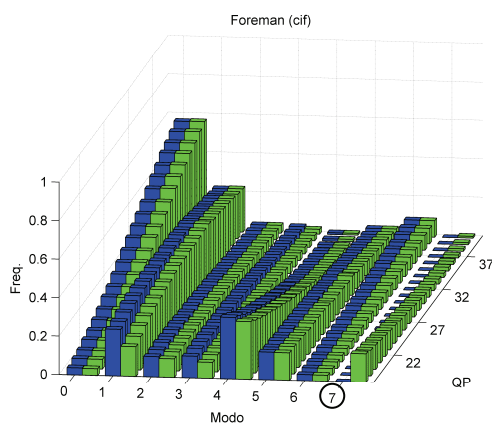
Fig. 5.7 – Coluna da esquerda: histograma dos tipos de MBs selecionados, para cada seqüência (sem modo *duplo-inter* em azul e com modo *duplo-inter* em verde). Coluna da direita: histograma da posição selecionada (*mix_block*).



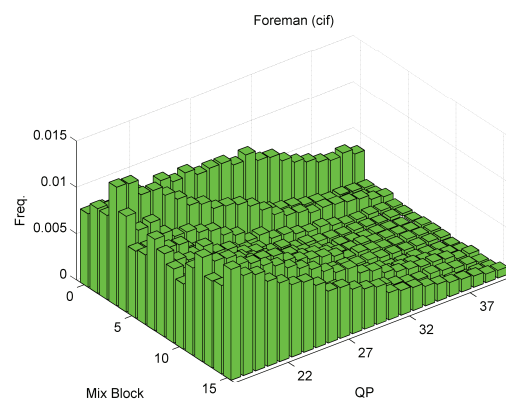
(e)



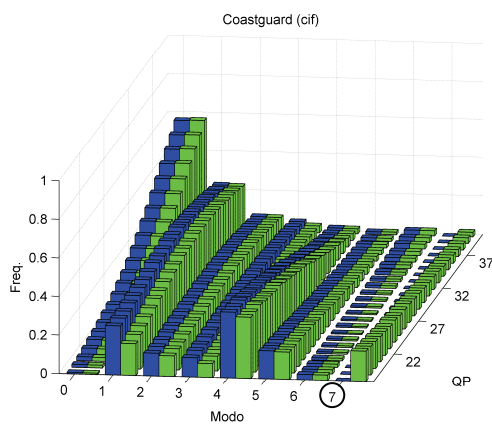
(f)



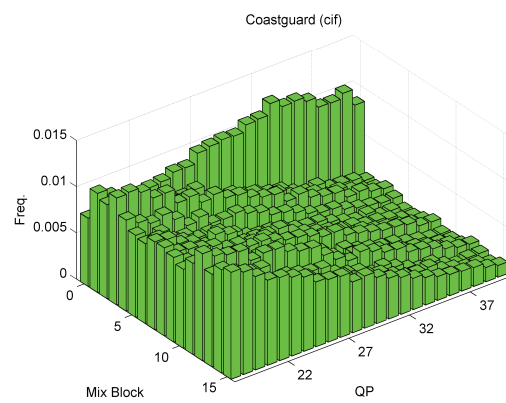
(g)



(h)

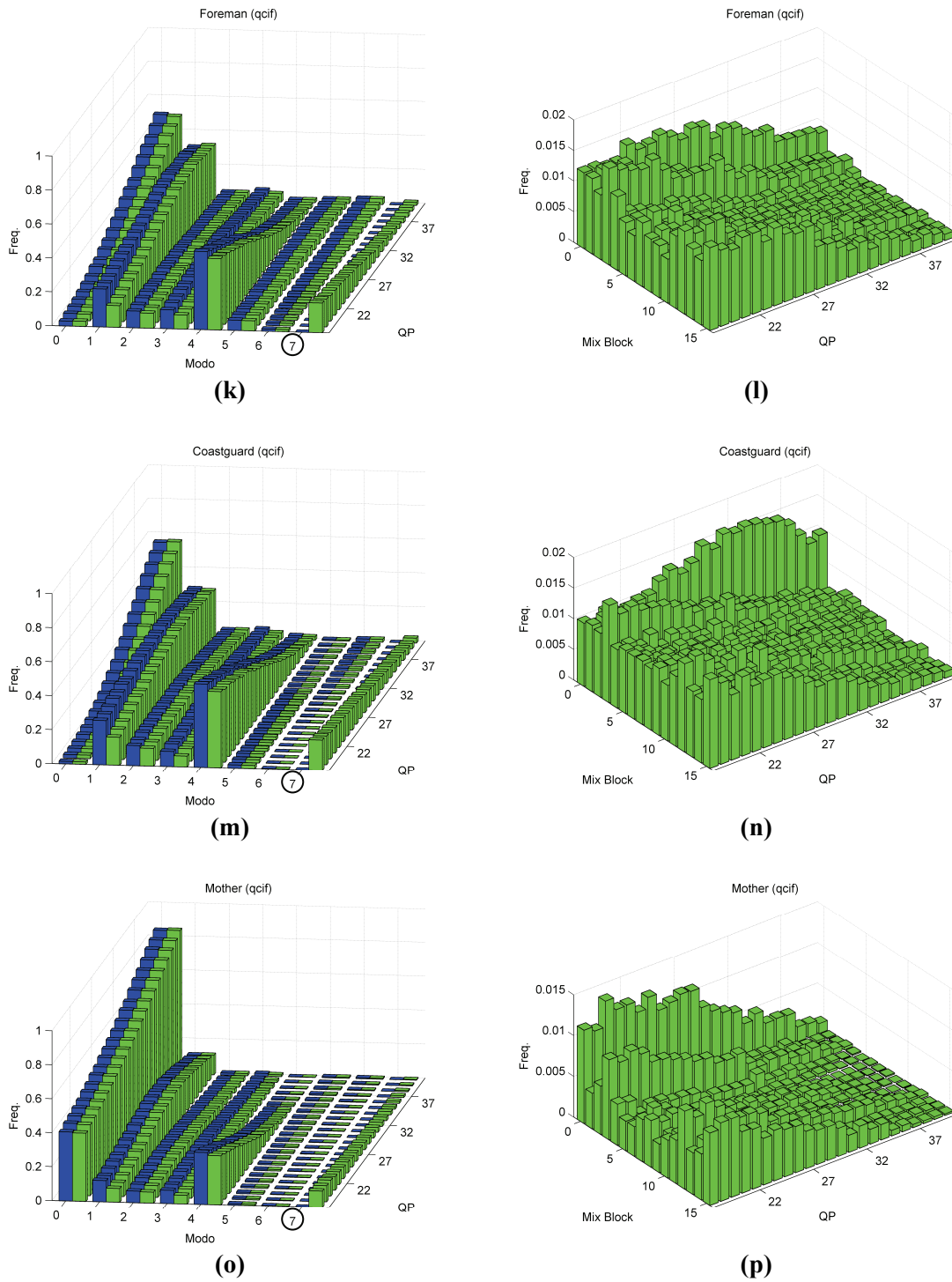


(i)

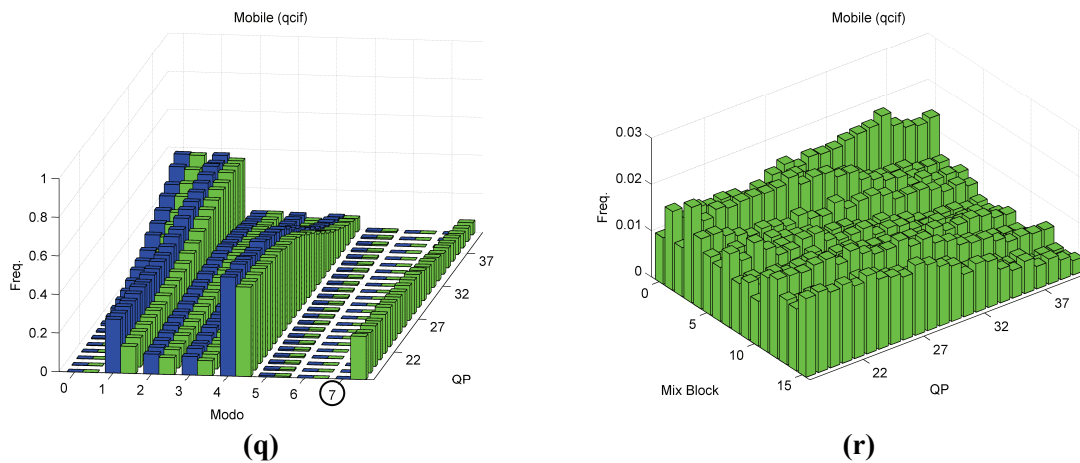


(j)

(continuação) Fig. 5.7 – Coluna da esquerda: histograma dos tipos de MBs selecionados, para cada sequência (sem modo *duplo-inter* em azul e com modo *duplo-inter* em verde). Coluna da direita: histograma da posição selecionada (*mix_block*).



(continuação) Fig. 5.7 – Coluna da esquerda: histograma dos tipos de MBs selecionados, para cada seqüência (sem modo *duplo-inter* em azul e com modo *duplo-inter* em verde). Coluna da direita: histograma da posição selecionada (*mix_block*).



(continuação) Fig. 5.7 – Coluna da esquerda: histograma dos tipos de MBs selecionados, para cada seqüência (sem modo *duplo-inter* em azul e com modo *duplo-inter* em verde). Coluna da direita: histograma da posição selecionada (*mix_block*).

Em relação às diversas seqüências avaliadas, há algumas particularidades nos histogramas que nos permite analisar as situações em que ocorrem os diferentes modos de MB do padrão (além do modo proposto). Na seqüência *Hall*, que é quase estática, o modo *skip* prevalece nos QPs mais altos, praticamente não sendo usados MBs do tipo *intra*. O bloco *duplo-inter* foi utilizado em quase 30% dos blocos usando $QP = 18$, mas com $QP < 25$ é usado em menos de 5% dos MBs.

Na seqüência *Stefan*, que possui muita movimentação da câmera e do objeto frontal (jogador) além de possuir um plano de fundo com uma textura complexa (platéia), ocorre uma maior utilização de modos *intra* e *inter* 8×8 . Para valores de QP maiores, os modos *inter* 8×8 são substituídos pelos modos *inter* 16×16 , que economizam bits por codificarem apenas um VM. O modo *duplo-inter* foi usado em 10% dos MBs nos QPs menores e em 3% nos QPs maiores.

No histograma da seqüência *Tennis* podemos notar uma maior utilização dos blocos *inter* 8×8 com os QPs menores, o que ocorre devido a existência de um afastamento lento da imagem (*zoom out*) e também devido à textura complexa do plano de fundo. No trecho final da seqüência, a câmera está estática, fazendo com que o modo *skip* seja também bastante utilizado, mesmo nos QPs mais baixos.

Apesar das seqüências *Foreman* e *Coastguard* serem bem diferentes, seus histogramas possuem perfis muito parecidos, tanto para os MBs do tipo *duplo-inter* quanto

para os de outros tipos. Nos QPs baixos prevalece o modo *inter* 8×8, sendo que o modo *inter* 16×16 é utilizado praticamente com a mesma frequência em todos os QPs.

