

**MARCOS MOECKE**

**CODIFICAÇÃO DE VÍDEO: PRIORIZAÇÃO DO MENOR  
CUSTO DE CODIFICAÇÃO NA OTIMIZAÇÃO EM  
TAXA-DISTORÇÃO**

**FLORIANÓPOLIS**

**2009**



**UNIVERSIDADE FEDERAL DE SANTA CATARINA**

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

**CODIFICAÇÃO DE VÍDEO: PRIORIZAÇÃO DO MENOR  
CUSTO DE CODIFICAÇÃO NA OTIMIZAÇÃO EM  
TAXA-DISTORÇÃO**

Tese submetida à  
Universidade Federal de Santa Catarina  
como parte dos requisitos para a obtenção do grau de  
Doutor em Engenharia Elétrica.

**MARCOS MOECKE**

Florianópolis, Novembro de 2009

Catálogo na fonte pela Biblioteca Universitária da  
Universidade Federal de Santa Catarina

M693p Moecke, Marcos

Priorização do menor custo de codificação na  
otimização em taxa distorção [tese] / Marcos  
Moecke ; orientador, Rui Seara. - Florianópolis,  
SC 2009.

213 f.: il., grafs., tabs.

Tese (doutorado) - Universidade Federal de  
Santa Catarina. Centro Tecnológico. Programa de  
Pós-Graduação em Engenharia Elétrica.

Inclui referências

1. Engenharia elétrica. 2. Processamento  
eletrônico de dados. 3. Algoritmos rápidos.  
4. Estimativa de movimento. 5. Otimização taxa  
distorção. 6. Modo de codificação. I. Seara, Rui.  
II. Universidade Federal de Santa Catarina.  
Programa de Pós-Graduação em Engenharia Elétrica.  
III. Título.

CDU 621.3

## **Codificação de Vídeo: Priorização do Menor Custo de Codificação na Otimização em Taxa-Distorção**

Marcos Moecke

‘Esta Tese foi julgada adequada para obtenção do Título de Doutor em Engenharia Elétrica, Área de Concentração em *Comunicações e Processamento 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. Roberto de Souza Salgado, Ph.D.  
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

---

Prof. Rui Seara, Dr.  
Presidente

---

Eng. Daniel Távora de Queiroz Cobra, Ph.D.

---

Prof. Eduardo Antônio Barros da Silva, Ph.D.

---

Prof. Max Henrique Machado Costa, Ph.D.

---

Prof. Orlando José Tobias, Dr.

---

Prof. Sidnei Noceti Filho, Dr.



Para  
Elisa, Filipe e Martha  
pelo amor,  
carinho,  
paciência,  
compreensão,  
apoio,  
pois tudo devo avocês.





## Agradecimentos

Apesar de um trabalho de tese ser um trabalho solitário, na vida nada se conquista sozinho, precisamos uns dos outros para alcançar o êxito. Muitas vezes, um simples gesto contribui para o nosso sucesso. Depois de quase uma década mergulhado neste trabalho gostaria de agradecer por diferentes razões a algumas pessoas e instituições.

Primeiro, agradeço: ao professor Rui Seara pela orientação e pelas leituras detalhadas das diversas versões desta tese; aos professores da(s) banca(s) do(s) exame(s) de qualificação, pelas ricas discussões e muitas sugestões que deram o azimute do trabalho; aos membros da banca examinadora da tese que pelo exame minucioso do trabalho enriqueceram-no com suas correções e sugestões; aos anônimos revisores dos artigos publicados decorrentes da tese, pois seus questionamentos ajudaram em alguns momentos a esclarecer pontos escuros e desvelar um horizonte que está coberto.

Também não posso esquecer-me dos colegas do LINSE/UFSC, especialmente o Elton, que sempre estiveram ali disponíveis nos momentos que deles precisei. Os colegas do IF-SC, e aqui não vou citar nomes, pois são muitos, pelo companheirismo e compreensão durante esse longo período. A direção do IF-SC, em suas várias instâncias, agradeço por permitir que me afastasse por muitos períodos para trabalhar na tese e concretizar este trabalho.

Aos muitos amigos e amigas (não vou citar nomes para não esquecer nenhum) que permaneceram ao meu lado, dando palavras de conforto, encorajamento, me apoiando e estando ao meu lado curtindo eventos, festas, férias, praias, trilhas, jogos e caminhadas, também o meu especial muito obrigado.

Essa tese não seria possível sem o apoio da minha família, especialmente a minha querida e amada esposa Elisa e meus filhos Filipe e Martha. Por isso, a eles dedico este trabalho, e digo “muito obrigado” pelo amor, carinho, compreensão, encorajamento e apoio que estiveram sempre presentes, tornando mais fácil suportar pacientemente o longo tempo de trabalho e vencer essa maratona.

Finalmente e principalmente agradeço a Deus pela vida, pela sabedoria, pelo amor, por sua presença, por me capacitar, por me guiar e manter, sem Ele nada disso seria possível.



Resumo da Tese apresentada à UFSC como parte dos requisitos necessários para a obtenção do grau de Doutor em Engenharia Elétrica.

## **Codificação de Vídeo: Priorização do Menor Custo de Codificação na Otimização em Taxa-Distorção**

**Marcos Moecke**

Novembro/2009

*Orientador:* Prof. Rui Seara, Dr.

*Área de Concentração:* Comunicações e Processamento de Sinais.

*Palavras-chave:* Algoritmos rápidos, estimação de movimento, vídeo em baixa taxa de bits, complexidade computacional, otimização taxa-distorção, modos de codificação.

*Número de Páginas:* 213.

*Resumo:* O presente trabalho propõe duas novas estratégias para compressão de sinais de vídeo através de algoritmos otimizados em taxa-distorção (RD), focando aplicações típicas de vídeo digital para operação em baixas taxas de bits. As estratégias propostas são implementadas em um codificador de vídeo baseado no padrão H.264, o qual apresenta uma alta complexidade computacional devido principalmente ao grande número de modos de codificação disponível. São apresentadas duas propostas de redução da complexidade, mantendo o desempenho RD próximo àquele do codificador H.264 otimizado em RD usando busca exaustiva. A primeira proposta (denominada *rate sorting and truncation* - RST) realiza o ordenamento tanto dos vetores de movimento (VMs) quanto dos modos de codificação em ordem ascendente de taxa de bits. O processo de codificação é interrompido quando a taxa de bits dos novos VMs e modos de codificação exceder à menor taxa já obtida para um pré-estabelecido nível de qualidade de imagem. Assim, um grande número de VMs e diversos modos de codificação são descartados antes que sejam avaliados. A segunda proposta consiste em um algoritmo rápido, baseado no perfil de distribuição de vetores do codificador H.264, para estimação de movimento (denominado *logarithmic diamond shape search* - LDSS). O uso da estratégia RST associada ao algoritmo LDSS reduz até 98% a carga computacional com perda marginal de desempenho RD.



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

## **Video Coding: Priority to Lower Coding Cost in Rate-Distortion Optimized**

**Marcos Moecke**

Novembro /2009

*Advisor:* Prof. Rui Seara, Dr.

*Area of Concentration:* Communications and Signal Processing.

*Keywords:* Fast algorithms, motion estimation, low bit-rate video coding, computational complexity, rate-distortion optimization, mode selection.

*Number of Pages:* 213.

*Abstract:* This research work proposes two new video compression strategies, aiming at typical low bit rate video applications using rate-distortion (RD) optimized algorithms. The proposed strategies are implemented on an H.264 video encoder, which has high computational complexity due mainly to the large number of coding modes available. Two approaches are presented for reducing the encoder computational complexity, maintaining the RD performance close to the full search RD optimized H.264 encoder. The first approach (termed rate sorting and truncation - RST) is based on sorting the motion vectors (MVs) and coding modes in an ascending rate order. This sorting and encoding process, which is stopped when the rate value exceeds the previous best rate for a required image quality level, allows the elimination of MVs and coding modes before checking their distortion. Apart from obtaining a significant complexity reduction, the process still remains optimized in RD sense. The second approach is an algorithm termed logarithmic diamond shape search (LDSS), which explores the MVs distribution profile for the RD optimized H.264 encoder. The use of the RST strategy associated with LDSS algorithm yields up to a 98% reduction in the computational burden, with insignificant RD performance loss.



## Lista de Figuras

Figura 1 - Implementação da DWT através de banco de filtros. _____	49
Figura 2 - Direções de predição para blocos $4 \times 4$ do H.264. _____	52
Figura 3 - Modo 3 de predição de pixels para blocos $4 \times 4$ do H.264. _	53
Figura 4 - Sequência de vídeo <i>Suzie</i> . (a) Componente de luminância do quadro #21. (b) Componente de luminância do quadro #22. (c) Diferença temporal entre os componentes de luminância dos quadros #21 e #22. _____	54
Figura 5 - Diagrama em blocos de um codificador baseado em modelos. _____	56
Figura 6 - FCCs do observador padrão: CIE 1951 para definição de 2 graus. _____	57
Figura 7 - Representação de imagem no espaço de cores RGB. (a) Vermelho, (b) Verde. (c) Azul. _____	58
Figura 8 - Representação de imagem no espaço de cores $Y C_B C_R$ . (a) Luminância $Y$ . (b) Crominância $C_B$ . (c) Crominância $C_R$ . _____	59
Figura 9 - Representação de imagem no espaço de cores $L^*a^*b^*$ . _____	60
Figura 10 - PSNR para CM com blocos de $16 \times 16$ , $8 \times 8$ , $4 \times 4$ e $2 \times 2$ . _____	69
Figura 11 - Imagens da DFD do quadro #96 para CM com blocos de $16 \times 16$ , $8 \times 8$ , $4 \times 4$ e $2 \times 2$ . _____	70
Figura 12 - Entropia média dos VMs para CM com blocos de $16 \times 16$ , $8 \times 8$ , $4 \times 4$ e $2 \times 2$ . _____	70
Figura 13 - PSNR para CM com janelas de procura de $\pm 2$ , $\pm 4$ , $\pm 7$ e $\pm 15$ pixels. _____	72
Figura 14 - Imagem da DFD do quadro #52, com janelas de procura de $\pm 2$ , $\pm 4$ , $\pm 7$ e $\pm 15$ pixels. _____	72
Figura 15 - PSNR do componente Y para CM com precisão de 1, $1/2$ e $1/4$ pixel, com interpolação bilinear (BL) e interpolação bicúbica (BC). _____	74
Figura 16 - Ganho de PSNR para CM com blocos $8 \times 8$ e precisão de $1/4$ pixel em relação a blocos $2 \times 2$ e precisão de 1 pixel, com interpolação bilinear (BL) e interpolação bicúbica (BC). _____	74
Figura 17 - Perda em qualidade das estratégias rápidas NTSS, BBGDS e DSS em relação ao algoritmo FS-DO. _____	77

- Figura 18 - Curvas taxa-distorção para a sequência *Mother&Daughter*. Valores obtidos com codificador JM7.3 usando somente o modo  $Inter_{8 \times 8}$  variando o parâmetro de quantização  $Q$  entre 24 e 36. \_\_ 85
- Figura 19 - Comparação de desempenho quadro-a-quadro das estratégias DO e RDO integradas ao codificador JM7.3 usando somente o modo  $Inter_{8 \times 8}$  e  $Q = 32$ . (Linha pontilhada) valores de PSNR (a) e taxa de bits (b) obtidos sem codificação da DFD. (Linha contínua) valores obtidos incluindo a codificação da DFD.86
- Figura 20 - Conjunto de vetores de movimento obtidos com o codificador JM7.3 usando apenas o modo  $Inter_{8 \times 8}$  para o quadro #15 da sequência *Mother&Daughter*. (a) Otimização em distorção. (b) Otimização em RD. \_\_\_\_\_ 87
- Figura 21 - Interpretação gráfica do processo de otimização RD, habilitando apenas o modo  $Inter_{16 \times 16}$  do codificador JM7.3. Os pontos mostrados correspondem aos VMs com resolução de 1 *pixel* (+), 1/2 *pixel* (o) e 1/4 de *pixel* (x). \_\_\_\_\_ 90
- Figura 22 - Interpretação gráfica do processo de otimização RD da EM com resolução de 1 *pixel* para diferentes valores de  $\lambda$ , habilitando apenas o modo  $Inter_{16 \times 16}$  do codificador JM7.3. \_\_\_\_\_ 91
- Figura 23 - Comparação quadro-a-quadro dos codificadores de vídeo hipotético CoderA e CoderB. (a) PSNR. (b) Taxa de bits. \_\_\_\_ 94
- Figura 24 - Comparação quadro-a-quadro da diferença de desempenho do codificador hipotético CoderB com relação ao CoderA. (Linha contínua) Medida de distorção LPSNR. (Linha pontilhada) Medida da distorção em PSNR. \_\_\_\_\_ 95
- Figura 25 - Comparação da diferença de desempenho quadro-a-quadro das estratégias DO e RDO usando as medidas PSNR e LPSNR. \_95
- Figura 26 - Representação gráfica da otimização RD para os modos de codificação do H.264 usando o codificador JM7.3-RDO com  $Q = 30$ . Quadro #15 da sequência *Mother&Daughter*. (a) MB #31 – modo ótimo  $Inter_{8 \times 8}$ . (b) MB #69 – modo ótimo  $Intra_{4 \times 4}$ . (c) MB #78 – modo ótimo  $Inter_{16 \times 16}$ . (d) MB #98 – modo ótimo *Skip*. \_\_\_\_\_ 99
- Figura 27 - Predição do VM a partir dos blocos vizinhos causais ( $b_A$ ,  $b_B$ ,  $b_C$ ,  $b_D$ ). \_\_\_\_\_ 106
- Figura 28 - Recíproco da taxa dos DVMs para codificação VLC. \_\_\_\_ 108



Figura 29 - Distribuição bidimensional média dos DVMs obtidos com o codificador JM7.3. (a) e (b) Otimização em distorção. (c) e (d) Otimização em taxa-distorção. (a) e (c) Quantização fina – alta taxa de bits, $Q = 15$ . (b) e (d) Quantização grosseira – baixa taxa de bits, $Q = 45$ . _____	109
Figura 30 - Particionamento dos macroblocos no H.264. (a) MB $16 \times 16$ . (b) subMB $8 \times 8$ pixels. _____	112
Figura 31 - Exemplos de MB-Skip (marcados com S) usando o codificador H.264 RDO FS. (a) Quadro #210 da sequência <i>Mother&amp;Daughter</i> com $Q = 30$ . (b) Quadro #237 da sequência <i>Foreman</i> com $Q = 28$ . _____	118
Figura 32 - Curva RD para o codificador H.264 RDO FS usando memória de longo prazo com 5 referências temporais para os 10 segundos iniciais da sequência <i>Foreman</i> . (Linha contínua) JM7.3 com o modo <i>Intra</i> habilitado. (Linha pontilhada) com modo <i>Intra</i> desabilitado. _____	124
Figura 33 - Exemplos de MBs codificados no modo <i>Intra</i> (marcados com I ou x) usando o codificador H.264 RDO FS. (a) Quadro #210 da sequência <i>Foreman</i> . (b) Quadro #177 da sequência <i>Carphone</i> . _____	126
Figura 34 - Curvas dos modelos de custo RD inicial <i>Intra</i> $J_1$ para diferentes níveis $k_i$ e valores experimentais obtidos usando o codificador H.264 RDO FS. (Linha cinza grossa) melhor ajuste do modelo de potência. _____	128
Figura 35 - Efeito do limiar de distorção na complexidade da EM. Para sequências de vídeo (a) <i>Akiyo</i> , (b) <i>News</i> , (c) <i>Mother&amp;Daughter</i> e (d) <i>Carphone</i> . _____	132
Figura 36 - Redução de complexidade para a estratégia RST usando a pré-seleção do modo <i>Skip</i> . (a) Vídeos adquiridos com câmera fixa. (b) Vídeos adquiridos com câmera móvel. _____	134
Figura 37 - Desempenho RD do codificador proposto com ST1, adotando a pré-eliminação de alguns modos <i>Intra</i> , para a sequência <i>Foreman</i> . _____	136
Figura 38 - Desempenho RD do codificador proposto com ST1, adotando a pré-eliminação de alguns modos <i>Intra</i> e das previsões de croma, para a sequência <i>Foreman</i> . _____	137
Figura 39 - Desempenho RD do codificador H.264 RDO FS e do codificador FullRST para a sequência <i>Mother&amp;Daughter</i> . _____	139

Figura 40 - Percentagem de estimações de movimento realizadas com tamanho de bloco $16 \times 16$ , $8 \times 8$ e $4 \times 4$ <i>pixels</i> para o codificador FullRST. (a) Sequência <i>Akiyo</i> . (b) Sequência <i>Mother&amp;Daughter</i> ._____	142
Figura 41 - Exemplo ilustrando as posições avaliadas para uma janela de procura de $\pm 8$ <i>pixels</i> e resolução 1 <i>pixel</i> . (a) e (c) Algoritmo TSS. (b) e (d) Algoritmo NTSS._____	150
Figura 42 - Exemplos ilustrando as posições avaliadas com o algoritmo DSS para uma janela de procura de $\pm 8$ <i>pixels</i> e resolução de 1 <i>pixel</i> . (a) VM = (+6, +2). (b) VM = (-1, +1)._____	152
Figura 43 - Desempenho RD e complexidade computacional das estratégias FS (JM7.3-w16), TSS, NTSS e DSS, com janela de procura de $\pm 16$ <i>pixels</i> e FS com janela de procura de $\pm 2$ <i>pixels</i> (JM7.3-w2). (a) Percentual de NPP relativo ao JM7.3-w16 para a sequência <i>Foreman</i> . (b) Curvas RD. _____	154
Figura 44 - Perda de desempenho RD medida em LPSNR das estratégias TSS, NTSS e DSS com relação ao JM7.3-RDO, para a sequência <i>Foreman</i> com $Q = 32$ ._____	155
Figura 45 - Variação do número de MBs <i>Intra</i> para as estratégias TSS, NTSS e DSS em relação ao JM7.3-RDO, para a sequência <i>Foreman</i> -10Hz com $Q = 32$ . _____	157
Figura 46 - Gabarito de procura inicial da estratégia DSS. _____	158
Figura 47 - Gabarito de procura inicial das estratégias: (a) TSS. (b) NTSS. _____	159
Figura 48 - Gabaritos de procura inicial da estratégia LDSS com 9 posições (a) LDSS <sub>1-2</sub> – GPI, (b) LDSS <sub>1-4</sub> – GPI, (c) LDSS <sub>1-8</sub> – GPI e com 17 posições (d) LDSS <sub>1-2-4-8</sub> – GPI . _____	162
Figura 49 - Ilustração de posições com resolução 1 <i>pixel</i> , avaliadas pelo algoritmo LDSS para uma janela de procura de $\pm 16$ <i>pixels</i> , com diversos GPIs. (a) e (b) LDSS <sub>1-2-4</sub> . (c) e (d) LDSS <sub>1-2</sub> . _____	164
Figura 50 - Desempenho RD do algoritmo FS (JM7.3-w16 e JM7.3-w2) e TSS, NTSS, DSS e LDSS com janela de procura de $\pm 16$ <i>pixels</i> , para a sequência <i>Foreman</i> ._____	167
Figura 51 - Redução de complexidade computacional dos algoritmos JM7.3-w2, TSS, NTSS, DSS e LDSS em relação ao JM7.3-w16. Resultados para a sequência <i>Foreman</i> com janela de procura de $\pm 16$ <i>pixels</i> . _____	167

Figura 52 - Desempenho RD do algoritmo FS (JM7.3-w16 e JM7.3-w2) e dos ARs DSS, LDSS <sub>1-8</sub> e LDSS-RST <sub>1-8</sub> com janela de procura de $\pm 16$ pixels, para a sequência <i>Foreman</i> . _____	168
Figura 53 - Redução de complexidade computacional dos algoritmos JM7.3-w2, TSS, NTSS, DSS e LDSS-RST com relação ao JM7.3-w16. Resultados para a sequência <i>Foreman</i> com janela de procura de $\pm 16$ pixels. _____	169
Figura 54 - Perda de desempenho RD medida em LPSNR das estratégias TSS, NTSS, DSS e LDSS-RST <sub>1-8</sub> com relação ao JM7.3-RDO, para a sequência <i>Foreman</i> com $Q = 32$ . _____	170
Figura 55 - Quadro inicial das sequências de vídeo em formato QCIF. _____	181
Figura 56 - Quadro #100 das sequências de vídeo em formato QCIF. _____	182
Figura 57 - Quadro #300 das sequências de vídeo em formato QCIF. _____	183
Figura 58 - Quadro inicial das sequências de vídeo em formato CIF. _____	184
Figura 59 - Quadro #300 das sequências de vídeo em formato CIF. _____	185
Figura 60 - Desempenho RD do codificador H.264 RDO FS e do codificador FullRST, sequência <i>Akiyo</i> . _____	195
Figura 61 - Desempenho RD do codificador H.264 RDO FS e do codificador FullRST, sequência <i>News</i> . _____	195
Figura 62 - Desempenho RD do codificador H.264 RDO FS e do codificador FullRST, sequência <i>Foreman</i> . _____	196
Figura 63 - Desempenho RD do codificador H.264 RDO FS e do codificador FullRST, sequência <i>Carphone</i> . _____	196
Figura 64 - Desempenho RD do codificador H.264 RDO FS e do codificador FullRST, sequência <i>Coastguard</i> . _____	197
Figura 65 - Desempenho RD do codificador H.264 RDO FS e do codificador FullRST, sequência <i>HallObjects</i> . _____	197



## Lista de Tabelas

Tabela 1 – Formatos padrões de sinais de vídeo digital _____	41
Tabela 2 – Coeficientes dos filtros Daubechies biortogonal 9/7 _____	50
Tabela 3 – Coeficientes dos filtros Le Gall 5/3 _____	50
Tabela 4 – Entropia média por <i>pixel</i> dos VMs e qualidade em PSNR __	75
Tabela 5 – Valores médios de PSNR e variação de PSNR em relação ao algoritmo FS-DO _____	76
Tabela 6 – Número médio de posições de procura por bloco e percentual médio de redução de complexidade em relação ao algoritmo FS-DO _____	78
Tabela 7 – Taxa de bits para a codificação dos VMs e percentagem de redução da entropia em relação ao FS-DO _____	78
Tabela 8 – Comparação da qualidade em PSNR da CM dos componentes de crominâncias $C_B$ e $C_R$ usando as abordagens $CM_1$ e $CM_2$ _____	80
Tabela 9 – Padrão de codificação da diferença dos VMs usando VLC do tipo <i>Exp-Golomb</i> _____	107
Tabela 10 – Percentagem de MBs em modo <i>Skip</i> para o codificador JM7.3-RDO _____	117
Tabela 11 – Bits obrigatórios na codificação MB- <i>Inter</i> para as partições segundo o padrão H.263++ _____	121
Tabela 12 – Bits obrigatórios na codificação MB- <i>Inter</i> para as partições segundo o padrão H.264 _____	122
Tabela 13 – Percentagem de MB codificados em modo <i>Intra</i> no codificador H.264 RDO FS _____	123
Tabela 14 – Percentagem de tempo de processamento despendido na seleção dos modos <i>Intra</i> e <i>Inter</i> usando um codificador H.264 RDO FS com memória de curto prazo (referência temporal única)_____	123
Tabela 15 – Percentagem de tempo de processamento despendido na seleção dos modos <i>Intra</i> e <i>Inter</i> usando um codificador H.264 RDO FS com memória de longo prazo (5 referências temporais)_____	124
Tabela 16 – Percentagem de redução de complexidade do processo EM com <i>Regra 1</i> _____	135
Tabela 17 – PSNR e $\Delta$ PSNR(dB) para o codificador FullRST (sem a <i>Regra 2</i> )_____	135

Tabela 18 – Percentagem média de redução na complexidade total usando a pré-eliminação do modo <i>Intra</i> para a sequência <i>Foreman</i> com taxas de bits de 50 até 130 kbps _____	137
Tabela 19 – Percentagem média de redução na complexidade total usando a pré-eliminação do modo <i>Intra</i> e a pré-eliminação da predição da cromaticidade para a sequência <i>Foreman</i> com taxas de bits de 50 até 130 kbps. _____	138
Tabela 20 – PSNR e $\Delta$ PSNR (dB) para o codificador FullRST (com a <i>Regra 2</i> ) _____	141
Tabela 21 – Taxa de bits (kbps) e $\Delta$ taxa de bits (%) para o codificador FullRST _____	141
Tabela 22 – Percentual médio de redução de complexidade da seleção do modo <i>Inter</i> e <i>Intra</i> do codificador FullRST em relação o FS com LT5 _____	143
Tabela 23 – Percentagem de redução de complexidade total para o codificador FullRST _____	144
Tabela 24 – Percentagem de redução de complexidade do processo EM com o codificador FullRST _____	144
Tabela 25 – Percentagem de redução de complexidade do cálculo do modo <i>Intra</i> para o codificador FullRST _____	145
Tabela 26 – Percentagem de MBs codificados no modo <i>Intra</i> com JM7.3-w16, JM7.3-w2, e com os ARs TSS, NTSS e DSS para $Q = 32$ , janela de procura de $\pm 16$ pixels e resolução de $1/4$ pixel _____	157
Tabela 27 – Percentagem de MB <i>Intra</i> para as estratégias FS, DSS, LDSS <sub>1-8</sub> e LDSS-RST <sub>1-8</sub> , obtidos com $Q = 32$ _____	169
Tabela 28 – Características das sequências de vídeo consideradas _____	180
Tabela 29 – Resultados de taxa-distorção e complexidade medida em NPP para a predição <i>Inter</i> e número de predições <i>Intra</i> . Sequência <i>Akiyo</i> -10Hz, formato QCIF _____	188
Tabela 30 – Resultados de taxa-distorção e complexidade medida em NPP para a predição <i>Inter</i> e número de predições <i>Intra</i> . Sequência <i>News</i> -10Hz, formato QCIF _____	189
Tabela 31 – Resultados de taxa-distorção e complexidade medida em NPP para a predição <i>Inter</i> e número de predições <i>Intra</i> . Sequência <i>Foreman</i> -10Hz, formato QCIF _____	190

Tabela 32 – Resultados de taxa-distorção e complexidade medida em NPP para a predição <i>Inter</i> e número de predições <i>Intra</i> . Sequência <i>Mother&amp;Daughter</i> -10Hz, formato QCIF _____	191
Tabela 33 – Resultados de taxa-distorção e complexidade medida em NPP para a predição <i>Inter</i> e número de predições <i>Intra</i> . Sequência <i>Carphone</i> -10Hz, formato QCIF _____	192
Tabela 34 – Resultados de taxa-distorção e complexidade medida em NPP para a predição <i>Inter</i> e número de predições <i>Intra</i> . Sequência <i>Coastguard</i> -10Hz, formato QCIF _____	193
Tabela 35 – Resultados de taxa-distorção e complexidade medida em NPP para a predição <i>Inter</i> e número de predições <i>Intra</i> . Sequência <i>HallObjects</i> -10Hz, formato QCIF _____	194





## Lista de Símbolos e Abreviaturas

$D$	distorção
$D_L$	limiar de distorção da qualidade
$J$	custo RD (custo lagrangeano)
$k_1$	nível de prioridade do modo <i>Inter</i>
$\lambda$	multiplicador de Lagrange
$m$	modo de codificação
$p$	partição do MB
$Q$	parâmetro de quantização
$R$	taxa de bits
4SS	busca em quatro passos ( <i>four step search</i> )
AN	algoritmo nulo sem compensação de movimento
AR(s)	algoritmo(s) rápido(s)
BBGDS	<i>block-based gradient descent search</i>
BC	interpolação bicúbica
BL	interpolação bilinear
BMA	algoritmo de casamento por blocos ( <i>block matching algorithm</i> )
CABAC	codificação aritmética binária adaptativa ao contexto ( <i>context-adaptive binary arithmetic coding</i> )
CAVLC	codificação de comprimento variável adaptativa ao contexto ( <i>context-adaptive variable length coding</i> )
CIE	<i>Commission Internationale de L'eclairage</i>
CIF	formato comum intermediário ( <i>common intermediate format</i> )
CM	compensação de movimento
CMYK	espaço de cores - ciano, magenta, amarelo e preto ( <i>cyan, magenta, yellow and black</i> )
dB	decibel
DCT	transformada discreta do cosseno ( <i>discrete cosine transform</i> )
DFD	erro de predição ( <i>displaced frame difference</i> )
DO	otimizado em distorção ( <i>distortion optimized</i> )
DSP	processador de sinais digitais ( <i>digital signal processor</i> )
DSS	procura em formato de diamante ( <i>diamond shape search</i> )
DVMs	diferença de vetores de movimento
DWT	transformada discreta wavelet ( <i>discrete wavelet transform</i> )

EM	estimação de movimento
FCC(s)	função(ões) de casamento de cores
fps	taxa de quadros por segundo ( <i>frames per second</i> )
FS	busca exaustiva ( <i>full search</i> )
FullRST	algoritmo que utiliza simultaneamente todas as estratégias
RST	
GP(s)	gabarito(s) de procura
GPI(s)	gabarito(s) de procura inicial
H	entropia do sinal
H.264	padrão de codificação de vídeo do ITU-T
HD	alta definição ( <i>high definition</i> )
HDTV	televisão de alta definição ( <i>high definition television</i> )
JM7.3	modelo de referência do padrão H.264.
JPEG	<i>Joint Photographic Experts Group</i>
JVT	<i>Joint Video Team</i>
kbps	quilo bits por segundo
L*a*b*	espaço de cores - CIELAB
LDSS	busca logarítmica em formato de diamante ( <i>logarithmic diamond shape search</i> )
LMAD	MAD lagrangeana
LMSD	MSD lagrangeana
LPSNR	razão sinal-ruído de pico lagrangeana
LT5	memória de longo prazo com cinco quadros de referência
MAD	média das diferenças absolutas ( <i>mean absolute differences</i> )
MB(s)	macrobloco(s)
MB-Inter	MB codificado no modo <i>Inter</i>
MB-Intra	MB codificado no modo <i>Intra</i>
MB-Skip	MB codificado no modo <i>Skip</i>
Mbps	Mega bits por segundo
MPEG	<i>Moving Pictures Experts Group</i>
MSD	média das diferenças quadráticas ( <i>mean squared differences</i> )
NiPP	número inicial de posições de procura
NPP	número de posições de procura
NTSS	nova busca em três passos ( <i>new three step search</i> )
PSNR	razão sinal-ruído de pico ( <i>peak signal-to-noise ratio</i> )
QCIF	um quarto do formato comum intermediário ( <i>quarter common intermediate format</i> )
RD	taxa-distorção ( <i>rate-distortion</i> )

RDO	otimizado em taxa-distorção ( <i>rate-distortion optimized</i> )
RGB	espaço de cores - vermelho, verde e azul ( <i>red, green and blue</i> )
RST	estratégia de ordenamento e truncamento por taxa de bits ( <i>rate sorting and truncation strategy</i> )
SAD	soma das diferenças absolutas ( <i>sum of absolute differences</i> )
SATD	soma das diferenças absolutas transformadas ( <i>sum of absolute transformed differences</i> )
SD	definição padrão ( <i>standard definition</i> )
SSD	soma das diferenças quadráticas ( <i>sum of squared differences</i> )
ST1	memória de curto prazo com referência única
SubMB(s)	submacrobloco(s)
SVH	sistema visual humano
TSS	busca em três passos ( <i>three step search</i> )
UVLC	código universal de comprimento variável ( <i>universal variable length code</i> )
VLBR	taxa muito baixa de bits ( <i>very low bit-rate</i> )
VLC(s)	tabela(s) de código de comprimento variável ( <i>variable length code</i> )
VM(s)	vetor(es) de movimento
YCBCR	espaço de cores - luminância e cromaticâncias de desvio para o azul e desvio para o vermelho ( <i>luma, blue-difference and red-difference</i> )



# Sumário

<b>Capítulo 1 Introdução</b>	<b>33</b>
1.1 Problema Investigado	33
1.2 Abordagem Proposta	34
1.3 Organização do Trabalho	35
1.4 Contribuições deste Trabalho	36
<b>Capítulo 2 Codificação de Vídeo</b>	<b>39</b>
2.1 Aquisição e Representação de Imagens e Vídeos	40
2.2 Necessidade de Compressão do Sinal de Vídeo	40
2.3 Quantização	42
2.3.1 Quantização Escalar	42
2.3.2 Quantização Vetorial	43
2.4 Codificação de Entropia	43
2.4.1 Codificação de Huffman e Tabelas VLCs	44
2.4.2 Codificação Aritmética	44
2.4.3 Codificação <i>Run-Length</i>	45
2.4.4 Codificação Adaptativa ao Contexto	45
2.5 Redundância Espacial	46
2.5.1 Transformada Discreta do Cosseno (DCT)	47
2.5.2 Transformada Inteira	47
2.5.3 Transformada Wavelet Discreta (DWT)	49
2.5.4 Matching Pursuits	51
2.5.5 Predição Espacial	51
2.6 Redundância Temporal	53
2.6.1 Técnicas Baseadas na Forma de Onda	54
2.6.2 Técnicas Baseadas em Objetos	55
2.6.3 Técnicas Baseadas em Modelos	55
2.7 Codificação da Cor	56
2.7.1 Espaços Padronizados pelo CIE	56
2.7.2 Espaços de Cores Usados na Codificação	57
2.7.3 A Cor nos Padrões de Codificação de Vídeo	60
2.8 Conclusões	61
<b>Capítulo 3 Estimação de Movimento</b>	<b>63</b>
3.1 Técnicas de Estimação de Movimento	64
3.2 Estimação de Movimento em Blocos	65
3.3 Funções de Distorção	66
3.4 Parâmetros que Afetam a Qualidade do BMA	68
3.4.1 Tamanho do Bloco	68

3.4.2	Tamanho da Janela de Procura	71
3.4.3	Precisão de Representação dos VMs e Interpolação	73
3.4.4	Estratégias Rápidas de EM	75
3.5	Compensação de Movimento da Crominância	79
3.6	Conclusões	80
<b>Capítulo 4 Otimização em Taxa-Distorção</b>		<b>83</b>
4.1	Otimização da Escolha dos VMs	84
4.1.1	Formas de Avaliação de Desempenho	84
4.1.2	Otimização Taxa-Distorção dos VMs	88
4.1.3	Interpretação Gráfica da Otimização RD dos VMs	89
4.2	Avaliação de Desempenho Quadro-a-Quadro	92
4.3	Seleção Otimizada do Modo de Codificação	96
4.3.1	Conjunto de Modos de Codificação	96
4.3.2	Interpretação Gráfica do Processo de Otimização do Modo de Codificação	98
4.4	Conclusões	101
<b>Capítulo 5 Codificação por Custo Crescente</b>		<b>103</b>
5.1	Introdução	103
5.2	Efeito da Predição dos VMs na Taxa de Bits	105
5.3	Distribuição Bidimensional dos VMs	107
5.4	Tamanhos de Bloco Usados na EM	111
5.5	Medida de Complexidade Computacional da EM	111
5.6	Estratégia RST na Estimação de Movimento	112
5.6.1	Ordenamento dos VMs por Taxa	112
5.6.2	Regra 1: Truncamento por Taxa	113
5.6.3	Regra 2: Truncamento por Qualidade	115
5.7	Estratégia RST na Seleção do Modo de Codificação	115
5.7.1	Pré-Seleção do Modo <i>Skip</i>	116
5.7.2	Pré-Eliminação de Partições no Modo <i>Inter</i>	119
5.7.3	Pré-Eliminação do Modo <i>Intra</i>	122
5.7.4	Seleção Final dos Modos de Codificação	129
5.8	Resultados Experimentais	130
5.8.1	Estimação de Movimento e Regras de Truncamento	130
5.8.2	Pré-Seleção do Modo <i>Skip</i>	131
5.8.3	Pré-Eliminação do Modo <i>Inter</i>	131
5.8.4	Pré-Eliminação do Modo <i>Intra</i>	135
5.8.5	Resultados Finais para a Abordagem RST	138
5.9	Conclusões	145

<b>Capítulo 6 Algoritmos Rápidos para EM-RDO</b>	<b>147</b>
6.1 Algoritmos Rápidos para EM	147
6.1.1 Algoritmos TSS e NTSS	148
6.1.2 Algoritmo DSS	151
6.1.3 Avaliação dos ARs para Otimização RD	153
6.2 LDSS: Algoritmo Rápido Otimizado em RD	159
6.2.1 Gabaritos de Procura Inicial	160
6.2.2 Procedimento de Busca	161
6.2.3 Resultados	166
6.3 Algoritmo LDSS-RST	168
6.4 Conclusões	171
<b>Capítulo 7 Conclusões e Trabalhos Futuros</b>	<b>173</b>
7.1 Sumário das Contribuições	173
7.2 Sugestões para Trabalhos Futuros	176
<b>Apêndice 1 Sequências de Vídeo Usadas</b>	<b>179</b>
<b>Apêndice 2 Resultados Complementares</b>	<b>187</b>
<b>Apêndice 3 Publicações da Tese</b>	<b>199</b>
<b>Referências Bibliográficas</b>	<b>201</b>





# Capítulo 1

## Introdução

### 1.1 Problema Investigado

Nas últimas três décadas a tecnologia de vídeo digital vem experimentando uma grande evolução começando a tomar um lugar de destaque nas mais diversas áreas de aplicação, nos setores empresarial, doméstico e transporte, através de videoconferência, *streaming* de vídeo, videotelefonia, treinamento e ensino a distância, monitoração de ambientes, jogos de computador, televisão digital, armazenamento digital de vídeos, dentre outras. Para que essa evolução prossiga na área da compressão e transmissão de informação, é necessário que haja o aprimoramento das atuais técnicas e tecnologias, bem como novas soluções sejam desenvolvidas. No que se refere ao processo de codificação de vídeo, um grande obstáculo a ser superado é a complexidade computacional, que tem crescido com a introdução de novas tecnologias. Essa complexidade afeta diretamente o tempo despendido na codificação do sinal de vídeo. O atraso máximo de codificação e decodificação bem como o *hardware* utilizado limitam a carga computacional do codificador.

Para que novas propostas de esquemas de codificação possam ser consideradas, é necessário que elas estejam inseridas nos padrões de âmbito mundial. Atualmente, o estado da arte em codificação de vídeo é representado pelo padrão mais recente de compressão de vídeo, que é o H.264 (ou MPEG-4 parte 10). Tal padrão apresenta um avanço significativo em relação aos padrões anteriores: H.263, MPEG-2 e MPEG-4 parte 2, pois introduziu diversas melhorias em quase todas as etapas do codificador, reduzindo significativamente a taxa de bits em relação aos seus predecessores. Uma das características principais desse padrão de codificação é a disponibilidade de um grande número de modos de codificação, os quais podem ser usados para adaptar o codificador ao conteúdo visual de cada região da imagem. No entanto, para a seleção da melhor forma de codificação e obtenção do desempenho máximo, é necessário considerar um processo de otimização em taxa-distorção (RD) para a escolha do modo de codificação. Tal processo geralmente envolve uma avaliação exaustiva

de todos os modos de codificação, exigindo a determinação da taxa de bits e da distorção para cada modo de codificação, implicando uma complexidade computacional muito elevada. Por isso, urge que haja o desenvolvimento de novas estratégias e algoritmos que reduzam consideravelmente a complexidade dos processos envolvidos, preservando o desempenho do codificador.

## 1.2 Abordagem Proposta

A exploração de redundância temporal através da estimação de movimentos (EM) em blocos (modo *Inter*) é, dentre as etapas que compõem o codificador de vídeo H.264, aquela de maior complexidade computacional como também a responsável por grande parte da melhoria de desempenho RD no referido padrão. Por isso, a EM tem sido alvo de diversos trabalhos de pesquisa visando reduzir sua complexidade. Outrossim, a grande variedade de modos de predição espacial, necessitando serem avaliados nos codificadores H.264, torna também complexa o processo de exploração da redundância espacial através da codificação no modo *Intra*.

O presente trabalho mostra que é viável obter uma alta taxa de compressão usando todos os modos de codificação *Intra* e *Inter*, sem que seja necessário avaliar exaustivamente todos eles.

### a) Proposta 1

Inicialmente apresentamos uma proposta de redução de complexidade computacional mantendo a codificação otimizada em RD. Essa proposta envolve a determinação dos vetores de movimento (VMs), a escolha do melhor tamanho de bloco e a seleção do melhor modo de codificação para cada região da imagem. A ideia básica da estratégia de ordenamento e truncamento por taxa de bits (*rate sorting and truncation* - RST) é ordenar os VMs e modos de codificação em ordem de custo crescente de codificação. Assim, quando nenhum outro VM puder melhorar o desempenho RD já obtido, o processo de EM para os VMs de maior taxa pode ser truncado. Para tal, a cada VM e modo é associado um custo RD inicial mínimo, o qual é utilizado pelas regras de truncamento para eliminar antecipadamente muitos desses modos de codificação, vetores de movimento e tamanhos de bloco, sem uma avaliação da distorção real, mantendo inalterado o desempenho RD.

### *b) Proposta 2*

Baseado na análise do perfil típico de distribuição dos VMs, obtidos com codificadores otimizados em RD, propomos também um algoritmo rápido de EM que pode ser usado em conjunto com a estratégia RST, a qual reduz em até 10 vezes a complexidade total do processo de EM sem perda significativa de desempenho. Dessa forma, é viabilizado o uso de codificadores com grande diversidade de modos de codificação para aplicações que exigem uma complexidade computacional limitada, seja devido ao *hardware* disponível e/ou pelo tempo máximo de codificação admissível.

## **1.3 Organização do Trabalho**

Esta tese é constituída de três capítulos de revisão, incluindo alguns novos desenvolvimentos, e dois capítulos descrevendo as contribuições principais do trabalho de pesquisa.

O Capítulo 2 apresenta as principais técnicas usadas na compressão de vídeo digital. Descreve sucintamente como cada etapa de um codificador de vídeo explora as diversas características do sinal, tais como a redundância espacial, temporal, de código e de codificação de cor. Este capítulo tem por objetivo mostrar que a taxa de compressão alcançada por um codificador depende da configuração adequada destas etapas.

No Capítulo 3, é apresentada a técnica de estimação de movimento pelo casamento de blocos (EM-BMA), usada para a redução da redundância temporal existente entre quadros sucessivos de uma sequência de vídeo. Os parâmetros: tamanho do bloco, janela de procura, precisão na representação de movimentos e algoritmos rápidos de EM são avaliados quanto à sua influência sobre a complexidade e o desempenho.

O Capítulo 4 trata de uma questão fundamental do processo de codificação de vídeo, que é o estabelecimento do compromisso entre os quesitos qualidade e taxa de bits. A abordagem considerada é a otimização em RD através da minimização da função de custo lagrangeana. A avaliação de desempenho RD é vista sob a ótica da análise de valores médios (em curvas taxa-distorção) e também da análise ao longo do tempo com gráficos de razão sinal-ruído de pico lagrangeana (LPSNR) medida quadro-a-quadro.

O Capítulo 5 apresenta a estratégia RST que reduz a complexidade computacional do codificador. Essa estratégia considera as taxas de bits envolvidas nas etapas de escolha dos vetores de

movimento, definição do tamanho do bloco e seleção do modo de codificação para interromper o processo de estimação de movimento e escolha do modo de codificação. A aplicação da estratégia em seqüências de videoconferência resulta em uma redução de complexidade computacional de até 96%.

No Capítulo 6 são examinados algoritmos rápidos que limitam o número de posições de procura, baseados em estratégias que avaliam os vetores de movimento a partir de gabaritos com posições predefinidas. Um algoritmo rápido cujo gabarito de procura é baseado no perfil típico de distribuição dos VMs de codificadores de vídeo otimizados em RD é apresentado e avaliado. Os resultados mostram que, na condição de otimização em RD, o seu desempenho é superior aos demais algoritmos, principalmente nos trechos com maior movimento.

Finalmente, no Capítulo 7, são apresentadas as conclusões gerais como também são sugeridos alguns pontos para continuidade deste trabalho.

#### 1.4 Contribuições deste Trabalho

Este trabalho de pesquisa evidencia que é possível usar de forma eficiente uma abordagem otimizada em RD. Também demonstra que a escolha do modo de codificação e o processo de EM-BMA podem ser realizados sem a necessidade de se recorrer à alta complexidade computacional dos algoritmos de busca exaustiva. As principais contribuições são:

- i) Redução de complexidade do processo de EM sem perda significativa de desempenho RD, através de uma estratégia de ordenamento e truncamento por custos dos VMs com regras de truncamento baseadas no custo e na qualidade da imagem;
- ii) Uso do custo RD inicial mínimo na pré-seleção do modo *Skip* e pré-eliminação de uma grande quantidade de modos de codificação *Inter* e *Intra* ;
- iii) Verificação da pequena influência das diferentes maneiras de prever a cor no modo *Intra*, com vistas ao desempenho RD do codificador H.264;
- iv) Proposta de um algoritmo de busca logarítmica em formato de diamante (LDSS) de EM. O algoritmo explora a dependência entre o custo de codificação dos VMs e o perfil típico de distribuição dos VMs em codificadores otimizados em RD;

- v) Aplicação conjunta do algoritmo LDSS com a estratégia RST, possibilitando uma redução significativa de complexidade do processo de EM, com pequena influência no desempenho RD.



## Capítulo 2

### Codificação de Vídeo

As técnicas de compressão de vídeo digital exploram características e redundâncias do sinal de vídeo. Para uma melhor compreensão de tais técnicas, inicialmente descrevemos como o sinal de vídeo é adquirido, e os formatos utilizados para representá-lo. Em seguida são descritas as etapas constituintes da maioria dos codificadores de vídeo, discutindo em cada etapa as características exploradas do sinal e as diferentes técnicas usadas nos codificadores. O processo de codificação geralmente pode ser decomposto em exploração da redundância temporal, redução da redundância espacial, quantização e codificação de entropia.

Na etapa de quantização ocorrem as principais perdas de informação do processo de compressão [1]. A codificação de entropia proporciona compressão sem perdas de informação explorando a redundância de código. Para a exploração da redundância espacial, diversas técnicas são usadas, dentre as quais o uso de transformações bidimensionais (2-D) que descorrelacionam *pixels*, tais como transformada cosseno discreta, transformada inteira, transformada *wavelet* discreta, e os *matching pursuits*. Nos codificadores de vídeo, essas técnicas são usadas tanto na codificação do quadro inicial como na codificação dos erros de previsão de quadros. Essas etapas são também aplicadas na compressão de imagens estáticas.

A redundância temporal é uma característica exclusiva dos sinais de vídeo, que advêm das variações nos valores dos *pixels* decorrente da aquisição sequencial da imagem de uma cena. Para explorá-la, existem diversas abordagens que podem ser usadas. As principais são as baseadas na compensação de movimento, sendo também consideradas técnicas baseadas na descrição de objetos como também transformações tridimensionais (3-D) nas quais a terceira dimensão é o tempo. A compensação de movimento é geralmente empregada em conjunto com uma transformada espacial 2-D. Como este trabalho é baseado nessa técnica, ela será estudada em detalhes no Capítulo 3.

Finalizando este capítulo, descrevemos sucintamente a forma de representar as cores e as abordagens consideradas para explorar as redundâncias existentes entre os componentes de cor.

## 2.1 Aquisição e Representação de Imagens e Vídeos

A aquisição de uma imagem estática em um suporte adequado para uma posterior observação tem sido realizada desde as primeiras fotografias em preto e branco, tendo evoluído para a aquisição do movimento no cinema e posteriormente para imagem em cores. Nesses processos, a aquisição da imagem é feita através da exposição de películas fotossensíveis que reagem às variações de luminosidade e de cores da cena. Para a reprodução do movimento, sequências de imagens estáticas são captadas e depois reproduzidas na mesma velocidade de aquisição, criando a ilusão de movimento devido ao efeito da persistência visual inerente ao sistema visual humano (SVH) [2].

A representação de uma imagem na forma digital é feita através de uma matriz 2-D de pontos obtidos pela amostragem espacial da imagem. Em sistemas monocromáticos cada ponto da imagem (*pixel*) é representado pela intensidade da luz, sendo cada *pixel* quantizado e representado através de bits. Nos sistemas em cores a intensidade da luz é processada por filtro vermelho, verde e azul, dando origem a três matrizes 2-D [3]. Dependendo da distância de observação, os *pixels* são percebidos como uma imagem contínua devido ao efeito da filtragem passa-baixas espacial efetuada pelo SVH. A capacidade do SVH em distinguir diferenças entre os níveis de luminosidade é limitada, sendo suficiente considerar 256 níveis para a representação da luminância em uma imagem monocromática, e a mesma quantidade de níveis para cada componente de cor em uma imagem colorida. No caso de sinais de vídeo a cena é ainda amostrada na dimensão temporal, obtendo três matrizes 3-D com duas dimensões espaciais e uma dimensão temporal. A taxa de quadros (fps – *frames per second*) pode variar dependendo da aplicação, no entanto, taxas menores do que 60 fps são usadas na maioria das aplicações.

## 2.2 Necessidade de Compressão do Sinal de Vídeo

Um sinal para televisão digital com resolução de  $720 \times 525$  *pixels* amostrado a uma taxa de 30 fps necessita de uma taxa bruta de bits de 173 Mbps (ver ITU-R 601 4:2:2 na Tabela 1). Essa taxa de bits inviabiliza praticamente qualquer aplicação de vídeo digital sem o uso de técnicas de compressão. Conforme o tipo de aplicação, diferentes formatos e taxas de quadros são usados. Para simplificar a comparação e determinação das reais necessidades de compressão dos sinais de vídeo, apresentamos na Tabela 1 as resoluções de bits,



espaciais e temporais, bem como as taxas brutas de bits usadas para diferentes padrões [4].

Tabela 1 – Formatos padrões de sinais de vídeo digital

<b>Formato</b>	<b>Resolução espacial</b>	<b>Bits por pixel<sup>c</sup></b>	<b>Resolução temporal<sup>f</sup></b>	<b>Taxa de bits (bruta)<sup>g</sup></b>
<b>CIF<sup>a</sup></b>	352 × 288	8+2+2	15 fps	18,2 Mbps
<b>QCIF<sup>b</sup></b>	176 × 144	8+2+2	10 fps	3,0 Mbps
<b>BT601<sup>c</sup> -4:2:2</b>	720 × 525	8+4+4	30 fps	173 Mbps
<b>BT601 -4:4:4</b>	720 × 525	8+8+8	30 fps	260 Mbps
<b>BT601 -4:2:2</b>	720 × 625	8+4+4	25 fps	172 Mbps
<b>BT601 -4:4:4</b>	720 × 625	8+8+8	25 fps	257 Mbps
<b>BT709<sup>d</sup> – 4:2:2</b>	1920 × 1080	8+4+4	30 fps	950 Mbps

<sup>a</sup> *Common Intermediate Format*. <sup>b</sup> *Quarter CIF*. <sup>c</sup> *SDTV - Standard-Definition Television*, <sup>d</sup> *HDTV - High-Definition Television*. <sup>e</sup> Número de bits utilizados para representar os *pixels* de luminância (Y) e crominância (C1 + C2). <sup>f</sup> Taxa de quadros por segundo. <sup>g</sup> Taxa de bits (bruta) = resolução horizontal × resolução vertical × resolução temporal × bits por *pixel*.

Os sistemas de vídeo digital podem ser classificados conforme a taxa de bits obtida após o processo de compressão. Assim, sinais de vídeos de taxa muito baixa de bits (VLBR – *very low bit-rate*) têm taxas menores do que 64 kbps; taxas médias de bits estão situadas entre 64 kbps e 1,5 Mbps, e taxas altas são consideradas aquelas maiores do que 1,5 Mbps. Em aplicações VLBR, como videoconferência e vídeo telefonia, geralmente são usados sinais com baixa resolução espacial e temporal. Em aplicações como videoconferência em taxas médias de bits e vídeo em CD-ROM, resoluções maiores são utilizadas. As altas taxas de bits geralmente são usadas em aplicações nas quais uma alta qualidade de imagem é exigida, tais como, gravação de vídeo em DVD e televisão digital, sendo usados os formatos ITU-R 601 e HDTV.

