

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

GUILHERME RIBEIRO CORRÊA

**Estudo e Desenvolvimento de Heurísticas e
Arquiteturas de Hardware para Decisão
Rápida do Modo de Codificação de Bloco
para o Padrão H.264/AVC**

Dissertação apresentada como requisito parcial
para a obtenção do grau de Mestre em Ciência
da Computação

Prof. Dr. Sergio Bampi
Orientador

Prof. Dr. Luciano Agostini
Co-orientador

Porto Alegre, agosto de 2010.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Corrêa, Guilherme Ribeiro

Estudo e Desenvolvimento de Heurísticas e Arquiteturas de Hardware para Decisão Rápida do Modo de Codificação de Bloco para o Padrão H.264/AVC / Guilherme Ribeiro Corrêa – Porto Alegre: Programa de Pós-Graduação em Computação, 2010.

95 f.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2010. Orientador: Sergio Bampi.

1. Compressão de vídeo. 2. H.264/AVC 3. Decisão de Modo 4. Hardware. I. Bampi, Sergio. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Pós-Graduação: Prof. Aldo Bolten Lucion

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do PPGC: Prof. Álvaro Freitas Moreira

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço ao Daniel Palomino, ao Cláudio Diniz e ao Roger Porto pelas suas contribuições essenciais para a conclusão deste trabalho. Também agradeço aos demais colegas e ex-colegas do GACI e do GME que participaram de discussões sobre este e outros trabalhos acadêmicos que realizei durante os últimos anos.

Agradeço ao Prof. Luciano Agostini pela orientação, pela revisão do texto e pelas injeções de motivação nos maus momentos. Ao Prof. Bampi, obrigado por ter me aceito como seu orientando no PPGC e por propiciar as condições para que eu desenvolvesse o meu trabalho.

Obrigado aos meus pais, à Amanda e ao Gonçalo por todo o amor e apoio durante este período. Obrigado também à Déia por prover a minha moradia e todas as condições necessárias para o meu bem-estar durante os meus estudos em Porto Alegre.

Agradeço também ao CNPq pela bolsa recebida durante todos os dezoito meses que antecederam a defesa da dissertação.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	9
LISTA DE FIGURAS	11
LISTA DE TABELAS	13
RESUMO.....	15
ABSTRACT.....	17
1 INTRODUÇÃO	19
2 COMPRESSÃO DE VÍDEO	23
2.1 Resolução e Amostragem	23
2.2 Espaço de Cores.....	23
2.3 Subamostragem de Cores em YCbCr.....	24
2.4 Redundância de Dados	24
2.4.1 Redundância Espacial	25
2.4.2 Redundância Temporal	25
2.4.3 Redundância Entrópica	25
2.5 Métricas de Distorção.....	25
2.6 O Padrão H.264/AVC.....	26
2.6.1 Terminologia	26
2.6.2 Perfis e Níveis	28
2.7 Núcleo do Codec H.264/AVC.....	29
2.7.1 Funcionamento Básico.....	29
2.7.2 Decisão de Modo.....	31
3 PREDIÇÕES INTRA-QUADRO E INTER-QUADROS	33
3.1 A Predição Intra-Quadro	33
3.1.1 Predição Intra-Quadro 4x4.....	35
3.1.2 Predição Intra-Quadro 16x16.....	38
3.1.3 Predição para Crominância	38
3.2 A Predição Inter-Quadros.....	39
3.2.1 Estimação de Movimento	40
3.2.2 Compensação de Movimento	42
3.3 Processo de Codificação	43
4 DECISÃO RÁPIDA DE MODO DE CODIFICAÇÃO	45
4.1 Decisão em Quadros I e Decisão em Quadros P.....	45
4.2 Decisão Rápida de Modo Intra-Quadro	46
4.2.1 Decisão Intra-Quadro com Bloco de Tamanho Fixo Baseada em Distorção	47

4.2.2	Decisão de Melhor Tamanho de Bloco Intra Baseada em Heterogeneidade.....	52
4.3	Decisão Rápida de Modo Inter-Quadros	57
4.3.1	Decisão 1: Detecção de Modo <i>SKIP</i> Baseada em Estacionaridade.....	59
4.3.2	Decisão 2: Detecção de Sub-Particionamento Baseada em Heterogeneidade....	61
4.3.3	Decisão 3: Tamanho de Particionamento com Base na Força de Borda	63
4.3.4	Decisão 4: Tamanho de Sub-Particionamento com Base na Força de Borda.....	66
4.4	Decisões Rápidas Integradas e Análise de Complexidade.....	68
4.5	Trabalhos Relacionados	71
4.5.1	Trabalho de Lee et al. (2004)	72
4.5.2	Trabalho de Sun et al. (2008)	72
4.5.3	Trabalho de Fengqin et al. (2008)	733
4.5.4	Trabalho de Kim et al. (2004b)	73
4.5.5	Trabalho de Lee et al. (2009)	73
4.5.6	Trabalho de Jeon et al. (2009)	744
4.5.7	Trabalho de Wu et al. (2005)	74
4.5.8	Trabalho de Shen et al. (2008)	74
4.5.9	Comparação com Trabalhos Relacionados	755
5	ARQUITETURAS PARA DECISÃO RÁPIDA DE MODO DE CODIFICAÇÃO	77
5.1	Arquiteturas para Decisão Intra-Quadro.....	77
5.1.1	Arquiteturas para Cálculo de Distorção.....	77
5.1.2	Arquitetura para Detecção de Heterogeneidade.....	82
5.1.3	Arquiteturas para Decisão de Melhor Modo Intra-Quadro.....	84
5.2	Arquiteturas para Decisão Inter-Quadros	85
5.2.1	Arquitetura para Cálculo de Força de Bordas em Partições	86
5.2.2	Arquitetura para Cálculo de Força de Bordas em Sub-Partições	86
5.2.3	Arquiteturas para Decisão de Melhor Modo Inter-Quadros	86
5.3	Análise dos Resultados de Síntese	88
6	CONCLUSÕES E TRABALHOS FUTUROS.....	89
	REFERÊNCIAS.....	91

LISTA DE ABREVIATURAS E SIGLAS

AVC	<i>Advanced Video Coding</i>
BAC	<i>Binary Arithmetic Coding</i>
BITSTREAM	Sequência de bits do vídeo codificado
CABAC	<i>Context-Based Adaptive Binary Arithmetic Coding</i>
CAVLC	<i>Context-Based Adaptive Variable Length Coding</i>
Cb	<i>Chrominance Blue</i>
CIF	<i>Common Intermediate Format</i>
CODEC	Codificador/Decodificador
Cr	<i>Chrominance Red</i>
DCT	<i>Discrete Cosine Transform</i>
DVD	<i>Digital Versatile Disk</i>
FDCT 2-D	<i>Bidimensional Discrete Cosine Transform</i>
FRExt	<i>Fidelity Range Extensions</i>
GACI	Grupo de Arquiteturas e Circuitos Integrados
GME	Grupo de Microeletrônica
GOP	<i>Group of Pictures</i>
HD	<i>High Definition</i>
HDTV	<i>High Definition Digital Television</i>
HSI	<i>Hue, Saturation, Intensity</i>
HVC	<i>High Efficiency Video Coding</i>
I_PCM	<i>Intra-frame Pulse Code Modulation</i>
IDR	<i>Instantaneous Decoding Refresh</i>
INTRA	Predição Intra-Quadro
INTER	Predição Inter-Quadros
ITU	<i>International Telecommunication Union</i>
ITU-T	<i>International Telecommunication Union - Telecommunications</i>
ISO	<i>International Organization for Standardization</i>

JVT	<i>Joint Video Team</i>
LUT	<i>Look-up Table</i>
MC	<i>Motion Compensation</i>
ME	<i>Motion Estimation</i>
MPEG	<i>Motion Picture Experts Group</i>
MSE	<i>Mean-Squared Error</i>
MVC	<i>Multiview Video Coding</i>
PSNR	<i>Peak Signal-to-Noise Ratio</i>
Q	<i>Quantização</i>
Q^{-1}	<i>Quantização Inversa</i>
QCIF	<i>Quarter Common Intermediate Format</i>
QP	<i>Quantization Parameter</i>
RD	<i>Rate-Distortion</i>
RDO	<i>Rate-Distortion Optimization</i>
RGB	<i>Red, Green, Blue</i>
SAD	<i>Sum of Absolute Differences</i>
SATD	<i>Sum of Absolute Transformed Differences</i>
SDTV	<i>Standard Television</i>
SI	<i>Switch-I</i>
SP	<i>Switch-P</i>
SSD	<i>Sum of Squared Differences</i>
SVC	<i>Scalable Video Coding</i>
T	<i>Transformadas</i>
T^{-1}	<i>Transformadas Inversas</i>
UFRGS	<i>Universidade Federal do Rio Grande do Sul</i>
VCEG	<i>Video Coding Experts Group</i>
VGA	<i>Video Graphics Array</i>
VHDL	<i>VHSIC Hardware Description Language</i>
VLC	<i>Variable Length Coding</i>
Y	<i>Luminance</i>

LISTA DE FIGURAS

Figura 2.1: Perfis do H.264/AVC	29
Figura 2.2: Diagrama em blocos de um codificador H.264/AVC	29
Figura 2.3: Diagrama em blocos de um decodificador H.264/AVC	31
Figura 3.1: Tipos de macrobloco Intra no padrão H.264/AVC	34
Figura 3.2: Resultado da predição intra-quadro 4x4. Quadro original (esquerda); Resíduo da predição intra-quadro 4x4 (direita).....	35
Figura 3.3: Identificação das amostras para a predição de um bloco 4x4.	35
Figura 3.4: Identificação das amostras para a predição de um bloco 4x4.	36
Figura 3.5: Modos de predição I16MB	38
Figura 3.6: Modos de predição para blocos 8x8 de crominância	39
Figura 3.7: Divisão do macrobloco em partições de macrobloco	40
Figura 3.8: Divisão do sub-macrobloco em partições	40
Figura 3.9: Uso de múltiplos quadros de referência	41
Figura 3.10: Processo de codificação e decisão de modo baseada em RDO.	43
Figura 4.1: Matriz de transformação A após simplificações e reduções ao primeiro quadrante.	53
Figura 4.2: Matriz de transformação C obtida a partir da matriz de transformação A através de fatoração.....	54
Figura 4.3: Matriz de escala E obtida a partir da matriz de transformação A através de fatoração.	55
Figura 4.4: Hierarquia proposta para decisão de tamanho de partição e sub-partição de macrobloco (modos inter-quadros).....	58
Figura 4.5: Aumento percentual da taxa de bits gerado pela decisão <i>SKIP</i> baseada em estacionaridade e pela decisão RDO sem modo <i>SKIP</i> , em relação à RDO completa. ...	61
Figura 4.6: Aumento do PSNR (em dB) gerado pela decisão <i>SKIP</i> baseada em estacionaridade e pela decisão RDO sem modo <i>SKIP</i> , em relação à RDO completa. ...	61
Figura 4.7: Valores de H (heterogeneidade) de cada macrobloco da sequência <i>MAN IN CAR</i> codificado sem sub-particionamento através da decisão RDO completa.	62
Figura 4.8: Valores de H (heterogeneidade) de cada macrobloco da sequência <i>MAN IN CAR</i> codificado com sub-particionamento através da decisão RDO completa.....	62
Figura 4.9: Pixels utilizados (em cinza) no cálculo de detecção de bordas vertical e horizontal para definição do particionamento de macrobloco	65
Figura 4.10: Pixels utilizados (em tons de cinza) no cálculo de detecção de bordas vertical e horizontal para definição do sub-particionamento de macrobloco.....	67
Figura 4.11: Processo de codificação e decisão de modo proposta.....	70
Figura 5.1: Arquitetura SAD com 16 amostras de entrada.	78
Figura 5.2: Arquitetura SATD com 16 amostras de entrada.....	78
Figura 5.3: Arquitetura SSD com 16 amostras de entrada.....	79

Figura 5.4: Arquitetura de SAD, SATD e SSD com compartilhamento de recursos lógicos	80
Figura 5.5: Operações para o cálculo de cada coeficiente antes do agrupamento	82
Figura 5.6: Operações para o cálculo de cada coeficiente após o agrupamento	83
Figura 5.7: Arquitetura da DCT 16x16 parcial	84
Figura 5.8: Arquitetura para decisão de modo intra-quadro	85
Figura 5.9: Arquitetura para cálculo de força de borda em partições.....	86
Figura 5.10: Arquitetura para cálculo de força de borda em sub-partições	86
Figura 5.11: Arquitetura para decisão de modo inter-quadros.....	87

LISTA DE TABELAS

Tabela 3.1: Modos de predição I4MB, baseada em (HUANG, 2005).....	37
Tabela 4.1: Ocorrência de modos intra-quadro e inter-quadros em quadros P de resolução CIF.....	46
Tabela 4.2: Comparação entre codificação com e sem utilização de modos intra-quadro em quadros P em vídeos de resolução CIF.	46
Tabela 4.3: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão intra-quadro 4x4 baseada em RDO e decisão intra-quadro 4x4 baseada em SAD.	48
Tabela 4.4: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão intra-quadro 16x16 baseada em RDO e decisão intra-quadro 16x16 baseada em SAD.	48
Tabela 4.5: Algoritmo utilizado para o cálculo da Hadamard 2-D 4x4 Direta.	49
Tabela 4.6: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão intra-quadro 4x4 baseada em RDO e decisão intra-quadro 4x4 baseada em SATD.	50
Tabela 4.7: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão intra-quadro 16x16 baseada em RDO e decisão intra-quadro 16x16 baseada em SATD.	50
Tabela 4.8: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão intra-quadro 4x4 baseada em RDO e decisão intra-quadro 4x4 baseada em SSD.	51
Tabela 4.9: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão intra-quadro 16x16 baseada em RDO e decisão intra-quadro 16x16 baseada em SSD.	51
Tabela 4.10: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão intra-quadro baseada em distorção e heterogeneidade do macrobloco original.	57
Tabela 4.11: Ocorrência de modos de codificação em quadros P de resolução CIF.	57
Tabela 4.12: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão de modo <i>SKIP</i> baseada em estacionaridade.	60
Tabela 4.13: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando detecção de sub-particionamento com base na heterogeneidade do macrobloco original.	63
Tabela 4.14: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão de tamanho de partição com base na detecção de borda.	66
Tabela 4.15: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão de tamanho de sub-partição com base na detecção de borda.	68
Tabela 4.16: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando todas as técnicas de decisão rápida propostas neste trabalho.	69
Tabela 4.17: Número de cálculos para a decisão de modo em um GOP IPPPPP.	75
Tabela 4.18: Comparação entre trabalhos relacionados e o conjunto de heurísticas proposto.	75
Tabela 5.1: Resultados das arquiteturas para cálculo de distorção em termos de área, frequência e potência implementadas como standard cells e em FPGA.	81

Tabela 5.2: Resultados das arquiteturas desenvolvidas em termos de recursos lógicos utilizados e frequência máxima para FPGAs Virtex 5 da Xilinx	88
--	----

RESUMO

O processo de compressão de vídeo é essencial para aplicações que utilizam vídeos digitais. O alto volume de informações contidas em um vídeo digital requer que um processo de compressão seja aplicado antes de este ser armazenado ou transmitido. O padrão H.264/AVC, estado-da-arte em termos de compressão de vídeo, introduziu um conjunto de ferramentas inéditas em relação a outros padrões, as quais possibilitam um ganho significativo em eficiência de compressão, diminuindo a taxa de bits sem perda na qualidade da imagem. Contudo, o preço deste ganho reside em um significativo aumento na complexidade de codificação.

No padrão H.264/AVC, a codificação pode acontecer de acordo com um dos treze modos de codificação intra-quadro ou de acordo com um dos oito tamanhos de bloco disponíveis para a predição inter-quadros. A escolha de melhor modo utilizada pelo software de referência do padrão (JM 17.1) é baseada em uma busca exaustiva pelo melhor modo, realizando a codificação repetidamente para todos os modos até que o menor custo em termos de taxa de bits e distorção seja encontrado. Esta decisão aumenta drasticamente o fluxo de codificação, muitas vezes impossibilitando a codificação de vídeos digitais em tempo real.

Neste contexto, a presente dissertação apresenta o estudo e o desenvolvimento de um conjunto de heurísticas que possibilitam a avaliação do melhor modo de codificação de bloco em um processo mais rápido que o usado pelo software de referência. Ao invés da realização do fluxo completo de codificação para todos os modos seguida por uma avaliação do melhor caso, propõe-se um conjunto de análises prévias que convergem para a decisão de apenas um modo de codificação. A redução atingida no número de repetições do processo de codificação foi de quarenta e sete vezes, ao custo de um aumento relativamente pequeno na taxa de bits. Quando comparada com outros trabalhos, a decisão rápida atingiu resultados expressivamente mais satisfatórios em termos de complexidade computacional, sem perda de qualidade ou aumento de taxa de bits significativo.

Foram desenvolvidas arquiteturas de hardware que implementam as heurísticas propostas. A arquitetura de decisão intra-quadro atingiu uma frequência máxima de 105 MHz, enquanto que a arquitetura de decisão inter-quadros apresentou uma frequência de 118 MHz para dispositivos FPGA Virtex 5 da Xilinx, sendo ambas capazes de processar vídeos de alta definição em tempo real.

Palavras-Chave: Compressão de Vídeo, H.264/AVC, Decisão de Modo de Codificação, Projeto de Hardware Digital.

Study and Development of Heuristics and Hardware Architectures for Fast Block Coding Mode Decision for the H.264/AVC Standard

ABSTRACT

The video compression process is essential in digital video applications, due to the extremely high data volume present in a digital video to be stored or transmitted through a physical link. H.264/AVC, the state-of-art video coding standard, introduces a set of novel features which lead to a significant gain in terms of compression efficiency, decreasing the bit-rate without image quality losses. However, the price of this gain resides at a high complexity increase.

In H.264/AVC, the encoding process can occur according to one of the thirteen intra-frame coding modes or according to one of the eight available inter-frames block sizes. In the reference software (JM 17.1), the choice of the best mode is performed through exhaustive executions of the whole encoding process. The mode which presents the lowest cost in terms of required bit-rate and image distortion is then chosen. This decision process increases significantly the encoding process, sometimes even forbidding its use in real time video coding applications.

Considering this context, this thesis presents a study and the development of a set of heuristics which allow the evaluation of the best coding mode in a process which is faster than the one used by the reference software. Instead of performing the whole encoding flow for all the possible modes followed by an evaluation of the best case, this work proposes a set of pre-analysis which converge to the selection of one encoding mode. The reduction achieved in the number of repetitions of the encoding process is of forty seven times, at the cost of a relatively small bit-rate increase. When compared to other works, the fast mode decision results are expressively more satisfactory in terms of computational complexity, with no image quality loss or significant bit-rate increase.

The hardware architectures which implement the proposed heuristics were also developed in this work. The architecture for intra-frame decision achieved a maximum frequency of 105 MHz, while the architecture for inter-frames decision presented a maximum frequency of 118 MHz for Virtex 5 FPGAs from Xilinx. They are both capable of processing high definition videos in real time.

Keywords: Video Coding, H.264/AVC, Coding Mode Decision, Digital Hardware Design, High Definition Video.

1 INTRODUÇÃO

A codificação de vídeos digitais tem se tornado, cada vez mais, uma área de pesquisa de grande importância na academia e na indústria. Isso se deve, principalmente, aos avanços tecnológicos que possibilitaram a utilização em massa deste tipo de vídeo através do uso de dispositivos eletrônicos móveis e fixos, como computadores pessoais e portáteis, televisores de alta definição, DVD *players*, celulares, câmeras digitais, entre outros.

Um vídeo digital é composto por uma série de imagens estáticas apresentadas em sequência a uma determinada taxa de amostragem. Quanto maior a taxa de amostragem, melhor a qualidade do vídeo e maior a quantidade de dados utilizados para representá-lo. O alto volume de dados presentes nos vídeos digitais tem impulsionado a pesquisa sobre técnicas de compressão a fim de facilitar a manipulação de tais vídeos pelos dispositivos existentes no mercado. Por exemplo, um vídeo de 10 minutos, com resolução de 720x480 pixels a 30 quadros por segundo (usado em televisão digital com definição normal – SDTV, assim como em DVDs) e utilizando 24 bits por pixel precisaria de 19 bilhões de bytes (19 GB) para ser armazenado (AGOSTINI, 2007). A taxa necessária para transmissão de um vídeo desta resolução seria próxima a 249 milhões de bits por segundo (249 Mbps). Como se pode perceber, tanto a taxa de transmissão quanto o armazenamento são intratáveis para as tecnologias atuais (AGOSTINI, 2007).

Grande parte dos dados que compõem o vídeo digital são redundantes e o seu volume pode ser diminuído através de técnicas de compressão. Isto significa que uma boa parte da enorme quantidade de dados necessários para representar o vídeo digitalizado é desnecessária. Assim, os padrões atuais de compressão de vídeo podem possibilitar que um vídeo alta resolução seja transmitido com a mesma taxa de transmissão de um vídeo de baixa resolução sem perda de qualidade da imagem.

O H.264/AVC é o padrão de compressão de vídeo considerado estado-da-arte, pois foi desenvolvido por um time de especialistas chamado JVT (*Joint Video Team*) (ITU, 2009) com o objetivo de dobrar a taxa de compressão em relação aos demais padrões existentes até então. Contudo, para que isto seja possível, o H.264/AVC apresenta uma grande complexidade computacional, impossibilitando, considerando a tecnologia atual, a sua utilização em *codecs* completos implementados em software que operem e possibilitem a transmissão do vídeo em tempo real (30 quadros de vídeo por segundo). Este aumento de complexidade tem impulsionado o meio acadêmico e a indústria a buscar soluções de hardware dedicadas à codificação e à decodificação de vídeo digital. Uma vasta quantidade de trabalhos que abordam soluções específicas de hardware para o *codec* H.264/AVC pode ser encontrada na literatura, como os trabalhos de Huang et al. (HUANG, 2005), Lin et al. (LIN, 2007) e Agostini et al. (AGOSTINI, 2007).

Para reduzir o volume de dados que representam um sinal de vídeo, os codificadores fazem uso abundante das técnicas de redução de redundância durante o processo de compressão. O codificador de vídeo do padrão H.264/AVC é composto por vários módulos que, direta ou indiretamente, têm como finalidade reduzir a redundância do sinal de vídeo: Estimção de Movimento (*Motion Estimation* – ME), Compensação de Movimento (*Motion Compensation* – MC), Predição Intra-Quadro, Transformadas Diretas (T) e Inversas (T^{-1}), Quantização Direta (Q) e Inversa (Q^{-1}), Codificação de Entropia e Filtro Redutor de Efeito de Bloco.

A função do módulo de predição intra-quadro é explorar e diminuir a redundância espacial. Assim, este módulo reutiliza informações já codificadas do quadro atual para realizar a predição do bloco que está sendo processado. A predição intra-quadro faz parte tanto do codificador quanto do decodificador do padrão H.264/AVC e realiza a codificação somente com base nas amostras espacialmente distribuídas em uma imagem. Já o módulo de predição inter-quadros é formado pela Estimção de Movimento (ME) e pela Compensação de Movimento (MC). Ele é responsável pela redução da redundância temporal, visto que opera realizando comparações entre blocos do quadro atual e dos quadros de referência previamente processados.

Devido à existência de diversos modos de codificação intra-quadro e inter-quadros definidos pelo padrão H.264/AVC, a tomada de decisão para selecionar o modo a ser utilizado é uma tarefa de grande complexidade que envolve a análise e a comparação da eficiência de codificação de todos os modos candidatos. Para atingir uma melhor eficiência de codificação, o H.264/AVC permite a utilização da técnica de Otimização Taxa-Distorção (RDO – *Rate-Distortion Optimization*) para encontrar o modo de codificação que apresenta o custo taxa-distorção (RD) mínimo (WIEGAND, 2003) (SULLIVAN, 1998). A taxa de bits pode ser obtida após a Codificação de Entropia, que constitui o último passo da codificação. Já a distorção da imagem é medida através de métricas de comparação que têm o objetivo de avaliar diferenças entre regiões de uma mesma imagem ou de imagens diferentes. As métricas de comparação podem realizar desde a simples diferença aritmética entre as imagens até cálculos mais complexos que procuram encontrar uma relação com o resultado visual da comparação (RICHARDSON, 2002). Como se pode perceber, a complexidade computacional do codificador H.264/AVC é dramaticamente aumentada por esta técnica de otimização, já que todos os modos de codificação devem ser aplicados e comparados em termos de taxa de bits e de distorção da imagem reconstruída para, então, a decisão do melhor modo de codificação ser tomada.

O objetivo deste trabalho é propor um conjunto de heurísticas e desenvolver as suas respectivas arquiteturas de hardware dedicadas à decisão rápida dos modos de codificação intra-quadro e inter-quadros, sendo capaz de codificar vídeos com resolução HD 1080p em tempo-real (30 quadros por segundo). Este trabalho está inserido no escopo de um projeto que tem como objetivo construir um *codec* H.264/AVC em hardware que tem como objetivo o processamento de vídeos de alta definição (HDTV) em tempo-real.

O texto foi organizado da seguinte forma. No capítulo 2, uma introdução sobre técnicas de compressão de vídeo e sobre o padrão H.264/AVC é apresentada. O capítulo 3 apresenta uma visão geral sobre as predições intra-quadro e inter-quadros. O capítulo 4 apresenta as heurísticas para decisão rápida de modo de codificação desenvolvidas, os resultados dos testes realizados em software para validá-las e comparações com trabalhos existentes na literatura. O capítulo 5 descreve as arquiteturas desenvolvidas,

os seus módulos e os resultados de síntese. O capítulo 6 conclui o trabalho e indica direções de trabalhos futuros.

2 COMPRESSÃO DE VÍDEO

Este capítulo apresenta conceitos básicos necessários para o entendimento dos capítulos seguintes e das arquiteturas desenvolvidas neste trabalho. Além disso, é apresentado o funcionamento geral do codificador H.264/AVC, com base nas informações disponibilizadas pela instituição que o definiu (ITU, 2007).

2.1 Resolução e Amostragem

Vídeos digitais são compostos por uma série de imagens estáticas chamadas quadros (ou *frames*) que, quando reproduzidas em sequência, podem induzir a percepção de movimento. Cada quadro é formado por uma matriz de blocos, os quais são formados por uma matriz de pontos chamados pixels. Quanto maior a quantidade de pixels utilizados para representar um quadro, maior é a resolução do vídeo e mais detalhada é a sua percepção. Os formatos mais comuns de resolução são o CIF (*Common Intermediate Format*), com 352x288 pixels, o QCIF (*Quarter Common Intermediate Format*), com 176x144 pixels, e o VGA (*Video Graphics Array*), com 640x480 pixels.

O número de quadros exibidos em um determinado período define a suavidade do movimento que os vídeos representam. Quanto maior o número de quadros exibidos, mais sutis são as diferenças entre os quadros e mais suaves são os movimentos representados. A este número de quadros, dá-se o nome de amostragem temporal, taxa de amostragem ou *frame rate*. Este valor é, geralmente, escolhido de forma que o olho humano não perceba a transição entre as imagens do vídeo. Taxas de captura usualmente utilizadas estão na faixa de 15 a 60 imagens capturadas por segundo (RICHARDSON, 2003).

2.2 Espaço de Cores

O sistema de visão do ser humano é composto por elementos chamados bastonetes e cones. Os primeiros são sensíveis à intensidade luminosa, enquanto que os cones são sensíveis às cores primárias. Assim, todas as cores são vistas como combinações das três cores primárias (vermelho, verde e azul). Milhares de cores distintas podem ser percebidas a partir de combinações de intensidades das cores primárias, mas apenas duas dúzias de tons de cinza são perceptíveis, os quais indicam variações na intensidade luminosa da imagem (GONZALES, 2003).

Os espaços de cores são usados para codificar digitalmente pixels dentro de quadros. Diversos espaços, tais como RGB, HSI e YCbCr (SHI, 1999) vêm sendo utilizados para representar imagens digitais. O RGB, um espaço de cores simples e bastante utilizado, divide os pixels em três componentes: R (*red* – vermelho), G (*green* – verde) e B (*blue* – azul). O YCbCr também é composto por três componentes. O componente de luminância, Y, representa o brilho da imagem, enquanto que os dois componentes de

crominância, Cb e Cr, representam o valor de desvio de uma cor a partir do cinza em direção ao azul e ao vermelho, respectivamente (BHASKARAN, 1997).

Como o sistema visual percebe melhor as mudanças na intensidade do brilho em uma imagem do que as mudanças na intensidade da cor, o sistema YCbCr foi desenvolvido buscando tratar a informação de cor completamente separada da informação de brilho, ao contrário do RGB. Assim, os dois tipos de informação podem ser manipulados de forma diferenciada pelos codificadores de imagens estáticas e de vídeos. No H.264/AVC, por exemplo, o sistema YCbCr permite representar os componentes de crominância usando menos amostras em relação aos componentes de luminância (RICHARDSON, 2002). Esta técnica é chamada de subamostragem de cores e é utilizada na maioria dos padrões de compressão de vídeo atuais.

2.3 Subamostragem de Cores em YCbCr

A eficiência da codificação aumenta significativamente quando a subamostragem de cor é utilizada, pois parte dos dados é simplesmente descartada sem causar qualquer impacto visual perceptível ao olho humano. Para tanto, existem diversas configurações da relação de subamostragem espacial das componentes de crominância em relação à componente de luminância. Os formatos mais comuns são o 4:4:4, o 4:2:2 e o 4:2:0.

No formato 4:4:4, para cada quatro amostras de luminância (Y), existem quatro amostras de crominância azul (Cb) e quatro amostras de crominância vermelha (Cr). Neste caso, nenhuma subamostragem é aplicada, pois a resolução dos três componentes é a mesma. No segundo caso, 4:2:2, existem duas amostras de crominância azul e duas amostras de crominância vermelha para cada quatro amostras de luminância. Já no formato 4:2:0, o mais usado no H.264/AVC, para cada quatro amostras de luminância, existe apenas uma amostra de cada tipo de crominância. O formato deveria, portanto, ser chamado 4:1:1. Contudo, por motivos históricos, esta não é a nomenclatura utilizada (RICHARDSON, 2003).

Considerando um vídeo codificado no formato 4:2:0, já que cada componente de crominância possui apenas um quarto das amostras do componente de luminância, pode-se dizer que a taxa de compressão é de 50% em relação a um vídeo no formato 4:4:4, utilizando apenas a subamostragem. Neste trabalho, a subamostragem utilizada é de formato 4:2:0, pois é o que apresenta maiores ganhos em compressão e, ao mesmo tempo, adéqua-se bem à capacidade de percepção do sistema visual humano. Além disso, esta sub-amostragem é utilizada na maioria dos perfis do padrão H.264/AVC (ITU, 2007).

2.4 Redundância de Dados

De forma geral, a codificação de vídeo tem o objetivo de reduzir as redundâncias existentes sob diferentes formas em um vídeo digital. São considerados redundantes todos os dados que não possuem informação relevante para a representação de uma imagem. Existem três tipos de redundância que podem ser exploradas na compressão de vídeos: redundância espacial, redundância temporal e redundância entrópica.

As técnicas de compressão podem reduzir as redundâncias gerando perda de informação ou não. Em geral, as técnicas de compressão com perda de informação exploram as limitações do sistema visual humano, selecionando apenas as informações que são mais perceptíveis, de forma que a imagem reconstruída (após a decodificação) possua uma diferença visual imperceptível ou pouco perceptível ao olho humano quando comparada com a imagem original.

2.4.1 Redundância Espacial

Também chamada de “redundância intra-quadro” ou “redundância inter-pixel”, é decorrente da correlação entre pixels existentes dentro de um mesmo quadro (GHANBARI, 2003). Se a correlação é percebida no domínio espacial, ou seja, se os valores de alguns pixels em um mesmo quadro são semelhantes, a redundância pode ser reduzida através da predição intra-quadro, presente na maioria dos padrões de codificação de vídeo atuais e que realiza predição sobre as amostras espacialmente distribuídas do quadro, enviando somente o resíduo da predição. Esta correlação é visível no domínio espacial, pois pixels vizinhos em um quadro tendem a ter valores semelhantes. No domínio das frequências, a redundância espacial é reduzida através de uma operação chamada de quantização (AGOSTINI, 2007), uma divisão inteira dos coeficientes gerados em uma etapa de transformação de resíduos, conforme explicado na próxima seção.

2.4.2 Redundância Temporal

Também chamada de “redundância inter-quadros”, é decorrente da correlação entre quadros temporalmente vizinhos (GHANBARI, 2003), isto é, quadros temporalmente próximos em um vídeo. Vários casos de redundância temporal podem ocorrer. Por exemplo, os valores de um conjunto de pixels podem não mudar de um quadro para outro ou podem variar pouco, como no caso de um fundo de uma imagem (parede, céu, campo, etc.). Além disso, o conjunto de pixels de determinado quadro pode reaparecer em outro quadro simplesmente deslocado em relação à posição anterior, como, por exemplo, o movimento de um objeto em uma cena. Uma redução eficiente da redundância temporal leva a altíssimos níveis de compressão de vídeo e, portanto, todos os padrões atuais procuram explorá-la.