A Tabela 5.4 mostra, para diferentes valores de QP, a frequência com que o modo *duplo-inter* foi escolhido. Na seqüência *Mobile*, esse modo foi escolhido com muita frequência, entre 16% e 22% dos MBs (dependendo do QP). Isso mostra que o modo *duplo-inter* é adequado às seqüências mais complexas como esta. A Tabela 5.5 mostra a frequência de utilização do modo *duplo-inter* com relação aos outros modos, para um QP fixo (QP = 27).

Em relação a frequência com que a partição menor ocupa as diversas posições, pode-se deduzir da análise dos histogramas da coluna da direita, que a posição correspondente ao primeiro bloco superior à esquerda (posição 0) é a mais utilizada, seguida pelas outras posições do canto do MB (3, 12 e 15), e pelas posições correspondentes aos blocos da margem esquerda e superior do MB (posições 1, 2, 4 e 8). Isto se deve em parte à vantagem que algumas destas posições possuem na predição do VM (a predição do VM nas outras posições é o VM da partição maior, ver seção 5.4).

Tabela 5.4 – Frequência com que o modo *duplo-inter* foi escolhido para diferentes QPs - Cenário 1

Seqüência	Resolução	QP = 18	QP = 21	QP = 24	QP = 27	QP = 30
<i>Hall</i>	QCIF	29,23%	13,20%	5,16%	2,90%	2,30%
<i>Stefan</i>	SIF	10,61%	9,59%	8,92%	8,15%	7,13%
<i>Tennis</i>	SIF	15,01%	10,28%	8,11%	6,62%	4,98%
<i>Foreman</i>	CIF	14,96%	11,78%	9,28%	6,82%	4,39%
<i>Coastguard</i>	CIF	15,81%	13,16%	11,40%	9,64%	7,54%
<i>Foreman</i>	QCIF	17,51%	14,80%	12,49%	9,47%	6,21%
<i>Coastguard</i>	QCIF	17,95%	15,06%	13,29%	11,49%	9,05%
<i>Mother</i>	QCIF	9,75%	8,10%	6,41%	4,75%	3,32%
<i>Mobile</i>	QCIF	22,39%	20,18%	19,65%	17,97%	15,59%

Tabela 5.5 – Frequência com que os diferentes modos foram escolhidos, para QP = 27 - Cenário 1

Seqüência	Resolução	Skip	16×16	16×8	8×16	8×8	<i>intra</i>	<i>duplo-inter</i>
<i>Hall</i>	QCIF	77,13%	8,67%	1,37%	1,45%	7,35%	1,13%	2,90%
<i>Stefan</i>	SIF	13,73%	14,49%	7,81%	4,46%	24,35%	27,01%	8,15%
<i>Tennis</i>	SIF	33,20%	12,76%	7,89%	5,57%	25,77%	8,19%	6,62%
<i>Foreman</i>	CIF	21,69%	27,02%	9,37%	7,92%	12,05%	15,13%	6,82%
<i>Coastguard</i>	CIF	10,93%	27,10%	12,83%	7,89%	22,92%	8,69%	9,64%
<i>Foreman</i>	QCIF	14,36%	25,58%	10,25%	10,34%	22,90%	7,10%	9,47%
<i>Coastguard</i>	QCIF	10,78%	27,35%	12,55%	7,97%	27,78%	2,08%	11,49%
<i>Mother</i>	QCIF	53,59%	17,58%	7,56%	6,20%	9,14%	1,18%	4,75%
<i>Mobile</i>	QCIF	2,13%	17,48%	11,97%	9,24%	40,52%	0,69%	17,97%

b) Cenário 2

Neste cenário, as seqüências de imagens são codificadas utilizando o código P_MIX de 5 bits para indicar o modo *duplo-inter* e a **predição alternativa** para o VM da **partição menor**.

Apesar de essas duas alterações em relação ao cenário 1 serem aplicadas simultaneamente, o efeito de cada uma pode ser analisado separadamente: o uso do código de 5 bits diminui a frequência de utilização do modo *duplo-inter*, enquanto a predição alternativa atua sobre a escolha das diferentes posições da partição menor (de acordo com a predição mais adequada).

Na coluna da esquerda da Fig. 5.8, as barras em azul correspondem à utilização dos diferentes modos na codificação padrão H.264 (Cenário 0), enquanto as barras em verde correspondem à utilização dos diferentes modos quando adicionado o modo *duplo-inter* (Cenário 2). A coluna da direita indica a frequência com que as possíveis posições da partição menor no modo *duplo-inter* são escolhidas.

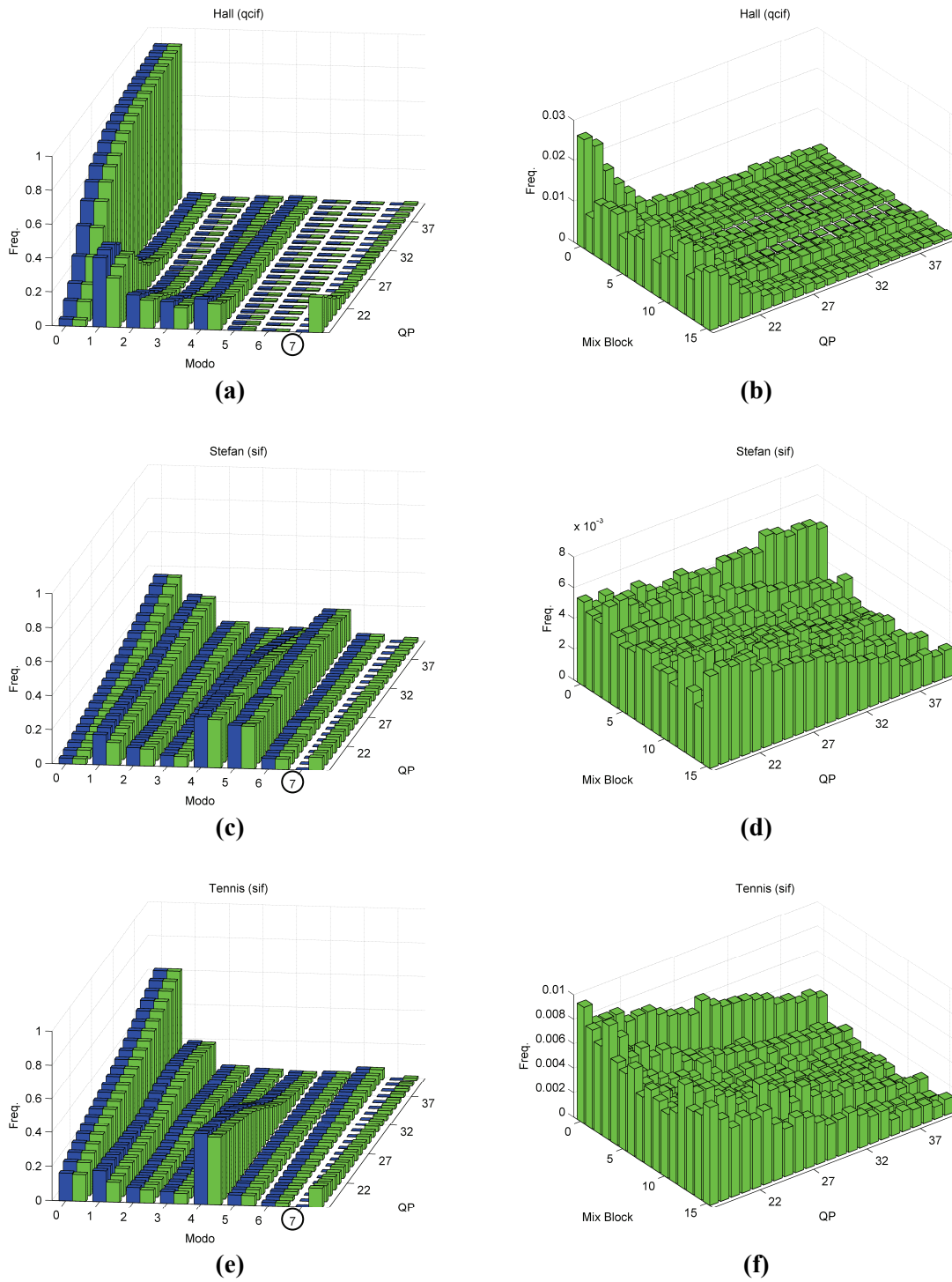
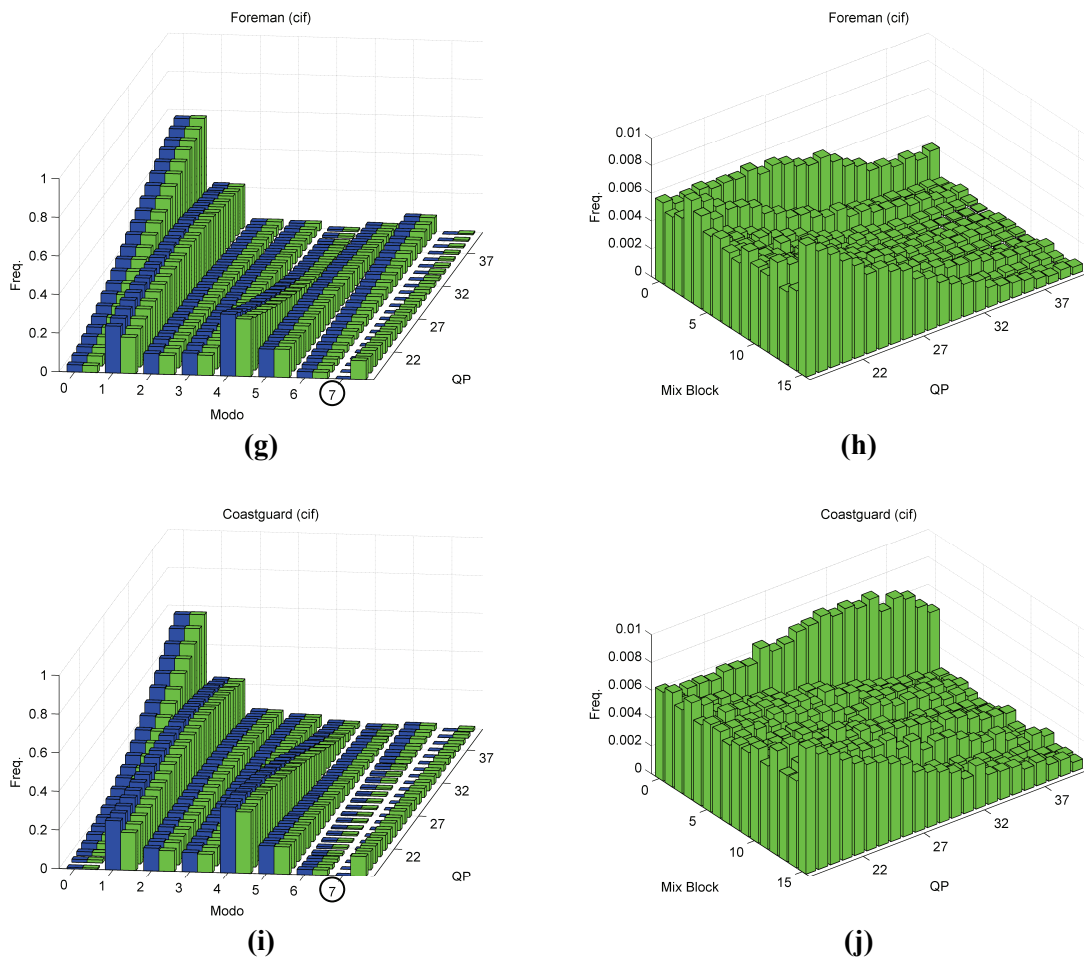


Fig. 5.8 – Coluna da esquerda: histograma dos tipos de MBs selecionados, para cada sequência (sem modo *duplo-inter* em azul e com modo *duplo-inter* em verde). Coluna da direita: histograma da posição selecionada (*mix_block*). O QP varia entre 18 e 40.