Para obter a compressão necessária para cada aplicação, as redundâncias estatísticas, espaciais, temporais e de cor existentes nos sinais de vídeo são exploradas, além disso, as limitações inerentes ao SVH ainda permitem aumentar a taxa de compressão sem que haja perda perceptível de qualidade. Se forem também toleradas perdas perceptíveis é possível aumentar ainda mais a taxa de compressão, podendo ocorrer nesse caso o aparecimento de artefatos visuais que prejudicam a qualidade da imagem [5]-[6]. Essa exploração das redundâncias do sinal associada com as características do SVH nos remete ao principal problema da codificação de sinais de imagem e vídeo que é determinar a quantidade de bits necessária para manter a qualidade de imagem aceitável para um observador humano.

Nas aplicações de vídeo anteriormente mencionadas, além da taxa de compressão e qualidade da imagem, a escolha da técnica de codificação usada é também influenciada por fatores tais como complexidade computacional, memória requerida, atraso de codificação e forma de acesso às informações.

## 2.3 Quantização

O objetivo da quantização é possibilitar a representação de valores reais através de códigos binários. Dentre as formas de quantização mais comuns, se destacam a quantização escalar e quantização vetorial [1]. Essa quantização consiste no mapeamento de um conjunto contínuo de valores  $X$  em um conjunto discreto  $Y$ ,  $Q: X \rightarrow Y$ . O mapeamento na direção inversa, chamada de dequantização, é denotado por  $Q^{-1}: Y \rightarrow X'$ . A combinação dos dois mapeamentos  $Q^{-1} \times Q: X \rightarrow X'$ , representa o mapeamento do espaço contínuo  $X$  sobre em subespaço discreto  $X'$ , que é um processo não-linear com perdas.

### 2.3.1 Quantização Escalar

A quantização escalar consiste no mapeamento, através de um operador  $Q_E[\cdot]$ , de um conjunto contínuo de infinitos valores de  $X = \{x \in \mathbb{R}\}$  em um conjunto finito discreto  $Y = \{y_0, y_1, \dots, y_{L-1} \mid y_i \in \mathbb{Z}\}$ . O mapeamento  $Q_E$  é definido através da partição do conjunto  $X$  em intervalos não sobrepostos  $\{X_i\} = (x_{i-1}, x_i)$ , onde os limites desse intervalo  $x_{i-1}$  e  $x_i$  são chamados de níveis de decisão. A cada intervalo  $\{X_i\}$  é associado um valor de reconstrução  $y_i$ .

O projeto de um quantizador escalar consiste na determinação dos níveis de decisão e dos valores de reconstrução que minimizam a distorção para uma determinada taxa de bits. O quantizador ótimo em termos do compromisso entre distorção e taxa de bits pode ser obtido através do algoritmo de Lloyd-Max [7], [8], que determina os valores de reconstrução de forma iterativa. Quando a distribuição de probabilidade é uniforme o quantizador ótimo é uniforme. Esse quantizador uniforme é muito usado em conjunto com um codificador de entropia no processo

de compressão de imagens devido à sua simplicidade de implementação e eficiência [9].

### 2.3.2 Quantização Vetorial

A quantização vetorial é um processo de discretização de um espaço vetorial pela sua partição em células e a representação de cada célula por um vetor de reconstrução. O quantizador vetorial  $Q_v[\cdot]$  mapeia um vetor  $\mathbf{x} \in \mathbb{R}^N$  em um conjunto finito  $\mathbf{Y} = \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{L-1} \mid \mathbf{y}_i \in \mathbb{Z}^N\}$  de vetores de reconstrução, onde  $\mathbf{x} \in C_i \Rightarrow Q_v[\mathbf{x}] = \mathbf{y}_i$ , para  $i = 0, 1, \dots, L-1$ . O conjunto de vetores  $\mathbf{Y}$  é conhecido por dicionário de códigos ou *codebook* e as células  $C_i$  podem ser interpretadas como poliedros no espaço  $\mathbb{R}^N$  que são delimitados pelos níveis de decisão.

O quantizador vetorial pode ser projetado através do algoritmo generalizado de Lloyd, conhecido por LBG [10], tendo como principais problemas a geração do dicionário de códigos, a memória para seu armazenamento e o tempo de procura do vetor na codificação [11].

## 2.4 Codificação de Entropia

Para explicar o conceito de entropia de um sinal, utilizaremos o modelo de Shannon, no qual o canal de comunicação transporta uma sequência infinita de símbolos pertencentes a um alfabeto finito [12]. Esses símbolos ocorrem segundo alguma lei de probabilidade e podem ser alterados no canal por ruídos de acordo com outra lei de probabilidade. Para o uso efetivo desse canal de comunicação na transmissão de informações, existem dois problemas básicos a serem resolvidos:

- i) representar o conteúdo do sinal de uma maneira mínima (codificação de fonte);
- ii) proteger o sinal para realizar a transmissão sem erros (codificação de canal).

Segundo os resultados de Shannon, esses problemas podem ser tratados separadamente, de modo que o conhecimento da estatística do sinal não é necessário para a codificação de canal, como também a codificação da fonte pode ser realizada sem o conhecimento da estatística do canal [12].

A entropia de um sinal pode ser entendida como a quantidade de informação do sinal. Se os símbolos  $x_i$  transmitidos em um canal têm

frequência de ocorrência dada pela probabilidade  $p_i$ , então, cada símbolo  $x_i$  transporta a quantidade de informação igual ao  $-\log_2 p_i$ . Assim, a entropia do sinal, ou quantidade de informação de um sinal, é definida como:

$$H = -\sum_{i=1}^N p_i \log_2 p_i \text{ bits por símbolo.} \quad (1)$$

O valor de entropia  $H$  é o limite teórico mínimo de taxa de bits que pode ser obtido por um codificador sem perdas, ou codificador de entropia [13]. A diferença entre a taxa de bits e o valor de entropia é a redundância de código, a qual é explorada pelos codificadores de entropia.

#### 2.4.1 Codificação de Huffman e Tabelas VLCs

Uma forma simples de se obter um código que resulte em uma taxa de bits próxima a entropia de um sinal é atribuir aos símbolos um número de bits inversamente proporcional a sua probabilidade de ocorrência. A codificação de Huffman se baseia na construção de uma estrutura que permite representar de forma eficiente tais símbolos. Para a implementação do código de Huffman é necessário conhecer *a priori* a probabilidade de ocorrência dos símbolos. Essa probabilidade pode ser obtida através da estatística de distribuição dos símbolos no próprio codificador [14].

Nas recomendações H.261 [15], H.263 [16] e H.264 [17], como também nos padrões MPEG-1 [18], MPEG-2 [19] e MPEG-4 [20]-[21], os códigos de Huffman são utilizados para codificar os coeficientes transformados, informações auxiliares e informações de movimento. Nesses casos a sua implementação é feita através de tabelas de código de comprimento variável (*variable length code* - VLC), as quais são projetadas para cada tipo de informação, ou ainda através de tabelas universais como as UVLC (*universal variable length code*) usadas no H.264. Também podem ser utilizadas tabelas regulares reversíveis, que adotam uma estrutura de codificação que permite a leitura dos bits no sentido direto e reverso, tornando a codificação mais tolerante a erros de transmissão [22], [23].

#### 2.4.2 Codificação Aritmética

Na codificação aritmética [24], [25], uma sequência de símbolos, denominada mensagem, é representada por um intervalo de números

reais entre 0 e 1. Para cada mensagem o valor binário de qualquer valor real dentro do intervalo pode ser transmitido como código. Quanto maior for o número de símbolos na mensagem, maior poderá ser a compressão obtida e mais a taxa de bits se aproximará do valor de entropia.

Essa codificação é utilizada no Anexo E da Recomendação H.263 [16] e no Anexo D do padrão JPEG [26], nos quais as tabelas VLC são substituídas pela codificação aritmética resultando em uma redução de até 5% na taxa de bits [23]. Como não há perdas, a qualidade final da imagem não é alterada e ocorre apenas um ganho em taxa de bits, à custa de uma maior complexidade de codificação e decodificação.

### **2.4.3 Codificação *Run-Length***

No processamento de imagem e vídeo, quando o resultado do uso de transformações lineares seguidas de quantização resulta em longas sequências de um único símbolo, um pré-processamento pode trazer em significativo aumento da taxa de compressão do codificador de entropia. Uma dessas possibilidades é o uso da codificação *run-length*, que consiste em substituir as sequências de símbolos repetidos por códigos que representam o seu comprimento [27]. Essa técnica é usada nos padrões atuais de vídeo para a codificação dos coeficientes das transformadas [15], [16], [18]-[20].

### **2.4.4 Codificação Adaptativa ao Contexto**

Sinais de vídeo naturais apresentam um comportamento estatístico não estacionário, dependendo fortemente do conteúdo da imagem e do processo de aquisição. Os métodos de codificação de entropia mencionados nas seções anteriores exploram parcialmente esse comportamento. Codificadores adaptativos baseados no contexto, tal como a CAVLC (*context-adaptive variable length coding*) e na CABAC (*context-adaptive binary arithmetic coding*), exploram as dependências estatísticas de maior ordem para obter uma melhoria significativa na eficiência em troca de um aumento de complexidade da codificação.

A CAVLC é utilizada no H.264 para a transmissão dos coeficientes quantizados após a sua varredura em ziguezague. Na CAVLC, tabelas de VLC para vários elementos sintáticos são chaveadas de acordo com elementos previamente transmitidos. Uma compressão adicional da taxa de bits é obtida, pois as VLCs são projetadas para se ajustar a estatística condicionante. Tipicamente se utilizam como condicionantes elementos sintáticos como: números de coeficientes não nulos, número de “1”s do final de uma sequência, valor dos coeficientes,

informação da polaridade dos “1”s, total de zeros e ainda os zeros antes do próximo coeficiente [28].

O processo de codificação CABAC consiste de três passos principais: binarização, modelagem do contexto e codificação aritmética binária. No primeiro passo um elemento sintático não binário é mapeado univocamente em uma sequência binária. Na modelagem do contexto, um modelo de probabilidade é selecionado de acordo com elementos sintáticos previamente codificados. Por último, o processo de codificação aritmética é realizado. A estimação da probabilidade e a codificação aritmética são realizadas usando abordagens baseadas em tabelas. Os resultados obtidos em [29] mostram que em sequências de vídeo típicas para televisão (*Canoe, Formula 1, Rugby, Mobile&Calendar e Football*), nas quais a CABAC melhor se aplica, o ganho obtido em taxa de bits em relação à codificação usando VLCs, se situa entre 9% e 14% na faixa de qualidade de vídeo aceitável (>30 dB).

## 2.5 Redundância Espacial

A compressão de imagens estáticas é baseada na redução de redundância espacial, redução de redundância de código e eliminação de características não percebidas pelo observador humano sob certas condições de observação [6], [30]. A redundância espacial decorre da existência de regiões nas quais os valores dos *pixels* vizinhos estão fortemente correlacionados. Existem diversas técnicas que permitem explorar a redundância espacial nas imagens estáticas, dentre as principais citamos a predição espacial dos *pixels* e o uso de transformadas que descorrelacionam os *pixels*.

As transformadas lineares mapeiam uma imagem em uma nova representação mais compacta de energia, visando concentrar grande parte da energia em um número reduzido de coeficientes. Essas transformadas devem ser inversíveis quando se deseja obter uma aproximação da imagem reconstruída através da inversão do processo. Os tipos de transformações lineares mais usados em compressão de imagens são: transformada cosseno discreta (DCT), transformada *wavelet* discreta (DWT), a transformada inteira e recentemente a transformada *matching pursuits*. A transformada de Karhunen-Loève, apesar de ser ótima em termos de minimização da distorção do erro quadrático médio do sinal reconstruído, não é utilizada na compressão de imagem por ter um desempenho similar à DCT com complexidade computacional muito superior [31].

Da mesma forma como são usadas para decorrelacionar imagens, essas transformadas também podem ser aplicadas na codificação do erro de predição entre quadros sucessivos na codificação de vídeo.

### 2.5.1 Transformada Discreta do Cosseno (DCT)

A DCT é uma transformada muito usada em codificação de imagens e corresponde à decomposição da imagem usando um conjunto de funções de base do tipo cosseno. A definição da transformada 2-D para um sinal  $N \times N$  é dada por

$$S_{u,v} = \frac{1}{4} c_u c_v \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} s_{x,y} \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N} \quad (2)$$

onde  $c_u, c_v = 1/\sqrt{2}$  para  $u, v = 0$  e  $c_u, c_v = 1$  para  $u, v = 1, \dots, N-1$  [16], [19]. Os coeficientes  $S_{u,v}$  correspondem às frequências vertical e horizontal da posição  $u, v$ . O coeficiente  $S_{0,0}$  é o componente contínuo do sinal, ou seja, seu valor médio e  $s_{x,y}$  é o valor do *pixel* da imagem na posição  $x, y$ .

Nas implementações de codificação de imagem e vídeo em que a DCT é considerada, a imagem é geralmente subdividida em blocos de tamanho  $N \times N$ . Como a transformação 2-D é separável, ela pode ser inicialmente aplicada a cada uma das  $N$  colunas e em seguida as  $N$  linhas obtidas. No entanto, em virtude de alguns valores  $S_{u,v}$  serem irracionais, a aplicação em cascata da transformada DCT e sua inversa pode resultar em descasamento (*drift*) entre os valores obtidos no codificador e no decodificador [32].

### 2.5.2 Transformada Inteira

A transformada inteira foi desenvolvida visando reduzir a complexidade computacional da DCT e eliminar o descasamento entre os valores obtidos no codificador e decodificador [32]. A transformada inteira  $\mathbf{H}$  usada no padrão H.264 é um exemplo no qual uma aproximação escalada da DCT  $\mathbf{H} = \text{round}\{\alpha \mathbf{H}_{\text{DCT}}\}$  é usada. Para uma DCT  $4 \times 4$  temos

$$\mathbf{H}_{\text{DCT}} = \begin{bmatrix} 0,5 & 0,5 & 0,5 & 0,5 \\ 0,6533 & 0,2706 & -0,2706 & -0,6533 \\ 0,5 & -0,5 & -0,5 & 0,5 \\ 0,2706 & -0,6533 & 0,6533 & -0,2706 \end{bmatrix}. \quad (3)$$

Então com  $\alpha = 2,5$  se tem

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}. \quad (4)$$

A grande vantagem dessa transformada é poder ser aplicada a blocos de *pixels*  $4 \times 4$  empregando apenas operações de adição e deslocamento de bits. Além do mais, apenas seis bits a mais do que os usados para cada *pixel* antes da transformação são necessários para armazenar os coeficientes transformados [32]. Para obter a transformação inversa  $\tilde{\mathbf{H}}_{\text{inv}}$ , dividem-se por 2 as funções de base ímpares de  $\mathbf{H}$  e transpõe-se a matriz resultante [32], de modo que

$$\tilde{\mathbf{H}}_{\text{inv}} = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix}. \quad (5)$$

Com a abordagem descrita em [32], é possível determinar a transformada direta e sua inversa para um bloco  $4 \times 4$  usando apenas números inteiros de 16 bits quando os *pixels* originais são representados por 8 bits. Devido à diferença no valor da norma (soma dos coeficientes) das funções de base (linhas) usadas, é necessário multiplicar os valores obtidos (após a aplicação da transformação inversa) pela matriz diagonal  $\mathbf{D}$  [32]

$$\mathbf{D} = \begin{bmatrix} 1/4 & 0 & 0 & 0 \\ 0 & 1/5 & 0 & 0 \\ 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 1/5 \end{bmatrix}. \quad (6)$$



### 2.5.3 Transformada Wavelet Discreta (DWT)

A transformada *wavelet* [33]-[35] permite representar um sinal qualquer como a combinação linear um conjunto de funções de base ortogonais  $w_{i,k}(t)$ , denominadas *wavelets*. Essas funções são pequenas ondas localizadas, com extensão espacial limitada e que são escaladas em frequência. Usando uma base *wavelet* ortogonal, o sinal contínuo  $x(t)$  pode ser representado por

$$x(t) = \sum_j \sum_k b_{j,k} w_{j,k}(t), \text{ com } w_{j,k}(t) = w(2^j t - k) \quad (7)$$

onde  $w_{j,k}(t)$  são as funções escaladas em frequência e deslocadas no tempo,  $j$  e  $k$  são as variáveis que definem a escala e posição e  $b_{j,k}$  são os coeficientes das *wavelet* obtidos por

$$b_{j,k} = \int_{-\infty}^{\infty} x(t)w_{j,k}(t)dt. \quad (8)$$

Após a aplicação da transformação, quando um coeficiente tem um alto valor para um determinado par  $(j, k)$ , isso indica que o sinal  $x(t)$  possui um importante componente nessa posição  $k$  e escala  $j$ . A partição do sinal em diferentes escalas conduz à propriedade de multirresolução dessa transformada.

A obtenção dos coeficientes *wavelet* na maioria das implementações é feita através da aplicação sucessiva de filtros passa-baixa  $H_a(z)$  passa-alta  $G_a(z)$  seguidas da dizimação por um fator 2, conforme ilustrado na Figura 1.

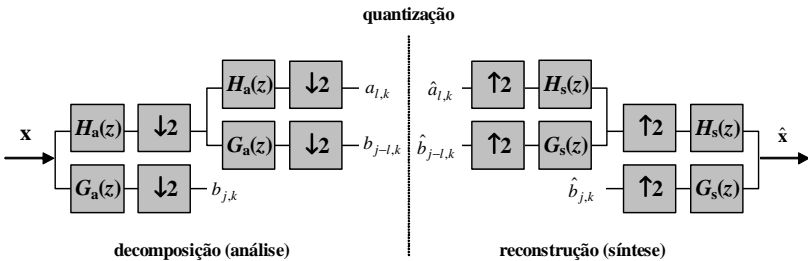


Figura 1 - Implementação da DWT através de banco de filtros.

Em imagens a transformação *wavelet* é geralmente efetuada através de duas decomposições 1-D sucessivas na direção horizontal e vertical. Para se obter uma transformada *wavelet* estruturada em árvore (decomposição de Mallat [33]) é necessário repetir a decomposição até atingir o nível de detalhe desejado.

O padrão de codificação de imagens estáticas JPEG 2000 [36], [37], utiliza para codificação com perdas uma DWT não inversível da família Daubechies biortogonal 9/7 (ver Tabela 2) [38] bem como filtro de Le Gall 5/3 (ver Tabela 3) [39] para a codificação sem perdas [40]. A decomposição utilizada é a de Mallat [33] e codificação dos coeficientes segue o método LZC (*layered zero coding*) [41], [42]. Uma das grandes vantagens do uso da DWT é a possibilidade de aumentar a taxa de compressão, sem que haja perdas perceptíveis, explorando o efeito do contraste local, a função de sensibilidade ao contraste e as características de mascaramento do SVH [43]. Isso é possível porque a DWT divide a imagem em bandas de frequência com diferentes resoluções de forma semelhante ao que ocorre no SVH.

Tabela 2 – Coeficientes dos filtros Daubechies biortogonal 9/7

k	filtros de análise		filtros de síntese	
	passa-baixa $H_a(z)$	passa-alta $G_a(z)$	passa-baixa $H_s(z)$	passa-alta $G_s(z)$
0	0,60294901823	1,1150870524	1,1150870524	0,60294901823
$\pm 1$	0,26686411844	-0,59127176311	0,59127176311	-0,26686411844
$\pm 2$	-0,07822326652	-0,05754352622	-0,05754352622	-0,07822326652
$\pm 3$	-0,01686411844	0,09127176311	-0,09127176311	0,01686411844
$\pm 4$	0,02674875741			0,02674875741

Tabela 3 – Coeficientes dos filtros Le Gall 5/3

k	filtros de análise		filtros de síntese	
	passa-baixa $H_a(z)$	passa-alta $G_a(z)$	passa-baixa $H_s(z)$	passa-alta $G_s(z)$
0	6/8	1	1	6/8
$\pm 1$	2/8	1/2	-1/2	-2/8
$\pm 2$	-1/8			-1/8

### 2.5.4 Matching Pursuits

O algoritmo de *matching pursuits* [44] expande o sinal  $f(t)$  usando um conjunto redundante de funções de base  $g_i(t) \in G$ , onde  $G$  é um dicionário de funções. Para representar um sinal  $f(t)$  pela combinação linear das funções (átomos) do dicionário, inicialmente é encontrada a função  $g_i(t)$  que tenha a maior correlação com o sinal e obter o coeficiente  $q_i$  dessa função através do produto interno

$$q_i = \langle f(t), g_i(t) \rangle. \quad (9)$$

Se a representação do sinal não é satisfatória, uma nova função  $g_j(t)$  que mais se aproxime do resíduo  $R(t)$  é buscada no dicionário.

$$R(t) = f(t) - q_i g_i(t) \quad (10)$$

O processo de busca de funções no dicionário é repetido até a obtenção da taxa de bits ou de um nível de energia residual desejado. Assim, após um total de  $M$  estágios, o sinal  $\hat{f}(t)$  pode ser aproximado pela combinação linear das funções do dicionário  $g_j(t)$ , através de

$$\hat{f}(t) = \sum_{i=1}^M q_i g_i(t). \quad (11)$$

Esta técnica apresenta propriedades que são adequadas para a representação de imagens, uma vez que a cada estágio é escolhida sempre a função que resulte na maior redução do erro entre o sinal original  $f(t)$  e o codificado  $\hat{f}(t)$ , possibilitando estabelecer de forma simples tanto um bom compromisso taxa-distorção (RD) quanto uma codificação progressiva do sinal. No entanto, a otimização do dicionário de funções de base e o algoritmo de escolha de tais funções são problemas não triviais que precisam ser resolvidos [45].

### 2.5.5 Predição Espacial

A predição espacial é uma forma de exploração de redundância espacial na qual os *pixels* de um bloco de são preditos a partir de *pixels* dos blocos vizinhos. No caso do padrão H.264, essa forma de predição é usada para a codificação de blocos para os quais nenhuma referência a quadros anteriores é feita (blocos *Intra*). Os valores preditos são

determinados segundo uma direção de predição predeterminada, sendo previstas oito direções de predição para blocos  $4 \times 4$  (ver Figura 2) e duas direções para blocos  $16 \times 16$  [17].

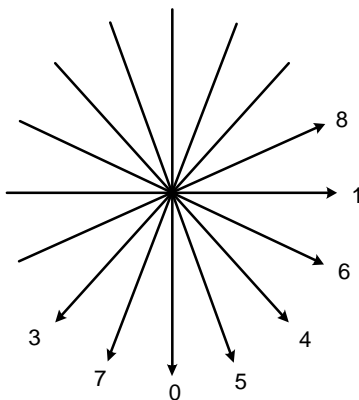


Figura 2 - Direções de predição para blocos  $4 \times 4$  do H.264.

O uso dessa técnica em conjunto com a transformada inteira (conforme o padrão H.264) resulta em um desempenho RD superior ao obtido aplicando as transformadas DCT e DWT (conforme os padrões JPEG e JPEG2000) [32].

Para exemplificar a forma como é feita a predição dos *pixels*, ilustramos a seguir a predição para a direção 3 (diagonal abaixo/esquerda), em um bloco  $4 \times 4$ . Os *pixels* preditos (**a** até **p**) são obtidos pela média ponderada de até 3 *pixels* da última linha dos blocos imediatamente acima deles (*pixel A* até **H**), conforme mostrado na Figura 3 [17]. Para as outras direções de predição, procedimentos similares são efetuados. Após a escolha da melhor predição, o erro entre o bloco predito e o bloco original é codificado utilizando uma das técnicas de compressão de imagens estáticas anteriormente descritas.

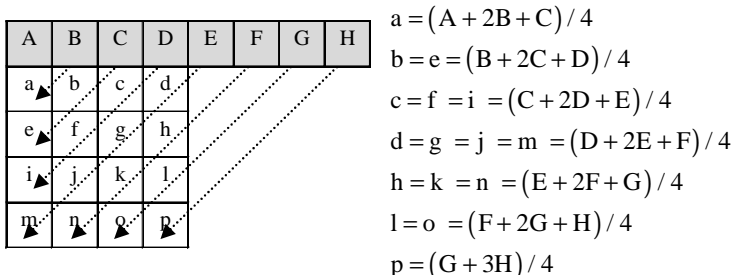


Figura 3 - Modo 3 de predição de pixels para blocos 4×4 do H.264.

## 2.6 Redundância Temporal

Na compressão de sinais de vídeo além da redundância espacial e de código, é explorada também a redundância temporal. O nível de redundância depende da forma de como os quadros são adquiridos variando com o tipo de cena e a taxa de quadros usada. A redundância temporal entre as imagens dos quadros sucessivos é correspondente a semelhança existente entre os *pixels* do quadro atual com *pixels* de quadros anteriores. As mudanças que ocorrem nos *pixels* da mesma posição espacial em quadros sucessivos podem se originar do: (1) movimento de objetos na cena (movimentos de translação, rotação, afastamento ou aproximação); (2) movimento de giro da câmera; (3) efeito de aproximação ou afastamento da cena (*zoom*); e (4) variações da iluminação.

Na Figura 4 (a) e (b) dois quadros sucessivos da sequência de vídeo *Suzie* são mostrados, nos quais a redundância temporal pode ser notada através da grande semelhança entre os quadros. A imagem de diferenças entre os quadros ( $\Delta Q$ ) é mostrada na Figura 4 (c) na qual as diferenças são indicadas através da intensidade da imagem (*pixels* com diferença nula são pretos). As diversas técnicas usadas para explorar essa redundância são classificadas como: (1) baseadas na forma de onda; (2) baseadas em objetos; e (3) baseadas em modelos [46]-[48].



Figura 4 - Sequência de vídeo *Suzie*. (a) Componente de luminância do quadro #21. (b) Componente de luminância do quadro #22. (c) Diferença temporal entre os componentes de luminância dos quadros #21 e #22.

### 2.6.1 Técnicas Baseadas na Forma de Onda

Dentre as técnicas baseadas na forma de onda podem-se destacar diversos métodos que utilizam transformadas *wavelet* 3-D, nas quais a dimensão temporal é vista como uma terceira dimensão da transformada [49]. Esses métodos são rápidos e eficientes, e possibilitam uma escalabilidade temporal, limitando também a propagação de erro entre quadros devido a sua não recursividade [50]. No entanto, tais técnicas têm um atraso de codificação maior em função da necessidade de armazenamento prévio dos quadros para a realização da filtragem temporal [51]. Esses métodos também apresentam um fraco desempenho em aplicações de VLBR [52].

Os codificadores híbridos formam outro grupo com grande eficiência na codificação de vídeo. Tais codificadores usam técnicas de predição para explorar a redundância temporal e uma transformada 2-D para reduzir a redundância espacial [52]. A predição do quadro atual é obtida a partir de quadros de referência anteriores (ou posteriores), sendo necessário codificar os parâmetros da predição juntamente com os erros de predição (DFD – *displaced frame difference*), sendo a estimação/compensação de movimento (EM/CM) uma das técnicas mais usadas para este fim. Para a codificação da DFD são geralmente consideradas as transformadas DCT [15]-[16], [18]-[19], DWT [53]-[56], *matching pursuits* [45], ou transformada inteira [17], conjugada com uma codificação de entropia. Nestes codificadores o principal problema em VLBR é a dificuldade de representação das bordas dos objetos em movimento, resultando em artefatos de bloqueio ou tremor [57]. Apesar disso, essa é a técnica usada em todos os padrões internacionais de codificação de vídeo (H.261, H.263, H.264, MPEG-1,

MPEG-2 e MPEG-4). Alguns desses codificadores, além do empregar a CM em conjunto com uma transformação 2-D, permitem explorar também a redundância espacial através do emprego dos diversos modos *Intra* em blocos nos quais a predição temporal obtida é inadequada [16], [17].

Por último existe ainda um grupo de técnicas que utiliza a transformada 3-D associada à EM/CM [58], [59]. Nesses casos a CM é usada para aumentar a correlação entre os quadros antes de realizar a filtragem temporal, melhorando o desempenho da transformação 3-D. Em aplicações com altas taxas de bits a distorção é reduzida quando comparada com codificadores que usam transformadas 2-D; no entanto, o uso dessas técnicas implica um aumento considerável de complexidade. [52].

### 2.6.2 Técnicas Baseadas em Objetos

Nesta classificação, agrupam-se as técnicas usam métodos de análise de imagens e de computação gráfica que segmentam a imagem em objetos visuais, aos quais são associados parâmetros como textura, forma e movimento [60]-[63]. A codificação de textura é geralmente feita através das transformações DCT ou DWT. Para a codificação de forma pode ser usado um *bitmap* do contorno [64] ou uma representação direta da forma por uma estrutura em árvore de blocos (*quad-tree*) [65]. A codificação de movimentos é feita através de parâmetros associados a cada objeto visual. A eficiência dessas depende muito do processo de segmentação [66] e da alocação dos bits para a codificação de cada parâmetro [67]. Uma importante consequência do uso dessa técnica é a possibilidade de acesso e manipulação em separado de cada objeto visual. Um exemplo de aplicação dessa técnica é o padrão MPEG-4 (Parte 2) [21].

### 2.6.3 Técnicas Baseadas em Modelos

A codificação baseada em modelos [68]-[71] obtém baixíssimas taxas de bits empregando técnicas de computação gráfica. Nessa abordagem a imagem é tratada como uma projeção em duas dimensões do mundo tridimensional. O codificador analisa a cena usando modelos de objetos visuais e codifica os parâmetros dos modelos. No decodificador a cena é sintetizada a partir dos parâmetros desses modelos, conforme ilustrado na Figura 5.

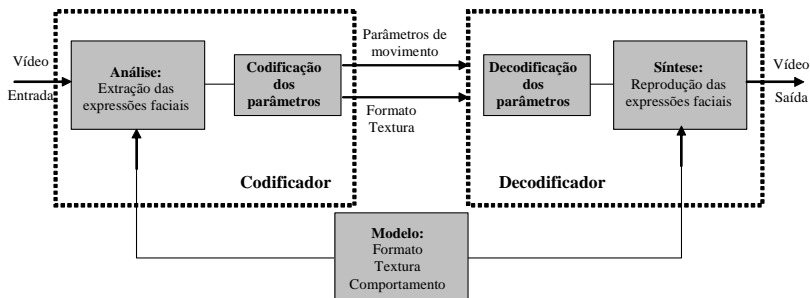


Figura 5 - Diagrama em blocos de um codificador baseado em modelos.

Como o conteúdo da sequência de imagens precisa ser conhecido *a priori*, a aplicação desta técnica fica normalmente limitada a sequências simples de videoconferência, tais como “cabeça e ombros” [67], [71]. O padrão MPEG-4 inclui esse tipo de técnica para animação facial e corporal dentro do contexto de codificação sintético-natural [4], [21], [67].

## 2.7 Codificação da Cor

A forma mais exata de representar uma cor é através do seu espectro de radiação. Essa representação não é usada em codificadores porque exige uma quantidade muito grande de bits, além de não se adequar a forma como o SVH processa a informação de cor. No SVH a cor é processada como uma soma ponderada da resposta de três tipos de cones (*S - short*, *M - medium* e *L - long wavelength*) que capturam a imagem na retina [3], por isso, é possível abranger toda a capacidade de distinção de cores do SVH através da descrição de cores por três componentes elementares. Os sistemas utilizados para a reprodução e codificação de imagens em cores, podem ser classificados em: aditivos, subtrativos. Nos sistemas aditivos a cor é reproduzida através da soma dos componentes de cores primárias: vermelho (*red - R*), verde (*green - G*) e azul (*blue - B*), enquanto nos sistemas subtrativos a cor é reproduzida através da remoção dos componentes espectrais não desejados de uma fonte de luz branca, sendo usados os pigmentos ciano (*C*), magenta (*M*), amarelo (*Y*) e preto (*K*) [3].

### 2.7.1 Espaços Padronizados pelo CIE

Com o objetivo de obter o modelo de representação independente do suporte no qual a imagem é mostrada, o CIE (*Commission Internationale de L'eclairage*) [72] realizou diversos experimentos para



estabelecer representações padrão de cores, as quais são denominadas de espaço de cores. Esses espaços são projeções (matemáticas) do espectro de radiação sobre uma tripla de coordenadas de cores. Para modelar a resposta do SVH às diferentes cores é usada a função de casamento de cores (FCC). A FCC indica para um observador humano típico, a equivalente combinação das cores primárias para cada cor do espectro de radiação. Essas FCCs permitem especificar a aparência da cor independente do suporte utilizado para mostrá-las. Na Figura 6 são mostradas as curvas das três FCCs  $\bar{x}(\lambda)$ ,  $\bar{y}(\lambda)$ ,  $\bar{z}(\lambda)$  de um observador padrão CIE 1951 [3], [43].

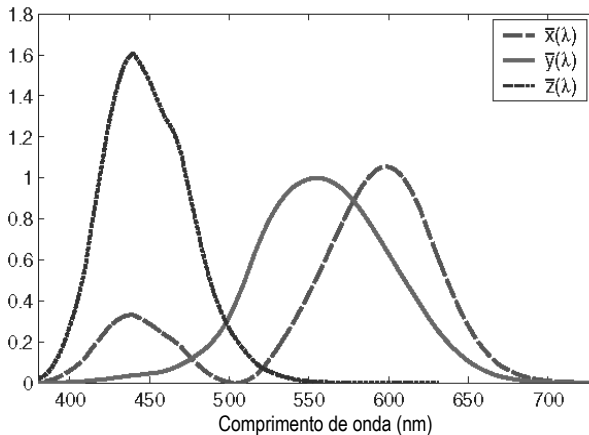


Figura 6 - FCCs do observador padrão: CIE 1951 para definição de 2 graus.

### 2.7.2 Espaços de Cores Usados na Codificação

Além do espaço de cores padronizado *RGB*, outros, podem ser obtidos através de transformações lineares e não-lineares desse espaço de cores. Na codificação de imagens o espaço de cores a ser empregado deve ter as seguintes características [43]:

- i) possibilitar a medição da diferença entre cores da mesma forma como um observador humano as percebe;
- ii) representar a cor da maneira mais compacta possível, reduzindo a sua redundância.

A Figura 7 ilustra a redundância de cor existente no espaço *RGB*. Note a presença de áreas claras e escuras nos componentes *R*, *G* e *B* nas mesmas posições espaciais. Os espaços de cores *RGB* [73] e *CMYK* [3]

não possuem nenhum das características mencionadas anteriormente, e por isso outros espaços de cores são utilizados na codificação de imagens [74], [75]. Durante a análise de desempenho realizada para o padrão JPEG, foi verificado que apenas as representações do tipo luminância-crominância possuem simultaneamente essas características. Dentre esses espaços de cores, aqueles que melhor se adaptam para a compressão da imagem são o  $L^*a^*b^*$  [73] que possui a característica (i) e o  $YC_B C_R$  [76] que atende melhor a característica (ii), concentrando grande parte da energia no componente de luminância. Como consequência, no  $YC_B C_R$ , a luminância é geralmente codificada com maior precisão, enquanto uma resolução menor é usada na codificação dos componentes de crominância.



Figura 7 - Representação de imagem no espaço de cores RGB. (a) Vermelho, (b) Verde. (c) Azul.

#### a) Espaço $YC_B C_R$

O espaço  $YC_B C_R$  definido pela recomendação ITU-R BT.601 [76] foi projetado especificamente com o propósito de obter uma grande redução da redundância de cor. A conversão de cores do espaço  $RGB$  para  $Y C_B C_R$  é feita da transformação linear

$$\begin{bmatrix} Y \\ C_B \\ C_R \end{bmatrix} = \begin{bmatrix} +0,299 & +0,587 & +0,114 \\ -0,167 & -0,331 & +0,5 \\ +0,5 & -0,419 & -0,081 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (12)$$

onde  $Y$  representa a luminância,  $C_B$  a crominância azul-amarelo,  $C_R$  a crominância verde-vermelho e  $R$ ,  $G$  e  $B$  são os componentes de cores primárias.

Na Figura 8 é mostrada a intensidade dos *pixels* dos componentes  $Y$ ,  $C_B$  e  $C_R$ . Note a redução da correlação entre os componentes. Apesar dessa redução, continuam existindo fortes dependências estatísticas de ordens superiores entre os componentes [77], as quais podem ser

exploradas atribuindo os mesmos parâmetros (borda, forma e movimento) aos objetos visuais no processo de codificação da imagem.

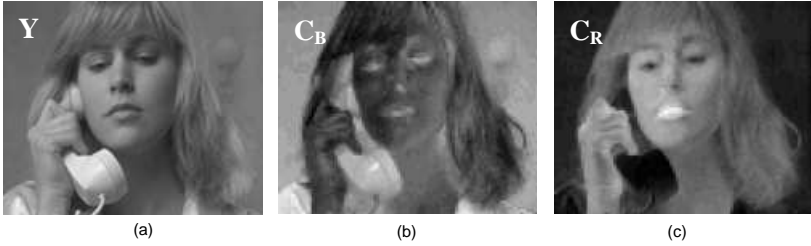


Figura 8 - Representação de imagem no espaço de cores  $YC_B C_R$ . (a) Luminância  $Y$ . (b) Crominância  $C_B$ . (c) Crominância  $C_R$ .

*b) Espaço  $L^*a^*b^*$*

O espaço de cores CIE 1976  $L^*a^*b^*$  (CIELAB) [73] é um espaço colorimétrico que permite uma adequada descrição de cores e também a medição da sua diferença. O CIELAB é um espaço Euclidiano com eixos ortogonais entre si. A luminância perceptual é representada por  $L^*$ , a crominância vermelho-verde representada por  $a^*$  e a crominância azul-amarelo representada por  $b^*$  (Figura 9). A transformação do espaço CIE 1931 para o  $L^*a^*b^*$  [78], [43] é dada por

$$\begin{aligned} L^* &= 116f(Y/Y_w) - 16 \\ a^* &= 500[f(X/X_w) - f(Y/Y_w)] \\ b^* &= 200[f(Y/Y_w) - f(Z/Z_w)] \end{aligned} \quad (13)$$

onde  $X$ ,  $Y$  e  $Z$  são os valores dos três estímulos no CIE 1931;  $X_w$ ,  $Y_w$  e  $Z_w$  são os valores do branco de referência; e

$$f(x) = \begin{cases} x^{1/3}, & x > 0,008856 \\ 7,787x + 16/116, & x \leq 0,008856 \end{cases} \quad (14)$$

A normalização em relação ao branco de referência é realizada para adaptar a luminância e crominância às condições de observação.

No espaço  $L^*a^*b^*$  a diferença de cor pode ser medida pela distância Euclidiana

$$\Delta E_{ab} = \sqrt{\Delta L^{*2} + \Delta a^{*2} + \Delta b^{*2}} \quad (15)$$

na qual a menor distorção perceptível pelo observador humano corresponde à  $\Delta E_{ab}^* = 2,3$  [73], [79].

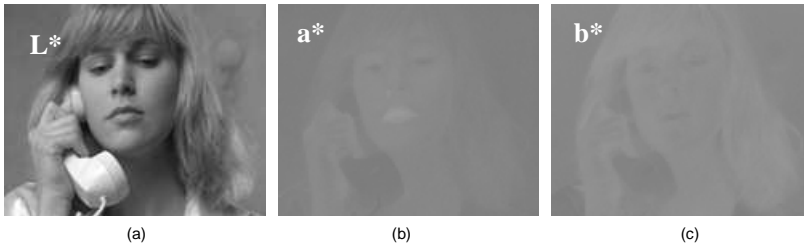


Figura 9 - Representação de imagem no espaço de cores  $L^*a^*b^*$ .

### 2.7.3 A Cor nos Padrões de Codificação de Vídeo

Nos codificadores que seguem os padrões MPEG-2, MPEG-4, H.261, H.263 e H.264, o espaço de cores usado é o  $YC_B C_R$ . Frequentemente a opção de amostragem dos componentes de crominância por um fator de 2 nas direções horizontal e vertical (4:2:0) é usada para obter uma maior taxa de compressão. Essa amostragem é usada para ajustar o codificador à função de sensibilidade ao contraste do SVH, que possui uma menor sensibilidade às altas frequências espaciais dos componentes de crominância [80], [81].

Para aumentar a taxa de compressão dos codificadores, a dependência estatística existente entre os componentes de crominância e luminância também pode ser explorada fazendo a compensação conjunta do movimento desses componentes. No entanto, quando a estimação de movimento, efetuada de acordo com o componente de luminância, não corresponde ao movimento real dos objetos, podem surgir artefatos de cor na imagem [82].

Dentre as publicações que discutem compressão de imagens e vídeo, a maioria não lida com os componentes de crominância. Poucos trabalhos apresentam resultados para tais componentes [50], [83]-[89]. A pouca importância dada à crominância é explicada em parte pela baixa taxa de bits (5% a 20% do total) que é necessária para a sua codificação [90]. No entanto, para VLBR, a melhoria de codificação da informação de cor se faz necessária para evitar a geração de artefatos e distorções de cor, produzindo uma melhora sensível na qualidade da imagem [82].

## 2.8 Conclusões

Conforme visto neste capítulo, a codificação e compressão de sinais de vídeo é um processo de fundamental importância para a viabilização de aplicações em VLBR nas quais uma alta taxa de compressão é necessária. Aplicações que permitem taxas de bits mais elevadas, como a televisão digital e a gravação de vídeo em DVD, também são beneficiadas pela maior compressão do sinal através da redução da taxa de bits para a mesma qualidade de imagem.

Mostramos ainda que a codificação de vídeo é realizada através de uma série de etapas entrelaçadas que exploram diferentes características do sinal tais como a redundância espacial, temporal, de cor e de código. Devido à interdependência dessas etapas, o desempenho RD de um codificador de vídeo é influenciado por todo o conjunto de técnicas e parâmetros usados. Por isso, para avaliar a validade de qualquer proposta é necessário que ela seja incluída em um codificador de vídeo completo, de modo que, os resultados possam ser comparados com um codificador de referência que reflita o estado da arte. Neste trabalho adotamos como modelo de referência o codificador H.264, implementado através do seu modelo de teste JM7.3 [91].



## Capítulo 3

### Estimação de Movimento

Neste capítulo prosseguimos o estudo de codificação de sinais de vídeo, focando agora na redução da redundância temporal através da técnica de estimação e compensação de movimento (EM/CM) em blocos. Para tal, são efetuados estudos dos fatores que influenciam na qualidade da imagem obtida após a CM, na complexidade computacional da EM/CM e na taxa de bits. A forma de realizar a CM dos componentes de luminância e crominância também é avaliada.

A técnica de EM/CM – BMA (*block matching algorithm*) é usada na maioria dos algoritmos (incluindo os padrões e recomendações internacionais) para obter a redução da redundância temporal. Em aplicações de VLBR, a qualidade da imagem obtida com codificadores que usam essa técnica é geralmente muito superior àquela obtida com outras técnicas, tais como fractais [92]-[94], transformações 3-D sem EM [49]-[51] e codificadores baseados em modelos [68]-[71]. A etapa da EM também merece uma atenção especial, pois é dentre as diversas etapas de um codificador de vídeo, a de maior complexidade computacional, exigindo um esforço computacional do codificador de até 80% do total [95].

Neste capítulo, inicialmente são discutidos os princípios básicos e terminologia empregada no processo de EM, especialmente as técnicas de casamento de blocos. Em seguida são discutidos os principais fatores que afetam a qualidade da imagem e a complexidade computacional do processo. Para tal, a qualidade da imagem, complexidade computacional e entropia dos vetores de movimento são avaliadas através da variação dos parâmetros ajustáveis na EM (tamanho de bloco, janela de procura, precisão da representação dos VMs, interpolação dos valores de *subpixel* e estratégia de procura). Finalmente, a compensação de movimento dos componentes de crominância é discutida. Neste capítulo todos os resultados apresentados são obtidos com um codificador otimizado em distorção (DO), sendo avaliadas as taxas de bits em conjunto com a distorção. A técnica de otimização RD baseada no multiplicador de Lagrange [96] será considerada a partir do próximo capítulo.

### 3.1 Técnicas de Estimação de Movimento

Nas técnicas de EM, o deslocamento dos objetos visuais capturado pela sequência de quadros de um vídeo é modelado através de informações de movimento que possibilitam obter uma predição do quadro atual a partir de *pixels* já codificados de outro quadro. Quando existe grande redundância temporal, o erro de predição (DFD) entre o quadro atual e o predito pela CM juntamente com as informações de movimento, são codificados no lugar dos *pixels* originais, resultando em altas taxas de compressão. Para tal, uma representação eficiente da informação de movimento e da DFD é de crucial importância nestes codificadores, sendo necessário estabelecer um adequado compromisso entre o detalhamento das informações de movimento e a quantidade de bits usados na codificação da DFD.

Antes de tratar das diferentes técnicas de EM, é necessário fazer uma distinção entre o campo de movimento 2-D e o fluxo óptico. O primeiro é uma projeção no plano do movimento que ocorre na cena, enquanto, o fluxo óptico está associado às variações espaço-temporais de intensidade da luminância. Como o objetivo em codificação de vídeo é reduzir a redundância temporal entre quadros, não há necessidade de uma descrição precisa do movimento, sendo suficiente à estimação do fluxo óptico. As variações de intensidade de luminância que resultam no fluxo óptico pode ser consequência de [47]:

- i) movimento dos objetos na cena (percebido como variações locais no fluxo óptico);
- ii) movimento global ou movimento de câmera (produzindo variação do fluxo óptico mesmo quando não existe qualquer movimento na cena);
- iii) variação das condições de iluminação.

Em aplicações típicas de VLBR, as imagens são adquiridas com câmeras fixas e geralmente a variação de iluminação é mínima. Nestes casos, a variação do fluxo óptico provém na maior parte do movimento de objetos da cena. Quando existe movimento de câmera, uma estimação de movimento em dois estágios global e local pode ser considerada [59].

Os métodos usados para efetuar a EM podem ser classificados em três grupos: (1) baseados no gradiente; (2) recursivos baseados no *pixel*; e (3) baseados em blocos. Estudos comparativos desses métodos [47] mostram que a EM em blocos apresenta resultados superiores aos outros dois métodos em termos de compactação de energia da DFD, sendo por isso empregado para a exploração da redundância temporal nos



codificadores de vídeo que seguem as recomendações e padrões internacionais (H.261, H.263, H.264, MPEG-1, MPEG-2 e MPEG-4).

Existem ainda alguns modelos complexos de estimação de movimento que permitem modelar a deformação, rotação, afastamento e aproximação de objetos [97], [98], assim como modelar às variações de iluminação [99]. No entanto, eles são pouco empregados devido à complexidade computacional, dificuldade na estimação de parâmetros e quantidade de informações adicionais necessárias.

### 3.2 Estimação de Movimento em Blocos

A técnica da EM em blocos é também denominada “algoritmo de casamento de blocos” (BMA - *block matching algorithm*). Nessa técnica, os quadros são representados por um conjunto de blocos de tamanho  $N \times N$  que são copiados de quadros de referência anteriores ou posteriores. O principal objetivo da técnica é reduzir a redundância temporal minimizando o erro entre o quadro atual e aquele construído com blocos copiados de outros quadros de referência. A informação de movimento é representada através de vetores de movimento (VMs), que indicam o número de *pixels* que cada bloco é deslocado nas dimensões  $x$  e  $y$ . Através da translação de pequenos blocos é possível obter uma estimação de movimento adequada mesmo quando o movimento não é de translação [47], [17]. Os VMs obtidos podem ter discrepância em relação ao movimento real dos objetos [47], produzindo artefatos e distorções na imagem. Para reduzir tais efeitos, o erro de predição (DFD) é codificado e uma filtragem após a codificação é usada para minimizar o efeito de blocagem [17].

A avaliação do melhor VM para cada bloco é feita utilizando uma função distorção que mede o erro de predição. Para se obter o mínimo global dessa função é necessário avaliar todos os possíveis deslocamentos de blocos em todos os quadros de referência. O algoritmo de busca exaustiva (*full search* - FS) apesar da sua elevada complexidade computacional é muito usado na EM devido a sua implementação simples [100]-[103] e garantia de encontrar o mínimo global. Várias técnicas de procura rápida têm sido propostas [103]-[111], com o objetivo de reduzir o esforço computacional do BMA; no entanto, muitas dessas técnicas não garantem a convergência para o mínimo global da função distorção. Algumas dessas técnicas serão analisadas neste capítulo, quanto a sua complexidade computacional, entropia dos VMs e qualidade da imagem.

Técnicas baseadas nas características de múltiplas escala do movimento, como as hierárquicas e múltiplas grades, têm sido também propostas com o objetivo de reduzir a complexidade computacional e a discrepância dos VMs em relação ao movimento real dos objetos [47], [112]-[114].

Para melhorar a qualidade da EM os parâmetros do BMA podem ser alterados reduzindo o tamanho dos blocos, aumentando a precisão da EM (*subpixel*), aumentando o número de quadros de referência, ou ainda alterando o tipo de interpolação utilizada para obter os valores de *subpixel*. Esse ajuste pode resultar em melhoria da qualidade da imagem predita, mas também em aumento de complexidade computacional e da entropia dos VMs. A escolha de uma adequada configuração desses parâmetros é necessária em um processo de otimização que leve em conta a taxa e a distorção [115].

### 3.3 Funções de Distorção

Para definir as funções de distorção que serão usadas neste trabalho são necessárias algumas definições iniciais. A intensidade do *pixel* na posição  $p = (x, y)$  para o instante de tempo  $t$  é dada por  $I(p, t)$ , sendo cada *pixel* representado por um conjunto de 3 valores para imagens coloridas. O VM, que indica o deslocamento durante o intervalo de tempo  $\Delta t$  entre quadros, é dado por  $d = (\Delta x, \Delta y)$ . O erro de predição para cada *pixel* é definido por

$$e(p, t, d) = I(p, t) - I(p - d, t - \Delta t). \quad (16)$$

A diferença absoluta e a energia dos  $N$  *pixels* dos blocos podem ser consideradas em termos de soma ou valores médios. Diferentes funções de distorção podem ser usadas para medir o grau de aproximação entre os blocos avaliados, entre as quais temos:

- i) soma das diferenças absolutas (*sum of absolute differences* – SAD)

$$SAD(d, t) = \sum_{p \in B} |e(p, t, d)| \quad (17)$$

onde  $B$  é o conjunto de *pixel* que compõe o bloco sob avaliação.

- ii) média das diferenças absolutas (*mean absolute differences* – MAD)

$$MAD(d,t) = \frac{SAD(d,t)}{N} \quad (18)$$

- iii) soma das diferenças quadráticas (*sum of squared differences* – SSD)

$$SSD(d,t) = \sum_{p \in B} e(p,t,d)^2 \quad (19)$$

- iv) média das diferenças quadráticas (*mean squared differences* – MSD)

$$MSD(d,t) = \frac{SSD(d,t)}{N} \quad (20)$$

- v) se a distorção precisa ser avaliada no domínio transformado, a predição pode ser obtida pela soma das diferenças absolutas transformadas (*sum of absolute transformed differences* – SATD) definida como

$$SATD(d,t) = \sum_{p \in B} |T[e(p,t,d)]| \quad (21)$$

onde  $T[\cdot]$  denota uma transformação 2-D ortonormal.

Em diversas avaliações, a qualidade da imagem é medida através da razão sinal-ruído de pico (*peak signal-to-noise ratio* – PSNR) dos componentes de luminância e crominâncias. Quando a imagem é representada com  $N$  bits, a PSNR de cada componente é obtida por

$$PSNR = 10 \log_{10} \left( \frac{(2^N - 1)^2}{MSD} \right) \quad (22)$$

onde  $2^N - 1$  é o valor máximo do componente de luminância ou crominância.

Na determinação da qualidade da imagem total em PSNR foi utilizada a média ponderada das PSNR de cada componente. Assim,

$$PSNR_{YCbCr} = \frac{4PSNR_Y + PSNR_{Cb} + PSNR_{Cr}}{6}. \quad (23)$$

### 3.4 Parâmetros que Afetam a Qualidade do BMA

Nesta seção são analisados os principais parâmetros que influenciam o desempenho da BMA, dentre eles: tamanho de bloco, janela de procura, precisão do VM e a estratégia de procura considerada. Cada um desses parâmetros é analisado separadamente visando determinar a sua influência sobre a qualidade, complexidade computacional e taxa de bits. Para a obtenção dos resultados são utilizadas simulações com sequências de imagens no padrão QCIF, representadas no espaço de cores  $YC_B C_R$  [19] usando-se 8 bits para cada componente de cor. Nas simulações, apenas um dos parâmetros é modificado, enquanto os demais são mantidos fixos. Os valores *default* desses parâmetros são: tamanho de bloco  $8 \times 8$ ; janela de procura  $\pm 7$  *pixels*; estratégia de procura FS e resolução  $\pm 1$  *pixel*. Nestas avaliações se considera que a codificação é sem perda, isolando assim o efeito das outras etapas do codificador de vídeo sobre o processo de EM. Para tal, os quadros de referência utilizados são sempre iguais aos originais.