2.4.3 Redundância Entrópica

A entropia é um conceito que relaciona a quantidade média de informação transmitida por símbolo do vídeo (SHI, 1999). Quanto maior a probabilidade de um símbolo ocorrer, menor é a quantidade de informação nova transmitida por este símbolo. Assim, os codificadores buscam transmitir o máximo de informação possível por símbolo codificado, ou seja, representar mais informações com um número menor de bits, sem inserir perdas de informação. A codificação de entropia atua, portanto, na representação dos dados do vídeo para reduzir sua redundância estatística através de diferentes técnicas e algoritmos de compressão sem perdas.

2.5 Métricas de Distorção

As métricas de distorção, ou critérios de comparação, são medidas utilizadas para medir a qualidade do vídeo codificado. Este é um parâmetro complexo de ser definido e avaliado, pois pode ser tanto subjetivo quanto objetivo. Os critérios subjetivos são os mais difíceis de medir e não serão abordados nesta dissertação, pois não são utilizados no *codec* do padrão H.264/AVC. Já os critérios objetivos se baseiam na comparação dos pixels da imagem original com os pixels da imagem após a decodificação (chamada de imagem reconstruída). Esta comparação é realizada quadro a quadro.

O critério mais conhecido e aceito pela comunidade é o PSNR (*Peak Signal-to-Noise Ratio*) (GHANBARI, 2003), definido pela equação (1).

$$PSNR_{dB} = 20 \cdot \log_{10} \left(\frac{MAX}{\sqrt{MSE}} \right) \quad (1)$$

Em (1), **MAX** é o valor máximo que uma amostra de luminância pode atingir ($2^n - 1$, sendo n o número de bits para representar cada amostra) e **MSE** é o erro médio quadrático (*Mean-Squared Error*), sendo definido em (2), onde m e n são as dimensões da imagem em *pixels* e O e R são a imagem original e reconstruída, respectivamente.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (R_{i,j} - O_{i,j})^2 \quad (2)$$

O MSE por si só é uma medida de similaridade entre dois quadros. Assim como esta medida, existem outras que têm a mesma finalidade, mas que realizam a função com maior ou menor precisão e complexidade, como o SAD (*Sum of Absolute Differences*) e o SSD (*Sum of Squared Differences*). Outras métricas de distorção serão abordadas no capítulo 4 desta dissertação.

2.6 O Padrão H.264/AVC

O H.264/AVC, foco deste trabalho, é o mais novo padrão de compressão de vídeo, desenvolvido pelo JVT (*Joint Video Team*), um grupo de especialistas do VCEG (*Video Coding Experts Group*) da ITU (ITU, 2009) e do MPEG (*Moving Picture Experts Group*) da ISO/IEC (ISO, 2009).

Diversos padrões serviram de base para o desenvolvimento do H.264/AVC, como o H.262 (ITU-T, 1994) e o H.263 (ITU-T, 2000). O H.264 (ITU-T, 2003) teve o seu *draft* final aprovado em outubro de 2003 (SULLIVAN, 2004). O nome do padrão adotado pela ITU-T foi H.264, enquanto que pela ISO/IEC, o nome utilizado foi MPEG-4 parte 10 – AVC (*Advanced Video Coding*). Assim, a comunidade científica passou a utilizar a denominação H.264/AVC para se referir ao padrão de forma equilibrada entre os nomes dados pela ITU-T e pela ISO/IEC.

O principal objetivo do desenvolvimento do padrão H.264/AVC foi aumentar a eficiência de codificação em relação aos padrões anteriores. Para isso, diversos recursos foram introduzidos no processo de codificação, tais como o uso de tamanho de blocos menores de 8x8 pixels, a predição intra-quadro realizada no domínio espacial, as transformadas Hadamard e DCT, a utilização de múltiplos quadros de referência para predição inter-quadros, a utilização de filtro redutor de efeito de bloco, entre outros. A utilização de tais ferramentas possibilitou um ganho significativo em compressão, ao preço de um aumento na complexidade do codificador.

Novas versões do padrão H.264/AVC foram aprovadas e publicadas nos últimos anos, após a primeira versão de 2003. Uma nova versão, chamada *Fidelity Range Extensions* (FRExt) (ITU-T, 2004), que suporta aplicações de vídeo com qualidade profissional foi proposta em 2004 e anexada como uma extensão do H.264/AVC em 2005 (ITU, 2005). Em 2007, o grupo JVT aprovou e publicou o padrão SVC (*Scalable Video Coding*), que dá suporte a vídeos escaláveis (ITU, 2007). Em 2009, o JVT especificou o MVC (*Multiview Video Coding*), uma extensão que possibilita a construção de *bitstreams* que representam mais de uma vista de uma cena de vídeo. Um novo padrão sucessor do H.264, o H.265 ou HVC (*High-efficiency Video Coding*) está atualmente em desenvolvimento e deve ter a sua primeira versão publicada em julho de 2012 (JCT-VC, 2010).

2.6.1 Terminologia

O padrão H.264/AVC utiliza o espaço de cores YCbCr apresentado na seção 2.2 e as sub-amostragens apresentadas na seção 2.3. O formato de sub-amostragem utilizado

depende do perfil de codificação utilizado. Uma imagem codificada pode ser formada a partir de um campo ou quadro, no caso de vídeos entrelaçados, ou a partir de um quadro, quando o vídeo codificado é progressivo (RICHARDSON, 2003). Neste trabalho, apenas vídeos progressivos são considerados e tratados. Assim, o significado de quadro e imagem será sempre o mesmo.

Um quadro é uma matriz de pixels, os quais são também chamados de amostras. Os pixels são agrupados em regiões chamadas de macrobloco. Cada macrobloco é formado por 16x16 amostras de luminância associadas a outras duas regiões de crominância, que possuem dimensões dependentes do perfil de codificação utilizado, conforme será apresentado na próxima seção. Os quadros também possuem informações de controle que indicam o tipo de codificação utilizada, o início de cada macrobloco, o tipo do macrobloco, entre outras.

Cada quadro é formado por um ou mais grupos de macroblocos organizados na ordem *raster* dentro de um quadro (ITU, 2007). Cada grupo de macroblocos é chamado de *slice* e representa uma entidade auto-contida para fins de codificação (PURI, 2004). Neste trabalho, para fins de simplificação, cada quadro é codificado utilizando somente um *slice*. O padrão define cinco tipos de *slice*: I (intra), P (preditivo), B (bi-preditivo), SI (*Switch-I*) ou SP (*Switch-P*). Os *slices* SI e SP são semelhantes aos *slices* I e P, porém usados em outras situações que não se aplicam a esta dissertação e, portanto, não serão abordados. Para previsões que usam quadros anteriormente processados como referência, como é o caso de *slices* P e B, estes quadros de referência ficam armazenados em duas listas, chamadas de lista 0 e lista 1 (ITU, 2007).

Slices do tipo I possuem apenas macroblocos do tipo I, que são codificados utilizando a predição intra-quadro a partir das amostras do *slice* atual. Esta codificação pode ser realizada sobre o macrobloco completo ou para cada bloco que o compõe, como será explicado com mais detalhes no próximo capítulo. Um quadro que contém apenas *slices* do tipo I (formados por macroblocos I) é o primeiro a ocorrer na sequência de decodificação. Um tipo especial de quadro, denominado IDR (*Instantaneous Decoder Refresh*), zera as listas 0 e 1 de referência, ou seja, todos os quadros posteriores a ele não podem fazer referências a quadros anteriores a ele, na ordem de exibição. Assim, este é o primeiro tipo de quadro a ser decodificado. Um tipo especial de macrobloco, chamado de I_PCM é utilizado sempre que os valores das amostras devem ser enviados diretamente, sem predição e posterior transformação. Em alguns casos raros, a utilização deste tipo de macrobloco pode levar a uma eficiência de codificação maior que os demais modos de codificação (RICHARDSON, 2003).

Slices do tipo P podem conter macroblocos do tipo I ou P. Macroblocos do tipo P são codificados através da predição inter-quadros a partir de quadros de referência já codificados. Os macroblocos podem ser divididos em partições de tamanho 16x16, 16x8, 8x16 ou 8x8 (que pode ser subdividido em 8x4, 4x8 ou 4x4). Cada partição é codificada utilizando, como referência, um quadro da lista 0. Caso exista uma sub-partição, esta deve ser codificada utilizando o mesmo quadro da lista 0 utilizado para codificar a partição de macrobloco.

Slices do tipo B podem conter macroblocos do tipo I, P ou B. Macroblocos do tipo B também são codificados com predição inter-quadros. Contudo, ao invés de utilizar apenas o quadro de uma lista de quadros de referência, este tipo de macrobloco também permite a utilização de dois quadros, um na lista 0 e um na lista 1.

O H.264/AVC apresenta ainda um tipo de macrobloco chamando *SKIP*, um tipo especial de macrobloco no qual nenhuma informação é transmitida no *bitstream*, exceto a sinalização de que o macrobloco é do tipo *SKIP*. Um macrobloco deste tipo pode ser utilizado em slices do tipo P ou do tipo B. O decodificador calcula um vetor de movimento para o macrobloco *SKIP* e reconstrói o macrobloco usando compensação de movimento usando o primeiro quadro de referência da lista 0 (RICHARDSON, 2003).

2.6.2 Perfis e Níveis

O padrão H.264/AVC suporta diferentes grupos de funções através de perfis que especificam o comportamento do codificador e do decodificador. Na primeira versão do H.264/AVC (ITU-T, 2003), os perfis *Baseline*, *Main* e *Extended* foram introduzidos.

O *Baseline* é um perfil voltado para aplicações móveis, como videoconferência e vídeo sem fio. Neste perfil, são suportadas as previsões intra-quadro e inter-quadros através do uso de *slices* do tipo I e P. A codificação de entropia é realizada através de códigos de comprimento variável adaptativos ao contexto (CAVLC). O perfil *Main* inclui o suporte para vídeo entrelaçado, o suporte à codificação inter-quadros utilizando *slices* do tipo B e utilizando previsão ponderada e o suporte à codificação de entropia utilizando codificação aritmética adaptativa ao contexto (CABAC). O perfil *Extended* é mais voltado para aplicações em *streaming* de vídeo e não suporta vídeo entrelaçado e CABAC, mas agrega modos para habilitar uma troca eficiente entre *bitstreams* codificados (através de *slices* do tipo SP e SI) e melhora a robustez a erros de transmissão (através do particionamento de dados).

A relação entre os elementos de luminância e crominância é a mesma nos três perfis definidos na primeira versão do padrão: 4:2:0. Isto significa que para cada 4 elementos de luminância, 1 elemento de crominância azul e 1 elemento de crominância vermelha são utilizados, gerando blocos de crominância com metade da resolução horizontal e vertical dos blocos de luminância. Em outras palavras, cada macrobloco é formado por 16x16 amostras de luminância (Y) e 8x8 amostras de cada crominância (Cb e Cr). Isso implica em uma subamostragem, com conseqüente diminuição do *bitstream* do vídeo, já na sua versão original. Nos três perfis do padrão, são utilizados 8 bits para representar cada amostra.

Quatro novos perfis foram desenvolvidos para dar suporte a vídeos profissionais de qualidade mais elevada, já que os perfis *Baseline*, *Main* e *Extended* foram criados para entretenimento. Os novos perfis foram chamados de FRExt (*Fidelity Range Extensions*), ou extensões para alcance de fidelidade, ou ainda, simplesmente, perfis *High* (SULLIVAN, 2004).

Os perfis *High* implementam algumas características que são inovadoras em relação aos perfis originais: suportam um tamanho de bloco adaptativo para a transformada (4x4 ou 8x8), matrizes de quantização baseadas em percepção e uma representação sem perdas de regiões específicas do conteúdo do vídeo (SULLIVAN, 2004). O primeiro perfil do FRExt, o *High* (HP), inclui vídeo com 8 bits por amostra e com relação de cor 4:2:0. O perfil *High* 10 (Hi10P) suporta todas ferramentas do perfil *High* e ainda vídeos com 8, 9 e 10 bits por amostra, também com uma relação de cor 4:2:0. O perfil *High* 4:2:2 (H422P) suporta todas ferramentas do *High* 10 e inclui ainda suporte à subamostragem de cor 4:2:2 (ITU, 2007). Finalmente, o perfil *High* 4:4:4 (H444P) dá suporte à relação de cor 4:4:4 (isto é, sem qualquer subamostragem), suporta vídeos de até 12 bits por amostra e suporta codificações sem perdas em regiões do vídeo. Parte das ferramentas suportadas pelos diferentes perfis está apresentada na Figura 2.1.

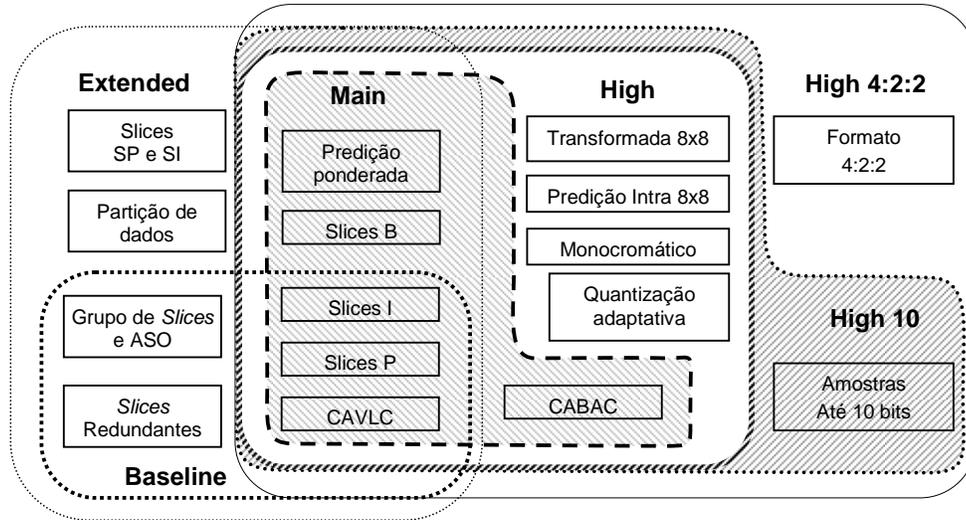


Figura 2.1: Perfis do H.264/AVC (ROSA, 2009).

Além da divisão em diversos perfis, o padrão H.264/AVC também define 16 diferentes níveis em função da taxa de processamento e da quantidade de memória necessária para cada implementação do codificador ou decodificador. Com a definição do nível utilizado, é possível deduzir o número máximo de quadros de referência e a máxima taxa de bits que podem ser utilizados (SULLIVAN, 2004).

2.7 Núcleo do Codec H.264/AVC

2.7.1 Funcionamento Básico

O núcleo do codificador H.264/AVC é composto pelos módulos de predição intra-quadro, predição inter-quadros, transformadas diretas e inversas (T e T^{-1}), quantização direta e inversa (Q e Q^{-1}), filtro redutor do efeito de bloco e codificação de entropia, conforme ilustra a Figura 2.2. Todos esses módulos aplicam técnicas que visam, direta ou indiretamente, reduzir as redundâncias presentes nos vídeos digitais.

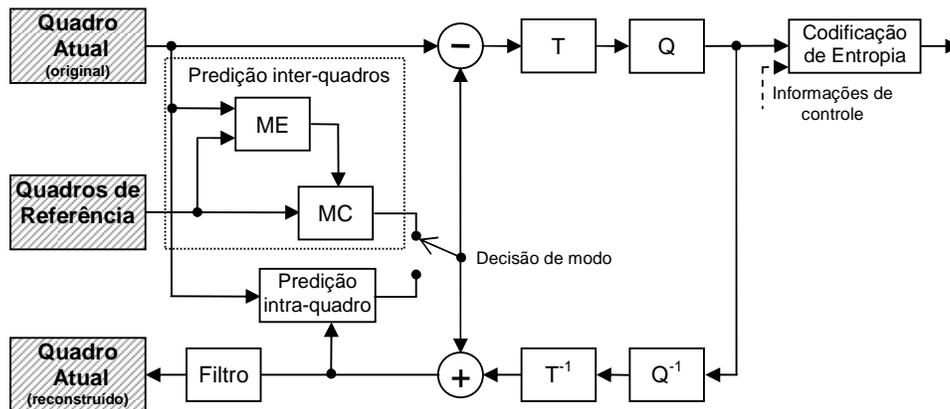


Figura 2.2: Diagrama em blocos de um codificador H.264/AVC (AGOSTINI, 2007).

Em resumo, o codificador H.264/AVC funciona da seguinte forma: cada macrobloco do quadro atual (original) é predito a partir dos macroblocos vizinhos processados anteriormente, através da predição intra-quadro, ou de um macrobloco contido em um

quadro de referência processado anteriormente, através da predição inter-quadros. A escolha de uma das duas predições é representada pela chave seletora de decisão de modo na Figura 2.2. O resultado da predição é subtraído do macrobloco original, gerando um macrobloco de resíduos, que são processados pelos módulos de transformadas (módulo T) e quantização (módulo Q), gerando os coeficientes quantizados. Os coeficientes quantizados, bem como as demais informações utilizadas no processo de decodificação (por exemplo, modos de predição e vetores de movimento utilizados) são enviados para a etapa de codificação de entropia, que gera uma sequência de bits que forma o macrobloco codificado. O conjunto de todos os macroblocos codificados e organizados em diferentes quadros forma o vídeo codificado, representado por uma sequência de bits denominada *bitstream*.

A predição inter-quadros é formada pela estimação de movimento (ME – *Motion Estimation*) e pela compensação de movimento (MC – *Motion Compensation*). Este módulo é responsável por reduzir a redundância temporal através da comparação dos blocos do quadro atual com os blocos existentes no quadro de referência. Quando o bloco do quadro de referência com maior semelhança com o bloco do quadro atual é encontrado, ele é escolhido para a predição e um vetor de movimento é gerado para indicar a sua posição no quadro de referência em relação à posição do bloco atual. Este módulo gera blocos do tipo P para *slices* do tipo P.

A predição intra-quadro reduz a redundância espacial, pois trabalha sobre comparações entre blocos pertencentes apenas ao quadro atual. Este módulo gera blocos do tipo I para *slices* do tipo I ou do tipo P. Tanto a predição inter-quadros quanto a predição intra-quadro serão explicadas com mais detalhes no capítulo 3 deste trabalho.

Após o bloco ser codificado através da predição intra-quadro ou inter-quadros, uma subtração entre os pixels do bloco original e do bloco predito é realizada para que a diferença causada pela predição seja obtida para ser codificada, já que esta diferença não pode ser desprezada. A esta diferença dá-se o nome de resíduo. O resíduo é enviado para os módulos de transformada (T) e quantização (Q), responsáveis por diminuir a redundância espacial no domínio das frequências. A codificação de entropia opera, por fim, sobre os resultados da quantização e reduz a redundância entrópica, que está relacionada com a forma como os dados estão codificados. Finalmente, é aplicado um filtro redutor de efeito de bloco para aumentar a qualidade subjetiva da imagem (AGOSTINI, 2007). Como não são foco deste trabalho, os módulos de transformada, quantização, codificação de entropia não serão explicados em mais detalhes neste trabalho. Mais informações sobre estes módulos podem ser encontradas nos trabalhos de Agostini et al. (2007) e Richardson (2003).

Como a codificação do vídeo gera perdas, o quadro codificado é diferente do quadro original após a decodificação (RICHARDSON, 2003). Assim, para que o cálculo de resíduos seja exatamente o mesmo no codificador e no decodificador, o codificador precisa realizar a reconstrução do quadro para que este sirva como referência para os próximos quadros a serem codificados, no caso da predição inter-quadros. Para a predição intra-quadro, o mesmo caso se aplica, pois as amostras dos blocos vizinhos já codificados devem ser utilizadas como referência para os próximos blocos. Em outras palavras, após a codificação, o codificador descarta as amostras originais e armazena, no seu lugar, as amostras reconstruídas.

A tarefa de reconstrução é realizada no codificador pelos módulos de quantização inversa, transformada inversa e filtro redutor de efeito de bloco. A codificação e a decodificação de entropia não são necessárias para a reconstrução do quadro porque a

sua utilização não gera perdas. Após as etapas de quantização inversa e transformada inversa, os resíduos reconstruídos são obtidos e podem ser somados ao resultado da predição intra-quadro ou inter-quadros. O filtro redutor de efeitos de bloco, opcional em outros padrões, mas de uso mandatório no H.264/AVC, é utilizado no caminho de reconstrução do quadro. Novamente, mais informações sobre este módulos podem ser encontradas nos trabalhos de Agostini et al. (2007) e Richardson (2003).

A Figura 2.3 apresenta o diagrama em blocos de um decodificador H.264/AVC, que é composto pelos blocos de predição intra-quadro, predição inter-quadros, transformadas e quantização inversas (T^{-1} e Q^{-1}), filtro redutor de efeitos de bloco e decodificação de entropia.

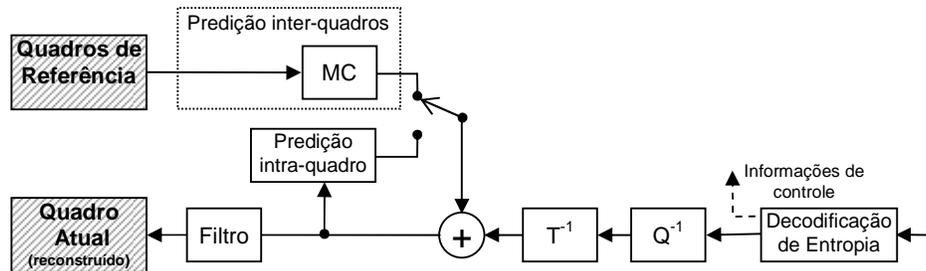


Figura 2.3: Diagrama em blocos de um decodificador H.264/AVC (AGOSTINI, 2007).

O funcionamento de um decodificador de vídeo H.264/AVC é similar ao caminho de reconstrução do codificador. O *bitstream* de entrada é decodificado pelo decodificador de entropia e os resíduos quantizados são processados pelo módulo Q^{-1} (quantização inversa) e, logo depois, pelo módulo T^{-1} (transformadas inversas). No *bitstream* também estão contidas informações de controle essenciais para predição e reconstrução do vídeo, tais como o tipo do macrobloco, modos de predição e vetores de movimento. A predição sinalizada no *bitstream* é realizada pelo módulo MC (da predição inter-quadros) ou de predição intra-quadro e o macrobloco predito é somado com os resíduos resultantes da etapa T^{-1} . A imagem reconstruída é posteriormente filtrada para ser exibida e armazenada para futuras referências.

2.7.2 Decisão de Modo

Para atingir uma codificação com uma boa relação entre a qualidade da imagem codificada e a eficiência da compressão, o codificador H.264/AVC deve realizar uma escolha correta entre as diversas formas de codificação que o padrão permite, tais como o tipo de predição (intra-quadro ou inter-quadros), o tamanho do bloco utilizado (16x16, 16x8, 8x16, 8x8, 8x4, 4x8 ou 4x4), se o modo *skip* será usado ou não, se será usada bi-predição, entre outras. Como as sequências de vídeo possuem, em geral, grande variação de conteúdo e de movimento, as decisões de modo e de tamanho de bloco são de extrema importância e devem ser bem tomadas, caso contrário os benefícios proporcionados pelo H.264/AVC podem ser reduzidos (PURI, 2004).

Embora não tenha sido normatizada pelo padrão H.264/AVC a técnica de Otimização Taxa-Distorção (RDO – *Rate-Distortion Optimization*) foi proposta com a finalidade de gerar a codificação com melhor relação entre a eficiência de compressão (taxa de bits) e a qualidade do vídeo codificado (distorção) (WIEGAND, 2003) (SULLIVAN, 1998). Embora este tipo de técnica gere resultados ótimos, todas as possíveis combinações de modos de codificação precisam ser calculadas e comparadas, aumentando drasticamente a complexidade do processo de codificação.

Em (3), \mathbf{D} representa o nível de distorção da imagem codificada e reconstruída, \mathbf{R} representa a taxa de bits e \mathbf{J} representa o custo taxa-distorção. O parâmetro λ é chamado de multiplicador de Lagrange e relaciona a distorção com a taxa de bits.

$$J = D + \lambda \cdot R \quad (3)$$

O objetivo da técnica de RDO é encontrar a combinação de modos de codificação que gera o menor custo \mathbf{J} , o que representa uma busca pela melhor relação entre a taxa de bits (quantidade de bits utilizados para a codificação) e a qualidade do vídeo (distorção expressa em decibéis) codificado.

Para tomar uma decisão ótima através da técnica de RDO, devem-se gerar os valores exatos dos custos \mathbf{J} , medindo os valores corretos de \mathbf{D} e \mathbf{R} . O enorme número de combinações de modos de codificação existentes no padrão H.264/AVC associado ao limite computacional dos sistemas atuais torna impraticável a avaliação exata dos custos \mathbf{J} , exigindo uma aproximação do seu valor com cálculos mais simples que permitam a codificação de vídeos em tempo-real. Em resumo, a complexidade do processo de escolha do modo de predição utilizado pode exigir recursos computacionais além dos disponíveis. Assim, para fins de aproximação, o valor de \mathbf{D} não é expresso em PSNR, mas em uma métrica de distorção mais simples.

Na técnica de RDO, a taxa de bits deve ser calculada através do número de bits utilizados para codificar as informações de controle (modos de predição, tipos de particionamento, entre outros) e resíduos após a codificação de entropia. Isto significa que para encontrar o custo \mathbf{J} mínimo, os resíduos de todos os modos de codificação devem ser transformados, quantizados e processados pela codificação de entropia, o que aumenta significativamente a complexidade do codificador.

Como se pode perceber, a complexidade de decisão torna a codificação em tempo-real um grande desafio, o que estimula a busca por heurísticas para tomada de decisão prévia no processo de codificação. A decisão rápida de modo de codificação é o foco deste trabalho e será amplamente discutida nos capítulos 4 e 5 desta dissertação.

Este capítulo apresentou conceitos básicos de compressão de vídeo, o padrão H.264/AVC e o núcleo do seu *codec*. O próximo capítulo apresenta os módulos de predição intra-quadro e inter-quadros com um grau de detalhamento maior, pois a sua compreensão é essencial para o desenvolvimento deste trabalho.

3 PREDIÇÕES INTRA-QUADRO E INTER-QUADROS

Os grandes índices de compressão apresentados pelo padrão H.264/AVC são possíveis graças à codificação preditiva realizada pelos módulos de predição intra-quadro e inter-quadros.

A predição intra-quadro, aplicada no domínio espacial, é uma inovação apresentada pelo padrão H.264/AVC. Ela é aplicada sobre todas as amostras (pixels) que compõem um macrobloco com o objetivo de diminuir a redundância espacial contida nos quadros. A predição inter-quadros é formada pelos módulos de estimação de movimento (ME) e compensação de movimento (MC). Ela é aplicada sobre as amostras com o intuito de reduzir a redundância espacial presente entre os quadros que compõem o vídeo digital. Após a predição intra-quadro ou inter-quadros ser realizada, o seu resultado é subtraído do macrobloco original, gerando apenas um resíduo que é processado pelas etapas seguintes do codificador.

Nas primeiras seções deste capítulo são apresentados os conceitos sobre a predição intra-quadro do padrão H.264/AVC, focando especificamente nos modos de codificação possíveis. Nas seções finais, a predição inter-quadros é detalhada, também com foco nos modos de codificação disponíveis. Informações mais aprofundadas sobre o funcionamento das predições intra-quadro e inter-quadros podem ser obtidas em (DINIZ, 2009) e (PORTO, 2008), respectivamente.

3.1 A Predição Intra-Quadro

A predição intra-quadro do padrão H.264/AVC é aplicada a todos os macroblocos do quadro atual, tanto para os componentes de luminância quanto para os de crominância. A predição de um macrobloco é realizada a partir de amostras já reconstruídas localizadas contidas dentro de um mesmo *slice* do quadro. Amostras reconstruídas são aquelas que já foram processadas pelo *loop* de reconstrução $T/Q/T^{-1}/Q^{-1}$ (como explicado na seção 2.7.1 do capítulo 2), mas que ainda não foram processadas pelo Filtro Redutor de Efeito de Bloco.

Dois tipos de macroblocos podem ser escolhidos quando a predição intra-quadro é utilizada: intra-quadro 4x4 (I4MB) e intra-quadro 16x16 (I16MB). No primeiro caso, as 16x16 amostras que compõem um macrobloco de luminância são divididas em blocos de 4x4 amostras que são preditos separadamente. No segundo caso, o macrobloco é predito por inteiro. A Figura 3.1 mostra as divisões de amostras de luminância nos dois tipos de macrobloco intra-quadro. A Figura 3.1 também apresenta a ordem de processamento dos blocos 4x4 que compõem um macrobloco codificado com predição intra-quadro 4x4. A ordem é a mesma utilizada pelas etapas de transformadas e quantização. Os componentes de crominância do macrobloco são codificados de acordo com apenas um tamanho (8x8) quando a sub-amostragem 4:2:0 é utilizada, não apresentando sub-divisões.

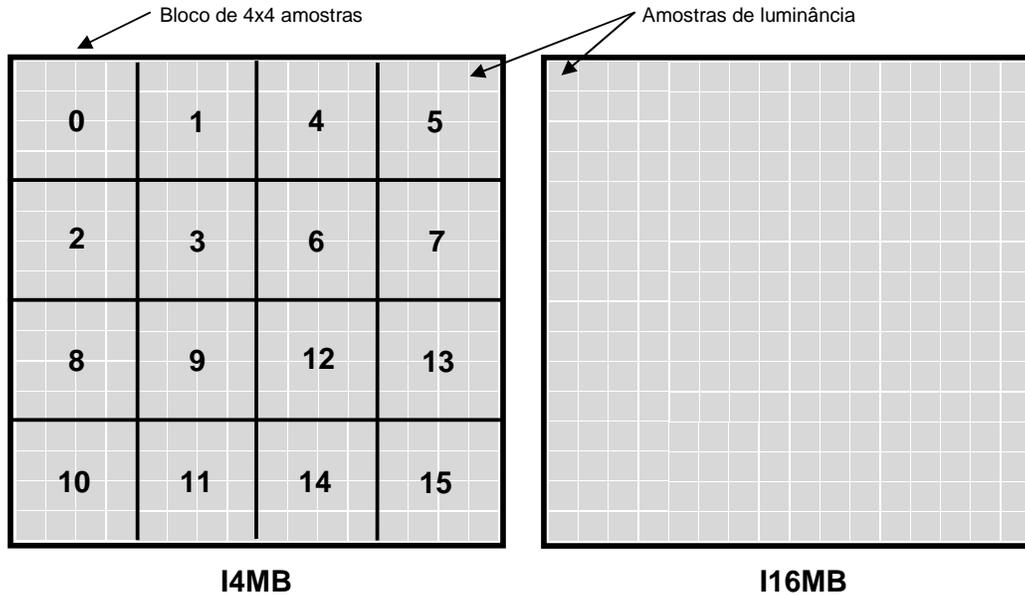


Figura 3.1: Tipos de macrobloco Intra no padrão H.264/AVC (DINIZ, 2009).

Cada tipo de macrobloco pode ser codificado através de um conjunto de modos de predição definidos pelo padrão. Macroblocos I4MB podem ser codificados através de nove modos de predição e macroblocos I16MB utilizam quatro modos. Os modos permitem que o bloco predito seja gerado através da cópia dos pixels vizinhos em diferentes sentidos. Dependendo do bloco sendo predito e dos seus pixels vizinhos, um tipo de cópia pode apresentar melhores resultados que outros tipos em termos de semelhança em relação à imagem original. Quanto maior a semelhança entre o bloco predito e o bloco original, menor é a amplitude dos resíduos e, portanto, menor é a quantidade de bits utilizados para representá-los. O tipo de macrobloco também influencia na quantidade de resíduos gerados, já que as predições realizadas sobre um bloco maior (I16MB) tende a apresentar uma quantidade maior de resíduos, especialmente em áreas muito heterogêneas da imagem. Esta questão será abordada com maiores detalhes no capítulo 4 desta dissertação.

A Figura 3.2 apresenta um quadro do vídeo *Foreman*, com resolução QCIF (176x144), que foi codificado utilizando apenas a predição intra-quadro 4x4. A imagem da esquerda mostra o quadro original e a imagem da direita mostra o resíduo resultante da predição (isto é, a diferença entre o quadro original e o quadro predito). Na imagem do resíduo (direita), as regiões em cinza representam diferenças iguais a zero entre as amostras originais e as amostras preditas. As regiões em tons de cinza mais claro ou mais escuro representam diferenças menores ou maiores que zero, respectivamente.