(continuação) Fig. 5.8 – Coluna da esquerda: histograma dos tipos de MBs selecionados, para cada seqüência (sem modo *duplo-inter* em azul e com modo *duplo-inter* em verde). Coluna da direita: histograma da posição selecionada (*mix_block*). O QP varia entre 18 e 40.

Comparando os resultados da seleção dos diversos modos (histogramas da coluna da esquerda) obtidos neste cenário 2, com aqueles obtidos para o cenário 1, podemos notar que houve uma diminuição na utilização do modo *duplo-inter* (índice 7) em todas as seqüências, devido à utilização do código `P_MIX` de 5 bits. Simultaneamente a frequência de utilização do modo 8×16 (índice 3) aumentou, ficando próxima à do modo 16×8 .

Analisando os histogramas da coluna da direita, pode-se perceber que não houve muita mudança com a introdução da **predição alternativa**. Houve uma diminuição de frequência em todas as posições (causada pela utilização do código de 5 bits), mas a utilização das diferentes posições da partição menor não ocorre de forma mais homogênea, indicando que a predição alternativa não foi eficiente nestes casos.

Tabela 5.6 – Freqüência com que os MBs foram escolhidos como tipo *duplo-inter*, para diferentes QPs, - Cenário 2

Sequência	Resolução	QP = 18	QP = 21	QP = 24	QP = 27	QP = 30
Hall	QCIF	20,96%	9,80%	3,52%	2,39%	1,66%
Stefan	SIF	7,69%	6,38%	5,82%	5,13%	4,31%
Tennis	SIF	11,30%	7,80%	5,98%	4,72%	3,46%
Foreman	CIF	9,76%	7,54%	5,68%	4,20%	2,61%
Coastguard	CIF	10,42%	8,48%	7,19%	5,93%	4,78%

Tabela 5.7 – Freqüência com que os MBs foram escolhidos nos diferentes modos, para QP = 27, no Cenário 2

Seqüência	Resolução	Skip	16×16	16×8	8×16	8×8	Intra	Duplo-Inter
Hall	QCIF	77,39%	8,53%	1,44%	1,84%	7,23%	1,18%	2,39%
Stefan	SIF	13,77%	15,89%	7,83%	5,71%	24,24%	27,43%	5,13%
Tennis	SIF	32,91%	13,58%	7,98%	7,49%	25,04%	8,28%	4,72%
Foreman	CIF	21,67%	27,20%	9,17%	10,27%	11,95%	15,54%	4,20%
Coastguard	CIF	10,94%	27,93%	12,92%	10,70%	22,82%	8,76%	5,93%

5.5.3. Ganho em Taxa-Distorção

As Figs. 5.9 e 5.10 mostram as curvas de taxa-distorção obtidas para os cenários 1 e 2, e os respectivos ganhos de PSNR que foram obtidos com as modificações propostas.

a) Cenário 1

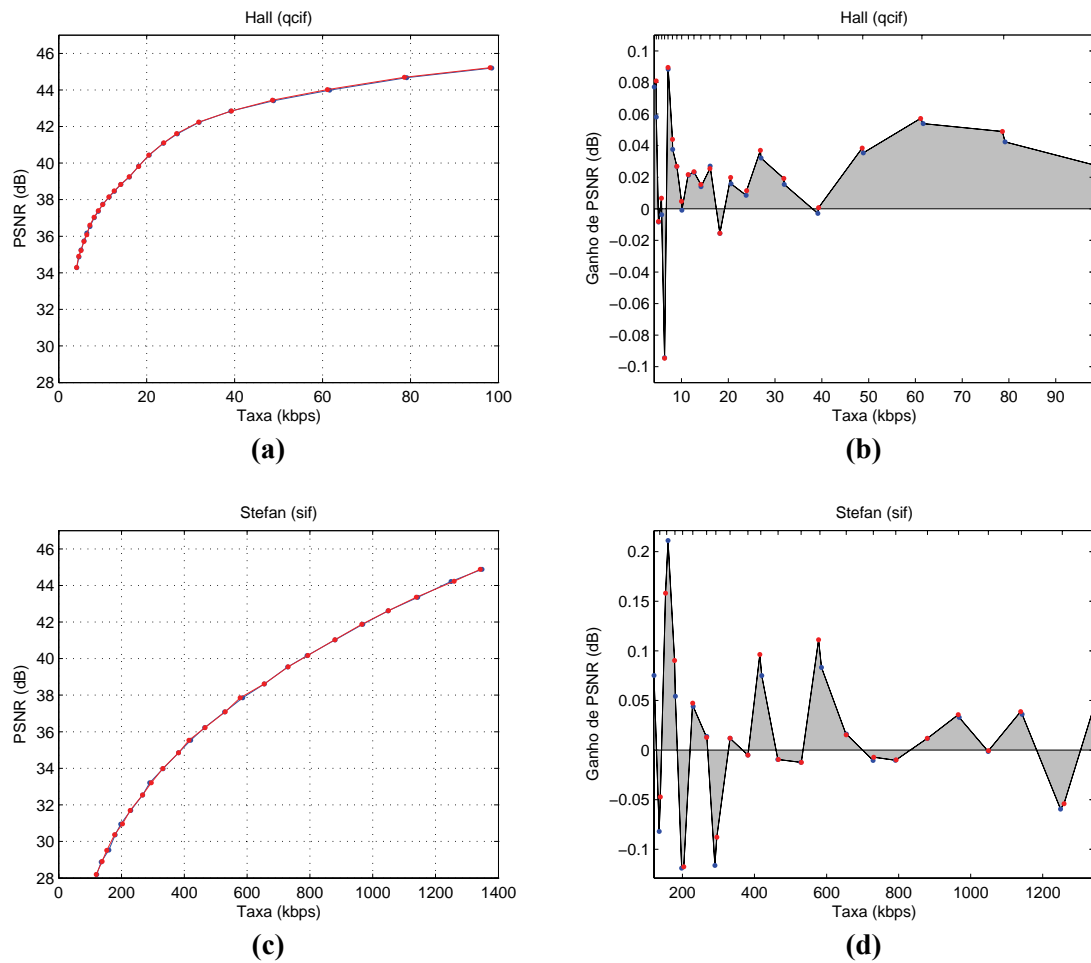
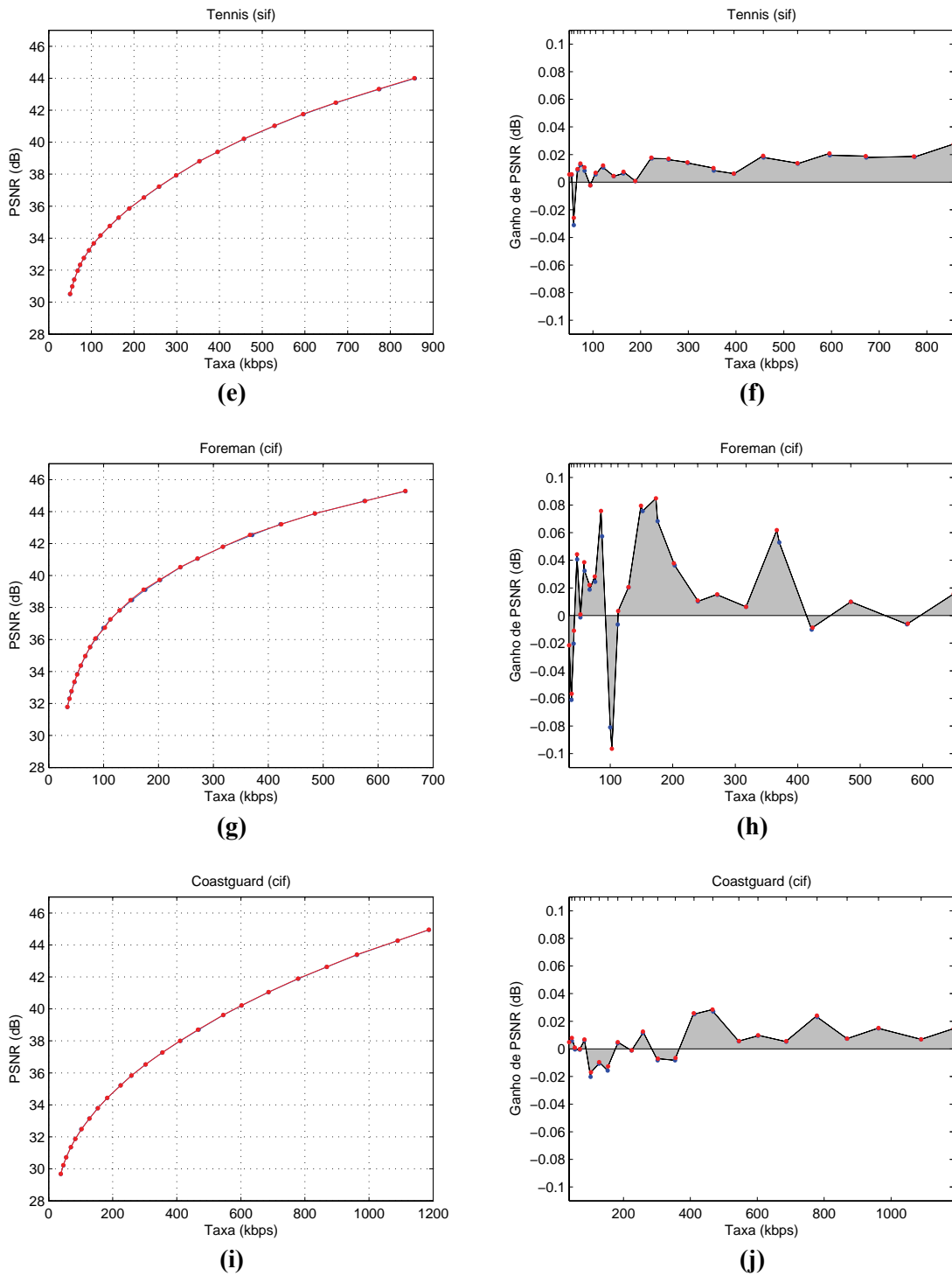
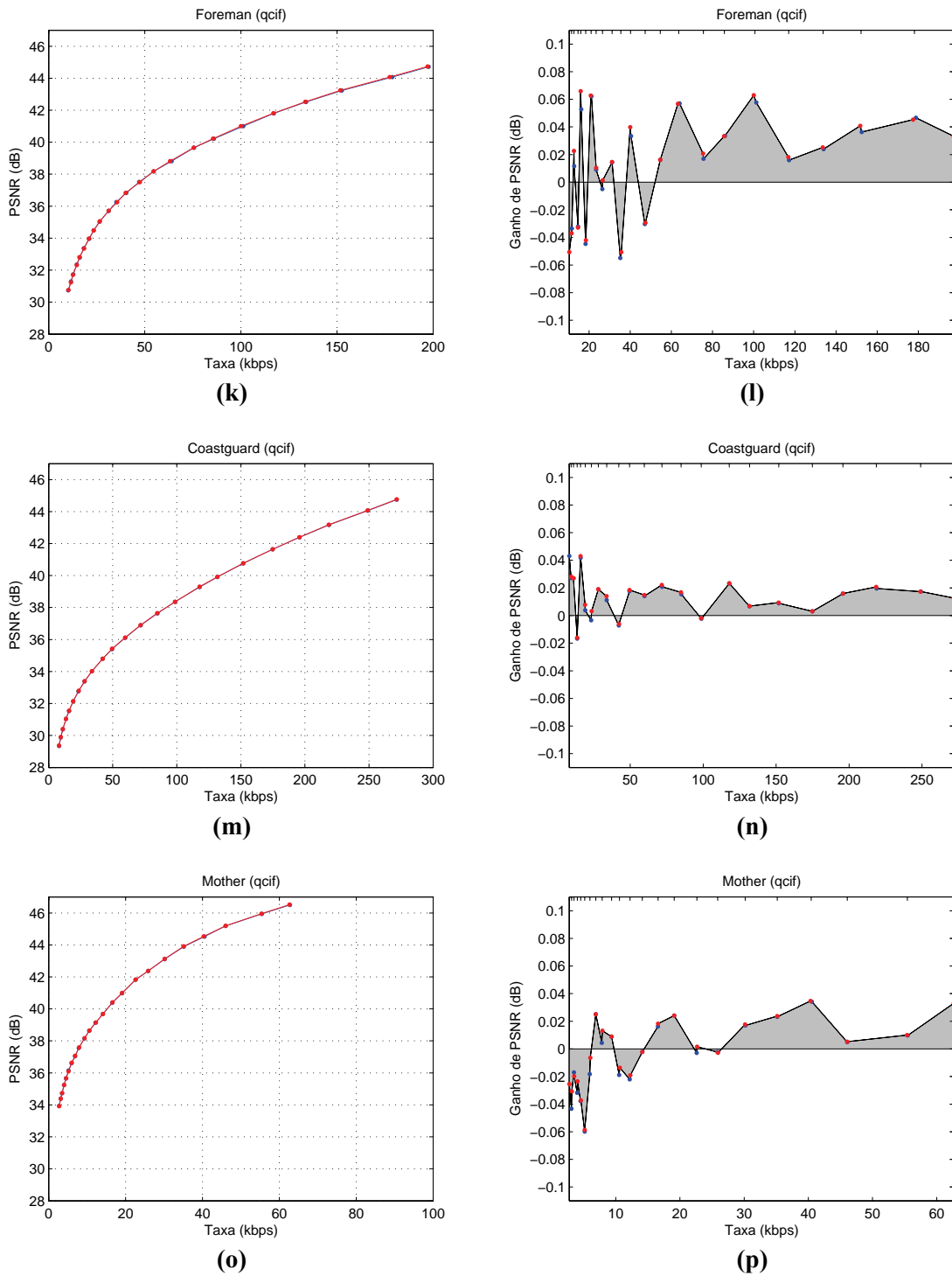