A qualidade da imagem é medida em PSNR dos componentes  $Y$ ,  $C_B$  e  $C_R$ . Em alguns casos, após a compensação de movimento, uma análise subjetiva da imagem ou da DFD é efetuada com o objetivo de ressaltar algumas características do processo de EM.

A taxa de bits necessária para a transmissão dos vetores de movimento  $D_{VM}$  é estimada considerando um codificador de entropia ideal. Assim,

$$D_{VM} = \frac{H_{VM} n_{VM}}{n_{pixel}} \quad (24)$$

onde  $H_{VM}$  é a entropia dos VMs,  $n_{VM}$  é o número de VM do quadro e  $n_{pixel}$  é o número de *pixels* do quadro. A determinação da entropia dos VMs é realizada através da entropia empírica obtida pelo histograma dos VMs de cada quadro.

#### 3.4.1 Tamanho do Bloco

O grau de influência do tamanho do bloco empregado na EM foi avaliado variando o tamanho entre  $16 \times 16$  e  $2 \times 2$ . Nessas avaliações foram usados os quadros #0 a #100 da sequência *MissAmerica*. No desempenho de qualidade quadro-a-quadro mostrado na Figura 10, é possível notar que ocorre uma melhoria significativa da qualidade com a redução do tamanho do bloco. Nos quadros com movimentos mais

amplios o ganho em qualidade para blocos  $2 \times 2$ , com respeito aos blocos  $16 \times 16$ , chega a 3 dB.

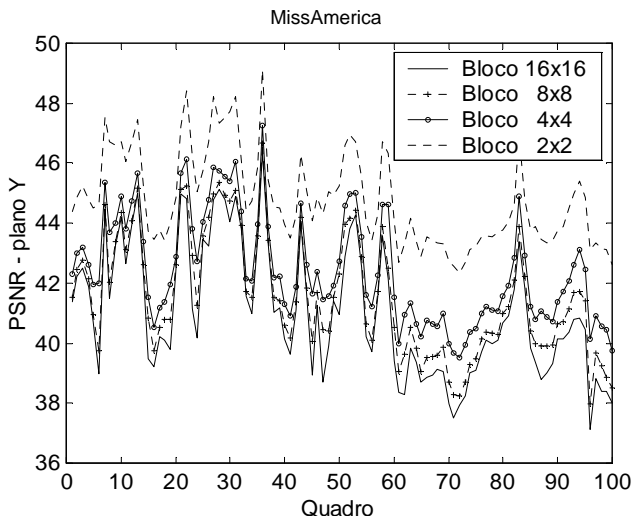


Figura 10 - PSNR para CM com blocos de  $16 \times 16$ ,  $8 \times 8$ ,  $4 \times 4$  e  $2 \times 2$ .

O principal motivo do aumento na qualidade da imagem em função da redução do tamanho do bloco é a melhoria da compensação de movimento nas bordas dos objetos móveis. Isso pode ser percebido comparando as imagens das DFDs<sup>1</sup> mostradas na Figura 11. Somente quando o tamanho do bloco é menor do que  $4 \times 4$  ocorre uma adequada compensação do movimento dos olhos da interlocutora. Com tamanhos maiores de blocos a compensação de movimento é eficiente apenas para o movimento da cabeça e ombros.

Os principais inconvenientes do uso de pequenos blocos na CM são o aumento exponencial da quantidade de VMs, o aumento da entropia dos VMs (ver Figura 12), e o aumento de complexidade de procura. O aumento da entropia dos VMs ocorre devido à melhor descrição do movimento dos objetos, podendo resultar também em um aumento da discrepância entre os VMs e o movimento real dos objetos. A solução geralmente usada para evitar esse aumento de entropia é adotar um esquema de EM com blocos de tamanho variável [16], [17], [116]-[118], no qual áreas planas são compensadas com o uso de blocos

<sup>1</sup> Na avaliação das imagens de DFD deve-se levar em conta que as diferenças nulas entre *pixel* são representadas pela cor cinza média e as diferenças positivas e negativas entre *pixel* são representadas por desvios em direção ao branco e preto, respectivamente.

de maior tamanho, enquanto bordas com movimento são descritas através de blocos menores.

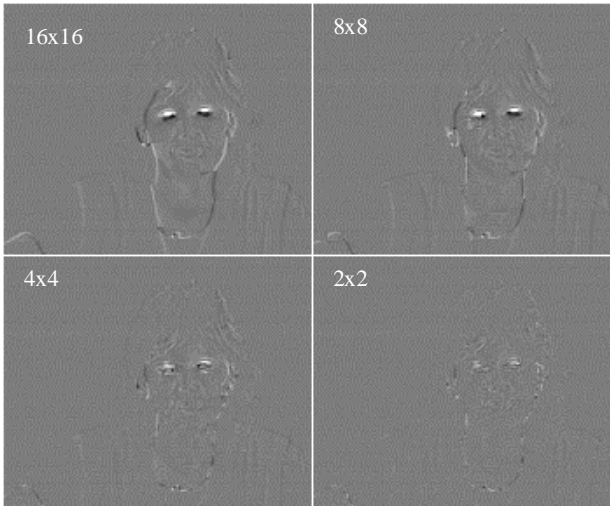


Figura 11 - Imagens da DFD do quadro #96 para CM com blocos de  $16 \times 16$ ,  $8 \times 8$ ,  $4 \times 4$  e  $2 \times 2$ .

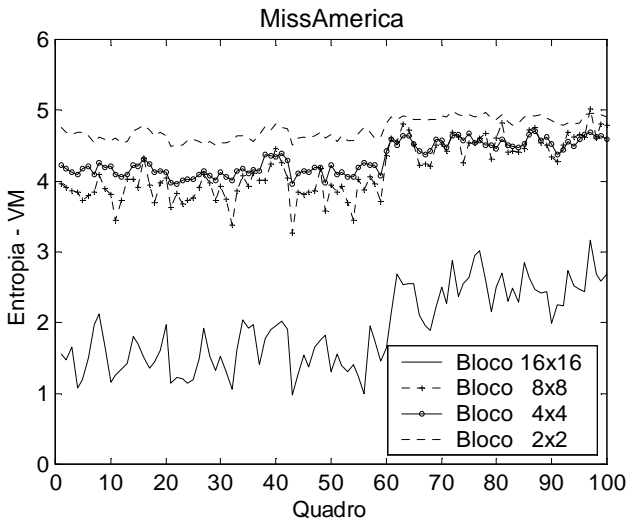


Figura 12 - Entropia média dos VMs para CM com blocos de  $16 \times 16$ ,  $8 \times 8$ ,  $4 \times 4$  e  $2 \times 2$ .

### 3.4.2 Tamanho da Janela de Procura

A janela de procura de um algoritmo BMA corresponde à região dentro do quadro de referência na qual é feita a procura do bloco que mais se aproxima do bloco de imagem em questão no quadro atual. A janela de procura define a variação máxima do valor do VM, ou seja, uma janela de  $\pm N$  pixels indica que o VM pode ter seu componente horizontal e vertical variando no intervalo  $[-N, +N]$ , existindo, portanto,  $(2N + 1)^2$  possíveis candidatos considerando uma resolução de  $\pm 1$  pixel.

Nesta avaliação, a janela de procura foi variada entre  $\pm 2$  e  $\pm 15$  pixels, sendo usados os quadros #0 a #100 da sequência *Suzie*. Essa sequência foi escolhida, pois exhibe trechos com pouco movimento (TRECHO1 entre os quadros #0 e #42 e TRECHO3 entre os quadros #80 e #100), e um trecho com movimentos amplos e rápidos (TRECHO2 entre os quadros #42 e #80). Os resultados de desempenho de qualidade obtidos (ver Figura 13) mostram que a qualidade da imagem praticamente não é influenciada pelo tamanho da janela de procura nos trechos com pouco movimento (TRECHO1 e 3). No trecho com movimentos amplos e rápidos (TRECHO2) a janela de procura maior resultou em uma melhor compensação de movimento, com um ganho médio de 2,9 dB para a janela de  $\pm 15$  pixels.

Comparando as imagens das DFDs mostradas na Figura 14, nota-se que a janela de procura de  $\pm 2$  pixels não é suficiente para se compensar os movimentos no quadro #52. O movimento da cabeça começa a ser compensado com uma janela de  $\pm 4$  pixels; no entanto, a adequada compensação de movimento da mão e monofone apenas ocorrem usando uma janela de no mínimo  $\pm 7$  pixels.

O principal inconveniente do aumento da janela de procura é a complexidade computacional envolvida. Por isso, o ideal é que a janela de procura esteja ajustada à amplitude dos movimentos que ocorrem entre os quadros. Geralmente se adota uma janela de procura fixa, implicando um desperdício de tempo computacional nos trechos ou blocos com pouco movimento. Uma solução para esse problema é o uso de algoritmos rápidos que limitem o número de posições de procura [119].

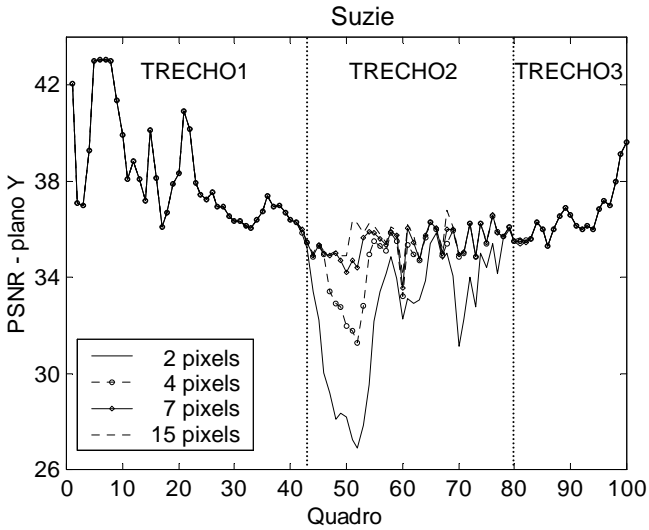


Figura 13 - PSNR para CM com janelas de procura de  $\pm 2$ ,  $\pm 4$ ,  $\pm 7$  e  $\pm 15$  pixels.

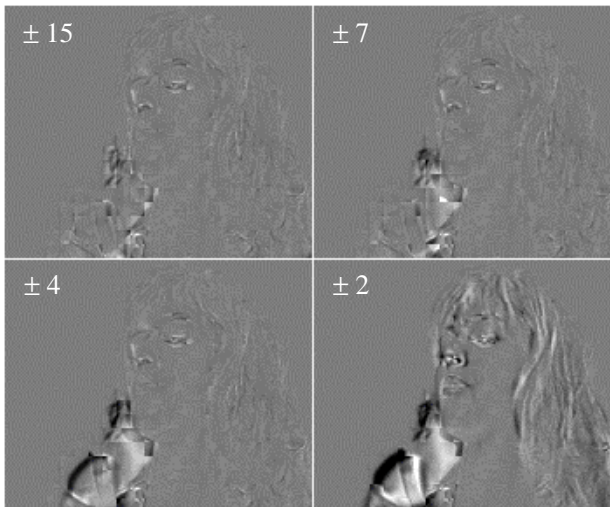


Figura 14 - Imagem da DFD do quadro #52, com janelas de procura de  $\pm 2$ ,  $\pm 4$ ,  $\pm 7$  e  $\pm 15$  pixels.



### 3.4.3 Precisão de Representação dos VMs e Interpolação

O aumento da precisão de representação dos VMs é conseguido através da interpolação dos valores entre *pixels* do quadro de referência antes de realizar a EM. O método de interpolação usado afeta diretamente a qualidade da EM e também a complexidade de procura. Nesta seção são consideradas as precisões de  $1/2$  e  $1/4$  *pixel* usando interpolação bilinear e bicúbica [120]. Estudos com outras formas de interpolação e diferentes precisões de representação dos VMs foram realizados durante o desenvolvimento dos modelos do teste do H.264 [28], mostrando resultados similares, levando a adoção da precisão de  $1/4$  *pixel* nesse padrão.

O aumento da precisão da EM de 1 *pixel* para  $1/2$  *pixel* quadruplica a complexidade da procura e, o aumento para  $1/4$  *pixel* aumenta a complexidade em 16 vezes. Para lidar com tal problema, geralmente se consideram monotônicas as variações em torno do ponto de mínimo da função distorção obtido com resolução de 1 *pixel* e se realiza um refinamento de EM de *subpixel*. A técnica de refinamento geralmente usada consiste da: (i) avaliação dos 8 VMs com variação de  $1/2$  *pixel* em relação ao melhor VM com resolução de 1 *pixel* e seleção do melhor VM; (ii) repetição do processo com resolução de  $1/4$  *pixel* em torno do melhor VM da etapa anterior.

Note que a utilização de um algoritmo FS com precisão de  $1/4$  *pixel*, no lugar da técnica de refinamento anteriormente descrita, resulta em aumento da qualidade da imagem nos blocos cuja função de distorção não é monotônica em torno do mínimo. No entanto, o uso de tal algoritmo FS, resultaria em um grande aumento de complexidade computacional, além de resultar em um campo de VMs menos uniforme e menos adequado a codificação.

A Figura 15 mostra os resultados de qualidade obtidos variando a precisão da EM para a sequência *MissAmerica* com blocos  $8 \times 8$  e janela de  $\pm 7$  *pixels*. Note que o aumento da precisão na EM produz um aumento significativo de qualidade. Um ganho médio de 1,9 dB é obtido com a precisão de  $1/2$  *pixel* usando interpolação bilinear e 2,1 dB com interpolação bicúbica. Aumentando a precisão para  $1/4$  *pixel*, os ganhos médios passam a ser, respectivamente, 3,0 dB e 3,5 dB. No quadro #73 a melhoria de qualidade atinge o valor de 6,3 dB.

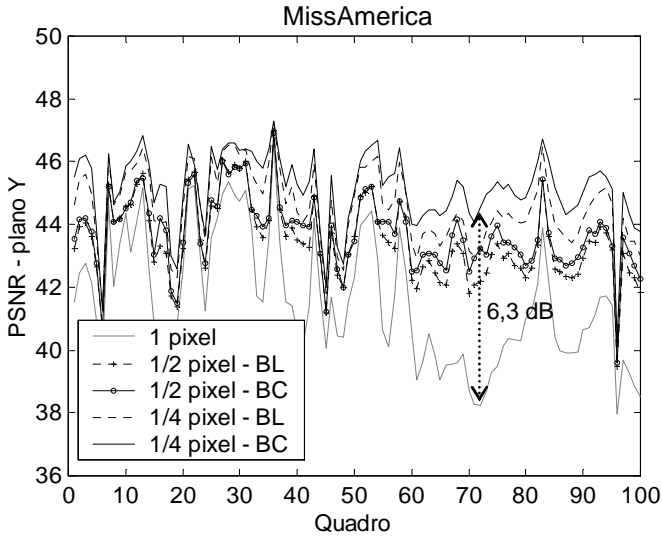


Figura 15 - PSNR do componente Y para CM com precisão de 1, 1/2 e 1/4 *pixel*, com interpolação bilinear (BL) e interpolação bicúbica (BC).

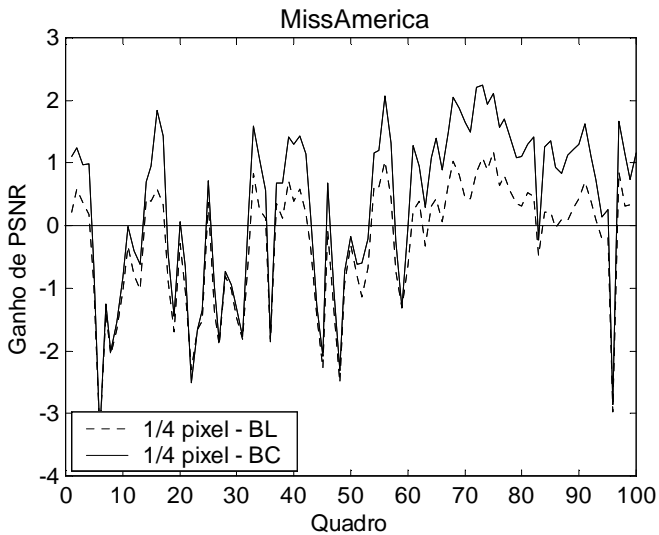


Figura 16 - Ganho de PSNR para CM com blocos  $8 \times 8$  e precisão de 1/4 *pixel* em relação a blocos  $2 \times 2$  e precisão de 1 *pixel*, com interpolação bilinear (BL) e interpolação bicúbica (BC).

Esta melhoria de qualidade obtida com o aumento da precisão para  $1/4$  *pixel* na EM (ver Figura 15) é superior àquela obtida reduzindo o tamanho do bloco para  $2 \times 2$  *pixels* (ver Figura 10). As diferenças de qualidade entre as duas situações são mostradas na Figura 16. Note que na maioria dos quadros (64%), o uso de precisão maior na EM resulta em maior qualidade que a diminuição do tamanho do bloco.

A influência da precisão, do tipo de interpolação usada e do tamanho do bloco na entropia dos VMs é mostrada na Tabela 4. Os resultados obtidos mostram que para uma qualidade semelhante em PSNR, a entropia dos VMs é menor no caso de blocos  $8 \times 8$  com precisão de  $1/4$  *pixel* do que para blocos  $2 \times 2$  com precisão de 1 *pixel*. Esses resultados apontam em direção contrária a adotada pelos codificadores de vídeo que reduzem o tamanho dos blocos sem primeiro aumentar a precisão da EM [16] e [116]. Por outro lado, para a mesma resolução a interpolação bicúbica resulta em qualidade de imagem superior à obtida com a interpolação bilinear. Por isso se conclui que o tipo de interpolação a ser usado e a precisão da mesma, têm importância fundamental no desempenho RD dos codificadores de vídeo.

Tabela 4 – Entropia média por *pixel* dos VMs e qualidade em PSNR

Precisão em <i>pixels</i>	Tipo de interpolação	bloco $2 \times 2$		bloco $4 \times 4$		bloco $8 \times 8$		bloco $16 \times 16$	
		entropia	PSNR	entropia	PSNR	entropia	PSNR	entropia	PSNR
1	-	1,184	44,80	0,270	42,46	0,065	41,58	0,007	41,06
$1/2$	<b>Bilinear</b>	1,517	46,59	0,392	44,34	0,091	43,46	0,014	42,86
$1/2$	<b>Bicúbica</b>	1,542	46,82	0,391	44,59	0,091	43,72	0,014	43,11
$1/4$	<b>Bilinear</b>	1,792	48,01	0,505	45,58	0,117	44,55	0,020	43,85
$1/4$	<b>Bicúbica</b>	1,836	48,66	0,505	46,20	0,117	45,14	0,020	44,39

\* Valores obtidos com algoritmo FS para os quadros  $Q_0$  a  $Q_{100}$  da sequência *MissAmerica*.

### 3.4.4 Estratégias Rápidas de EM

As estratégias rápidas de EM [104]-[109] são algoritmos que visam reduzir a complexidade computacional do processo de EM com um comprometimento mínimo da qualidade da imagem obtida em comparação com o algoritmo FS. A maioria dessas estratégias restringe o número de posições de procura (NPP) baseada nas premissas: (i) monotonicidade da função distorção; (ii) distribuição centralizada dos mínimos da função distorção. Tais premissas podem ser consideradas válidas em sequências de vídeo que empregam câmera fixa e têm poucos objetos em movimento, por exemplo, aplicações de

videoconferência e monitoração de ambientes. No entanto, em casos nos quais essas premissas não são válidas a principal consequência do uso das estratégias rápidas é uma redução de qualidade da imagem.

Para determinar o efeito da estratégia usado na EM, foram avaliadas as estratégias frequentemente citadas na literatura tais como: TSS (*three step search*) [105], NTSS (*new three step search*) [106], 4SS (*four step search*) [107], BBGDS (*block-based gradient descent search*) [108], 2-DLS (*2-dimensional logarithmic search*) [104], DSS (*diamond shape search*) [100] e UHexagonS [109]. Nas comparações, usou-se como referência de qualidade o algoritmo FS e também como limite de qualidade mínima o “algoritmo nulo” (AN), que não realiza qualquer CM (cópia do quadro anterior). A estratégia TSS é frequentemente usada para codificadores DO devido à sua simplicidade e seu bom desempenho. As estratégias NTSS e 4SS são um aprimoramento que explora a concentração dos VMs em torno da posição central. Baseadas na mesma premissa concentração dos VMs as estratégias BBGDS e DSS reduzem ainda mais a complexidade de EM.

A comparação das diferentes estratégias foi realizada usando os 100 quadros iniciais das sequências: *MissAmerica*, *Akiyo*, *Suzie*, *Salesman*, *Carphone* e *Foreman* em formato QCIF a uma taxa de quadros de 10 fps. Os resultados de qualidade medidos em PSNR são apresentados na Tabela 5. Note que todas as estratégias rápidas avaliadas trazem como consequência uma redução de qualidade em relação ao FS, sendo as estratégias NTSS e BBGDS as que apresentam a menor perda de qualidade.

Tabela 5 – Valores médios de PSNR e variação de PSNR em relação ao algoritmo FS-DO

Sequência	PSNR ( $\Delta$ PSNR)					
	FS-DO	TSS	4SS	NTSS	BBGDS	DSS
<i>MissAmerica</i>	41,57	41,21 (0,36)	41,27 (0,3)	41,53 (0,04)	41,52 (0,05)	41,46 (0,11)
<i>Akiyo</i>	44,61	44,5 (0,11)	44,52 (0,09)	44,61 (0,00)	44,61 (0,00)	44,59 (0,02)
<i>Suzie</i>	36,87	36,18 (0,69)	36,34 (0,53)	36,68 (0,19)	36,59 (0,28)	36,25 (0,62)
<i>Salesman</i>	40,87	40,66 (0,21)	40,74 (0,13)	40,84 (0,03)	40,83 (0,04)	40,79 (0,08)
<i>Carphone</i>	35,25	34,62 (0,63)	34,76 (0,49)	35,08 (0,17)	35,04 (0,21)	34,84 (0,41)
<i>Foreman</i>	34,44	41,21 (0,36)	41,27 (0,3)	41,53 (0,04)	41,52 (0,05)	41,46 (0,11)
<b>Média <math>\Delta</math>PSNR</b>		0,51	0,39	0,12	0,15	0,34

Para ilustrar a variação de desempenho quadro-a-quadro das estratégias rápidas em relação ao FS, são mostrados na Figura 17 os resultados para a sequência *Suzie*. Note que a maior perda de qualidade ocorre nos trechos em que acontecem movimentos mais amplos (TRECHO2). Isto acontece porque em muitos blocos desse trecho, a premissa da monotonicidade da função distorção não é verdadeira e consequentemente o VM selecionado pela estratégia rápida é distante do mínimo global da função distorção. Nesse caso a estratégia NTSS é a que resulta na menor perda de qualidade, devido ao uso de um gabarito de procura inicial com resolução de 4 *pixels* [106].

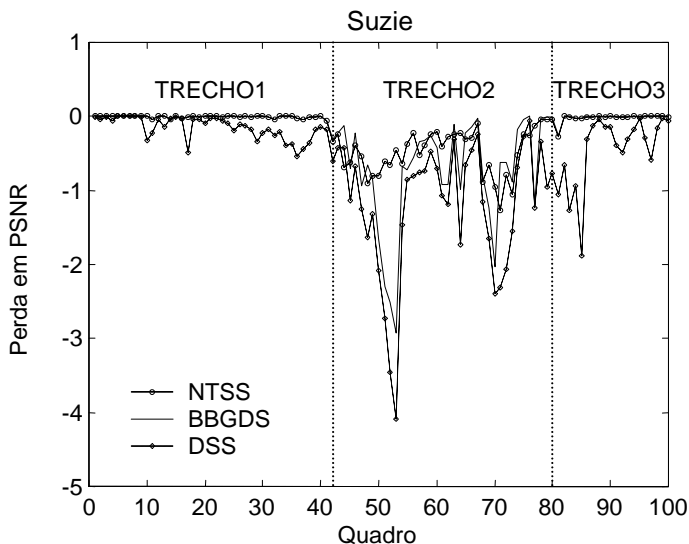


Figura 17 - Perda em qualidade das estratégias rápidas NTSS, BBGDS e DSS em relação ao algoritmo FS-DO.

A redução de complexidade obtida com o uso destas estratégias pode ser estimada através do número de posições de procura (NPP) por bloco que cada algoritmo analisa (ver Tabela 6). Os algoritmos DSS e BBGDS apresentam a menor complexidade, a qual é consequência da maior rapidez na determinação de ausência de movimento.

Tabela 6 – Número médio de posições de procura por bloco e percentual médio de redução de complexidade em relação ao algoritmo FS-DO

Sequência	NPP (% redução complexidade)					
	FS-DO	TSS	4SS	NTSS	BBGDS	DSS
<i>MissAmerica</i>	225	25 (11,1%)	18,1 (8%)	19,5 ( 8,7%)	11 (4,9%)	6,7 (3,0%)
<i>Akiyo</i>	225	25 (11,1%)	16,5 (7,3%)	16,5 (7,3%)	8,5 (3,8%)	4,5 (2,0%)
<i>Suzie</i>	225	25 (11,1%)	18,1 (8,0%)	19,4 (8,6%)	11,6 (5,2%)	7,1 (3,2%)
<i>Salesman</i>	225	25 (11,1%)	17,1 (7,6%)	17,2 (7,6%)	9,2 (4,1%)	5,2 (2,3%)
<i>Carphone</i>	225	25 (11,1%)	18,3 (8,1%)	20 (8,9%)	11,7 (5,2%)	7,4 (3,3%)
<i>Foreman</i>	225	25 (11,1%)	18,1 (8,0%)	19,5 (8,7%)	11 ( 4,9%)	6,7 (3,0%)
<b>% média</b>		11,1%	7,9%	8,3%	4,7%	2,8%

A taxa de bits usada na codificação dos VMs é mostrada na Tabela 7. Note que dentre as estratégias avaliadas, a DSS foi aquela que resultou na maior redução de entropia dos VMs. O favorecimento dos VMs de menor custo de codificação nas estratégias que consideram a premissa da distribuição centralizada dos VMs é o principal motivo da diferença de entropia das várias estratégias em relação ao FS. O algoritmo FS sempre resulta na maior entropia, pois seleciona o VM de menor distorção. Note que por isso, a melhor estratégia de EM para um codificador de vídeo nem sempre (ou quase nunca) é a FS-DO, quando a otimização é apenas em distorção.

Tabela 7 – Taxa de bits para a codificação dos VMs e percentagem de redução da entropia em relação ao FS-DO

Sequência	Entropia x 1000 (% redução)					
	FS-DO	TSS	4SS	NTSS	BBGDS	DSS
<i>MissAmerica</i>	66,2	49,6 ( 74,9%)	44,2 ( 66,8%)	40,5 ( 61,2%)	35,2 ( 53,2%)	31,8 ( 48%)
<i>Akiyo</i>	1,5	1,3 ( 86,7%)	1,3 ( 86,7%)	1,2 ( 80,0%)	1,2 ( 80,0%)	1,2 ( 80,0%)
<i>Suzie</i>	40,1	38,6 ( 96,3%)	36,5 ( 91,0%)	34,4 ( 85,8%)	34 ( 84,8%)	30,8 ( 76,8%)
<i>Salesman</i>	7,2	6,1 ( 84,7%)	5,9 ( 81,9%)	5,7 ( 79,2%)	5,6 ( 77,8%)	5,4 ( 75,0%)
<i>Carphone</i>	57,2	48 ( 83,9%)	44,4 ( 77,6%)	41,6 ( 72,7%)	38,8 ( 67,8%)	35,2 ( 61,5%)
<i>Foreman</i>	48,8	48,6 ( 99,6%)	44,6 ( 91,4%)	37,5 ( 76,8%)	35,2 ( 72,1%)	28 ( 57,4%)
<b>% média</b>		87,7%	82,6%	76,0%	72,6%	66,5%

\* Entropia obtida dividindo a taxa de bits pelo número de bits do bloco usado na CM. Valores multiplicados pelo fator 1000.

A escolha da melhor estratégia de EM depende dos seguintes fatores: (i) capacidade computacional do sistema no qual a estratégia está implementada; (ii) taxa de bits disponível (iii) tipo de aplicação. Em algumas situações a estratégia que apresenta um pior desempenho na etapa de CM pode levar a um melhor compromisso taxa-distorção, devido à menor entropia dos seus VMs. Assim, a melhor opção não é necessariamente aquela que apresenta a melhor qualidade de imagem após a CM. Na maioria dos casos, principalmente quando baixas taxas de bits são consideradas, é possível obter para uma mesma taxa de bits uma qualidade muito superior ao algoritmo FS-DO, usando um processo de otimização RD, conforme será mostrado no próximo capítulo.

### 3.5 Compensação de Movimento da Crominância

Nos padrões de vídeo atuais a CM dos componentes de crominância é sempre realizada de forma conjunta com o componente de luminância, usando para tal os VMs obtidos através da EM realizada na luminância. Nos casos em que os componentes de crominância são amostrados, os VMs empregados na CM da crominância são escalados de acordo com o tipo de amostragem usada [16]. Nas avaliações realizadas nesta seção, empregamos sequências de vídeo em formato QCIF, com espaço de cores  $YC_B C_R$  e amostragem 4:2:0 [19], [121].

Nesta seção, são analisadas duas possibilidades de realização da CM para os componentes  $C_B$  e  $C_R$ . A abordagem  $CM_1$  para compensação de movimento dos componentes  $C_B$  e  $C_R$  é realizada usando o valor médio dos VMs obtidos para o componente Y, conforme Anexo F do padrão H.263 [16]. Na segunda abordagem  $CM_2$ , que é adotada no padrão H.264, a compensação de movimento dos blocos  $N/2 \times N/2$  pixels dos componentes  $C_B$  e  $C_R$  é realizada usando a metade do valor dos VMs obtidos para Y. Os resultados da Tabela 8 mostram que a abordagem  $CM_2$  apresenta um significativo ganho de qualidade sobre a abordagem  $CM_1$  [82]. Como a informação usada nas duas abordagens é a mesma esse ganho é conseguido sem nenhum acréscimo de bits.

Tabela 8 – Comparação da qualidade em PSNR da CM dos componentes de crominâncias  $C_B$  e  $C_R$  usando as abordagens  $CM_1$  e  $CM_2$ 

Sequência	PSNR-Y	Compensação de movimento $CM_1$			Compensação de movimento $CM_2$		
		PSNR- $C_B$	PSNR- $C_R$	PSNR	PSNR- $C_B$	PSNR- $C_R$	PSNR
<i>MissAmerica</i>	42,50	39,88	43,35	42,20	39,97	43,79	42,29
<i>Suzie</i>	39,30	52,99	52,69	43,81	53,12	52,96	43,88
<i>Mother&amp;Daughter</i>	37,60	44,23	44,78	39,90	45,07	45,92	40,23
<i>Salesman</i>	39,44	46,13	48,26	42,03	46,37	48,56	42,12
<i>Carphone</i>	33,90	45,58	46,38	37,93	46,98	47,45	38,34
<i>Foreman</i>	34,11	43,67	45,17	37,55	44,73	45,95	37,85

\* Valores obtidos para os quadros #0 a #25 de cada sequência.

### 3.6 Conclusões

Neste capítulo, o processo de estimação de movimento do tipo BMA foi analisado com o objetivo de avaliar a influência de cada parâmetro do processo de EM no desempenho e complexidade. Foram realizadas avaliações alterando os parâmetros: tamanho de bloco, janela de procura, precisão da representação do movimento, estratégia de procura dos VMs e forma de representação da cor. Para que o processo de EM pudesse ser avaliado independentemente das demais etapas do codificador de vídeo, a análise desses parâmetros foi realizada com um codificador otimizado em distorção sem perdas. Nas avaliações foram considerados os quesitos: qualidade objetiva (PSNR), qualidade subjetiva, complexidade computacional e taxa de bits da informação de movimento. Dentre os resultados obtidos destacamos os de maior importância. Em primeiro lugar, se constatou que existe uma considerável influência do tamanho do bloco sobre a qualidade, principalmente, nas bordas dos objetos em movimento. Quanto menor o tamanho do bloco usado melhor é a qualidade; no entanto, o custo de codificação dos VMs e a complexidade computacional do processo de EM aumentam demasiadamente à medida que o tamanho do bloco é reduzido. Para contornar tal problema, frequentemente são usados diferentes tamanhos de blocos conforme a quantidade de movimento da cena. Nos próximos capítulos esse tipo de solução é analisado no contexto de um codificador de vídeo completo que adota blocos de tamanho de variável. Em segundo lugar, o tamanho da janela de procura tem uma influência “exponencial” sobre a complexidade computacional



dos algoritmos de EM-BMA. No entanto, o aumento da janela de procura só resulta em melhoria de qualidade se na cena considerada existirem movimentos reais de amplitude que a justifiquem. Na maioria dos casos, o aumento da janela não acarreta ganho significativo na qualidade de imagem codificada; porém, se ela for menor do que a amplitude dos movimentos ocorrerá uma expressiva perda de qualidade. Por isso, o ideal é utilizar um tamanho de janela que esteja em conformidade com a amplitude dos movimentos na cena. No Capítulo 6, o problema de complexidade exponencial é tratado com o emprego de algoritmos rápidos, os quais permitem resolvê-lo de maneira satisfatória. Em terceiro lugar, também se constatou que o aumento da precisão dos VMs resulta em uma significativa melhoria de qualidade da imagem com um modesto aumento no custo de codificação dos VMs. Verificou-se que o tipo de interpolação usado na determinação dos valores de *subpixel* influencia significativamente à qualidade da CM. Assim, se concluiu que no processo de codificação o aumento da precisão dos VMs deve ser avaliado antes de se reduzir o tamanho dos blocos usados, pois resulta em um menor acréscimo de complexidade computacional e taxa de bits. Ressaltamos que muitos trabalhos da literatura desconsideram por completo tal influência, deixando inclusive de mencionar a precisão usada na CM na descrição dos resultados.

A avaliação dos diferentes algoritmos rápidos (ARs) de EM sob condições de otimização de distorção, mostrou que todos os ARs apresentam perda de qualidade em relação ao algoritmo FS-DO. No entanto, em termos de desempenho RD, a redução no custo de codificação dos VMs obtido com os ARs, muitas vezes conduz a uma solução melhor do que o próprio algoritmo FS-DO.

Quanto à forma de CM da crominância, concluímos que a compensação de movimento conjunta é a solução mais adequada. Quando as grades de amostragem usadas para os componentes de crominância e luminância são diferentes (tal como na amostragem 4:2:0), é necessário que os VMs utilizados na CM da crominância sejam ajustados, mantendo o mesmo número de blocos de crominância e luminância [82].



## Capítulo 4

### Otimização em Taxa-Distorção

Geralmente a codificação de vídeo tem como objetivo obter a menor taxa de bits possível mantendo uma qualidade satisfatória da imagem. A complexidade computacional dos atuais codificadores tem crescido a cada novo padrão surgido, devido à existência de muitos parâmetros do processo de codificação que precisam ser ajustados. Além dos fatores discutidos no capítulo anterior relacionados ao processo de EM, a taxa de compressão e a qualidade da imagem que podem ser obtidos dependem ainda da correta configuração das diversas etapas do codificador, tais como compensação de movimento, redução da redundância espacial com transformações 2-D, codificação do erro de previsão, quantização, codificação de entropia e codificação da cor. A otimização dessas etapas pode ser feita considerando como critério apenas a distorção ou, adicionalmente, a taxa de bits. Em aplicações de VLBR, tais como videoconferência, videotelefonia e monitoração de ambientes, nas quais a taxa de bits é limitada, a otimização apenas da distorção implica perda de qualidade final da imagem. Apesar disso, diversos trabalhos [104]-[108], [122] consideravam a distorção como critério único de decisão na escolha dos VMs durante a etapa da EM e na avaliação de codificadores, levando a conclusões equivocadas sobre desempenho dos codificadores. Em tais trabalhos, o algoritmo FS-DO é frequentemente considerado como referência ótima para fins de comparação. Por outro lado, os trabalhos [123]-[133] que buscam um compromisso adequado entre taxa e distorção, produzem sempre uma menor distorção para uma determinada taxa de bits. Dentre esses trabalhos, o uso dos multiplicadores de Lagrange na otimização do compromisso taxa-distorção tem apresentado bons resultados [115], [123]-[131].

O objetivo geral deste capítulo é avaliar o ganho de desempenho obtido pelos codificadores que utilizam um processo de EM otimizado em RD (denominados aqui RDO), assim como estudar o uso dos multiplicadores de Lagrange na otimização da escolha do VM e modo de codificação. A Seção 4.1 compara os resultados obtidos usando apenas a otimização em distorção (DO) com aqueles obtidos usando RDO, apresentando uma interpretação gráfica do processo RDO de

escolha dos VMs. Na Seção 4.2, é apresentada uma proposta de medida de desempenho RD baseada no custo lagrangeano para a avaliação quadro-a-quadro do desempenho RD. Na Seção 4.3, uma interpretação gráfica do processo de otimização RD de seleção do modo de codificação é apresentada.

#### **4.1 Otimização da Escolha dos VMs**

Com o objetivo de ilustrar a necessidade da otimização em RD, a diferença de desempenho entre algoritmos de estimação de movimento DO e RDO é avaliada. A Figura 18 mostra os valores de taxa e distorção obtidos para ambos os algoritmos utilizando o modelo de teste JM7.3 [91], habilitando apenas o modo de predição  $Inter_{8x8}$ . Nota-se que o algoritmo RDO apresenta um ganho significativo sobre o algoritmo DO em toda a faixa de taxa considerada. Testes com outras configurações e sequências de vídeo mostram que a otimização em RD sempre supera em desempenho RD os algoritmos DO.

##### **4.1.1 Formas de Avaliação de Desempenho**

Na literatura da área [125], [129], o desempenho de codificadores de vídeo é geralmente comparado usando curvas RD que consideram valores médios de taxa e distorção; no entanto, para a completa avaliação de um codificador é necessário que a qualidade seja avaliada quadro-a-quadro, pois uma série de quadros de baixa qualidade afeta consideravelmente a qualidade percebida pelo observador humano. Neste caso, para se ter uma medida objetiva, geralmente a comparação quadro-a-quadro da qualidade é feita em PSNR fixando a taxa de bits. No entanto, esse procedimento resulta em redução do desempenho RD do codificador, pois impede que a taxa de bits e a qualidade se ajustem ao conteúdo do sinal de vídeo.

A Figura 19 ilustra a variação quadro-a-quadro da distorção e taxa de bits para o par de pontos destacados na Figura 18. Na Figura 19(a) nota-se que para todos os quadros da sequência de vídeo o emprego da estratégia DO resulta em qualidade PSNR superior à estratégia RDO. Isso ocorre tanto na qualidade medida após a compensação de movimento, quanto na qualidade final considerando a codificação da DFD. Por outro lado, na Figura 19(b), note que esse ganho resultou em um exagerado aumento da taxa de bits. Esse aumento é consequência da seleção de VMs que minimizam a distorção sem levar em conta o seu custo de codificação.

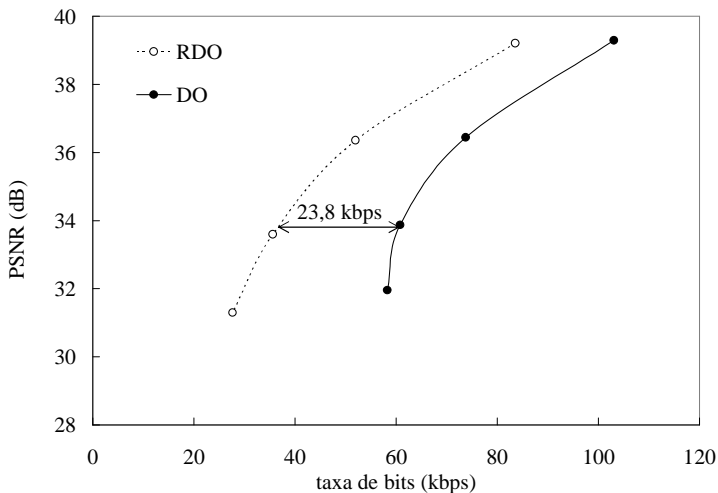


Figura 18 - Curvas taxa-distorção para a sequência *Mother&Daughter*. Valores obtidos com codificador JM7.3 usando somente o modo *Inter*<sub>8x8</sub> variando o parâmetro de quantização *Q* entre 24 e 36.

Note que a análise destas duas medidas não permite chegar a nenhuma comparação de desempenho RD. Por outro lado, as curvas RD (obtidos com valores médios, ver Figura 18) não revelam se há quadros com baixo desempenho RD dentro de uma sequência de vídeo. Uma solução adotada por muitos autores [95], [134]-[137] é fixar um dos fatores de avaliação (qualidade ou taxa de bits) enquanto mede-se o outro fator. O problema é que para que este método seja aplicado é necessário que o codificador permita controlar com precisão a taxa de bits ou qualidade, além disso, o uso desse controle geralmente resulta em um desempenho RD inferior ao obtido quando a taxa e a qualidade podem se ajustar quadro-a-quadro ao conteúdo visual [138].

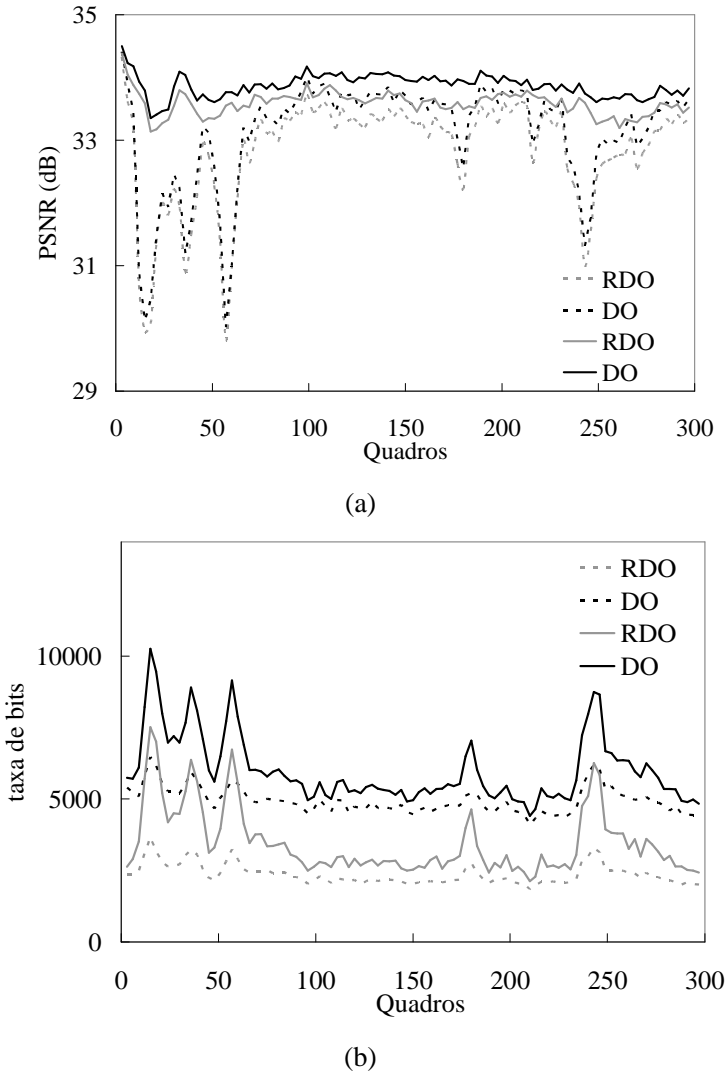
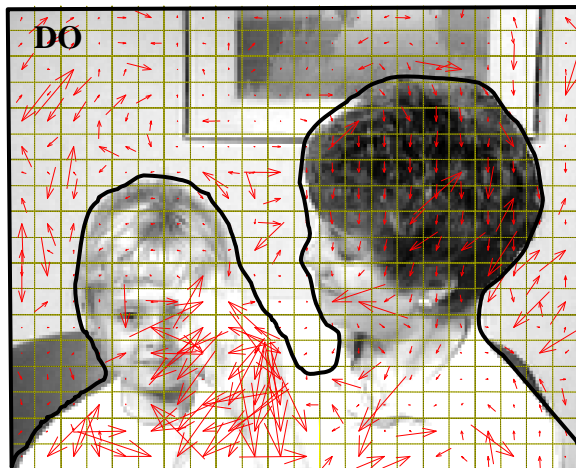
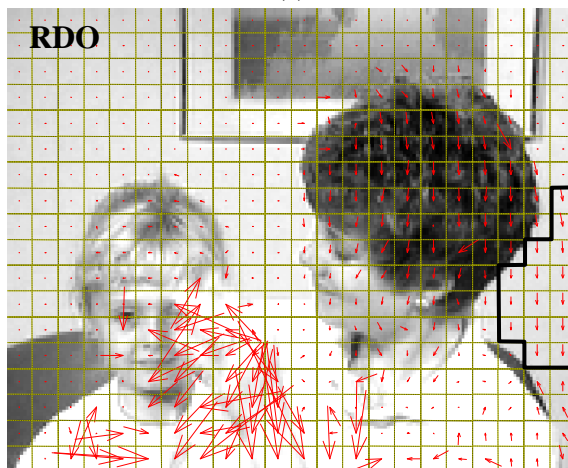


Figura 19 - Comparação de desempenho quadro-a-quadro das estratégias DO e RDO integradas ao codificador JM7.3 usando somente o modo  $Inter_{8 \times 8}$  e  $Q = 32$ . (Linha pontilhada) valores de PSNR (a) e taxa de bits (b) obtidos sem codificação da DFD. (Linha contínua) valores obtidos incluindo a codificação da DFD.



(a)



(b)

Figura 20 - Conjunto de vetores de movimento obtidos com o codificador JM7.3 usando apenas o modo  $Inter_{8 \times 8}$  para o quadro #15 da sequência *Mother&Daughter*. (a) Otimização em distorção. (b) Otimização em RD.

Em função da necessidade de avaliar o desempenho RD não apenas em valores médios finais principalmente durante o estudo das diversas possibilidades de codificação, é proposta na Seção 4.2 uma medida de desempenho que permite realizar a comparação

quadro-a-quadro sem a necessidade de fixar a taxa de bits ou qualidade. Assim cada proposta sob avaliação, é livre para se ajustar a taxa mais adequada ao conteúdo da cena, e não é necessário controlar com precisão a taxa de bits ou qualidade.

Para entender a grande diferença entre as estratégias DO e RDO, é ilustrado na Figura 20 o principal fator causador da diferença de desempenho em taxa de bits das estratégias. Nessa figura nota-se que a estratégia DO utiliza VMs não nulos em blocos nos quais não existe qualquer movimento, que minimizam a distorção em regiões cujas variações de *pixels* são causadas exclusivamente pelos ruídos de aquisição da imagem. Por outro lado, nota-se que na estratégia RDO os VMs não nulos são apenas usados em blocos com movimento real [ver Figura 20 (b)], com exceção de alguns blocos nos quais o VM predito é usado. Isso ocorre porque o VM no padrão H.264 é codificado através da diferença entre o VM predito e o VM real, assim o menor custo de codificação corresponde ao vetor de diferença nulo. Outra consequência da estratégia RDO é a obtenção de um conjunto de VMs mais uniforme e, portanto de menor custo de codificação [129].

#### 4.1.2 Otimização Taxa-Distorção dos VMs

Conforme já mencionado, na codificação de vídeo a RDO tem mostrado o melhor desempenho dentre os vários processos de otimização [115] e [131]. Essa otimização é geralmente realizada através da minimização da função custo RD dada pela seguinte função de custo lagrangeano:

$$J = D + \lambda R \quad (25)$$

onde  $D$  representa a distorção da imagem,  $R$  é a taxa de bits e  $\lambda$  é multiplicador de Lagrange. O fator  $J_r = \lambda R$  pode ser considerado o custo RD correspondente a taxa de bits. O multiplicador  $\lambda$  permite estabelecer um compromisso entre taxa de bits e distorção.

Quando a função taxa-distorção (curva RD) é convexa, o resultado do processo de otimização é uma solução ótima independentemente do valor escolhido para  $\lambda$  [139], o qual pode ser variado entre 0 e  $\infty$  para determinar a taxa de bits ou qualidade desejada. A escolha do valor de  $\lambda$  pode ser realizada através métodos iterativos como da biseccção [140], curvas de Bezier [95], ou mesmo ser baseado em parâmetros experimentais conforme discutido em [126] e [141]. Neste último caso, funções que relacionam o multiplicador de



Lagrange  $\lambda$  com o parâmetro de quantização  $Q$  podem ser usadas [96], [115], [138].

Nos codificadores que seguem o padrão H.263, usando como medida de distorção a SAD, o valor de  $\lambda$  pode ser determinado a partir de [96]

$$\lambda_1 = 0,92Q \quad (26)$$

enquanto nos codificadores que seguem o padrão H.264 [142] a função usada é

$$\lambda_1 = 0,92 \times 2^{(Q-12)/6}. \quad (27)$$

Por outro lado, se a distorção é medida em SSD, deve-se usar um multiplicador de Lagrange [115], [142]

$$\lambda_2 = \lambda_1^2. \quad (28)$$

#### 4.1.3 Interpretação Gráfica da Otimização RD dos VMs

O processo de otimização em RD da EM usando o multiplicador de Lagrange pode ser interpretado de forma gráfica [143]. Nessa interpretação as taxas de bits  $R_i$  e as distorções  $D_i$  obtidas para cada VMs são posicionadas em um gráfico. Traçando uma linha reta como uma função da taxa de bits para cada ponto  $(R_i, D_i)$  neste gráfico obtém-se várias retas paralelas

$$D_i = J_i - \lambda R_i. \quad (29)$$

O processo de otimização em RD consiste em encontrar a reta que tem o menor coeficiente linear  $J_i$  para o coeficiente angular  $-\lambda$ . Assim o custo RD ótimo é dado por

$$J_{opt} = \arg \min_{\forall_i} (J_i | \lambda). \quad (30)$$

Para ilustrar tal procedimento, é utilizado o exemplo de um codificador H.264 no qual o processo de EM é dividido em etapas com diferentes resoluções. Na primeira, os valores de  $R_i$  e  $D_i$  são obtidos através de um algoritmo de busca exaustiva (*full search* – FS) que avalia todos os VMs com resolução de 1 *pixel* em uma janela de procura de

$\pm w$  pixels. Em seguida, o VM que minimiza (30) é selecionado e os refinamentos de *subpixel* são efetuados. Na Figura 21, o processo RDO é ilustrado para o macrobloco número 0 (MB #0) do quadro #15 da sequência *Mother&Daughter*. Nesse caso, o VM que minimiza o custo RD para  $\lambda = 54,4$  é o VM nulo ( $VM_{null}$ ).

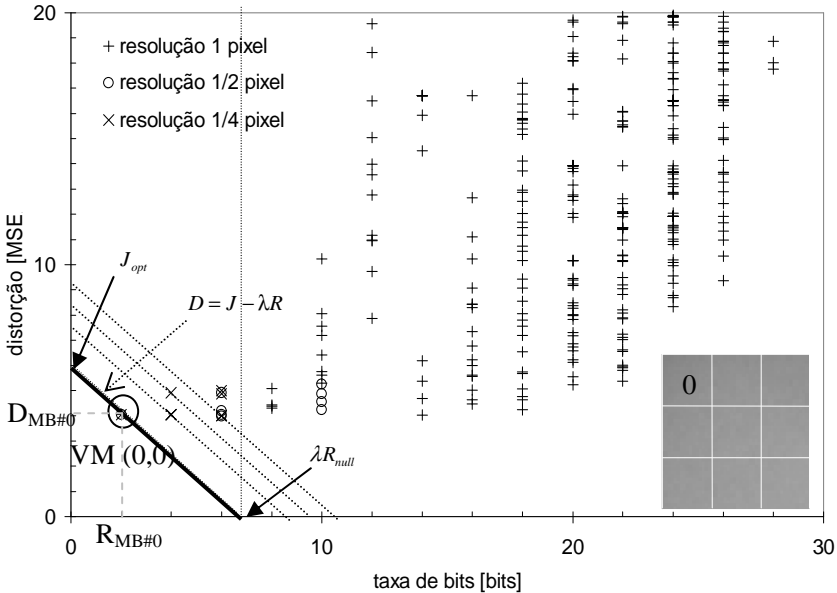


Figura 21 - Interpretação gráfica do processo de otimização RD, habilitando apenas o modo *Inter*<sub>16x16</sub> do codificador JM7.3. Os pontos mostrados correspondem aos VMs com resolução de 1 *pixel* (+), 1/2 *pixel* (o) e 1/4 de *pixel* (x).