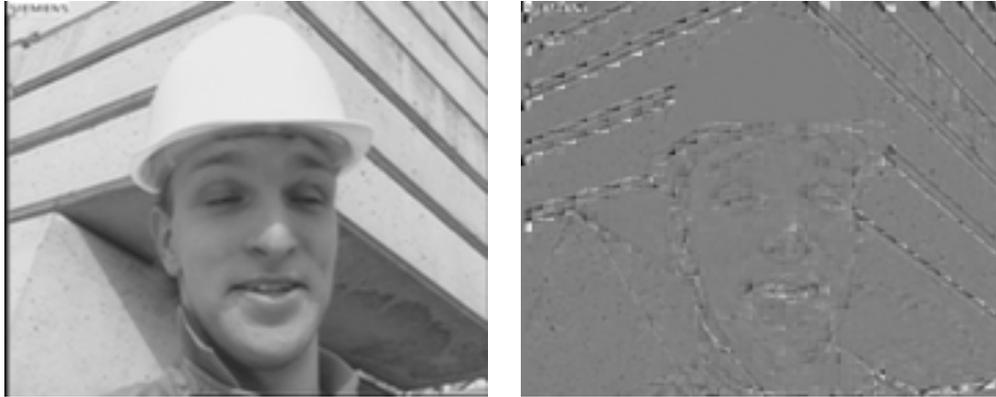


Figura 3.2: Resultado da predição intra-quadro 4x4. Quadro original (esquerda); Resíduo da predição intra-quadro 4x4 (direita) (DINIZ, 2009).

3.1.1 Predição Intra-Quadro 4x4

Um macrobloco deve ser predito utilizando partições de 4x4 pixels (I4MB) quando a imagem possui uma grande granularidade de detalhes. A predição intra-quadro 4x4 pode ser realizada de acordo com nove modos de predição. Cada um dos modos estabelece um sentido de cópia das amostras vizinhas, o qual pode variar entre horizontal e vertical em diferentes ângulos. A Figura 3.3 ilustra as amostras envolvidas na predição de um bloco 4x4. As amostras identificadas de A a M representam os pixels vizinhos utilizados para a predição. As amostras em cinza (numeradas) compõem o bloco 4x4 predito.

M	A	B	C	D	E	F	G	H
I	0,0	0,1	0,2	0,3				
J	1,0	1,1	1,2	1,3				
K	2,0	2,1	2,2	2,3				
L	3,0	3,1	3,2	3,3				

Figura 3.3: Identificação das amostras para a predição de um bloco 4x4.

A Figura 3.4 ilustra os nove modos possíveis da predição intra-quadro 4x4. Os modos, numerados de 0 a 8, representam a direção de cópia das amostras de referência para formação do bloco predito. O modo 0 faz uma simples cópia das amostras que compõem a borda superior (A, B, C e D) para todas as posições do bloco. Da mesma forma, o modo 1 faz a cópia das amostras que compõem a borda lateral esquerda (I, J, K e L). O modo 2 (DC) calcula a média aritmética entre todas as amostras de referência e copia a média para todas as posições do bloco. Os modos 3, 4, 5, 6, 7 e 8 realizam uma interpolação linear das amostras vizinhas conforme a direção das setas apresentadas na Figura 3.4 e copia esses valores para o bloco predito (DINIZ, 2009).

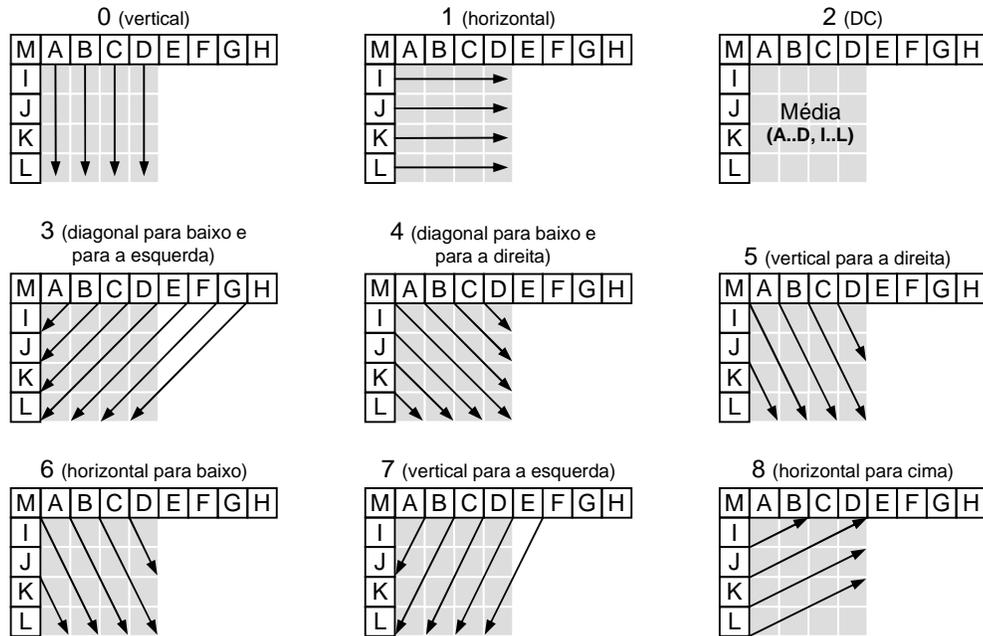


Figura 3.4: Modos de predição I4MB (AGOSTINI, 2007).

A Tabela 3.1, baseada em (HUANG, 2005), detalha todas as operações realizadas em cada um dos modos de predição intra-quadro 4x4 (DINIZ, 2009). Os valores nas colunas de L a H indicam as constantes que multiplicam as amostras vizinhas. Por exemplo, no modo 0 (vertical), as amostras vizinhas A, B, C e D são multiplicadas por 1. A coluna Soma indica uma constante que deve ser somada à amostra vizinha multiplicada pela constante. A coluna Desloca indica o número de posições à direita que devem ser deslocados os resultados das somas e a coluna Posições indica o grupo de amostras previstas que correspondem ao valor calculado. As duas últimas operações (soma e deslocamento) são utilizadas para fins de arredondamento.

Na maior parte das linhas da Tabela 3.1, o resultado gerado representa o valor de mais de uma amostra prevista. Por exemplo, o valor das amostras (0,1) e (1,0) previstas com o modo 3 é determinado por $(B + 2C + D + 2) \gg 2$. Algumas operações são aplicadas para gerar outras amostras em modos diferentes. Por exemplo, a amostra (1,3) do modo 5 e as amostras (1,1) e (3,0) do modo 7 também são determinadas através do cálculo $(B + 2C + D + 2) \gg 2$.

Tabela 3.1: Modos de predição I4MB, baseada em (HUANG, 2005).

Modo I4MB	L	K	J	I	M	A	B	C	D	E	F	G	H	Soma	Desloca	Posições (y, x)
0						1								0	>>0	(0,0) (1,0) (2,0) (3,0)
							1							0	>>0	(0,1) (1,1) (2,1) (3,1)
								1						0	>>0	(0,2) (1,2) (2,2) (3,2)
									1					0	>>0	(0,3) (1,3) (2,3) (3,3)
1				1										0	>>0	(0,0) (0,1) (0,2) (0,3)
			1											0	>>0	(1,0) (1,1) (1,2) (1,3)
		1												0	>>0	(2,0) (2,1) (2,2) (2,3)
	1													0	>>0	(3,0) (3,1) (3,2) (3,3)
2	1	1	1	1		1	1	1	1				4	>>3	Todas posições	
3						1	2	1						2	>>2	(0,0)
							1	2	1					2	>>2	(0,1) (1,0)
								1	2	1				2	>>2	(0,2) (1,1) (2,0)
									1	2	1			2	>>2	(0,3) (1,2) (2,1) (3,0)
										1	2	1		2	>>2	(1,3) (2,2) (3,1)
											1	2	1	2	>>2	(2,3) (3,2)
4	1	2	1											2	>>2	(3,0)
		1	2	1										2	>>2	(2,0) (3,1)
			1	2	1									2	>>2	(1,0) (2,1) (3,2)
				1	2	1								2	>>2	(0,0) (1,1) (2,2) (3,3)
					1	2	1							2	>>2	(0,1) (1,2) (2,3)
						1	2	1						2	>>2	(1,3)
5					1	1								1	>>1	(0,0) (2,1)
						1	1							1	>>1	(0,1) (2,2)
							1	1						1	>>1	(0,2) (2,3)
								1	1					1	>>1	(0,3)
		1	2	1										2	>>2	(3,0)
			1	2	1									2	>>2	(2,0)
				1	2	1								2	>>2	(1,0) (3,1)
					1	2	1							2	>>2	(1,1) (3,2)
6						1	2	1						2	>>2	(1,2) (3,3)
							1	2	1					2	>>2	(1,3)
	1	1												1	>>1	(3,0)
		1	1											1	>>1	(2,0) (3,2)
			1	1										1	>>1	(1,0) (2,2)
				1	1									1	>>1	(0,0) (1,2)
	1	2	1											2	>>2	(3,1)
		1	2	1										2	>>2	(2,1) (3,3)
7			1	2	1									2	>>2	(1,1) (2,3)
				1	2	1								2	>>2	(0,1) (1,3)
					1	2	1							2	>>2	(0,2)
						1	2	1						2	>>2	(0,3)
						1	1							1	>>1	(0,0)
							1	1						1	>>1	(0,1) (2,0)
								1	1					1	>>1	(0,2) (2,1)
									1	1				1	>>1	(0,3) (2,2)
8						1	2	1						2	>>2	(2,3)
							1	2	1					2	>>2	(1,0)
								1	2	1				2	>>2	(1,1) (3,0)
									1	2	1			2	>>2	(1,2) (3,1)
										1	2	1		2	>>2	(1,3) (3,2)
											1	2	1	2	>>2	(3,3)
	1													0	>>0	Demais posições
	1	1												1	>>1	(1,2) (2,0)
		1	1										1	>>1	(0,2) (1,0)	
			1	1									1	>>1	(0,0)	
3	1												2	>>2	(1,3) (2,1)	
1	2	1											2	>>2	(0,3) (1,1)	
	1	2	1										2	>>2	(0,1)	

3.1.2 Predição Intra-Quadro 16x16

Quando a predição realizada é do tipo intra-quadro 16x16, isto é, sobre um macrobloco completo, quatro modos de codificação podem ser utilizados. Todos os modos utilizam as 32 amostras de luminância que compõem as bordas superior e à esquerda do macrobloco predito (H e V na Figura 3.5), chamadas amostras vizinhas. A Figura 3.5 ilustra as predições possíveis e os tipos de cópia que elas representam.

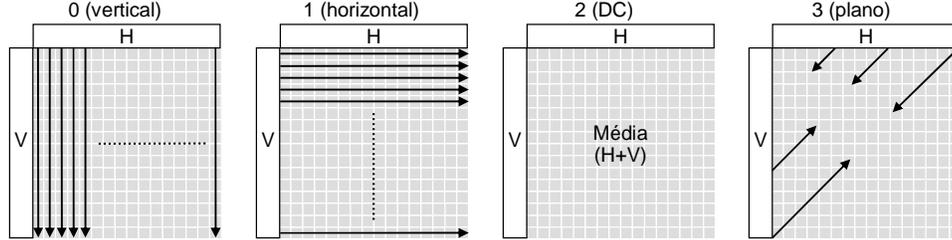


Figura 3.5: Modos de predição I16MB (AGOSTINI, 2007).

Nos modos 0 e 1, as amostras de H e V são copiadas nos sentidos vertical e horizontal, respectivamente, para todo o macrobloco predito, assim como na predição Intra 4x4. No modo 2, as amostras preditas são calculadas através da média simples entre as amostras de H e V , conforme em (19).

$$DC = \left(\sum_{x'=0}^{15} p[x', -1] + \sum_{y'=0}^{15} p[-1, y'] + 16 \right) \gg 5 \quad (19)$$

Em (19), $\mathbf{p}[y, \mathbf{x}]$ é uma amostra localizada na posição indicada pela linha \mathbf{y} e pela coluna \mathbf{x} do macrobloco. A amostra $[0,0]$ localiza-se no canto superior esquerdo, de forma que, quando o índice é igual a -1 , a amostra está no macrobloco vizinho (esquerdo ou superior), que já foi codificado e reconstruído. O valor resultante (**DC**) é então copiado para todas as amostras do macrobloco predito.

O modo 3 (plano) é o mais complexo, pois seu cálculo envolve uma função linear que utiliza três parâmetros (\mathbf{a} , \mathbf{b} e \mathbf{c}) pré-calculados para cada amostra do macrobloco predito, como definido em (20).

$$P[y, x] = \text{Clip}((a + b \cdot (x - 7) + c \cdot (y - 7) + 16) \gg 5) \quad (20)$$

Em (20), $\mathbf{P}[y, \mathbf{x}]$ é a amostra predita, **Clip** é uma função de truncagem (ou arredondamento) para limitar os valores na faixa de representação da imagem e os parâmetros \mathbf{a} , \mathbf{b} e \mathbf{c} são definidos em (21), (22) e (23), respectivamente. Os valores de \mathbf{H} e \mathbf{V} em (22) e (23) estão definidos em (24) e (25), respectivamente.

$$a = 16 \cdot (p[-1, 15] + p[15, -1]) \quad (21)$$

$$b = (5 \cdot H + 32) \gg 6 \quad (22)$$

$$c = (5 \cdot V + 32) \gg 6 \quad (23)$$

$$H = \sum_{x'=0}^7 (x'+1) \cdot (p[-1, 8+x'] - p[-1, 6-x']) \quad (24)$$

$$V = \sum_{y'=0}^7 (y'+1) \cdot (p[8+y', -1] - p[6-y', -1]) \quad (25)$$

3.1.3 Predição para Crominância

As amostras de crominância são preditas de forma similar às amostras de luminância codificadas com o modo intra-quadro 16x16, com exceção do número de pixels

envolvidos. Também são utilizados quatro modos de predição com os mesmos nomes, mas com numeração trocada em relação à predição intra-quadro 16x16, como ilustra a Figura 3.6 (AGOSTINI, 2007).

Neste trabalho, a sub-amostragem utilizada é do tipo 4:2:0 e, portanto os blocos de croma são de dimensão 8x8. Desta forma, existem 8 amostras vizinhas à esquerda e 8 amostras vizinhas acima do bloco predito, de forma que 16 amostras são utilizadas nos cálculos das amostras preditas, ao invés de 32. O mesmo modo é sempre escolhido para ambos os canais de croma (Cb e Cr).

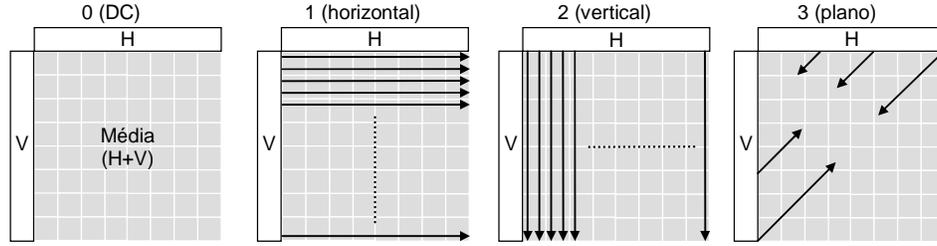


Figura 3.6: Modos de predição para blocos 8x8 de croma (AGOSTINI, 2007).

Na predição para croma, o modo DC corresponde ao modo 0. Ele é calculado através de uma média aritmética simples sobre todas as 16 amostras vizinhas. O cálculo realizado é similar ao realizado em (19), com exceção do arredondamento e do deslocamento que são realizados através de uma soma com 8 e de um deslocamento à direita de 4 bits neste caso, já que o número de amostras vizinhas envolvidas é menor.

O modo plano para croma é similar ao modo plano da predição I16MB. A predição de uma amostra é definida em (26), onde os parâmetros a , b e c estão definidos em (27), (28) e (29), respectivamente. Os valores de H e V em (28) e (29) estão definidos em (30) e (31), respectivamente.

$$P[y, x] = \text{Clip}((a + b \cdot (x - 3) + c \cdot (y - 3) + 16) \gg 5) \quad (26)$$

$$a = 16 \cdot (p[-1, 7] + p[7, -1]) \quad (27)$$

$$b = (34 \cdot H + 32) \gg 6 \quad (28)$$

$$c = (34 \cdot V + 32) \gg 6 \quad (29)$$

$$H = \sum_{x'=0}^3 (x'+1) \cdot (p[4+x', -1] - p[2-x', -1]) \quad (30)$$

$$V = \sum_{y'=0}^3 (y'+1) \cdot (p[-1, 4+y'] - p[-1, 2-y']) \quad (31)$$

3.2 A Predição Inter-Quadros

A predição inter-quadros é composta por dois módulos: a estimação de movimento (ME – *Motion Estimation*) e a compensação de movimento (MC – *Motion Compensation*). A estimação de movimento está presente apenas no codificador H.264/AVC, enquanto que a compensação existem em ambos codificador e decodificador. O nome mais adequado para a estimação de movimento seria “estimativa de movimento”. Contudo, a literatura técnica utiliza, em maioria esmagadora, o primeiro termo. Por esta razão, este também será utilizado neste trabalho.

3.2.1 Estimação de Movimento

A maior complexidade computacional do padrão H.264/AVC encontra-se na estimação de movimento devido às diversas técnicas inovadoras inseridas no padrão e presentes neste módulo. Por outro lado, a estimação de movimento é a principal responsável pela grande eficiência de codificação e pelos altos níveis de compressão proporcionados pelo H.264/AVC (WIEGAND, 2003) (RICHARDSON, 2003).

O objetivo da estimação de movimento é aplicar diversas técnicas para diminuir a redundância temporal entre os quadros de um vídeo. Para isso, ela procura encontrar, em uma determinada área de pesquisa (em um ou mais quadros de referência), o bloco de melhor casamento, isto é, maior semelhança, com o bloco sendo predito. Quando o bloco de melhor casamento é encontrado, um vetor de movimento que indica a posição deste bloco é gerado e codificado junto com o bloco.

A estimação de movimento é realizada apenas sobre as amostras de luminância dos macroblocos. Como os componentes de crominância possuem metade da resolução dos componentes de luminância (8x8 amostras), os vetores de movimento para crominância são calculados simplesmente através de uma divisão por dois dos vetores de luminância.

A principal característica do padrão que leva ao bom desempenho da predição inter-quadros é a utilização de blocos de tamanhos variáveis. Assim, a busca pelo bloco com melhor casamento nos quadros de referência pode ser realizada através de comparações com partições de um macrobloco, além do macrobloco inteiro. O padrão H.264/AVC permite o uso de partições dos seguintes tamanhos: 16x16 (isto é, o macrobloco inteiro), 8x16, 16x8 e 8x8, como ilustrado na Figura 3.7.

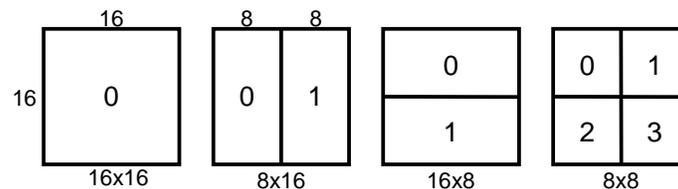


Figura 3.7: Divisão do macrobloco em partições de macrobloco (AGOSTINI, 2007).

Nos casos em que a partição é do tipo 8x8, um novo nível de particionamento pode ocorrer e novas partições, chamadas partições de sub-macrobloco, são utilizadas. As partições de sub-macrobloco podem ser dos seguintes tamanhos: 8x8, 4x8, 8x4 e 4x4, conforme apresentado na Figura 3.8.

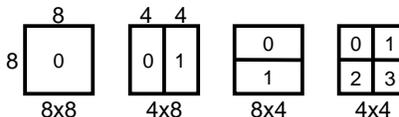


Figura 3.8: Divisão do sub-macrobloco em partições (AGOSTINI, 2007).

Os blocos de crominância são divididos da mesma forma que os de luminância, mas com metade das dimensões. Por exemplo, se o tamanho de bloco de luminância utilizado for 16x8, o bloco de crominância utilizado possui dimensões 8x4.

Para que a estimação de movimento apresente uma boa eficiência, a decisão de tamanho de bloco deve ser realizada corretamente. Quando um quadro é codificado com partições de grandes dimensões (16x16, 16x8, 8x16), o número de vetores de movimento utilizados é menor do que seria quando são utilizados partições de tamanhos menores (8x4, 4x8, 4x4). O número de bits utilizados para codificar vetores de movimento é, portanto, menor quando partições de tamanhos grandes são escolhidas.

Por outro lado, quanto maior o tamanho de partição escolhido, maior tende a ser o resíduo gerado após a predição, especialmente em áreas de grande granularidade, isto é, pouco homogêneas. Assim, quadros codificados com a maioria das partições em tamanhos menores utilizam mais bits para codificar os vetores de movimento, mas menos bits para codificar os resíduos. Por outro lado, quadros codificados com a maioria das partições em tamanhos maiores utilizam menos bits para codificar os vetores de movimento, mas mais bits para codificar os resíduos.

Como se pode perceber, a decisão do melhor tamanho de partição para a predição inter-quadros é de extrema importância para que o codificador H.264/AVC apresente um bom desempenho em termos de qualidade da imagem e de taxa de bits utilizados. A melhor escolha é realizada através da análise dos resultados de estimação de movimento com todos os tamanhos de partição e aquele que apresentar a melhor eficiência de codificação é escolhido. A complexidade computacional envolvida neste processo é extremamente alta, visto que os processos de estimação de movimento e de compensação de movimento devem ser realizados repetidamente para que todos os tamanhos de partição sejam comparados.

Outras inovações no padrão H.264/AVC também estão presentes no módulo de estimação de movimento, como a predição com precisão fracionária para os vetores de movimento. Como, em geral, os movimentos em um vídeo que acontecem de um quadro para outro não correspondem a posições inteiras de pixel, o uso de posições fracionárias possibilita que um melhor casamento seja encontrado para o bloco sendo predito. O H.264/AVC possibilita a utilização de vetores de movimento com valores fracionários de $\frac{1}{2}$ e $\frac{1}{4}$ de pixel. As amostras fracionárias de $\frac{1}{2}$ pixel são calculadas através de interpolações das amostras em posições inteiras utilizando um filtro FIR de seis *taps*. Já as amostras fracionárias de $\frac{1}{4}$ de pixel são calculadas através de interpolações das amostras fracionárias de $\frac{1}{2}$ pixel (calculadas no passo anterior) e das amostras em posições inteiras, usando uma média aritmética simples entre dois pontos. Como os elementos de crominância possuem metade da resolução dos elementos de luminância, a predição com precisão fracionária é realizada, na prática, com posições de até $\frac{1}{8}$ de pixel.

O uso de múltiplos quadros de referência é também outra característica importante do H.264/AVC implementada no módulo de estimação de movimento. O padrão define duas listas de quadros de referências que permitem que sejam utilizados quadros localizados temporalmente antes ou depois do quadro sendo codificado, como mostrado na Figura 3.9.

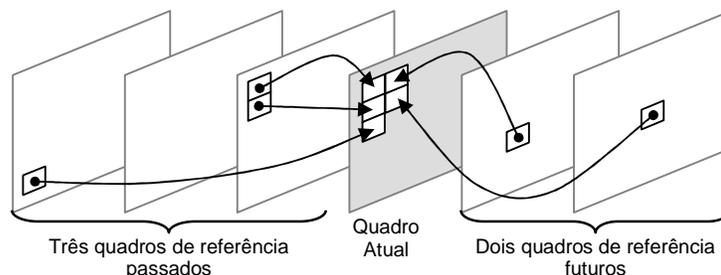


Figura 3.9: Uso de múltiplos quadros de referência (AGOSTINI, 2007).

A lista 0 contém, em ordem, o quadro passado mais próximo, seguido de quaisquer outros quadros passados, seguidos de quaisquer quadros futuros. A lista 1 contém, em ordem, o quadro futuro mais próximo, seguido de quaisquer outros quadros futuros,

seguidos de quaisquer outros quadros passados. Como o H.264/AVC permite realizar a codificação em uma ordem completamente diferente da ordem de exibição dos quadros na tela, as listas 0 e 1 devem conter quadros que já tenham sido codificados, incluindo aí os quadros futuros.

Slices do tipo I são codificados sem estimação de movimento, pois realizam apenas predição intra-quadro sobre as amostras do *slice* atual. Já os *slices* do tipo P podem ser codificados através da estimação de movimento, da predição intra-quadro ou através de macroblocos do tipo *SKIP*. A estimação de movimento pode ser realizada sobre quadros passados ou futuros armazenados na lista 0. Macroblocos do tipo *SKIP* são gerados quando o custo taxa-distorção para codificar o macrobloco é mais elevado do que o custo de não enviar informação alguma sobre o macrobloco (KANNANGARA, 2005). Quando o decodificador encontra um macrobloco tipo *SKIP* em um *slice* P, o decodificador reconstrói o macrobloco a partir do primeiro quadro armazenado na lista 0, calculando seu vetor de movimento a partir dos vetores vizinhos. Se o macrobloco *SKIP* estiver em um tipo *slice* B, então o macrobloco é reconstruído no decodificador usando predição direta (SAHAFI, 2005).

As partições de macrobloco em um *slice* do tipo B podem ser codificadas através da predição inter-quadros realizada de diversas formas. A estimação de movimento pode ser realizada com um quadro armazenado na lista 0, um quadro armazenado na lista 1 ou ainda com uma média entre as amostras de um quadro da lista 0 e um quadro da lista 1. Esta última forma é chamada de predição bi-preditiva. O modo *SKIP* também pode ser utilizado, assim como nos *slices* de tipo P. Além disso, uma forma de predição chamada estimação direta também pode ser realizada, na qual nenhum vetor de movimento é transmitido. Ao invés disso, o decodificador calcula os vetores da lista 0 e da lista 1 baseado nos vetores previamente codificados.

Como se pode perceber, o grande número de possibilidades de codificação introduzidas pelo padrão H.264/AVC aumenta muito a sua complexidade em relação aos padrões mais antigos. Contudo, como o padrão normatiza apenas o decodificador, a aplicação de todas estas técnicas não é mandatória no projeto dos codificadores, tornando a sua implementação menos restrita e possibilitando que várias das suas características possam ser simplesmente ignoradas na construção do codificador para que a sua complexidade seja reduzida.

O codificador utilizado neste trabalho realiza a estimação de movimento considerando apenas um quadro de referência, ignorando a predição bi-preditiva sugerida pelo padrão H.264/AVC. Essa simplificação diminui consideravelmente a complexidade do padrão, tornando possível a sua implementação em hardware. Por outro lado, tal simplificação pode gerar uma redução na eficiência da codificação.

3.2.2 Compensação de Movimento

O módulo de compensação de movimento (MC) está presente tanto no codificador quanto no decodificador do padrão H.264/AVC. A estimação de movimento tem o papel de encontrar, nos quadros de referência, o bloco que apresenta o melhor casamento com o bloco atual, gerando um vetor de movimento. A compensação de movimento realiza o papel inverso, isto é, a partir do vetor de movimento, localiza o bloco de melhor casamento no quadro de referência e o copia para montar o quadro predito. Depois disso, a MC realiza uma subtração entre as amostras do quadro predito e do quadro atual, gerando o quadro de resíduos que passa pelas transformadas.

Como explicado na seção anterior, diversas simplificações podem ser realizadas na estimação de movimento para diminuir a complexidade do codificador. Da mesma forma, a compensação de movimento do codificador pode ser simplificada para corresponder às simplificações realizadas na estimação de movimento. Por exemplo, uma compensação de movimento com suporte a predição de precisão fracionária não é necessária em um codificador que implementa apenas uma estimação de movimento com posições inteiras de pixel. No decodificador, contudo, a MC deve dar suporte a todas as possibilidades de codificação inter-quadros que o padrão H.264/AVC oferece, tais como o uso partições de macrobloco de múltiplos tamanhos, o uso de múltiplos quadros de referência, predição com precisão fracionária, predição bi-preditiva, entre outras.

3.3 Processo de Codificação

Durante a decodificação de um vídeo segundo o padrão H.264/AVC, a predição ocorre de acordo com os modos de predição sinalizados no *bitstream* durante a codificação e de acordo com o tipo do macrobloco. É o codificador de vídeo que toma a decisão de qual modo deve ser utilizado para codificar cada macrobloco. Entretanto, essa decisão não é normatizada e, portanto, a escolha do algoritmo de decisão dos modos de predição é livre.

A Figura 3.10 mostra o processo de codificação realizado até que o melhor modo seja determinado, conforme proposto na técnica de RDO. O caminho representado pelos blocos em cinza é percorrido uma vez para cada um dos modos de predição existentes. Já o módulo de decisão de modo, em cor branca, é utilizado apenas uma vez durante todo o fluxo de codificação, ao contrário dos demais. As setas em linhas contínuas apresentam apenas o fluxo de execução do codificador. As setas com linhas pontilhadas apresentam o envio de dados da codificação para a etapa de decisão de modo.

A primeira etapa, representada pelo bloco Predição Intra/Inter, realiza os cálculos de modos para cada macrobloco considerando a predição intra-quadro ou inter-quadros. No caso da predição intra-quadro, esta etapa realiza os cálculos apresentados na seção 3.1 para todos os modos I4MB e I16MB de luminância e 8x8 de crominância. No caso da predição inter-quadros, o bloco com melhor casamento é procurado nos quadros de referência com todos os tamanhos de partição (16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4), conforme explicado na seção 3.2. Esta etapa é seguida pela geração de resíduos, a qual realiza a subtração entre os valores das amostras do macrobloco original e do macrobloco predito. Em seguida, são realizadas as operações de transformadas e quantização. Finalmente, os coeficientes resultantes da etapa de quantização e as informações de tipo do macrobloco e modos de predição são codificados pelo bloco de codificação de entropia.

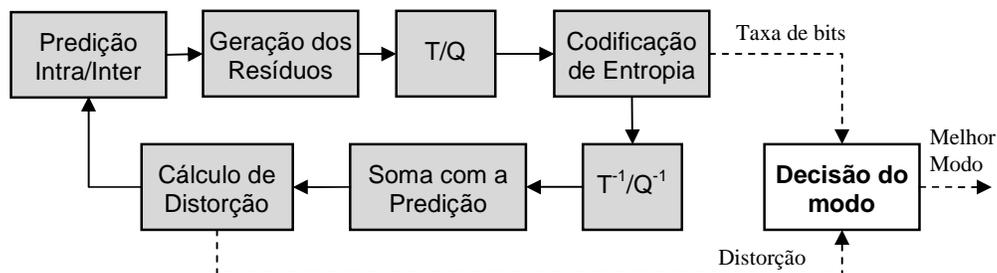


Figura 3.10: Processo de codificação e decisão de modo baseada em RDO.

Os coeficientes quantizados são processados pelo bloco T^{-1}/Q^{-1} (Transformadas e Quantização Inversas) para serem somados às amostras preditas correspondentes no macrobloco predito, gerando o macrobloco reconstruído, que será usado para predição de outros macroblocos. Esta reconstrução é o mesmo processo realizado no decodificador e garante que as predições feitas no codificador sejam as mesmas realizadas na decodificação. Por fim, a distorção pode ser calculada com base nas amostras do macrobloco reconstruído e nas amostras do macrobloco original. A partir das informações de taxa de bits e de distorção geradas pela utilização de cada um dos modos de predição, o codificador escolhe o modo com menor custo taxa-distorção, conforme explicado na seção 2.7.2 desta dissertação.

Na predição I16MB, um modo de predição é escolhido para cada macrobloco, gerando um macrobloco de resíduos. Os resíduos são divididos em blocos 4x4 para serem processados pelos blocos T/Q , T^{-1}/Q^{-1} e Soma com a Predição. No caso da predição I4MB, um modo de predição é escolhido localmente para cada bloco 4x4, sendo que este bloco 4x4 é processado pelos módulos de geração de resíduos, T/Q , T^{-1}/Q^{-1} e Soma com a Predição. Desta forma, a predição I16MB usa como referência amostras reconstruídas de macroblocos vizinhos. Já na predição I4MB, cada bloco 4x4 usa como referência as amostras reconstruídas dos blocos 4x4 anteriormente processados, gerando uma dependência de dados maior. A predição inter-quadros, por outro lado, não apresenta tal dependência de dados, pois a codificação de um bloco não depende dos blocos vizinhos, mas sim dos quadros codificados anteriormente.

Durante a seleção de modo da predição intra-quadro, os cálculos relativos a todos os modos são realizados, conforme definido pelo padrão. Dois tipos de busca podem acontecer: a busca completa ou a busca rápida. A busca completa (*Full Search*) calcula todos os modos de predição intra-quadro possíveis, enquanto que a busca rápida utiliza algoritmos rápidos que aplicam alguma heurística baseada na informação contida no vídeo para escolher apenas alguns dos modos de predição intra-quadro, diminuindo a complexidade da decisão de modo. Pan (2007), Chang (2007), Li (2007) e Tsai (2008) propõem algoritmos rápidos para selecionar somente alguns modos de predição intra-quadro a serem comparados na etapa de decisão do modo. Os trabalhos de Suh (2005) e Kuo (2008) utilizam o algoritmo de busca completa.