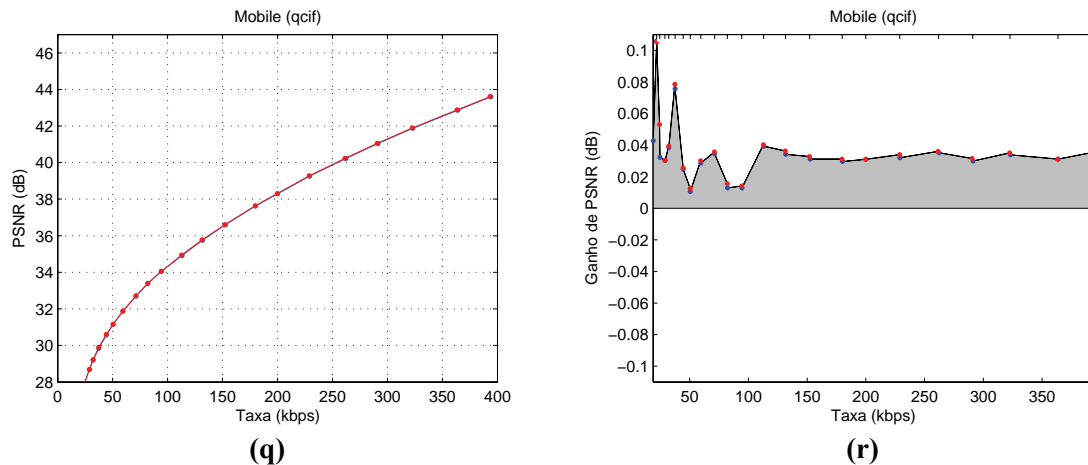
Fig. 5.9 – Curvas taxa-distorção da codificação no Cenário 0 (padrão H.264) e da codificação no Cenário 1 (com modo *duplo-inter*, coluna da esquerda). Diferença entre a distorção obtida no Cenário 1 e a obtida no Cenário 0 (coluna da direita). Os “marcadores” do eixo superior indicam o QP, que varia de 40 (mais à esquerda) a 18 (mais à direita).



(continuação) Fig. 5.9 – Curvas taxa-distorção da codificação no Cenário 0 (padrão H.264) e da codificação no Cenário 1 (com modo *duplo-inter*, coluna da esquerda). Diferença entre a distorção obtida no Cenário 1 e a obtida no Cenário 0 (coluna da direita). Os “marcadores” do eixo superior indicam o QP, que varia de 40 (mais à esquerda) a 18 (mais à direita).



(continuação) Fig. 5.9 – Curvas taxa-distorção da codificação no Cenário 0 (padrão H.264) e da codificação no Cenário 1 (com modo *duplo-inter*, coluna da esquerda). Diferença entre a distorção obtida no Cenário 1 e a obtida no Cenário 0 (coluna da direita). Os “marcadores” do eixo superior indicam o QP, que varia de 40 (mais à esquerda) a 18 (mais à direita).



(continuação) Fig. 5.9 – Curvas taxa-distorção da codificação no Cenário 0 (padrão H.264) e da codificação no Cenário 1 (com modo duplo-inter, coluna da esquerda). Diferença entre a distorção obtida no Cenário 1 e a obtida no Cenário 0 (coluna da direita). Os “marcadores” do eixo superior indicam o QP, que varia de 40 (mais à esquerda) a 18 (mais à direita).

Nota-se que, apesar da alta taxa de utilização, o ganho obtido com o modo *duplo-inter* não é muito significativo (ganho médio de 0,014 dB), apesar de atingir máximos entre 0,11 dB (para a seqüência *Mobile*) e 0,21 dB (para a seqüência *Stefan*). Isso se deve a pouca diferença na taxa ou na distorção do modo proposto em relação ao modo que ele substituiu.

Em alguns casos, principalmente com as taxas de bits menores, obteve-se uma piora no desempenho TD. Isso pode ocorrer porque o processo de OTD é realizado isoladamente para cada MB, sem levar em consideração os efeitos das escolhas feitas para cada MB sobre os MBs futuros. Por exemplo, para um determinado MB, a escolha do modo *duplo-inter* pode implicar na redução da quantidade de bits e aumento da distorção. No entanto, se em quadros futuros este MB for codificado no modo *skip*, a distorção se manterá ou aumentará, fazendo com que a qualidade total diminua sem que haja uma compensação em termos de taxa de bits. Já um modo que resultasse em uma taxa maior com uma distorção menor poderia obter, neste caso, uma melhor relação taxa-distorção nos quadros futuros.

Tabela 5.8 – Ganhos médio e máximo de PSNR no Cenário 1 em relação à codificação padrão

Seqüência	Resolução	Ganho Médio	Ganho Máximo
<i>Hall</i>	QCIF	0,02 dB	0,09 dB
<i>Stefan</i>	SIF	0,01 dB	0,21 dB
<i>Tennis</i>	SIF	0,01 dB	0,03 dB
<i>Foreman</i>	CIF	0,01 dB	0,08 dB
<i>Coastguard</i>	CIF	0,00 dB	0,03 dB
<i>Foreman</i>	QCIF	0,01 dB	0,07 dB
<i>Coastguard</i>	QCIF	0,01 dB	0,04 dB
<i>Mother</i>	QCIF	0,00 dB	0,03 dB
<i>Mobile</i>	QCIF	0,04 dB	0,11 dB

b) Cenário 2

Em termos de taxa-distorção os resultados obtidos no Cenário 2 são muito próximos ao do Cenário 1 para todas as seqüências.