Pela análise dos pontos de mesmo custo RD que o  $VM_{null}$  (linha reta destacada com linha grossa na Figura 21) se verifica que a maior parte das avaliações de taxa e distorção geralmente realizadas é completamente desnecessária. No exemplo mostrado, o custo RD obtido com o  $VM_{null}$  só pode ser reduzido usando VMs que sejam codificados com custo RD menor do que  $\lambda R_{null}$ . Assim, todos VMs com custo de codificação maior do que  $R_{null}$  possuem também custo RD superior ao  $VM_{null}$  e, por isso, podem ser descartados sem a necessidade de avaliar a distorção resultante. No Capítulo 5, essa característica será usada para

reduzir a complexidade do codificador mantendo o processo otimizado em RD. Baseado nessa interpretação gráfica também se pode analisar a influência do valor do multiplicador de Lagrange  $\lambda$  na escolha dos VMs ótimos. A Figura 22 mostra os valores de  $R_i$  e  $D_i$  obtidos para o MB #19 do quadro #15 da sequência *Mother&Daughter*. As linhas de mesmo custo RD que passam pelos VMs ótimos mostradas para valores de  $\lambda = 54,4$ ;  $548,3$  e  $5527$ . A dependência do custo RD com  $\lambda$  faz com que VMs de menor custo tenham maior preferência à medida que uma menor taxa de bits é desejada (maior valor de  $\lambda$ ).

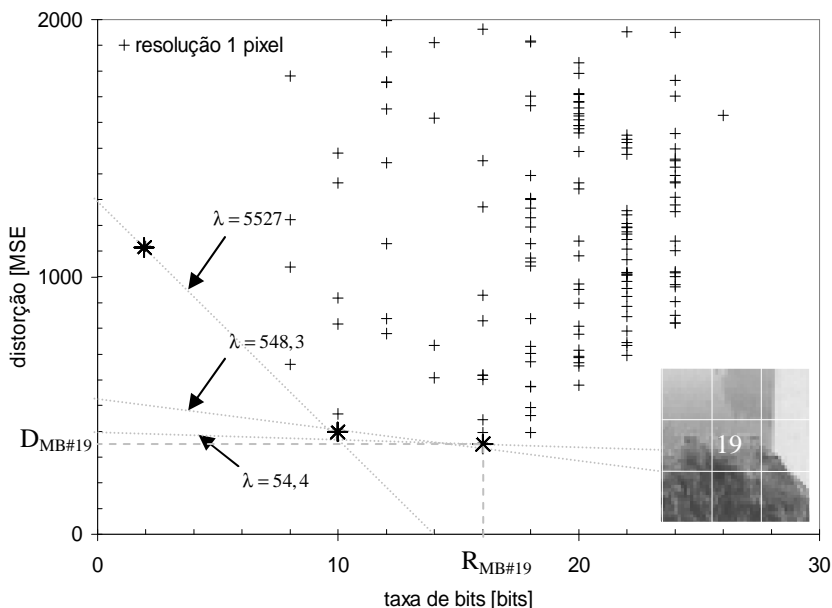


Figura 22 - Interpretação gráfica do processo de otimização RD da EM com resolução de 1 *pixel* para diferentes valores de  $\lambda$ , habilitando apenas o modo *Inter*<sub>16x16</sub> do codificador JM7.3.

Outro aspecto notado é a grande diferença entre os valores finais de distorção obtidos para o MB #0 ( $D_{MB\#0}$  na Figura 21) e o MB #19 ( $D_{MB\#19}$  na Figura 22), ambos com  $\lambda = 54,4$ . A distorção no MB #0 corresponde a apenas a 1% da distorção no MB #19, indicando que a determinação do VM ótimo no primeiro caso tem menor importância do que no segundo. Essa diferença nos valores de distorção sugere que o processo de EM de um MB pode ser interrompido quando é atingido um

limiar de qualidade considerada “suficiente”. Dessa forma se pode evitar a determinação da distorção de novos VMs quando não se tem mais muito a ganhar. No Capítulo 5, será mostrado como o emprego desse limiar de qualidade pode reduzir a complexidade computacional sem perda significativa de desempenho RD.

## 4.2 Avaliação de Desempenho Quadro-a-Quadro

Conforme foi ilustrado na Figura 19, a avaliação do desempenho RD quadro-a-quadro não pode considerar em separado a distorção e taxa de bits. Para superar tal problema frequentemente é fixada uma taxa de bits por quadro e a distorção é medida. No entanto, essa abordagem leva a uma redução da qualidade em quadros que exigem uma maior taxa e a um desperdício de bits em quadros que podem ser codificados com taxa de bits menor.

Neste trabalho uma solução alternativa é apresentada para evitar a fixação da taxa de bits (ou da qualidade). A proposta é avaliar o desempenho a cada quadro através de uma medida de custo RD dada por:

$$J = D_d + \lambda^d R \quad (31)$$

onde  $D_d$  é a distorção medida em MAD ( $d = 1$ ) ou MSD ( $d = 2$ ) e  $R$  é a quantidade de bits usada na codificação do quadro. Dependendo da medida de distorção considerada em  $D_d$ , a medida  $J$  neste caso pode ser denominada de MAD lagrangeana (LMAD) ou a MSD lagrangeana (LMSD).

Quando a medida  $J$  é usada no processo de EM de um bloco, considera-se  $R$  como a taxa de bits necessária para codificação da informação de movimento e  $D_d$  a distorção entre o bloco original e aquele obtido com a CM. Por outro lado, se a avaliação é de todo o processo de codificação do quadro, então  $R$  será a taxa de bits correspondente à codificação completa do quadro e  $D_d$  os valores de distorção do quadro codificado em relação ao original.

Frequentemente o PSNR é usado na medição da qualidade de imagens no lugar das medidas de distorção MSD e MAD. Neste caso, o PSNR lagrangeano é definido por:

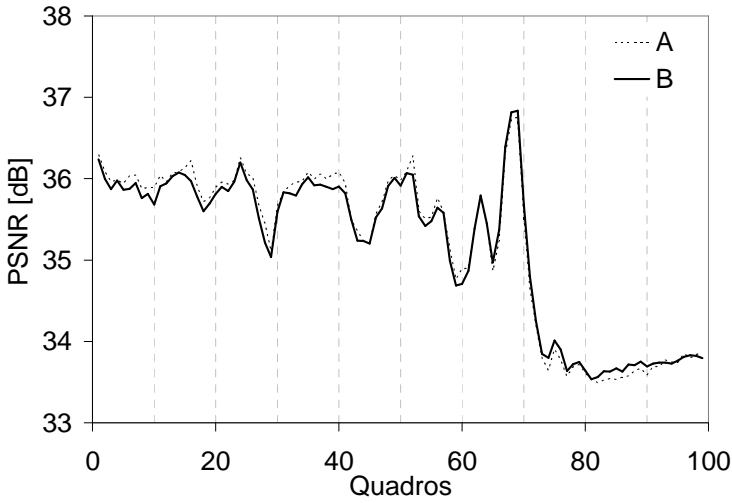
$$\text{LPSNR} = 10 \log_{10} \left( \frac{2^{2N}}{J} \right). \quad (32)$$

A vantagem do uso da medida LPSNR no lugar do PSNR é considerar a taxa de bits junto com a distorção. Dessa forma é possível comparar diretamente quadro-a-quadro o desempenho RD de duas estratégias sem que seja necessário fixar nem a taxa de bits nem a qualidade.

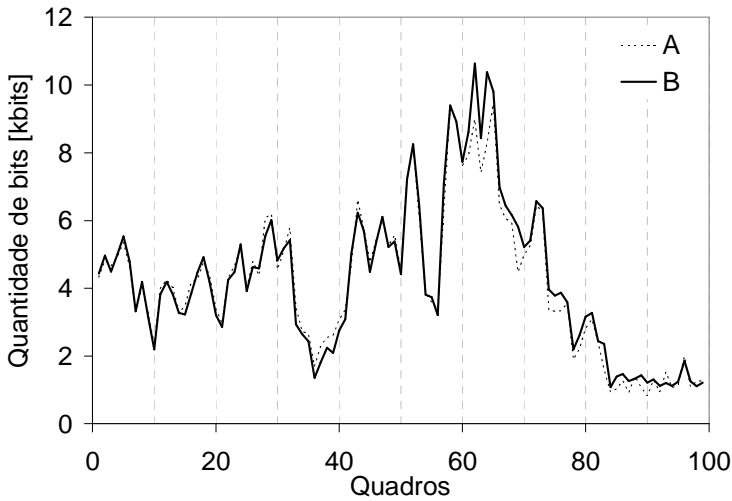
Para ilustrar as vantagens da LPSNR sobre as outras medidas de distorção, usamos como exemplo os resultados de variação de PSNR e taxa de bits (ver Figura 23) obtidos com dois codificadores de vídeo hipotéticos CoderA e CoderB. Nesses gráficos percebem-se apenas pequenas diferenças entre os dois codificadores tanto no PSNR quanto na taxa de bits. Analisando apenas a variação do PSNR [Figura 23 (a)] se nota que o CoderA resulta em uma melhor qualidade no trecho inicial da sequência de vídeo (até o quadro #62), enquanto o CoderB apresenta uma melhor qualidade no trecho entre os quadros #63 e #92.

Na Figura 24 a diferença entre os valores medidos de PSNR e LPSNR dos mesmos codificadores de vídeo CoderA e CoderB é mostrada. Note que usando a medida LPSNR se pode constatar o ganho (valores positivos) ou perda (valores negativos) de desempenho RD a cada quadro. Por exemplo, verifica-se que apesar do CoderB ter uma qualidade de imagem superior em PSNR ao CoderA entre os quadros #63 e #92, mas o seu desempenho RD é pior neste trecho. Esse pior desempenho RD é consequência da maior taxa de bits usada pelo CoderB na codificação desses quadros [ver Figura 23(b)].

O exemplo mostrado na Figura 19 também permite ilustrar as vantagens da medida LPSNR em relação ao PSNR. Realizando a mesma comparação, ou seja, calculando a diferença entre os valores medidos de PSNR e LPSNR da abordagem RDO e DO, se constata que ocorre uma perda de qualidade em PSNR com o emprego da abordagem RDO (ver linha pontilhada da Figura 25). No entanto, nota-se que o desempenho RD da abordagem RDO (linha contínua na Figura 25) é sempre superior a abordagem DO.



(a)



(b)

Figura 23 - Comparação quadro-a-quadro dos codificadores de vídeo hipotético CoderA e CoderB. (a) PSNR. (b) Taxa de bits.

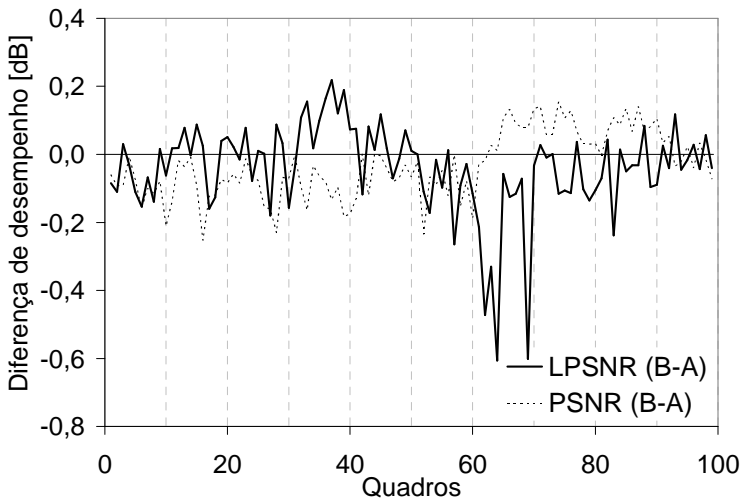


Figura 24 - Comparação quadro-a-quadro da diferença de desempenho do codificador hipotético CoderB com relação ao CoderA. (Linha contínua) Medida de distorção LPSNR. (Linha pontilhada) Medida da distorção em PSNR.

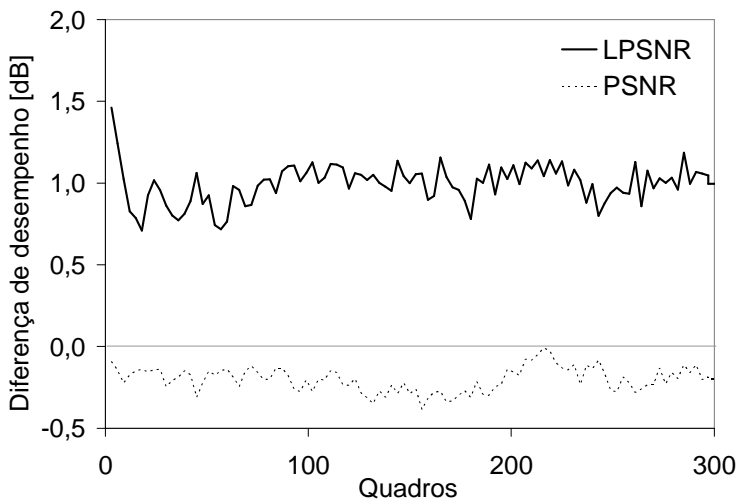


Figura 25 - Comparação da diferença de desempenho quadro-a-quadro das estratégias DO e RDO usando as medidas PSNR e LPSNR.

### 4.3 Seleção Otimizada do Modo de Codificação

Na seção anterior, foi tratado o problema da otimização dos VMs. Para que todo o processo de codificação do vídeo seja otimizado em RD é necessário que a seleção do modo de codificação para cada bloco de imagem também seja otimizada em RD. Nos codificadores de vídeo que dispõem de diversos modos para a codificação de cada bloco de imagem, eles são selecionados de acordo com o conteúdo visual da imagem. Para que a escolha seja otimizada em RD, é necessário levar em conta a dependência de cada bloco com os blocos já codificados e a sua interação com blocos ainda não codificados. A escolha otimizada do modo de codificação resulta em uma elevadíssima carga computacional [129], [131], [144], e para reduzir essa complexidade geralmente é desconsiderada a influência do quadro (e bloco) atual sobre os futuros [141]. O emprego dessa abordagem causal simplifica o problema, mas impede a exploração de todas as dependências entre quadros (e blocos), resultando em um desempenho RD inferior ao ótimo.

A otimização RD da seleção dos modos de codificação pode ser realizada determinando para cada modo  $m$  o custo RD

$$J(m) = D_{\text{REC}}(m) + \lambda R_{\text{REC}}(m) \quad (33)$$

onde  $R_{\text{REC}}(m)$  é a taxa de bits necessária para a codificação do bloco (incluindo a indicação do tipo de modo, informações de movimento e codificação da DFD);  $D_{\text{REC}}(m)$  é a distorção entre o bloco original e do bloco codificado.

A seleção do modo ótimo  $m_{\text{opt}}$  de cada bloco de imagem é feita minimizando-se o custo RD

$$m_{\text{opt}} = \arg \min_{\forall m \in \mathbf{M}} (J(m)) \quad (34)$$

onde  $\mathbf{M} = \{m_1, m_2, \dots\}$  é o conjunto de modos que o codificador aceita.

#### 4.3.1 Conjunto de Modos de Codificação

O uso de diversos modos em um codificador de vídeo visa adaptar a codificação ao conteúdo da imagem de cada bloco. Por exemplo, em áreas quase estacionárias ou estacionárias de uma imagem, se pode simplesmente copiar blocos do quadro anterior (usando o modo *Skip*). Para se obter uma predição adequada em blocos que sofreram movimento de translação (usando o modo *Inter*), copiam-se blocos de



quadros anteriores (e ou posteriores) deslocados por um VM; se não houver nenhuma informação nesses quadros que permita obter uma satisfatória predição do bloco atual, utilizam-se a informação do quadro atual para fazer tal predição espacial através do uso do modo *Intra*.

No caso dos padrões H.263 [16] e H.264 [17], o conjunto de modos disponíveis para a codificação dos macrobloco (MB) em quadros preditos é

$$\mathbf{M}_{H.263} = \{Skip, Intra, Inter_{16 \times 16}, Inter_{8 \times 8}\} \quad (35)$$

e

$$\mathbf{M}_{H.264} = \{Skip, Inter_{16 \times 16}, Inter_{16 \times 8}, Inter_{8 \times 16}, Inter_{8 \times 8}, Intra_{16 \times 16}, Intra_{4 \times 4}\}. \quad (36)$$

Os modos do tipo *Inter* utilizam como predição uma cópia deslocada de *pixels* de um quadro de referência anterior, enquanto os modos *Intra* utilizam uma predição espacial direcional obtida a partir dos *pixels* vizinhos ao bloco no quadro atual. Em ambos os modos, o erro de predição é codificado para reduzir a distorção. No modo *Skip* nenhuma informação de predição é codificada, havendo uma pequena diferença entre os padrões H.263 e H.264 para esse modo. No H.263, é feita a cópia direta dos *pixels* do quadro anterior [16], enquanto no H.264 a cópia dos *pixels* é deslocada de acordo com VM predito [17].

O padrão H.264, para permitir uma melhor adaptação do modo de codificação ao conteúdo de cada bloco, tem uma maior variedade de modos de codificação do que o H.263. O conjunto de modos do H.264 (36) define apenas os modos básicos, pois para cada modo  $Inter_{8 \times 8}$ ,  $Intra_{16 \times 16}$  e  $Intra_{8 \times 8}$  existem muitas variedades. O modo  $Inter_{8 \times 8}$  pode ser dividido em quatro tipos de submacroblocos  $Inter_{8 \times 8}$ ,  $Inter_{8 \times 4}$ ,  $Inter_{4 \times 8}$  e  $Inter_{4 \times 4}$ , resultando em um total de 260 possibilidades de codificação em modo *Inter* de cada MB. No modo  $Intra_{16 \times 16}$ , o componente de luminância pode ser codificado com 4 variantes espaciais (DC, vertical, horizontal e plana) enquanto no  $Intra_{4 \times 4}$  existem 9 variantes espaciais (ver Figura 2) [17]. Cada uma delas pode ter a predição de cor realizada de 4 maneiras distintas resultando, portanto, em um total de 52 variantes de codificação no modo *Intra*.

Para selecionar os modos de codificação de cada MB, mantendo a codificação otimizada em RD, é necessário determinar a taxa de bits  $R_{REC}$  e a distorção  $D_{REC}$  de cada variante dos modos *Inter* e *Intra*. A

escolha da melhor combinação de modos é obtida calculando o custo RD (33) e minimizando (34) para o MB. Esse processo de otimização em RD é de grande complexidade computacional quando uma estratégia de busca exaustiva do melhor modo é usada.

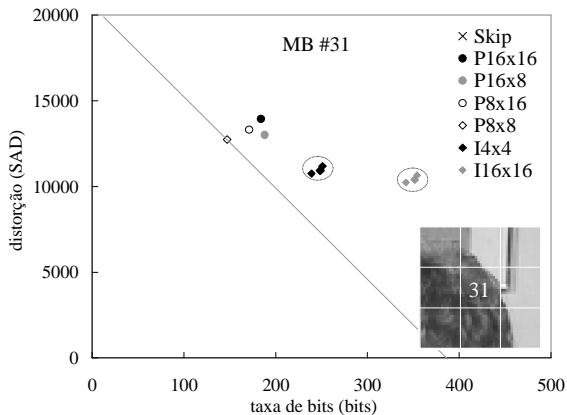
### 4.3.2 Interpretação Gráfica do Processo de Otimização do Modo de Codificação

A otimização em RD da seleção do modo de codificação do MB pode ser interpretada de forma gráfica de maneira similar a usada para interpretar a otimização dos VMs. Para tal, as taxas de bits  $R_{\text{REC}}(m_i)$  e as distorções  $D_{\text{REC}}(m_i)$  obtidas são representadas no gráfico RD. Traçando as retas

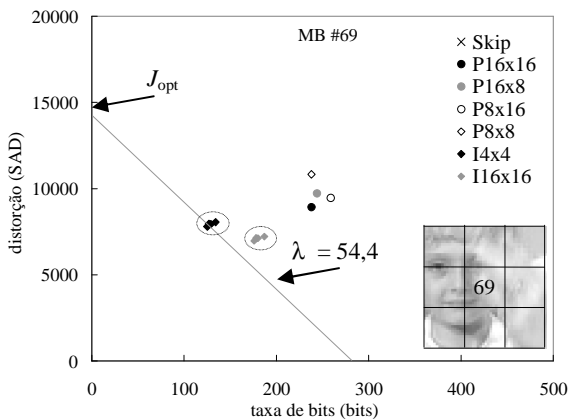
$$D_{\text{REC}}(m_i) = J(m_i) - \lambda R_{\text{REC}}(m_i) \quad (37)$$

obtem-se para cada multiplicador  $\lambda$  linhas paralelas que passam pelos pontos  $[R_{\text{REC}}(m_i), D_{\text{REC}}(m_i)]$ , onde  $m_i \in \mathbf{M}$  é um modo de codificação admitido pelo padrão. O processo de otimização em RD consiste em encontrar a reta com menor coeficiente linear  $J(m_i)$ , ou seja, minimizar (34).

A Figura 26 ilustra essa interpretação gráfica para alguns MBs do quadro #15 da sequência *Mother&Daughter*. Para simplificar a visualização, apenas é mostrado o modo *Skip*, os melhores resultados obtidos para os modos *Inter*<sub>16×16</sub>, *Inter*<sub>16×8</sub>, *Inter*<sub>8×16</sub> e *Inter*<sub>8×8</sub>, e a melhor predição do componente de luminância usando os modos *Intra*<sub>4×4</sub> e *Intra*<sub>16×16</sub> para cada uma das 4 formas de predição espacial da cor. No exemplo, a minimização do custo RD resulta na seleção dos modos *Inter*<sub>8×8</sub>, *Intra*<sub>4×4</sub>, *Inter*<sub>16×16</sub> e *Skip* para os MBs #31, #69, #78 e #98, respectivamente. Note que devido à otimização em desempenho RD, o modo ótimo escolhido na maioria das vezes não corresponde à solução com menor distorção mostrando que o uso da otimização em distorção conduz a perda de desempenho RD.

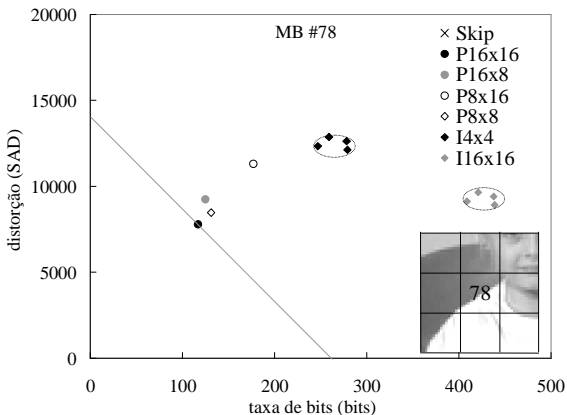


(a)

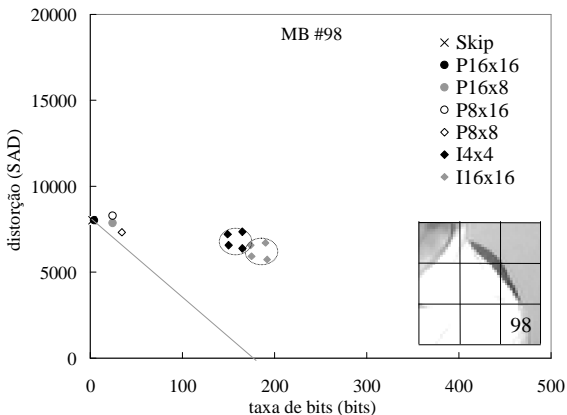


(b)

Figura 26 - Representação gráfica da otimização RD para os modos de codificação do H.264 usando o codificador JM7.3-RDO com  $Q = 30$ . Quadro #15 da sequência *Mother&Daughter*. (a) MB #31 – modo ótimo  $Inter_{8 \times 8}$ . (b) MB #69 – modo ótimo  $Intra_{4 \times 4}$ . (c) MB #78 – modo ótimo  $Inter_{16 \times 16}$ . (d) MB #98 – modo ótimo  $Skip$ .



(c)



(d)

Figura 26 (cont) - Representação gráfica da otimização RD para os modos de codificação do H.264 usando o codificador JM7.3-RDO com  $Q = 30$ . Quadro #15 da sequência *Mother&Daughter*. (a) MB #31 – modo ótimo  $Inter_{8 \times 8}$ . (b) MB #69 – modo ótimo  $Intra_{4 \times 4}$ . (c) MB #78 – modo ótimo  $Inter_{16 \times 16}$ . (d) MB #98 – modo ótimo  $Skip$ .

Outro aspecto a ser destacado, neste exemplo ilustrativo, são as mínimas variações de custo RD obtidos para as diferentes maneiras de predição de cor usadas nos modos  $Intra_{4 \times 4}$  e  $Intra_{16 \times 16}$ . As variações de taxa de bits e distorção em função da forma de predição da cor são muito similares, conforme mostram os agrupamentos de pontos

representados por losangos pretos ( $Intra_{4 \times 4}$ ) e cinza ( $Intra_{16 \times 16}$ ) destacadas na Figura 26. Essas características são verificadas em todas as sequências de vídeos analisadas e pode ser atribuída a mínima influência que os componentes de crominância exercem sobre distorção e taxa total de bits de um MB, principalmente, quando a resolução espacial da crominância é menor do que a da luminância. Essa semelhança de desempenho RD das diferentes predições de cor é abordada no Capítulo 5, no qual é proposto um algoritmo que pré-elimina alguns modos *Intra*, reduzindo a complexidade computacional com mínima perda no desempenho RD.

#### 4.4 Conclusões

Neste capítulo, foram avaliados os processos otimizados em RD para a EM e seleção do modo de codificação. Foi também enfatizado que a otimização em RD apresenta um desempenho muito superior ao da otimização em distorção. Também foi proposta uma nova forma de avaliar o desempenho quadro-a-quadro da codificação de vídeo, permitindo a comparação de desempenho RD.

A interpretação gráfica da otimização RD do processo EM e da seleção do modo de codificação para os MB permite destacar algumas características que podem ser usadas na redução de complexidade computacional dos codificadores de vídeo otimizados em RD, sem que haja perda de desempenho RD ou, pelo menos, perda insignificante.

Destacamos os diferentes graus de importância dos diversos blocos de uma imagem na qualidade final de um quadro. Tal diferença é evidenciada pela ampla variação dos valores de distorção entre os blocos que compõem um quadro. Em decorrência dessa característica, é proposto no próximo capítulo um algoritmo no qual a distorção do bloco seja usada como critério para determinar a necessidade de avaliação ou não de novos VMs e modos de codificação.

Foi também evidenciado que no processo de EM muitos VMs não precisam ser avaliados, por possuírem um custo RD mínimo superior ao do melhor VM já obtido. Consequentemente tais VMs podem ser descartados durante o processo de EM sem que seja necessário determinar a correspondente distorção. Essa característica, para ser adequadamente explorada, requer uma estratégia de EM que avalie os VMs na ordem determinada pelo seu custo de codificação, conforme será mostrado no próximo capítulo.

Outro aspecto mostrado é a influência mínima que as diferentes maneiras de predição espacial da cor têm sobre o custo RD nos MBs codificados em modo *Intra*. Essa característica pode ser explorada descartando antecipadamente alguns desses modos de predição, devido à distorção mínima já obtida para o MB com outros modos *Inter* e *Intra* durante o processo de otimização.

Para a avaliação de desempenho RD quadro-a-quadro, foi proposto o uso da distorção lagrangeana no lugar das medidas distorção MAD, MSD e PSNR geralmente usadas. A medida lagrangeana (LMSD, LMAD ou LPSNR) que considera conjuntamente a taxa de bits e a distorção foi apresentada, permitindo comparar diretamente o desempenho RD a cada quadro sem a necessidade de fixar a taxa de bits.

# Capítulo 5

## Codificação por Custo Crescente

O desempenho de codificadores de vídeo que empregam o processo de EM e a seleção do modo de codificação otimizados em RD [102], [115], [141] é superior àqueles que otimizam apenas a distorção [145]. A implementação dos processos otimizados em RD é relativamente simples quando poucos modos de codificação precisam ser avaliados e a janela de procura do VM é pequena; no entanto, quando os codificadores usam um grande número de modos de codificação, tais como aqueles baseados no padrão H.264, o processo de otimização se torna bastante complexo. Muitas abordagens vêm sendo propostas na literatura para reduzir tal complexidade. Algumas delas são discutidas (de forma resumida) na introdução deste capítulo. Além disso, uma nova proposta explorando a dependência do custo RD em relação à taxa de bits, visando simplificar tanto o processo de EM como a seleção do modo de codificação é também apresentada.

### 5.1 Introdução

A complexidade da otimização RD é evitada no modelo de teste TMN8 [101], através do favorecimento do VM nulo sempre que sua distorção é próxima a do VM ótimo. De maneira similar em [146] e [147] o processo de EM é terminado antecipadamente se o modo *Skip* tem custo menor do que o de um limiar predefinido. Em [148], o *Joint Video Team (JVT)* propõe o uso da predição *Inter* de blocos  $16 \times 16$  para pré-selecionar o modo *Skip*. Em [129], uma janela de procura menor é usada, limitando a EM otimizada em RD no entorno da posição do VM nulo, do VM predito e dos VMs dos blocos vizinhos. Em [127], o processo de EM é otimizado apenas em distorção para uma resolução de 1 *pixel*, seguido de uma EM otimizada em RD com resolução de *subpixel* em torno do melhor VM.

Todas as abordagens anteriores visam reduzir a complexidade do processo de otimização RD da EM, sem aliviar a complexidade computacional da seleção do modo de codificação. Outros algoritmos, denominados de algoritmos rápidos na literatura, usam um esquema de grade de pontos iniciais para a procura do melhor VM. No entanto, esses

algoritmos podem ser armadilhados por mínimos locais de distorção e por isso não são considerados como otimizados em RD [149].

Há abordagens que limitam o número de modos avaliados, como em [150], no qual são avaliados apenas os modos *Inter* dos MBs usados nos MBs vizinhos espaciais do quadro atual ou vizinhos temporais no quadro anterior, conduzindo a uma redução de carga computacional.

Outros trabalhos propõem redução de complexidade na predição do modo *Intra*. As propostas discutidas em [151] e [152] reduzem o número predições *Intra*, usando a direção das bordas, enquanto a proposta dada em [153] usa a informação direcional da textura para evitar o processo de predição. Em [154], o melhor modo *Inter* é usado para descartar alguns dos modos *Intra*, considerando para tal que a direção das bordas locais e os movimentos dos objetos afetam a escolha do modo *Intra*. Em [155], a predição do modo *Intra* para blocos de  $16 \times 16$  *pixels* é usada para reduzir o número de predições candidatas a ser avaliadas nos blocos de  $4 \times 4$  *pixels*. Em [156], uma técnica de estimação da taxa de bits no domínio transformado é usada para substituir o custo lagrangeano. Em [157] e [158], o uso do operador de Sobel para detectar regiões homogêneas assim como a direção local das bordas é proposto pelo JVT para selecionar o conjunto mais provável de modos *Intra* e *Inter*. Em [159], um método de poda dos modos de codificação, baseado nas dependências estatísticas entre os MBs, é proposto com o objetivo de descartar antecipadamente alguns dos modos *Intra* e *Inter*.

Além dessas estratégias mencionadas anteriormente, muitas outras vêm sendo discutidas para reduzir a complexidade computacional do processo de codificação RDO do padrão H.264, buscando o melhor desempenho RD sem avaliar a taxa de bits e distorção para cada um dos modos permitidos. Por exemplo, na estratégia proposta em [160], dependendo da monotonicidade da superfície de custo RD obtida com os modos *Inter*  $16 \times 16$ ,  $8 \times 8$  e  $4 \times 4$ , os demais modos *Inter* são descartados. Nessa mesma proposta a energia do erro é usada como critério para descartar os modos *Intra*.

Por outro lado, em nosso trabalho [161], são explorados os problemas de complexidade do processo de EM e de seleção dos modos de codificação do tipo *Inter* e *Intra*, visando aplicações em baixa taxa de bits, para as quais geralmente a capacidade computacional disponível é limitada. A abordagem proposta neste trabalho, denominada ordenamento e truncamento por taxa de bits (*rate sorting and truncation* – RST), é baseada no ordenamento dos VMs e modos de codificação em



ordem ascendente de taxa bits. Com esse processo de ordenamento, a codificação pode ser truncada quando a taxa de novos VMs ou modos de codificação exceder a menor taxa já obtida para um determinado nível de qualidade de imagem. Assim, muitos VMs e modos de codificação são descartados antes de ser avaliada sua distorção. Apesar de se obter uma considerável redução na complexidade do processo, este ainda permanece otimizado em RD, pois apenas VMs e modos de codificação que não podem melhorar o desempenho RD são descartados. Além disso, uma redução adicional de complexidade pode ser obtida se alguma perda de desempenho RD é tolerada, parando o processo de codificação quando um limiar de qualidade for atingido [161].

Para comparar o desempenho RD e a complexidade computacional da estratégia RST com uma referência padrão, ela foi integrada ao codificador de referência JM7.3 do *Heinrich Hertz Institute* [91], através do acréscimo das modificações necessárias. Na validação da estratégia RST, são consideradas sequências de vídeo típicas (ver Apêndice 1), incluindo sequências com câmera em movimento, câmera fixa, sequências com pouco movimento e outras com alta quantidade de movimento. A avaliação de complexidade computacional do algoritmo de EM é realizada usando o número de pontos de procura (NPP) [129], enquanto o número de modos de predição espacial é usado para avaliar a complexidade da seleção do modo *Intra*.

## 5.2 Efeito da Predição dos VMs na Taxa de Bits

Em codificação de vídeo de baixas taxas de bits os vetores de movimento (VMs) e o tamanho do bloco usado são os principais fatores que repercutem na taxa de bits ao final do processo de EM. O tamanho do bloco estabelece o número de vetores que será usado para compensar o movimento entre quadros de um vídeo. Os VMs  $\vec{v} = (x, y)$  são compostos dos componentes  $x$  e  $y$  que especificam o deslocamento dos *pixels* do bloco do quadro atual em relação ao quadro de referência. Geralmente, a EM é iniciada pelos blocos do canto superior esquerdo da imagem e os VM preditos  $\vec{v}_p = (x_p, y_p)$  de cada bloco são obtidos determinando o valor mediano dos componentes  $x$  e  $y$  dos blocos vizinhos causais  $b_A$ ,  $b_B$  e  $b_C$  conforme mostrado na Figura 27. Quando um desses blocos está situado fora da borda da imagem ou é codificado

no modo *Intra*, então o VM do bloco  $b_D$  ou o VM nulo  $\vec{v} = (0,0)$  é usado no lugar do VM faltante.

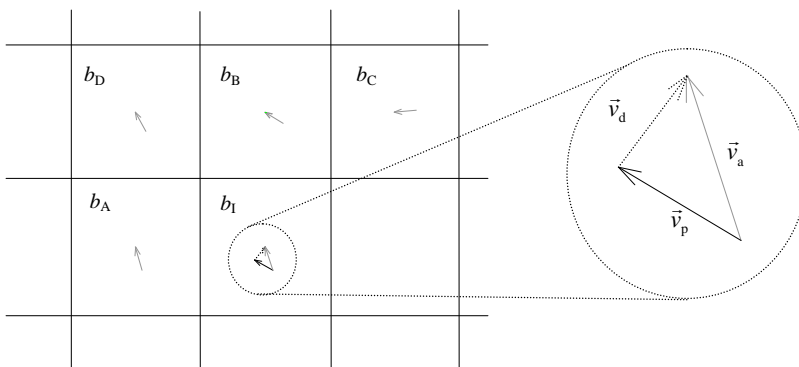


Figura 27 - Predição do VM a partir dos blocos vizinhos causais ( $b_A$ ,  $b_B$ ,  $b_C$ ,  $b_D$ )<sup>II</sup>.

Nos padrões de vídeo, o processo de EM/CM é geralmente iniciado pelo canto superior esquerdo e prossegue percorrendo as linhas de blocos. Após determinado o vetor predito  $\vec{v}_p = (x_p, y_p)$ , a diferença desse vetor em relação ao VM atual  $\vec{v}_a = (x_a, y_a)$  é obtido por

$$\vec{v}_d = (x_d, y_d) = (x_a - x_p, y_a - y_p). \quad (38)$$

O uso da predição do VM é crucial na minimização da taxa de bits, mas cria uma dependência entre os blocos, dificultando o processo de otimização em RD. Tal problema é considerado na implementação discutida em [141]; no entanto, para simplificar o processo de otimização, a influência dos VMs sobre os blocos ainda não codificados é muitas vezes desconsiderada levando a um desempenho subótimo. Um exemplo do uso de abordagem subótima são os modelos de teste JM do JVT [91]. Note que, se for considerada a influência de cada VM sobre os próximos blocos no quadro atual e nos quadros posteriores, seria certamente possível melhorar o desempenho RD do codificador; no entanto, isso tornaria a complexidade computacional do processo de EM demasiadamente elevada.

<sup>II</sup> Os VMs são representados por setas proporcionais ao seu tamanho em *pixels*.

Nos padrões H.263 e H.264, a diferença de vetores de movimento (DVMs) obtida em (38) é codificada usando tabelas VLCs ou codificação aritmética. No padrão H.264 as VLCs usadas são do tipo *Exp-Golomb* [17], conforme mostrado na Tabela 9. Nesse caso a taxa de bits dos DVMs  $R(\vec{v}_d)$  depende diretamente dos componentes  $x_d$  e  $y_d$  dos DVMs. Assim,

$$R(\vec{v}_d) = R(x_d) + R(y_d) \quad (39)$$

onde  $R(x_d)$  e  $R(y_d)$  representam a taxa dos componentes  $x$  e  $y$  do vetor predito, respectivamente.

Tabela 9 – Padrão de codificação da diferença dos VMs usando VLC do tipo *Exp-Golomb*

Componente do VM em <i>pixel</i>	Prefixo	Sufixo <sup>III</sup>	Número de bits
0	1		1
±1/4	01	$s$	3
±1/2 e ±3/4	001	$b_1s$	5
±1 até ±1 3/4	0001	$b_2b_1s$	7
±2 até ±3 3/4	00001	$b_3b_2b_1s$	9
±4 até ±7 3/4	000001	$b_4b_3b_2b_1s$	11
...	...	...	...

### 5.3 Distribuição Bidimensional dos VMs

No processo de EM a distribuição dos VMs nas dimensões  $x$  e  $y$  irá depender do movimento existente entre os quadros sucessivos da sequência de vídeo. Em sequências com movimentos amplos ocorrem com frequência VMs de maior tamanho; enquanto naquelas com movimentos reduzidos a maioria dos vetores é nula. No entanto, no padrão H.264, como os VMs são codificados através dos DVMs, ocorre uma tendência de concentração dos valores próximos ao DVM nulo, independente do movimento, pois a predição dos VMs tende a minimizar a diferença de vetores de movimento [162].

Uma estimativa da distribuição estatística dos DVMs em sequências de videoconferência em baixas taxas de bits pode ser obtida

<sup>III</sup> O bit  $s$  representa o sinal ('0' para positivo e '1' para negativo). Os bits  $b_N \dots b_2 b_1$  assumem os valores de 0...00 até 1...11 de acordo com o tamanho do componente.

considerando os processos de EM otimizados ou em distorção ou em taxa-distorção. A distribuição bidimensional de ocorrência (histograma bidimensional) dos DVMs é obtida para valores médios dos 100 primeiros quadros de seis sequências típicas (*Akiyo*, *News*, *Silent*, *Foreman*, *Mother&Daughter* e *Carphone*) em formato QCIF (ver Figura 29). Os resultados são obtidos usando o modelo de teste JM7.3, primeiramente, sem habilitar a otimização RD (DO) e, em seguida, habilitando a otimização RD (RDO). Usando otimização em distorção [ver Figura 29 (a) e (b)] é obtida uma distribuição Laplaciana aproximadamente circular simétrica [162]; enquanto usando a RDO [ver Figura 29 (d) e (e)] ocorre uma concentração de DVMs nos eixos principais  $x$  e  $y$ . Note que quanto mais grosseira é a quantização, maior é a concentração de DVMs nos eixos.

A Figura 28 ilustra o recíproco da taxa  $R(v)^{-1}$  dos DVMs obtido usando a VLC do padrão H.264. Comparando a distribuição RDO (ver Figura 29) com o recíproco da taxa dos DVMs, é observado que há uma forte dependência entre a ocorrência dos DVMs e o recíproco da taxa. Essa dependência indica que é preferível avaliar primeiro os vetores com menor taxa, uma vez que a probabilidade de escolha é maior para esses vetores. A abordagem proposta neste trabalho, denominada RST com regras de truncamento por taxa e qualidade, é baseada neste argumento, podendo o processo de EM para os VMs de maior taxa quando nenhum outro VM puder melhorar o desempenho RD já obtido.

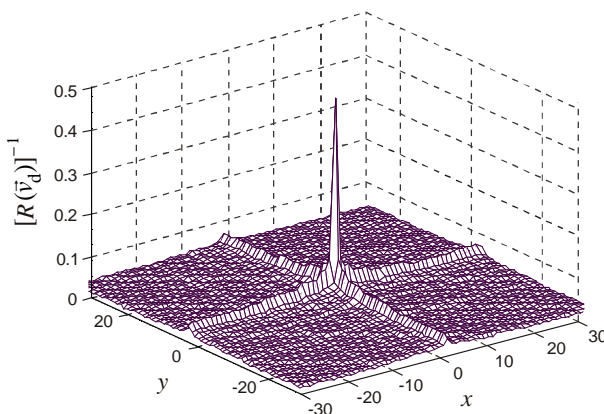


Figura 28 - Recíproco da taxa dos DVMs para codificação VLC.

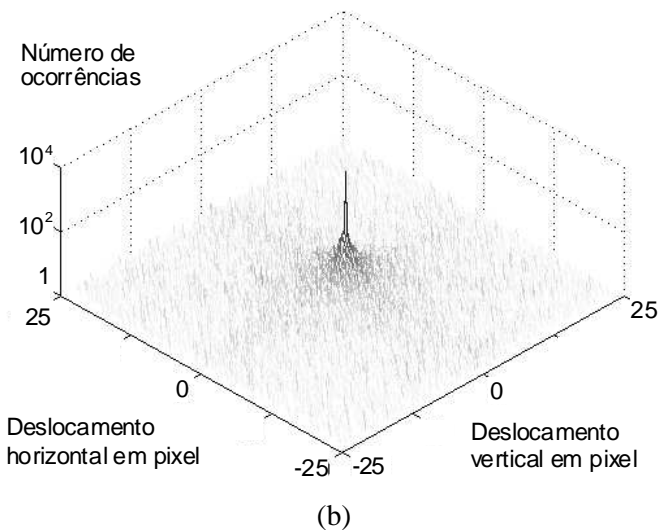
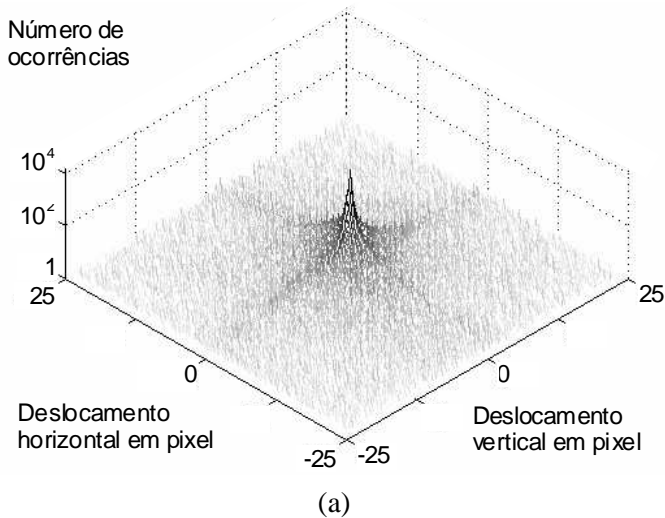
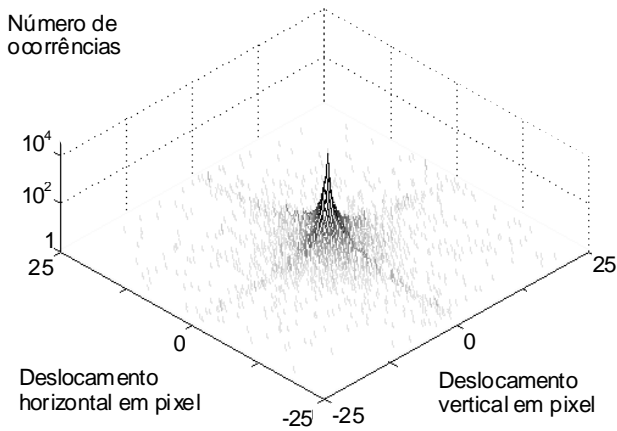
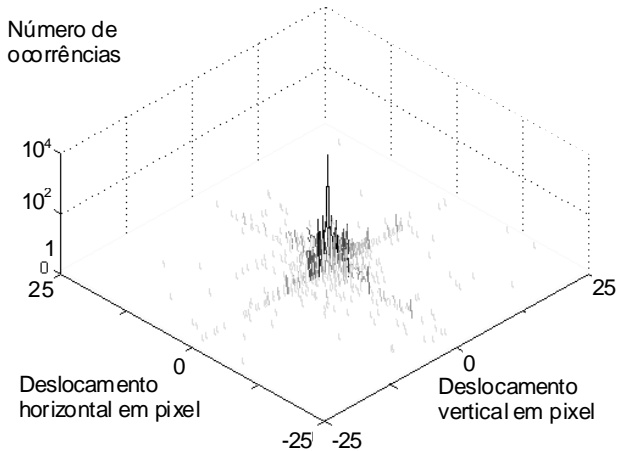


Figura 29 - Distribuição bidimensional média dos DVMs obtidos com o codificador JM7.3. (a) e (b) Otimização em distorção. (c) e (d) Otimização em taxa-distorção. (a) e (c) Quantização fina – alta taxa de bits,  $Q = 15$ . (b) e (d) Quantização grosseira – baixa taxa de bits,  $Q = 45$ .



(c)



(d)

Figura 29 (cont) - Distribuição bidimensional média dos DVMs obtidos com o codificador JM7.3. (a) e (b) Otimização em distorção. (c) e (d) Otimização em taxa-distorção. (a) e (c) Quantização fina – alta taxa de bits,  $Q = 15$ . (b) e (d) Quantização grosseira – baixa taxa de bits,  $Q = 45$ .

#### 5.4 Tamanhos de Bloco Usados na EM

A taxa dos VMs depende do número de vetores usados em um quadro. Nos padrões H.263 [16], MPEG-2 [19] e MPEG-4 [21] cada quadro é dividido em macrobloco (MB) de  $16 \times 16$  *pixels* que, por sua vez, podem ser subdivididos em blocos de  $8 \times 8$ . Nesses padrões a EM é realizada usando os blocos de  $16 \times 16$  ou  $8 \times 8$  *pixels*. Por outro lado, o padrão H.264/AVC [17] tem opções de partição do MB adicionais (ver Figura 30), podendo ser subdivididos em blocos menores de até  $4 \times 4$  *pixels*. Através desse particionamento, o padrão H.264 explora os benefícios do uso de blocos de tamanho variável, discutidos em [116]-[118], melhorando o desempenho RD em relação aos codificadores que seguem os padrões anteriores. No entanto, essa grande variedade de partições de MBs torna o processo RDO muito mais complexo, uma vez que a EM precisa ser repetida para cada partição de MB.

#### 5.5 Medida de Complexidade Computacional da EM

A complexidade computacional do processo de EM é frequentemente avaliada contabilizando o número de acessos à memória e de operações de soma, multiplicação, comparação e deslocamento de bits [40]. Esse tipo de avaliação é dependente do algoritmo usado, sendo necessário levar em conta a arquitetura do processador no qual o algoritmo é implementado. Outra forma de obter uma estimativa de complexidade é medindo o tempo de processamento [23], [50], [134], [163], [164]. Nesse caso, é importante que todos os algoritmos a serem comparados sejam implementados de forma otimizada e que os resultados sejam obtidos na mesma plataforma de *hardware*. Essa última restrição dificulta a comparação dos resultados com algoritmos apresentados na literatura, pois é necessário obter os seus códigos fonte ou então implementá-los novamente.

Neste trabalho é usado o tempo de processamento e adicionalmente uma medida simplificada de complexidade que consiste na determinação do número de pontos de procura (NPP) analisados no processo de EM. O NPP corresponde ao número de medições de distorção realizadas para cada bloco ou MB, sendo por isso usada por muitos autores [45], [106], [107], [122], [129]. Note que para cada VM avaliado é necessário determinar a distorção entre o bloco do quadro atual e o bloco do quadro de referência deslocado, sendo que esse procedimento é a etapa de maior complexidade do processo de EM. Se um único tamanho de bloco é utilizado, a comparação de complexidade

usando o NPP é direta; no entanto, se diferentes tamanhos de bloco são usados, o cálculo do NPP precisa ser ajustado de acordo com a partição do MB empregada. Para fins de avaliação, a complexidade computacional das propostas apresentadas neste trabalho é sempre comparada com a do algoritmo de referência H.264 FS RDO (JM7.3 sem modificações).

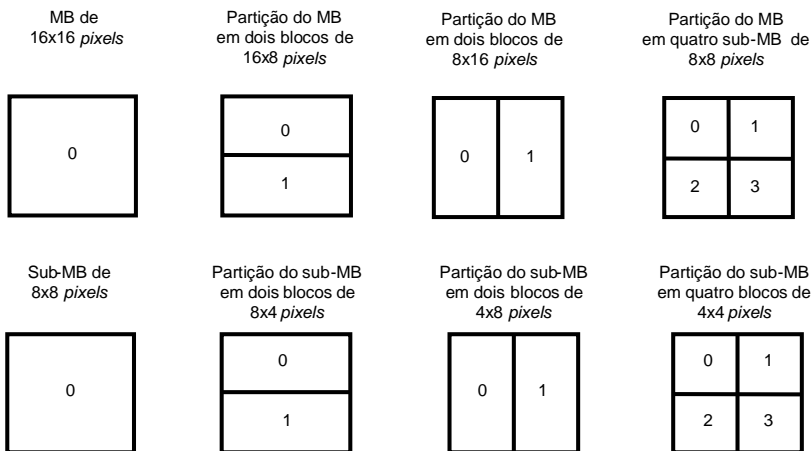


Figura 30 - Particionamento dos macroblocos no H.264. (a) MB 16×16. (b) subMB 8×8 pixels.

## 5.6 Estratégia RST na Estimação de Movimento

A obtenção do melhor compromisso RD no processo de EM implica a busca do VM de menor custo para uma dada distorção. A proposta da estratégia RST é economizar carga computacional do processo de EM ordenando os VMs por taxa de bits e parando o processo quando um desempenho RD melhor não puder ser mais obtido ou a qualidade alcançada já atingiu o valor mínimo estabelecido. Para implementar essa estratégia o ordenamento dos VMs e as regras de truncamento precisam ser estabelecidos.

### 5.6.1 Ordenamento dos VMs por Taxa

Em codificadores RDO, o ordenamento dos VMs e a avaliação em primeiro lugar dos VMs com menor taxa é vantajosa, pois é possível reduzir a complexidade computacional do processo de EM sem afetar seu desempenho RD, estabelecendo-se uma regra de truncamento



baseada na taxa. Note que, se os VM são preditos conforme explicado na Seção 5.2, a taxa da DVM deve ser usada. Para implementar a regra de truncamento baseada na taxa, primeiramente, o conjunto de todos os possíveis VMs  $\mathbf{V} = \{\vec{v}_1, \vec{v}_2, \dots\}$  é dividido em subconjuntos  $\mathbf{S}_{k_i}$  com a mesma taxa  $k_i$ , tal que

$$\mathbf{S}_{k_i} = \{\vec{v}_1, \vec{v}_2, \dots\} \mid \vec{v}_j \in \mathbf{S}_{k_i} \Leftrightarrow R(\vec{v}_j) = k_i \quad (40)$$

onde  $R(\vec{v}_j)$  é a taxa do VM  $\vec{v}_j$ .