Na predição inter-quadros, existem diversos algoritmos que podem ser utilizados para busca nos quadros de referência. Estes algoritmos variam desde a busca exaustiva (BHASKARAN, 1997) até algoritmos sub-ótimos que reduzem muito a complexidade da busca, mas que produzem impactos negativos na eficiência da codificação. Dentre estes algoritmos rápidos é possível citar a busca circular (TOURAPIS, 1999), a busca logarítmica (JAIN, 1981), a busca em três passos (LI, 1994), a busca espiral (KHUN, 1999) (KROUPIS, 2005), a busca em diamante (YI, 2005), a busca hierárquica (CHU, 1997), entre outros.

A decisão do modo de predição, foco deste trabalho, escolhe um dos tipos do modo de predição gerados durante a etapa de Predição Intra/Inter (I4MB, I16MB, P16x16, P16x8, P8x16, P8x8, P8x4, P4x8 e P4x4, além do modo *SKIP*). Para a realização de tal escolha, o software de referência do H.264/AVC utiliza a complexa técnica de Otimização Taxa-Distorção, apresentada na seção 2.7.2. O capítulo 4 deste trabalho apresenta uma série de heurísticas desenvolvidas com o objetivo de acelerar a decisão de modo e de possibilitar a sua implementação em hardware.

4 DECISÃO RÁPIDA DE MODO DE CODIFICAÇÃO

Para atingir a melhor eficiência de codificação, o software de referência do H.264/AVC emprega a técnica de Otimização Taxa-Distorção (*Rate-Distortion Optimization* – RDO) para encontrar o modo de codificação que apresenta o custo taxa-distorção (RD) mínimo, através do uso do coeficiente de Lagrange (WIEGAND, 2003) (SULLIVAN, 1998). Entretanto, a complexidade computacional do codificador H.264/AVC é dramaticamente aumentada por esta técnica de otimização, conforme explicado anteriormente. Para lidar com este problema, é importante desenvolver um método eficiente para seleção de modo de predição com complexidade inferior ao do RDO, visando à aceleração do tempo de codificação e com impacto mínimo na qualidade e na taxa de compressão atingida pelo RDO.

Diversos trabalhos têm explorado características intrínsecas dos vídeos digitais a fim de simplificar o processo de decisão de modo e possibilitar a utilização do codificador H.264/AVC em aplicações de tempo-real. A grande maioria desses trabalhos, contudo, está direcionada para aplicações em software e não leva em consideração o fluxo de execução do codificador H.264/AVC quando implementado em hardware.

As primeiras seções deste capítulo apresentam as heurísticas desenvolvidas para a decisão de modo intra-quadro, com foco em uma futura implementação em hardware. Em seguida, são apresentadas as idéias utilizadas no processo de decisão rápida inter-quadros. Após a apresentação de cada heurística proposta, são mostrados os resultados da sua aplicação isolada, comparando-os com a decisão baseada na técnica de RDO em termos de taxa de bits e de PSNR. Todos os dados de simulações apresentados neste capítulo foram obtidos através da execução do software de referência do padrão H.264/AVC (versão 17.1) (SUHRING, 2010), com vídeos tipicamente utilizados para análises em trabalhos científicos.

4.1 Decisão em Quadros I e Decisão em Quadros P

Um grande número de trabalhos que abordam a decisão de modo de codificação para o padrão H.264/AVC explora o fato de que uma parcela muito pequena de macroblocos é codificada com algum modo intra-quadro em quadros do tipo P. No trabalho de Huang (HUANG, 2009), uma análise realizada para duas sequências de vídeo de resolução QCIF (176x144 pixels) mostrou que apenas 0,11% dos modos escolhidos para quadros P são do tipo intra-quadro. Pan (PAN, 2007) mostra que a quantidade de modos intra-quadro em quadros P varia de 0,05% a 1,4%, também considerando apenas vídeos QCIF, e propõe a utilização de apenas modos do tipo inter-quadros em quadros P.

A Tabela 4.1 mostra uma avaliação da ocorrência média de modos intra-quadro em quadros P que foi desenvolvida neste trabalho. Oito vídeos diferentes com resolução CIF (352x288 pixels) foram utilizados. Apenas 3,8% dos macroblocos foram

codificados com algum modo intra-quadro, uma parcela bastante pequena, em comparação à ocorrência de 96,2% de modos inter-quadros.

Tabela 4.1: Ocorrência de modos intra-quadro e inter-quadros em quadros P de resolução CIF.

Modos	Ocorrência	Frequência
<i>Intra-quadro</i>	10.895	3,8%
<i>Inter-quadros</i>	276.601	96,2%

A partir das frequências de modos apresentadas na Tabela 4.1 e de posteriores simulações para determinar o impacto da simplificação proposta por Pan (PAN, 2007), verificou-se que uma decisão entre modos intra-quadro e inter-quadros poderia, de fato, ser eliminada sem a inserção de perdas na eficiência da codificação. A Tabela 4.2 mostra uma comparação, em termos de PSNR e taxa de bits, entre a codificação realizada com e sem a utilização de modos intra-quadro em quadros P. A limitação só tem efeitos sobre os quadros P, já que os quadros I são, naturalmente, compostos apenas por macroblocos intra-quadro. Em média, a taxa de bits aumentou apenas 0,39% e a qualidade da imagem não sofreu alterações significativas. Esses valores justificam, portanto, a utilização de apenas macroblocos do tipo inter-quadros em quadros P neste trabalho. Em todas as simulações realizadas e relatadas nesta e nas próximas seções, a sequência de quadros considerada é IPP (isto é, dois quadros I separados por dois quadros P).

Tabela 4.2: Comparação entre codificação com e sem utilização de modos intra-quadro em quadros P em vídeos de resolução CIF.

Vídeo	(Intra+Inter) em P		Apenas Inter em P		(Intra+Inter) x Inter	
	Taxa de bits	PSNR (dB)	Taxa de bits	PSNR (dB)	+ Taxa de bits (%)	+ PSNR (dB)
<i>BUS</i>	799.824	35,88	804.192	35,90	0,55	0,01
<i>CITY</i>	536.512	35,67	536.672	35,67	0,03	0,00
<i>FOOTBALL</i>	876.168	35,82	885.808	35,84	1,10	0,02
<i>FOREMAN</i>	333.928	37,66	336.008	37,67	0,62	0,01
<i>HARBOUR</i>	901.752	35,56	902.696	35,56	0,10	0,00
<i>ICE</i>	289.624	39,96	290.696	39,97	0,37	0,01
<i>MOBILE</i>	1.321.616	35,57	1.321.808	35,57	0,01	0,00
<i>SOCCKER</i>	591.968	35,56	593.776	35,57	0,31	0,01
MÉDIA	706.424	36,46	708.957	36,47	0,39	0,01

Nesta dissertação, o problema da decisão de modos foi dividido, portanto, em dois: decisão intra-quadro e decisão inter-quadros. Cada um deles, por sua vez, foi subdividido em decisões mais simples, conforme apresentado nas próximas seções.

4.2 Decisão Rápida de Modo Intra-Quadro

Com a finalidade de simplificar o problema complexo da decisão de modo intra-quadro, este foi dividido em quatro passos mais simples, descritos a seguir:

1. Decisão do melhor modo de predição intra-quadro 4x4 para cada um dos 16 blocos 4x4 que compõem o macrobloco de luminância;
2. Decisão do melhor modo de predição intra-quadro 16x16 para o macrobloco de luminância;
3. Decisão do melhor modo de predição para o macrobloco de crominância;

4. Decisão entre o melhor modo intra-quadro 4x4 e o melhor modo intra-quadro 16x16 de luminância (isto é, decisão de melhor tamanho de bloco intra-quadro).

Como os itens citados não são interdependentes, eles não definem a ordem de execução da decisão proposta. Pelo contrário, a decisão pode ser realizada em qualquer ordem ou, ainda, em processos concorrentes. As decisões de melhores modos intra-quadro 4x4, intra-quadro 16x16 e de crominância (itens 1, 2 e 3, respectivamente) propostas são baseadas apenas em cálculo de distorção, como será explicado na próxima seção. A proposta para decisão entre os melhores modos para blocos de tamanhos diferentes (item 4) é realizada com base no cálculo de heterogeneidade do macrobloco, conforme será explicado na seção 4.2.2.

4.2.1 Decisão Intra-Quadro com Bloco de Tamanho Fixo Baseada em Distorção

Uma série de simulações foi realizada com a finalidade de analisar a diferença, em termos de taxa de bits e qualidade da imagem gerada, entre a decisão de modo baseada na técnica de RDO e a decisão de modo baseada apenas no valor de distorção quando o tamanho de bloco é o mesmo para todos os modos. As análises foram realizadas com vídeos de resolução CIF (352x288 pixels), levando em consideração apenas quadros I e apenas blocos 4x4 (ou apenas blocos 16x16).

Os dados obtidos nas análises, apresentados nas próximas subseções, mostraram que a diferença entre as duas abordagens é muito pequena em termos de taxa de bits e qualidade da imagem. Isto acontece porque o modo de codificação que resulta em resíduos próximos dos mínimos tende a gerar uma quantidade de informação codificada menor, já que valores menores de resíduos são transformados e, após, tendem a ser quantizados para zero. Contudo, para que esta abordagem seja utilizada, o tamanho de bloco utilizado na codificação deve ser fixo. Caso contrário, a informação de codificação de cada bloco também deve ser levada em consideração (e não apenas os resíduos gerados), já que um bloco intra-quadro 16x16 possui a mesma quantidade de informação de codificação que um bloco intra-quadro 4x4, mas 16 vezes mais resíduos.

As escolhas de melhor modo intra-quadro 4x4 e de melhor modo intra-quadro 16x16 podem ser realizadas, portanto, levando em consideração apenas a distorção da imagem codificada. No software de referência, a distorção é medida como o PSNR (*Peak Signal-to-Noise Ratio*) entre o bloco original e o bloco codificado e reconstruído. Contudo, o cálculo de distorção entre esses dois blocos faria com que a decisão fosse tomada somente após a realização do caminho composto por predição intra-quadro, cálculo de resíduos, transformadas, quantização, quantização inversa, transformadas inversas, reconstrução e filtro. Além disso, como a decisão é realizada como uma comparação entre todos os modos, este caminho deveria ser percorrido uma vez para cada modo de codificação. Para solucionar este problema, estabelecemos, neste trabalho, que o cálculo de distorção é realizado entre o bloco original e bloco predito, isto é, a saída da predição intra-quadro.

Neste trabalho, foram avaliadas as seguintes métricas de distorção: Soma das Diferenças Absolutas (SAD), Soma das Diferenças Absolutas Transformadas (SATD) e Soma das Diferenças Quadráticas (SSD), conforme mostram as subseções a seguir.

4.2.1.1 Decisão Baseada na Soma das Diferenças Absolutas (SAD)

A Soma das Diferenças Absolutas (SAD – *Sum of Absolute Differences*) é uma métrica de distorção de baixa complexidade largamente utilizada em codificadores de vídeo H.264/AVC de baixa e alta resolução. A função de SAD é calculada para cada

bloco através da comparação entre as amostras (pixels) de cada bloco predito de acordo com um modo de codificação e as amostras do bloco do quadro atual (original), conforme mostrado em (32).

$$SAD = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |P_{i,j} - O_{i,j}| \quad (32)$$

Em (32), \mathbf{P} é um bloco predito por algum modo de codificação intra-quadro, \mathbf{O} é o bloco original e \mathbf{m} e \mathbf{n} são as dimensões do bloco em número de amostras. Para realizar a decisão de modo de codificação para cada bloco, os valores de SAD são calculados para cada modo candidato e, então, comparados.

As Tabelas 4.3 e 4.4 mostram os resultados obtidos para cada um dos vídeos com a aplicação da decisão baseada apenas no valor de SAD para blocos intra-quadro 4x4 e intra-quadro 16x16, respectivamente. As últimas colunas da tabela apresentam comparações entre a decisão baseada em SAD e a decisão baseada em RDO. É importante destacar que os tamanhos de blocos também foram limitados a apenas 4x4 (ou 16x16, no caso da Tabela 4.4) na decisão baseada em RDO. Percebe-se, na última linha de ambas as tabelas, que a decisão baseada em SAD para blocos intra-quadro 4x4 e intra-quadro 16x16 gerou um aumento médio de 5,29% e 2,41% na taxa de bits e uma redução de 0,28 dB e 0 dB na qualidade da imagem, respectivamente.

Tabela 4.3: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão intra-quadro 4x4 baseada em RDO e decisão intra-quadro 4x4 baseada em SAD.

Vídeo	Decisão RDO		Decisão por SAD		RDO x SAD	
	Taxa de bits	PSNR (dB)	Taxa de bits	PSNR (dB)	+ Taxa de bits (%)	+ PSNR (dB)
<i>BUS</i>	1.371.896	36,39	1422360	36,01	3,68	-0,37
<i>CITY</i>	1.195.576	36,23	1243248	35,86	3,97	-0,37
<i>CREW</i>	507.792	39,07	555000	38,88	9,30	-0,18
<i>FOOTBALL</i>	1.381.480	36,21	1425136	35,82	3,16	-0,39
<i>FOREMAN</i>	664.696	38,08	716704	37,93	7,82	-0,16
<i>ICE</i>	548.192	40,23	588032	40,22	7,27	-0,01
<i>MOBILE</i>	2.267.080	36,27	2335088	35,89	2,99	-0,39
<i>SOCCER</i>	1.124.104	35,95	1171104	35,60	4,18	-0,35
MÉDIA	1.132.602	37,30	1.182.084	37,026	5,29	-0,28

Tabela 4.4: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão intra-quadro 16x16 baseada em RDO e decisão intra-quadro 16x16 baseada em SAD.

Vídeo	Decisão RDO		Decisão por SAD		RDO x SAD	
	Taxa de bits	PSNR (dB)	Taxa de bits	PSNR (dB)	+ Taxa de bits (%)	+ PSNR (dB)
<i>BUS</i>	1.531.264	36,00	1.551.792	36,00	1,34	0,00
<i>CITY</i>	1.337.528	35,94	1.361.952	35,95	1,83	0,01
<i>CREW</i>	598.968	38,99	611.136	39,00	2,03	0,01
<i>FOOTBALL</i>	1.563.352	35,81	1.590.896	35,82	1,76	0,01
<i>FOREMAN</i>	888.880	37,80	925.376	37,80	4,11	0,00
<i>ICE</i>	663.928	40,10	692.336	40,11	4,28	0,00
<i>MOBILE</i>	2.507.528	35,80	2.557.648	35,79	2,00	0,00
<i>SOCCER</i>	1.235.720	35,63	1.259.232	35,63	1,90	0,00
MÉDIA	1.290.896	37,01	1.318.796	37,01	2,41	0,00

4.2.1.2 Decisão Baseada na Soma das Diferenças Absolutas Transformadas (SATD)

A Soma das Diferenças Absolutas Transformadas (SATD – *Sum of Absolute Transformed Differences*) é uma métrica de distorção mais complexa que o SAD, pois realiza, além das subtrações entre as amostras **O** e **P**, uma série de operações de soma e subtração presentes no módulo de transformadas do codificador H.264/AVC. A transformada utilizada neste trabalho foi a Hadamard 2-D 4x4 Direta, definida pela série de operações apresentadas na Tabela 4.5 (AGOSTINI, 2007).

Na Tabela 4.5, os valores **W_n** representam os resultados obtidos após a subtração entre os valores de entrada (isto é, a diferença entre as amostras originais e as amostras preditas). As diferenças **W_n** transformadas estão apresentadas na última coluna da Tabela 4.5 como **S_n**.

Tabela 4.5: Algoritmo utilizado para o cálculo da Hadamard 2-D 4x4 Direta.

$a_0 = W_0 + W_4$	$b_0 = a_0 + a_1$	$c_0 = b_0 + b_1$	$S_0 = (c_0 + c_1)/2$
$a_1 = W_8 + W_{12}$	$b_1 = a_2 + a_3$	$c_1 = b_2 + b_3$	$S_1 = (c_0 - c_1)/2$
$a_2 = W_1 + W_5$	$b_2 = a_4 + a_5$	$c_2 = b_0 - b_1$	$S_2 = (c_2 - c_3)/2$
$a_3 = W_9 + W_{13}$	$b_3 = a_6 + a_7$	$c_3 = b_2 - b_3$	$S_3 = (c_2 + c_3)/2$
$a_4 = W_2 + W_6$	$b_4 = a_0 - a_1$	$c_4 = b_4 + b_5$	$S_4 = (c_4 + c_5)/2$
$a_5 = W_{10} + W_{14}$	$b_5 = a_2 - a_3$	$c_5 = b_6 + b_7$	$S_5 = (c_4 - c_5)/2$
$a_6 = W_3 + W_7$	$b_6 = a_4 - a_5$	$c_6 = b_4 - b_5$	$S_6 = (c_6 - c_7)/2$
$a_7 = W_{11} + W_{15}$	$b_7 = a_6 - a_7$	$c_7 = b_6 - b_7$	$S_7 = (c_6 + c_7)/2$
$a_8 = W_0 - W_4$	$b_8 = a_8 - a_9$	$c_8 = b_8 + b_9$	$S_8 = (c_8 + c_9)/2$
$a_9 = W_8 - W_{12}$	$b_9 = a_{10} - a_{11}$	$c_9 = b_{10} + b_{11}$	$S_9 = (c_8 - c_9)/2$
$a_{10} = W_1 - W_5$	$b_{10} = a_{12} - a_{13}$	$c_{10} = b_8 - b_9$	$S_{10} = (c_{10} - c_{11})/2$
$a_{11} = W_9 - W_{13}$	$b_{11} = a_{14} - a_{15}$	$c_{11} = b_{10} - b_{11}$	$S_{11} = (c_{10} + c_{11})/2$
$a_{12} = W_2 - W_6$	$b_{12} = a_8 + a_9$	$c_{12} = b_{12} + b_{13}$	$S_{12} = (c_{12} + c_{13})/2$
$a_{13} = W_{10} - W_{14}$	$b_{13} = a_{10} + a_{11}$	$c_{13} = b_{14} + b_{15}$	$S_{13} = (c_{12} - c_{13})/2$
$a_{14} = W_3 - W_7$	$b_{14} = a_{12} + a_{13}$	$c_{14} = b_{12} - b_{13}$	$S_{14} = (c_{14} - c_{15})/2$
$a_{15} = W_{11} - W_{15}$	$b_{15} = a_{14} + a_{15}$	$c_{15} = b_{14} - b_{15}$	$S_{15} = (c_{14} + c_{15})/2$

Fonte: (AGOSTINI, 2007)

Após transformados, os valores são, por fim, somados conforme demonstrado em (33), gerando o valor final de SATD. Em (33), **m** e **n** são as dimensões do bloco transformado e **S_i** representa o bloco de coeficientes calculados a partir da Tabela 4.5.

$$SATD = \sum_i^{m \times n} |S_i| \quad (33)$$

As Tabelas 4.6 e 4.7 exibem os resultados em taxa de bits e PSNR obtidos para cada um dos vídeos codificados com a decisão de modo baseada no valor de SATD para blocos intra-quadro 4x4 e intra-quadro 16x16, respectivamente. Em comparação à aplicação da técnica de RDO, a decisão baseada em SATD gerou um aumento médio de 0,6% na taxa de bits e uma redução de 0,26 dB na qualidade da imagem para blocos intra-quadro 4x4, conforme mostra a Tabela 4.6, e um aumento médio de 0,1% na taxa de bits, sem redução na qualidade da imagem, para blocos intra-quadro 16x16, conforme mostra a Tabela 4.7. Como se pode perceber, a utilização desta métrica apresenta resultados bastante superiores aos da decisão baseada em SAD, ao preço de uma complexidade mais elevada.

Tabela 4.6: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão intra-quadro 4x4 baseada em RDO e decisão intra-quadro 4x4 baseada em SATD.

Vídeo	Decisão RDO		Decisão por SAD		RDO x SATD	
	Taxa de bits	PSNR (dB)	Taxa de bits	PSNR (dB)	+ Taxa de bits (%)	+ PSNR (dB)
<i>BUS</i>	1.371.896	36,39	1.384.656	36,06	0,93	-0,33
<i>CITY</i>	1.195.576	36,23	1.203.472	35,93	0,66	-0,30
<i>CREW</i>	507.792	39,07	509.000	38,93	0,24	-0,14
<i>FOOTBALL</i>	1.381.480	36,21	1.387.056	35,91	0,40	-0,30
<i>FOREMAN</i>	664.696	38,08	668.688	37,95	0,60	-0,13
<i>ICE</i>	548.192	40,23	550.176	40,00	0,36	-0,23
<i>MOBILE</i>	2.267.080	36,27	2.297.064	35,90	1,32	-0,38
<i>SOCCER</i>	1.124.104	35,95	1.127.000	35,68	0,26	-0,27
MÉDIA	1.132.602	37,30	1.140.889	37,04	0,60	-0,26

Tabela 4.7: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão intra-quadro 16x16 baseada em RDO e decisão intra-quadro 16x16 baseada em SATD.

Vídeo	Decisão RDO		Decisão por SAD		RDO x SATD	
	Taxa de bits	PSNR (dB)	Taxa de bits	PSNR (dB)	+ Taxa de bits (%)	+ PSNR (dB)
<i>BUS</i>	1.531.264	36,00	1.532.112	36,00	0,06	0,00
<i>CITY</i>	1.337.528	35,94	1.338.832	35,94	0,10	0,00
<i>CREW</i>	598.968	38,99	599.752	38,99	0,13	0,00
<i>FOOTBALL</i>	1.563.352	35,81	1.564.496	35,81	0,07	0,00
<i>FOREMAN</i>	888.880	37,80	890.512	37,80	0,18	0,00
<i>ICE</i>	663.928	40,10	664.672	40,10	0,11	0,00
<i>MOBILE</i>	2.507.528	35,80	2.508.608	35,80	0,04	0,00
<i>SOCCER</i>	1.235.720	35,63	1.236.712	35,63	0,08	0,00
MÉDIA	1.290.896	37,01	1.291.962	37,01	0,10	0,00

4.2.1.3 Decisão Baseada na Soma das Diferenças Quadráticas (SSD)

A Soma das Diferenças Quadráticas (SSD – *Sum of Squared Differences*) é uma métrica mais complexa que as anteriores, pois envolve multiplicações, além de somas sucessivas. A SSD é calculada através da equação apresentada em (34), onde \mathbf{P} é o bloco candidato predito, \mathbf{O} é o bloco original e \mathbf{m} e \mathbf{n} são as dimensões do bloco em número de amostras.

$$SSD = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (P_{i,j} - O_{i,j})^2 \quad (34)$$

As Tabelas 4.8 e 4.9 exibem os resultados para a decisão de modo baseada no valor de SSD para blocos intra-quadro 4x4 e intra-quadro 16x16, respectivamente. A decisão baseada em SSD gerou um aumento médio de 2,56% na taxa de bits e uma redução de 0,28 dB na qualidade da imagem para blocos intra-quadro 4x4, conforme mostra a Tabela 4.8. Para blocos intra-quadro 16x16, houve um aumento médio de 1,31% na taxa de bits, sem redução na qualidade da imagem, conforme mostra a Tabela 4.9.

Tabela 4.8: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão intra-quadro 4x4 baseada em RDO e decisão intra-quadro 4x4 baseada em SSD.

Vídeo	Decisão RDO		Decisão por SSD		RDO x SSD	
	Taxa de bits	PSNR (dB)	Taxa de bits	PSNR (dB)	+ Taxa de bits (%)	+ PSNR (dB)
<i>BUS</i>	1.371.896	36,39	1.405.880	36,01	2,48	-0,38
<i>CITY</i>	1.195.576	36,23	1.226.760	35,87	2,61	-0,36
<i>CREW</i>	507.792	39,07	526.552	38,90	3,69	-0,16
<i>FOOTBALL</i>	1.381.480	36,21	1.409.344	35,82	2,02	-0,39
<i>FOREMAN</i>	664.696	38,08	688.152	37,94	3,53	-0,15
<i>ICE</i>	548.192	40,23	558.640	40,18	1,91	-0,05
<i>MOBILE</i>	2.267.080	36,27	2.315.768	35,89	2,15	-0,38
<i>SOCCER</i>	1.124.104	35,95	1.148.112	35,61	2,14	-0,34
MÉDIA	1.132.602	37,30	1.159.901	37,03	2,56	-0,28

Tabela 4.9: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão intra-quadro 16x16 baseada em RDO e decisão intra-quadro 16x16 baseada em SSD.

Vídeo	Decisão RDO		Decisão por SSD		RDO x SSD	
	Taxa de bits	PSNR (dB)	Taxa de bits	PSNR (dB)	+ Taxa de bits (%)	+ PSNR (dB)
<i>BUS</i>	1.531.264	36,00	1544440	36,01	0,86	0,01
<i>CITY</i>	1.337.528	35,94	1354416	35,94	1,26	0,00
<i>CREW</i>	598.968	38,99	608008	39,00	1,51	0,01
<i>FOOTBALL</i>	1.563.352	35,81	1579104	35,81	1,01	0,00
<i>FOREMAN</i>	888.880	37,80	903064	37,81	1,60	0,01
<i>ICE</i>	663.928	40,10	678760	40,10	2,23	0,00
<i>MOBILE</i>	2.507.528	35,80	2532008	35,79	0,98	0,00
<i>SOCCER</i>	1.235.720	35,63	1248384	35,63	1,02	0,01
MÉDIA	1.290.896	37,01	1306023	37,01	1,31	0,00

4.2.1.4 Comparação Entre Decisões Baseadas em Distorção

De modo geral, a redução na qualidade da imagem e o aumento da taxa de bits gerados pela utilização da decisão baseada em distorção foram bastante pequenos, levando em consideração a enorme diminuição da complexidade computacional envolvida no processo de decisão. Ao invés de realizar a decisão de melhor modo após percorrer o caminho composto pela predição intra-quadro, geração de resíduos, transformadas diretas, quantização direta, codificação de entropia, quantização inversa, transformadas inversas e reconstrução, a decisão é realizada após um caminho mais curto, composto apenas pela predição intra-quadro e pela geração de resíduos (ver Figura 2.2).

A métrica de distorção SAD é aquela que apresenta menor complexidade computacional, pois realiza apenas subtrações e somas sucessivas. Por outro lado, esta é a que apresentou o maior aumento na taxa de bits, em comparação à utilização da técnica de RDO (5,29%, no caso de blocos 4x4).

A métrica SATD apresenta uma complexidade maior no processo de decisão, pois acrescenta à SAD a complexidade computacional de uma transformada Hadamard direta. A qualidade da decisão realizada por esta métrica é, contudo, muito mais alta do que a decisão baseada em SAD (aumento de apenas 0,6% na taxa de bits, no caso de blocos 4x4). A decisão por esta métrica é ainda bastante mais simples que a decisão

RDO, já que o caminho percorrido para a realização da decisão é composto apenas pela predição intra-quadro, pela geração de resíduos e pela transformada Hadamard direta.

SSD utiliza multiplicadores ou somas sucessivas para elevar ao quadrado o resultado das subtrações, aumentando significativamente a complexidade em relação à SAD. Apesar disso, os resultados desta decisão ainda são piores do que os obtidos através da utilização de SATD (aumento de 2,56% na taxa de bits, para bloco 4x4).

A partir desta análise, conclui-se que a decisão baseada em SAD é mais recomendada para implementações em hardware em que a utilização de recursos lógicos é limitada e em que a baixa complexidade computacional é necessária para a realização de uma decisão rápida. SAD é mais simples e econômica que as demais métricas e, ao mesmo tempo, ainda apresenta uma baixa redução na eficiência da decisão. Caso a utilização de recursos não seja limitada, a métrica SATD é a mais recomendada. Todos os resultados apresentados nas próximas seções levam em consideração a utilização da métrica SAD.

4.2.2 Decisão de Melhor Tamanho de Bloco Intra Baseada em Heterogeneidade

Quanto maior o nível de detalhamento, também chamado de heterogeneidade, de uma região da imagem, maior tende a ser a variação da informação residual gerada após a predição intra-quadro. Uma grande variação dos valores dos resíduos gera, após as operações de transformada, coeficientes de altos valores concentrados nas regiões de baixas frequências do bloco (isto é, na região em torno da diagonal superior), o que aumenta a quantidade de bits utilizados para codificar o bloco (LEE, 2009). Além disso, valores maiores distribuídos no bloco transformado acabam por gerar uma maior perda de qualidade na imagem durante o passo de quantização. Portanto, em sequências de vídeo que contêm áreas de grande detalhamento, blocos de menores dimensões são mais adequados, pois permitem a identificação de áreas homogêneas em regiões menores (LEE, 2009).

Conforme explicado anteriormente, o software de referência do H.264/AVC executa todos os modos intra-quadro 4x4 e todos os modos intra-quadro 16x16 para encontrar o menor custo taxa-distorção. Contudo, a comparação entre todos os modos é impraticável quando a codificação em tempo-real para vídeos de alta resolução é uma exigência. Nesta seção, propõe-se uma decisão de tamanho de bloco independente dos resultados da predição. Assim, a escolha entre os modos com mesmo tamanho de bloco continua sendo realizada apenas com base na distorção, conforme proposto na seção 4.2.1, e a escolha de melhor tamanho de bloco passa a ser realizada em paralelo, com base na análise da heterogeneidade do bloco original.

4.2.2.1 Desenvolvimento de uma Aproximação Inteira para a DCT 16x16

O padrão H.264/AVC utiliza as transformadas Hadamard e Discreta do Cosseno (DCT) com a finalidade de concentrar a maior parte da informação da imagem nas componentes de mais alta frequência do bloco. Os resultados obtidos com a DCT são, em geral, mais satisfatórios que os obtidos com a Hadamard. A DCT opera sobre um bloco de entrada \mathbf{X} de $N \times N$ amostras e gera um bloco \mathbf{Y} de $N \times N$ coeficientes transformados. As operações de uma transformada DCT podem ser descritas em termos de uma matriz de transformação \mathbf{A} . A DCT direta de um bloco de amostras é dada por (35), onde \mathbf{X} é a matriz de amostras, \mathbf{Y} é a matriz de coeficientes e \mathbf{A} é a matriz de transformação.

$$Y = AXA^T \quad (35)$$

Em (35), os elementos que compõem a matriz de transformação \mathbf{A} e a sua transposta \mathbf{A}^T são calculados de acordo com (36).

$$A_{ij} = C_i \cos \frac{(2j+1)i\pi}{2N}, \quad \text{onde } C_i = \sqrt{\frac{1}{N}} \text{ se } (i=0) \text{ e } C_i = \sqrt{\frac{2}{N}} \text{ se } (i>0) \quad (36)$$

O padrão H.264/AVC implementa apenas a DCT 4x4, que processa os resíduos das duas previsões (inter-quadros e intra-quadro). Para o desenvolvimento da heurística de identificação de heterogeneidade, uma matriz de transformação \mathbf{A} da DCT 16x16 precisou ser calculada com base nas equações (35) e (36). A matriz \mathbf{A} resultante possui 256 elementos diferentes; contudo, como a função cosseno é simétrica e se repete a cada 2π , os valores da matriz de transformação \mathbf{A} podem ser simplificados. Todos os elementos que compõem a matriz \mathbf{A} foram, portanto, reduzidos ao primeiro quadrante do círculo trigonométrico, diminuindo o número de elementos de 256 para 16. A matriz \mathbf{A} após simplificações e redução ao primeiro quadrante está apresentada na Figura 4.1, na qual os elementos de \mathbf{a} a \mathbf{q} representam os valores expressos nas equações de (37) a (52).

a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a
b	c	d	e	f	g	h	i	-i	-h	-g	-f	-e	-d	-c	-b
j	l	m	n	-n	-m	-l	-j	j	l	m	n	-n	-m	-l	-j
c	f	i	-g	-d	-b	-e	-h	h	e	b	d	g	-i	-f	-c
o	p	-p	-o	-o	-p	p	o	o	p	-p	-o	-o	-p	p	o
d	i	-e	-c	-h	f	b	g	-g	-b	-f	h	c	e	-i	-d
l	-n	-j	-m	m	j	n	-l	-l	n	j	m	-m	-j	-n	l
e	-g	-c	i	b	h	-d	-f	f	d	-h	-b	-i	c	g	-e
q	-q	-q	q	q	-q	-q	q	q	-q	-q	q	q	-q	-q	q
f	-d	-h	b	-i	-c	g	e	-e	-g	c	i	-b	h	d	-f
m	-j	n	l	-l	-n	j	-m	-m	j	-n	-l	l	n	-j	m
g	-b	f	h	-c	e	i	-d	d	-i	-e	c	-h	-f	b	-g
p	-o	o	-p	-p	o	-o	p	p	-o	o	-p	-p	o	-o	p
h	-e	b	-d	g	i	-f	c	-c	f	-i	-g	d	-b	e	-h
n	-m	l	-j	j	-l	m	-n	-n	m	-l	j	-j	l	-m	n
i	-h	g	-f	e	-d	c	-b	b	-c	d	-e	f	-g	h	-i

Figura 4.1: Matriz de transformação \mathbf{A} após simplificações e reduções ao primeiro quadrante.

$$a = \frac{1}{4} = 0,25 \quad (37)$$

$$f = \frac{1}{2} \cos \frac{9\pi}{32} \approx 0,32 \quad (42)$$

$$b = \frac{1}{2} \cos \frac{\pi}{32} \approx 0,5 \quad (38)$$

$$g = \frac{1}{2} \cos \frac{11\pi}{32} \approx 0,24 \quad (43)$$

$$c = \frac{1}{2} \cos \frac{3\pi}{32} \approx 0,48 \quad (39)$$

$$h = \frac{1}{2} \cos \frac{13\pi}{32} \approx 0,15 \quad (44)$$

$$d = \frac{1}{2} \cos \frac{5\pi}{32} \approx 0,44 \quad (40)$$

$$i = \frac{1}{2} \cos \frac{15\pi}{32} \approx 0,05 \quad (45)$$

$$e = \frac{1}{2} \cos \frac{7\pi}{32} \approx 0,39 \quad (41)$$

$$j = \frac{1}{2} \cos \frac{2\pi}{32} \approx 0,49 \quad (46)$$

$$l = \frac{1}{2} \cos \frac{6\pi}{32} \approx 0,42 \quad (47)$$

$$m = \frac{1}{2} \cos \frac{10\pi}{32} \approx 0,28 \quad (48)$$

$$n = \frac{1}{2} \cos \frac{14\pi}{32} \approx 0,1 \quad (49)$$

$$o = \frac{1}{2} \cos \frac{4\pi}{32} \approx 0,46 \quad (50)$$

$$p = \frac{1}{2} \cos \frac{12\pi}{32} \approx 0,19 \quad (51)$$

$$q = \frac{1}{2} \cos \frac{8\pi}{32} \approx 0,35 \quad (52)$$

A heurística utilizada consiste em identificar a heterogeneidade do bloco 16x16 transformado através da análise da concentração dos coeficientes nas regiões de baixa frequência do bloco. Caso a soma dos coeficientes distribuídos nas regiões de baixa frequência, com exceção do primeiro coeficiente (DC), possua um alto valor, a imagem tende a apresentar um grande nível de detalhamento e deve ser particionada em blocos de 4x4 pixels. Caso contrário, deve ser codificada como um bloco de 16x16 pixels.