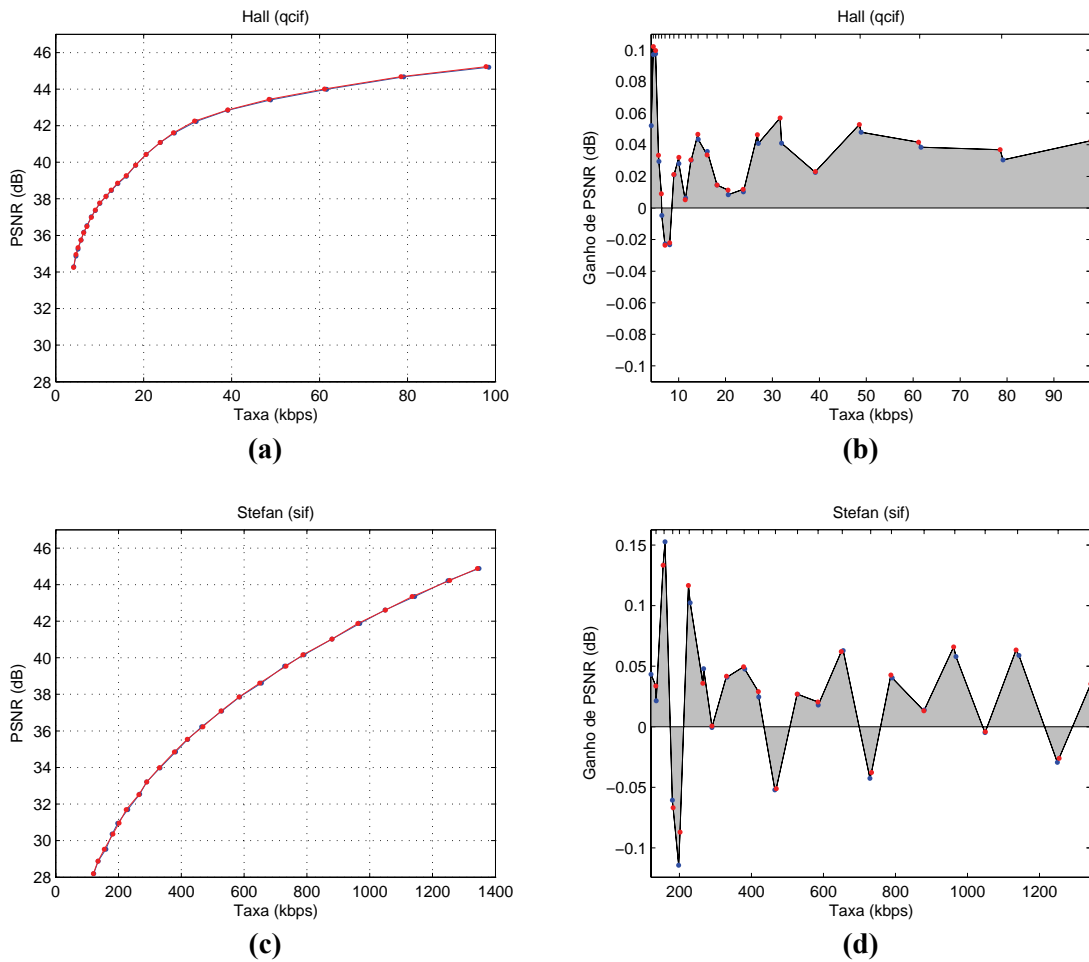
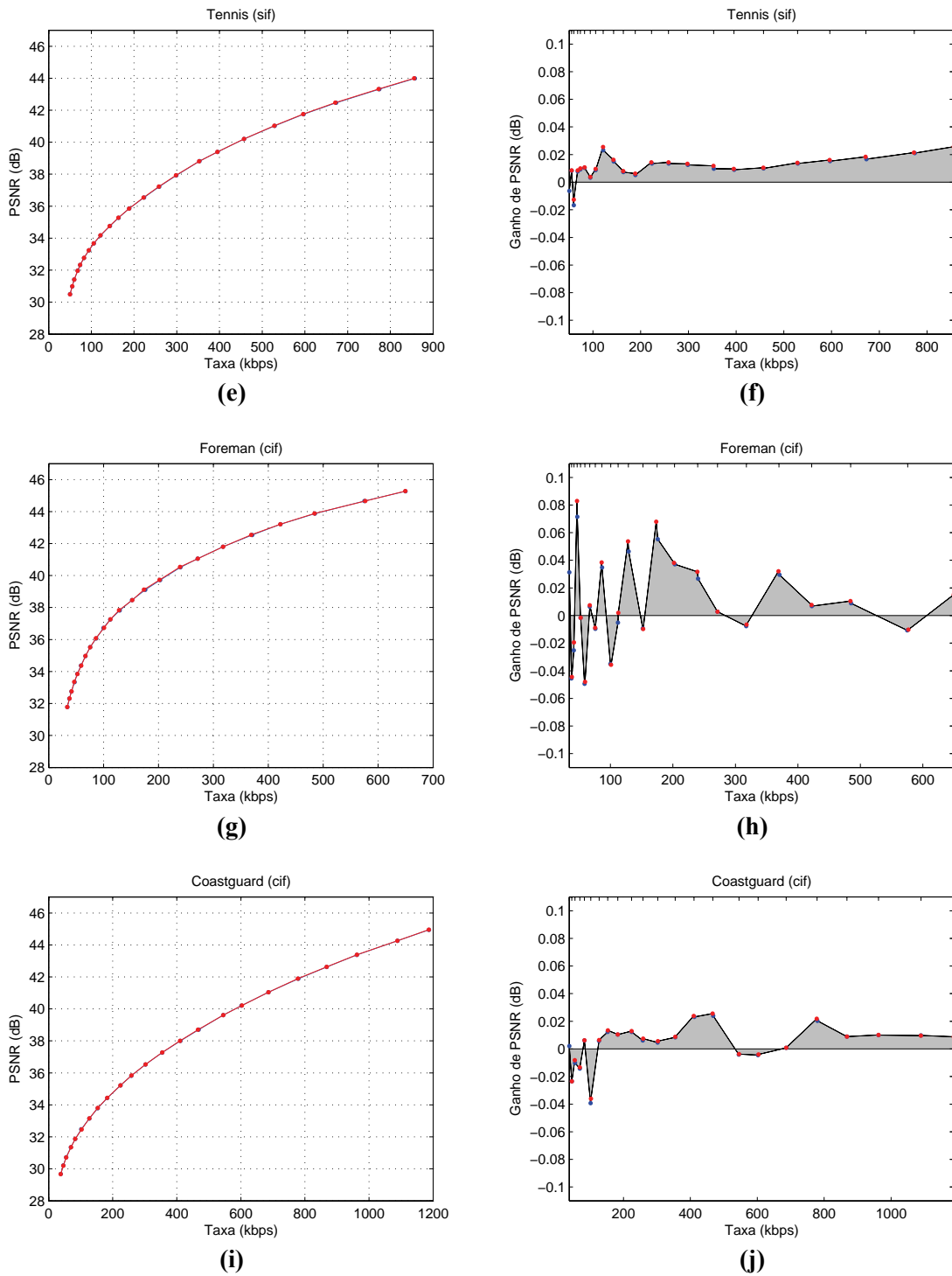


Fig. 5.10 – Curva taxa-distorção da codificação normal e da codificação com modo *duplo-inter* (coluna da esquerda). Diferença entre a distorção obtida no modo *duplo-inter* e a obtida no modo normal (coluna da direita - quanto mais, melhor). Os “marcadores” do eixo superior indicam o QP, que varia de 40 (mais à esquerda) e 18 (mais à direita).



(continuação) Fig. 5.10 – Curva taxa-distorção da codificação normal e da codificação com modo *duplo-inter* (coluna da esquerda). Diferença entre a distorção obtida no modo *duplo-inter* e a obtida no modo normal (coluna da direita - quanto mais, melhor). Os “marcadores” do eixo superior indicam o QP, que varia de 40 (mais à esquerda) e 18 (mais à direita).

Tabela 5.9 – Ganhos médio e máximo de PSNR no Cenário 2 em relação à codificação padrão

Seqüência	Resolução	Ganho Médio	Ganho Máximo
<i>Hall</i>	QCIF	0,03 dB	0,10 dB
<i>Stefan</i>	SIF	0,02 dB	0,15 dB
<i>Tennis</i>	SIF	0,01 dB	0,03 dB
<i>Foreman</i>	CIF	0,01 dB	0,08 dB
<i>Coastguard</i>	CIF	0,00 dB	0,03 dB

5.6. Conclusões

Considerando o ganho médio obtido para todas as seqüências, o MB *duplo-inter* teve um baixo desempenho; porém, para seqüências como *Stefan* e *Mobile*, que são mais complexas e possuem bastante movimento, o MB *duplo-inter* obteve algum ganho. Esse resultado era esperado, pois esse modo foi proposto para obter ganhos em seqüências com tais características. Nessas mesmas seqüências, o MB *duplo-inter* foi utilizado com grande freqüência, mostrando que há uma correlação entre sua utilização e o ganho obtido.

Foi proposto um modo alternativo de predição de VM para a partição menor do MB *duplo-inter*, porém essa predição não causou mudanças significativas nos resultados. Além disso, foram avaliados dois diferentes códigos para indicar que o MB é do tipo *duplo-inter*, um com 3 bits e outro com 5 bits. Apesar de haver uma grande diferença na utilização desse modo (o modo *duplo-inter* é menos utilizado com o código de 5 bits), os resultados em termos de ganho são bem parecidos.

2ª PROPOSTA DE MODIFICAÇÃO: CODIFICAÇÃO COM MB “HÍBRIDO” INTER/INTRA

6.1. Introdução

A codificação proposta neste capítulo é útil quando apenas uma parte do MB pode ser predita a partir dos quadros anteriores. A Fig. 6.1 ilustra um MB que se beneficiaria da codificação híbrida.

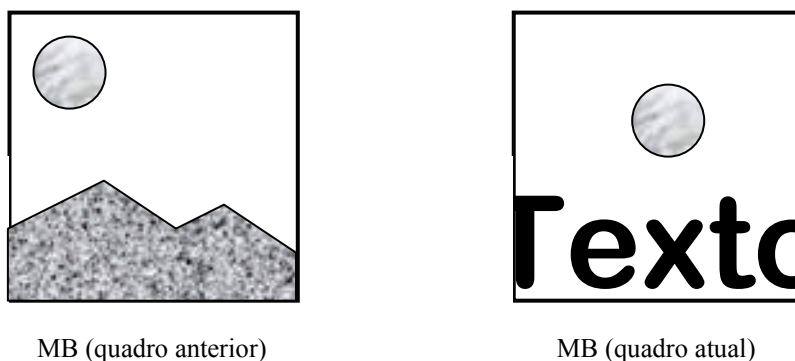


Fig. 6.1 – Exemplo de um MB que pode se beneficiar da codificação com MB híbrido, pois possui uma região que pode ser predita através da predição *inter* (círculo) e outra região que se beneficiaria mais da predição *intra*.

O modo de codificação de MB proposto possibilita a codificação de subMBs⁵ *intra* 8×8 dentro de MBs *inter* 16×16, permitindo uma maior flexibilidade de escolha no processo de OTD. A possibilidade de se utilizar subMBs *intra* e *inter* no mesmo MB já havia sido considerada no projeto inicial de particionamento de MB do H.264 [36], porém, abandonada por razões de complexidade no decodificador [37]. Propomos então a criação

⁵ Submacroblocos, equivalem à ¼ do macrobloco, ou seja, 8×8 amostra *luma* e correspondentes amostras *croma*.

de um novo tipo de subMB, o P_I_8x8, que é um bloco 8x8, porém codificado de forma *intra*. Assim, poderemos ter MBs *híbridos*, possibilitando o codificador dispor uma maior flexibilidade, com uma granularidade menor de decisão entre *intra/inter*. Poderemos assim ter MBs do seguinte tipo:

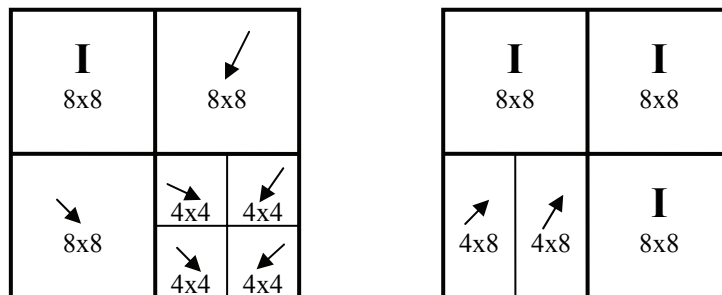


Fig. 6.2 – Exemplos de MBs *híbridos inter/intra* possíveis.