Em seguida, os subconjuntos  $\mathbf{S}_{k_i}$  são ordenados, de acordo com as taxas  $k_1 < k_2 < \dots$ , de modo que um conjunto ordenado de VMs é obtido

$$\mathbf{P} = \{\mathbf{S}_{k_1} \prec \mathbf{S}_{k_2} \prec \dots\} \mid \mathbf{S}_{k_i} \prec \mathbf{S}_{k_j} \Leftrightarrow k_i < k_j \quad (41)$$

onde  $\prec$  indica a precedência entre os subconjuntos [161].

Note que se uma tabela VLC é usada para codificar os VMs, este ordenamento ocorre apenas uma vez antes do início do processo de EM e a complexidade computacional do processo de codificação não é afetada, uma vez que o custo computacional do ordenamento é desprezível.

### 5.6.2 Regra 1: Truncamento por Taxa

O processo de EM otimizado em RD consiste em encontrar para cada bloco de imagem o VM candidato que minimize a função custo RD

$$\vec{v}_{\text{opt}} = \arg \min_{\vec{v} \in \mathbf{V}} \{D_d(\vec{v}) + \lambda R(\vec{v})\} \quad (42)$$

onde  $\vec{v}$  representa um VM pertencente o conjunto  $\mathbf{V}$  de todos VM e  $\vec{v}_{\text{opt}}$  é o vetor ótimo.

Uma vez ordenados os VMs de acordo com (41), o processo de EM consiste em analisar os VMs de cada subconjunto  $\mathbf{S}_{k_i}$  e selecionar dentre eles o vetor  $\vec{v}_{k_{\text{opt}}}$  que produz a menor distorção, uma vez que todos esses vetores têm a mesma taxa  $k_i$ . O custo RD mínimo  $J_{k_i}(\vec{v}_{k_{\text{opt}}})$  para este subconjunto é então

$$J_{k_i}(\vec{v}_{k_{\text{opt}}}) = D(\vec{v}_{k_{\text{opt}}}) + \lambda k_i. \quad (43)$$

Após a avaliação de cada subconjunto  $\mathbf{S}_{k_i}$ , o VM ótimo provisório  $\vec{v}_{\text{opt}}$  é aquele que tem o menor custo RD  $J(\vec{v}_{\text{opt}})$  dentre os subconjuntos avaliados. Antes de avaliar um novo subconjunto a regra de truncamento verifica se o custo RD inicial mínimo  $J_{\text{init}}(k_{i+1}) = \lambda k_{i+1}$  do próximo subconjunto  $\mathbf{S}_{k_{i+1}}$  é maior do que o custo  $J(\vec{v}_{\text{opt}})$ . Assim, se

$$J(\vec{v}_{\text{opt}}) \leq J_{\text{init}}(k_{i+1}) \quad (44)$$

então o processo de EM é terminado e o custo RD ótimo é dado por  $J(\vec{v}_{\text{opt}}) = J(\vec{v}_{\text{opt}})$ . Note que o uso desta regra de truncamento reduz o espaço de procura dos VMs preservando o processo RDO, uma vez que todos os VMs não avaliados têm custo maior do que  $J(\vec{v}_{\text{opt}})$ .

Como exemplo da aplicação desta regra, considere o caso mostrado na Figura 21. Primeiramente, o VM nulo é avaliado. Em seguida, todos os VMs correspondentes a pontos à direita da linha vertical tracejada são eliminados, pois seu custo RD inicial mínimo é superior ao custo RD do VM nulo. Dessa forma, apenas mais quatro VMs (aqueles que estão à esquerda da linha vertical tracejada) precisam ser avaliados para se determinar o VM ótimo.

Esta estratégia também pode ser adotada quando múltiplos quadros de referência [17], [28], [142], [164], [165], [166] são usados. Nesse caso, a taxa de bits para a codificação da referência temporal  $R(r)$  é incluída antes de os subconjuntos  $\mathbf{S}_{k_i}$  serem ordenados em (41), calculando

$$k_i = R(\vec{v}, r) = R(r) + R(\vec{v}) \quad (45)$$

e o processo de EM RDO escolhe o par ótimo  $\{\vec{v}_{\text{opt}}, r_{\text{opt}}\}$  dentre todos os pares possíveis. Assim,

$$\{\vec{v}_{\text{opt}}, r_{\text{opt}}\} = \arg \min_{\vec{v} \in \mathbf{V}, r \in \mathbf{R}} \{D(\vec{v}, r) + \lambda R(\vec{v}, r)\} \quad (46)$$

onde  $r$  é o quadro de referência pertencente ao conjunto  $\mathbf{R}$  de todos os quadros anteriores permitidos e  $r_{\text{opt}}$  é a referência temporal do vetor ótimo.

Como consequência do uso de múltiplos quadros de referência em codificadores que usam a estratégia RST, o processo de EM alterna a procura dos VMs nos quadros de referência dependendo da taxa de bits  $R(\vec{v}, r)$ . A introdução da regra de truncamento nesse caso reduz o espaço de procura em ambas às dimensões espaciais  $x$ ,  $y$  e na dimensão temporal  $r$ . O nível de redução do espaço de busca depende do valor de  $\lambda$  e da imagem a ser codificada.

### 5.6.3 Regra 2: Truncamento por Qualidade

Conforme mostrado anteriormente a influência do processo de EM na qualidade da imagem obtida após a CM depende do bloco codificado. Assim, de acordo com a imagem e a qualidade mínima requerida, é possível estabelecer uma regra de truncamento, usando um limiar de distorção  $D_L$  para parar a EM quando uma qualidade mínima desejada já foi alcançada. A adoção dessa regra na estratégia RST reduz significativamente a complexidade computacional do processo à custa de uma pequena perda de desempenho RD. Portanto, o processo de EM pode ser interrompido se

$$J(\vec{v}_{\text{opt}}) \leq J_{\text{init}}(k_{i+1}) + D_L \quad (47)$$

onde  $J(\vec{v}_{\text{opt}})$  é o menor custo RD para o limiar de distorção  $D_L$ . Se  $D_L = 0$ , a *Regra 1* é considerada e a *Regra 2* não é aplicada.

## 5.7 Estratégia RST na Seleção do Modo de Codificação

Codificadores de vídeo híbridos admitem múltiplos modos de codificação (*Skip*, *Inter*, *Intra*) para cada bloco de imagem e suportam vários tamanhos de bloco nos modos *Inter* e *Intra*. Para selecionar a melhor partição de MB nesses codificadores é necessário determinar a taxa de bits e a distorção para cada modo. Por isso o uso de um grande número de modos de codificação e diversos tamanhos de bloco nos codificador que seguem o padrão H.264 leva a uma alta carga computacional quando o processo é RDO.

Com a abordagem RST é possível eliminar antecipadamente alguns modos de codificação, considerando no cálculo do custo RD os

bits que são usados para indicar o modo de codificação  $R(m)$  e a taxa usada para codificar o conjunto de VMs  $R(v)$  de cada bloco que constitui o MB. Dependendo da sintaxe do fluxo de bits do codificador, pode-se obter uma taxa mínima inicial  $R_{\text{init}}(m_i)$  para cada modo de partição  $m_i$ , que corresponde ao número de bits obrigatórios para a codificação do MB. Assim, um custo RD inicial mínimo  $J_{\text{init}}(m_i)$  pode ser determinado por

$$J_{\text{init}}(m_i) = \lambda R_{\text{init}}(m_i). \quad (48)$$

Os valores de custo RD para todos os modos  $m_i$  podem então ser ordenados

$$\mathbf{M} = \{m_1 < m_2 \dots\} \mid m_i < m_j \Leftrightarrow J_{\text{init}}(m_i) \leq J_{\text{init}}(m_j) \quad (49)$$

$$\mathbf{J}_{\text{init}} = \{J_{\text{init}}(m_1), J_{\text{init}}(m_2), \dots\} \quad (50)$$

e este conjunto ordenado  $\mathbf{J}_{\text{init}}$  pode ser usado para eliminar antecipadamente alguns modos de codificação de maior custo RD.

O modo *Skip*, que tem o menor custo RD inicial mínimo, é o primeiro modo  $m_1$  no arranjo  $\mathbf{M}$ , os modos *Intra* são os últimos do arranjo  $\mathbf{M}$ , enquanto os modos *Inter* têm a posição dependente do número de blocos que formam o MB. Note que este ordenamento é realizado apenas uma vez antes de iniciar a codificação do vídeo, não afetando a complexidade computacional.

### 5.7.1 Pré-Seleção do Modo *Skip*

Na codificação de vídeo otimizada em RD, a frequência de uso do modo *Skip* aumenta à medida que a taxa total de bits é reduzida e uma menor qualidade de imagem é permitida. A percentagem de MBs codificados no modo *Skip* (*MB-Skip*) depende da quantidade de movimento e de complexidade da imagem. A Tabela 10 mostra essa percentagem para diversas sequências de vídeo considerando três valores do parâmetro de quantização  $Q$ . Em imagens com fundo estático nas quais o movimento existente é devido aos interlocutores, como por exemplo, *Akiyo* e *Mother&Daughter*, o percentual de uso do modo *Skip* é alto. Note que o uso de uma quantização mais grosseira eleva ainda mais esse percentual. Por outro lado, em cenas com câmera

não estacionária, tais como *Coastguard*, *Foreman* e *Carphone*, esse percentual é bem menor.

Tabela 10 – Percentagem de MBs em modo *Skip* para o codificador JM7.3-RDO

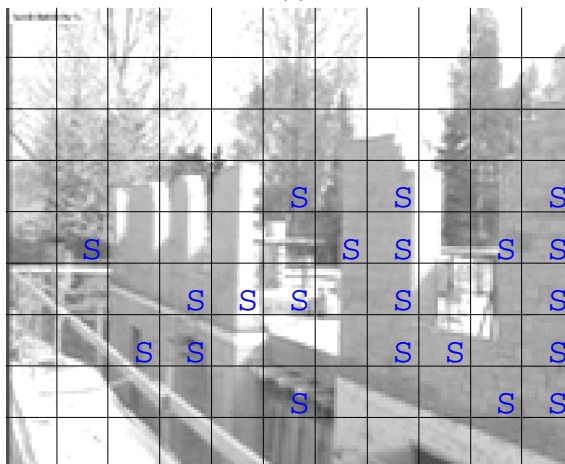
Sequências	Parâmetro de Quantização, $Q$		
	20	30	40
<i>Akiyo</i>	68,2 %	79,1 %	89,7 %
<i>Mother&amp;Daughter</i>	35,5 %	51,3 %	77,2 %
<i>Foreman</i>	7,4 %	23,0 %	45,0 %
<i>Coastguard</i>	2,3 %	15,8 %	53,4 %
<i>Carphone</i>	9,7 %	28,4 %	56,5 %
<b>Média</b>	<b>24,6%</b>	<b>39,5 %</b>	<b>64,4 %</b>

A explicação para a grande quantidade de MBs *Skip* pode encontrada através da análise de algumas sequências de vídeo. A Figura 31 (a) ilustra um vídeo com fundo estático, no qual ocorrem apenas pequenos movimentos, levando a 94% de MB-*Skip*. No vídeo ilustrado na Figura 31 (b) são encontrados 24% de MB-*Skip* apesar ser uma sequência de vídeo no qual a câmera está em movimento. No primeiro caso, o grande número de MB-*Skip* é naturalmente esperado; porém, no segundo caso, o percentual é inesperado, pois não existem MBs estáticos. O modo *Skip* nesse caso apenas ocorre porque o padrão H.264 associa o VM predito aos MB-*Skip*, assim o bloco de imagem do quadro anterior é copiado levando em conta o deslocamento indicado pelo VM predito.

Apesar desta grande percentagem de MBs codificados no modo *Skip*, geralmente a decisão sobre o seu uso é postergada até que o custo RD de todos os modos (*Inter* e *Intra*) tenha sido determinado e verificado que o modo *Skip* é o de menor custo RD. Usando a abordagem RST a escolha do modo *Skip* pode ser antecipada em muitos MBs, evitando a EM (modos *Inter*) e a predição dos modos *Intra*. A quantidade de MBs pré-selecionados e conseqüente redução de complexidade dependem do conteúdo da imagem codificada, do parâmetro de quantização  $Q$  e do custo RD obtido para o modo *Skip*.



(a)



(b)

Figura 31 - Exemplos de MB-Skip (marcados com S) usando o codificador H.264 RDO FS. (a) Quadro #210 da sequência *Mother&Daughter* com  $Q = 30$ . (b) Quadro #237 da sequência *Foreman* com  $Q = 28$ .

Para decidir sobre a pré-seleção do modo *Skip*, inicialmente, é calculado o custo RD do modo *Skip*

$$J(\text{Skip}) = D(\text{Skip}) + \lambda R(\text{Skip}) \quad (51)$$

onde  $D(Skip)$  é a distorção entre o MB do quadro atual e os *pixels* apontados pelo VM predito no quadro de referência e  $R(Skip)$  é a taxa do MB para modo *Skip*. Em seguida, o custo RD  $J(Skip)$  é comparado com o custo RD inicial mínimo  $J_{init}(m_2)$ , onde  $m_2$  é o modo  $Inter_{16 \times 16}$ . Assim, o modo *Skip* é pré-selecionado, se

$$J(Skip) \leq J_{init}(m_2) + D_L \quad (52)$$

pois todos os outros modos têm um custo RD maior do que o modo *Skip*. Note que  $D_L$  é o limiar de qualidade definido anteriormente em (47).

No caso do H.263, o modo *Skip* é codificado com um bit  $R(Skip) = 1$ . Para o modo  $m_2$  são necessários no mínimo 4 bits para a indicação do modo e 2 bits adicionais para codificação do VM predito, portanto  $R_{init}(m_2) = 6$ . Substituindo esses valores em (48), (51) e (52), obtemos a condição para a qual o modo *Skip* é pré-selecionado. Assim,

$$D(Skip) \leq 0,85Q^2[R(m_2) - R(Skip)] + D_L \quad (53)$$

ou

$$D(Skip) \leq 4,25Q^2 + D_L. \quad (54)$$

No caso do H.264, o modo *Skip* é codificado através de um código *run-length* que especifica o número de MBs *Skip* sequenciais [17]; dessa forma, o número de bits necessário por MB é menor ou igual a um. Para o modo  $m_2$ , ao menos 4 bits são necessários para indicar o modo e 2 bits para codificar o VM predito, portanto  $R_{init}(m_2) = 4$ . Assim, o modo *Skip* é pré-selecionado quando

$$D(Skip) \leq 3,4 \times 2^{(Q-12)/6} + D_L. \quad (55)$$

Note que o parâmetro de quantização  $Q$  controla a prioridade do modo *Skip* privilegiando-o em baixas taxas de bits.

### 5.7.2 Pré-Eliminação de Partições no Modo *Inter*

Em codificadores de vídeo que usam vários tamanhos de bloco, uma estrutura de macrobloco (MB) é definida para agrupá-los e construir um bloco final de  $16 \times 16$  *pixels*. Assim,

$$MB = \{B_1, B_2, \dots\} \quad (56)$$

onde  $B_i$ ,  $i = 1, 2, \dots$ , são os blocos.

Na otimização RD o codificador geralmente realiza o processo de EM para cada tamanho de bloco, obtendo o VM ótimo  $\vec{v}_{B_i, \text{opt}}$  para cada bloco  $B_i$ . Uma vez que o processo de EM seja completado, o custo RD  $J_{MB}(p)$  é calculado para cada partição do MB. Assim,

$$J_{MB}(p) = \sum_{\forall B_i \in MB} D(\vec{v}_{B_i, \text{opt}}) + \lambda \left[ \sum_{\forall B_i \in MB} R(\vec{v}_{B_i, \text{opt}}) + R(p) \right] \quad (57)$$

onde  $D(\vec{v}_{B_i, \text{opt}})$  denota a distorção após a compensação de movimento,  $R(\vec{v}_{B_i, \text{opt}})$  é a taxa do VM ótimo de cada bloco e  $R(p)$  é o número de bits usados para designar a partição  $p$ . Ao final do processo de otimização, a partição do MB que resultar no menor custo RD é selecionada.

*a) Pré-eliminação de tamanhos de bloco*

Usando a estratégia RST, algumas combinações de blocos acima mencionadas podem ser pré-eliminadas. Para tal, o processo de EM é feito na ordem de teste definida em (49), considerando um custo RD inicial mínimo associado a cada partição. Assim,

$$J_{\text{init}}(m|p) = \lambda \left[ \sum_{\forall B_i \in MB} R(\vec{v}_{B_i, \text{pred}}) + R_{\text{init}}(m|p) \right] \quad (58)$$

onde  $R_{\text{init}}(m|p)$  representa o número mínimo de bits necessário para dividir o MB na partição  $p$  do modo *Inter* e  $R(\vec{v}_{B_i, \text{pred}})$  é a taxa mínima do VM predito para o bloco  $B_i$ .

O custo RD inicial mínimo obtido é proporcional ao número de VMs preditos e ao número de blocos que formam o MB. Quando a EM inicia com blocos maiores, os menores podem ser evitados se a seguinte condição é satisfeita:

$$\arg \min_{m_i \in \mathbf{M} \mid i < j} \{J_{MB, \text{min}}(m_i)\} \leq J_{\text{init}}(m_j) + D_L \quad (59)$$



onde  $m_j$  é o modo *Inter* sob avaliação e  $\arg \min_{m_i \in \mathbf{M} \mid i < j} \{J_{\text{MB}, \min}(m_i)\}$  é o custo



RD mínimo obtido até então.

Quando (59) é verdadeira a EM é finalizada, uma vez que o custo RD obtido é menor do que o custo RD inicial mínimo para as partições com blocos menores ainda não avaliadas no arranjo  $\mathbf{M}$ .

*b) Aplicação aos padrões H.263 e H.264*

Para aplicar a estratégia RST em codificadores que seguem os padrões de vídeo H.263 e H.264, o custo RD inicial mínimo (58) precisa ser determinado. O número de bits obrigatórios por MB para cada tipo de partição *Inter* é mostrado nas Tabelas 11 e 12. Esses valores asseguram a otimização em RD do processo de EM, porém custos RD iniciais maiores baseados na estatística da imagem podem ser também considerados visando uma maior redução de complexidade da EM, desde que seja admitida uma pequena perda no desempenho RD.

Tabela 11 – Bits obrigatórios na codificação MB-*Inter* para as partições segundo o padrão H.263++

partição do MB	número de MVs	bits por MVs <sup>IV</sup>	bits por modo <sup>V</sup>	outros bits <sup>VI</sup>	ref. (bits) <sup>VII</sup>	$J_{\text{init}}(m)$
<i>Inter</i> <sub>16x16</sub> 	1	2	1	3	$R_r$ <sup>VIII</sup>	$\lambda(6 + R_r)$
<i>Inter</i> <sub>8x8</sub> 	4	10	3	3	$4R_r$	$\lambda(18 + 4R_r)$

<sup>IV</sup> O menor VM diferença para blocos 16x16 é (0, 0). Para blocos 8x8 pelo menos um VM deve ser não-nulo.








<sup>V</sup> Ver VLC Table.8 [16].

<sup>VI</sup> 1 bit para *coded\_macroblock\_indication* (COD) e  
2 bits para *coded\_block\_pattern\_for\_luminance* (CBPY).

<sup>VII</sup> Ver Table U.1 [16].

<sup>VIII</sup>  $R_r$  é o número de bits necessário para codificar a referência temporal

Tabela 12 – Bits obrigatórios na codificação MB-*Inter* para as partições segundo o padrão H.264

partição do MB	número de MVs	bits por MVs <sup>IX</sup>	bits por modo <sup>X</sup>	outros bits <sup>XI</sup>	ref. (bits) <sup>XII</sup>	$J_{\text{init}}(m)$
<i>Inter</i> <sub>16x16</sub> 	1	4	1	2	$R_r$	$\lambda(7 + R_r)$
<i>Inter</i> <sub>16x8</sub> 	2	6	3	2	$2R_r$	$\lambda(11 + 2R_r)$
<i>Inter</i> <sub>8x16</sub> 	2	6	3	2	$2R_r$	$\lambda(11 + 2R_r)$
<i>Inter</i> <sub>8x8</sub> 	4	10	9	2	$4R_r$	$\lambda(21 + 4R_r)$
<i>Inter</i> <sub>8x4</sub> 	5	12	11	2	$4R_r$	$\lambda(25 + 4R_r)$
<i>Inter</i> <sub>4x8</sub> 	5	12	11	2	$4R_r$	$\lambda(25 + 4R_r)$
<i>Inter</i> <sub>4x4</sub> 	7	16	11	2	$4R_r$	$\lambda(29 + 4R_r)$

### 5.7.3 Pré-Eliminação do Modo *Intra*

Para avaliar se o uso do modo *Intra* em algum MBs resulta em um desempenho RD superior ao obtido com os modos *Skip* e *Inter*, é necessário codificar o MB usando todas as direções de predição do modo *Intra* e comparar o seu custo RD com aquele obtido para os outros modos.

A variedade de modos *Intra* disponível depende do tipo de codificador empregado. Apenas um modo *Intra* é disponível para o codificador H.263 *baseline* e 3 modos *Intra* são disponíveis se o Anexo I do H.263 é considerado. No caso de codificador H.264, existem 592 combinações de modos *Intra*, resultando em uma elevadíssima complexidade computacional do processo de predição nesse tipo de codificador [151], [167], [168].

#### a) Medindo a importância do modo *Intra*

Para avaliar o efeito do uso do modo *Intra* em codificadores de vídeo, a frequência de ocorrência e diferença de desempenho RD devido ao seu uso ou não foram determinados usando o codificador H.264 RDO

<sup>IX</sup> Considera-se pelo menos um VM diferente do VM predito.

<sup>X</sup> Considera-se o menor número de bits. Por exemplo: quando o modo *Inter*<sub>4x4</sub> é usado, considera-se apenas um sub-MB dividido em 4 blocos 4x4, enquanto os outros sub-MBs são codificados no modo *Inter*<sub>8x8</sub>, conforme ilustrado na primeira coluna da Tabela 12.

<sup>XI</sup> 1 bit para *coded\_block\_pattern* e 1 bit para *mb\_skip\_run*.

<sup>XII</sup> Número mínimo de bits para a codificação da referência usada no bloco.

FS. Evidências estatísticas mostram que independente do conteúdo ou da qualidade requerida para a imagem, a quantidade de MBs codificados em modo *Intra* permanece abaixo de 5% (ver Tabela 13). Quando são consideradas apenas as sequências de videoconferência com câmera fixa, tais como *Mother&Daughter* e *Akiyo*, essa percentagem é ainda menor, abaixo de 0,5%. Resultados similares de percentagem de uso do modo *Intra* têm sido obtidos para outras sequências de vídeo por outros autores [159].

Tabela 13 – Percentagem de MB codificados em modo *Intra* no codificador H.264 RDO FS

Sequências	Parâmetro de Quantização, $Q$		
	20	20	20
<i>Akiyo</i>	0,0 %	0,0 %	0,0 %
<i>Mother&amp;Daughter</i>	0,3 %	0,3 %	0,3 %
<i>Carphone</i>	2,9 %	2,9 %	2,9 %
<i>Foreman</i>	1,8 %	1,8 %	1,8 %
<i>Coastguard</i>	1,6 %	1,6 %	1,6 %
<b>Média</b>	<b>1,3 %</b>	<b>1,3 %</b>	<b>1,3 %</b>

Apesar desta baixa percentagem de MBs codificados no modo *Intra*, o tempo de processamento despendido pelo codificador na sua avaliação é muito alto. Por exemplo, quando uma referência temporal única é usada na EM, até 38% do tempo total de processamento é utilizado na avaliação dos modos *Intra* (ver Tabela 14).

Tabela 14 – Percentagem de tempo de processamento despendido na seleção dos modos *Intra* e *Inter* usando um codificador H.264 RDO FS com memória de curto prazo (referência temporal única)

Sequências	Tempo de processamento	
	modo <i>Intra</i>	modo <i>Inter</i>
<i>Akiyo</i>	37,5 %	59,7 %
<i>Mother&amp;Daughter</i>	33,9 %	63,4 %
<i>Foreman</i>	31,3 %	65,1 %
<i>Coastguard</i>	30,1 %	67,4 %
<i>Carphone</i>	33,8 %	63,3 %
<b>Média</b>	<b>33,3 %</b>	<b>63,8 %</b>

Tabela 15 – Percentagem de tempo de processamento despendido na seleção dos modos *Intra* e *Inter* usando um codificador H.264 RDO FS com memória de longo prazo (5 referências temporais)

Sequências	Tempo de processamento	
	modo <i>Intra</i>	modo <i>Inter</i>
<i>Akiyo</i>	14,7 %	85,4 %
<i>Mother&amp;Daughter</i>	11,4 %	87,9 %
<i>Foreman</i>	10,8 %	88,7 %
<i>Coastguard</i>	8,1 %	90,1 %
<i>Carphone</i>	9,0 %	87,5 %
<b>Média</b>	<b>10,8 %</b>	<b>87,9 %</b>

Mesmo com memória de longo prazo de 5 quadros de referência temporal, o codificador ainda consome entre 8% a 15% (ver Tabela 15) do tempo de processamento na avaliação dos modos *Intra*. A redução do percentual de tempo de processamento consumido na predição *Intra*, nesse caso, é devido a uma maior complexidade do processo de EM.

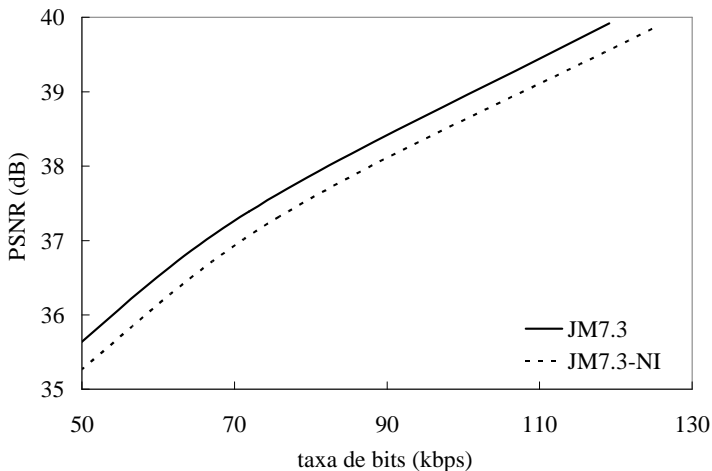


Figura 32 - Curva RD para o codificador H.264 RDO FS usando memória de longo prazo com 5 referências temporais para os 10 segundos iniciais da sequência *Foreman*. (Linha contínua) JM7.3 com o modo *Intra* habilitado. (Linha pontilhada) com modo *Intra* desabilitado.

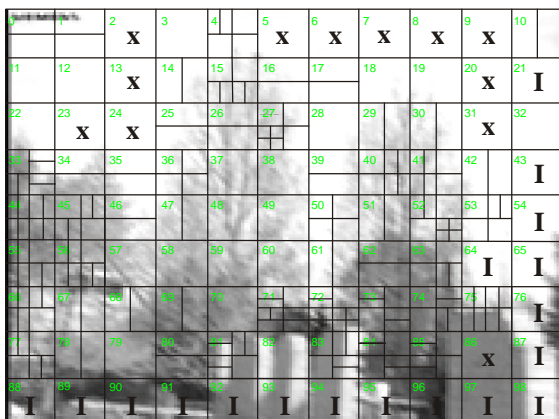
Comparando o número de MBs efetivamente codificados no modo *Intra* com o percentual de tempo despendido na sua avaliação, se percebe que o tempo não é bem empregado. Por outro lado, a codificação *Intra* não pode ser simplesmente deixada de lado, uma vez que em algumas situações o ganho em desempenho RD devido ao seu uso pode resultar em até 0,4 dB (ver Figura 32).

*b) Situações nas quais o modo Intra é utilizado*

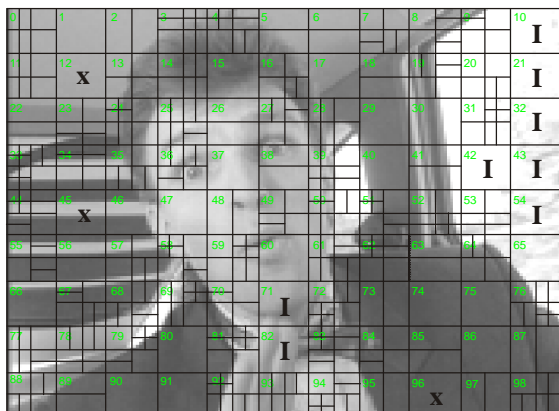
Para um melhor entendimento da importância do modo *Intra* nos codificadores que seguem o padrão H.264, na Figura 33 são ilustrados algumas situações nas quais os MBs são codificados no modo *Intra*. Essas situações podem ser agrupadas em:

- i) MBs que representam objetos que estavam fora do campo de visão nos quadros anteriores e se tornaram visíveis devido ao movimento de câmera [por exemplo, os MBs assinalados com **I** na borda na Figura 33(a)].
- ii) MBs que representam objetos que estavam escondidos e que apareceram devido ao movimento de translação ou rotação de algum objeto da cena [por exemplo, os MBs assinalados com **I**, correspondentes a mão na Figura 33(b)].
- iii) MBs correspondentes a novos objetos que não estavam anteriormente na cena [por exemplo, os MBs assinalados com **I** correspondentes a paisagem na janela do carro na Figura 33(b)].
- iv) MBs que representam objetos já existentes na cena para os quais há uma adequada predição *Inter*; no entanto, o uso da predição *Intra* resulta em um desempenho RD superior [os MBs assinalados com **x** na Figura 33(a) e (b)].

Nas situações (i) a (iii), o uso do modo *Intra* para a codificação do MB é a solução mais indicada, uma vez que estes MBs não possuem uma predição *Inter* adequada. Já na situação (iv), eventualmente, o uso do modo *Inter* no lugar do modo *Intra* poderia resultar em uma melhor predição dos VMs, uma vez que os MBs *Intra* não têm VMs associados e assim prejudicam a predição dos VMs dos MBs vizinhos.



(a)



(b)

Figura 33 - Exemplos de MBs codificados no modo *Intra* (marcados com **I** ou **x**) usando o codificador H.264 RDO FS. (a) Quadro #210 da sequência *Foreman*. (b) Quadro #177 da sequência *Carphone*.

### c) Pré-eliminação da predição de luminância

Conforme mostrado nas seções anteriores, o modo *Intra* é pouco usado nos codificadores RDO (ver Tabela 13); no entanto, em algumas sequências de vídeo eles têm uma grande importância no desempenho RD. Com o objetivo de evitar a sua avaliação na maioria dos casos, uma

estratégia RST que permite pré-eliminar várias direções de predição de luminância do modo *Intra* é apresentada nesta seção. A estratégia consiste em pré-eliminar os modos *Intra* quando o seu custo RD inicial mínimo é superior ao menor custo RD já obtido para o mesmo MB com outro modo de codificação. Inicialmente, o menor custo RD  $J_{\text{MB,min}}(m)$  obtido com os modos *Skip* e *Inter* é comparado com o custo RD inicial mínimo  $J_{\text{init}}(m_{\text{Intra}}) = \lambda R_{\text{init}}(m_{\text{Intra}})$  de cada modo *Intra*. Em codificadores H.264, essa taxa mínima  $R_{\text{init}}(m_{\text{Intra}})$  é 9 bits para o modo  $\text{Intra}_{16 \times 16}$  e 27 bits para os modos  $\text{Intra}_{4 \times 4}$ . Essa estratégia permite pré-eliminar todos os modos *Intra* para MBs com

$$J_{\text{init}}(m_{\text{Intra}}) \geq J_{\text{MB,min}}(m) \quad (60)$$

mantendo o mesmo desempenho RD.

Adicionalmente, resultados experimentais obtidos com várias sequências de vídeo mostram que é possível estabelecer um custo RD inicial *Intra*  $J_1$  superior a este mínimo, levando a uma considerável redução de complexidade computacional com uma pequena perda no desempenho RD. Tal fato pode ser justificado pelo uso do modo *Intra* nas situações (i) a (iii) anteriormente mencionados, nas quais há a necessidade de codificação de muitos coeficientes da transformada inteira do erro de predição, exigindo uma taxa de bits muito superior a mínima. Na situação (iv), na qual poucos coeficientes são codificados, o uso de uma  $J_1$  maior resulta em aumento da prioridade para codificação no modo *Skip* e *Inter*, podendo melhorar a predição dos VMs subsequentes.

O custo RD inicial *Intra*  $J_1$  é obtido considerando o modelo de distorção obtido para a faixa de taxa de bits da aplicação. A Figura 34 mostra os resultados obtidos para várias sequências QCIF (*Foreman*, *Carphone*, *Coastguard*, *News*, *Akiyo*, *Mother&Daughter* e *HallObjects*) com o codificador H.264 RDO FS. Para manter a perda de desempenho equilibrada em toda a faixa de taxa de bits, é adotado o seguinte modelo de custo  $J_1$ :

$$J_1 = k_1 Q^4 \times 10^{-3} \quad (61)$$

onde  $k_1$  define o nível de prioridade *Inter* sobre os modos *Intra*. Na expressão (61), a potência 4 em  $Q^4$  é o valor inteiro arredondado do resultado obtido para o melhor ajuste do modelo de potência.

A Figura 34 mostra as curvas do modelo para diferentes valores de  $k_1$ . Note que quanto menor for  $k_1$ , mais próximo será o desempenho RD da estratégia RST em relação ao codificador RDO FS (ver Figura 37). Se um valor de  $k_1$  muito alto for usado, o modo *Intra* será preterido e o desempenho do codificador RST se aproximará do obtido pelo codificador com o modo *Intra* desabilitado.

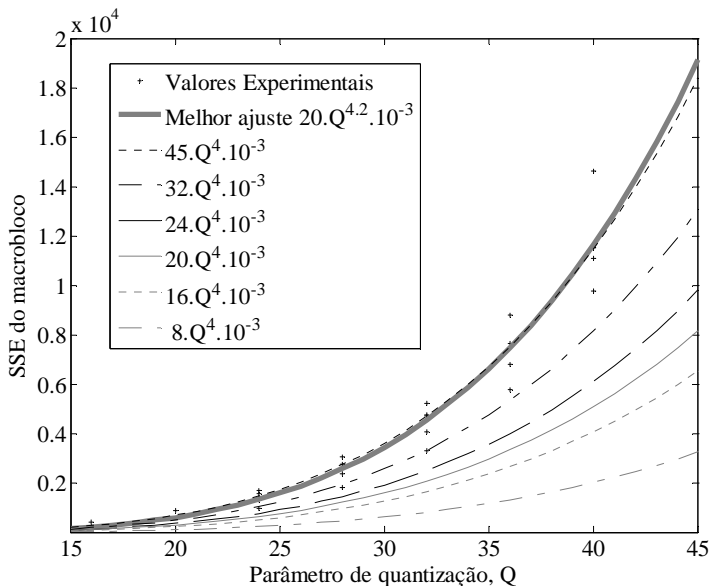


Figura 34 - Curvas dos modelos de custo RD inicial *Intra*  $J_1$  para diferentes níveis  $k_1$  e valores experimentais obtidos usando o codificador H.264 RDO FS. (Linha cinza grossa) melhor ajuste do modelo de potência.

#### d) Pré-eliminação da predição de cor

Nos codificadores baseados no padrão H.264, além das várias possibilidades de predições do componente de luminância, existem para cada uma delas também 4 formas de predição espacial para os componentes de crominância  $p_{DC}$ ,  $p_V$ ,  $p_H$  e  $p_P$  (DC, Vertical, Horizontal e Plana) [17]. Essas predições de crominância aumentam ainda mais a



complexidade final do processo de predição *Intra*. Através da análise do custo RD de várias sequências de vídeo, percebeu-se que as diferentes predições de crominância resultam em variações mínimas de custo RD entre elas, conforme ilustram os agrupamentos de pontos destacados na Figura 26. Por isso, uma adicional simplificação de complexidade da seleção dos modos *Intra* pode ser conseguida avaliando-se apenas uma das predições de crominância  $p_{DC}$  e descartando as outras se

$$J_{MB,p_{DC}}(m_{Intra}) > J_{MB,min}(m) \quad (62)$$

onde  $J_{MB,p_{DC}}(m_{Intra})$  denota o custo RD do MB codificado no modo *Intra* com predição de crominância  $p_{DC}$  e  $J_{MB,min}(m)$  representa o mínimo custo RD com os modos *Skip* e *Inter*.

Nesta condição a perda de desempenho RD é mínima, pois as outras 3 formas de predição de crominância descartadas ( $p_V$ ,  $p_H$  e  $p_P$ ) têm um custo RD similar ao da predição  $p_{DC}$ . Note que se a condição (62) não é satisfeita todas as predições são avaliadas.

#### 5.7.4 Seleção Final dos Modos de Codificação

A seleção final do modo de codificação de cada MB na estratégia RST é feita avaliando-se o custo RD dos modos que não foram descartados nas etapas anteriores. Essas etapas são pré-seleção do modo *Skip*, pré-eliminação dos modos *Inter*, pré-eliminação da predição da luminância e pré-eliminação da predição de crominância dos modos *Intra*. Nenhuma avaliação adicional será necessária se o modo *Skip* é pré-selecionado, caso contrário, o modo ótimo  $m_{opt}$  do MB é escolhido dentre os modos  $m_i$  que não foram descartados, minimizando o custo RD

$$m_{opt} = \min_{m_i \in M} \{D_{2,REC}(m_i | Q) + \lambda^2 R_{REC}(m_i | Q)\} \quad (63)$$

onde  $D_{2,REC}$  é a distorção medida em SSD entre os *pixels* originais e os reconstruídos e  $R_{REC}$  é o número total de bits usados.

O critério de minimização é o mesmo usado na estratégia FS [91], porém, na estratégia RST ele é calculado sobre um número reduzido de modos. Note que se o modo *Skip* não for pré-selecionado, então ele pode ainda ser selecionado nesta etapa, desde que ele seja o de menor custo RD dentre todos os modos de codificação.

## 5.8 Resultados Experimentais

Para avaliar o desempenho estratégia RST proposta, são consideradas algumas sequências de vídeo QCIF bem conhecidas a taxa de quadros de 10 fps. Essas sequências incluem três cenas típicas de videoconferência usando câmera fixa: *Akiyo*, *News*, e *Mother&Daughter*, três cenas com câmera não estacionária: *Coastguard*, *Foreman*, e *Carphone* e uma cena de vigilância eletrônica com câmera fixa *HallObjects*. A abordagem proposta foi implementada, através de uma modificação na implementação JM7.3 RDO FS do H.264, sendo o codificador *baseline* dessa implementação [91] usado com referência nas avaliações de desempenho. Na configuração de ambos os codificadores é usada uma estrutura de grupo de imagens do tipo IPPP..., memória de longo prazo com 5 quadros de referência (LT5) ou memória de curto prazo com referência única (ST1), seleção do modo otimizada em RD, EM com precisão de  $\frac{1}{4}$  de pixel e janela de procura de  $\pm 16$  pixels e codificação de entropia com VLC. O parâmetro de quantização  $Q$  foi variado para escolher ou a taxa ou a distorção desejada. A PSNR foi calculada através da média dos 100 quadros iniciais de cada sequência de vídeo avaliada.

Para analisar a estratégia RST, é considerada a procura dos VMs (incluindo a *Regra 1* e *Regra 2*) e a seleção do modo de codificação (incluindo a pré-seleção do modo *Skip*, pré-eliminação dos modos *Inter* e *Intra*). A abordagem proposta é avaliada visando verificar a redução de complexidade obtida e o ganho ou perda de desempenho RD.

### 5.8.1 Estimação de Movimento e Regras de Truncamento

A Figura 35 mostra a variação de complexidade computacional em relação ao desempenho em PSNR com a aplicação das regras de truncamento por taxa (*Regra 1*) e por qualidade (*Regra 2*). O uso da *Regra 1* (com  $D_L$  nulo) na abordagem RST resulta em uma significativa redução de complexidade, de até 90% em algumas sequências de vídeo. Aplicando também a *Regra 2* (com valores de  $D_L$  não nulos) é possível limitar ainda mais a complexidade da EM, obtendo uma redução adicional de até 10% sem aumento significativo da distorção (menor do que 0,1 dB) para valores de  $D_L$  menores do que 200 (ver Tabela 20). A Figura 35 mostra que para melhores qualidades de imagem uma menor redução de carga computacional é obtida pelo uso da *Regra 1*. Note que o uso da *Regra 2* compensa esta menor redução, afetando em maior nível a complexidade computacional nas imagens de melhor qualidade.

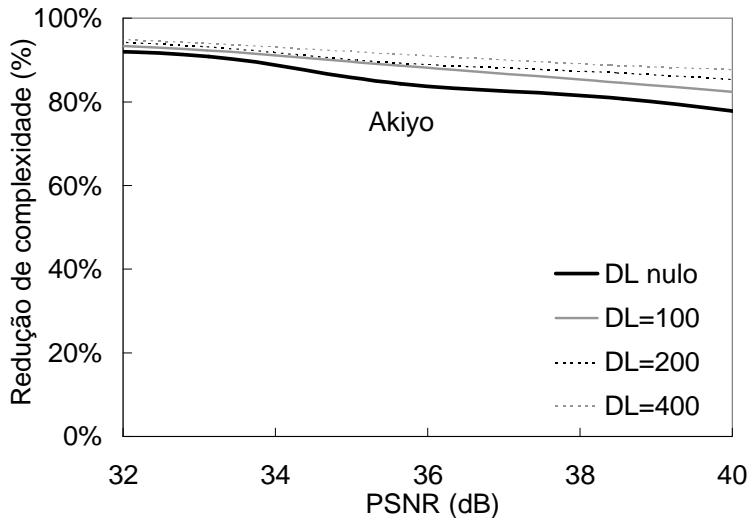
### 5.8.2 Pré-Seleção do Modo *Skip*

Aplicando a pré-seleção do modo *Skip* da abordagem RST em sequências típicas de videoconferência resulta em até 80% de MBs pré-selecionados [ver Figura 36(a)]. Em sequências de vídeo adquiridas com câmera não estacionária, essa percentagem permaneceu significativa (até 30%) [ver Figura 36(b)]. Note que essa redução de complexidade é obtida sem perda de desempenho RD, uma vez que a pré-seleção do modo *Skip* é aplicada apenas quando nenhum outro modo pode melhorar o desempenho.

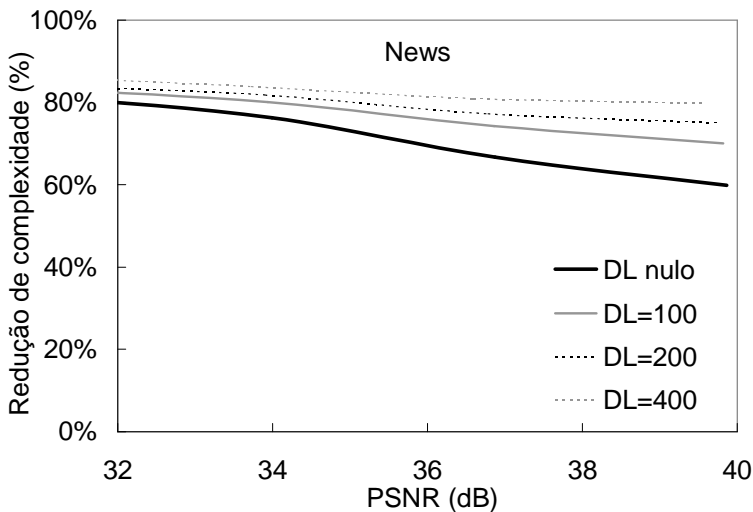
### 5.8.3 Pré-Eliminação do Modo *Inter*

A Tabela 16 mostra o ganho em complexidade computacional obtido com o uso da abordagem RST aplicando simultaneamente a *Regra 1* na EM, a pré-seleção do modo *Skip* e a pré-eliminação dos modos *Inter*. Note que o ganho é maior para valores maiores de  $Q$  quando a qualidade da imagem é menor. A maior redução de complexidade é obtida para as sequências de câmera fixa (até 94%); no entanto, mesmo para câmeras não estacionárias uma considerável redução de complexidade é obtida (de 10 a 65%). Note que essa economia de carga computacional é obtida mantendo o processo otimizado em RD, conforme o desempenho em PSNR mostrado na Tabela 17.

Aplicando também a *Regra 2* (codificador FULLRST, conforme definido na Seção 5.8.5), é obtida uma redução adicional de complexidade sem perda significativa de desempenho RD (menor do que 0,1 dB). A Tabela 24 mostra a percentagem de redução de complexidade, enquanto a Tabela 20 ilustra o desempenho em PSNR.

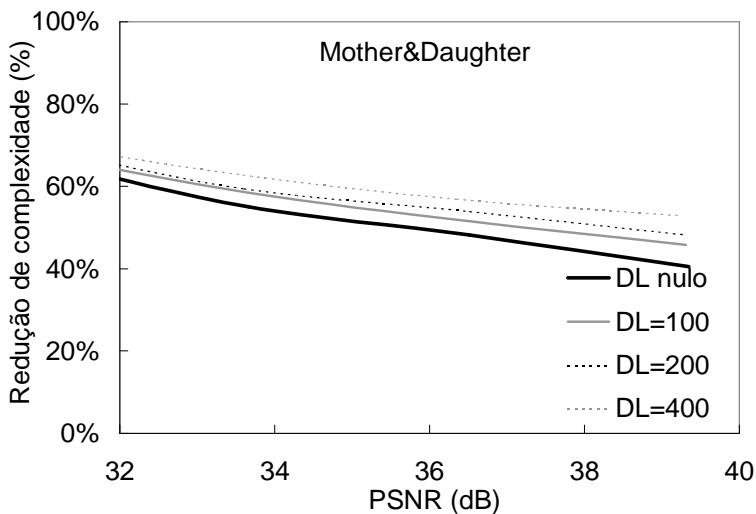


(a)

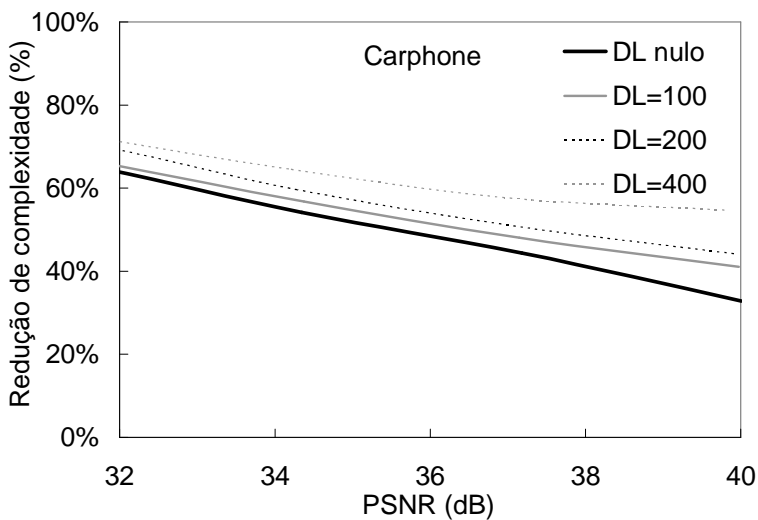


(b)

Figura 35 - Efeito do limiar de distorção na complexidade da EM. Para seqüências de vídeo (a) *Akiyo*, (b) *News*, (c) *Mother&Daughter* e (d) *Carphone*.

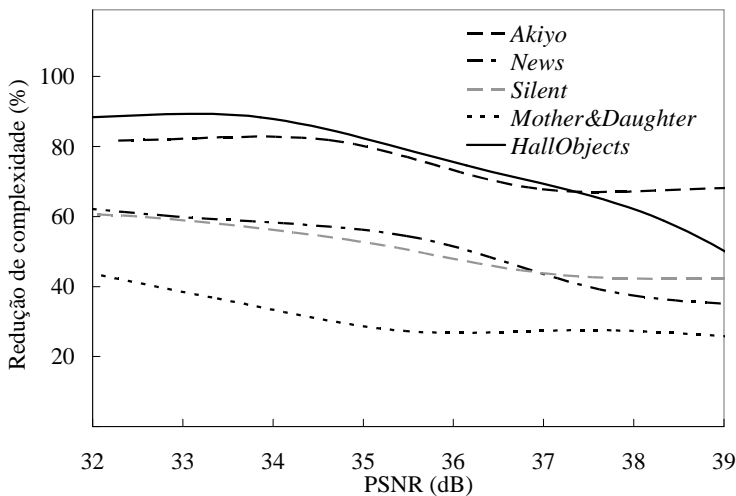


(c)

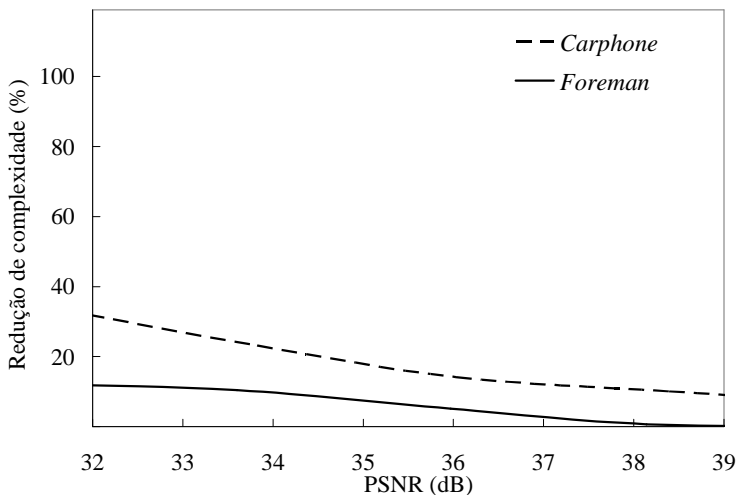


(d)

Figura 35 (cont) - Efeito do limiar de distorção na complexidade da EM. Para seqüências de vídeo (a) *Akiyo*, (b) *News*, (c) *Mother&Daughter* e (d) *Carphone*.



(a)



(b)

Figura 36 - Redução de complexidade para a estratégia RST usando a pré-seleção do modo *Skip*. (a) Vídeos adquiridos com câmera fixa. (b) Vídeos adquiridos com câmera móvel.

Tabela 16 – Percentagem de redução de complexidade do processo EM com Regra 1

Sequências	Parâmetro de quantização, $Q$			
	24	28	32	36
<i>HallObjects</i>	65,0 %	82,5 %	89,1 %	94,1 %
<i>Akiyo</i>	74,6 %	80,8 %	84,3 %	90,8 %
<i>News</i>	59,8 %	66,6 %	76,2 %	81,4 %
<i>Mother&amp;Daughter</i>	40,5 %	48,2 %	55,0 %	65,2 %
<i>Carphone</i>	32,9 %	44,4 %	54,2 %	64,7 %
<i>Foreman</i>	22,7 %	32,9 %	41,1 %	50,4 %
<i>Coastguard</i>	9,8 %	18,6 %	33,4 %	50,5 %
<b>Média</b>	<b>43,6 %</b>	<b>53,4 %</b>	<b>61,9 %</b>	<b>71,0 %</b>

Tabela 17 – PSNR e  $\Delta$ PSNR(dB) para o codificador FullRST (sem a Regra 2)

Sequências	Taxa de bits (kbps)**			
	96	64	48	24
<i>HallObjects</i>	*	*	39,3 (0,03)	34,9 (0,09)
<i>Akiyo</i>	*	*	*	37,6 (0,03)
<i>News</i>	*	37,4 (0,01)	35,4 (0)	*
<i>Mother&amp;Daughter</i>	38,4 (-0,02)	36,3 (-0,03)	34,7 (-0,01)	31,5 (0,01)
<i>Carphone</i>	39,4 (-0,01)	37,2 (0,01)	35,7 (-0,02)	31,9 (-0,06)
<i>Foreman</i>	37,1 (0,03)	34,9 (0,09)	33,1 (0,08)	*
<i>Coastguard</i>	33,1 (0,05)	31,6 (0,04)	30,7 (0,03)	*
<b><math>\Delta</math>PSNR Média</b>	<b>0,01</b>	<b>0,03</b>	<b>0,02</b>	<b>0,02</b>

\* valores não calculados

\*\* valores entre parênteses são as variações de PSNR em relação ao FS ( $\Delta$ PSNR)

#### 5.8.4 Pré-Eliminação do Modo *Intra*

O uso do custo RD inicial *Intra*  $J_1$  para pré-eliminar os modos *Intra* leva a uma redução de carga computacional em todas as sequências avaliadas. Para ilustrar esse ganho, é usada como exemplo a sequência *Foreman*, por essa ser dentre as sequências avaliadas, a mais sensível em relação à codificação *Intra* (ver Tabela 13). Nesse caso,

usando um nível de prioridade *Inter*  $k_l = 16$ , obteve-se uma redução de 20% na complexidade (ver Tabela 18) com uma perda de desempenho RD menor do que 0,1 dB (ver Figura 37). Para as outras seqüências avaliadas a redução de complexidade é similar a obtida com a seqüência *Foreman* para os níveis de prioridade  $k_l$  variando entre 8 e 32, sem perda significativa de desempenho RD. Note que, conforme ilustrado na Figura 37, a função custo RD inicial *Intra*, proposta em (61) balanceou a diferença de qualidade, permitindo controlar a perda em toda a faixa de taxa de bits.