Para fins de implementação da transformada em hardware, uma aproximação da DCT foi elaborada com base na matriz apresentada na Figura 4.1. Aos moldes das aproximações inteiras realizadas com as transformadas Hadamard e DCT 4x4 implementadas no padrão H.264/AVC (RICHARDSON, 2003), a matriz de transformação \mathbf{A} da DCT 16x16 foi fatorada, de forma que os seus valores pudessem ser representados na forma de número inteiro ou de fração de inteiro com denominador em potência de 2 (isto é, obtível através de simples deslocamentos). A DCT fatorada pode ser representada através da equação (53), na qual \mathbf{CXC}^T é o núcleo da transformada e \mathbf{E} é a matriz de escala obtida através da fatoração da matriz \mathbf{A} . O sinal \otimes indica uma multiplicação escalar.

$$Y = (C X C^T) \otimes E \quad (53)$$

A Figura 4.2 apresenta a matriz de transformação \mathbf{C} obtida após a fatoração, composta apenas por números inteiros e divisões possíveis através de deslocamentos. Como se pode perceber, os 16 valores diferentes apresentados na Figura 4.1 foram reduzidos, por aproximação, a apenas 8 valores diferentes.

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1/2	1/2	1/4	1/8	-1/8	-1/4	-1/2	-1/2	-1	-1	-1	-1
1	1	1/2	1/8	-1/8	-1/2	-1	-1	1	1	1/2	1/8	1/8	-1/2	-1	-1
1	1/2	1/8	-1/2	1	-1	-1	-1/4	1/4	1	1	1	1/2	-1/8	-1/2	-1
1	1/2	-1/2	-1	-1	-1/2	1/2	1	1	1/2	-1/2	-1	-1	-1/2	1/2	-1
1	1/8	-1	-1	-1/4	1/2	1	1/2	-1/2	-1	-1/2	1/4	1	1	-1/8	-1
1	-1/8	-1	-1/2	1/2	1	1/8	-1	-1	1/8	1	1/2	-1/2	-1	-1/8	1
1	-1/2	-1	1/8	1	1/4	-1	-1/2	1/2	1	-1/4	-1	-1/8	1	1/2	-1
1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1
1/2	-1	-1/4	1	-1/8	-1	1/2	1	-1	-1/2	1	1/8	-1	1/4	1	-1/2
1/2	-1	1/8	1	-1	-1/8	1	-1/2	-1/2	1	-1	-1/8	1	1/8	-1	1/2
1/2	-1	1/2	1/4	-1	1	1/8	-1	1	-1/8	-1	1	-1/4	-1/2	1	-1/2
1/2	-1	1	-1/2	-1/2	-1	-1	1/2	1/2	-1	-1	-1/2	-1/2	1	-1	1/2
1/4	-1	1	-1	1/2	1/8	-1/2	1	-1	1/2	-1/8	-1/2	1	-1	1	-1/4
1/8	-1/2	1	-1	1	-1	1/2	-1/8	-1/8	1/2	-1	1	-1	1	-1/2	1/8
1/8	-1/4	1/2	-1/2	1	-1	1	-1	1	-1	1	-1	1/2	-1/2	1/4	-1/8

Figura 4.2: Matriz de transformação \mathbf{C} obtida a partir da matriz de transformação \mathbf{A} através de fatoração.

A Figura 4.3 mostra os elementos que compõem a matriz de escala **E**, também obtida através do processo de fatoração. Os valores de **a** a **q** correspondem aos mesmos valores definidos nas equações de (37) a (52).

a ²	ba	ja	ba	oa	ba	ja	ba	qa	ba	ja	ba	oa	ba	ja	ba
ab	b ²	jb	b ²	ob	b ²	jb	b ²	qb	b ²	jb	b ²	ob	b ²	jb	b ²
aj	bj	j ²	bj	oj	bj	j ²	bj	qj	bj	j ²	bj	oj	bj	j ²	bj
ab	b ²	jb	b ²	ob	b ²	jb	b ²	qb	b ²	jb	b ²	ob	b ²	jb	b ²
ao	bo	jo	bo	o ²	bo	jo	bo	qo	bo	jo	bo	o ²	bo	jo	bo
ab	b ²	jb	b ²	ob	b ²	jb	b ²	qb	b ²	jb	b ²	ob	b ²	jb	b ²
aj	bj	j ²	bj	oj	bj	j ²	bj	qj	bj	j ²	bj	oj	bj	j ²	bj
ab	b ²	jb	b ²	ob	b ²	jb	b ²	qb	b ²	jb	b ²	ob	b ²	jb	b ²
aq	bq	jq	bq	oq	bq	jq	bq	q ²	bq	jq	bq	oq	bq	jq	bq
ab	b ²	jb	b ²	ob	b ²	jb	b ²	qb	b ²	jb	b ²	ob	b ²	jb	b ²
aj	bj	j ²	bj	oj	bj	j ²	bj	qj	bj	j ²	bj	oj	bj	j ²	bj
ab	b ²	jb	b ²	ob	b ²	jb	b ²	qb	b ²	jb	b ²	ob	b ²	jb	b ²
ao	bo	jo	bo	o ²	bo	jo	bo	qo	bo	jo	bo	o ²	bo	jo	bo
ab	b ²	jb	b ²	ob	b ²	jb	b ²	qb	b ²	jb	b ²	ob	b ²	jb	b ²
aj	bj	j ²	bj	oj	bj	j ²	bj	qj	bj	j ²	bj	oj	bj	j ²	bj
ab	b ²	jb	b ²	ob	b ²	jb	b ²	qb	b ²	jb	b ²	ob	b ²	jb	b ²

Figura 4.3: Matriz de escala **E** obtida a partir da matriz de transformação **A** através de fatoração.

Após todas as simplificações e fatorações realizadas, os valores que compõem a matriz **C** (e a sua transposta **C^T**) do núcleo da transformada (**CXC^T**) podem ser obtidos apenas através de subtrações, somas e deslocamentos. As equações para a obtenção dos 256 termos do núcleo da transformada foram obtidas através do software Maple 14, com base nas matrizes **C** e **C^T** calculadas e considerando um bloco com amostras de entrada composto por variáveis de **X₀** a **X₂₅₆**. Para fins de elucidação, a equação (54) apresenta o cálculo realizado para a obtenção do quinto elemento da primeira linha da matriz de coeficientes **Y**.

$$\begin{aligned}
Y_4 = & -(X_{100}/4) + (X_{203}/4) + (X_{187}/4) + (X_{171}/4) + (X_{235}/4) + (X_{219}/4) + (X_{107}/4) - (X_{250}/2) + (X_{139}/4) + \\
& (X_{155}/4) + (X_{123}/4) + X_{204} + X_{172} + X_{188} + X_{236} + X_{220} + X_{108} - (X_{154}/2) - X_{169} - (X_{120}/2) + X_{150} + (X_{197}/2) + \\
& (X_{181}/2) + (X_{149}/2) + (X_{133}/2) + (X_{117}/2) + (X_{101}/2) - (X_{244}/4) - (X_{228}/4) - (X_{212}/4) - (X_{196}/4) - (X_{180}/4) - \\
& (X_{164}/4) - (X_{148}/4) - (X_{132}/4) - (X_{116}/4) + X_{112} + X_{128} + X_{144} + X_{160} + X_{176} + X_{192} + X_{208} + X_{224} + X_{240} + \\
& (X_{113}/8) + (X_{129}/8) + (X_{145}/8) + (X_{161}/8) + (X_{177}/8) + (X_{193}/8) + (X_{209}/8) + (X_{225}/8) + (X_{241}/8) - X_{114} - X_{130} - \\
& X_{146} - X_{162} - X_{178} - X_{194} - X_{210} - X_{226} - X_{242} - X_{115} - X_{131} - X_{147} - X_{163} - X_{179} - X_{195} - X_{211} + (X_{251}/4) + X_{156} \\
& + X_{140} + X_{124} + (X_{231}/2) - (X_{104}/2) + (X_{247}/2) + (X_{199}/2) + (X_{183}/2) + (X_{215}/2) - (X_{216}/2) - (X_{232}/2) - X_{105} - \\
& (X_{248}/2) - (X_{152}/2) - (X_{136}/2) - (X_{184}/2) - (X_{200}/2) - (X_{168}/2) + X_{166} + X_{134} + X_{198} + X_{182} + (X_{229}/2) + \\
& (X_{213}/2) + X_{102} + X_{118} + (X_{245}/2) + (X_{135}/2) + (X_{151}/2) + (X_{119}/2) + (X_{167}/2) + X_{230} + X_{246} + X_{214} + (X_{103}/2) \\
& - (X_{170}/2) - (X_{186}/2) - (X_{138}/2) - (X_{122}/2) - (X_{218}/2) - (X_{202}/2) - (X_{234}/2) - X_{153} - X_{185} - X_{137} - X_{121} - X_{249} - \\
& X_{233} - (X_{106}/2) - X_{201} - X_{239} - X_{223} - X_{207} - X_{191} - X_{255} - (X_{110}/8) - (X_{142}/8) - (X_{126}/8) + X_{237} - (X_{238}/8) + X_{253} \\
& - (X_{158}/8) + X_{252} + X_{221} + X_{205} + X_{109} - (X_{254}/8) - X_{111} - (X_{222}/8) - X_{175} - (X_{174}/8) - (X_{190}/8) - X_{127} - (X_{206}/8) \\
& + X_{157} - X_{159} + X_{173} - X_{143} - X_{217} + (X_{165}/2) + X_{125} + X_{141} + X_{189} - X_9 - (X_{88}/2) - (X_{72}/2) - (X_{56}/2) - (X_{40}/2) - \\
& (X_{24}/2) - (X_8/2) + (X_{87}/2) + (X_{71}/2) + X_0 + X_{16} + X_{32} + X_{48} + X_{64} + X_{80} + X_{96} + (X_1/8) + (X_{17}/8) + (X_{33}/8) \\
& + (X_{49}/8) + (X_{65}/8) + (X_{81}/8) + (X_{97}/8) - X_2 - X_{18} - X_{34} - X_{50} - X_{66} - X_{82} - X_{98} - X_3 - X_{19} - X_{35} - X_{51} - X_{67} - \\
& X_{83} - X_{99} - (X_4/4) - (X_{20}/4) - (X_{36}/4) - (X_{52}/4) - (X_{68}/4) - (X_{84}/4) + (X_5/2) + (X_{21}/2) + (X_{37}/2) + (X_{53}/2) + \\
& (X_{69}/2) + (X_{85}/2) + X_6 + X_{22} + X_{38} + X_{54} + X_{70} + X_{86} + (X_7/2) + (X_{23}/2) + (X_{39}/2) + (X_{55}/2) - X_{243} - X_{227} - \\
& X_{25} - X_{41} - X_{57} - X_{73} - X_{89} - (X_{10}/2) - (X_{26}/2) - (X_{42}/2) - (X_{58}/2) - (X_{74}/2) - (X_{90}/2) + (X_{11}/4) + (X_{27}/4) + \\
& (X_{43}/4) + (X_{59}/4) + (X_{75}/4) + (X_{91}/4) + X_{12} + X_{28} + X_{44} + X_{60} + X_{76} + X_{92} + X_{13} + X_{29} + X_{45} + X_{61} + X_{77} + \\
& X_{93} - (X_{14}/8) - (X_{30}/8) - (X_{46}/8) - (X_{62}/8) - (X_{78}/8) - (X_{94}/8) - X_{15} - X_{31} - X_{47} - X_{63} - X_{79} - X_{95}
\end{aligned} \tag{54}$$

Como se pode perceber, apesar de todas as simplificações realizadas, o número de operações aritméticas associadas ao cálculo de apenas um coeficiente é extremamente grande. Contudo, maiores simplificações podem ser realizadas em nível de implementação de hardware, conforme será explicado no próximo capítulo desta dissertação. Além disso, nem todos os coeficientes precisam ser calculados para a análise da heterogeneidade, conforme explicado na próxima seção.

4.2.2.2 Cálculo do Nível de Heterogeneidade da Imagem

A análise do nível de heterogeneidade de uma imagem é realizada apenas sobre a região de baixa frequência do bloco e, portanto, apenas alguns coeficientes transformados precisam ser levados em consideração no cálculo. Uma série de testes mostrou que a utilização dos coeficientes que compõem a primeira linha e a primeira coluna do bloco, com exceção do elemento DC (primeiro elemento), é suficiente para definir o grau de heterogeneidade da imagem, conforme proposto por (LEE, 2009). Isto significa que a DCT completa não precisa ser implementada para que a decisão seja realizada, já que apenas 30 dos 256 coeficientes precisam ser totalmente calculados. Além disso, a utilização dos coeficientes que compõem a primeira linha e a primeira coluna do bloco facilita o compartilhamento de recursos lógicos durante a implementação da solução como uma arquitetura de hardware, conforme será explicado no próximo capítulo.

O somatório realizado para o cálculo da heterogeneidade com base nos coeficientes de baixa frequência está apresentado em (55), onde \mathbf{H} representa o grau de heterogeneidade do bloco e \mathbf{Y} representa o bloco transformado. Quanto mais alto o valor de \mathbf{H} , mais heterogêneo tende a ser o bloco original.

$$H = \sum_{i=1}^{15} |Y(i,0)| + \sum_{j=1}^{15} |Y(0,j)| \quad (55)$$

Após exaustivas execuções do software de referência com vídeos de resolução CIF, valores médios de \mathbf{H} foram encontrados para cada um dos tamanhos de bloco intra-quadro. A partir desses valores, um conjunto de valores limite foi derivado para testar a escolha de blocos 4x4 e 16x16. Todas as decisões de tamanho de bloco intra-quadro realizadas pelo software de referência através da técnica de RDO foram salvas e comparadas com as decisões realizadas quando a heurística foi utilizada. Após um grande número de simulações nas quais o valor limite entre a escolha de blocos 4x4 e 16x16 variou entre 2000 e 9000, verificou-se que o maior número de decisões semelhantes às realizadas com RDO foi obtido com um valor limite 2800.

O algoritmo que define o melhor tamanho de bloco intra-quadro pode ser definido, resumidamente, de acordo com o algoritmo abaixo. Este algoritmo considera que o melhor modo intra-quadro é uma escolha entre o melhor modo intra-quadro 16x16 e o melhor modo 4x4, ambos definidos no passo de decisão anterior com base na distorção.

$$H = \sum_{i=1}^{15} |Y(i,0)| + \sum_{j=1}^{15} |Y(0,j)|$$

Se $H(\text{macrobloco_atual}) < 2800$

$\text{melhor_modo_INTRA} = \text{melhor_modo_I16MB}$

Senão

$\text{melhor_modo_INTRA} = \text{melhor_modo_I4MB}$

-- (macrobloco é formado por 16 blocos 4x4)

A Tabela 4.10 exibe os resultados para a decisão combinada de melhor modo intra-quadro baseada em distorção por SAD (melhor modo 4x4 e melhor modo 16x16) e em heterogeneidade (melhor tamanho de bloco). Como se pode perceber, a utilização das duas técnicas combinadas apresentou um aumento médio de 3,96% na taxa de bits e uma redução de apenas 0,3 dB na qualidade da imagem, em comparação à decisão baseada em RDO.

Tabela 4.10: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão intra-quadro baseada em distorção e heterogeneidade do macrobloco original.

Vídeo	Decisão RDO		Decisão por SAD & DCT		RDO x SAD & DCT	
	Taxa de bits	PSNR (dB)	Taxa de bits	PSNR (dB)	+ Taxa de bits (%)	+ PSNR (dB)
<i>BUS</i>	1.369.328	36,39	1.417.912	36,04	3,55	-0,35
<i>CITY</i>	1.193.560	36,24	1.241.008	35,87	3,98	-0,37
<i>CREW</i>	491.760	39,14	534.240	38,91	8,64	-0,23
<i>FOOTBALL</i>	1.380.760	36,22	1.426.184	35,85	3,29	-0,37
<i>FOREMAN</i>	652.568	38,11	700.704	37,92	7,38	-0,19
<i>ICE</i>	500.320	40,45	533.760	40,28	6,68	-0,17
<i>MOBILE</i>	2.264.904	36,27	2.332.496	35,88	2,98	-0,39
<i>SOCCER</i>	1.115.384	35,95	1.137.064	35,64	1,94	-0,31
MÉDIA	1.121.073	37,35	1.165.421	37,05	3,96	-0,30

4.3 Decisão Rápida de Modo Inter-Quadros

Assim como na decisão rápida de modo intra-quadro, a decisão inter-quadros foi dividida com a finalidade de simplificar um problema complexo em passos mais simples. As decisões inter-quadros propostas também não são interdependentes e podem, portanto, ser realizadas em paralelo quando implementadas em hardware. Ao final de todos os processos de decisão, os seus resultados são combinados para gerar uma decisão única de melhor modo.

A decisão de modo inter-quadros consiste na escolha do melhor particionamento e do melhor sub-particionamento de macrobloco, explicados no capítulo 3 e apresentados nas Figuras 3.7 e 3.8, respectivamente. No caso de aplicações em software, a análise prévia da ocorrência média dos modos de codificação em quadros P pode auxiliar na definição de uma ordem preferencial de avaliação e decisão, diminuindo o tempo de execução necessário para o processo de decisão. A partir de testes realizados com vídeos de resolução CIF codificados com base na decisão RDO, as frequências médias foram obtidas para cada modo, conforme apresentado na Tabela 4.11.

Tabela 4.11: Ocorrência de modos de codificação em quadros P de resolução CIF.

Modo	Ocorrências	Frequência
<i>SKIP</i>	83.645	29.1%
<i>P16x16</i>	83.651	29.1%
<i>P16x8</i>	30.364	10.6%
<i>P8x16</i>	33.187	11.5%
<i>SMB8x8</i>	27.251	9.5%
<i>SMB8x4</i>	7.700	2.7%
<i>SMB4x8</i>	8.472	2.9%
<i>SMB4x4</i>	2.331	0.8%
<i>Intra-quadro</i>	10.895	3.8%

Como se pode perceber na Tabela 4.11, os modos *SKIP* e *P16x16* constituem, juntos, quase 60% da totalidade de modos escolhidos. Os modos *P16x8* e *P8x16* aparecem em seguida, com ocorrência em semelhante frequência e os modos de sub-particionamento aparecem por último. Quanto maior o tamanho do particionamento (e do sub-particionamento) do macrobloco, maior é a sua ocorrência, o que confirma a característica dos vídeos digitais possuírem bastantes áreas homogêneas e estacionárias.

A sequência hierárquica de decisão inter-quadros proposta neste trabalho está apresentada na Figura 4.4. Como explicado anteriormente, em casos de implementações em software, a ordem proposta pode diminuir o tempo necessário para a realização da decisão inter-quadros. Em casos de implementações em hardware, todos os níveis de decisão podem ser realizados em paralelo, embora a ordem hierárquica possa auxiliar em implementações sequenciais e que visem à redução do consumo de energia.

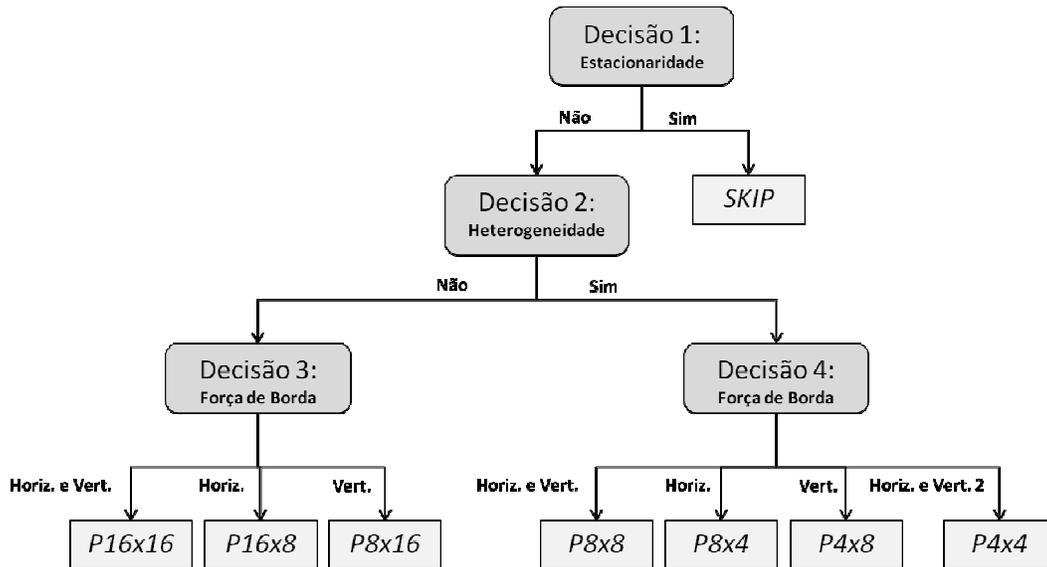


Figura 4.4: Hierarquia proposta para decisão de tamanho de partição e sub-partição de macrobloco (modos inter-quadros).

As decisões apresentadas na Figura 4.4 estão detalhadas nas próximas seções deste capítulo. A Decisão 1 realiza a primeira análise sobre o macrobloco original e o macrobloco de referência, detectando regiões estacionárias entre um quadro e outro do vídeo. A Decisão 2, baseada em análise de heterogeneidade, é semelhante à decisão de tamanho de bloco intra-quadro e define se serão utilizadas sub-partições de macrobloco ou não. As Decisões 3 e 4 são semelhantes e analisam a diferença entre os pixels que compõem o macrobloco original para detectar bordas horizontais e verticais na imagem.

Apesar de não avaliada neste trabalho, a predição fracionária pode ser suportada pelo conjunto de heurísticas proposto para a decisão inter-quadros. A decisão pelo uso de *quarter* e *half* pixel pode ser realizada, por exemplo, de acordo com uma heurística simples baseada apenas na distorção entre a imagem predita e a imagem original para um determinado tamanho de bloco. Caso a distorção seja menor quando alguma das predições fracionárias é utilizada, a predição utilizando predição inteira é descartada. A decisão entre predição inteira e fracionária integrada com as heurísticas apresentadas nas próximas seções será avaliada em trabalhos futuros.

Todos os testes apresentados nas próximas seções foram realizados com vídeos de resolução HD 1080p (1920x1088 pixels).

4.3.1 Decisão 1: Detecção de Modo *SKIP* Baseada em Estacionaridade

Sequências de vídeos apresentam uma característica natural de serem compostas por movimentos constantes e, principalmente, por regiões temporalmente estacionárias. Esta característica é bastante notória, em especial, nas áreas de fundo das imagens (WU, 2005). Diversas técnicas são utilizadas para a detecção prévia de modo *SKIP*, como o cálculo adiantado do custo taxa-distorção (LIU, 2008) (MA, 2008), a previsão de blocos totalmente quantizados para zero (WANG, 2007) (KIM, 2004a) e a detecção de estacionaridade em uma sequência de quadros (WU, 2005) (YU, 2008).

Na decisão RDO, o modo *SKIP* é selecionado quando o custo taxa-distorção de não codificar qualquer resíduo é menor do que os custos dos outros modos possíveis, que codificam os resíduos. Quando a decisão RDO não é utilizada, a especificação do padrão H.264/AVC define que um conjunto de 4 condições sejam satisfeitas para que o modo *SKIP* seja utilizado. Contudo, de acordo com Yu (2008), uma das condições já é, por si só, um bom indicador da utilização do modo *SKIP*: os coeficientes transformados do modo *SKIP* são todos quantizados para zero. Sabe-se que quanto menor for a amplitude dos valores de resíduos, maior é a tendência de tais valores serem quantizados para zero após serem transformados. Com base nesta condição, alguns trabalhos vêm explorando a idéia de estacionaridade das sequências de quadros.

A estacionaridade é uma forma simples e eficiente de identificar o grau de similaridade entre cada quadro de um vídeo e o seu quadro de referência. A estacionaridade (em inglês, *stationarity*) é também chamada de imobilidade (ou *stillness*, em inglês) da sequência de quadros. No trabalho de Wu (WU, 2005), a estacionaridade é detectada a partir da comparação dos quadros temporalmente vizinhos através da utilização da métrica SAD. Caso o SAD seja menor que um limite calculado previamente, a região da imagem é considerada estacionária e o macrobloco é codificado com modo *SKIP*. Esta ideia foi adotada neste trabalho, conforme explicado a seguir.

Com a finalidade de encontrar um valor limite para ser usado na detecção de macroblocos estacionários, uma série de testes em software com vídeos de resolução HD 1080p foi realizada, considerando a decisão realizada com base na técnica de RDO. Para todos os vídeos analisados, o valor de SAD entre o macrobloco do quadro atual e o macrobloco co-localizado no quadro de referência foi calculado. A partir daí, os valores de SAD dos casos em que o modo escolhido foi *SKIP* e os valores de SAD dos casos em que o modo escolhido foi diferente de *SKIP* foram agrupados e ambas as médias foram calculadas. Os valores médios obtidos guiaram uma série de simulações que definiram o valor limite de SAD para a detecção de macroblocos estacionários. As simulações variaram os valores limite de 200 a 700 e possibilitaram a definição do valor 500 como o melhor limite entre macroblocos estacionários e não estacionários. Em outras palavras, na decisão utilizada neste trabalho, o SAD entre o macrobloco original e o macrobloco de referência é calculado e comparado com o valor 500. Caso o SAD seja menor que 500, o macrobloco é estacionário e deve ser codificado com o modo *SKIP*. Caso contrário, o macrobloco não é estacionário e deve ser codificado com outro modo inter-quadros.

A Tabela 4.12 apresenta os resultados obtidos com a decisão de modo *SKIP* baseada em análise de estacionaridade e os compara com a aplicação da técnica de RDO. É importante notar que apenas a decisão *SKIP* foi realizada com base na estacionaridade.

Caso a estacionaridade não seja detectada, o processo de RDO é ainda aplicado para selecionar um dentre os demais modos restantes. A aplicação do valor limite igual a 500 resultou num aumento médio de 0,04 dB na qualidade da imagem gerada e aumento de 1,75% na quantidade de bits utilizados para codificar o vídeo. Isto acontece porque a análise de estacionaridade permite uma decisão de modo aproximada, de forma que, em alguns vídeos, um número menor de macroblocos *SKIP* pode ser selecionado, em comparação à aplicação da técnica de RDO.

Tabela 4.12: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão de modo *SKIP* baseada em estacionaridade.

Vídeo	Decisão RDO		Decisão <i>SKIP</i> por Estacionaridade		RDO <i>x</i> Estacionaridade	
	Taxa de bits	PSNR (dB)	Taxa de bits	PSNR (dB)	+ Taxa de bits (%)	+ PSNR (dB)
<i>MAN IN CAR</i>	2.138.352	42,43	2.301.488	42,51	7,63	0,07
<i>RIVERBED</i>	15.729.560	38,71	15.585.336	38,71	-0,92	0,00
<i>ROLLING TOMATOES</i>	3.427.472	40,36	3.869.736	40,41	12,90	0,05
<i>RUSH HOUR</i>	3.711.680	41,47	3.859.688	41,53	3,99	0,05
<i>STATION 2</i>	4.407.176	39,64	4.586.352	39,70	4,07	0,06
<i>SUNFLOWER</i>	7.105.888	42,58	7.162.712	42,60	0,80	0,01
<i>TRACTOR</i>	11.344.312	38,92	11.393.720	38,94	0,44	0,01
<i>TRAFFIC</i>	8.657.232	39,37	8.752.600	39,40	1,10	0,04
MÉDIA	7.065.209	40,44	7.188.954	40,47	1,75	0,04

As Figuras 4.5 e 4.6 apresentam gráficos que comparam os resultados médios de codificação com decisão *SKIP* baseada em estacionaridade e com decisão RDO sem a utilização de macroblocos *SKIP*. Nos dois gráficos, Seq1-8 representam as 8 sequências de vídeo listadas na Tabela 4.12. As barras da Figura 4.5 representam o aumento ou a diminuição da taxa de bits, em termos percentuais, gerados por cada uma das abordagens, em comparação à aplicação da decisão RDO completa. Como se pode perceber, em alguns casos a heurística proposta gera uma taxa de bits inferior à utilização da decisão RDO completa. Nos casos em que isso acontece, a heurística levou à escolha de mais macroblocos *SKIP* do que a técnica de RDO e, portanto, menos resíduos foram codificados. Em geral, esta redução da taxa de bits vem acompanhada de uma diminuição da qualidade do vídeo, pois uma quantidade menor de resíduos é codificada, gerando uma distorção maior na imagem reconstruída.

Na Figura 4.6, a mesma comparação é realizada em termos de PSNR (em dB). Como se pode perceber, a utilização da decisão *SKIP* baseada em estacionaridade gerou uma redução na taxa de bits em todos os casos, em comparação à decisão RDO sem *SKIP*. Além disso, com exceção do primeiro vídeo, a qualidade da imagem (PSNR) ficou muito próxima à obtida com a técnica de RDO completa. Como já era esperado, a heurística sempre gera vídeos com qualidade inferior à técnica RDO sem *SKIP*, pois quanto menor o número de macroblocos codificados como *SKIP* em uma imagem, maior tende a ser a sua qualidade. Como a heurística proposta permite a utilização de macroblocos *SKIP*, ao contrário da técnica de RDO com modo *SKIP* desativado, a qualidade da imagem tende a ser sempre menor.

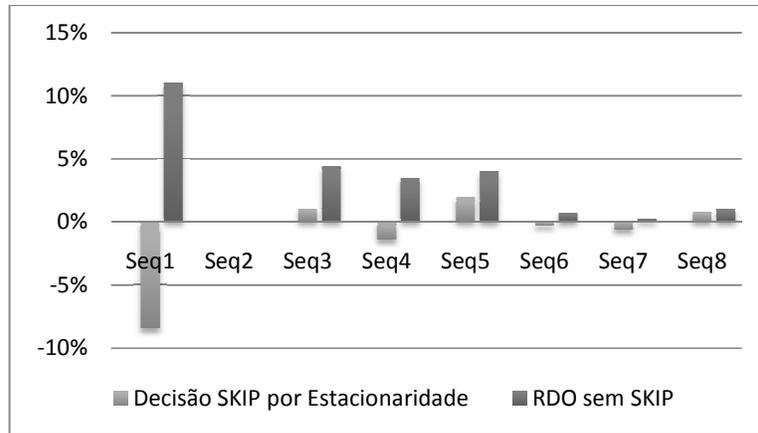


Figura 4.5: Aumento percentual da taxa de bits gerado pela decisão *SKIP* baseada em estacionaridade e pela decisão RDO sem modo *SKIP*, em relação à RDO completa.