O MB *híbrido* proposto é uma variação do MB de tipo P_8x8 (*mb_type* = 3). De acordo com o padrão H.264, um MB pode ser dividido em 4 subMBs *inter* de 8x8, sendo que o MB *híbrido* permite que alguns desses subMBs sejam codificados como *intra*.

6.2. Alteração dos Processos do Codificador

Para acomodar o novo modo na codificação, o encadeamento dos processos do codificador (e decodificador) foi alterado, com a inserção da predição híbrida. Esse processo utiliza tanto as amostras do quadro atual original (predição *intra*) quanto as dos quadros anteriores reconstituídas⁶ (predição *inter*), como mostra a Fig. 6.3.

⁶ As amostras que estarão disponíveis no *decodificador*, pertencentes ao sinal decodificado. São obtidas através da codificação e subseqüentes decodificação.

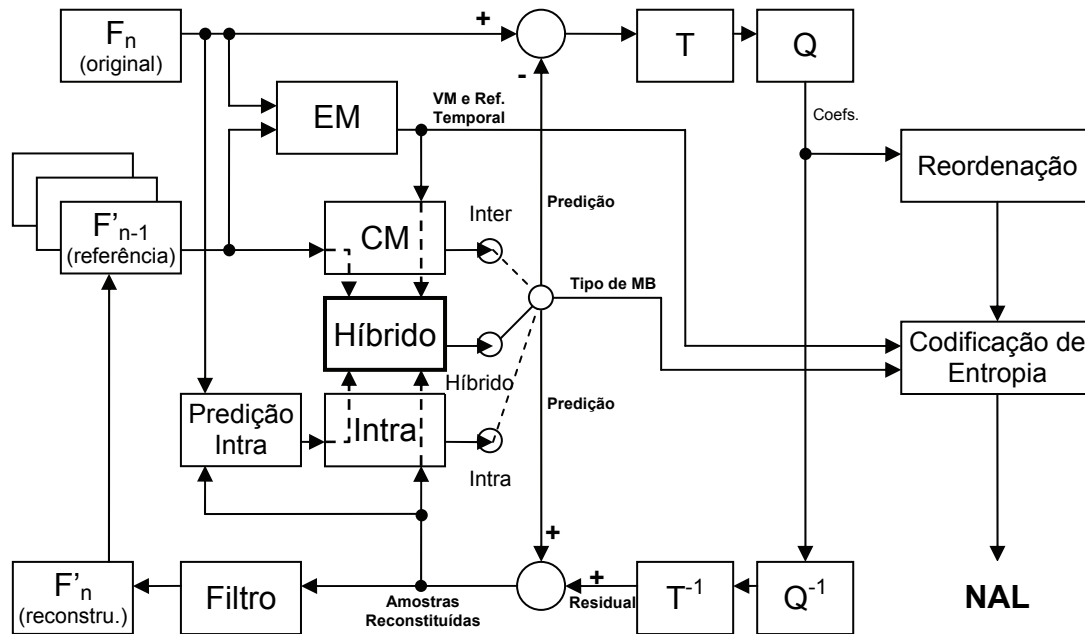


Fig. 6.3 – Esquematisação dos processos do codificador com a inserção do modo *híbrido*. As setas tracejadas indicam que o processo de codificação do MB *híbrido* recebe as mesmas entradas que o processo de CM (quadros anteriores e VMs) e de codificação *intra* (predição *intra* e amostras reconstituídas).

Note que o processo *híbrido* não emprega o processo de predição *intra* original do H.264, pois deve considerar as amostras reconstituídas⁷ dos outros subMBs do mesmo MB, uma vez que parte do MB pode usar predição *inter*.

6.3. Alterações no Fluxo de Bits

Para a codificação do novo tipo de subMB, deve-se alocar um código de bits que não entre em conflito com os códigos de bits existentes. No MB de tipo $P_{8 \times 8}$, logo após a especificação do `mb_type`, seguem 4 códigos, referentes aos tipos de cada um dos subMBs. Por exemplo, para o MB da Fig. 6.4, os quatro SubMBs são codificados através de 1, 010, 011, 00100 (representando, respectivamente, os tipos 0, 1, 2, e 3). O padrão especifica que a sintaxe para a codificação destes valores é o $ue(v)$ (vide Seção 4.3.1).

⁷ As amostras que estarão disponíveis no **decodificador**, pertencentes ao sinal decodificado. São obtidas através da codificação e subsequente decodificação.

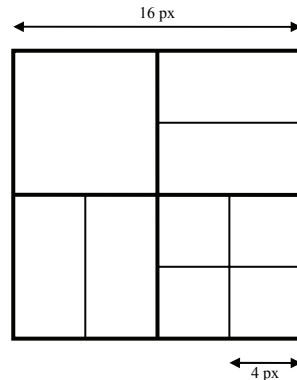


Fig. 6.4 - MB com 4 partições 8x8 e com 4 particionamentos diferentes para cada subMB: 8x8 para o subMB 0 (ao topo e à esquerda), 8x4 para o subMB 1 (ao topo e à direita), 4x8 para o subMB 2 (abaixo e à esquerda) e 4x4 (abaixo e à direita).

Como o $ue(v)$ é uma codificação “aberta”, ou seja, não existe um valor limite, então no caso do nosso novo subMB $P_I_8 \times 8$, simplesmente atribuímos a ele o próximo valor disponível (valor 4), que tem o código 00101.

6.3.1. Codificação Original do H.264

Com a finalidade de comparar com as alterações propostas, nesta seção é descrita a codificação dos MBs do tipo $P_8 \times 8$, **de acordo com o padrão H.264**. Os elementos de sintaxe de tais MBs são codificados na seguinte seqüência:

1. O próprio tipo do MB (mb_type , que neste caso é 3 ou 4, dependendo do item nº.3 a seguir).
2. Quatro elementos codificando os tipos de cada um dos subMBs 8x8 (sub_mb_type).
3. Quatro elementos codificando a referência temporal de cada subMB (ref_idx_l0). Assim, é possível que dentro do mesmo MB tenhamos subMBs cujas predições *inter* se referem a quadros diferentes. Caso todos os subMBs estejam referenciando o quadro 0 da lista 0 (que é geralmente o quadro anterior, uma situação que ocorre com frequência), estes valores são omitidos, economizando-se 4 bits; isso é sinalizado utilizando-se um mb_type do tipo 4. Caso contrário utiliza-se o mb_type do tipo 3.

4. Uma quantidade variável de elementos (de 8 a 32) codificando os VMs. Para cada VM, é transmitido um par de valores que representam os componentes x e y da diferença entre o vetor corrente e o vetor **predito** a partir dos vizinhos (em quartos de amostra *luma*). O número total de vetores de movimento depende dos tipos dos subMBs (codificados no item nº.2), sendo um vetor para um subMB do tipo $P_L0_8 \times 8$, dois para os tipos $P_L0_8 \times 4$ e $P_L0_4 \times 8$ e quatro para o tipo $P_L0_4 \times 4$. Por exemplo, para o MB da Fig. 6.4, temos um total de 9 vetores de movimento.
5. CBP, padrão de bloco **codificado**, um elemento formado por 6 bits, dos quais:
 - 4 bits indicam quais dos 4 subMBs possuem coeficientes da codificação dos residuais (caso não possuam, é utilizada apenas a predição *inter*);
 - 2 bits indicam se a codificação do residual do *croma* possui coeficientes (00), se possui apenas coeficientes DC (01) ou se possui coeficientes DC e AC (10).

Este número de 6 bits do CBP é mapeado para um outro valor tabelado antes da codificação de entropia, para melhor representar sua distribuição em frequência.

6. Um parâmetro Δ_{QP} que indica a variação no parâmetro de quantização do MB (geralmente 0, considerando apenas 1 bit).
7. Todos os coeficientes *luma*, se houverem.
8. Todos os coeficientes *croma DC* (se houverem), seguidos pelos coeficientes *croma AC*.

6.3.2. Alteração Proposta na Codificação

Para a inserção do modo *híbrido*, é proposto alterar o fluxo de bits da seguinte forma:

- O número de referências temporais (ref_idx_l0) depende do número de subMBs *intra*, uma vez que esses subMBs não possuem referência temporal.

- Nenhum elemento de vetor de movimento é transmitido para subMBs *intra*. Para fins de estimação de movimento, tanto na codificação quanto na decodificação, assume-se implicitamente que o VM dos subMBs *intra* é equivalente ao VM que seria utilizado se o MB fosse do tipo *skip*.
- Acréscimo de um novo valor para **sub_mb_type**, o tipo 4 (P_I_8×8).
- Inserção de um novo tipo de elemento: o **modo de predição**⁸ *intra* para os subMBs P_I_8×8. Esses elementos são inseridos após o item 4 descrito na seção anterior (entre os vetores de movimento e o CBP). Para cada subMB P_I_8×8, são codificados 4 modos de predição *intra*, um para cada bloco 4×4. A codificação dos modos de predição é idêntica à codificação original do padrão H.264. A Tabela 4.3 mostra os possíveis modos de predição, que estão ilustrados na Fig. 4.9.

Na codificação do modo de predição *intra*, é calculado o modo de predição **mais provável**, baseado nos modos de predição do bloco imediatamente acima e do bloco imediatamente à esquerda. Se o modo escolhido é o modo mais provável, ele é codificado com apenas um bit, caso contrário, é codificado com tamanho fixo de 4 bits. Equivale à uma predição do próprio **modo de predição** *intra*, baseada nos modos utilizados nos vizinhos.

No processo de OTD de um MB *híbrido*, antes de serem testadas as predições *intra* em um subMB, deve ser realizado o processo de reconstituição dos outros subMBs (no padrão H.264, esse processo de reconstituição é realizada apenas para MBs completos). Essa modificação também deve ser feita no decodificador, implicando apenas em uma reordenação das amostras a serem reconstituídas, sem aumento da complexidade computacional do mesmo.

6.4. Resultados

Para avaliar este novo modo de codificação, são utilizadas as mesmas seqüências de vídeo do Capítulo 5. O modelo de referência utilizado como base para tais alterações foi o

⁸ Vide Seção 3.3.1.

JM 10.2. Realizamos os testes inserindo a proposta apenas na codificação de quadros do tipo P, limitando também a codificação de bits para o tipo CAVLC.

6.4.1. Ganho em Taxa-Distorção

O resultado final do ganho em termos de taxa-distorção obtido está mostrado na Fig. 6.5.