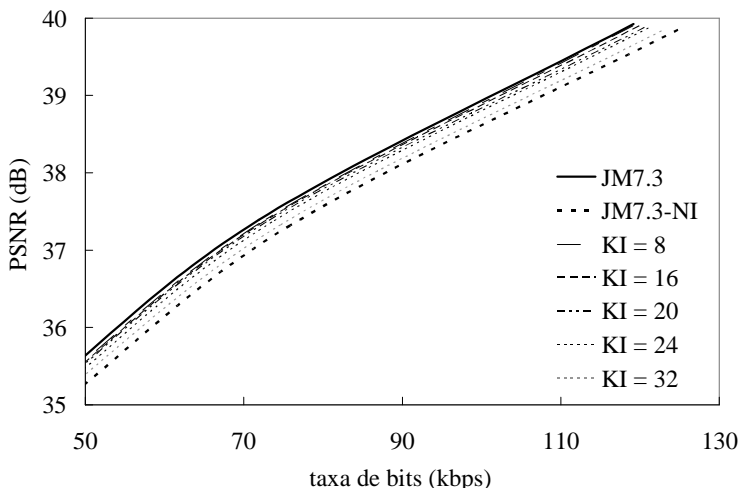


Figura 37 - Desempenho RD do codificador proposto com ST1, adotando a pré-eliminação de alguns modos *Intra*, para a seqüência *Foreman*.

Combinando a pré-eliminação dos modos *Intra* com a pré-eliminação das predições de crominância, uma redução adicional de complexidade é obtida sem causar uma mudança significativa no desempenho RD. Usando um nível de prioridade *Inter*  $k_l = 16$ , uma redução de até 26% no tempo de processamento é obtida (ver Tabela 19) com mínima mudança no desempenho RD para a seqüência *Foreman* (ver Figura 38). Resultados similares são obtidos para as outras seqüências de vídeo avaliadas.



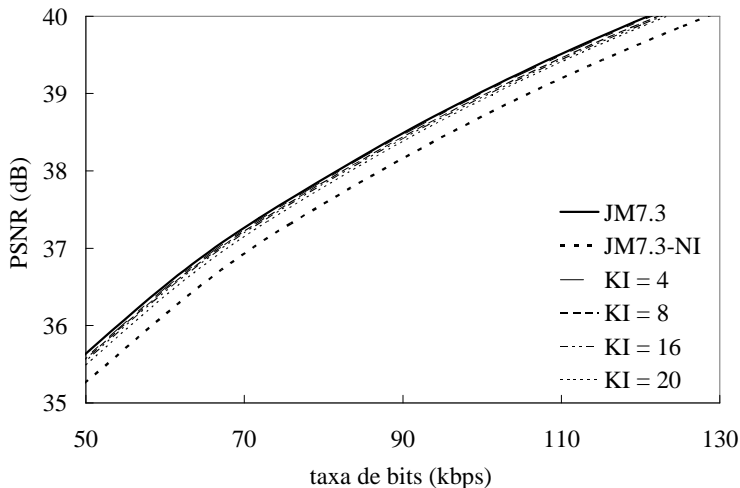


Figura 38 - Desempenho RD do codificador proposto com ST1, adotando a pré-eliminação de alguns modos *Intra* e das previsões de crominância, para a sequência *Foreman*.

Tabela 18 – Percentagem média de redução na complexidade total usando a pré-eliminação do modo *Intra* para a sequência *Foreman* com taxas de bits de 50 até 130 kbps

Nível de prioridade <i>Inter</i>	Redução de complexidade
$k_1 = 8$	6,1 %
$k_1 = 16$	19,5 %
$k_1 = 20$	25,8 %
$k_1 = 24$	27,3 %
$k_1 = 32$	30,5 %
Sem modo <i>Intra</i>	31,0 %

Tabela 19 – Percentagem média de redução na complexidade total usando a pré-eliminação do modo *Intra* e a pré-eliminação da predição da crominância para a sequência *Foreman* com taxas de bits de 50 até 130 kbps.

Nível de prioridade <i>Inter</i>	Redução de complexidade
$k_1 = 4$	21,5 %
$k_1 = 8$	22,4 %
$k_1 = 16$	25,7 %
$k_1 = 20$	27,1 %
Sem modo <i>Intra</i>	31,2 %

### 5.8.5 Resultados Finais para a Abordagem RST

Para obter os resultados finais usando todas as possibilidades de redução de complexidade computacional da estratégia RST, são aplicadas simultaneamente a procura dos VMs (*Regra 1* e *Regra 2*) e seleção do modo de codificação (pré-seleção do modo *Skip*, pré-eliminação do modo *Inter* e *Intra*). Este codificador, doravante denominado FullRST, será usado para codificar sequências de vídeo com câmera fixa e não estacionária, com o modo LT5 habilitado em ambos os codificadores FS e FullRST. Para obter o melhor compromisso entre desempenho RD e complexidade computacional, o limiar de qualidade e o nível de prioridade *Inter* são ajustados para  $D_L = 200$  e  $k_l = 16$ , respectivamente.

#### a) Comparação de desempenho RD

Conforme mencionado anteriormente, a comparação de desempenho RD entre dois codificadores pode ser feita através das curvas RD. Na Figura 39 são mostradas as curvas RD obtidas com os codificadores de referência FS e o FullRST para a sequência *Mother&Daughter*. Para essa sequência as diferenças de desempenho foram mínimas. Diferenças semelhantes também ocorrem nas demais sequências avaliadas (ver Apêndice 2). Note que confrontando as curvas RD não é possível estabelecer uma comparação numérica de desempenho, pois os pontos obtidos para traçar as curvas têm taxas de bits e distorções diferentes.

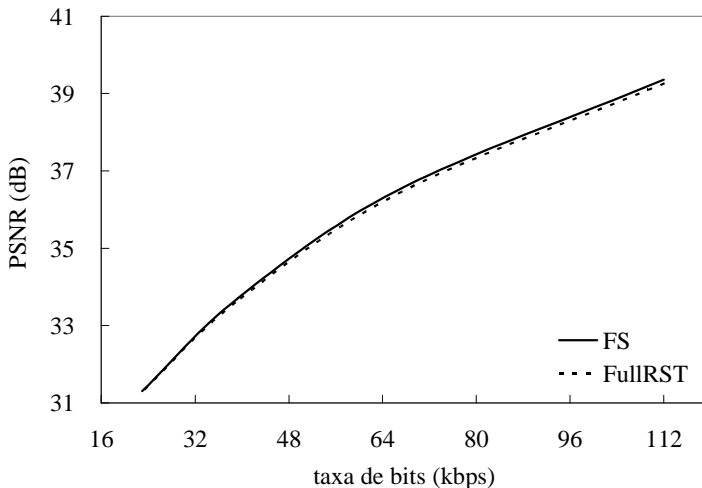


Figura 39 - Desempenho RD do codificador H.264 RDO FS e do codificador FullRST para a sequência *Mother&Daughter*.

Para solucionar tal problema, uma alternativa usada na avaliação do padrão H.264 [142] é fixar a taxa de bits e comparar a distorção obtida. O controle da taxa de bits é feito através da variação do parâmetro de quantização  $Q$ , incrementando o seu valor a partir de um quadro da sequência para obter o ajuste fino de taxa. A determinação do quadro em que ocorre a mudança do valor de  $Q$  é realizada por um processo iterativo. No presente trabalho, para evitar a abordagem iterativa, determinam-se os valores de taxa de bits ou de qualidade desejada, através do algoritmo de Neville [120], usando os valores de taxa  $R_{Q_i}$  e distorção  $D_{Q_i}$  (ver Apêndice 2), obtidos para um conjunto predefinido de parâmetros de quantização  $\{Q_1, Q_2, Q_3, Q_4\}$ . Dessa forma, fixando as taxas de bits em 24, 48, 64 e 96 kbps, e usando os pontos  $\{R_{Q_i}, D_{Q_i}\}$  da segunda e terceira coluna do Apêndice 2 de cada sequência de vídeo, são obtidos os valores de qualidade para as taxas correspondentes. De maneira análoga são obtidos os valores de taxa de bits para qualidade de 34, 36, 38 e 40 dB. Com esse procedimento é possível comparar o desempenho RD em uma dada taxa ou qualidade (situada entre os valores  $Q_1$  e  $Q_4$ ), sem recodificar iterativamente inúmeras vezes a sequência de vídeo correspondente.

Os resultados de PSNR média e a perda de desempenho  $\Delta$ PSNR (valores entre parênteses) são apresentados na Tabela 20, mostrando que a perda de desempenho é mínima (menor do que 0,1 dB) para as diferentes taxas de bits e seqüências de vídeo consideradas. Em duas seqüências (*HallObjects* e *Coastguard*) é observado um pequeno ganho (menor do que 0,1 dB), o qual creditamos ao uso do modo *Inter* no lugar do modo *Intra*, principalmente, na situação (iv) descrita na Seção 5.7.3. A Tabela 21 mostra que pode ocorrer um pequeno acréscimo na taxa de bits (menor do que 3%) quando a PSNR varia entre 34 a 40 para as mesmas seqüências de vídeo. Note que, sob algumas condições, ocorre uma pequena redução da taxa de bits (de até 3%) a qual creditamos ao mesmo motivo anteriormente aventado.

### b) Comparação de complexidade computacional

Para avaliar a redução de complexidade computacional obtida, é determinado o percentual de avaliações de VMs realizadas nos codificadores FS e FullRST (ver Figura 40). A Figura 40 mostra o percentual de blocos avaliados no modo *Inter*<sub>16x16</sub>, *Inter*<sub>8x8</sub> e *Inter*<sub>4x4</sub>. Note que a percentagem de blocos avaliados no modo *Inter*<sub>4x4</sub> é muito pequena, principalmente, quando uma qualidade menor é aceitável. Essa redução ocorre porque o custo RD inicial mínimo depende do valor de  $Q$ , tornando o processo de EM mais simples em baixas taxas de bits. Apesar da grande redução de complexidade, não existe qualquer perda no desempenho RD, uma vez que os tamanhos de bloco descartados (sem a realização da EM) são aqueles que já não podem melhorar o desempenho RD devido ao seu custo RD inicial mínimo ser maior do que o custo RD obtido com a partição em blocos já avaliados.

Para obter a estimativa de complexidade do algoritmo de EM, os valores de NPP para cada tipo de bloco  $N \times M$  são obtidos e mostrados no Apêndice 2. O valor do NPP total do MB é obtido pela soma ponderada dos  $NPP_{N \times M}$  de todos os tamanhos de bloco

$$NPP_{\text{Total}} = \sum_{N \times M \in \mathbf{TB}} \frac{1}{NM} NPP_{N \times M} \quad (64)$$

onde o  $\mathbf{TB} = \{16 \times 16, 16 \times 8, 8 \times 16, 8 \times 8, 8 \times 4, 4 \times 8, 4 \times 4\}$  é o conjunto de todos os tamanhos de bloco em que os MBs podem ser particionados e o fator de ponderação  $1/NM$  corresponde ao recíproco do número de *pixels* no bloco.

Tabela 20 – PSNR e  $\Delta$ PSNR (dB) para o codificador FullRST (com a Regra 2)

Sequências	Taxa de bits (kbps)**			
	96	64	48	24
<i>HallObjects</i>	*	*	39,3 (0,1)	34,9 (0,09)
<i>Akiyo</i>	*	*	*	37,5 (-0,01)
<i>News</i>	*	37,3 (-0,08)	35,4 (-0,05)	*
<i>Mother&amp;Daughter</i>	38,4 (-0,1)	36,2 (-0,09)	34,7 (-0,07)	31,5 (0)
<i>Carphone</i>	39,4 (-0,08)	37,1 (-0,06)	35,6 (-0,11)	31,9 (-0,1)
<i>Foreman</i>	37 (-0,02)	34,8 (0)	33 (0)	*
<i>Coastguard</i>	33,1 (0,02)	31,6 (0,03)	30,6 (0,01)	*
<b><math>\Delta</math>PSNR Média</b>	<b>-0,04</b>	<b>-0,04</b>	<b>-0,02</b>	<b>0,00</b>

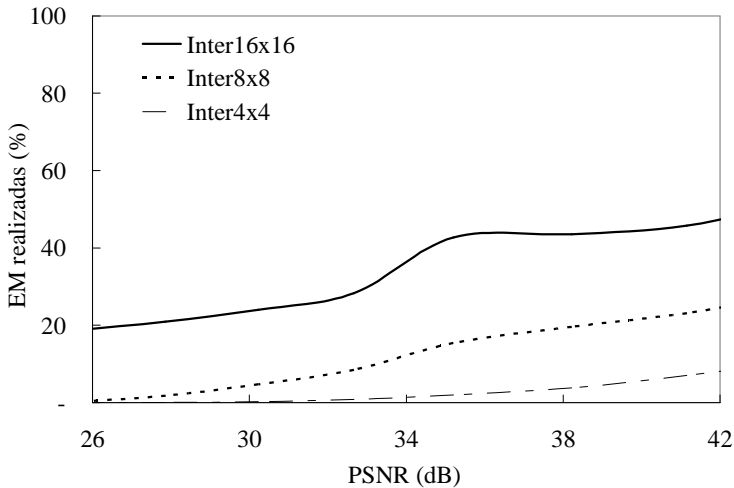
\* valores não calculados

\*\* valores entre parênteses são as variações de PSNR em relação ao FS ( $\Delta$ PSNR)Tabela 21 – Taxa de bits (kbps) e  $\Delta$ taxa de bits (%) para o codificador FullRST

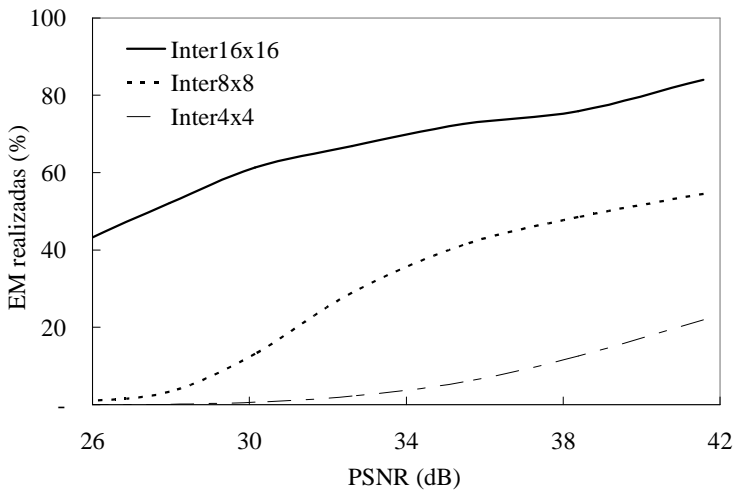
Sequências	PSNR (dB)**			
	34	36	38	40
<i>HallObjects</i>	21,8 (-1,33 %)	27,6 (-1,27 %)	37,5 (-1,75 %)	55,1 (-3,12 %)
<i>Akiyo</i>	*	19,1 (-0,67 %)	25,7 (0,77 %)	35,2 (1,57 %)
<i>News</i>	39,3 (0,99 %)	52,7 (0,78 %)	70,7 (1,38 %)	94,6 (2,59 %)
<i>Mother&amp;Daughter</i>	42,2 (1,19 %)	61,7 (1,67 %)	89,2 (1,82 %)	127,7 (1,75 %)
<i>Carphone</i>	35,5 (2,7 %)	51,8 (1,44 %)	74,4 (1,12 %)	104,4 (2,13 %)
<i>Foreman</i>	56,3 (-0,05 %)	79,6 (0,08 %)	112,5 (0,45 %)	157 (0,98 %)
<i>Coastguard</i>	127,6 (0,33 %)	194,3 (0,37 %)	269,9 (-0,03 %)	350,4 (-0,8 %)
<b><math>\Delta</math>taxa média de bits</b>	<b>0,64 %</b>	<b>0,34 %</b>	<b>0,54 %</b>	<b>0,73 %</b>

\* valores não calculados

\*\* valores entre parênteses são as variações da taxa de bits em relação ao FS ( $\Delta$ taxa de bits)



(a)



(b)

Figura 40 - Percentagem de estimações de movimento realizadas com tamanho de bloco  $16 \times 16$ ,  $8 \times 8$  e  $4 \times 4$  pixels para o codificador FullRST. (a) Sequência *Akiyo*. (b) Sequência *Mother&Daughter*.

A redução de complexidade da avaliação do modo *Intra* é estimada comparando o número de predições de modo *Intra* (*N-Intra*) realizadas com o codificador de referência (FS) e a estratégia RST proposta (FullRST). No FS todas as possibilidades de codificação *Intra* e *Inter* são sempre avaliadas, resultando em valores constantes de *NPP* e *N-Intra*. No FullRST esses valores variam conforme a complexidade de movimento da cena. Através dos resultados obtidos se constatou que ocorre uma efetiva redução de complexidade do processo de EM (modos *Inter*) e também do número de avaliações de modos *Intra*. O quadro resumo dos percentuais médios de redução do NPP para cada tamanho de bloco e o percentual de redução do número de modos *Intra* avaliados é apresentado na Tabela 22. Note que uma maior redução de complexidade ocorre quando é usada uma taxa de bits menor. O uso da estratégia RST no processo de EM leva a uma redução de complexidade entre 55% a 74% e a pré-eliminação do modo *Intra* na estratégia RST reduz a carga computacional de 70% a 82%.

Tabela 22 – Percentual médio de redução de complexidade da seleção do modo *Inter* e *Intra* do codificador FullRST em relação o FS com LT5

Parâmetro de quantização	Modo <i>Inter</i>							Modo <i>Intra</i>			
	16×16	16×8	8×16	8×8	8×4	4×8	4×4	Total	16×16	4×4	Total
24	37,1%	39,1%	39,4%	60,2%	65,1%	65,1%	77,6%	<b>54,8%</b>	66,2%	74,3%	<b>70,2%</b>
28	41,0%	44,1%	44,5%	63,8%	70,5%	70,5%	85,4%	<b>60,0%</b>	70,3%	77,7%	<b>74,0%</b>
32	44,9%	49,8%	49,7%	70,0%	78,9%	78,9%	92,5%	<b>66,4%</b>	74,5%	80,2%	<b>77,4%</b>
36	49,8%	57,3%	57,1%	79,7%	88,7%	88,8%	96,6%	<b>74,0%</b>	79,9%	83,7%	<b>81,8%</b>

A Tabela 23 mostra que através da combinação das duas estratégias RST para o modo *Inter* e *Intra*, o codificador FullRST reduz a complexidade em até 98% em relação ao codificador H.264 RDO FS. A redução de complexidade devido à pré-eliminação do modo *Inter* e pré-eliminação do modo *Intra* é mostrada, respectivamente, nas Tabela 24 e Tabela 25 . Note que ambas pré-eliminações resultam em uma maior redução de complexidade computacional para uma taxa de bits menor (maior  $Q$ ).

Tabela 23 – Percentagem de redução de complexidade total para o codificador FullRST

Sequências	Parâmetro de quantização, $Q$			
	24	28	32	36
<i>HallObjects</i>	95,7 %	96,7 %	97,4 %	98,0 %
<i>Akiyo</i>	95,7 %	96,2 %	96,9 %	98,1 %
<i>News</i>	93,1 %	93,0 %	95,0 %	95,3 %
<i>Mother&amp;Daughter</i>	81,8 %	84,2 %	86,2 %	89,6 %
<i>Carphone</i>	83,7 %	85,9 %	88,9 %	91,9 %
<i>Foreman</i>	77,6 %	80,6 %	84,7 %	85,5 %
<i>Coastguard</i>	68,3 %	71,5 %	77,9 %	84,7 %
<b>Média</b>	<b>85,1 %</b>	<b>86,9 %</b>	<b>89,6 %</b>	<b>91,9 %</b>

Tabela 24 – Percentagem de redução de complexidade do processo EM com o codificador FullRST

Sequências	Parâmetro de quantização, $Q$			
	24	28	32	36
<i>HallObjects</i>	87,4 %	93,3 %	94,5 %	96,2 %
<i>Akiyo</i>	84,0 %	87,0 %	89,4 %	93,2 %
<i>News</i>	75,1 %	77,3 %	81,8 %	84,4 %
<i>Mother&amp;Daughter</i>	48,3 %	54,0 %	59,4 %	68,6 %
<i>Carphone</i>	44,3 %	50,9 %	59,6 %	70,3 %
<i>Foreman</i>	29,6 %	36,3 %	45,1 %	51,7 %
<i>Coastguard</i>	15,1 %	21,1 %	34,8 %	53,6 %
<b>Média</b>	<b>54,8 %</b>	<b>60,0 %</b>	<b>66,4 %</b>	<b>74,0 %</b>



Tabela 25 – Percentagem de redução de complexidade do cálculo do modo *Intra* para o codificador FULLRST

Sequências	Parâmetro de quantização, $Q$			
	24	28	32	36
<i>HallObjects</i>	68,8 %	77,5 %	81,8 %	85,7 %
<i>Akiyo</i>	78,0 %	82,3 %	85,2 %	88,9 %
<i>News</i>	73,3 %	75,5 %	79,6 %	82,7 %
<i>Mother&amp;Daughter</i>	71,8 %	73,9 %	75,9 %	79,3 %
<i>Carphone</i>	71,4 %	74,7 %	78,4 %	83,8 %
<i>Foreman</i>	66,8 %	68,9 %	73,5 %	78,8 %
<i>Coastguard</i>	61,4 %	65,0 %	67,3 %	73,1 %
<b>Média</b>	<b>70,2 %</b>	<b>74,0 %</b>	<b>77,4 %</b>	<b>81,8 %</b>

## 5.9 Conclusões

Neste capítulo, foi apresentada uma estratégia de truncamento aplicada a codificadores de vídeo otimizados em RD. Tal abordagem visa reduzir a carga computacional de ambos os processos de estimação de movimento e seleção do modo de codificação de cada MB. A estratégia RST permite antecipar o final dos processos, acima mencionados, evitando o cálculo da taxa e da distorção para muitos modos candidatos, quando o custo RD atual é ótimo. Na fase de seleção dos modos de codificação, é feita a pré-seleção do modo *Skip*, a pré-eliminação de diversos particionamentos do MB no modo *Inter* e pré-eliminação do cálculo de previsões espaciais do modo *Intra*. Além disso, um número considerável de candidatos de VMs não precisa ser avaliado, uma vez que seu custo RD inicial mínimo é maior do que o valor ótimo previamente encontrado. A estratégia RST proposta permite reduzir a área de procura tanto na dimensão espacial quanto na dimensão temporal. Quando a *Regra 2*, (truncamento por qualidade) não é aplicada, a complexidade computacional é reduzida sem perda de desempenho RD; por outro lado, aplicando a *Regra 2*, uma economia adicional de complexidade é obtida com uma perda insignificante de PSNR.

Neste capítulo, foi também apresentado um procedimento que coteja o desempenho RD de diferentes codificadores de vídeo através da estimação das taxas de bits ou qualidade, obtidos previamente com

parâmetros de quantização prefixados. Com tal procedimento, se evitou a recodificação iterativa da sequência de vídeo que geralmente é necessária para se obter uma taxa de bits ou qualidade desejada. Esse tipo de abordagem, apesar de não fornecer um fluxo de bits que possa ser decodificado, permitiu comparar codificadores diferentes. Usando essa abordagem, mostrou-se que em relação ao FS, o codificador FullRST apresenta uma variação mínima de taxa de bits (menor do que 3%) para qualidades situadas entre 35 e 39 dB para as sequências QCIF consideradas. Da mesma forma, foi insignificante a perda ou ganho de qualidade em PSNR para taxas de bits entre 16 e 96 kbps. Assim, conclui-se que a estratégia RST é eficiente para reduzir a complexidade computacional com mínima perda em desempenho RD.

## Capítulo 6

### Algoritmos Rápidos para EM-RDO

No capítulo anterior foi apresentada a estratégia RST que reduz significativamente a complexidade computacional do processo de EM, mantendo o desempenho RD praticamente inalterado em relação ao codificador de referência H.264 FS RDO. Neste capítulo, é mostrado como obter uma maior redução de complexidade de procura usando os ARs na EM, os quais evitam a busca exaustiva em toda a janela de procura. A maioria das abordagens de EM rápida tem sido avaliada sob a condição de otimização da distorção (DO) [104]-[108]. Integramos algumas dessas abordagens ao codificador JM7.3 otimizado em RD (RDO) [91] com o objetivo de avaliar a eficiência na redução de complexidade computacional e o seu desempenho RD. Como nesses algoritmos o processo de procura dos VMs não é exaustivo, o VM selecionado nem sempre é o ótimo. Apesar disso, a redução de complexidade obtida geralmente compensa a perda em desempenho RD principalmente em situações em que o tempo de codificação é limitado, como para as aplicações de vídeo *on-line*.

Neste capítulo, o perfil típico de distribuição dos VMs obtido pelo codificador FS RDO, que apresenta uma concentração de VMs ao longo dos eixos, é usado para obtenção de um novo algoritmo rápido. O algoritmo proposto é denominado busca logarítmica em formato de diamante (LDSS – *logarithmic diamond shape search*), pois o seu processo de procura dos VMs é baseado em um gabarito de procura em formato de diamante, cujo tamanho é reduzido em uma razão logarítmica à medida que ocorre a convergência para o VM “ótimo”.

No final do capítulo o desempenho RD do uso em conjunto da estratégia RST com o algoritmo de busca LDSS proposto é avaliado, mostrando que uma solução de baixa complexidade pode obter um desempenho RD próximo ao do algoritmo de referência H.264 FS RDO.

#### 6.1 Algoritmos Rápidos para EM

A maioria das propostas de algoritmos rápidos desenvolvidas para a condição de otimização de distorção é baseada nas premissas: (i) monotonicidade da função distorção e (ii) distribuição centralizada dos

mínimos da função distorção [104]-[108]. Em sequências típicas de videoconferência, tais premissas podem ser consideradas parcialmente válidas e o VM obtido é na maioria das vezes o ótimo; contudo, em algumas situações o emprego desses ARs pode levar a seleção de um VM afastado do VM ótimo, degradando o desempenho RD em alguns blocos. O desempenho dos ARs apresenta variações que dependem do tipo de movimento apresentado na sequência de vídeo. Com o objetivo de avaliar as diferentes estratégias de AR, serão discutidos os algoritmos mais frequentemente citados na literatura: TSS [105], NTSS [106] e DSS [100] e os seus desempenhos serão comparados com aquele obtido com o algoritmo de referência H.264 FS RDO com a mesma janela de procura. Para se poder estabelecer uma comparação em condições semelhantes de complexidade computacional, foi usada uma versão do algoritmo de referência com janela de procura reduzida.

### 6.1.1 Algoritmos TSS e NTSS

O algoritmo de busca em três passos (TSS - *three step search*) [105] é uma estratégia frequentemente usada em codificadores DO devido à sua simplicidade. A base da proposta é a hipótese da monotonicidade da função distorção. Um aprimoramento do TSS é o algoritmo da nova busca em três passos (NTSS - *new three step search*) [106], que além da monotonicidade da função distorção, explora também a concentração dos VMs em torno do centro da janela de procura existente na maioria das sequências de vídeo. A concentração dos VMs em torno do centro da janela de procura ocorre devido ao movimento conjunto de diversos blocos em objetos visuais cujo contorno se estende por vários blocos, ou então, pela ausência de movimento em muitos blocos de um quadro em relação aos quadros anteriores. Essa concentração é maior quando o centro da janela de procura é o VM predito (ver Seção 5.2).

Nas estratégias TSS e NTSS a procura é feita em passos com uma janela de procura de  $\pm 8$  pixels. O gabarito de procura usado é um quadrado com 8 posições, conforme mostra a Figura 41. No primeiro passo são avaliadas as posições correspondentes ao centro da janela de procura (posição marcada com “0”) e as definidas pelo gabarito que têm dimensão horizontal e vertical igual à metade da janela de procura (posições marcadas com “1”). No NTSS são também avaliadas as 8 posições vizinhas ao centro da janela de procura, sendo usada uma regra de truncamento que interrompe a EM no primeiro passo se a posição central é a melhor posição dentre as 17 posições avaliadas. Se uma posição vizinha a central é a melhor opção, então as posições vizinhas

dessa posição são avaliadas e a melhor posição é escolhida [ver Figura 41 (b)]. Caso qualquer outra posição do gabarito seja a escolhida, o centro de procura é deslocado para essa posição e o tamanho do gabarito é reduzido à metade. Nesse caso, o algoritmo só para quando as posições do gabarito são vizinhas ao centro de procura.

Na estratégia TSS, a complexidade computacional do algoritmo é dada pelo número de pontos de procura

$$NPP_{TSS} = 1 + 8 \log_2 w \quad (65)$$

que depende apenas da janela de procura  $w$ . Assim, para uma janela de  $\pm 8$  pixels é obtido  $NPP_{TSS} = 25$ .

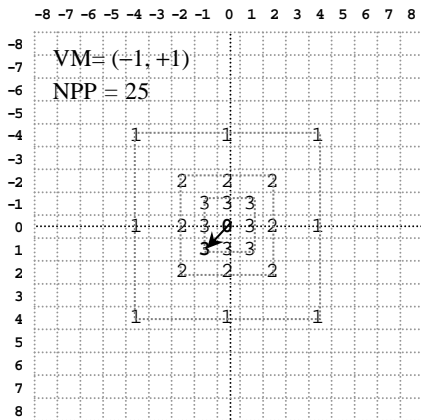
Na estratégia NTSS o número de pontos de procura depende do VM escolhido no primeiro passo e da janela de procura usada. Assim, se o centro da janela de procura for escolhido,  $NPP_{NTSS} = 17$ ; se uma posição vizinha ao centro é escolhida no primeiro passo, então apenas outras 3 (ou 5) posições são avaliadas; caso contrário

$$NPP_{NTSS} = 1 + 8 (\log_2 w + 1) \quad (66)$$

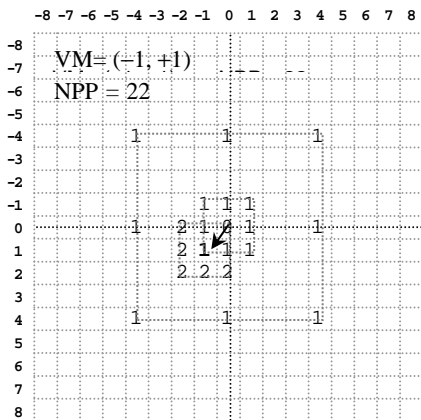
que depende apenas da janela de procura  $w$ . Assim, para uma janela de  $\pm 8$  pixels, é obtido  $NPP_{NTSS} = 33$ .

Na Figura 41, são mostrados exemplos da seleção de dois VMs pelas estratégias TSS e NTSS. Na Figura 41 (a) e (b), o VM selecionado é próximo ao centro da janela de procura levando a uma menor complexidade computacional com o algoritmo NTSS. Nas Figura 41 (c) e (d), o VM selecionado está afastado do centro da janela de procura levando a uma menor complexidade com o algoritmo TSS. Em seqüências típicas de vídeo, devido à distribuição centralizada dos mínimos da função distorção, o NTSS geralmente resulta em menor complexidade computacional do que o TSS [106].

Em codificadores que adotam o FS na etapa de resolução de *pixel* inteiro, é frequentemente usada uma abordagem baseada no TSS para refinamento do VM para resolução de *subpixel*. Nessa situação o desempenho não é afetado significativamente pela estratégia TSS uma vez que a função distorção é geralmente monotônica em torno do mínimo.



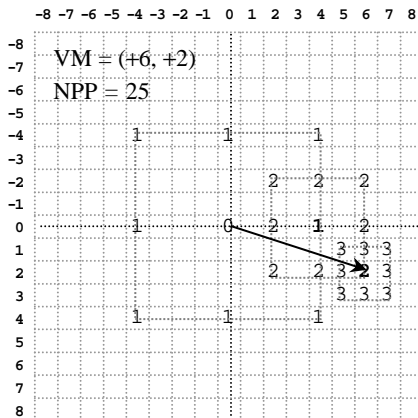
(a)



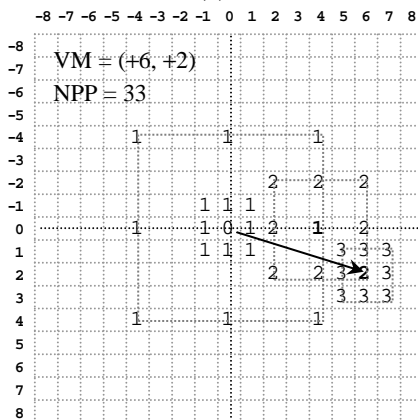
(b)

Figura 41 - Exemplo ilustrando as posições avaliadas<sup>XIII</sup> para uma janela de procura de  $\pm 8$  pixels e resolução 1 pixel. (a) e (c) Algoritmo TSS. (b) e (d) Algoritmo NTSS.

<sup>XIII</sup> Os números à esquerda e acima do diagrama são o valor do componente horizontal e vertical dos VMs. A posição central (marcada com "0") corresponde ao VM nulo e as posições marcadas com "1" correspondem aos VMs avaliados no primeiro passo.



(c)



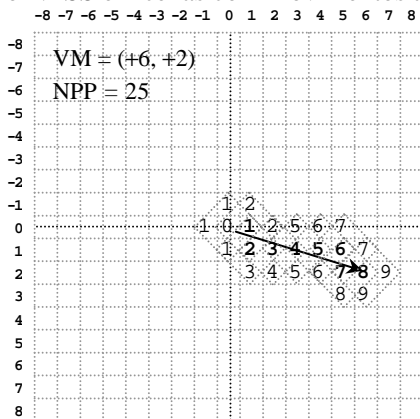
(d)

Figura 41 (cont)- Exemplo ilustrando as posições avaliadas para uma janela de procura de  $\pm 8$  pixels e resolução 1 pixel. (a) e (c) Algoritmo TSS. (b) e (d) Algoritmo NTSS.

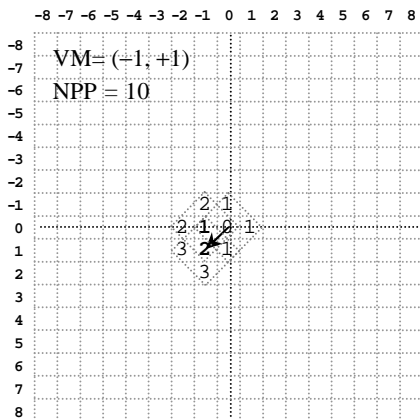
### 6.1.2 Algoritmo DSS

A estratégia de procura em formato de diamante (DSS – *diamond shape search*) [100] foi adotada pelo ITU-T a partir do modelo de teste TMN-8 [101]. Ela também é baseada na centralização dos VMs e na monotonicidade da função distorção, mas devido ao seu gabarito com menos posições de procura, resulta em uma maior redução de complexidade da EM. O gabarito de procura usado tem formato de

diamante. O melhor VM dentro as posições avaliadas se torna o novo centro de procura e um novo passo de procura é efetuado usando o mesmo gabarito deslocado. A busca termina quando o centro de procura é o VM de melhor desempenho. Em virtude de se usar apenas quatro posições no gabarito de procura, essa estratégia fornece um excelente desempenho para cenas com pouco movimento, mas perde em desempenho para o NTSS em cenas com movimentos amplos.



(a)



(b)

Figura 42 - Exemplos ilustrando as posições avaliadas com o algoritmo DSS para uma janela de procura de  $\pm 8$  pixels e resolução de 1 pixel. (a)  $VM = (+6, +2)$ . (b)  $VM = (-1, +1)$ .



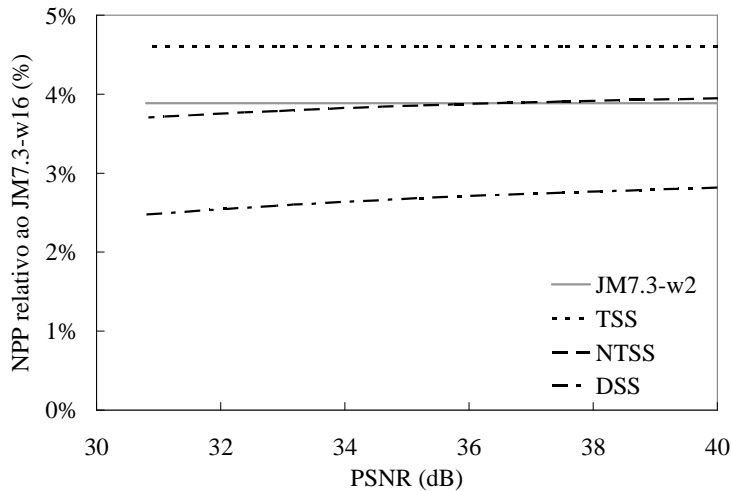
A Figura 42 apresenta dois exemplos que mostram as posições avaliadas pela estratégia DSS. No caso (a) o VM selecionado está distante da posição inicial de procura, já no caso (b), o VM selecionado está próximo ao centro da janela de procura. No DSS a complexidade da EM depende do caminho de busca seguido e do tamanho do VM selecionado. A principal deficiência do algoritmo DSS é ser facilmente armadilhado por mínimos locais da função de distorção, encontrando um VM final que não é o ótimo. Essa deficiência é mais acentuada quando existem na cena movimentos amplos e variados.

### 6.1.3 Avaliação dos ARs para Otimização RD

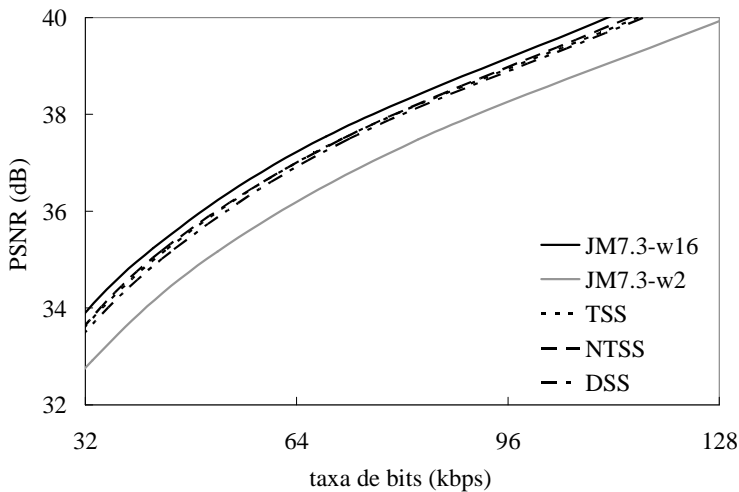
Para avaliar o desempenho dos ARs no processo de EM otimizado em RD, o custo RD conforme indicado em (25) foi usado para a seleção do melhor VM. As estratégias TSS, NTSS e DSS foram integradas no codificador JM7.3-RDO, sendo necessário acrescentar um quarto passo de procura no TSS e NTSS para adaptá-los a janela de procura de  $\pm 16$  pixels. Após determinar o VM “ótimo” com resolução de 1 pixel, são realizados refinamentos de 1/2 e 1/4 de pixel para a obtenção do VM com resolução de 1/4 de pixel. Na fase de refinamento, o critério RD também é usado na escolha dos VMs. O VM predito foi considerado o centro inicial de procura e foi usada uma memória de longo prazo com 5 referências temporais. O modo *Skip* e todas as possibilidades de modos *Inter* e *Intra* foram habilitados no codificador. A configuração IPPP foi usada na codificação dos quadros e foram avaliados os 100 quadros iniciais das sequências: *Akiyo*, *News*, *Silent*, *Carphone Mother&Daughter* e *Foreman* em formato QCIF-10Hz e das sequências *Paris* e *Mobile&Calendar* em formato CIF-10Hz.

#### a) Desempenho médio RD e complexidade computacional

O desempenho dos ARs para janela de procura de  $\pm 16$  pixels foi comparado com o codificador de referência H.264 FS RDO considerando a mesma janela de procura (JM7.3-w16) e também com janela reduzida de  $\pm 2$  pixels (JM7.3-w2). Durante a EM a função de distorção usada foi a SATD (com transformada Hadamard  $4 \times 4$  conforme descrito em [168]). Na seleção do modo de codificação a função de distorção usada foi SSD, a qual também foi usada na determinação das curvas RD. No quesito complexidade computacional, foi usada como medida de avaliação, o NPP determinado segundo (64).



(a)



(b)

Figura 43 - Desempenho RD e complexidade computacional das estratégias FS (JM7.3-w16), TSS, NTSS e DSS, com janela de procura de  $\pm 16$  pixels e FS com janela de procura de  $\pm 2$  pixels (JM7.3-w2). (a) Percentual de NPP relativo ao JM7.3-w16 para a sequência *Foreman*. (b) Curvas RD.

A Figura 43(a) ilustra a redução de complexidade obtida no processo de EM com os ARs. Note que a redução é da ordem de 20 a 45 vezes em relação ao JM7.3-w16. Essa redução de complexidade é similar a obtida com o JM7.3-w2, o qual resulta em sequências com movimentos mais amplos em perda significativa de desempenho RD [ver Figura 43(b)]. A diferença de desempenho RD entre os ARs e o JM7.3-w16 é tanto maior quanto mais amplo forem os movimentos na cena. Nas sequências analisadas a maior perda de desempenho ocorreu na sequência *Foreman*; nas sequências *News*, *Paris* e *Silent* a perda foi baixa (<0,1 dB) e em *Akiyo* e *Mother&Daughter* a perda foi mínima.

### b) Desempenho quadro-a-quadro

A comparação de algoritmos através de valores médios mostrada na subseção anterior, não permite verificar a diferença de desempenho ao longo dos quadros e determinar os quadros com maior perda. Para avaliar os ARs ao longo das sequências medimos o seu desempenho RD utilizando o procedimento descrito na Seção 4.2.

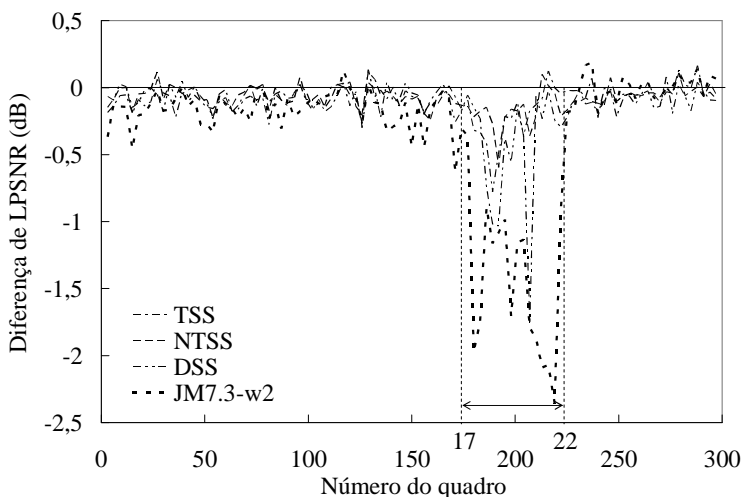


Figura 44 - Perda de desempenho RD medida em LPSNR das estratégias TSS, NTSS e DSS com relação ao JM7.3-RDO, para a sequência *Foreman* com  $Q = 32$ .

A Figura 44 mostra a variação da LPSNR relativa ao codificador JM7.3-w16 para a sequência *Foreman* com  $Q = 32$ . Note que na maioria dos quadros o desempenho de todos os ARs é similar ao algoritmo de referência; porém, entre os quadros #177 e #225 (trecho

com ampla movimentação de câmera) ocorre uma grande redução no desempenho RD dos ARs, principalmente, no DSS e no FS com janela reduzida. Para valores de  $Q = 24, 28$  e  $40$  é observado o mesmo comportamento de desempenho dentro da sequência, indicando que o trecho assinalado é o principal causador da diferença de desempenho dentre os ARs.

*c) Influência do modo Intra no desempenho RD*

Avaliando a perda de desempenho dos ARs nas sequências com movimentos amplos (*Silent, Carphone, Foreman*), é possível notar que ela decorre da escolha de VMs associados a mínimos locais. Isso acontece porque o VM escolhido para o MB pode resultar em um custo RD total superior ao obtido com sua codificação no modo *Intra*. Consequentemente, o modo *Inter* é preterido na fase de escolha do modo de codificação, mesmo que o MB possa ser adequadamente predito por compensação de movimento.

A redução de desempenho dos ARs, geralmente, está relacionada diretamente com o número de MB que são codificados no modo *Intra*. Para explicitar tal relação, na Tabela 26 é indicada a percentagem de MB codificados no modo *Intra* com cada tipo de estratégia. Note que para a sequência *Foreman* ocorre um aumento na percentagem de MBs *Intra* de 8,4% (para o algoritmo JM7.3-w16) para 11,3% e 19,8% quando a estratégia DSS e o algoritmo JM7.3-w2 são usados respectivamente. Note que o aumento nessa percentagem é menor quando a estratégia NTSS é usada (9,5%). Ou seja, o desempenho RD dos algoritmos rápidos piora com o aumento do número de MB-*Intra*, assim o NTSS que apresenta o melhor desempenho RD também tem o menor aumento no número de MB-*Intra*.

Na Figura 45 é mostrada a variação quadro-a-quadro do número de MBs *Intra*. A forte correlação existente entre essa perda de desempenho e o aumento da codificação de MBs no modo *Intra* é confirmada comparando o aumento de MBs *Intra* com a perda de desempenho RD (Figura 44).

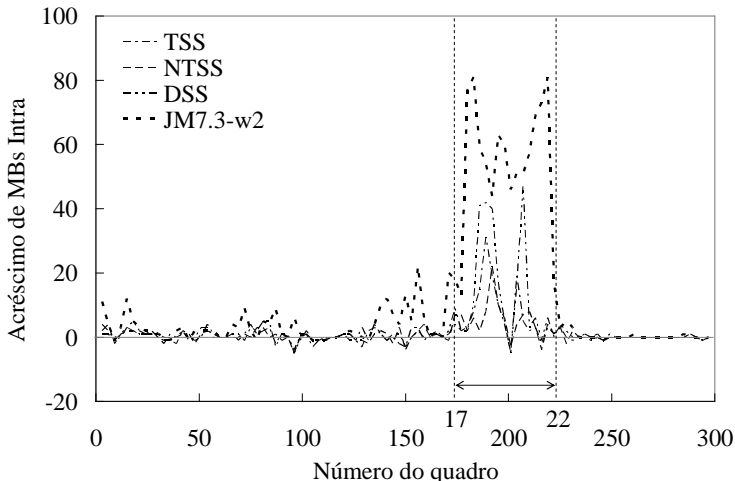


Figura 45 - Variação do número de MBs *Intra* para as estratégias TSS, NTSS e DSS em relação ao JM7.3-RDO, para a sequência *Foreman*-10Hz com  $Q = 32$ .

Tabela 26 – Percentagem de MBs codificados no modo *Intra* com JM7.3-w16, JM7.3-w2, e com os ARs TSS, NTSS e DSS para  $Q = 32$ , janela de procura de  $\pm 16$  *pixels* e resolução de  $1/4$  *pixel*

Sequências	Algoritmos				
	JM7.3-w16	JM7.3-w2	TSS	NTSS	DSS
<i>Foreman</i>	8,35%	19,84%	9,85%	9,50%	11,27%
<i>Mother&amp;Daughter</i>	0,80%	1,13%	0,90%	0,87%	1,04%
<i>Akiyo</i>	0,00%	0,01%	0,00%	0,00%	0,00%
<i>News</i>	2,40%	3,23%	2,84%	2,78%	2,78%
<i>Silent</i>	3,94%	6,29%	4,75%	4,68%	5,00%
<i>Paris</i>	2,00%	3,37%	2,33%	2,33%	2,59%
<i>Carphone</i>	6,25%	7,52%	6,39%	6,60%	6,64%
<b>Diferença média</b>		<b>2,52%</b>	<b>0,47%</b>	<b>0,43%</b>	<b>0,80%</b>

d) *Fatores que determinam o desempenho e complexidade*

Pela análise de desempenho médio e também o desempenho quadro-a-quadro, o melhor AR em todas as sequências analisadas foi o

NTSS, enquanto os piores foram o TSS e DSS. A redução de desempenho RD da estratégia DSS em relação à TSS e NTSS pode ser atribuída tanto ao gabarito de procura inicial (GPI) usado quanto ao gabarito de procura (GP) usado. No caso do DSS, o uso de GPI e GP pequenos facilitam o armadilhamento do processo de EM nos mínimos locais da função RD situados próximos ao centro da janela de procura (ver Figura 46). As estratégias TSS e NTSS usam GPIs e GPs com posições mais afastadas e, por isso, resultam em um melhor desempenho RD. O GPI usado em cada AR está representado na Figura 47, juntamente com o número inicial de posições de procura (NiPP), que é um fator determinante na complexidade computacional dos ARs. Note que dentre os ARs analisados, o DSS é o que apresenta a menor complexidade e também possui o menor NiPP.

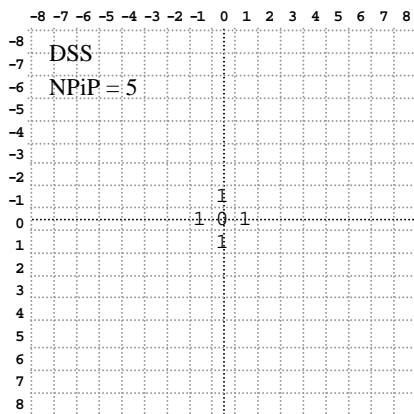
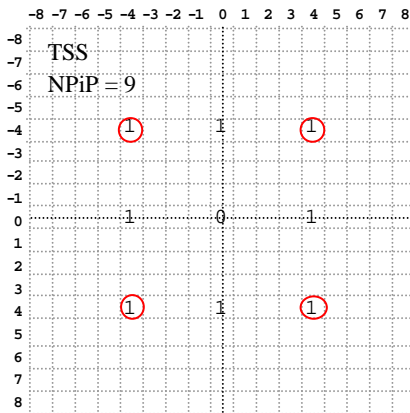
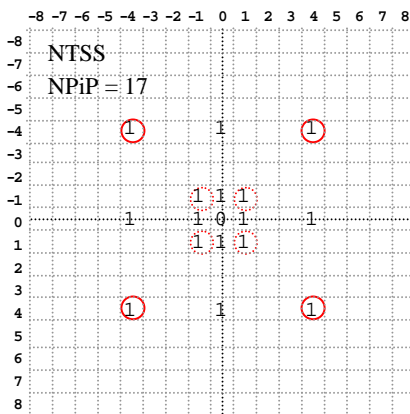


Figura 46 - Gabarito de procura inicial da estratégia DSS.



(a)



(b)

Figura 47 - Gabarito de procura inicial das estratégias: (a) TSS. (b) NTSS.

### 6.2 LDSS: Algoritmo Rápido Otimizado em RD

Na Seção 6.1, foi constatado que o emprego de ARs, tais como o TSS, NTSS e DSS reduzem em até 45 vezes a complexidade do processo de EM. No entanto, o uso desses ARs resulta frequentemente na escolha do modo *Intra* para a codificação de MBs para os quais existem VMs que produzem um custo RD muito inferior. Isso é decorrente da forma como a busca não exaustiva pelo melhor VM é

feita. A seleção do modo *Intra* no lugar de um modo *Inter* em alguns MBs é um dos principais fatores de perda de desempenho RD dos ARs em relação ao JM7.3-RDO.