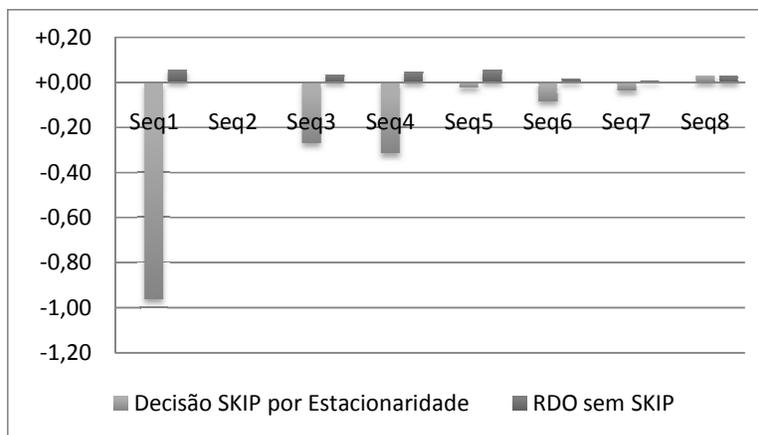


Figura 4.6: Aumento do PSNR (em dB) gerado pela decisão *SKIP* baseada em estacionaridade e pela decisão RDO sem modo *SKIP*, em relação à RDO completa.

4.3.2 Decisão 2: Detecção de Sub-Particionamento Baseada em Heterogeneidade

Além de serem compostas por muitas regiões estacionárias, sequências naturais de vídeo também são compostas em grande parte por regiões espacialmente homogêneas. De acordo com Yu (YU, 2004) e Choi (CHOI, 2006), essas regiões homogêneas tendem a apresentar movimentos em direções similares, enquanto que as regiões heterogêneas apresentam movimentos desordenados. Isto acontece porque as regiões homogêneas, em geral, pertencem a um objeto em comum da imagem, o qual é composto por partes que movem-se juntas, continuamente, na mesma direção e sentido. Assim, as regiões homogêneas são, em geral, codificadas em partições de macrobloco com tamanhos grandes, como 16x16, 16x8 e 8x16. Em uma implementação sequencial, se a heterogeneidade de um macrobloco pode ser identificada em um passo anterior à decisão, a análise de todos os outros tamanhos de partição menores pode ser evitada.

Como apresentado na seção 4.2.2, a heterogeneidade de um macrobloco pode ser detectada a partir da análise de alguns coeficientes gerados após a aplicação de uma transformada direta, como a Hadamard e a DCT. Outros trabalhos, como o de Shen (2008), propõem ainda a utilização do Operador de Sobel para criar um mapa de bordas e, a partir deste, detectar a heterogeneidade da imagem. O Operador de Sobel gera

resultados bastante eficientes, mas a sua complexidade computacional é maior que a de uma transformada parcial e, portanto, a sua utilização foi evitada neste trabalho.

A decisão de tamanho de bloco intra-quadro baseada em heterogeneidade foi utilizada como base na detecção de sub-particionamento de macrobloco. Testes semelhantes foram realizados com vídeos de resolução HD 1080p para encontrar valores médios de \mathbf{H} (ver equação (25)) nos casos em que o sub-particionamento foi utilizado e nos casos em que o sub-particionamento não foi utilizado.

A Figura 4.7 mostra um gráfico em que cada ponto representa um macrobloco da sequência de vídeo *MAN IN CAR* codificado como *P16x16*, *P16x8* ou *P8x16* (isto é, sem sub-particionamento) pela decisão RDO. O eixo x representa apenas a numeração do macrobloco e o eixo y apresenta os valores \mathbf{H} calculados para cada um dos macroblocos. Como se pode perceber, a grande maioria dos macroblocos não sub-particionados apresentaram um valor \mathbf{H} inferior a 10.000.

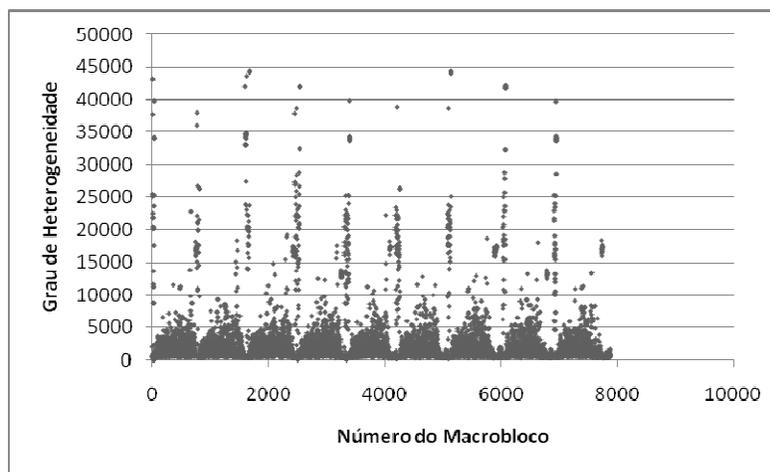


Figura 4.7: Valores de \mathbf{H} (heterogeneidade) de cada macrobloco da sequência *MAN IN CAR* codificado sem sub-particionamento através da decisão RDO completa.

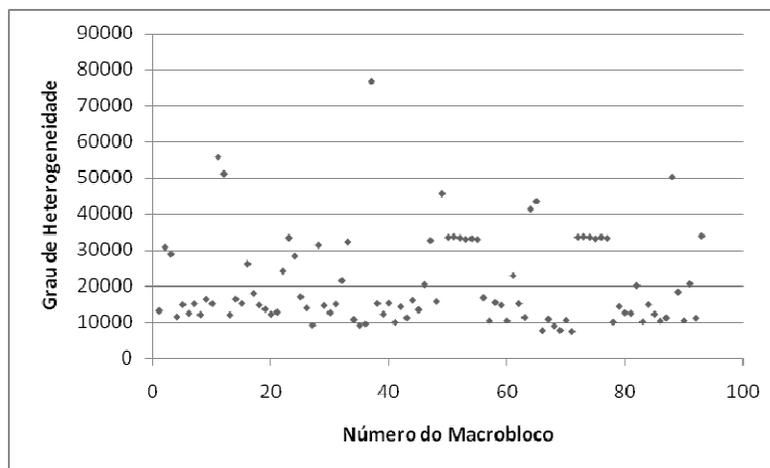


Figura 4.8: Valores de \mathbf{H} (heterogeneidade) de cada macrobloco da sequência *MAN IN CAR* codificado com sub-particionamento através da decisão RDO completa.

A Figura 4.8 apresenta um gráfico semelhante para macroblocos da mesma sequência de vídeo codificados com sub-particionamento pela decisão RDO. A maioria dos macroblocos apresenta um valor \mathbf{H} superior a 10.000. O padrão se repete para todos

os 8 vídeos utilizados para simulações, o que permite concluir que o valor limite em torno de 10.000 é um bom indicador para detecção de sub-particionamento.

Em suma, na detecção de sub-particionamento proposta, o valor da heterogeneidade do macrobloco (H) é calculado conforme explicado na seção 4.2.2.2 e comparado com o valor 10.000. Caso o valor H seja menor que 10.000, o macrobloco é definido como homogêneo e deve ser codificado sem sub-particionamento de macrobloco. Caso contrário, o macrobloco não é homogêneo e deve ser codificado com a utilização de sub-particionamento. Nota-se, neste ponto, que a decisão hierárquica apresentada na Figura 4.4 apresenta uma limitação: a não possibilidade de utilização de uma sub-partição $P8x8$ dentro de apenas uma partição $P16x8$ ou $P8x16$. Caso a utilização de um sub-particionamento $P8x8$ seja decidida, este será utilizado em ambas metades do macrobloco. Esta limitação acaba por gerar, em alguns casos, um pequeno aumento na taxa de bits, já que o codificador envia a informação de codificação de duas partições $P8x8$ ao invés de uma partição $P16x8$ ou $P8x16$. A qualidade da imagem tende a não sofrer alterações, já que ambas as partições possuem vetores de movimento que apontam para a mesma (ou semelhante) direção e sentido.

A Tabela 4.13 apresenta os resultados obtidos com a detecção de sub-particionamento baseada na análise de heterogeneidade os compara com a aplicação da técnica de RDO. Nos resultados apresentados, apenas a decisão de utilização de sub-particionamento foi realizada com base na heterogeneidade, ou seja, as decisões de melhor particionamento ainda são realizadas através do processo de RDO. A aplicação do valor limite igual a 10.000 resultou em um aumento de 0,62% na quantidade de bits utilizados para codificar o vídeo. A perda de qualidade da imagem foi desprezível, ficando próxima a zero.

Tabela 4.13: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando detecção de sub-particionamento com base na heterogeneidade do macrobloco original.

Vídeo	Decisão RDO		Detecção de Sub-Partic. por Heterog.		RDO x Detecção de Sub-Partic.	
	Taxa de bits	PSNR (dB)	Taxa de bits	PSNR (dB)	+ Taxa de bits (%)	+ PSNR (dB)
<i>MAN IN CAR</i>	2.138.352	42,43	2.135.464	42,44	-0,14	0,00
<i>RIVERBED</i>	15.729.560	38,71	15.837.128	38,72	0,68	0,00
<i>ROLLING TOMATOES</i>	3.427.472	40,36	3.425.976	40,36	-0,04	0,00
<i>RUSH HOUR</i>	3.711.680	41,47	3.749.352	41,47	1,01	0,00
<i>STATION 2</i>	4.407.176	39,64	4.428.856	39,64	0,49	0,00
<i>SUNFLOWER</i>	7.105.888	42,58	7.149.072	42,59	0,61	0,01
<i>TRACTOR</i>	11.344.312	38,92	11.409.976	38,92	0,58	0,00
<i>TRAFFIC</i>	8.657.232	39,37	8.733.856	39,37	0,89	0,00
MÉDIA	7.065.209	40,44	7.109.071	40,44	0,62	0,00

4.3.3 Decisão 3: Tamanho de Particionamento com Base na Força de Borda

Após a decisão de utilização ou não de sub-partição de macrobloco, o tamanho do particionamento de macrobloco é decidido com base na diferença entre os pixels que compõem as linhas e as colunas do macrobloco. Conforme explicado na seção anterior, todas as partes que compõem os objetos presentes em um vídeo movem-se, em geral, na mesma direção e sentido, sendo agrupadas em partições de macrobloco maiores. Assim, a detecção de arestas de objetos nas áreas limítrofes das partições permite a definição de qual o melhor tamanho de particionamento para o macrobloco. Neste trabalho,

considera-se que objetos diferentes possuem, em geral, valores de luminância diferentes e, portanto, podem ser distinguidos através de análises simples sobre as amostras de luminância do macrobloco. Entende-se que esta abstração leva a erros de decisão nos casos em que objetos diferentes possuem valores de luminância semelhantes. Contudo, os resultados apresentados no final desta seção mostram que, apesar da existência de uma parcela de erros de decisão, a heurística ainda apresenta resultados satisfatórios.

Neste trabalho propõe-se a utilização da detecção de arestas de uma forma semelhante à detecção de arestas do Filtro Redutor de Efeito de Bloco utilizado no padrão H.264/AVC. O Filtro Redutor de Efeito de Bloco é um filtro adaptativo, isto é, ele permite distinguir uma aresta real da imagem (que não é filtrada) de um artefato gerado por um passo elevado de quantização (que é filtrado). Contudo, a decisão de tamanho de particionamento proposta neste trabalho é mais simples, já que são as arestas reais da imagem que precisam ser detectadas e, além disso, a quantização ainda não foi aplicada e não gerou bordas indesejáveis.

A detecção é aplicada nas bordas verticais e horizontais das possíveis partições de macrobloco, como ilustra a Figura 4.9. As bordas de particionamento vertical e horizontal estão destacadas como linhas mais fortes e os pixels utilizados no cálculo da detecção de borda estão destacados em cinza. Oito amostras de cada linha (ou coluna) são utilizadas no cálculo de detecção de borda. A Figura 4.9 apresenta um exemplo de oito amostras ($\mathbf{p}_3, \mathbf{p}_2, \mathbf{p}_1, \mathbf{p}_0, \mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2$ e \mathbf{q}_3) utilizadas em uma linha e em uma coluna para o cálculo. A equação (56) define o somatório realizado para determinar a força de borda vertical e a equação (57) define o cálculo da força de borda horizontal, onde \mathbf{O} é o conjunto de pixels do macrobloco original, i é a linha do macrobloco e j é a coluna do macrobloco. Como se pode perceber pelas equações, os pixels são comparados de dois em dois e em um sentido de fora para dentro do macrobloco, isto é, \mathbf{p}_3 é comparado com \mathbf{q}_3 , \mathbf{p}_2 é comparado com \mathbf{q}_2 e assim por diante. Isto permite que tanto mudanças abruptas quanto mudanças relativamente graduais no valor das amostras sejam encontradas.

$$BV = \sum_{i=0}^{16} \sum_{j=4}^7 \left(O_{i,j} - O_{i,(15-j)} \right) \quad (56)$$

$$BH = \sum_{j=0}^{16} \sum_{i=4}^7 \left(O_{i,j} - O_{(15-i),j} \right) \quad (57)$$

A partir dos valores de borda horizontal (\mathbf{BH}) e borda vertical (\mathbf{BV}) obtidos nas equações (56) e (57), são realizadas comparações com valores limite para determinação do tamanho de partição de macrobloco. Testes realizados com o software de referência e com a aplicação da heurística variaram o valor limite para a diferença entre \mathbf{BH} e \mathbf{BV} em uma faixa de 0 a 150. O valor 80 foi aquele que retornou os melhores resultados de PSNR e taxa de bits para a maioria dos vídeos, em comparação com a decisão baseada em RDO. Desta forma, a decisão proposta de particionamento para cada macrobloco segue o algoritmo apresentado abaixo. Caso o primeiro teste seja verdadeiro, a diferença entre \mathbf{BH} e \mathbf{BV} é pequena e, portanto, não há diferença significativa entre os pixels que compõem as metades do macrobloco. Neste caso, o tamanho de partição é 16x16. Caso o segundo teste seja verdadeiro, o valor de \mathbf{BH} é suficientemente maior que o valor de \mathbf{BV} e, portanto, a borda de particionamento horizontal é mais forte. Assim, o tamanho de partição deve ser 16x8. No último caso, o contrário é válido e, portanto, o tamanho de partição deve ser 8x16.

Se $|BH - BV| \leq 80$ Então

$melhor_modo_INTER = P16x16$

Senão Se $BH - BV > 80$ Então

$melhor_modo_INTER = P16x8$

-- (partição 16x16 é formada por dois blocos 16x8)

Senão Se $BV - BH > 80$ Então

$melhor_modo_INTER = P8x16$

-- (partição 16x16 é formada por dois blocos 8x16)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0					p ₃	p ₂	p ₁	p ₀	q ₀	q ₁	q ₂	q ₃				
1																
2																
3																
4	p ₃															
5	p ₂															
6	p ₁															
7	p ₀															
8	q ₀															
9	q ₁															
10	q ₂															
11	q ₃															
12																
13																
14																
15																

Figura 4.9: Pixels utilizados (em cinza) no cálculo de detecção de bordas vertical e horizontal para definição do particionamento de macrobloco.

A Tabela 4.14 apresenta os resultados obtidos para decisão de tamanho de partição com base na análise de bordas internas do macrobloco. Como nas seções anteriores, apenas a decisão de tamanho de partição foi realizada com base na detecção de borda, isto é, todas as outras decisões ainda são tomadas através da aplicação de RDO. A aplicação da técnica resultou em um aumento de 3,59% na quantidade de bits utilizados para codificar o vídeo e em um aumento de 0,02 dB na qualidade da imagem. Em alguns casos, a decisão proposta apresentou uma leve queda na qualidade da imagem, enquanto que em outros casos a qualidade sofreu melhoria. Conforme já explicado, isto acontece pelo fato de a decisão proposta não ser tão precisa quanto a decisão RDO e, portanto, selecionar uma maior ou uma menor quantidade de blocos menores.

Tabela 4.14: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão de tamanho de partição com base na detecção de borda.

Vídeo	Decisão RDO		Decisão de Partição p/ Borda		RDO x Borda	
	Taxa de bits	PSNR (dB)	Taxa de bits	PSNR (dB)	+ Taxa de bits (%)	+ PSNR (dB)
<i>MAN IN CAR</i>	2.138.352	42,43	2.138.720	42,40	0,02	0,04
<i>RIVERBED</i>	15.729.560	38,71	15.737.280	38,67	0,05	0,04
<i>ROLLING TOMATOES</i>	3.427.472	40,36	3.553.224	40,33	3,67	0,03
<i>RUSH HOUR</i>	3.711.680	41,47	3.899.904	41,44	5,07	0,03
<i>STATION 2</i>	4.407.176	39,64	4.545.296	39,61	3,13	0,03
<i>SUNFLOWER</i>	7.105.888	42,58	7.532.872	42,59	6,01	-0,01
<i>TRACTOR</i>	11.344.312	38,92	12.220.032	38,96	7,72	-0,03
<i>TRAFFIC</i>	8.657.232	39,37	8.923.240	39,36	3,07	0,01
MÉDIA	6.933.349	40,49	7.187.744	40,48	3,59	0,02

4.3.4 Decisão 4: Tamanho de Sub-Particionamento com Base na Força de Borda

Da mesma forma que a decisão de tamanho de partição, a decisão de tamanho de sub-partição é realizada com base na diferença entre os pixels que compõem as linhas e as colunas dos sub-macroblocos. Testes semelhantes aos descritos na seção 4.3.3 foram realizados para obtenção dos valores de borda vertical e borda horizontal de cada sub-macrobloco que compõe o macrobloco.

A Figura 4.10 apresenta a estrutura do macrobloco dividida em quatro estruturas menores, os sub-macroblocos de tamanho 8x8. Em cada um dos sub-macroblocos, os cálculos de borda vertical e horizontal são realizados para posterior comparação e decisão de tamanho de sub-partição. Contudo, de forma diferente do cálculo de borda para macroblocos apresentado anteriormente, o cálculo de borda para sub-macrobloco opera sobre 4 amostras por vez, ao invés de 6. Além disso, para que também seja possível a identificação de blocos de tamanho 4x4, as bordas são calculadas em duas partes: a parte superior (ou esquerda) e a parte inferior (ou direita).

As equações (62) e (63) apresentam os cálculos para bordas verticais e horizontais do primeiro sub-macrobloco de cada macrobloco. Para diferenciá-las das equações apresentadas para borda de macrobloco, são utilizadas as variáveis **SBV** (sub-borda horizontal) e **SBH** (sub-borda vertical) para representar as bordas verticais e horizontais de sub-macrobloco, respectivamente. As duas partes de cada borda são representadas como **SBV1** (sub-borda vertical, parte 1), **SBV2** (sub-borda vertical, parte 2), **SBH1** (sub-borda horizontal, parte 1) e **SBH2** (sub-borda horizontal, parte 2), apresentadas (para o primeiro sub-macrobloco) nas equações (58), (59), (60) e (61), respectivamente.

$$SBV1 = \sum_{i=0}^4 \sum_{j=2}^3 (O_{i,j} - O_{i,(7-j)}) \quad (58)$$

$$SBV2 = \sum_{i=4}^7 \sum_{j=2}^3 (O_{i,j} - O_{i,(7-j)}) \quad (59)$$

$$SBH1 = \sum_{j=0}^4 \sum_{i=2}^3 (O_{i,j} - O_{(7-i),j}) \quad (60)$$

$$SBH2 = \sum_{j=4}^7 \sum_{i=2}^3 (O_{i,j} - O_{(7-i),j}) \quad (61)$$

$$SBV = SBV1 + SBV2 \quad (62)$$

$$SBH = SBH1 + SBH2 \quad (63)$$

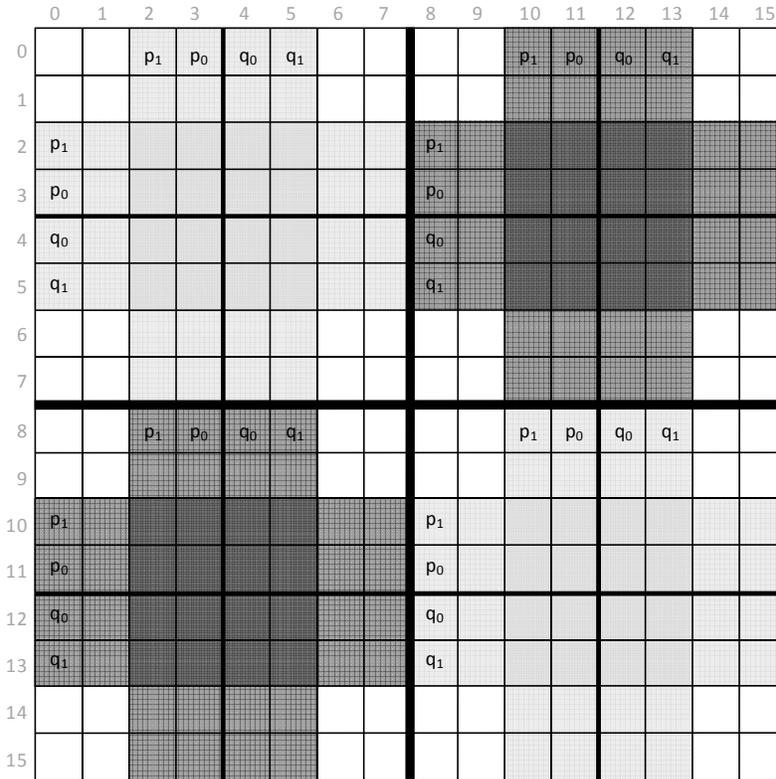


Figura 4.10: Pixels utilizados (em tons de cinza) no cálculo de detecção de bordas vertical e horizontal para definição do sub-particionamento de macrobloco.

A partir dos valores de sub-borda horizontal (**SBH**) e sub-borda vertical (**SBV**) obtidos nas equações (56) e (57), são realizadas comparações com valores limite para determinação do tamanho da sub-partição. Após determinado o tamanho da sub-partição, novos testes foram realizados com o valor das metades de sub-borda para determinar a existência ou a inexistência de partições de tamanho 4x4. A decisão para cada sub-macrobloco segue o algoritmo apresentado abaixo. Caso o primeiro teste seja verdadeiro, isto é, quando a diferença entre **SBH** e **SBV** for pequena, não há diferença significativa entre os pixels que compõem as duas metades do sub-macrobloco. Neste caso, o tamanho de partição é 8x8. Caso contrário, se o segundo teste for verdadeiro, o valor de **SBH** é suficientemente maior que o valor de **SBV** e, portanto, a borda de sub-particionamento horizontal é mais forte. Assim, o tamanho de partição deve ser 8x4 ou menor. Um terceiro teste é realizado para definir a existência de uma partição menor (4x4). Caso a primeira ou a segunda metade da sub-borda vertical seja suficientemente grande, a sub-partição é de tamanho 4x4. Caso contrário, o tamanho é 8x4. O último caso funciona de forma semelhante e define sub-partições de tamanho 4x8 ou 4x4, realizando os testes sobre as metades de sub-borda horizontal.

Se $|SBH - SBV| \leq 40$ Então

$melhor_modo_INTER = P8x8$

Senão Se $SBH - SBV > 40$ Então

Se $SBV1 > 20$ OU $SBV2 > 20$ Então

$melhor_modo_INTER = P4x4$

-- (sub-partição 8x8 é formada por quatro blocos 4x4)

Senão

$melhor_modo_INTER = P8x4$

-- (sub-partição 8x8 é formada por dois blocos 8x4)

Senão Se $SBV - SBH > 40$ Então

Se $SBH1 > 20$ OU $SBH2 > 20$ Então

$melhor_modo_INTER = P4x4$

-- (sub-partição 8x8 é formada por quatro blocos 4x4)

Senão

$melhor_modo_INTER = P4x8$

-- (sub-partição 8x8 é formada por dois blocos 4x8)

A Tabela 4.15 apresenta os resultados obtidos para decisão de tamanho de sub-partição com base na análise de bordas internas do sub-macrobloco. Somente a decisão de tamanho de sub-partição foi realizada com base na detecção de borda, isto é, todas as outras decisões ainda são tomadas através da aplicação de RDO. A aplicação da técnica resultou em um aumento de 0,34% na quantidade de bits utilizados para codificar o vídeo e em um aumento de 0,02 dB na qualidade da imagem. Conforme apresentado na Tabela 4.11, sub-partições de macrobloco não representam a parcela mais significativa dos modos escolhidos em quadros do tipo P. Assim, a aplicação de uma decisão aproximada e menos precisa que a técnica RDO não levou a uma diferença significativa em termos de taxa de bits e qualidade da imagem gerada.

Tabela 4.15: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando decisão de tamanho de sub-partição com base na detecção de borda.

Vídeo	Decisão RDO		Decisão de Sub-Partição por Borda		RDO x Borda	
	Taxa de bits	PSNR (dB)	Taxa de bits	PSNR (dB)	+ Taxa de bits (%)	+ PSNR (dB)
<i>MAN IN CAR</i>	2.138.352	42,43	2.096.136	42,44	-1,97	0,00
<i>RIVERBED</i>	15.729.560	38,71	15.747.760	38,63	0,12	0,08
<i>ROLLING TOMATOES</i>	3.427.472	40,36	3.460.384	40,36	0,96	0,00
<i>RUSH HOUR</i>	3.711.680	41,47	3.729.568	41,45	0,48	0,03
<i>STATION 2</i>	4.407.176	39,64	4.423.496	39,65	0,37	-0,01
<i>SUNFLOWER</i>	7.105.888	42,58	7.152.712	42,56	0,66	0,02
<i>TRACTOR</i>	11.344.312	38,92	11.435.984	38,89	0,81	0,03
<i>TRAFFIC</i>	8.657.232	39,37	8.666.616	39,36	0,11	0,01
MÉDIA	6.933.349	40,44	7.089.082	40,42	0,34	0,02

4.4 Decisões Rápidas Integradas e Análise de Complexidade

Os testes finais foram realizados com todas as heurísticas integradas no software de referência do padrão H.264/AVC. Para isto, a opção de decisão através da técnica RDO

continuou habilitada no software, mas foi alterada de forma que todas as decisões fossem realizadas de acordo com as heurísticas propostas neste trabalho.

O software de referência realiza a decisão RDO através de um laço de repetição sobre todos os possíveis modos de codificação. O custo taxa-distorção mínimo é salvo em uma variável que é atualizada a cada passo do laço. No final do laço, o modo com menor custo taxa-distorção é escolhido. Para implementar a decisão rápida, as decisões anteriores foram inseridas no software e testes para verificar cada uma das condições foram inseridos no início do laço. Por exemplo, no caso de decisão de modo *SKIP*, o valor de SAD entre o macrobloco original e o macrobloco de referência é testado no início do laço para verificar a estacionaridade do primeiro. Caso a estacionaridade seja verificada, os custos de todos os outros modos diferentes de *SKIP* são fixados com o máximo valor possível e o laço é interrompido. Isto força que a decisão ocorra de acordo com o método proposto.

A ordem de testes acontece de acordo com as frequências apresentadas na Tabela 4.11. Os modos mais frequentes são testados primeiro, no início do laço e são seguidos pelos modos que aparecem com menor frequência. Isto garante realização de uma escolha estatisticamente mais correta e mais rápida.

A Tabela 4.16 apresenta os resultados obtidos para decisão de modo de codificação de acordo com todos os métodos apresentados nas seções anteriores. Como em todas as outras tabelas de resultados, as primeiras colunas apresentam os resultados obtidos através da técnica de RDO completa, as colunas centrais apresentam os resultados obtidos através do método proposto e as colunas finais mostram uma comparação entre as duas abordagens em termos de taxa de bits (percentuais) e qualidade da imagem gerada (em PSNR). A aplicação de todos os métodos resultou em um aumento médio de 7,53% na quantidade de bits utilizados para codificar o vídeo e em um aumento de 0,05 dB na qualidade da imagem (em PSNR).

Tabela 4.16: Taxa de bits e qualidade (PSNR) para vídeos codificados utilizando todas as técnicas de decisão rápida propostas neste trabalho.

Vídeo	Decisão RDO		Decisão Rápida Proposta		RDO x Decisão Rápida	
	Taxa de bits	PSNR (dB)	Taxa de bits	PSNR (dB)	+ Taxa de bits (%)	+ PSNR (dB)
<i>MAN IN CAR</i>	2.138.352	42,43	2.171.544	42,40	1,55	0,04
<i>RIVERBED</i>	15.729.560	38,71	15.981.912	38,55	1,60	0,16
<i>ROLLING TOMATOES</i>	3.427.472	40,36	3.740.704	40,32	9,14	0,04
<i>RUSH HOUR</i>	3.711.680	41,47	4.066.768	41,40	9,57	0,08
<i>STATION 2</i>	4.407.176	39,64	4.630.136	39,60	5,06	0,04
<i>SUNFLOWER</i>	7.105.888	42,58	7.636.112	42,59	7,46	-0,01
<i>TRACTOR</i>	11.344.312	38,92	11.897.392	38,87	4,88	0,06
<i>TRAFFIC</i>	8.657.232	39,37	10.650.976	39,33	23,03	0,03
MÉDIA	6.933.349	40,44	7.596.943	40,38	7,53	0,05

Como se pode perceber, os resultados de aumento de taxa de bits anteriores foram acumulados e geraram um aumento médio de 7,53%. A qualidade da imagem, por outro lado, manteve-se bastante parecida à obtida através da técnica de RDO. Isto acontece porque o software de referência procura melhorar a qualidade da imagem gerada por uma codificação menos eficiente através do envio um número maior de resíduos. Assim, a eficiência menor acaba por refletir apenas em uma taxa de bits mais alta,

conservando a qualidade da imagem. É importante notar que em um dos casos (vídeo *TRAFFIC*), o aumento da taxa de bits apresentou uma característica discrepante em relação aos demais, que ficaram entre 1% e 10%. Em trabalhos futuros, as características deste vídeo serão analisadas com a finalidade de identificar formas de adaptabilidade dos valores limites utilizados nas heurísticas, visando à redução do impacto da sua utilização.

O aumento da taxa de bits é ainda bastante pequeno frente à enorme redução da complexidade computacional envolvida no processo de decisão. Conforme apresentado na Figura 3.10, no capítulo 3, a decisão de modo em uma codificação baseada na técnica de RDO é realizada apenas após a codificação completa para todos os modos possíveis. Conforme ilustra a Figura 4.11, a decisão proposta neste trabalho pode ser realizada após a etapa de geração de resíduos, seguida apenas por uma nova etapa composta pelo cálculo de DCT, no caso da predição intra-quadro, ou pela detecção de estacionaridade, cálculo de DCT e detecção de bordas, no caso da predição inter-quadros. O laço de repetição apresentado na Figura 3.10 é eliminado, simplificando enormemente o processo de decisão de modo de codificação.

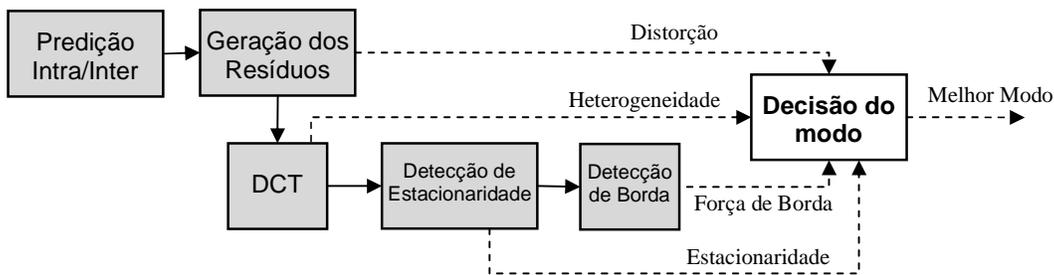


Figura 4.11: Processo de codificação e decisão de modo proposta.

Considerando a decisão de modos para codificação das componentes de luminância, os próximos parágrafos realizam uma comparação entre a complexidade da decisão RDO e da decisão baseada no conjunto de heurísticas proposto.

Conforme apresentado no capítulo 2 desta dissertação, a codificação intra-quadro pode ser realizada de acordo com 4 modos de codificação intra-quadro 16x16 e 9 modos de codificação intra-quadro 4x4. Levando em consideração o fluxo de codificação baseado em RDO da Figura 3.10, a decisão de melhor modo para um macrobloco pertencente a um quadro I acontece, portanto, após a avaliação de todos os modos 16x16 para o macrobloco e de todos os 9 modos 4x4 para cada um dos 16 blocos que compõem o macrobloco. Em termos de complexidade computacional, isto significa que o laço apresentado na Figura 3.10 é repetido 4 vezes para a predição intra-quadro 16x16 e 144 vezes para a predição intra-quadro 4x4 (9 modos vezes 16 blocos 4x4 em um macrobloco), totalizando 148 iterações para a decisão ser realizada.