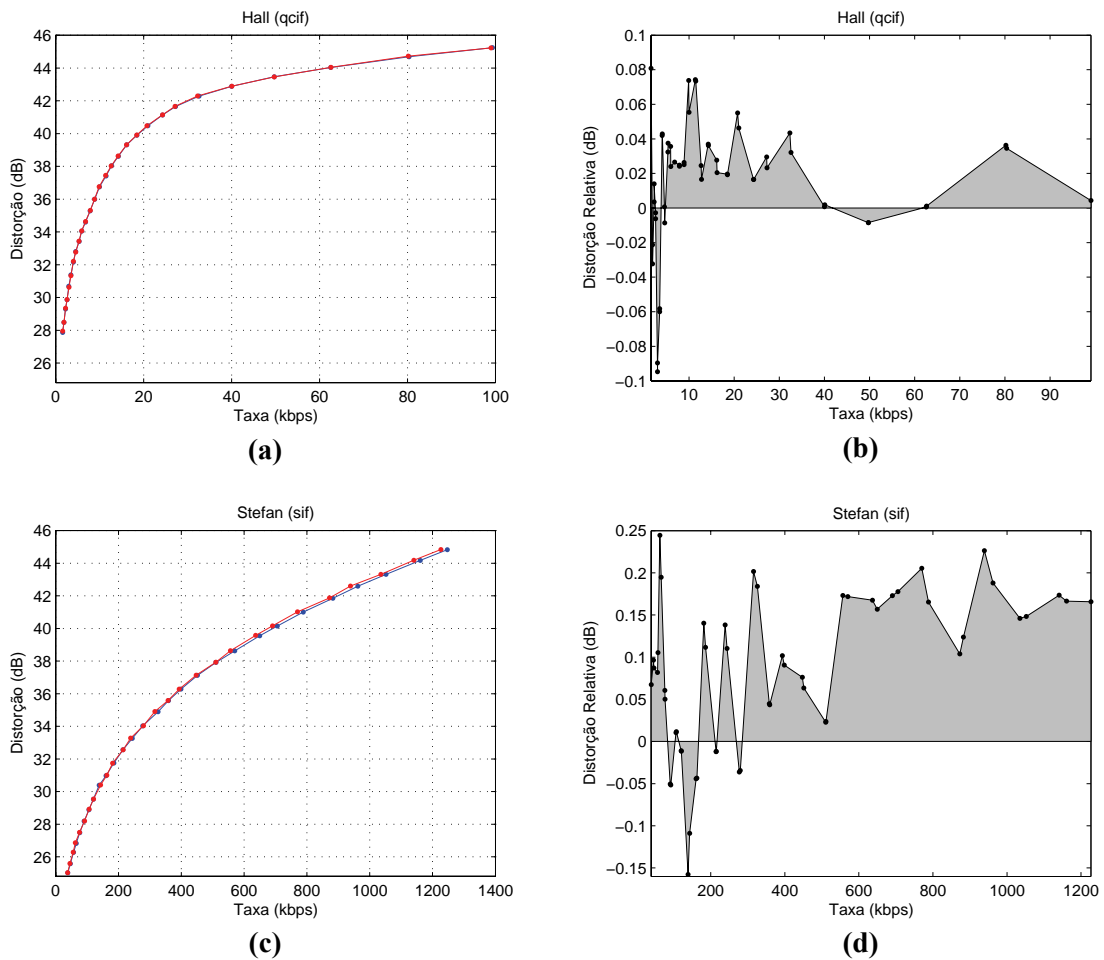
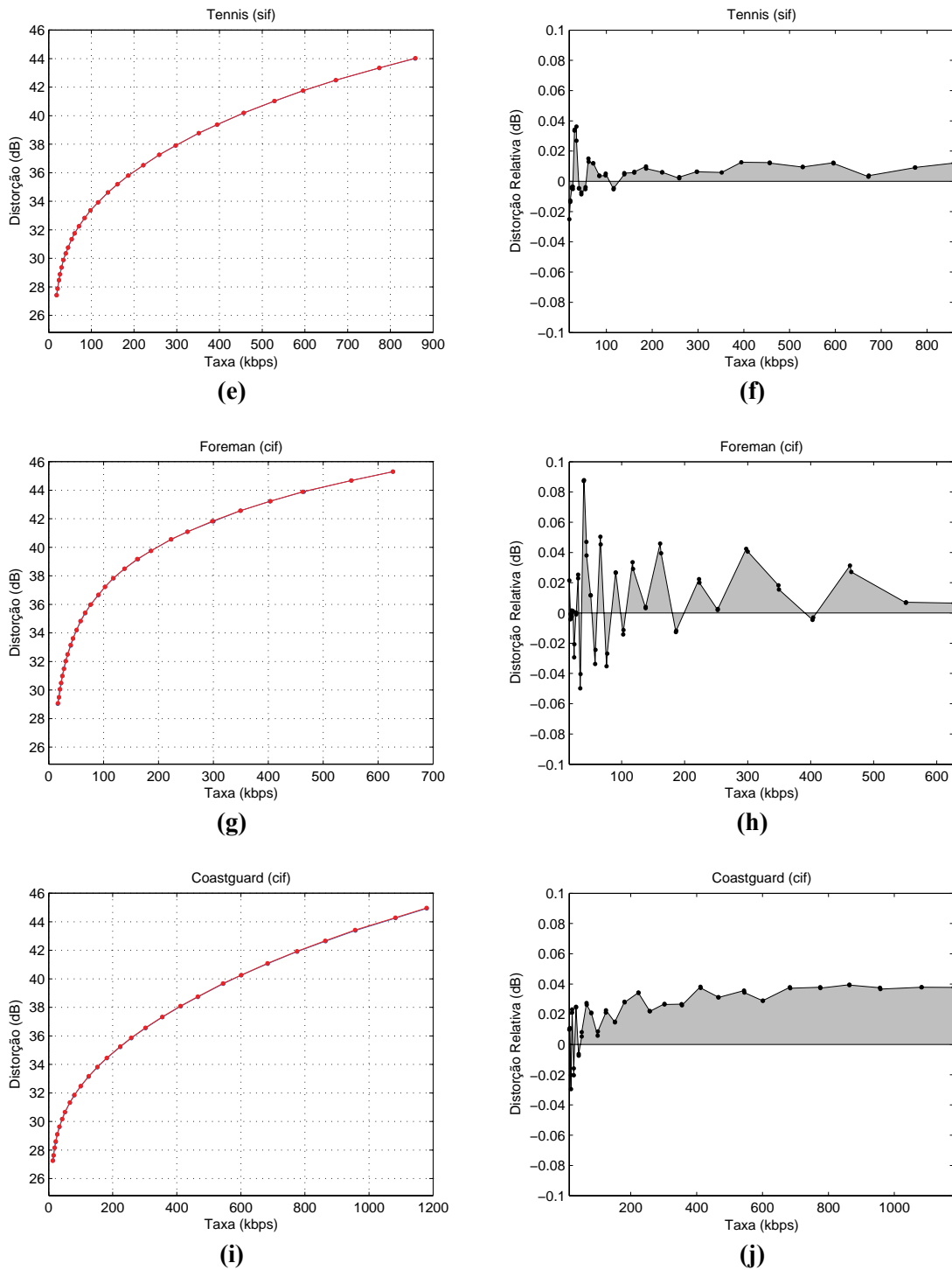


Fig. 6.5 – Comparação das curvas taxa-distorção obtidas para a codificação padrão e a codificação com o modo *híbrido inter/intra*.



(continuação) Fig. 6.5 – Comparação das curvas taxa-distorção obtidas para a codificação padrão e a codificação com o modo híbrido *inter/intra*.

Como podemos notar, os resultados dependem muito do conteúdo da seqüência. A seqüência que apresenta o melhor resultado é a *Stefan*, com um significativo ganho, próximo a 0,25 dB. Esse resultado deve-se principalmente ao fundo complexo (espectadores) e à movimentação relevante da câmera. Tal resultado é esperado, pois a

referida seqüência representa uma situação típica em que há a mistura de pequenos elementos que podem ser preditos a partir de quadros anteriores e outros que não podem ser.

6.5. Conclusões

A codificação com o modo de MB *híbrido* permite associar, em um mesmo MB, duas formas de codificação que anteriormente eram mutuamente excludentes: a codificação com predição *inter-quadros* (que utiliza vetores de movimento e referências temporais) e a codificação com predição *intra-quadro* (que utiliza as amostras vizinhas de um bloco). A predição *inter* é mais adequada para regiões cujas amostras apenas sofrem um deslocamento, enquanto a *intra* é apropriada para regiões que ainda não apareceram em quadros anteriores. Para MBs que apresentam simultaneamente essas duas situações, o codificador do padrão H.264 tem que optar por uma das duas predições, sendo que nenhuma é adequada; a introdução do MB *híbrido* permite que se particione o MB e codifique cada subMB utilizando a melhor predição (*intra* ou *inter*).

A codificação com o MB *híbrido* possibilita ganhos moderados na qualidade de vídeo (medida através do PSNR), principalmente em seqüências que possuem movimentação em conjunto com o aparecimento ou modificação de “objetos”, como a da seqüência *Stefan*.

CONCLUSÃO

7.1. Discussão

Nesta dissertação, foram propostos dois novos modos de codificação de macroblocos para serem incluídos no padrão H.264. Uma descrição detalhada dos processos principais que afetam ou são afetados pelas modificações é também apresentada. As modificações são realizadas alterando-se o *software* de referência do H.264, o qual já tem sido modificado por diversos autores, acrescentando características ao longo do processo de desenvolvimento do referido padrão e suas subseqüentes extensões.

Alguns ganhos obtidos com as alterações sugeridas são significativos, de até 0,25 dB em algumas seqüências, podendo ser bem explorados na filosofia de desenvolvimento do H.264 (na qual várias melhorias com pequenos ganhos vêm sendo somados). Essa filosofia de desenvolvimento é mais bem ilustrada se considerarmos a evolução dos protótipos utilizados durante o processo de padronização do H.264 (Fig. 1.2): foi obtido um ganho de 2 dB entre o TML-1 e o JM 6.1e, que é a soma dos pequenos ganhos obtidos entre as versões intermediárias.

As modificações sugeridas nessa dissertação podem servir de inspiração ou de base para novas alterações ou modos mais avançados que possam resultar em mais ganhos para o referido padrão. O modo *híbrido* mostrou-se mais promissor e poderá ser também reconsiderado nas discussões para o futuro padrão H.265.

7.2. Propostas para Trabalhos Futuros

Em relação ao desempenho de compressão dos padrões anteriores, o H.264 obteve bastante ganho – uma conseqüência disso é que se necessita de processos cada vez mais

complexos para se conseguir mais ganhos no desempenho de codificação. Durante o desenvolvimento deste trabalho de dissertação, surgiram algumas idéias para outras possíveis melhorias do H.264. Apresentamos, inicialmente, sugestões relacionadas à engenharia de *software* do próprio codificador de referência, que poderiam trazer benefícios, tais como ciclos de desenvolvimento mais rápidos; são apresentadas, em seguida, as sugestões que são uma extensão ao trabalho desenvolvido.

a) Codificador e Decodificador de Referência (*lencod/ldecod*)

O modelo de referência é escrito em linguagem C, e o código é muito “denso”, possuindo muitas interconexões entre as sub-rotinas, muitas variáveis globais e baixa modularidade (i.e., uma alteração em uma sub-rotina pode ter efeitos que são indesejados ou obscuros, em diversas outras sub-rotinas). Isso inevitavelmente dificulta a compreensão, manutenção e extensão do modelo.