Analisando também os ARs citados em termos de adaptação de seu GPI ao perfil típico de distribuição dos VMs, é observado que eles são adaptados à distribuição obtida com a otimização da distorção [Figura 29 (a) e (b)], mas não àquela obtida com a otimização em RD [Figura 29 (c) e (d)]. Note que o GPI em formato de diamante (usado no DSS) se ajusta melhor a essa distribuição do que o GPI em formato quadrado (usado no TSS e NTSS). Nesse caso, existem posições situadas fora dos eixos (marcados com um círculo na Figura 47) para as quais a probabilidade de seleção é muito baixa o que leva a um desperdício de recurso computacional. Por outro lado, o algoritmo 2-DLS, que também utiliza um GPI em formato de diamante, realiza a procura inicial nas 4 posições localizadas sobre os eixos correspondentes à metade da janela de procura. Tais posições têm probabilidade de escolha muito menor que o GPI do algoritmo DSS. Para superar tais deficiências, é proposto neste trabalho um AR que tem o GPI adaptado ao perfil típico de distribuição dos VMs de um codificador JM7.3-RDO. O AR proposto tem os GPs e os GPIs assemelhados à forma de diamante com seu tamanho definido em escala logarítmica, por isso, é denominado de AR com procura logarítmica em formato de diamante (*logarithmic diamond shape search* – LDSS) [119], diferenciando-se do algoritmo DSS que tem um GP de tamanho fixo e do algoritmo 2-DLS que utiliza um GPI fixo. A grande vantagem do algoritmo LDSS, em relação a seus concorrentes, é a sua menor susceptibilidade ao armadilhamento pelos mínimos locais; visto que, ele antecipa a avaliação dos VMs de menor custo (com maior probabilidade de ocorrência) e posterga as posições de baixa probabilidade de ocorrência.

### 6.2.1 Gabaritos de Procura Inicial

O algoritmo LDSS usa um conjunto diversificado de GPIs, todos eles com posições de procura inicial localizadas sobre os eixos principais. Essas posições são indicadas por índices descritos como  $2^\alpha$ , para  $\alpha = 0, 1, 2, \dots$ , os quais correspondem à distância em *pixel* até o centro da janela de procura, por exemplo: um índice 2 no GPI indica que ele possui 4 posições localizadas sobre os eixos principais e afastadas de 2 *pixels* do centro  $[(0, +2), (0, -2), (+2, 0), (-2, 0)]$ , da mesma forma um índice 4 indica as posições  $[(0, +4), (0, -4), (+4, 0), (-4, 0)]$ . No LDSS, o



uso de diferentes GPIs possibilita uma adaptação da estratégia de EM ao conteúdo da cena e à capacidade computacional disponível. Em regiões da imagem com baixa quantidade de movimento se pode adotar GPIs com índices baixos (exemplos: LDSS<sub>1-2</sub> e LDSS<sub>1-4</sub>, ver Figura 48); enquanto, em regiões com maior quantidade de movimento se pode usar GPIs com índices maiores (exemplos: LDSS<sub>1-8</sub> e LDSS<sub>1-2-4-8</sub>, ver Figura 48). A predição da quantidade de movimento pode ser feita a partir dos quadros e blocos anteriores. De acordo com a janela de procura também se pode incluir no GPI os índices 16, 32 e 64, ampliando sua abrangência e evitando, assim, que o algoritmo seja logrado por mínimos locais [119].

O valor NiPP, que corresponde ao número mínimo de posições de procura do algoritmo é o fator de maior influência na complexidade computacional. Comparando o GPI do NTSS [Figura 47(b)] com o do LDSS<sub>1-2-4-8</sub> [Figura 48(d)] se percebe que apesar de possuírem o mesmo NiPP, o LDSS deve resultar em melhor desempenho por ter as posições de procura adaptadas a distribuição típica dos VMs de um codificador JM7.3-RDO.

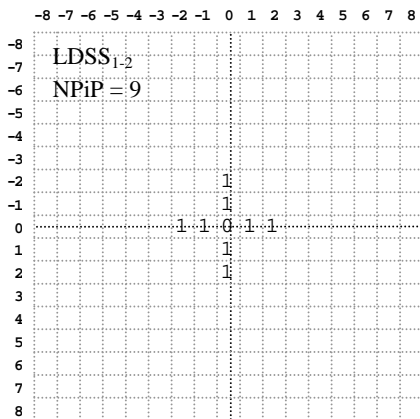
### 6.2.2 Procedimento de Busca

O algoritmo LDSS, assim como os demais ARs avaliados, usa o vetor predito como posição central da janela de procura. Na primeira fase da procura são avaliadas todas as posições do GPI. Se a posição central da janela de procura é selecionada nessa fase e o GPI inclui o índice 1, então o processo de EM com resolução de 1 *pixel* é finalizado. Caso contrário, a posição selecionada é usada como centro do GP, cujo tamanho é determinado pela distância  $d$  da posição selecionada ao centro da janela de procura. O GP em formato de diamante é denotado GP <sub>$n$</sub> , onde o índice  $n$  é dado por  $n = \log_2(d)$ , indicando o número de reduções que serão aplicadas ao gabarito até a finalização do processo de procura. A Figura 49 ilustra alguns exemplos de GPs do LDSS.

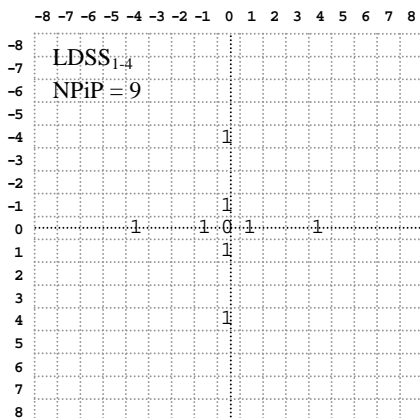
Na segunda fase, as posições definidas pelo GP são avaliadas. Se uma das posições do GP for selecionada, então o gabarito é deslocado para essa nova posição e a procura é repetida, mantendo o mesmo tamanho de gabarito. Se a posição central do gabarito é selecionada, o GP é reduzido à metade, fazendo  $n = n - 1$  e uma nova fase de procura é iniciada. O processo de EM prossegue até que o índice do GP seja zero e a posição central do GP é selecionada, conforme ilustrado na Figura 49.

Concluída a EM com resolução de 1 *pixel* é aplicada a mesma forma de refinamento (1/2 e 1/4 de *pixel*) usada nos outros ARs.

Os processos de busca da estratégia LDSS com resolução de 1 *pixel* para dois VMs final e dois GPIs diferentes GPIs são ilustrados na Figura 49. A velocidade do processo de EM depende tanto do GPI adotado quanto do VM final escolhido, de modo que GPIs com menor número de posições geralmente aumentam a velocidade da EM; porém, tendem a piorar o desempenho RD. Os GPIs com índices maiores têm menor tendência de ser armadilhados por mínimos locais.

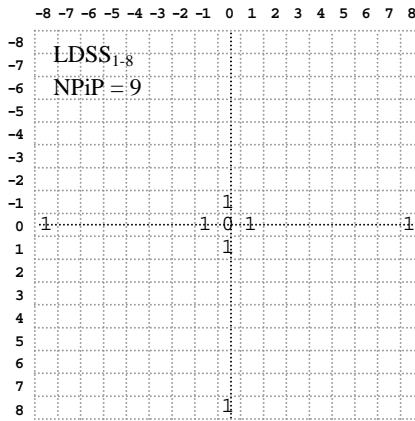


(a)

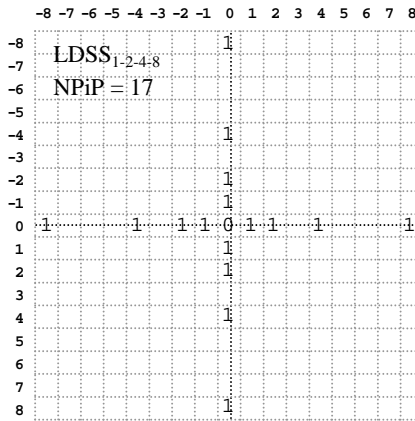


(b)

Figura 48 - Gabaritos de procura inicial da estratégia LDSS com 9 posições (a) LDSS<sub>1-2</sub> – GPI, (b) LDSS<sub>1-4</sub> – GPI, (c) LDSS<sub>1-8</sub> – GPI e com 17 posições (d) LDSS<sub>1-2-4-8</sub> – GPI .

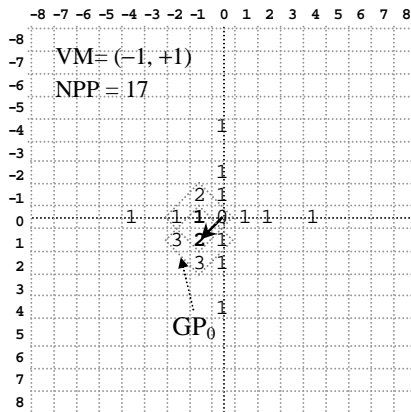


(c)

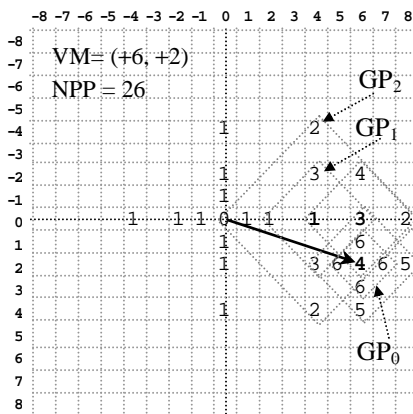


(d)

Figura 48 (cont) - Gabaritos de procura inicial da estratégia LDSS com 9 posições (a) LDSS<sub>1-2</sub> – GPI, (b) LDSS<sub>1-4</sub> – GPI, (c) LDSS<sub>1-8</sub> – GPI e com 17 posições (d) LDSS<sub>1-2-4-8</sub> – GPI .



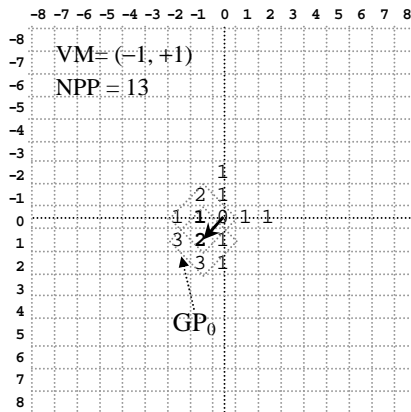
(a)



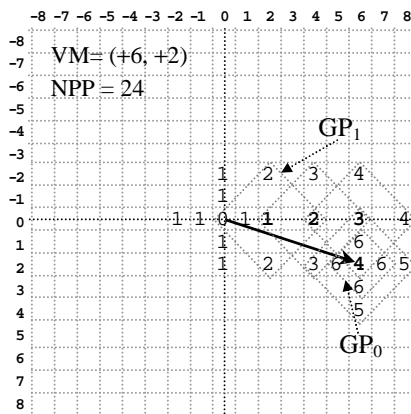
(b)

Figura 49 - Ilustração de posições com resolução 1 *pixel*, avaliadas pelo algoritmo LDSS para uma janela de procura de  $\pm 16$  *pixels*, com diversos GPis.<sup>XIV</sup> (a) e (b) LDSS<sub>1-2-4</sub>. (c) e (d) LDSS<sub>1-2</sub>.

<sup>XIV</sup> Os números no interior do diagrama indicam as posições avaliadas a cada passo de procura. As posições selecionadas como centro do GP a cada passo estão indicadas em negro.



(c)



(d)

Figura 49 (cont)- Ilustração de posições com resolução 1 *pixel*, avaliadas pelo algoritmo LDSS para uma janela de procura de  $\pm 16$  *pixels*, com diversos GPIs.<sup>XV</sup> (a) e (b) LDSS<sub>1-2-4</sub>. (c) e (d) LDSS<sub>1-2</sub>.

<sup>XV</sup> Os números no interior do diagrama indicam as posições avaliadas a cada passo de procura. As posições seleccionadas como centro do GP a cada passo estão indicadas em negro.

### 6.2.3 Resultados

Para avaliar a estratégia proposta, ela foi integrada ao codificador JM7.3-RDO, e comparada com o algoritmo de referência JM7.3-w16 e JM7.3-w2. Nas avaliações foram usadas as sequências de vídeo listadas no Apêndice 1.

#### a) *Desempenho RD e complexidade do LDSS*

Nestes testes foram realizadas simulações empregando a estratégia LDSS sob as mesmas condições de avaliação e sequências de vídeos consideradas na avaliação dos demais ARs da Seção 6.1. A influência dos diversos GPIs da estratégia LDSS foi avaliada, tanto em desempenho quanto em complexidade. Nesta avaliação o algoritmo LDSS apresentou um desempenho RD superior ao da estratégia DSS e equivalente aos dos demais algoritmos rápidos aqui considerados. Para as sequências de vídeo *Akiyo*, *News*, *Mother&Daughter*, *Silent* e *Paris*, a perda de desempenho RD observada, relativa ao JM7.3-w16 foi insignificante. A Figura 50 mostra os resultados obtidos para a sequência *Foreman*, na qual se pode notar que o desempenho RD do algoritmo LDSS é um pouco superior ao DSS em toda a faixa de taxa de bits considerada. Para as outras sequências avaliadas, o desempenho RD do algoritmo LDSS é semelhante ao observado para o NTSS e muito próximo ao JM7.3-w16.

A redução de complexidade computacional da estratégia LDSS em relação ao JM7.3-w16 está associado diretamente ao GPI utilizado. Conforme mostra a Figura 51 essa complexidade pode ser ajustada entre a do algoritmo NTSS (máxima) e a do DSS (mínima). Dentre os GPIs de menor complexidade, o melhor desempenho RD é obtido com o LDSS<sub>1-8</sub>, o qual é mais adaptado para cenas com movimentos amplos. Para as demais sequências avaliadas, os resultados de complexidade foram similares aos mostrados na Figura 51 para a sequência *Foreman*.

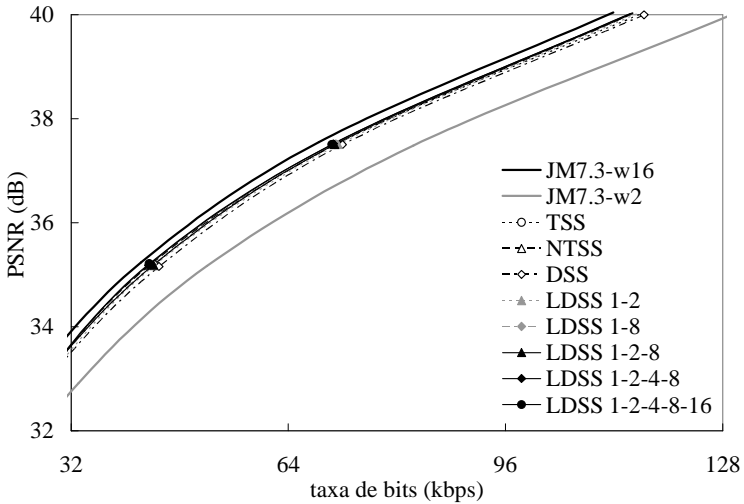


Figura 50 - Desempenho RD do algoritmo FS (JM7.3-w16 e JM7.3-w2) e TSS, NTSS, DSS e LDSS com janela de procura de  $\pm 16$  pixels, para a sequência *Foreman*.

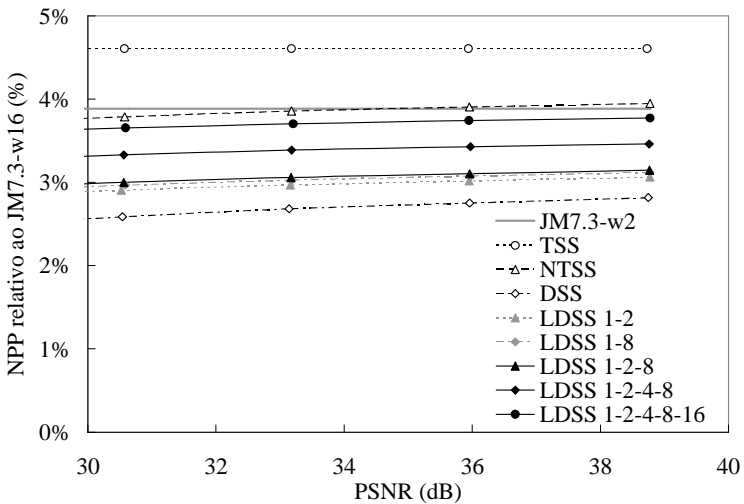


Figura 51 - Redução de complexidade computacional dos algoritmos JM7.3-w2, TSS, NTSS, DSS e LDSS em relação ao JM7.3-w16. Resultados para a sequência *Foreman* com janela de procura de  $\pm 16$  pixels.

### 6.3 Algoritmo LDSS-RST

Uma vez que as características exploradas pela proposta de AR e pela estratégia RST são diferentes, também realizamos testes para avaliar o desempenho do algoritmo LDSS-RST, o qual utiliza conjuntamente as propostas LDSS e RST apresentadas neste trabalho.

#### a) Desempenho médio RD e complexidade computacional

Os resultados obtidos para o LDSS-RST indicam que a incorporação da abordagem RST possibilita uma maior redução de complexidade da EM sem afetar o desempenho RD. Em baixas taxas de bits a complexidade do LDSS-RST chega a ser inferior a da estratégia DSS, conforme mostra a Figura 53. Note que simultaneamente com essa redução ocorre um ganho no desempenho RD em relação ao DSS (ver Figura 52), tornando o algoritmo LDSS-RST a opção de AR mais eficiente dentre os analisados, principalmente em baixas taxas de bits, quando sua complexidade computacional é inferior ao DSS.

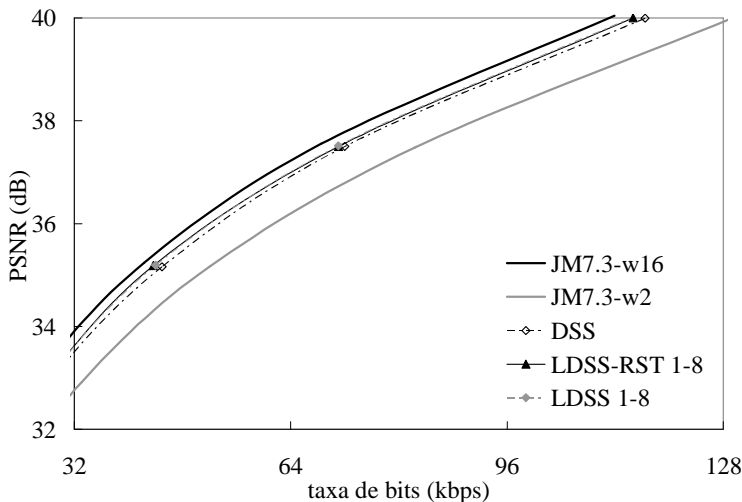


Figura 52 - Desempenho RD do algoritmo FS (JM7.3-w16 e JM7.3-w2) e dos ARs DSS, LDSS<sub>1-8</sub> e LDSS-RST<sub>1-8</sub> com janela de procura de  $\pm 16$  pixels, para a sequência *Foreman*.



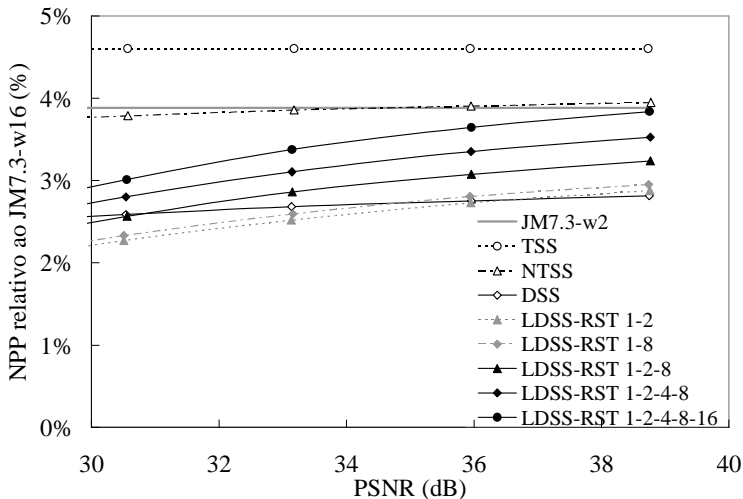


Figura 53 - Redução de complexidade computacional dos algoritmos JM7.3-w2, TSS, NTSS, DSS e LDSS-RST com relação ao JM7.3-w16. Resultados para a sequência *Foreman* com janela de procura de  $\pm 16$  pixels.

Tabela 27 – Percentagem de MB *Intra* para as estratégias FS, DSS, LDSS<sub>1,8</sub> e LDSS-RST<sub>1,8</sub>, obtidos com  $Q = 32$

Sequência	Algoritmo			
	JM7.3-w16	DSS	LDSS <sub>1,8</sub>	LDSS-RST <sub>1,8</sub>
<i>Foreman</i>	8,35%	11,27%	10,27%	9,88%
<i>Mother&amp;Daughter</i>	0,80%	1,04%	0,94%	1,02%
<i>Akiyo</i>	0,00%	0,00%	0,00%	0,00%
<i>News</i>	2,40%	2,78%	2,80%	2,83%
<i>Silent</i>	3,94%	5,00%	4,68%	4,63%
<i>Paris</i>	2,00%	2,59%	2,33%	2,42%
<i>Carphone</i>	6,25%	6,64%	6,21%	6,41%
<b>Diferença média</b>		<b>0,80%</b>	<b>0,50%</b>	<b>0,49%</b>

#### b) Número de MBs *Intra*

Em relação à percentagem de MB *Intra* do algoritmo LDSS<sub>1,8</sub>, a Tabela 27 mostra que tal percentagem é menor do que a do algoritmo

DSS, sendo este um dos fatores que contribui na melhoria do desempenho RD. O uso da abordagem RST combinada com o algoritmo LDSS afeta minimamente o desempenho RD (ver Figura 52), resultado que também aparece na percentagem de MBs *Intra* que permanece praticamente inalterado (ver Tabela 27).

c) *Desempenho quadro-a-quadro*

Usando a mesma metodologia de avaliação dos outros ARs testados, é realizada a comparação de desempenho RD quadro-a-quadro entre o algoritmo LDSS-RST e os demais algoritmos. A Figura 54 mostra que no trecho mais crítico da sequência *Foreman* (entre os quadros 177 e 225), a estratégia LDSS-RST melhora significativamente o desempenho RD, comparado com o do DSS (seu concorrente em termos de complexidade computacional). Com respeito ao DSS, o algoritmo LDSS-RST<sub>1-8</sub> apresenta um ganho de desempenho RD em torno 1,23dB nas proximidades do quadro #207. Em relação aos ARs, TSS e NTSS, a proposta LDSS-RST exibe um desempenho similar, porém com uma complexidade consideravelmente inferior.

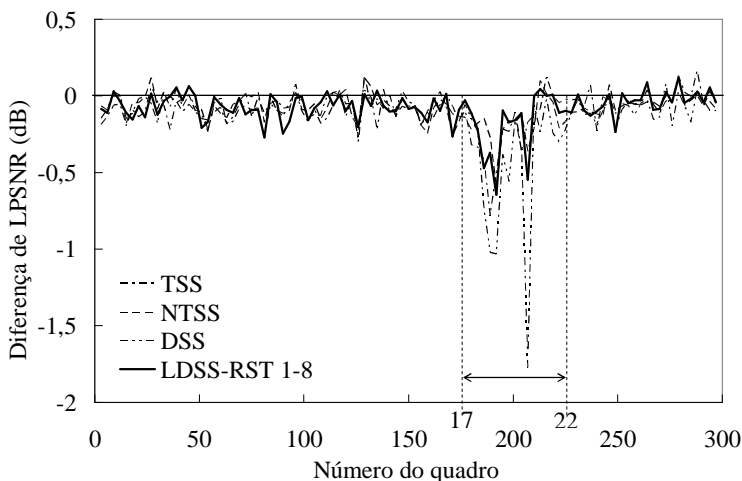


Figura 54 - Perda de desempenho RD medida em LPSNR das estratégias TSS, NTSS, DSS e LDSS-RST<sub>1-8</sub> com relação ao JM7.3-RDO, para a sequência *Foreman* com  $Q = 32$ .

## 6.4 Conclusões

A análise dos algoritmos TSS, NTSS e DSS, projetados especificamente para reduzir a complexidade da EM na condição de otimização da distorção, mostra que tais algoritmos se utilizados em codificadores de vídeo RDO têm um desempenho RD similar ao obtido com o algoritmo FS-RDO, quando a quantidade de movimento entre os quadros da sequência de vídeo é baixa. A grande vantagem desses ARs é a significativa redução de complexidade computacional da EM comparada ao algoritmo FS-RDO. No entanto, nas sequências com maior quantidade de movimento, esses algoritmos apresentam perda de desempenho RD, sendo que dentre os ARs analisados o algoritmo de menor complexidade (DSS) é o pior nesse quesito. Essa perda pode ser atribuída ao tamanho muito reduzido do seu GPI e do GP usados (eles têm apenas 3 *pixels*). Nos outros ARs com GPIs e GPs mais amplos, a perda de desempenho RD é menor.

A partir da avaliação quadro-a-quadro dos ARs também foi verificado que existe uma forte correlação entre a perda de desempenho e o aumento do número de MBs codificados no modo *Intra*. O aprisionamento dos ARs em mínimos da função custo RD é uma das principais causas que leva a escolha do modo *Intra* mesmo quando existem no espaço de busca VMs que levariam a um melhor desempenho para o modo *Inter*.

Para se conseguir um adequado compromisso entre o desempenho RD e a complexidade computacional, foi proposto o AR denominado LDSS, que apresenta um desempenho superior ou similar aos demais ARs, porém, exibindo uma complexidade computacional menor. Dentre as principais características desse AR, está o uso de GPIs adequados ao perfil de distribuição típico dos VMs de codificadores RDO. Tais GPIs, juntamente com os GPs em formato de diamante, conseguem antecipar a avaliação dos VMs de menor custo e consequentemente evitar a avaliação de um grande número de VMs cuja probabilidade de escolha é baixa. Assim, com o algoritmo LDSS é possível simplificar o processo de EM reduzindo o número de MBs nos quais o modo *Intra* é usado desnecessariamente.

No presente trabalho, a escolha do GPI foi fixada *a priori* para todos os MBs, sendo analisados separadamente os resultados obtidos com cada GPI. O GPI do LDSS<sub>1,8</sub> foi aquele que logrou o melhor compromisso entre complexidade e desempenho RD. Estima-se que as informações já codificadas no fluxo de bits permitam obter uma

predição da quantidade de movimento do MB, selecionando adaptativamente o GPI a ser usado.

Adicionalmente, foi possível reduzir ainda mais a complexidade computacional do processo de EM sem alterar o desempenho RD usando simultaneamente o algoritmo LDSS e a estratégia RST na EM. Com o algoritmo LDSS-RST foi possível codificar as sequências consideradas obtendo um desempenho próximo ao obtido com codificador JM7.3-RDO, porém, com complexidade computacional de apenas 2% do codificador de referência. Assim, concluímos que a combinação das duas propostas LDSS e RST é uma opção interessante para a codificação de vídeo, especialmente em aplicações de baixa taxa de bits, que exigem uma codificação com baixa complexidade computacional.

# Capítulo 7

## Conclusões e Trabalhos Futuros

### 7.1 Sumário das Contribuições

Este capítulo apresenta uma revisão das principais resultados e contribuições deste trabalho de pesquisa. Também são discutidas algumas questões que permanecem em aberto para futuras extensões e aperfeiçoamentos. O trabalho de pesquisa teve como foco principal o desenvolvimento de uma estratégia de codificação de vídeo de baixa complexidade computacional, mantendo o desempenho RD próximo àquele obtido com os codificadores otimizados em RD de alta complexidade.

#### *a) Conclusões Gerais*

Foi mostrado que a codificação de vídeo é um processo complexo, no qual muitas etapas cooperam para a obtenção do desempenho RD. Para codificar o sinal de vídeo com uma adequada taxa de compressão, são exploradas características do sinal de vídeo, tais como a redundância espacial, temporal, cor e de código. O uso das limitações do sistema visual humano permite aumentar a compressão, mantendo a distorção quase imperceptível ao observador humano (dependendo das condições de observação). Em virtude da interação do conjunto de etapas usado na codificação, a validação de qualquer estratégia nova deve ser realizada integrando-a em um codificador de vídeo completo. Somente nessa condição o desempenho RD e sua complexidade computacional podem ser efetivamente comparados com os de outras técnicas existentes.

No entanto, na fase inicial de análise de estratégias muitas vezes um modelo de teste que implemente apenas a etapa sob análise é usado com frequência, para isolar o efeito das demais etapas, permitindo avaliar em detalhes os pontos positivos e negativos obtidos. Usando um codificador otimizado apenas em distorção, foram analisados os principais fatores que afetam a complexidade do processo de EM. Concluiu-se que a escolha adequada do tamanho da janela de procura, o tamanho de bloco e a precisão da representação dos VMs são de fundamental importância tanto para o desempenho quanto para a

complexidade computacional. A partir dos resultados obtidos, podemos destacar os seguintes pontos:

- i) Blocos menores devem ser usados nas bordas dos objetos em movimento. O tamanho da janela de procura deve ser suficiente para abranger todos os movimentos dos objetos visuais no campo de visão mostrado pela sequência de vídeo;
- ii) Uma janela de procura fixa com tamanho de 16 ou 32 *pixels* é suficiente. No entanto, o ideal seria ajustar o tamanho da janela dinamicamente devido à complexidade computacional ser uma função exponencial em relação ao tamanho da janela. O algoritmo LDSS, proposto neste trabalho, usa diferentes GPIs para atender a esse requisito, permitindo usar janelas de até 128 *pixels*, sem comprometer a complexidade do processo;
- iii) A precisão com que os VMs são representados é outro fator essencial do processo de EM. O aumento da precisão implica uma melhoria significativa de qualidade da imagem a um custo de codificação e complexidade computacional muito menor do que para os demais fatores desse processo.

Também foi constatado que a técnica de otimização RD baseada no multiplicador de Lagrange [115] produz uma qualidade muito superior àquela obtida com codificadores otimizados em distorção para a mesma taxa de bits.

Verificou-se também que as curvas RD geralmente usadas na avaliação de desempenho dos codificadores RDO podem falsamente indicar uma equivalência entre codificadores (para uma determinada condição de codificação e sequência de vídeo) quando na realidade existem diferenças significativas de desempenho RD em alguns trechos do vídeo. Para permitir tal avaliação, foi proposta uma medida de distorção lagrangeana LPSNR que permite acompanhar quadro-a-quadro o desempenho RD dos codificadores sem a necessidade de fixação da taxa de bits. O uso dessa medida de distorção possibilita ajustar o codificador para que logre o máximo desempenho a cada quadro. Os resultados obtidos com tal medida sugerem que ela seja usada como instrumento de comparação complementar as curvas RD, principalmente, na fase de desenvolvimento e avaliação de novas estratégias e algoritmos de codificação, pois permite diferenciar os codificadores detectando seus pontos fortes e fracos em termos de desempenho RD.

### b) Principais Contribuições

Também foi mostrado que dentre todas as etapas de um codificador de vídeo H.264-RDO, o processo de EM-BMA e a avaliação dos diversos modos *Intra* são as etapas que apresentam a maior complexidade. Este trabalho apresentou a estratégia RST e o algoritmo rápido de EM denominado LDSS, os quais efetivamente reduzem a complexidade, mantendo o desempenho RD próximo ao do codificador de referência FS RDO. Ambas as propostas exploram a conexão entre o custo de codificação dos VMs e o perfil típico de distribuição dos VMs em codificadores otimizados em RD.

A estratégia RST, baseada no ordenamento e truncamento por custos dos VMs com regras de truncamento baseadas no custo e na qualidade da imagem, possibilitou descartar antecipadamente uma grande parcela dos VMs e uma significativa percentagem dos modos de codificação *Intra*, sem a necessidade de avaliar a distorção, mantendo o processo otimizado em RD. Para possibilitar o truncamento, o custo RD inicial mínimo foi usado na pré-seleção do modo *Skip* e pré-eliminação de uma grande quantidade de modos de codificação *Inter* e *Intra*. Na média de todas as sequências avaliadas foram obtidas reduções de complexidade no processo de EM-BMA entre 55% e 74%, chegando a uma redução de até 96% para sequências com menor movimento. O peso na carga computacional total do processo de predição do modo *Intra* que representa até 38% no codificador JM7.3-RDO foi reduzido para apenas 5% usando a estratégia RST. Simultaneamente foi demonstrada a pequena influência no desempenho RD das diferentes maneiras de predição de cor no modo *Intra*.

Avaliaram-se também os algoritmos rápidos de EM adaptados ao codificador RDO. Uma expressiva redução de até 97% na complexidade computacional foi obtida, sem produzir perdas significativas de desempenho RD para a maioria das sequências de vídeo testadas. Dentre os ARs avaliados, o algoritmo de busca logarítmica em formato de diamante (LDSS) proposto neste trabalho, foi aquele que apresentou o melhor compromisso entre complexidade e desempenho. Através da avaliação quadro-a-quadro, se constatou que o algoritmo LDSS também tem um desempenho superior aos demais ARs nos trechos mais críticos das sequências consideradas, onde há a maior perda de desempenho em relação ao algoritmo de referência. A melhoria do desempenho e a menor complexidade do algoritmo LDSS podem ser atribuídas ao emprego de um gabarito de procura mais adaptado ao perfil típico de distribuição dos VMs do codificador RDO.

Finalizando, foi avaliado o uso conjunto do algoritmo LDSS e da estratégia RST (LDSS-RST). Comparado essa abordagem com o codificador de referência RDO foi obtida uma redução de até 98% na complexidade do processo de EM e seleção dos modos de codificação, principalmente, para taxas de bits menores. Portanto, conclui-se que o codificador LDSS-RST é uma opção sedutora para a codificação de vídeo em aplicações nas quais a complexidade computacional é limitada, uma vez que, com uma complexidade de até 50 vezes menor é possível manter o desempenho RD próximo ao codificador H.264-RDO.

## 7.2 Sugestões para Trabalhos Futuros

Como contribuições futuras ao tema pesquisado, diversas são as possibilidades de aperfeiçoamento deste trabalho, que merecem o investimento de recursos financeiros e humanos. Destacamos a seguir alguns pontos específicos da estratégia RST e do algoritmo LDSS proposto que deveriam ser melhor estudados visando obter um maior ganho de desempenho e/ou uma maior redução de complexidade.

- i) Avaliar a estratégia RST em vídeos de maiores taxas de bits, tais como definição padrão (*standard definition* – SD) e alta definição (*high definition* – HD), tanto em termos de desempenho RD quanto em relação à redução de complexidade computacional. Nessas aplicações, o erro de predição é o fator predominante na taxa de bits final.
- ii) Outro desafio é estender a abordagem RST para o uso em conjunto com a codificação aritmética binária adaptativa ao contexto (*context adaptive binary arithmetic coding* – CABAC) e a codificação VLC adaptativa ao contexto (*context adaptive variable-length coding* – CAVLC), nas quais a taxa dos MVs devem ser calculadas e ordenadas para cada novo contexto.
- iii) Investigar uma maneira de selecionar a cada MB o GPI a ser usado pelo algoritmo LDSS, usando para tal uma predição da quantidade de movimento obtida com os quadros e blocos já codificados.
- iv) Desenvolver um estudo detalhado do processo de interpolação utilizado na determinação dos valores de *subpixel*, uma vez que foi constatada uma forte influência do tipo de interpolação sobre a qualidade da CM, sem implicar aumento de taxa de bits.
- v) Implementar o algoritmo e estratégias propostas neste trabalho usando processadores de sinais digitais (DSP) para se obter um codificador e decodificador que operem em tempo real.



Finalmente, tirando proveito da redução de complexidade computacional obtida com a estratégia RST, é sugerido que novos modos de codificação sejam incorporados aos codificadores H.264, os quais seriam apenas considerados quando o desempenho RD (ou qualidade) em um MB estiver abaixo de um valor mínimo. Essa proposta se torna viável com a utilização do algoritmo LDSS-RST, da medida LPSNR e das regras de truncamento por custo e qualidade. Essas regras farão com que os novos modos de codificação sejam apenas avaliados para um pequeno número de MBs que tem a maior influência na perda de desempenho RD total de cada quadro. Para tal, é sugerido que os novos modos sejam avaliados somente quando for patente que o processo de EM (modo *Inter*) e a predição espacial (modo *Intra*) com os modos antigos não alcançar um adequado desempenho RD para o MB. Como possíveis modos de codificação a serem incorporados aos codificadores, são sugeridos o uso das resoluções de  $1/8$ ,  $1/16$  e  $1/32$  *pixels*, com novas outras funções de interpolação, o particionamento do MB em blocos de até  $2 \times 2$  *pixels*, o uso de transformações afins (de escala e rotação) aplicadas ao MB [71], [97] e a codificação de parte do MB no modo *Inter* e parte no modo *Intra*.



# **Apêndice 1**

## **Sequências de Vídeo Usadas**

Descrição do conteúdo e tipo de cena mostrada em cada sequência de vídeo usada nas avaliações durante o trabalho. Na Figura 55 a Figura 59 são mostrados alguns quadros das sequências usadas, visando facilitar a comparação de resultados com propostas futuras.

Tabela 28 – Características das sequências de vídeo consideradas

<b>Sequência</b>	<b>Quadros</b>	<b>Duração</b>	<b>Formato</b>	<b>Características</b>
<i>Akiyo</i>	300	10 seg.	QCIF	Câmera fixa, pessoa em movimento lento, fundo artificial, sequência típica de videoconferência.
<i>Miss America</i>	150	5 seg.	QCIF	Câmera fixa, pessoa em movimento lento, sequência típica de videoconferência.
<i>News</i>	300	10 seg.	QCIF	Câmera fixa, pessoas em movimento lento, fundo artificial, sequência típica de videoconferência.
<i>Mother &amp; Daughter</i>	300	10 seg.	QCIF	Câmera fixa, em pessoas com movimentos lentos, sequência típica de videoconferência.
<i>Salesman</i>	257	8,57 seg.	QCIF	Câmera fixa, em pessoa com fundo artificial, sequência típica de videoconferência.
<i>Silent</i>	451	15,03 seg.	QCIF	Câmera fixa, em pessoa com movimentos rápidos, fundo com detalhes espaciais, sequência típica de videoconferência.
<i>Suzie</i>	150	5 seg.	QCIF	Câmera fixa, em pessoa com movimentos rápidos, sequência típica de videoconferência.
<i>Carphone</i>	382	12,73 seg.	QCIF	Câmera com movimento rápido, pessoa em movimento, fundo com paisagem passando.
<i>Foreman</i>	300	10 seg.	QCIF	Câmera com movimento rápido, cena com movimento do interlocutor; movimento de translação da câmera no final.
<i>HallObjects</i>	300	10 seg.	QCIF	Câmera fixa, movimento de pessoas e objetos, sequência típica de monitoração de ambientes.
<i>Container</i>	300	10 seg.	QCIF	Câmera fixa, movimento de objetos, paisagem e rio com fundo, sequência típica de monitoração de ambientes.
<i>Coastguard</i>	300	10 seg.	QCIF	Câmera com movimento rápido, objetos em movimento rápido, paisagem e rio com fundo.
<i>Mobile &amp; Calendar</i>	300	10 seg.	CIF	Câmera com movimento lento lateral e zoom; movimento complexo; fortes detalhes espaciais e de cor.
<i>Paris</i>	1065	35,5 seg.	CIF	Câmera fixa, pessoa em movimento lento, fundo detalhado, sequência típica de videoconferência.

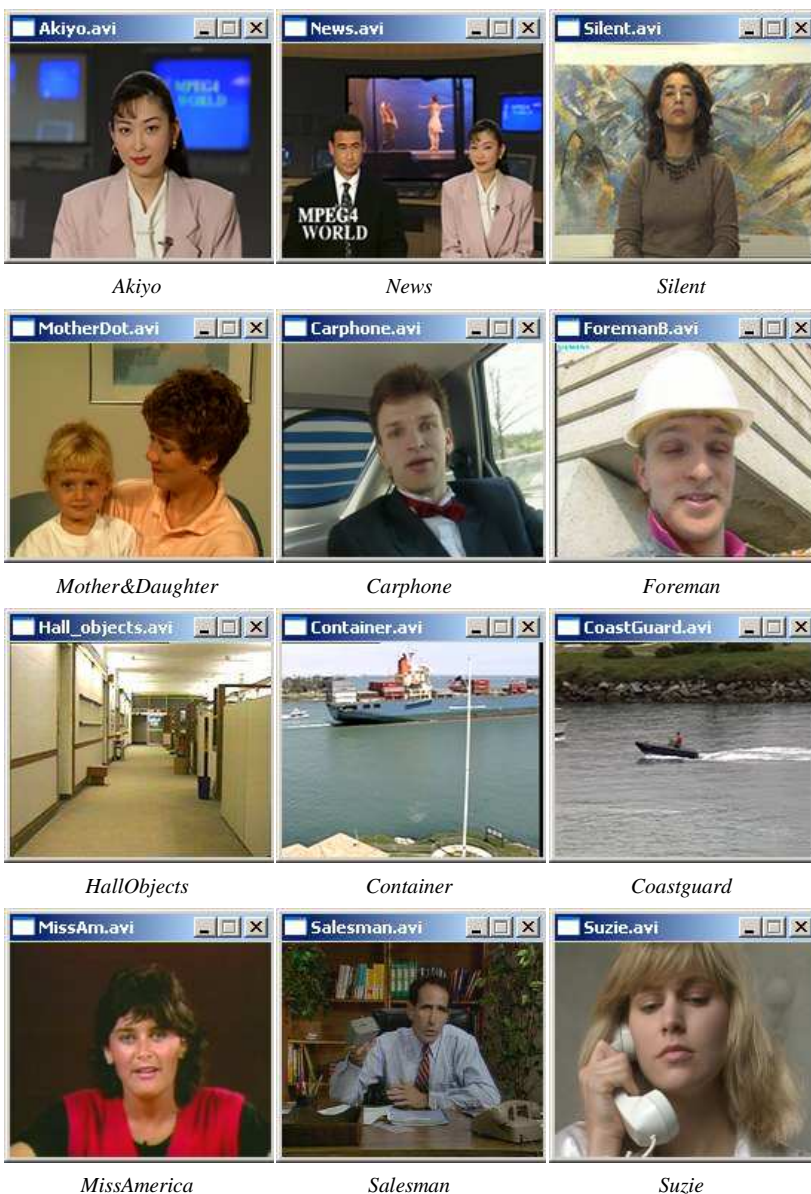


Figura 55 - Quadro inicial das sequências de vídeo em formato QCIF.

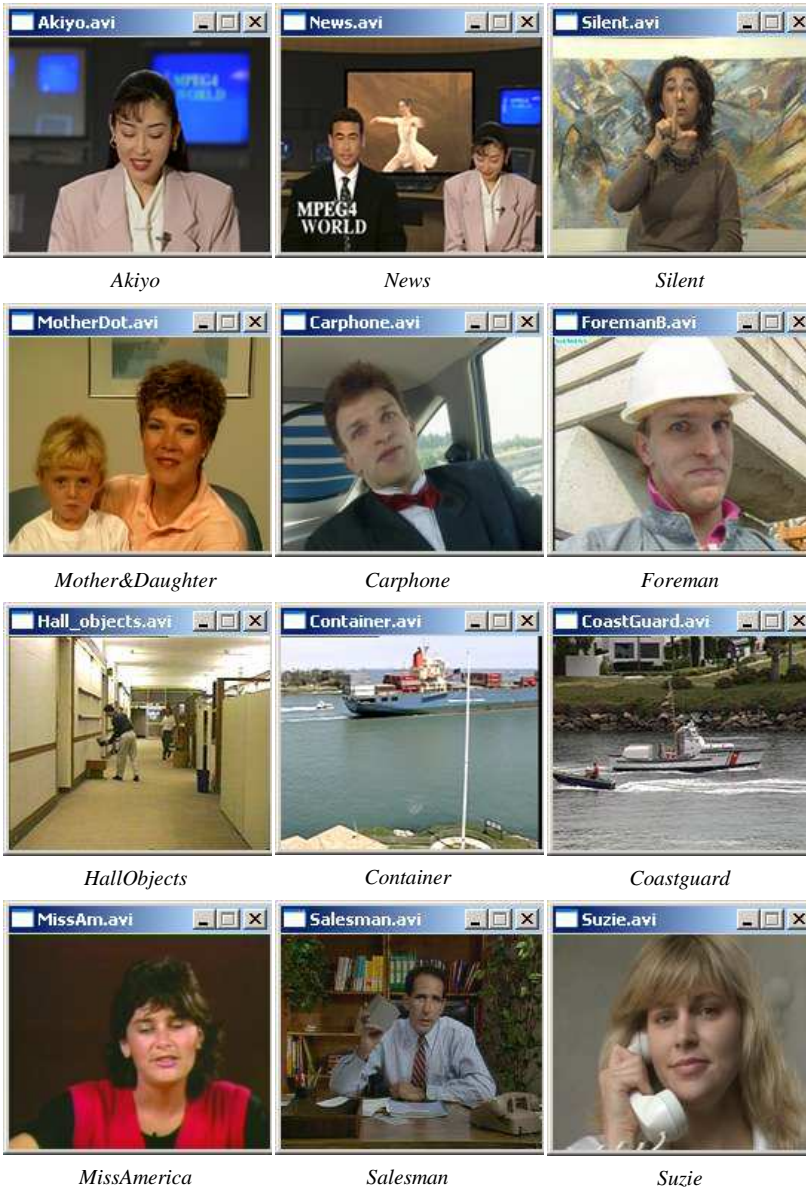


Figura 56 - Quadro #100 das seqüências de vídeo em formato QCIF.

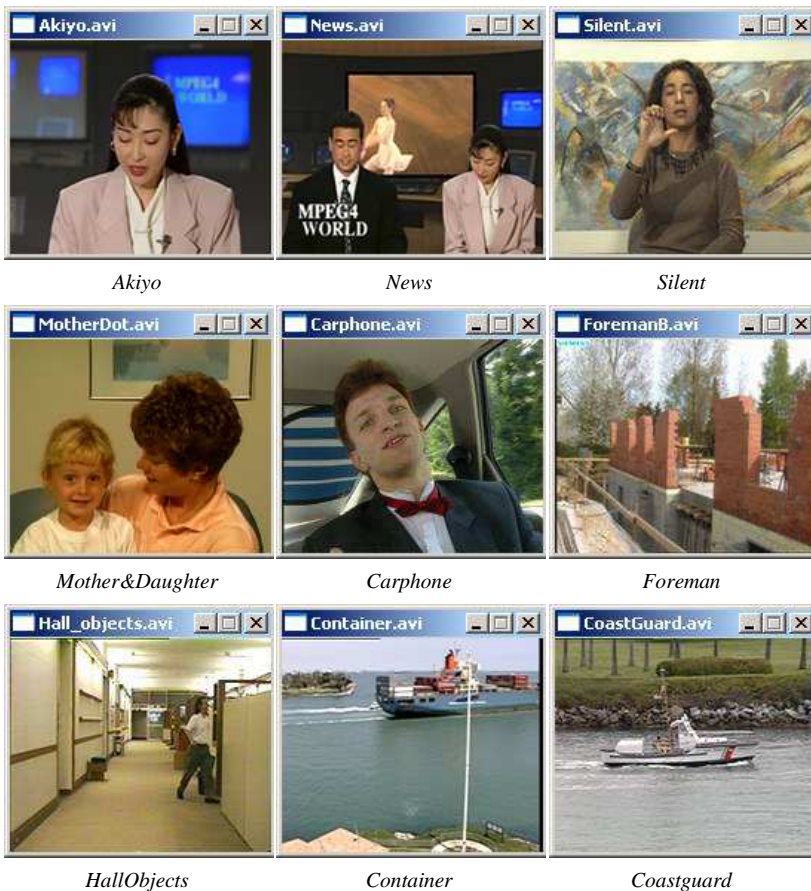


Figura 57 - Quadro #300 das seqüências de vídeo em formato QCIF.





*Mobile & Calendar*



*Paris*

Figura 58 - Quadro inicial das seqüências de vídeo em formato CIF.





*Mobile&Calendar*



*Paris*

Figura 59 - Quadro #300 das sequências de vídeo em formato CIF.



## Apêndice 2

### Resultados Complementares

Os resultados da Tabela 29 a Tabela 35 foram obtidos utilizando o codificador H.264 RDO FS e o codificador FullRST usando um limiar de qualidade  $D_L = 200$  e nível de prioridade do modo *Inter*  $k_l = 16$ . As curvas R-D mostradas nas Figura 60 a Figura 65 foram obtidas utilizando os dados de distorção e taxa de bits da Tabela 29 a Tabela 35.