A codificação inter-quadros é composta, conforme apresentado no capítulo 2, por 8 modos de codificação (*SKIP*, *P16x16*, *P16x8*, *P8x16*, *P8x8*, *P8x4*, *P4x8* e *P4x4*). Também considerando o fluxo da Figura 3.10, a decisão para um macrobloco inter-quadros em um quadro P é realizada após a avaliação dos 4 primeiros modos (*SKIP*, *P16x16*, *P16x8* e *P8x16*) para cada macrobloco e dos 4 últimos modos para cada uma das 4 sub-partições 8x8 que compõem o macrobloco. Em outras palavras, são realizadas 4 iterações no laço da Figura 3.10 para os modos de macrobloco e 16 iterações para os modos de sub-partição de macrobloco, totalizando 20 repetições. Além disso, como o padrão H.264/AVC define que quadros P também podem conter macroblocos intra-

quadro, todas as 148 iterações para modos intra-quadro também são realizadas, totalizando 168 repetições.

Considerando o conjunto de heurísticas proposto e o fluxo de execução da Figura 4.11, a decisão de melhor modo para um macrobloco em quadro I acontece após o cálculo de resíduos para os modos e cálculo da DCT. Isto significa que, caso a heurística que detecta heterogeneidade decida que o melhor tamanho de bloco é 16x16, o laço de codificação é percorrido apenas uma vez com o modo 16x16 escolhido para o macrobloco. Caso o tamanho de bloco escolhido seja 4x4, o laço é percorrido 16 vezes, uma vez para cada bloco 4x4.

Em macroblocos inter-quadros em quadros P, a decisão de melhor tamanho de partição e de sub-partição acontece após a avaliação da heterogeneidade, forças de bordas e estacionaridade. O laço de codificação completo é percorrido, portanto, apenas uma vez para o macrobloco inter-quadros considerando o tamanho de partição e os tamanhos de sub-partição decididos.

Levando em consideração um grupo de imagens (GOP – *Group of Pictures*) composto por um quadro I e cinco quadros P (sequência IPPPPP) em um vídeo de resolução HD 1080p (8.160 macroblocos por quadro), a codificação com decisão baseada na técnica de RDO realiza 8.062.080 iterações no laço apresentado na Figura 3.10 (148 repetições para cada macrobloco no quadro I e 168 repetições para cada macrobloco nos quadros P). Por outro lado, quando considerado o mesmo GOP, um vídeo de mesma resolução e a codificação baseada no conjunto de heurísticas proposto, a codificação do GOP completo ocorre em apenas 171.360 iterações (16 repetições para cada macrobloco no quadro I, no pior caso, e 1 iteração para cada macrobloco nos quadros P). Assim é possível concluir que o conjunto de heurísticas proposto leva a uma redução da ordem de 47 vezes na complexidade de codificação. Esse ganho é muito elevado e justifica as pequenas perdas de PSNR e de taxa de bits causadas pelo uso das heurísticas propostas. A Tabela 4.17 resume todos os números apresentados nos últimos parágrafos, comparando as duas técnicas de decisão em níveis diferentes de unidades de codificação da imagem (blocos, partições, sub-partições, macroblocos, quadros e GOPs).

Tabela 4.17: Número de cálculos para a decisão de modo em um GOP IPPPPP.

Iterações	RDO	Decisão Proposta
Bloco Intra-Quadro 4x4	144	16
Bloco Intra-Quadro 16x16	4	1
Macrobloco I (pior caso)	148	16
Partições em MB Inter-Quadros	4	1
Sub-partições em MB Inter-Quadros	16	1
Macrobloco P	168	1
Quadro I	1.207.680	130.560
Quadro P	1.370.880	8.160
GOP IPPPPP	8.062.080	171.360

4.5 Trabalhos Relacionados

Nesta seção, o conjunto de heurísticas proposto é comparado com outros trabalhos encontrados na literatura técnica que também visam à redução da complexidade do processo de decisão de modo de codificação. Inicialmente, um resumo de cada trabalho é apresentado.

4.5.1 Trabalho de Lee et al. (2004)

Neste trabalho, Lee et al. (2004) realiza uma decisão rápida de modo de codificação com base em informações estatísticas da ocorrência de modos em quadros do tipo P. De acordo com Lee et al. (2004), em média, 3% dos modos de codificação escolhidos em quadros P são do tipo intra-quadro. Desta forma, a análise de modos inter-quadros antes dos modos intra-quadro associada à aplicação de heurísticas para a decisão reduz o tempo de codificação sem que haja uma queda significativa na qualidade da imagem ou um grande aumento na taxa de bits.

Em primeiro lugar, é analisada a aplicação do modo *SKIP* através de 4 condições. Caso todas as condições sejam verdadeiras, o modo *SKIP* é selecionado e os demais modos não são avaliados. Caso contrário, os demais modos inter-quadros são avaliados antes dos modos intra-quadro. A avaliação dos modos intra-quadro só acontece se uma condição é satisfeita: o grau de correlação espacial entre os blocos do quadro atual é maior que o grau de correlação temporal entre os blocos do quadro atual e do seu vizinho. Para medir as duas correlações, resultados intermediários já disponíveis do processo de codificação são utilizados. A correlação espacial é calculada a partir do valor das bordas entre o bloco atual e os seus adjacentes e a correlação temporal é calculada a partir da quantidade de bits utilizados para codificar o resíduo obtido com o melhor modo inter-quadros.

Quando apenas a detecção prévia de modo *SKIP* foi utilizada, houve uma redução de 31,8% do número de cálculos do custo taxa-distorção. A qualidade da imagem não sofreu degradação significativa. Por outro lado, quando a detecção prévia de modo *SKIP* foi utilizada junto à decisão de modo intra-quadro seletiva, notou-se uma redução de 71,7% na quantidade de cálculos do custo taxa-distorção. Nos termos de análise de complexidade utilizados nesta dissertação, isto significa que houve uma redução média de 3,5 vezes no número de iterações do laço de codificação apresentado na Figura 3.10. A redução média na qualidade da imagem foi de 0,034 dB, o que é pouco significativo. O trabalho não apresenta qualquer análise acerca dos efeitos sobre a taxa de bits.

4.5.2 Trabalho de Sun et al. (2008)

O trabalho de Sun et al. (2008) propõe uma decisão rápida voltada apenas para a codificação intra-quadro. Os autores defendem que nem todos os modos precisam ter o seu custo taxa-distorção calculados, principalmente aqueles com valores de SATD muito altos. Os modos que apresentam um valor alto para SATD são descartados da avaliação de melhor modo, pois têm tendência de apresentar uma alta taxa de bits.

O algoritmo proposto é dividido em três etapas. A primeira etapa calcula o SATD para todos os modos intra-quadro e seleciona os modos com menor distorção. A segunda etapa calcula a variância dos coeficientes transformados e elimina os modos cujos coeficientes são muito diferentes. A última etapa ordena os modos de acordo com o valor de SATD (primeiro os modos com menor distorção, depois os modos com maior distorção) e realiza o processo RDO considerando este conjunto de modos ordenados.

Quando apenas a eliminação de modos a partir do valor de SATD é utilizada, o número de cálculos taxa-distorção diminuiu, em média, 50%. Quando ambas as técnicas foram combinadas (eliminação de modos por SATD e eliminação de modos por variância dos coeficientes transformados), o número de cálculos foi reduzido em 79%. Nos termos de complexidade utilizados nesta dissertação, isto significa que a proposta de Sun et al. (2008) levaria a uma redução média de 4,7 vezes no número de iterações do laço de codificação apresentado na Figura 3.10. Contudo, como a segunda etapa da

decisão é baseada na análise dos coeficientes transformados, as transformadas também devem ser executadas algumas vezes, uma vez para cada modo selecionado pela primeira etapa do algoritmo. A redução média na qualidade da imagem ficou em torno de 0,136 dB e o aumento na taxa de bits foi de apenas 0,69%.

4.5.3 Trabalho de Fengqin et al. (2008)

A análise dos valores dos coeficientes transformados de cada bloco é realizada neste trabalho para determinar os modos intra-quadro candidatos a serem avaliados através da decisão baseada em RDO. De acordo com Fengqin et al. (2008), o bloco de coeficientes transformados tende a assumir uma formação de valores característica de acordo com cada modo da predição intra-quadro utilizado.

A partir das formações identificadas, a probabilidade de cada modo pode ser calculada através da média ponderada dos coeficientes transformados. Os autores propõem um conjunto de equações para calcular a probabilidade de cada modo. Quanto menor a média de coeficientes, maior é a probabilidade de o modo ser utilizado. Os modos com menor probabilidade são descartados. Um método de prevenção de difusão de erro de decisão é aplicado através da inserção periódica de quadros com decisão RDO completa.

A aplicação do algoritmo proposto por Fengqin et al. (2008) gerou uma redução menor que 0,02 dB na qualidade da imagem e um aumento médio de 3% na taxa de bits dos vídeos codificados. Em comparação à decisão baseada em RDO, a técnica proposta realiza 5,92 vezes menos cálculos de custo taxa-distorção. É importante notar que a técnica proposta realiza operações sobre os coeficientes transformados, de forma que a etapa de transformadas da Figura 3.10 ainda é executada repetidamente para a avaliação do melhor modo.

4.5.4 Trabalho de Kim et al. (2004b)

Neste trabalho, Kim et al. (2004b) também utiliza o cálculo de SATD utilizado por Sun et al. (2008) para eliminar os modos intra-quadro menos prováveis. O trabalho também propõe a utilização de uma estimativa de taxa de bits baseada na quantidade de coeficientes quantizados para zero após a etapa de quantização.

Neste caso, a técnica proposta realiza operações sobre os coeficientes transformados e quantizados, de forma que a etapa de transformadas e de quantização da Figura 3.10 é executada para cada um dos modos selecionados pela avaliação de SATD.

A aplicação da técnica proposta por Kim et al. (2004b) gerou uma redução de apenas 0,07 dB na qualidade da imagem. O trabalho não apresenta qualquer avaliação relacionada à taxa de bits gerada nem à quantidade de cálculos de custo taxa-distorção. Contudo, a redução em tempo de execução foi de aproximadamente 85%.

4.5.5 Trabalho de Lee et al. (2009)

O trabalho propõe a diminuição da complexidade da decisão de modo intra-quadro através da análise dos componentes de baixa frequência do bloco de coeficientes transformados. A proposta se baseia na análise da homogeneidade para a determinação do melhor tamanho de bloco através da aplicação da transformada DCT 16x16 sobre o macrobloco original, conforme utilizado nesta dissertação. Contudo, o trabalho de Lee et al. (2009) realiza apenas a decisão de melhor tamanho de bloco, de forma que o melhor modo intra-quadro 4x4 ou o melhor modo intra-quadro 16x16 ainda é escolhido através da decisão baseada em RDO. Para limitar a quantidade de modos analisados, o

trabalho propõe a utilização do valor de SATD para eliminar os modos intra-quadro menos prováveis.

O trabalho de Lee et al. (2009) não apresenta resultados em termos de número de cálculos do custo taxa-distorção. Contudo, a análise do algoritmo permite deduzir que a redução do número de cálculos ficou, em média, em torno de 25%, o que significa que o número de repetições do laço apresentado na Figura 3.10 foi diminuído em 4 vezes. A qualidade da imagem sofreu uma redução de apenas 0,088 dB, enquanto que a taxa de bits apresentou um aumento de 0,09% em relação à decisão baseada em RDO.

4.5.6 Trabalho de Jeon et al. (2009)

O trabalho de Jeon et al. (2009) se baseia no fato de que, em geral, os pixels que compõem uma mesma borda local possuem valores semelhantes. O algoritmo proposto realiza, portanto, a decisão de melhor modo intra-quadro de acordo com os gradientes direcionais do bloco de acordo com cada modo. Uma série de equações que calcula os gradientes é apresentada no trabalho. Os resultados das equações são comparados, para cada bloco, com valores limite pré-calculados através de simulações. Os dois modos com menor gradiente são selecionados como melhores e utilizados, junto com o modo DC, na escolha RDO.

De acordo com Jeon et al. (2009), o número de cálculos de custo taxa-distorção foi diminuído 12 vezes no melhor caso e 6 vezes no pior caso. A diminuição na qualidade da imagem foi de 0,148 dB e o aumento da taxa de bits ficou em torno de 4%.

4.5.7 Trabalho de Wu et al. (2005)

O trabalho de Wu et al. (2005) apresenta um conjunto de heurísticas para decisão de modo completa, isto é, intra-quadro e inter-quadros. Os autores observam que regiões homogêneas e estáticas prevalecem em sequências naturais de vídeo e são codificadas como *P16x16* ou *SKIP*. A detecção de bordas na imagem através da utilização do Operador de Sobel é utilizada para determinar a homogeneidade dos blocos e a detecção de estacionaridade através do cálculo de SAD é utilizada para determinar o modo *SKIP*, conforme utilizado nesta dissertação.

O algoritmo desenvolvido diminui a complexidade da codificação através da finalização prévia do processo de RDO quando alguma das condições especificadas é alcançada. Caso nenhuma das condições seja alcançada (isto é, caso não seja detectada estacionaridade ou homogeneidade no macrobloco), todos os modos ainda precisam ser checados. Apesar de não apresentar resultados em termos de número de cálculos de custos, a análise do algoritmo proposto permite notar que no, pior caso, a decisão proposta não oferece nenhuma redução no número de cálculos de custo taxa-distorção. No melhor caso, a redução no cálculo de custos para modos inter-quadros fica em torno de 6 vezes. A taxa de bits sofreu um aumento de 0,6% e a perda em PSNR da qualidade da imagem foi de 0,03 dB.

4.5.8 Trabalho de Shen et al. (2008)

Shen et al. (2008) realiza a decisão com base na continuidade espacial do campo de movimento. Uma série de simulações realizadas neste trabalho levou à conclusão de que regiões com movimento contínuo têm uma maior probabilidade de serem codificadas com partições 16x16, 16x8 e 8x16, regiões estáticas tendem a ser codificadas com partição 16x16 (modo *SKIP*) e regiões com pouco movimento contínuo são, em geral, codificadas com tamanho 8x8. Regiões sem movimento contínuo tendem a utilizar tamanhos de partição menor que 8x8. Assim como Wu et al. (2005), este trabalho

também utiliza o Operador de Sobel para identificar as bordas dos objetos que compõem a imagem e, assim, prever a sua continuidade espacial.

A aplicação da heurística proposta resultou em uma queda de apenas 0,07 dB na qualidade da imagem codificada e em uma redução de 0,27% na taxa de bits. No melhor caso, a decisão inter-quadros é realizada em apenas 2 iterações do laço apresentado na Figura 3.10. No pior caso, o laço de codificação é percorrido para todos os modos. Apesar de a redução de complexidade aplicar-se apenas à decisão inter-quadros, o tempo de execução da codificação foi reduzido em 54%, já que a grande maioria dos modos utilizados em quadros P é do tipo inter-quadros.

4.5.9 Comparação com Trabalhos Relacionados

Nesta seção, os algoritmos para decisão rápida de modo de codificação apresentados nas seções anteriores são comparados ao conjunto de heurísticas proposto nesta dissertação.

A Tabela 4.17 apresenta os resultados dos trabalhos relacionados em termos de aumento da taxa de bits, redução da qualidade da imagem (PSNR) e redução da quantidade de cálculos de custo taxa-distorção, em comparação aos resultados obtidos através da técnica RDO. A última coluna da Tabela 4.18 especifica o escopo da decisão proposta em cada um dos trabalhos, isto é, se a decisão é realizada para modos intra-quadro, para modos inter-quadros ou para ambos os tipos de modo.

Tabela 4.18: Comparação entre trabalhos relacionados e o conjunto de heurísticas proposto.

Algoritmo para Decisão Rápida	Diminuição da Qualidade (PSNR, em dB)	Aumento na Taxa de Bits (%)	Redução no Número de Cálculos (vezes)	Tipo de Predição
Lee et al. (2004)	0,034	-	3,5	Intra e Inter
Sun et al. (2008)	0,136	0,69	4,7	Intra
Fengqin et al. (2008)	0,020	3,00	5,9	Intra
Lee et al. (2009)	0,088	0,09	4,0	Intra
Jeon et al. (2009)	0,148	4,00	6,0	Intra
Wu et al. (2005)	-0,030	0,60	6,0	Intra e Inter
Shen et al. (2008)	0,070	0,27	0,0	Inter
ESTE TRABALHO	-0,050	7,53	47,0	Intra e Inter

Os números apresentados para quantidade de cálculos referem-se ao pior caso e, como se pode perceber na Tabela 4.18, o trabalho proposto nesta dissertação apresentou a maior redução. Enquanto os outros trabalhos apresentaram reduções no número de iterações de 3,5 a 6 vezes (em relação à aplicação de RDO), este trabalho apresentou uma redução de 47 vezes.

O custo deste ganho reside, contudo, no aumento da taxa de bits em torno de 7,5% em relação à utilização de RDO. Apesar de ser o mais alto dentre todos os aumentos notados, este valor não representa uma grande perda, frente ao enorme ganho em redução de complexidade computacional. Outras técnicas, como a proposta por Jeon et al. (2009), apresentaram um aumento na taxa de bits relativamente alto, mas não conseguiram reduzir o número de cálculos realizados em uma escala tão grande quanto a obtida no trabalho apresentado nesta dissertação.

A redução da qualidade da imagem em todas as abordagens ficou abaixo de 0,15 dB e a deste trabalho ficou entre as mais baixas de todas.

5 ARQUITETURAS PARA DECISÃO RÁPIDA DE MODO DE CODIFICAÇÃO

As arquiteturas que realizam as funções das heurísticas apresentadas no capítulo 4 são descritas neste capítulo. Assim como naquele capítulo, as soluções arquiteturais foram divididas em duas partes: decisão intra-quadro e decisão inter-quadros. As arquiteturas foram todas descritas em VHDL e sintetizadas para dispositivos FPGA da família Virtex 5 da Xilinx (XILINX, 2010). As arquiteturas que realizam cálculo de distorção também foram avaliadas em termos de dissipação de potência para implementação em *standard cells* da tecnologia TSMC 0,18 μm .

5.1 Arquiteturas para Decisão Intra-Quadro

Esta seção apresenta as arquiteturas referentes às decisões propostas na seção 4.2 do capítulo 4. Foram implementadas, inicialmente, as arquiteturas que realizam os cálculos de distorção para a decisão intra-quadro com bloco de tamanho fixo, conforme proposto na seção 4.2.1. Apesar de a métrica SAD ter sido a escolhida para utilização na decisão rápida de modo de codificação, as outras duas métricas também foram avaliadas para fins de comparação em termos de frequência e de utilização de recursos lógicos.

Em seguida, é apresentada a arquitetura que calcula a heterogeneidade do macrobloco para a decisão de melhor tamanho de bloco intra-quadro, conforme proposto na seção 4.2.2.

5.1.1 Arquiteturas para Cálculo de Distorção

Foram implementadas as arquiteturas de SAD, SATD e SSD para o cálculo de distorção em blocos de tamanho 4x4, conforme mostram as próximas subseções. Todas as arquiteturas possuem 16 amostras de entrada por ciclo (8 amostras do bloco original e 8 amostras do bloco predito). Além disso, uma arquitetura que implementa as três métricas com compartilhamento de recursos é apresentada (CORRÊA, 2010) (DINIZ, 2010).

5.1.1.1 Arquitetura para Cálculo de SAD

A arquitetura SAD é composta por oito subtratores, oito calculadores de módulo e uma árvore de somadores com três níveis. Como o cálculo de SAD é realizado em dois ciclos (um para cada metade do bloco 4x4), um acumulador é necessário na saída do último somador, como mostra a Figura 5.1. No final dos dois ciclos, o valor de SAD é gerado para o bloco 4x4 completo.

Uma segunda versão com um *pipeline* de 6 estágios foi desenvolvida. Neste caso, cada passo do *pipeline* foi composto por apenas um operador aritmético (subtração, cálculo de módulo e soma).

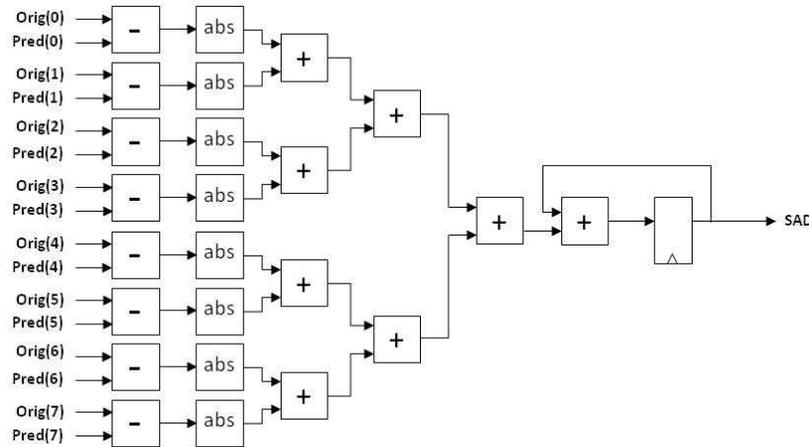


Figura 5.1: Arquitetura SAD com 16 amostras de entrada.

5.1.1.2 Arquitetura para Cálculo de SATD

A arquitetura SATD também é composta por oito subtratores. Contudo, como o cálculo de SATD envolve a realização de uma transformada Hadamard 4x4, os resultados das subtrações de todas as amostras do bloco 4x4 devem estar disponíveis para que os cálculos sejam realizados. Assim, um grupo de 16 registradores foi inserido logo após as subtrações. Depois de dois ciclos, todas as diferenças estão carregadas nos registradores e, portanto, podem ser transformadas. Quatro grupos de 16 somadores/subtratores, correspondentes às colunas da Tabela 4.5, foram adicionados para a realização da transformada Hadamard 4x4. Por razões de simplificação de representação, as conexões que ligam as saídas e as entradas dos grupos não foram mostradas na Figura 5.2. A Hadamard 4x4 também possui uma divisão por dois após o último grupo de somas, conforme mostra a última coluna da Tabela 4.5. Na arquitetura, esta divisão foi realizada através dos 16 deslocamentos de um bit à direita representados na Figura 5.2 através do símbolo \gg . Assim como o cálculo de SAD, o cálculo de SATD termina com o cálculo de valores absolutos e uma árvore de somadores.

Uma versão com *pipeline* de um operador por estágio também foi desenvolvida para a arquitetura de SATD. Neste caso, a arquitetura apresentada na Figura 5.2 foi dividida, portanto, em 10 estágios.

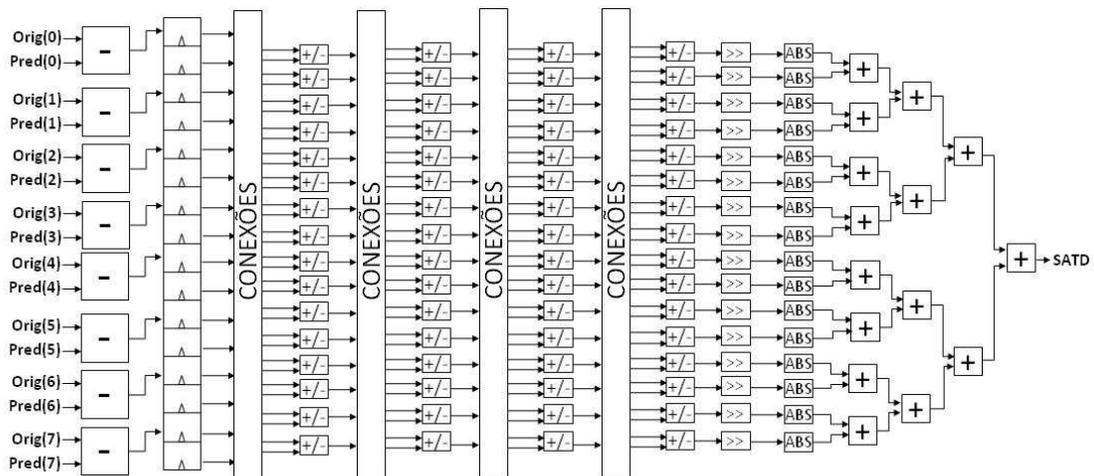


Figura 5.2: Arquitetura SATD com 16 amostras de entrada.

5.1.1.3 Arquitetura para Cálculo de SSD

Na arquitetura SSD, os resultados das subtrações são elevados ao quadrado através de uma multiplicação realizada por uma árvore de somas. A etapa da multiplicação ocorre da seguinte forma: inicialmente, todas as parcelas são geradas através de operações AND bit a bit. Após a geração das oito parcelas (cujo atraso é de apenas uma AND, já que todas operam em paralelo), elas são somadas duas a duas (quatro somas em um primeiro passo, duas somas em um segundo passo e uma soma em um terceiro passo).

Após a operação de multiplicação ser realizada para as 16 diferenças resultantes da subtração entre o bloco original e o bloco predito, os resultados são somados em uma estrutura de árvore de somas, como nas outras métricas. Assim como no cálculo de SAD, o cálculo de SSD acontece em dois ciclos. No primeiro ciclo, é calculado um resultado parcial relativo à primeira metade do bloco 4x4. No segundo ciclo, o resultado parcial é somado com o cálculo de SSD relativo à segunda metade do bloco 4x4. Por este motivo, um acumulador é utilizado após a árvore de somas, conforme mostra a Figura 5.3.

Uma versão com um *pipeline* de 8 estágios (um operador por estágio) também foi desenvolvida para arquitetura de SSD.

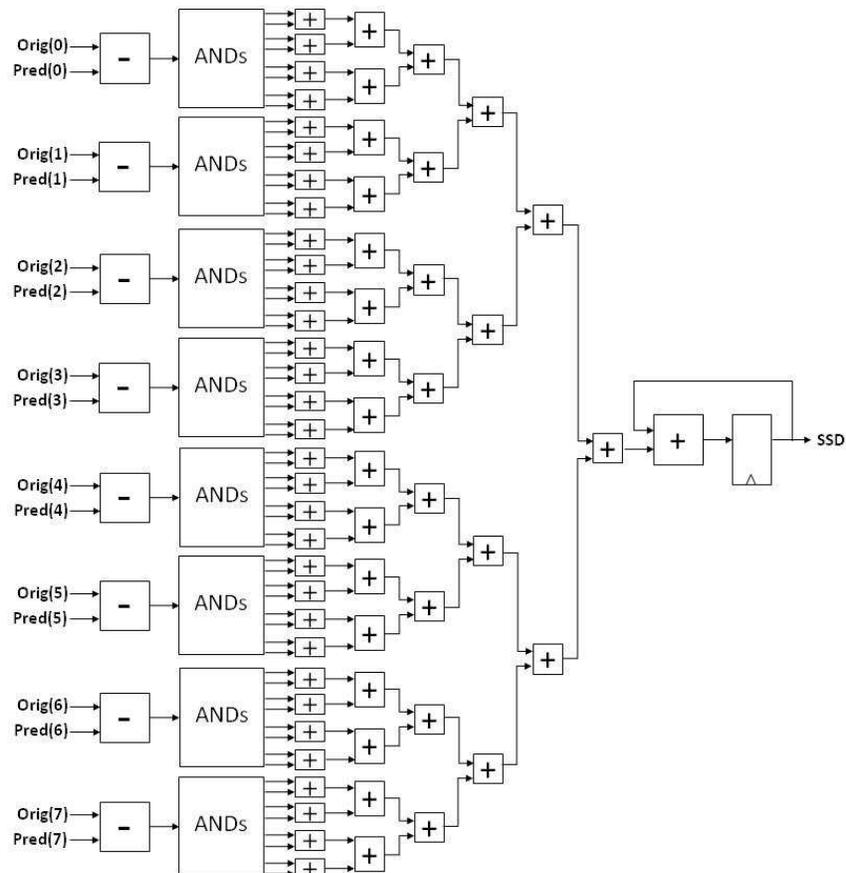


Figura 5.3: Arquitetura SSD com 16 amostras de entrada.

5.1.1.4 Arquitetura para Cálculo de SAD, SATD e SSD

Além das arquiteturas dedicadas a cada uma das métricas de distorção apresentadas nas seções anteriores, este trabalho apresenta uma arquitetura hierárquica que permite escolher, dentre um conjunto de critérios de similaridade, qual deles será utilizado para a decisão do modo de codificação intra-quadro (CORRÊA, 2010). Ao permitir escolher qual dos critérios é utilizado, a arquitetura também permite escolher a precisão do cálculo de distorção, a potência dissipada e a energia gasta no processo de codificação, já que essas características diferem de métrica para métrica. Assim, a escolha do modo de codificação pode ser adaptado às diferentes condições de um codificador, tais como baixa dissipação de potência/consumo de energia ou a requisitos de alta qualidade de codificação.

O objetivo desta arquitetura com três métricas de distorção é aproveitar os recursos lógicos disponíveis (operadores e elementos de armazenamento) e utilizá-los no cálculo de mais de um critério de similaridade, reduzindo a área ocupada e a potência dissipada pelo circuito, em comparação à utilização de simples instâncias das três métricas. Assim, para o projeto desta arquitetura foram investigados quais os registradores e quais os operadores aritméticos (somadores e subtratores) são utilizados em mais de uma métrica.

A Figura 5.4 apresenta a arquitetura desenvolvida. Como se pode perceber, cada estágio está separado do estágio anterior através de uma barreira de registradores, formando um extenso *pipeline*. Como exemplo de reutilização de recursos, as árvores de somadores das arquiteturas de SATD e SSD são implementadas com os mesmos somadores e registradores, já que nunca são utilizadas ao mesmo tempo.

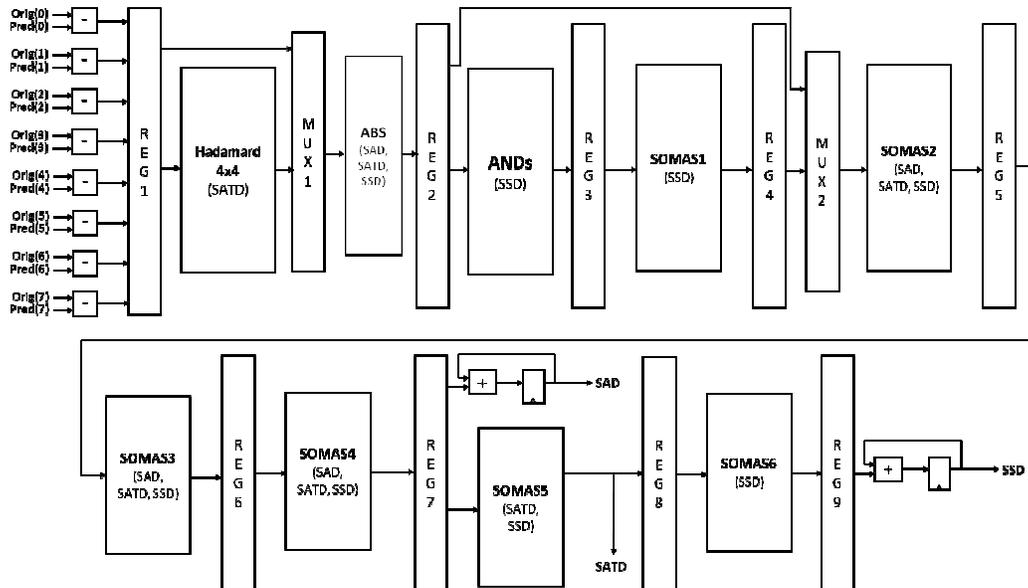


Figura 5.4: Arquitetura de SAD, SATD e SSD com compartilhamento de recursos lógicos.

No cálculo de SAD, as subtrações iniciais são realizadas e armazenadas nos primeiros registradores temporais (REG1). O primeiro grupo de multiplexadores (MUX1) seleciona as diferenças para o cálculo de módulos, cujo resultado é armazenado em outra barreira de registradores (REG2). Em seguida, o segundo grupo multiplexador (MUX2) seleciona a saída de REG2 e realiza as somas da árvore de

somadores (SOMAS2, SOMAS3 e SOMAS4). Da mesma forma, os módulos subtratores, REG1, Hadamard 4x4, MUX1, ABS, REG2, MUX2, SOMAS2, REG5, SOMAS3, REG6, SOMAS4, REG7 e SOMAS5 são utilizados para gerar o valor de SATD. Por fim, o valor de SSD é gerado através da utilização de todos os módulos, com exceção do calculador de Hadamard 4x4 e do acumulador para cálculo de SAD.

Como nem todos os registradores são utilizados durante o cálculo de todas as métricas, foi aplicada a técnica de *clock gating* com o objetivo de diminuir a atividade de chaveamento do *clock* em tais registradores e, por consequência, diminuir a potência dinâmica dissipada. Desta forma, três sinais de *clock* foram criados, além do sinal de *clock* original: clk_SAD, clk_SATD e clk_SSD. Registradores utilizados no cálculo de apenas uma das métricas foram descritos em processos sensíveis aos *clocks* correspondente a essas métricas. Registradores utilizados no cálculo de duas métricas foram descritos em processos sensíveis aos dois *clocks* correspondentes. Já os registradores utilizados no cálculo de todas as métricas são sensíveis ao *clock* original. Os novos sinais de *clock* são gerados através do sinal de *clock* original e do sinal de controle que indica qual critério está sendo calculado no momento.