Para facilitar a implementação de novos processos no codificador de referência e modificação dos processos já existentes, um passo importante seria reescrever o código do codificador, provavelmente em uma linguagem de mais alto nível (como, por exemplo, C++ ou desenvolvendo uma toolbox de H.264 para o Matlab[®]), focando em extensibilidade e manutenibilidade.

b) Continuação do Trabalho Proposto

Uma continuação direta do trabalho realizado seria integrar as duas modificações propostas (codificação com MB *duplo-inter* e com MB *hibrido*), para avaliar o ganho obtido em relação aos ganhos independentes somados.

Na primeira proposta de modificação, testamos também uma predição alternativa de VM que, em princípio, não obteve um ganho muito diferente do alcançado com a predição padrão; entretanto, foi mostrado que há pontos negativos no processo de predição de VM que podem ainda ser mais explorados.

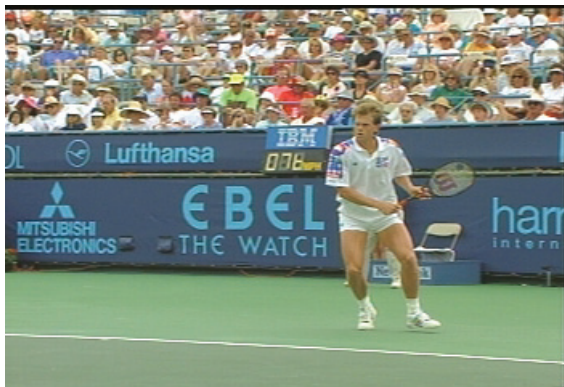
Em relação aos MBs *hibridos*, provavelmente poderíamos ter mais algum ganho também se possibilitássemos a predição *intra* 8×8 junto com a transformada 8×8 (devido aos tamanhos serem iguais). Por outro lado, também poderia ser testado uma menor granularidade na escolha entre *inter* e *intra*, por exemplo, para blocos 4×4.

APÊNDICE A

SEQUÊNCIAS DE TESTE



(a)



(b)



(c)



(d)



(e)

Fig. A.1 – Primeiro quadro de cada uma das seqüências de vídeo utilizadas para os testes. (a) Hall QCIF. (b) Stefan SIF. (c) Tennis SIF. (d) Foreman CIF. (e) Coastguard CIF.



(f)



(g)

(continuação) Fig. A.1 – Primeiro quadro de cada uma das seqüências de vídeo utilizadas para os testes. (f) *Mother & Daughter QCIF*. (g) *Mobile & Calendar QCIF*.

O conteúdo das seqüências está descrito na Tabela A.1.

Tabela A.1 – Descrição das sequências utilizadas nos testes

Nome	Resolução	Descrição
Hall	QCIF (176×144)	Câmera totalmente parada. Movimentação de duas pessoas e objetos em um corredor.
Stefan	SIF (352×240)	Rápida movimentação lateral da câmera. Movimentação rápida dos objetos de frente (tenista e bola). Fundo com muito detalhe espacial (platéia).
Tennis	SIF (352×240)	<i>Zoom out</i> da câmera no início, mudança de cena e câmera estática no final. Fundo texturizado, com detalhe espacial, porém homogêneo. Movimentação da bola e dos jogadores.
Foreman	CIF (352×288) e QCIF	Movimentação rápida da câmera e do rosto de uma pessoa.. Rotação da câmera para uma cena de uma construção.
Coastguard	CIF e QCIF	Movimentação da câmera e movimentação horizontal de barcos em um rio. Oclusão parcial de objetos. Movimento de água.
Mobile & Calendar	QCIF (176×144)	<i>Zoom out</i> da câmera, movimentação global da câmera, vários objetos se movimentando independentemente, rotação de alguns objetos, bastante textura.
Mother & Daughter	QCIF (176×144)	Câmera parada, pouca movimentação, sem cortes, pessoa falando como em uma videoconferência.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ITU-T Recommendation H.262 | ISO/IEC 13818-2 (MPEG-2), *Information Technology – Generic Coding of Moving Pictures and Associated Audio Information – Part 2: Video*, Int’l Telecomm. Union | Int’l Organization for Standardization, Geneva, Switzerland, 1995.
- [2] C. Fogg, MPEG-2 FAQ: Questions that Should Be Frequently Asked About MPEG. <http://bmrc.berkeley.edu/frame/research/mpeg/mpeg2faq.html> (current Feb., 2007)
- [3] ITU-T Recommendation H.264 | ISO/IEC 14496-10:2005 (MPEG-4), *Advanced Video Coding for Generic Audiovisual Services*, Int’l Telecomm. Union | Int’l Organization for Standardization, Geneva, Switzerland, 2005.
- [4] N. Kamaci and Y. Altunbasak, “Performance Comparison of the Emerging H.264 Video Coding Standard with the Existing Standards,” *IEEE Int’l Conf. on Multimedia and Expo.*, 5-9 July 2003, vol. 1, pp. 345-348.
- [5] J. Golston and A. Rao, Video Compression: System Trade-Offs with H.264, VC-1 and Other Advanced CODECs, white paper, Texas Instruments, Aug. 2006.
- [6] D. Marpe, T. Wiegand and S. Gordon, “H.264/MPEG4-AVC fidelity range extensions: tools, profiles, performance, and application areas,” *IEEE Int’l Conf. on Image Processing*, 11-14 Sept. 2005, pp. 593-596.
- [7] Sistema Brasileiro de TV Digital, Recomendações para o Modelo de Referência (SW) Codificador e Decodificador de Vídeo H.264/AVC, Brazil, 2006.
- [8] ITU-T Recommendation H.263, *Video Coding for Low Bit Rate Communication*, Int’l Telecomm. Union, Geneva, Switzerland, 1998.

-
- [9] I.E.G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*. John Wiley & Sons, West Sussex: 2003.
- [10] T. Wiegand et al., “Overview of the H.264/AVC Video Coding Standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560-576, July 2003.
- [11] G.B. Tandberg, “Video Compression after H.264 (Some personal reflections),” slide presentation, IMTC Forum, Eibsee, Germany, May 2005.
- [12] R. Reddy, “Advances in Video Compression Standards: H.265 (Personal Reflections),” slide presentation, 8th Texas Instruments Developer Conference India, Bangalore, India, Nov. 2005. <http://tii.developerconference.ext.ti.com/downloads/ratna-reddy.pdf> (current Dec. 2006)
- [13] ITU-R Recommendation BT.601-6, *Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios*, Int’l Telecomm. Union, Geneva, Switzerland, 2007.
- [14] K.R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, Boston: 1990.
- [15] H. Malvar, A. Hallapuro, M. Karczewicz et al., “Low-Complexity Transform and Quantization in H.264/AVC,” *IEEE Trans. Circuits Syst. Video Technol.*, v. 13, p. 637-644, Jul. 2003.
- [16] Aravind Al, B.P. Rao, S.S. Kudva et al., “Quality and Complexity Comparison of H.264 Intra Mode with JPEG2000 and JPEG,” *IEEE Int’l Conf. on Image Processing*, 24-27 Oct. 2004, pp. 525- 528.
- [17] T. Halbach and M. Wien, “Concepts and performances of next generation video compression standardization,” *Proc. Nordic Sig. Proc. Symp.*, Oct. 2002.
- [18] D. Marpe, V. George, H.L. Cycon et al., “Performance Evaluation of Motion-JPEG2000 in Comparison With H.264/AVC Operated in Pure Intra Coding Mode,” *SPIE Conf. Wavelet Application in Industrial Processing*, Feb. 2004, pp. 128-137.

-
- [19] P. List, A. Joch, J. Lainema et al., “Adaptive Deblocking Filter,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 614-619, Jul. 2003.
- [20] G.J. Sullivan and T. Wiegand, “Video Compression – From Concepts to the H.264/AVC Standard,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 18-31, Jan. 2005.
- [21] ITU-T & ISO/IEC JTC 1. *Generic Coding of Moving Pictures and Associated Audio Information – Part 1: Systems*. ITU-T Rec. H.222.0 & ISO/IEC 13818-1 (MPEG-2 Systems), Nov. 1994.
- [22] ITU-T. *Narrow-band Visual Telephone Systems and Terminal Equipment*. ITU-T Rec. H.320, 1999.
- [23] ITU-T. *Packet-based multimedia communications systems*. ITU-T Rec. H.323, 1998.
- [24] ITU-T. *Terminal for low bit rate multimedia communication*. ITU-T Rec. H.324, 1996.
- [25] J. Rosenberg, H. Schulzrinne, G. Camarillo et al. *SIP: Session Initiation Protocol*. IETF RFC 3261, 2002.
- [26] H. Schulzrinne, S. Casner, R. Frederick et al. *RTP: A transport protocol for real-time applications*. IETF RFC 3261, 2002.
- [27] S. Wenger, “H.264/AVC over IP,” *IEEE Trans. Circuits Syst. Video Technol.*, v. 13, p. 645-656, Jul. 2003.
- [28] C.E. Shannon, “Communications in the Presence of Noise,” *Proc. IEEE*, vol. 86, no. 2, pp. 447-457, Feb. 1998.
- [29] J. Rissanen and G.G. Langdon Jr, “Universal modeling and coding,” *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 12-23, Jan. 1981.

-
- [30] D. Marpe, H. Schwatz and T. Wiegand, “Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard,” *IEEE Trans. Circuits Syst. Video Technol.*, v. 13, p. 620-636, Jul. 2003.
- [31] T. Wiegand, H. Schwarz A. Joch et al. Rate-Constrained Coder Control and Comparison of Video Coding Standards. *IEEE Trans. Circuits Syst. Video Technol.*, v. 13, p. 688-703, Jul. 2003.
- [32] A. Ortega; K. Ramchandran, “Rate-distortion methods for image and video compression,” *IEEE Signal Processing Mag.*, vol. 15, no. 6, pp. 23–50, Nov. 1998.
- [33] G.J. Sullivan and T. Wiegand, “Rate-distortion optimization for video compression,” *IEEE Signal Processing Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.
- [34] M. Moecke and R. Seara, “Rate-distortion optimized video coding with stopping rules: quality and complexity,” *IEEE Int’l Conf. on Image Processing*, 24-27 Oct. 2004, pp. 753–756.
- [35] ITU-T & ISO/IEC JVT, “H.264/AVC Reference Software Encoder”, <http://iphome.hhi.de/suehring/tml/download/> (current Dec., 2006).
- [36] ITU-T Video Coding Experts Group (ITU-T SG16 Q.6), “Tree-structured macroblock partition”, Document: VCEG-O17r3, 2004. http://ftp3.itu.ch/av-arch/jvt-site/2001_12_Pattaya/VCEG-O17r3.doc (current Dec., 2006)
- [37] ITU-T Video Coding Experts Group (ITU-T SG16 Q.6), “Some Potential Enhancements of H.264/AVC”, Document: VCEG-X03r1, 2004. http://ftp3.itu.ch/av-arch/video-site/0410_Pal/VCEG-X03r1.doc (current Dec., 2006)