Tabela 29 – Resultados de taxa-distorção e complexidade medida em NPP para a predição *Inter* e número de predições *Intra*. Sequência *Akiyo*-10Hz, formato QCIF

Q	taxa (kbps)	PSNR (dB)	PSNRY (dB)	PSNRU (db)	PSNRV (dB)	<i>Inter (NPP)</i>							<i>Intra (N)</i>		
						16x16	16x8	8x16	8x8	8x4	4x8	4x4	TOTAL	16x16	4x4
<i>Akiyo, FS RDO, QCIF, 10Hz, 10 Quadros</i>															
24	43,45	42,21	41,36	43,43	44,37	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
28	27,89	39,58	38,61	41,04	42,01	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
32	18,58	37,06	35,75	38,92	40,43	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
36	12,47	34,66	33,21	36,75	38,39	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
<i>Akiyo, FullRST, QCIF, 10Hz, DL = 200, kl = 16, 10 Quadros</i>															
24	43,30	42,12	41,25	43,40	44,34	1180	2310	2292	1835	2886	2791	2928	4833	0,894	0,694
28	27,66	39,48	38,48	40,99	41,97	1056	1969	1962	1394	1872	1862	1509	3931	0,726	0,548
32	17,95	36,93	35,57	38,88	40,41	894	1659	1628	1135	1342	1318	698	3198	0,591	0,477
36	12,02	34,55	33,05	36,72	38,40	608	1118	1104	653	583	566	218	2040	0,427	0,371

Tabela 30 – Resultados de taxa-distorção e complexidade medida em NPP para a predição *Inter* e número de predições *Intra*. Sequência *News-10Hz*, formato QCIF

Q	taxa (kbps)	PSNR (dB)	PSNRY (dB)	PSNRU (db)	PSNRV (dB)	<i>Inter (NPP)</i>							<i>Intra (N)</i>		
						16x16	16x8	8x16	8x8	8x4	4x8	4x4	TOTAL	16x16	4x4
<b><i>News, FSRDO, QCIF, 10Hz, 10 Quadros</i></b>															
24	91,168	40,8387	39,917	42,245	43,119	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
28	59,84	37,996	36,93	39,76	40,51	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
32	39,24	35,5333	34,05	38,05	38,94	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
36	25,25	32,8882	31,18	35,81	36,80	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
<b><i>News, FullRST, QCIF, 10Hz, DL = 200, kl = 16, 10 Quadros</i></b>															
24	91,032	40,7023	39,733	42,173	43,109	1708	3347	3389	3161	5008	5025	6233	7510	1,075	0,848
28	58,94	37,8677	36,76	39,75	40,42	1554	3025	3005	3332	4734	4574	4528	6849	0,987	0,777
32	38,79	35,4128	33,91	38,02	38,83	1348	2533	2561	2511	3259	3101	2532	5476	0,830	0,639
36	24,86	32,6818	30,92	35,70	36,70	1308	2337	2335	1872	2052	1906	1271	4687	0,701	0,549

Tabela 31 – Resultados de taxa-distorção e complexidade medida em NPP para a predição *Inter* e número de predições *Intra*. Sequência *Foreman-10Hz*, formato QCIF

Q	taxa (kbps)	PSNR (dB)	PSNRY (dB)	PSNRU (db)	PSNRV (dB)	<i>Inter (NPP)</i>							<i>Intra (N)</i>		
						16x16	16x8	8x16	8x8	8x4	4x8	4x4	TOTAL	16x16	4x4
<b><i>Foreman, FS RDO, QCIF, 10Hz, 10 Quadros</i></b>															
24	134,024	40,5298	39,084	42,425	44,418	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
28	86,09	38,1588	36,46	40,78	42,35	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
32	54,49	35,9672	33,81	39,60	40,96	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
36	35,23	33,6537	31,22	38,25	38,78	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
<b><i>Foreman, FullRST, QCIF, 10Hz, DL = 200, KI = 16, 10 Quadros</i></b>															
24	133,76	40,4972	39,029	42,413	44,454	4190	8048	7982	11409	18776	19036	22701	21204	1,400	0,995
28	85,69	38,1392	36,42	40,78	42,37	4132	7706	7629	10072	15294	15456	16374	19186	1,315	0,927
32	53,85	35,9147	33,74	39,55	40,96	3883	7023	7036	8397	11570	11708	10034	16549	1,127	0,785
36	34,60	33,5705	31,12	38,26	38,71	3781	6524	6615	6985	8333	8425	5660	14546	0,912	0,616

Tabela 32 – Resultados de taxa-distorção e complexidade medida em NPP para a predição *Inter* e número de predições *Intra*. Sequência *Mother&Daughter-10Hz*, formato QCIF

Q	taxa (kbps)	PSNR (dB)	PSNRY (dB)	PSNRU (db)	PSNRV (dB)	<i>Inter (NPP)</i>							<i>Intra (N)</i>		
						16x16	16x8	8x16	8x8	8x4	4x8	4x4	TOTAL	16x16	4x4
<b><i>Mother&amp;Daughter, FSRDO, QCIF, 10Hz, 10 Quadros</i></b>															
24	112,008	40,3252	39,362	42,177	42,326	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
28	66,67	37,7978	36,51	40,36	40,40	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
32	39,31	35,5063	33,70	39,23	39,01	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
36	23,00	33,4273	31,30	37,60	37,77	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
<b><i>Mother&amp;Daughter, FullRST, QCIF, 10Hz, DL = 200, kl = 16, 10 Quadros</i></b>															
24	112,12	40,2542	39,268	42,153	42,3	2907	5732	5555	8076	15113	15017	19946	15583	1,160	0,871
28	67,30	37,75	36,47	40,25	40,39	2805	5409	5244	7248	12480	12444	12885	13865	1,085	0,798
32	38,93	35,4283	33,60	39,20	38,98	2668	5052	4928	6712	9760	9670	7373	12227	1,011	0,725
36	22,30	33,369	31,19	37,67	37,79	2510	4430	4303	4459	5110	5077	3341	9474	0,848	0,646

Tabela 33 – Resultados de taxa-distorção e complexidade medida em NPP para a predição *Inter* e número de predições *Intra*. Sequência *Carphone-10Hz*, formato QCIF

Q	taxa (kbps)	PSNR (dB)	PSNRY (dB)	PSNRU (db)	PSNRV (dB)	<i>Inter (NPP)</i>							<i>Intra (N)</i>		
						16x16	16x8	8x16	8x8	8x4	4x8	4x4	TOTAL	16x16	4x4
<b><i>Carphone, FS RDO, QCIF, 10Hz, 10 Quadros</i></b>															
24	102,976	40,9493	40,045	42,504	43,012	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
28	63,82	38,4233	37,20	40,62	41,13	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
32	38,04	36,1075	34,48	39,18	39,53	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
36	23,34	33,8717	31,87	37,56	38,20	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
<b><i>Carphone, FullRST, QCIF, 10Hz, DL = 200, kl = 16, 10 Quadros</i></b>															
24	102,92	40,8533	39,911	42,429	43,047	3558	6821	6870	8397	13647	13616	14140	16795	1,147	0,912
28	63,43	38,3388	37,10	40,58	41,04	3396	6280	6296	7405	10838	10704	9073	14796	1,023	0,800
32	37,57	36,0065	34,29	39,26	39,61	3116	5500	5618	5771	7084	6973	4577	12161	0,859	0,696
36	23,02	33,7718	31,71	37,59	38,21	2645	4377	4416	3549	3682	3544	1793	8945	0,643	0,523



Tabela 34 – Resultados de taxa-distorção e complexidade medida em NPP para a predição *Inter* e número de predições *Intra*. Sequência *Coastguard-10Hz*, formato QCIF

Q	taxa (kbps)	PSNR (dB)	PSNRY (dB)	PSNRU (db)	PSNRV (dB)	<i>Inter (NPP)</i>							<i>Intra (N)</i>		
						16x16	16x8	8x16	8x8	8x4	4x8	4x4	TOTAL	16x16	4x4
<b><i>Coastguard, FSRDO, QCIF, 10Hz, 10 Quadros</i></b>															
24	244,216	40,34	37,349	46,039	46,605	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
28	136,47	37,6938	34,30	44,14	44,81	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
32	60,53	35,2752	31,38	42,51	43,64	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
36	26,64	33,276	29,08	41,66	41,67	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
<b><i>Coastguard, FullRST, QCIF, 10Hz, DL = 200, kl = 16, 10 Quadros</i></b>															
24	243,488	40,3307	37,321	46,036	46,664	4305	8503	8444	14049	27105	27110	40037	25570	1,672	1,109
28	136,54	37,7067	34,29	44,16	44,92	4298	8320	8325	13586	24808	24948	24525	23770	1,482	1,042
32	59,18	35,244	31,32	42,45	43,72	4256	7730	7792	11107	16797	17229	9469	19640	1,283	1,077
36	26,00	33,2092	29,00	41,70	41,56	3983	6406	6527	6561	6728	6878	3282	13996	0,983	0,955

Tabela 35 – Resultados de taxa-distorção e complexidade medida em NPP para a predição *Inter* e número de predições *Intra*. Sequência *HallObjects*-10Hz, formato QCIF

Q	taxa (kbps)	PSNR (dB)	PSNRY (dB)	PSNRU (db)	PSNRV (dB)	<i>Inter (NPP)</i>							<i>Intra (N)</i>		
						16x16	16x8	8x16	8x8	8x4	4x8	4x4	TOTAL	16x16	4x4
<b><i>Hall Objects, FS RDO, QCIF, 10Hz, 10 Quadros</i></b>															
24	55,888	40,4902	39,917	40,791	42,482	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
28	35,42	38,4093	37,57	39,24	40,93	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
32	24,30	36,2518	34,86	37,98	40,08	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
36	16,70	33,9482	32,01	36,56	39,09	4305	8610	8610	17220	34440	34440	68880	30135	3,606	3,606
<b><i>Hall Objects, FuIRST, QCIF, 10Hz, DL = 200, kl = 16, 10 Quadros</i></b>															
24	52,96	40,2388	39,805	40,218	41,995	1092	1946	1964	988	1630	1598	1808	3811	1,193	1,055
28	34,46	38,2828	37,50	38,98	40,70	536	972	993	625	1056	1026	1413	2025	0,881	0,742
32	23,74	36,1172	34,78	37,69	39,91	449	757	756	546	937	900	1299	1653	0,724	0,587
36	16,27	33,8453	31,85	36,52	39,15	301	536	538	446	644	628	656	1151	0,565	0,464

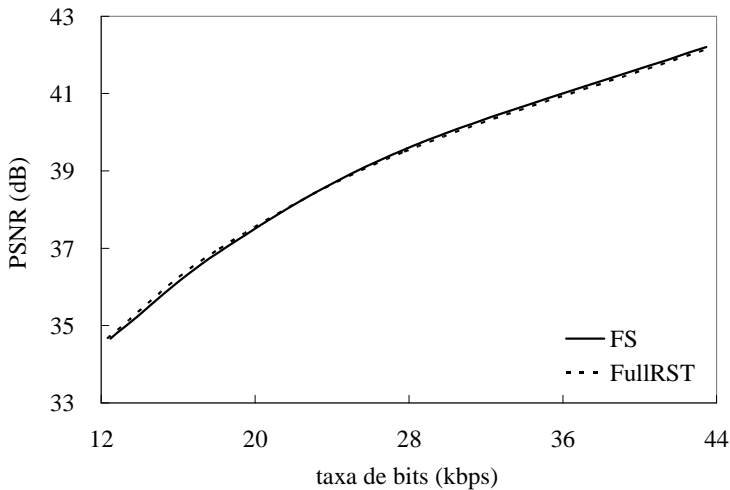


Figura 60 - Desempenho RD do codificador H.264 RDO FS e do codificador FullRST, sequência *Akiyo*.

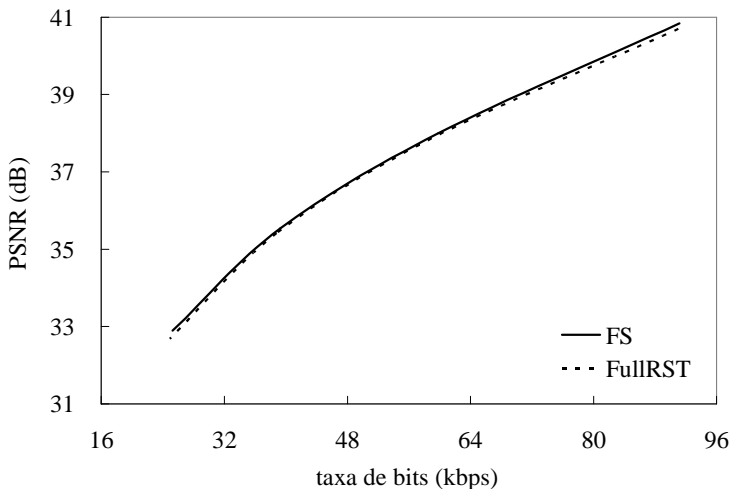


Figura 61 - Desempenho RD do codificador H.264 RDO FS e do codificador FullRST, sequência *News*.

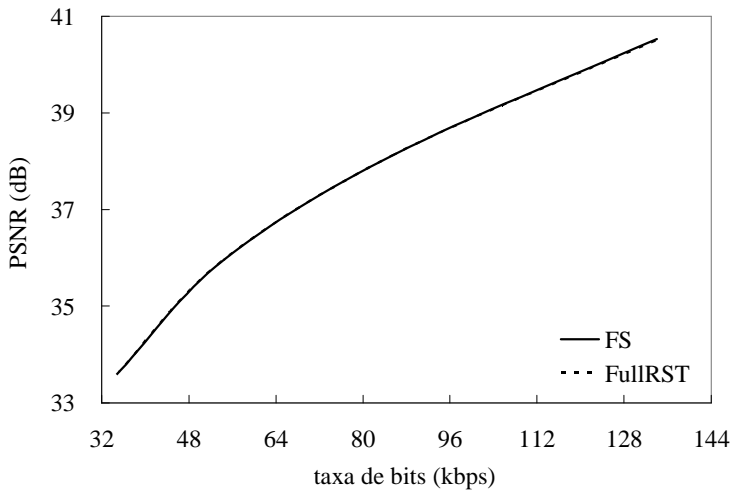


Figura 62 - Desempenho RD do codificador H.264 RDO FS e do codificador FullRST, sequência *Foreman*.

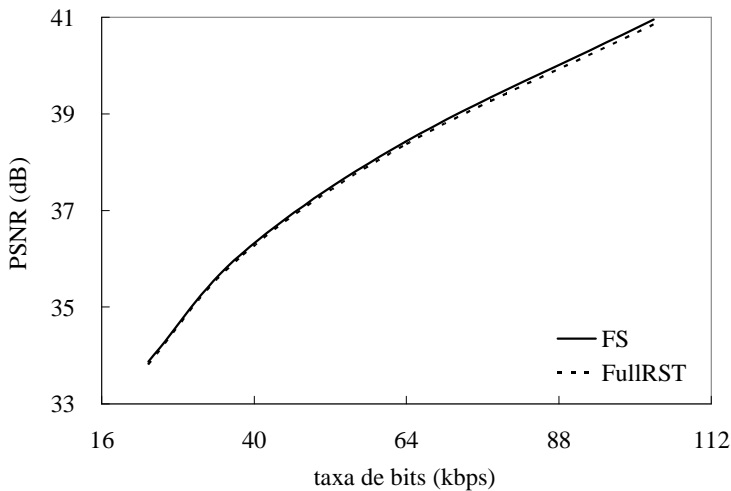


Figura 63 - Desempenho RD do codificador H.264 RDO FS e do codificador FullRST, sequência *Carphone*.

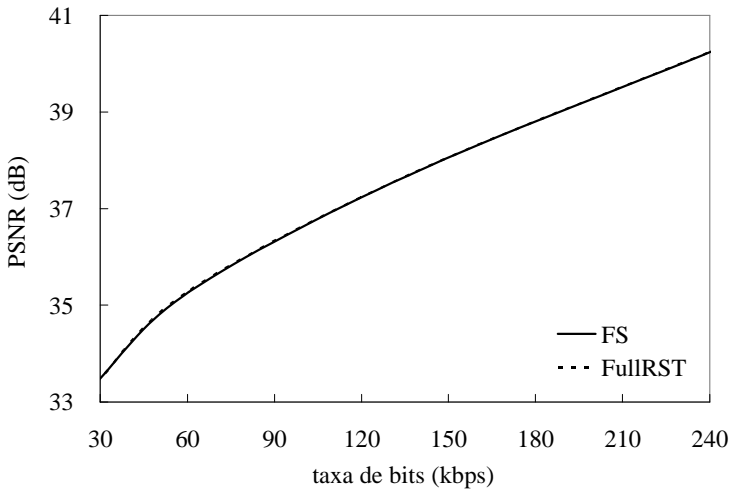


Figura 64 - Desempenho RD do codificador H.264 RDO FS e do codificador FullRST, sequência *Coastguard*.

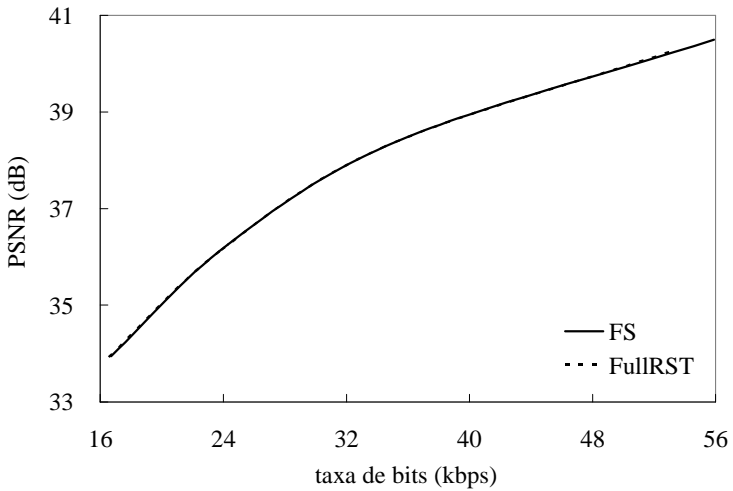


Figura 65 - Desempenho RD do codificador H.264 RDO FS e do codificador FullRST, sequência *HallObjects*.



## Apêndice 3

### Publicações da Tese

- [1] M. Moecke e R. Seara, "Melhorando a Compensação de Movimentos da Crominância para Codificadores de Vídeo Baseados na Recomendação H.263," *Anais XX Simpósio Brasileiro de Telecomunicações*, Rio de Janeiro, RJ, Out. 2003, pp. 231-236.
- [2] M. Moecke e R. Seara, "Rate-Distortion Optimized Video Coding with Stopping Rules: Quality and Complexity," in *Proc. IEEE Int. Conf. Image Processing*, Singapore, Oct. 2004, pp. 753-756.
- [3] M. Moecke e R. Seara, "Algoritmo Rápido de Estimação de Movimento Otimizado em Taxa-Distorção para Codificação de Vídeo com Baixa Taxa de Bits," *Anais XXII Simpósio Brasileiro de Telecomunicações*, Campinas, SP, Set. 2005, pp. 120-125.
- [4] M. Moecke e R. Seara, "Redução de Complexidade Através da Ordenação de Custo dos Processos de Codificação em Vídeo Digital". *Anais XXV Simpósio Brasileiro de Telecomunicações*, Recife, PE, Set. 2007 pp 1-6.
- [5] E. T. M. Manoel, M. Moecke, e R. Seara, "Codificação de Macrobloco Duplo Inter no Padrão H.264". *Anais XXV Simpósio Brasileiro de Telecomunicações*, Recife, PE, Set. 2007 pp 1-6.
- [6] M. Moecke e R. Seara, "Sorting Rate in Video Encoding Process for Complexity Reduction," *IEEE Trans. Circuits Syst. Video Technol*, aceito para publicação.





## Referências Bibliográficas

- [1] R. M. Gray e D. L. Neuhoff, “Quantization,” *IEEE Trans. Information Theory*, vol. 44, no. 6, pp. 1-63, Oct. 1998.
- [2] S. Winkler, “Issues in Vision Modeling for Perceptual Video Quality Assessment,” *Signal Processing*, vol. 78, no. 2, pp. 231-252, Oct. 1999.
- [3] G. Sharma, M. Vreth, e H. Trussell, “Color Imaging for Multimedia,” *Proc. IEEE*, vol. 86, no. 6, pp. 1088-1108, June 1998.
- [4] R. V. Cox, B. G. Haskell, Y. Lecun et al., “Scanning the Technology – On the Application of Multimedia Processing to Communications,” *Proc. IEEE*, vol. 86, no. 5, pp. 755-823, May 1998.
- [5] M. Yuen e H. R. Wu, “A Survey of Hybrid MC/DPCM/DCT Video Coding Distortion,” *Signal Processing*, vol. 70, no. 3, pp. 247-278, 1998.
- [6] M. Moecke, “*Codificação de Vídeo Colorido para Taxas Baixas de Bits, Explorando as Características do Sistema Visual Humano*,” Florianópolis, 2002, Exame de Qualificação (Doutorado em Engenharia Elétrica) – Centro Tecnológico, Universidade Federal de Santa Catarina.
- [7] S. P. Lloyd, “Least Squares Quantization in PCM,” *IEEE Trans. Information Theory*, vol. 28, no. 2, pp. 127-135, Mar. 1982.
- [8] J. Max, “Quantizing for Minimum Distortion,” *IRE Trans. Information Theory*, vol. 6, no. 1, pp. 7-12, Mar. 1960.
- [9] A. K. Jain, *Fundamental of Digital Image Processing*. Prentice-Hall, Englewood Cliffs, NJ: 1989.
- [10] Y. Linde, A. Buzo, e R. M. Gray, “An Algorithm for Vector Quantization,” *IEEE Trans. Comm.*, vol. 28, no. 1, pp. 84-95, Jan. 1980.
- [11] C. Huang e R. Harris, “A Comparison of Several Vector Quantization Codebook Generation Approaches,” *IEEE Trans. Image Processing*, vol. 2, no. 1, pp. 108-112, Jan. 1993.
- [12] C. E. Shannon, “Communications in the Presence of Noise,” *Proc. IEEE*, vol. 86, no. 2, pp. 447-457, Feb. 1998.
- [13] O. Egger, P. Fleury, T. Ebrahimi, e M. Kunt, “High-Performance Compression of Visual Information – A Tutorial Review – Part I; Still Pictures,” *Proc. IEEE*, vol. 87, no. 6, pp. 976-1011, June 1999.
- [14] D. A. Huffman, “A Method for the Construction of Minimum-Redundancy Codes,” *Proc. IRE*, vol. 40, no. 2, pp. 1098-1101, Sept. 1952.
- [15] ITU-T Recommendation H.261, *Codes for Audiovisual Services at px64 kbit/s*, Int. Telecomm. Union, Geneva, Switzerland, 1990.

- [16] ITU-T Recommendation H.263, *Video Coding for Low Bit Rate Communication*, Int. Telecomm. Union, Geneva, Switzerland, 1998.
- [17] ITU-T Recommendation H.264, “*Advanced Video Coding for Generic Audiovisual Services*,” Int. Telecomm. Union, Geneva, Switzerland, 2003.
- [18] ISO/IEC 11172-2 (MPEG-1), *Information Technology — Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s — Part 2: Video*, Int. Organization for Standardization, Aug. 1993.
- [19] ITU-T Recommendation H.262 | ISO/IEC 13818-2 (MPEG-2), *Information Technology - Generic Coding of Moving Pictures and Associated Audio Information: Video*, Int. Telecomm. Union | Int. Organization for Standardization, Geneva, Switzerland, 1995.
- [20] ISO/IEC JTC1/SC29/WG11 Doc. N3747, “MPEG-4 Overview - (V.16 – La BauleVersion)”, La Baule, Oct. 2000.
- [21] ISO/IEC JTC1/SC29/WG11 Doc. N1902, “ISO/IEC 14496-2 (Committee Draft) - Information Technology - Coding of Audio-Visual Objects: Visual,” Int. Organization for Standardization, Fribourg, Oct. 1997.
- [22] J. Wen e J. D. Villasenor, “A Class of Reversible Variable Length Codes for Robust Image and Video Coding,” in *Proc. IEEE Int. Conf. Image Processing*, St. Barbara, CA, USA, vol. 2, Oct. 1997, pp. 65-68.
- [23] G. Côté, B. Erol, M. Gallant et al., “H.263+: Video Coding at Low Bit Rates,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 7, pp. 849-866, Nov. 1998.
- [24] P. G. Howard e J. S. Vitter, “Arithmetic Coding for Data Compression,” *Proc. IEEE*, vol. 82, no. 6, pp. 857-865, June 1994.
- [25] I. H. Witten, R. M. Neal, e J. G. Cleary, “Arithmetic Coding for Data Compression,” *Comm. ACM*, vol. 30, no. 6, pp. 520-540, June 1987.
- [26] ITU-T Recommendation T.81| ISO/IEC 10918-1 (JPEG), *Information Technology – Digital Compression and Coding of Continuous-tone Still Images – Requirements and Guidelines*, Int. Telecomm. Union, Geneva, Switzerland, 1992.
- [27] K. A. S. Immink, “Runlength-Limited Sequences,” *Proc. IEEE*, vol. 78, no. 11, pp. 1745-1759, Nov. 1990.
- [28] T. Wiegand, G. Sullivan, G. Bjontegaard et al., “Overview of the H.264/Video Coding Standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560-576, July 2003.
- [29] D. Marple, H. Schwarz, e T. Wiegand, “Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620-635, July 2003.

- [30] Z. Wei, Y. Fu, Z. Gao et al., “Visual Compander in Wavelet-Based Image Coding,” *IEEE Trans. Consumer Electronics*, vol. 44, no. 4, pp. 1261-1266, Nov. 1998.
- [31] I. R. Greenshields e J. A. Rosiene, “A Fast Wavelet-Based Karhunen–Loeve Transform,” *Pattern Recognition*, vol. 31, no. 7, pp. 839–845, July 1998.
- [32] H. S. Malvar, A. Hallapuro, M. Karczewicz et al., “Low-Complexity Transform and Quantization in H.264/AVC”, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 598-603, July 2003.
- [33] S. G. Mallat, “A Theory for Multiresolution Signal Decomposition: The Wavelet Representation,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674-693, July 1989.
- [34] K. Ramchandran, M. Vetterli, e C. Herley, “Wavelet, Subband Coding, and Best Bases,” *Proc. IEEE*, vol. 84, no. 4, pp. 541-560, Apr. 1996.
- [35] M. Vetterli e C. Herley, “Wavelet and Filter Bank: Theory and Design,” *IEEE Trans. Signal Processing*, vol. 40, no. 9, pp. 2207-2232, Sept. 1992.
- [36] ITU-T Recommendation T.800 - ISO/IEC FCD 15444-1, *Information Technology – JPEG 2000 Image Coding System*, v 1.0, Int. Telecomm. Union, Geneva, Switzerland, 2000.
- [37] A. Skodras, C. Christopoulos, e T. Ebrahimi, “The JPEG2000 Still Image Compression Standard,” *IEEE Signal Processing Mag.*, vol. 18, no. 5, pp. 36-58, Sept. 2001.
- [38] M. Antonini, M. Barlaud, P. Mathieu et al., “Image Coding Using the Wavelet Transform,” *IEEE Trans. Image Processing*, vol. 1, no. 2, pp. 205-220, Apr. 1992.
- [39] L. Gall e A. Tabatabai, “Sub-band Coding of Digital Images Using Symetric Short Kernel Filters and Arithmetic Coding Techniques,” in *Proc. IEEE Int. Acoustics, Speech, Signal Processing*, New York, USA, Apr. 1988, pp. 761-765.
- [40] M. D. Adams e F. Kossentini, “Reversible Integer-to-Integer Wavelet Transforms for Image Compression: Performance Evaluation and Analysis,” *IEEE Trans. Image Processing*, vol. 9, no. 6, pp. 1010-1024, June 2000.
- [41] D. Taubman, “High Performance Scalable Image Compression with EBCOT,” *IEEE Trans. Image Processing*, vol. 9, no. 7, pp. 1158-1170, July 2000.
- [42] D. Taubman e A. Zakhor, “Multirate 3-D Subband Coding of Video,” *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 572-588, Sept. 1994.
- [43] M. Nadenau, “Integration of Human Color Vision Models into High Quality Image Compression,” Lausanne, 2000, Doctoral Dissertation - Département D’Électricité, École Polytechnique Fédérale de Lausanne.

- [44] S. Mallat e Z. Zhang, "Matching Pursuits with Time-Frequency Dictionaries," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.
- [45] R. Neff e A. Zakhor, "Very Low Bit-Rate Video Coding Based on Matching Pursuits," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 158-171, Feb. 1997.
- [46] T. Ebrahimi e M. Kunt, "Visual Data Compression for Multimedia Applications," *Proc. IEEE*, vol. 86, no. 6, pp. 1109-1125, June 1998.
- [47] F. Dufaux and F. Moscheni, "Motion Estimation Techniques for Digital TV: a Review and a New Contribution," *Proc. IEEE*, vol. 83, no. 6, pp. 858-876, June 1995.
- [48] T. Ebrahimi, E. Reusens, e W. Li, "New Trends in Very Low Bitrate Video Coding," *Proc. IEEE*, vol. 83, no. 6, pp. 877–891, June 1995.
- [49] G. Karlsson e M. Vetterli, "Three-Dimensional Sub-band Coding of Video," in *Proc. IEEE Int. Acoustics, Speech, Signal Processing*, New York, USA, Apr. 1988, pp. 1100-1103.
- [50] B. -J. Kim, Z. Xiong, e W. A. Pearlman, "Low Bit-Rate Scalable Video Coding with 3-D Set Partitioning in Hierarchical Trees (3-D SPIHT)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 8, pp. 1374-1387, Dec, 2000.
- [51] C. I. Podilchuk, N. S. Jayant, e N. Farvardi, "Three-Dimensional Subband Coding of Video," *IEEE Trans. Image Processing*, vol. 4, no. 2, pp. 125-139, Feb. 1995.
- [52] G. J. Conklin e S. S. Hemami, "A Comparison of Temporal Scalability Techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 6, pp. 909-919, Sept. 1999.
- [53] S. A. Martucci, I. Sodagar, T. Chiang et al., "A Zerotree Wavelet Video Coder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 109-118, Feb. 1997.
- [54] D. Marpe e H. L. Cycon, "Very Low Bit-Rate Video Coding Using Wavelet-Based Techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 85-94, Feb. 1999.
- [55] H. -W. Park e H. -S. Kim, "Motion Estimation Using Low-Band-Shift Method for Wavelet-Based Moving-Picture Coding," *IEEE Trans. Image Processing*, vol. 9, no. 4, pp. 577-587, Apr. 2000.
- [56] K. Cinkler, "Very Low Bit-Rate Wavelet Video Coding," *IEEE J. Select. Areas Comm.*, vol. 16, no. 1, pp. 4-11, Jan. 1998.
- [57] B. Ostermann, "Differences between an Object-Based Analysis-Synthesis Coder and a Block-Based Hybrid Coder," in *Proc. IEEE Int. Conf. Image Processing*, Washington, DC, USA, vol. 2, Oct. 1995, pp. 398-401.

- [58] J. -R. Ohm, "Three-Dimensional Subband Coding with Motion Compensation," *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 559-571, Sept. 1994.
- [59] A. Wang, Z. Xiong, P. A. Chou et al., "Three-Dimensional Wavelet Coding of Video with Global Motion Compensation," in *Proc. Data Compression Conf.*, Snowbird, UT, USA, Mar. 1999, pp. 404-413.
- [60] H. G. Musmann, M. Hotter, e J. Ostermann, "Object-Oriented Analysis-Synthesis Coding of Moving Objects," *Signal Processing: Image Comm.*, vol. 1, no. 2, pp. 117-138, Oct. 1989.
- [61] M. Hotter, "Object-Oriented Analysis-Synthesis Coding Based on Moving Two-Dimensional Objects," *Signal Processing: Image Comm.*, vol. 2, no. 4, pp. 409-428, Dec. 1990.
- [62] P. Salembier, L. Torres, F. Meyer et al., "Region-Based Video Coding Using Mathematical Morphology," *Proc. IEEE*, vol. 83, no. 6, pp. 843-857, June 1995.
- [63] G. M. Schuster e A. K. Katsaggelos, "A Video Compression Scheme with Optimal Bit Allocation among Segmentation, Motion and Residual Error," *IEEE Trans. Image Processing*, vol. 6, no. 11, pp. 1487-1502, Nov. 1997.
- [64] F. Bossen e T. Ebrahimi, "A Simple and Efficient Binary Shape Coding Technique Based on Bitmap Representation," in *Proc. IEEE Int. Acoustics, Speech, Signal Processing*, Munich, Germany, vol. 4, Apr. 1997, pp. 3129-3132.
- [65] C. R. Dyer, A. Rosenfeld, e H. Samet, "Region Representation: Bondary Codes from Quadrees," *Comm. ACM*, vol. 23, no. 3, pp. 171-179, Mar. 1980.
- [66] N. Herodotou, K. N. Plataniotis, e A. N. Venetsanopoulos, "A Color Segmentation Scheme for Object-Based Video Coding," in *Proc. IEEE Symp. Advances in Digital Filtering and Signal Processing*, Victoria, BC, Canada, June 1998, pp. 25-29.
- [67] B. Girod, K. B. Younes, R. Bernstein et al., "Recent Advances in Video Compression," in *Proc. IEEE Int. Symp. Circuits Syst.*, Atlanta, USA, vol. 2, May 1996, pp. 580-583.
- [68] D. Pearson, "Developments in Model-Based Video Coding," *Proc. IEEE*, vol. 83, no. 6, pp. 892-906, June 1995.
- [69] K. Aizawa and T. S. Huang, "Model-Based Image Coding: Advanced Video Coding Techniques for Very Low Bit-Rate Applications," *Proc. IEEE*, vol. 83, no. 2, pp. 259-271, Feb. 1995.
- [70] H. Li, A. Lundmark, and R. Forchheimer, "Image Sequence Coding at Very Low Bitrates: A Review," *IEEE Trans. Image Processing*, vol. 3, no. 9, pp. 589-609, Sept. 1994.

- [71] Y. -C. Li e Y. -C. Chen, “A Hybrid Model-Based Image Coding System for Very Low Bit-Rate Coding,” *IEEE J. Select. Areas Comm.*, vol. 16, no. 1, pp. 28-41, Jan. 1998.
- [72] International Commission on Illumination, “*Colorimetry*,” Central Bureau of the CIE, Vienna, Austria, CIE Pub. 15.2, 1986.
- [73] B. Hill, T. Roger, e F. W. Vorhagen, “Comparative Analysis of the Quantization of Color Spaces on the Basis of the CIELAB Color Difference Formula,” *ACM Trans. on Graphics*, vol. 16, no. 2, pp. 109–154, Apr. 1997.
- [74] W. K. Pratt, “Spatial Transform Coding of Color Images,” *IEEE Trans. Comm.*, vol. 9, no. 12, pp. 980–992, Dec. 1971.
- [75] K. Jack, *Video Demystified: A Handbook for the Digital Engineer*. 3 ed. LLH Technology, Vancouver, Canada: 2001.
- [76] ITU-R Recommendation BT.601-5, *Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-Screen 16:9 Aspect Ratios*, Int. Telecomm. Union, Geneva, Switzerland, 1995.
- [77] J. O. Limb e C. B. Rubinstein, “Plateau Coding of the Chrominance Component of Color Picture Signals,” *IEEE Trans. Comm.*, vol. 22, no. 3, pp. 812-820, June 1974.
- [78] U. Gargi, R. Kasturi, S. Strayer et al., “An Evaluation of Color Histogram Based Methods in Video Indexing,” Dept. of Computer Science and Engineering, Penn State University, University Park, PA, *Tech. Rep. CSE-96-053*, Sept. 1996, 30 p.
- [79] International Commission on Illumination, “*Industrial Color Difference Evaluation*,” Central Bureau of the CIE, Vienna, Austria, CIE Pub. 15.2, 1995.
- [80] W. Zeng, S. Daly, e S. Lei, “An Overview of the Visual Optimization Tools in JPEG 2000,” *Signal Processing: Image Comm.*, vol. 17, no. 1, pp. 85-104, Jan. 2002.
- [81] R. Lakhan e P. Vimal, “Spatial Frequency Discrimination: A Comparison of Achromatic and Chromatic Conditions,” *Vision Research*, vol. 42, no. 5, pp. 599-611, Mar. 2002.
- [82] M. Moecke e R. Seara, “Melhorando a Compensação de Movimentos da Crominância para Codificadores de Vídeo Baseados na Recomendação H.263,” *Anais XX Simpósio Brasileiro de Telecomunicações*, Rio de Janeiro, RJ, Out. 2003, pp. 231-236.
- [83] S. -C. Han e J. W. Woods, “Adaptive Coding of Moving Objects for Very Low Bit Rates,” *IEEE J. Select. Areas Comm.*, vol. 16, no. 1, pp. 56-70, Jan. 1998.

- [84] A. A. Kassim e L. Zhao, "Rate-Scalable Object-Based Wavelet Codec with Implicit Shape Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 7, pp. 1068-1079, Oct. 2000.
- [85] Y. -Q. Zhang e S. Zafar, "Motion-Compensated Wavelet Transform Coding for Color Video Compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, no. 3, pp. 285-296, Sept. 1992.
- [86] K. Shen e E. J. Delp, "Color Image Compression Using an Embedded Rate Scalable Approach," in *Proc. IEEE Int. Conf. Image Processing*, St. Barbara, CA, USA, vol. 3, Oct. 1997, pp. 34-37.
- [87] A. Bist, A. Jacquin, e C. Podilchuk, "Encoding of the Chrominance Signals in 3D Subband-Based Video Coding," in *Proc. IEEE Int. Acoustics, Speech, Signal Processing*, Minneapolis, MN, USA, vol. 5, Apr. 1993, pp. 329-332.
- [88] L. Karam e C. Podilchuk, "Chroma Coding for Video at Very Low Bit Rates," in *Proc. IEEE Int. Conf. Image Processing*, Washington, DC, USA, vol. 1, Oct. 1995, pp. 562-565.
- [89] A. Dumitras e F. Kossentini, "FANN-Based Video Chrominance Subsampling," in *Proc. IEEE Int. Acoustics, Speech, Signal Processing*, Seattle, WA, USA, vol. 2, May 1998, pp. 1077-1080.
- [90] C. Du e Y. He, "Early Detection of All Zero Chroma Blocks in H.263," in *Proc. 5th Int. Conf. Signal Processing*, Beijing, China, vol. 2, Aug. 2000, pp. 1110-1114.
- [91] JVT JM73 implementation. Disponível em <http://bs.hhi.de/~suehring/tml/download/jm73.zip>
- [92] B. Wohlberg e G. De Jager, "A Review of the Fractal Image Coding Literature," *IEEE Trans. Image Processing*, vol. 8, no. 12, pp. 1716-1729, Dec. 1999.
- [93] M. S. Lazar e L. T. Bruton, "Fractal Block Coding of Digital Video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 3, pp. 297-308, June 1994.
- [94] K. U. Barthel e T. Voeye, "Three-Dimensional Fractal Video Coding," in *Proc. IEEE Int. Conf. Image Processing*, Washington, DC, USA, vol. 3, Oct. 1995, pp. 260-263.
- [95] A. Schuster e A. Katsaggelos, "Fast and Efficient Mode and Quantizer Selection in the Rate Distortion Sense for H.263," in *Proc. Visual Comm. Image Processing*, Orlando, FL, USA, vol. 2727, Mar. 1996, pp. 784-795.
- [96] T. Wiegand e B. Girod, "Lagrangian Multiplier Selection in Hybrid Video Coder Control," in *Proc. IEEE Int. Conf. Image Processing*, Thessaloniki, Greece, Oct. 2001, pp. 542-545

- [97] C. S. Fuh e P. Maragos, "Affine Models for Image Matching and Motion Detection," in *Proc. IEEE Int. Acoustics, Speech, Signal Processing*, Toronto, Canada, vol. 4, May 1991, pp. 2409-2412.
- [98] V. Seferidis e M. Ghanbari, "General Approach to Block Matching Motion Estimation," *Opt. Eng.*, vol. 32, no. 7, pp. 1464-1474, July 1993.
- [99] P. Treves e J. Konrad, "Motion Estimation and Compensation under Varying Illumination," in *Proc. IEEE Int. Conf. Image Processing*, Austin, TX, USA, vol. 1, Nov. 1994, pp. 373-377.
- [100] M. Gallant, G. Cote, e F. Kossentini, "An Efficient Computation-Constrained Block-Based Motion Estimation Algorithm for Low Bit Rate Video Coding," *IEEE Trans. Image Processing*, vol. 8, no. 12, pp. 1816-1823, Dec. 1999.
- [101] ITU-T, *Video Codec Test model Near-Term, version 8 (TMN8)*, Int. Telecomm. Union, Sunriver, OR, USA, Sept. 1997.
- [102] ITU-T, *Test model 12 / Appendix II of H.263, version 3 (TMN12)*, Int. Telecomm. Union, Portland, OR, USA, Aug. 2000.
- [103] F. Kossentini e Y. Lee, "Computation-Constrained Fast MPEG-2 Encoding," *IEEE Signal Processing Letters*, vol. 4, no. 8, pp. 224-226, Aug. 1997.
- [104] J. Jain e A. Jain, "Displacement Measurement and its Application in Interframe Image Coding," *IEEE Trans. Comm.*, vol. 29, no. , pp. 1799-1808, Dec. 1981.
- [105] T. Koga, K. Linuma, A. Hirano et al., "Motion Compensated Interframe Coding for Video Conferencing," in *Proc. National Telecomm. Conf.*, New Orleans, LA, USA, vol. G5, Nov. 1981, pp. 3.1-3.5.
- [106] R. Li, B. Zeng, e M. L. Liou, "A New Three Step Search Algorithm for Block Motion Estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 4, pp. 438-442, Aug. 1994.
- [107] L. M. Po and W. C. Ma, "A Novel Four Step Search Algorithm for Fast Block Motion Estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 313-317, June 1996.
- [108] L. K. Liu e E. Feig, "A Block Based Gradient Descent Search Algorithm for Block Motion Estimation in Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 4, pp. 419-422, Aug. 1996.
- [109] W. Li e E. Salari, "Successive Elimination Algorithm for Motion Estimation," *IEEE Trans. Image Processing*, vol. 4, no. 1, pp. 105-107, Jan. 1995.
- [110] R. T. DelPietro e M. H. M. Costa. "A Pattern Matching Algorithm Applied to the Estimation of Motion Vectors in Video Compression," In *Proc. SBrT/IEEE International Telecomm. Symposium 2002*, Natal, Set. 2002.



- [111] C. -G. Kim, I. -J. Lee, e S. -D. Kim, “Reduced Uneven Multi-hexagon-grid Search for Fast Integer Pel Motion Estimation in H.264/AVC,” LNCS 4633, pp. 708-714, 2007.
- [112] J. Lee and B. W. Dickinson, “Subband Video Coding with Scene-adaptive Hierarchical Motion Estimation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 3, pp. 459-466, Apr. 1999.
- [113] M. -J. Chen, L. -G. Chen, R. -M. Weng et al., “Efficient Hierarchical Motion Estimation Algorithm Based on Visual Pattern Block Segmentation,” in *Proc. IEEE Int. Symp. Circuits Syst.*, Hong Kong, vol. 2, June 1997, pp. 1429-1432.
- [114] J. Malo, F. J. Ferri, J. Albert et al., “Splitting Criterion for Hierarchical Motion Estimation Based on Perceptual Coding,” *Electronics Letters*, vol. 34, no. 6, pp. 541-543, Mar.1998.
- [115] G. Sullivan e T. Wiegand, “Rate-Distortion Optimization for Video Compression,” *IEEE Signal Processing Mag.*, vol. 15, no. 11, pp. 74-90, Nov. 1998.
- [116] J. Zhang, M. O. Ahmad, e M. N. S. Swamy, “A New Variable Size Block Motion Compensation,” in *Proc. IEEE Int. Conf. Image Processing*, St. Barbara, CA, USA, vol. 2, Oct. 1997, pp. 164-167.
- [117] H. -S. Oh e H. -K. Lee, “Adaptive Adjustment of the Search Window for Block-Matching Algorithm with Variable Block Size,” *IEEE Trans. Consumer Electronics*, vol. 44, no. 3, pp. 659-666, Aug. 1998.
- [118] I. Rhee, G. R. Martin, S. Muthukrishnan et al., “Quadtree-Structured Variable-Size Block-Matching Motion Estimation with Minimal Error,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 1, pp. 42-50, Feb. 2000.
- [119] M. Moecke e R. Seara, “Algoritmo Rápido de Estimção de Movimento Otimizado em Taxa-Distorção para Codificação de Vídeo com Baixa Taxa de Bits,” *Anais XXII Simpósio Brasileiro de Telecomunicações*, Campinas, SP, Set. 2005, pp. 120-125.
- [120] W. H. Press, S. A. Teukosky, W. T. Vetterling et al., *Numerical Recipes in C: The Art of Scientific Computing*. 2 ed. Cambridge Univ. Press, Cambridge, UK: 1992.
- [121] ITU-R Recommendation BR.1356, *User Requirements for Application of Compression in Television Production*, Int. Telecomm. Union, Geneva, Switzerland, 1998.
- [122] M. C. Chen e A. N. Willson, Jr., “A High Accuracy Predictive Logarithmic Motion Estimation Algorithm for Video Coding,” in *Proc. IEEE Int. Symp. Circuits Syst.*, Seattle, WA, USA, vol. 1, Apr. 1995, pp. 617-620.

- [123] H. Song e C. -C. Jay Kuo, "Rate Control for Low-Bit-Rate Video via Variable-Encoding Frame Rates," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 4, pp. 512-521, Apr. 2001.
- [124] Y. Yang e S. S. Hemami, "Rate-Distortion Optimization for VBR Environments Using a Motion Compensated Wavelet Coder," in *Proc. IEEE Int. Conf. Image Processing*, Kobe, Japan, Oct. 1999, pp. 352-356.
- [125] S. Y. Hu, M. C. Chen, e A. N. Willson Jr., "A Fast Rate-Distortion Optimization Algorithm for Motion Compensated Video Coding," in *Proc. IEEE Int. Symp. Circuits Syst.*, Hong Kong, vol. 2, June 1997, pp. 1349-1352.
- [126] F. Kossentini, Y. -W. Lee, M. J. T. Smith et al., "Predictive RD Optimized Motion Estimation for Very Low Bit-Rate Video Coding," *IEEE J. Select. Areas Comm.*, vol. 15, no. 9, pp. 1752-1763, Dec. 1997.
- [127] G. M. Schuster e A. K. Katsaggelos, "A Theory for the Optimal Bit Allocation Between Displacement Vector Field and Displaced Frame Difference," *IEEE J. Select. Areas Comm.*, vol. 15, no. 9, pp. 1739-1751, Dec. 1997.
- [128] G. M. Schuster e A. K. Katsaggelos, "A Video Compression Scheme with Optimal Bit Allocation among Segmentation, Motion, and Residual Error," *IEEE Trans. Image Processing*, vol. 6, no. 11, pp. 1487-1502, Nov. 1997.
- [129] M. C. Chen e A. N. Willson, Jr., "Rate-Distortion Optimal Motion Estimation Algorithms for Motion-Compensated Transform Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 2, pp. 147-158, Apr. 1998.
- [130] J. Ribas-Corbera e S. Lei, "Rate Control in DCT Video Coding for Lowdelay Video Communication," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 172-185, Feb. 1999.
- [131] Y. Yang e S. S. Hemami, "Generalized Rate-Distortion Optimization for Motion-Compensated Video Coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 6, pp. 942-955, Sept. 2000.
- [132] Y. Yang e S. S. Hemami, "Rate-Constrained Variable Block Size Motion Estimation and Perceptual Coding," in *Proc. IEEE Int. Conf. Image Processing*, St. Barbara, CA, USA, vol. 1, Oct. 1997, pp. 81-84.
- [133] G. J. Sullivan e R. L. Baker, "Rate-Distortion Optimized Motion Compensation for Video Compression Using Fixed or Variable Size Blocks," in *Proc. IEEE Global Telecomm. Conf. 1991*, Phoenix, USA, vol. 3.3.1, Dec. 1991, pp. 85-90.

- [134] S. M. Akramullah, I. Ahmad, e M. L. Liou, "Optimization of H.263 Video Encoding Using a Single Processor Computer: Performance Tradeoffs and Benchmarking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 8, pp. 901-914, Aug. 2001.
- [135] J. Hartung, A. Jacquin, J. Pawlyk et al., "Object-Oriented H.263 Compatible Video Coding Platform for Conferencing Applications," *IEEE J. Select. Areas Comm.*, vol. 16, no. 1, pp. 42-55, Jan. 1998.
- [136] S. C. Yoon, K. Ratakonda, e N. Ahuja, "Low Bit-Rate Video Coding with Implicit Multiscale Segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 7, pp. 1115-1129, Oct. 1999.
- [137] S. -C. Han e J. W. Woods, "Adaptive Coding of Moving Objects for Very Low Bit Rates," *IEEE J. Select. Areas Comm.*, vol. 16, no. 1, pp. 56-70, Jan. 1998.
- [138] W. Chung, F. Kossentini, e M. J. T. Smith, "An Efficient Motion Estimation Technique Based on a Rate-Distortion Criterion," in *Proc. IEEE Int. Acoustics, Speech, Signal Processing*, Atlanta, USA, vol. 4, May 1996, pp. 1926-1929.
- [139] Y. Shoham e A. Gersho, "Efficient Bit Allocation for an Arbitrary Set of Quantizers," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 9, pp. 1445-1453, Sept. 1988.
- [140] K. Ramchandran e M. Vetterli, "Best Wavelet Packet Bases in a Rate-Distortion Sense," *IEEE Trans. Image Processing*, vol. 2, no. 1, pp. 160-174, Apr. 1993.
- [141] T. Wiegand, M. Lightstone, D. Mukherjee et al., "Rate-Distortion Optimized Mode Selection for Very Low it Rate Video Coding and the Emerging H.263 Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 2, pp. 182-190, Apr. 1996.
- [142] T. Wiegand, H. Schwarz, A. Joch et al., "Rate-Constrained Coder Control and Comparison of Video Coding Standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 688-703, July 2003.
- [143] K. Ramchandran, A. Ortega, e M. Vetterli, "Bit Allocation for Dependent Quantization with Application to Multiresolution and MPEG Video Coder," *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 533-545, Sept. 1994.
- [144] D. Mukherjee e S. K. Mitra, "Combined Mode Selection and Macroblock Step Adaptation for H.263 Video Encoder," in *Proc. IEEE Int. Conf. Image Processing*, St. Barbara, CA, USA, vol. 2, Oct. 1997, pp. 37-40.
- [145] ISO/IEC JTC1/SC29/WG11 Doc. AVC-491, "Coded Representation of Picture and Audio Information—MPEG-2 Test Model 5," International Organization for Standardization, Sydney, Australia, Apr. 1993.

- [146] L. Yang, K. Yu, J. Li et al., “An effective variable block-size early termination algorithm for H.264 video coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 6, pp. 784–788, Jun. 2005.
- [147] C. S. Kannangara, I. E. G. Richardson, M. Bystrom et al., “Low-complexity skip prediction for H.264 through Lagrangian cost estimation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 2, pp. 202–208, Feb. 2006.
- [148] ISO/IEC MPEG and ITU-T VCEG Joint Video Team, JVT-I020, “Fast Mode Decision for H.264,” 2003.
- [149] Z. Chen, J. Xu, Y. He et al., “Fast integer-pel and fractional-pel motion estimation for H.264/AVC,” *J. Vis. Commun. Image Represent.*, vol. 17, no. 2, pp. 264–290, Apr. 2006.
- [150] A. Ahmad, N. Khan, S. Masud et al., “Efficient block size selection in H.264 video coding standard,” *Electron. Lett.*, vol. 40, no. 1, pp. 19–21, Jan. 8, 2004.
- [151] F. Pan, X. Lin, S. Rahardja et al., “Fast Intra Mode Decision Algorithm for H.264/AVC Video Coding,” in *Proc. IEEE Int. Conf. Image Processing*, Singapore, Oct. 2004, pp. 781-784.
- [152] Y. Zhang, F. Dai e S. Lin, “Fast 4x4 intra-prediction mode selection for H.264”, in *Proc. Int. Conf. Multimedia and Expo*, Taipei, Taiwan, Jun. 2004, pp. 1151-1154.
- [153] F. Fu, X. Lin, e L. Xu, “Fast intra prediction algorithm in H.264/AVC,” in *Proc. 7th Int. Conf. Signal Processing*, Beijing, China, Sep. 2004, pp. 1191-1194.
- [154] J.-H. Kim, B.-G. Kim, e C.-S. Cho, “Fast intra-mode prediction algorithm in H.264/AVC video coding,” *IEICE Trans. Info and Systems*, vol. E90-D, no. 8, pp. 1320-1323, Aug. 2007.
- [155] J. S. Park e H. J. Song, “Fast selective intra mode decision in H.264/AVC,” in *Proc. Consumer Communications and Networking Conf.*, Las Vegas, NV, Jan. 2006, pp. 1068-1072.
- [156] S. Sun e S. Chen, “A novel fast intra-mode decision algorithm for H.264/AVC,” in *Proc. Congress Image and Signal Processing*, Sanya, China, May 2008, pp. 324-328.
- [157] ISO/IEC MPEG e ITU-T VCEG Joint Video Team, JVT-K049, “Joint Model Reference Encoding Methods and Decoding Concealment Methods,” 2004.
- [158] ISO/IEC MPEG e ITU-T VCEG Joint Video Team, JVT-I020, “Fast INTER Mode Selection,” 2003.

- [159] J. Lee e B. Jeon, “Pruned mode decision based on variable block sizes motion compensation for H.264,” *Lecture Note in Computer Sciences - Multimedia Interactive Protocols and Systems (MIPS)*, vol. 2899, pp. 410-418, Nov. 2003.
- [160] P. Yin, H.-Y. Cheong, A. M. T. Torapis et al., “Fast Mode Decision and Motion Estimation for JVT/H.264,” in *Proc. IEEE Int. Conf. Image Processing*, Singapore, Sept. 2003, vol.3, pp. 853-856.
- [161] M. Moecke e R. Seara, “Rate-Distortion Optimized Video Coding with Stopping Rules: Quality and Complexity,” in *Proc. IEEE Int. Conf. Image Processing*, Singapore, Oct. 2004, pp. 753-756.
- [162] B. Zeng, R. Li, e M. L. Liou, “Optimization of Fast Block Motion Estimation Algorithms,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 6, pp. 833-844, Dec. 1997.
- [163] S.-J. Choi e J. W. Woods, “Motion-Compensated 3-D Subband Coding of Video,” *IEEE Trans. Image Processing*, vol. 8, no. 2, pp. 155-167, Feb. 1999.
- [164] T. Wiegand, X. Zhang, e B. Girod, “Long-Term Memory Motion-Compensated Prediction,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 70-84, Feb. 1999.
- [165] ITU-T Recommendation H.263, *Video Coding for Low Bit Rate Communication: Annex U – Enhanced Reference Picture Section Mode*, Int. Telecomm. Union, Geneva, Switzerland, 2000.
- [166] A. Tamhankar e K. R. Rao, “An Overview of H.264/Mpeg4 Part 10,” in *Proc. 4<sup>th</sup> EURASIP Conf.*, Zagreb, Croatia, July 2003, pp. 1-51.
- [167] C. Kim, H.-H. Shih, e C.-C. J. Kuo, “Feature-Based Intra-Prediction Mode Decision for H.264,” in *Proc. IEEE Int. Conf. Image Processing*, Singapore, Oct. 2004, pp. 769-772.
- [168] C.-L. Yang, L.-M. Po, e W.-H. Lam, “A Fast H.264 Intra Prediction Algorithm Using Macroblock Properties,” in *Proc. IEEE Int. Conf. Image Processing*, Singapore, Oct. 2004, pp. 461-464.