5.1.1.5 Resultados Obtidos

A Tabela 5.1 mostra os resultados obtidos em termos de utilização de recursos lógicos e frequência máxima atingida por cada uma das arquiteturas quando implementadas para o dispositivo FPGA XC5VLX30 da família Virtex 5 da Xilinx. Além disso, as duas últimas colunas apresentam os resultados obtidos em termos de área e potência dissipada quando as arquiteturas foram implementadas em *standard cells* da tecnologia TSMC 0,18 μm . Os resultados de potência foram obtidos através da ferramenta Power Compiler, da Synopsys (SYNOPTSYS, 2010), e os resultados de área foram obtidos através da ferramenta LeonardoSpectrum, da Mentor Graphics.

Tabela 5.1: Resultados das arquiteturas para cálculo de distorção em termos de área, frequência e potência implementadas como *standard cells* e em FPGA.

Versão	#LUTs	F _{máx} (MHz)	Área (μm^2)	Potência (mW) @ 100 MHz
SAD s/ pipeline	208	469,4	35.394	2,1
SATD s/ pipeline	1449	94,3	296.032	14,78
SSD s/ pipeline	1148	111,5	148.605	15,08
SAD+SATD+SSD c/ pipeline sem compartilhamento	2236	415,6	480.789	65 @SAD 65 @SATD 65 @SSD
SAD+SATD+SSD c/ pipeline com compartilhamento	1816	362,8	407.057	12,9 @ SAD 22,9 @ SATD 19,0 @ SSD

Em ambas as tecnologias, a arquitetura de SAD é a que apresenta a menor área (ou número de LUTs) e a maior frequência máxima. Além disso, a arquitetura SAD é a que apresenta a menor dissipação de potência quando implementada em *standard cells*. A arquitetura com as três métricas que compartilha recursos lógicos apresentou uma área um pouco menor do que a arquitetura com três métricas sem compartilhamento de recursos, representando uma boa opção para uso em dispositivos nos quais os requisitos de energia e qualidade da imagem são variáveis.

A Figura 5.6 apresenta as mesmas operações reordenadas de forma que os padrões pudessem ser percebidos de uma maneira mais fácil. As 256 operações por linha apresentadas na Figura 5.5 foram reordenadas e agrupadas em 16 grupos de 16 operações na Figura 5.6. As 16 primeiras colunas da Figura 5.6 mostram as 16 primeiras operações (reordenadas) realizadas no cálculo de cada um dos 24 coeficientes. Como se pode perceber, para cada coluna, uma grande parte das operações nas 24 linhas são idênticas e, portanto, não precisam ser repetidas. Por exemplo, os cálculos que utilizam a primeira amostra do macrobloco são 12 somas, 6 deslocamentos de 1 bit à direita e 6 somas, 2 deslocamentos de 2 bits à direita e 2 somas e 4 deslocamentos de 3 bits à direita e 4 somas, conforme pode-se perceber na primeira coluna da Figura 5.6. Após o agrupamento, as operações repetidas podem ser ignoradas e o conjunto relativo à primeira amostra resume-se a 1 soma, 1 deslocamento de um bit à direita e 1 soma, 1 deslocamento de 2 bits à direita e 1 soma e 1 deslocamento de 3 bits à direita e 1 soma.

O agrupamento descrito levou a uma redução de 6144 operações (256 operações por coeficiente) para apenas 373 operações.

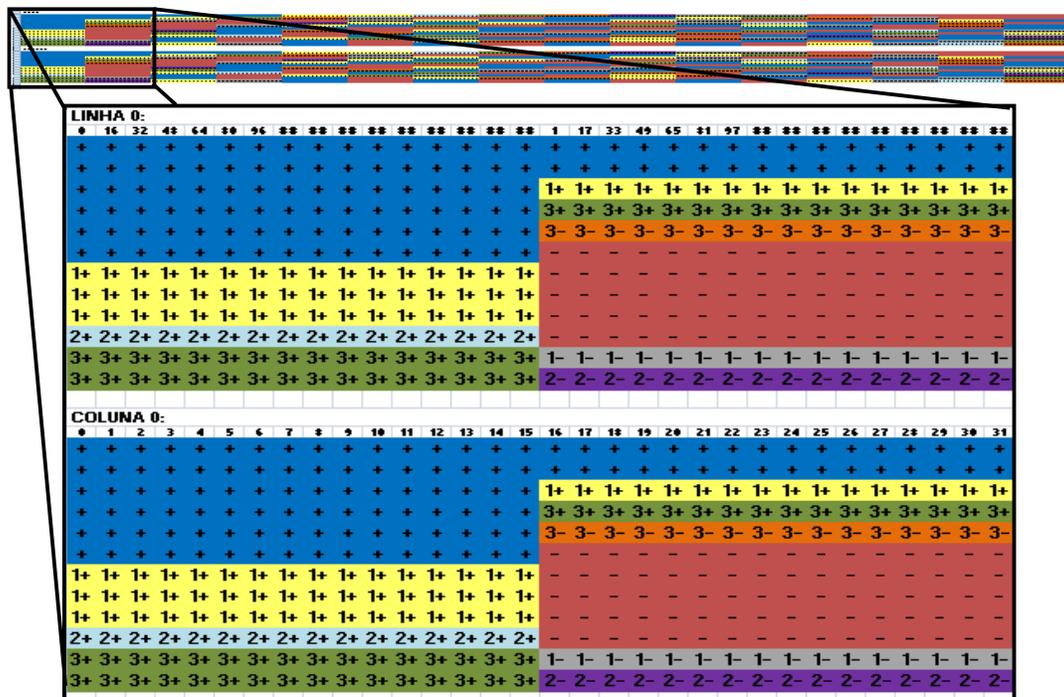


Figura 5.6: Operações para o cálculo de cada coeficiente após o agrupamento.

Para maiores simplificações, os deslocamentos foram realizados após as somas ou subtrações que possuem deslocamentos iguais. Desta forma, apenas operações aritméticas são realizadas nos primeiros passos do cálculo de cada coeficiente, postergando os deslocamentos iguais a serem realizados sobre os resultados das operações e não sobre os seus operandos.

A Figura 5.7 ilustra a arquitetura desenvolvida para o cálculo dos coeficientes da DCT 16x16 parcial. A entrada da arquitetura é composta pelas 256 amostras que compõem o macrobloco original. Como os cálculos realizados para os coeficientes da primeira linha do macrobloco são diferentes dos cálculos realizados para os coeficientes da primeira coluna do macrobloco, um seletor é utilizado para ordenar as amostras de entrada que são entregues aos módulos somadores (MS, na Figura 5.7). Os módulos somadores são responsáveis por realizar as somas/subtrações das 16 amostras que

compõem cada um dos agrupamentos de 16 colunas apresentados na Figura 5.6. A árvore de somadores realiza a soma dos resultados parciais gerados pelos módulos MS e também realiza os deslocamentos que foram postergados. Inicialmente, são gerados e armazenados no *buffer* de saída os valores dos 12 coeficientes que pertencem à primeira linha do macrobloco. Logo depois, o processo é repetido para o cálculo dos coeficientes da primeira coluna do macrobloco. Neste caso, o controle seleciona uma nova ordenação no conjunto de operandos que são entregues aos módulos somadores e 12 outras posições no *buffer* de saída do módulo.

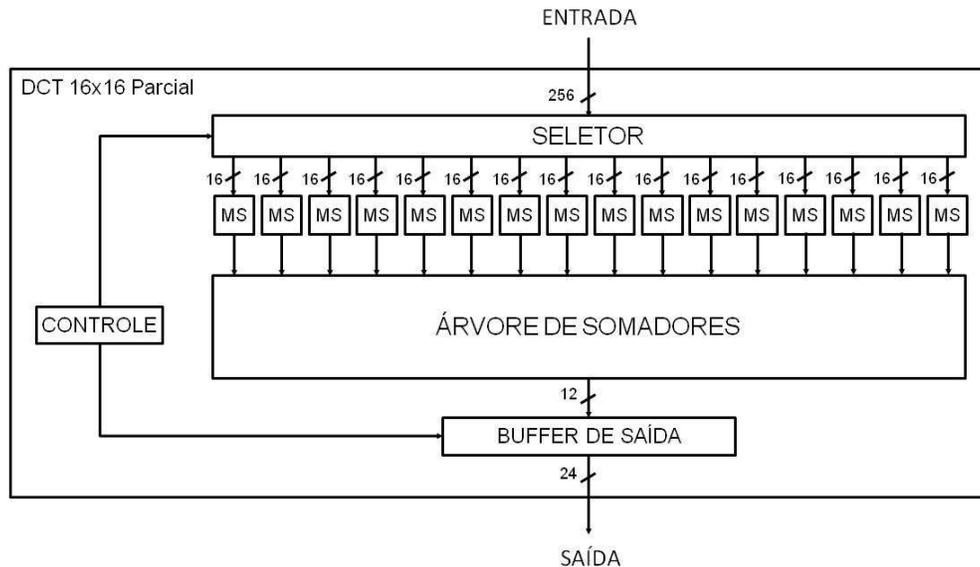


Figura 5.7: Arquitetura da DCT 16x16 parcial.

Os módulos MS são compostos por 4 estágios de *pipeline*, assim como a árvore de somadores. Desta forma, 8 ciclos são necessários para o cálculo dos coeficientes pertencentes à primeira linha e 8 ciclos são necessários para o cálculo dos coeficientes pertencentes à primeira coluna, totalizando 16 ciclos para o cálculo dos 24 coeficientes utilizados para a decisão de tamanho de bloco intra-quadro.

A arquitetura foi sintetizada com a ferramenta ISE 10.1 da Xilinx para o dispositivo FPGA XC5VLX30 da família Virtex 5. Foram utilizadas 5198 LUTs. A frequência máxima de operação atingida foi de 543 MHz.

5.1.3 Arquiteturas para Decisão de Melhor Modo Intra-Quadro

Por apresentar a menor complexidade e utilização de recursos lógicos dentre todas as arquiteturas desenvolvidas para cálculo de distorção, o cálculo de SAD foi utilizado na arquitetura que decide o melhor modo intra-quadro. A integração entre as instâncias da arquitetura de SAD e a arquitetura da DCT 16x16 parcial foi realizada de forma que os cálculos de melhor tamanho de bloco e de distorção fossem realizados em paralelo.

Nove instâncias da arquitetura SAD 4x4 foram utilizadas para o cálculo de distorção dos nove modos intra-quadro 4x4 e quatro instâncias de SAD 4x4 foram utilizadas para o cálculo de distorção dos quatro modos intra-quadro 16x16, conforme mostrado na Figura 5.8. Os valores de distorção para os 9 modos 4x4 são calculados em 2 ciclos para cada um dos 16 blocos 4x4, totalizando 32 ciclos. Já a distorção para cada um dos 4 modos 16x16 é calculada em 32 ciclos, pois cada macrobloco é composto por 16 blocos 4x4.

Os 24 coeficientes gerados pela DCT 16x16 parcial são calculados em 16 ciclos e somados em uma estrutura de árvore de somas em 2 ciclos. O resultado final é, então, comparado com o valor limite proposto na seção 4.2.2.2. São utilizados, portanto, 35 ciclos para a decisão completa do melhor modo intra-quadro.

A Figura 5.8 mostra a arquitetura completa para a decisão intra-quadro, que foi sintetizada com a ferramenta ISE 10.1 da Xilinx para o dispositivo FPGA XC5VLX30 da família Virtex 5. Na Figura 5.8, a saída do módulo de decisão pode ser apenas um modo, no caso da predição intra-quadro 16x16, ou um vetor com 16 modos, um para cada bloco 4x4, no caso da predição intra-quadro 4x4. Foram utilizadas 7940 LUTs na arquitetura completa e a frequência máxima de operação atingida foi de 105 MHz.

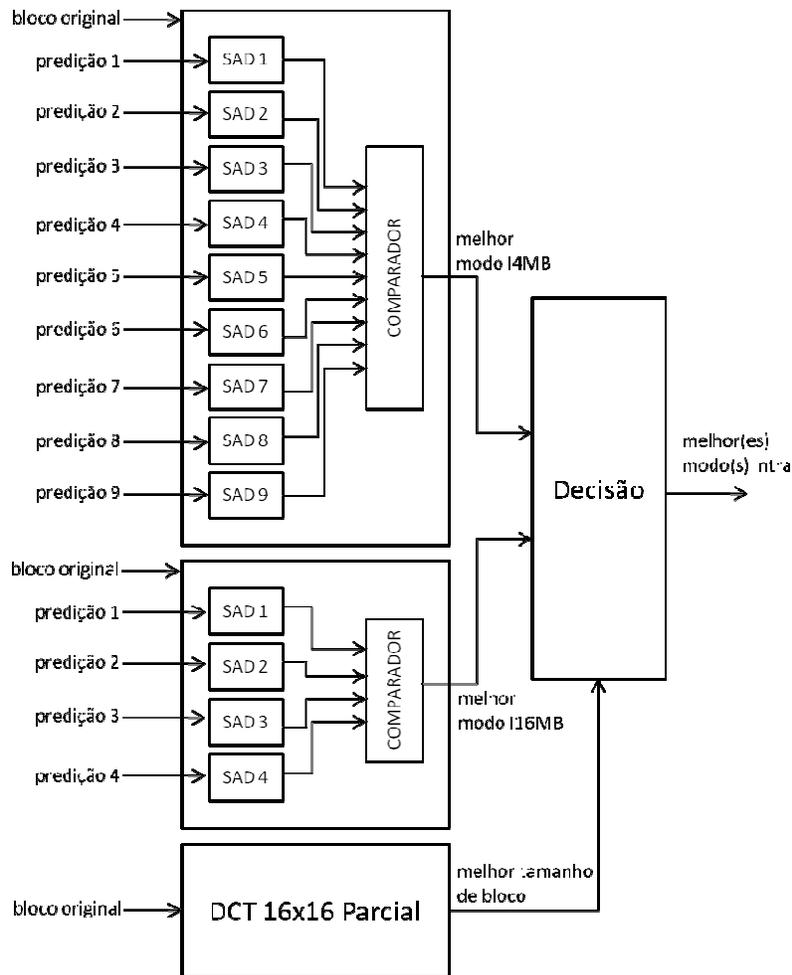


Figura 5.8: Arquitetura para decisão de modo intra-quadro.

5.2 Arquiteturas para Decisão Inter-Quadros

Conforme explicado anteriormente, a decisão inter-quadros foi dividida em decisão de modo *SKIP* baseada em estacionaridade, decisão de sub-particionamento baseada em heterogeneidade e decisões de tamanho de partição e de tamanho de sub-partição baseadas em força de borda.

A arquitetura para detecção de estacionaridade é baseada no cálculo de distorção e utiliza o mesmo módulo de SAD apresentado na seção 5.1.1.1. A decisão de sub-

particionamento baseada em heterogeneidade é realizada pela mesma arquitetura da DCT 16x16 parcial apresentada na seção 5.1.2.

As próximas seções apresentam as arquiteturas para cálculo de força de borda desenvolvidas para as decisões de tamanho de partição e de tamanho de sub-partição, que são diferentes das soluções arquiteturais apresentadas até este ponto do texto.

5.2.1 Arquitetura para Cálculo de Força de Bordas em Partições

A arquitetura desenvolvida para o cálculo de força de bordas em partições é similar à apresentada para o cálculo de distorção por SAD. Contudo, apenas 8 amostras do bloco original (P3, P2, P1, P0, Q0, Q1, Q2 e Q3) compõem a entrada da arquitetura, sendo 4 delas pertencentes a cada lado da borda. A arquitetura, apresentada na Figura 5.9, utiliza 16 ciclos para calcular a força de borda vertical e 16 ciclos para calcular a força de borda horizontal do bloco. O cálculo da força de borda em cada linha é realizado em um ciclo e o resultado é armazenado em um acumulador. A decisão de tamanho de partição é realizada por um módulo de decisão, conforme explicado na seção 5.2.3.

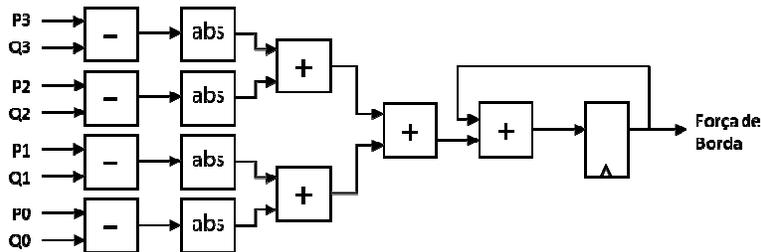


Figura 5.9: Arquitetura para cálculo de força de borda em partições.

5.2.2 Arquitetura para Cálculo de Força de Bordas em Sub-Partições

Como se pode perceber na Figura 5.10, este módulo é bastante similar ao apresentado na seção anterior. Contudo, são utilizadas apenas 4 amostras do bloco original (P1, P0, Q0 e Q1) para o cálculo de força de borda em sub-partições, sendo 2 delas pertencentes a cada lado da borda, de forma que apenas um somador é necessário (além do acumulador), ao invés de uma árvore com dois níveis de soma.

Conforme explicado na seção 4.3.2, o cálculo da borda de sub-partição é realizado em duas partes: a força da metade superior (ou à esquerda) e a força da metade inferior (ou à direita) da sub-partição. São utilizados, portanto, 4 ciclos para computar a força de borda de cada metade da sub-partição, totalizando 8 ciclos para o cálculo da força de borda da sub-partição em cada sentido. São instanciados, portanto, 8 módulos calculadores de força de borda em sub-partição, sendo cada um deles responsável por metade da borda de cada uma das 4 sub-partições 8x8.

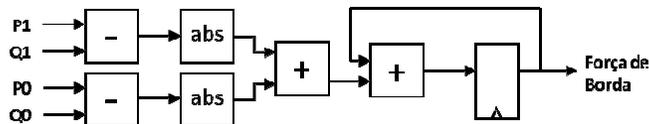


Figura 5.10: Arquitetura para cálculo de força de borda em sub-partições.

5.2.3 Arquiteturas para Decisão de Melhor Modo Inter-Quadros

Os módulos que compõem a decisão de melhor modo inter-quadros foram integrados na estrutura mostrada pela Figura 5.11. A estacionaridade entre o

macrobloco original e o macrobloco de referência é medida através do calculador de SAD 4x4 que realiza o cálculo para um macrobloco em 32 ciclos (2 ciclos para cada bloco 4x4).

O cálculo das forças de borda horizontal e vertical de partição também é realizado em 32 ciclos (16 ciclos para cada sentido). Já o cálculo das forças de borda horizontal e vertical cada sub-partição é realizado em apenas 8 ciclos (4 ciclos para cada sentido). A DCT 16x16 parcial é calculada em 16 ciclos e a árvore de somas realiza o cálculo do grau de heterogeneidade em 2 ciclos, conforme explicado anteriormente.

O módulo de decisão é composto por comparadores que operam sobre os valores calculados pelos módulos precedentes e os valores-limite apresentados na seção 4.2, além dos multiplexadores que selecionam os modos de acordo com os resultados das comparações. No final da decisão, apenas um modo é disponibilizado na saída, no caso de não haver sub-particionamento. Caso contrário, um conjunto de 4 modos é disponibilizado na saída, um para cada sub-macrobloco 8x8.

A arquitetura completa foi sintetizada com a ferramenta ISE 10.1 da Xilinx para o dispositivo FPGA XC5VLX30 da família Virtex 5. Foram utilizadas 6432 LUTs e a frequência máxima de operação atingida foi de 118 MHz.

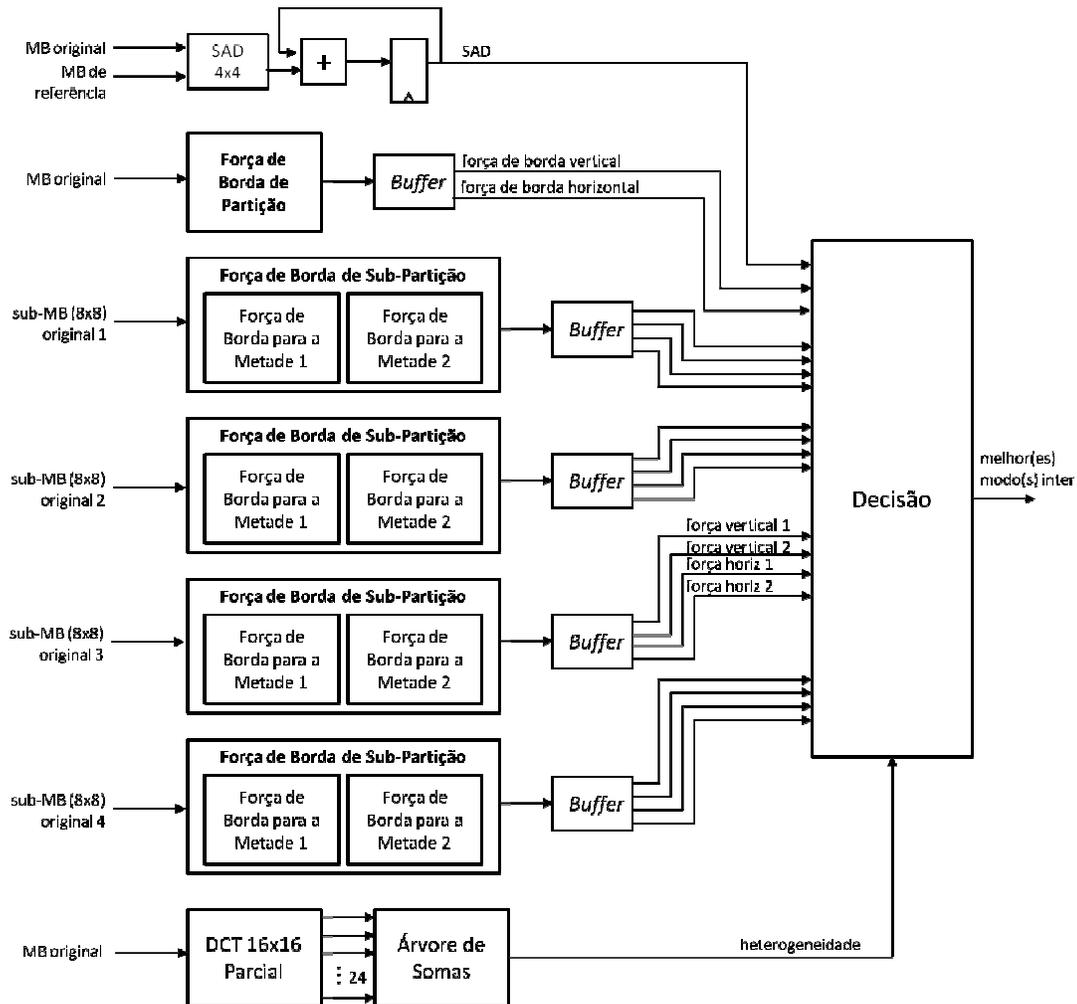


Figura 5.11: Arquitetura para decisão de modo inter-quadros.

5.3 Análise dos Resultados de Síntese

A Tabela 5.2 apresenta um resumo dos resultados de síntese obtidos para as arquiteturas descritas neste capítulo. Os resultados para as arquiteturas de decisão intra-quadro e inter-quadros são apresentados, bem como os resultados para os módulos que as compõem.

Tabela 5.2: Resultados das arquiteturas desenvolvidas em termos de recursos lógicos utilizados e frequência máxima para FPGAs Virtex 5 da Xilinx.

Arquitetura	#LUTs	Slice Reg.	LUT-FF pairs	F _{máx} (MHz)
Decisão Intra-Quadro	7940	3111	1798	105,0
Cálculo de SAD	208	13	13	469,4
Deteção de Heterogeneidade	5198	4617	3098	543,0
Decisão Inter-Quadros	6432	2071	1692	118,0
Força de Borda (Partição)	130	91	35	157,7
Força de Borda (Sub-partição)	64	50	22	200,0

Como se pode perceber, o módulo de deteção de heterogeneidade é o responsável pela maior parte da utilização de recursos lógicos do FPGA. Em trabalhos futuros, pretendem-se investigar alternativas arquiteturais para a redução desta solução. Também se percebe que a limitação da frequência máxima de operação encontra-se nos módulos que realizam as comparações e decisões baseadas nos valores fornecidos pelos módulos que calculam distorção, heterogeneidade e força de borda, já que estes apresentam frequências superiores às das arquiteturas completas.

Até o momento de término desta dissertação, apenas as arquiteturas de cálculo de SAD e deteção de heterogeneidade haviam sido validadas. A metodologia utilizada para a validação de ambas as arquiteturas foi a comparação entre os resultados obtidos com simulações através da ferramenta ModelSim (MENTOR, 2010) e os resultados obtidos através de execuções de um software em linguagem C desenvolvido com a mesma finalidade da arquitetura. Os próximos passos deste trabalho consistem na validação completa do conjunto de arquiteturas.

Levando em consideração que a menor frequência obtida foi a da arquitetura de decisão intra-quadro (105 MHz), que também é a arquitetura que utiliza mais ciclos para realizar a decisão, pode-se dizer que um quadro HD 1080p é processado pela arquitetura de decisão de modo em 285.600 ciclos (35 ciclos para cada um dos seus 8.160 macroblocos). Isto significa que a arquitetura é capaz de processar até 367 quadros por segundo, atingindo com folga o objetivo de 24 a 30 quadros por segundo para codificação de vídeo em tempo real.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou uma investigação e o desenvolvimento de soluções que visam à redução do processo de decisão do melhor modo de codificação de vídeo segundo o padrão H.264/AVC. A decisão de modo proposta pelo software de referência do padrão (SUHRING, 2010) consiste na execução de um fluxo de codificação completo para cada possível modo existente, limitando drasticamente o desempenho do codificador.

A principal contribuição deste trabalho encontra-se no desenvolvimento de heurísticas testadas através de modificações no software de referência do padrão H.264/AVC. As análises iniciais mostraram que a ocorrência de modos intra-quadro em quadros P é mínima, de forma que a decisão pôde ser dividida em dois processos separados: a decisão intra-quadro para quadros I e a decisão inter-quadros para quadros P.

A decisão intra-quadro proposta foi dividida em uma etapa de decisão de melhor modo para tamanho fixo de bloco baseada em distorção e em uma etapa de decisão de melhor tamanho de bloco baseada em heterogeneidade. Esta abordagem gerou resultados bastante próximos aos obtidos com a técnica de RDO. A decisão inter-quadros foi dividida em uma etapa de detecção de estacionaridade para decisão de modo *SKIP*, detecção de heterogeneidade para sub-particionamento e análise de bordas internas das partições e sub-partições para decisão de tamanho de particionamento e sub-particionamento. Os resultados obtidos em termos de taxa de bits e de PSNR também se aproximaram bastante aos gerados pela decisão RDO.

Quando todas as técnicas propostas para decisão foram combinadas, notou-se um aumento de 7,53% na taxa de bits e um aumento de 0,05 dB na qualidade da imagem. O aumento na taxa de bits é relativamente baixo quando comparado à diminuição na ordem de 47 vezes no número de cálculos para decisão de modo proporcionada pela utilização das heurísticas. O laço de repetição que realiza a codificação completa para todos os modos possíveis segundo a decisão RDO foi reduzido a apenas uma iteração de codificação através da utilização das heurísticas propostas. Em comparação com trabalhos relacionados, a decisão rápida proposta neste trabalho mostrou-se expressivamente menos complexa, apresentando codificação com qualidade de imagem equivalente ou superior aos outros trabalhos. A taxa de bits sofreu acréscimo de até 4% em comparação a outros trabalhos, um valor ainda pequeno quando levada em consideração a grande redução da complexidade computacional.

Após as análises de desempenho de cada heurística separadamente e de todas as heurísticas integradas, uma arquitetura de hardware para o módulo de decisão de modo intra-quadro e uma arquitetura para decisão de modo inter-quadros foram descritas em VHDL e sintetizadas para dispositivos FPGA Virtex 5 da Xilinx. Dentre os módulos que compõem as arquiteturas, a DCT 16x16 parcial foi a que apresentou a maior

utilização de recursos lógicos. A frequência máxima da arquitetura para decisão intra-quadro desenvolvida ficou em torno de 105 MHz, enquanto que a arquitetura para decisão inter-quadros pode operar em uma frequência de até 118 MHz. Ambas as arquiteturas atingem o requisito de realizar a codificação em tempo real com folga.

Durante o período de elaboração deste trabalho, foram publicados dois artigos que descrevem parte das heurísticas e dos resultados obtidos:

- *Comparative Analysis of Parallel SAD Calculation Hardware Architectures for H.264/AVC Video Coding* (IEEE LASCAS 2010) (DINIZ, 2010) e
- *Adaptive Distortion Metric Architecture for H.264/AVC Video Coding* (SIM 2010) (CORRÊA, 2010).

Dois outros trabalhos foram submetidos e estão em fase de avaliação:

- *Homogeneity and Distortion-Based Intra Mode Decision Architecture for H.264/AVC* (IEEE ICECS 2010) e
- *Fast Hierarchical Mode Decision for H.264/AVC Video Coding Standard* (Multimedia Tools and Applications Journal – Springer).

Como trabalhos futuros, pretende-se realizar uma série de atividades relacionadas à investigação de heurísticas e às arquiteturas desenvolvidas, tais como:

- Avaliar a detecção de heterogeneidade através da utilização de Operador de Sobel e a sua aplicação nas decisões de tamanho de bloco intra-quadro e de utilização de sub-particionamento em macroblocos P;
- Avaliar a detecção de bordas internas de partições e sub-partições através da utilização de outras técnicas de detecção de bordas, como o Operador de Sobel, o Operador de Prewitt e o Algoritmo de Canny;
- Validar completamente e integrar as arquiteturas de decisão intra-quadro e de decisão inter-quadros com os módulos de predição intra-quadro e de estimação de movimento, respectivamente;
- Investigar o desempenho e a adaptabilidade das heurísticas a bi-predição (isto é, habilitando a possibilidade de quadros B);
- Investigar o desempenho das heurísticas com a utilização de predição fracionária (*half* e *quarter* pixel);
- Investigar estratégias de redução de área, especialmente para o módulo de detecção de heterogeneidade (DCT 16x16 parcial).

REFERÊNCIAS

- AGOSTINI, L. V., **Desenvolvimento de Arquiteturas de Alta Performance Dedicadas à Compressão de Vídeo Segundo o Padrão H.264**. Tese (Doutorado) - Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, RS, 2007.
- BHASKARAN, V.; KONSTANTINIDES, K. **Image and Video Compression Standards: Algorithms and Architectures**. 2nd ed. Boston: Kluwer Academic Publishers, 1997.
- CHANG, C.-H.; et al. A Quality Scalable H.264/AVC Baseline Intra Encoder for High Definition Video Applications. In: Workshop on Signal Processing Systems (SiPS), IEEE, 2007. **Proceedings...** Shanghai: IEEE, 2007.
- CHOI, I.; et al. Fast coding mode selection with rate-distortion optimization for MPEG-4 Part-10 AVC/H.264. **IEEE Trans. Circuits Syst. Video Technol.**, v. 16, n. 12, p. 1557-1561, dez. 2006.
- CHU, C. et al. Hierarchical Global Motion Estimation/Compensation in Low Bitrate Video Coding. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, ISCAS, 1997. **Proceedings...** New York: IEEE, 1997. p. 1149-1152.
- CORRÊA, G.; DINIZ, C.; AGOSTINI, L.; BAMPI, S.. **Adaptive Distortion Metric Architecture for H.264/AVC Video Coding**. In: South Symposium on Microelectronics (SIM), 2010, **Proceedings...** Porto Alegre: EDIPUCRS, 2010.
- DINIZ, C., **Arquitetura de Hardware Dedicada para a Predição Intra-Quadro em Codificadores do Padrão H.264/AVC de Compressão de Vídeo**. Dissertação (Mestrado) - Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, RS, 2009.
- DINIZ, C.; CORRÊA, G.; SUSIN, A.; BAMPI, S.. **Comparative Analysis of Parallel SAD Calculation Hardware Architectures for H.264/AVC Video Coding**. In: Latin American Symposium on Circuits and Systems (LASCAS), IEEE, 2010, **Proceedings...** Foz do Iguaçu: IEEE, 2010.
- FENGQIN, W. et al. Fast intra mode decision algorithm in H.264/AVC using characteristics of transformed coefficients. In: International Conference on Visual Information Engineering, VIE, 2008. **Proceedings...** Xian: IEEE, 2008.
- GHANBARI, M. **Standard Codecs: Image Compression to Advanced Video Coding**. United Kingdom: The Institute of electrical Engineers, 2003.
- GONZALEZ, R.; WOODS, R. **Processamento de Imagens Digitais**. São Paulo: E. Blücher, 2003.

HUANG, Y.-W.; et al. A 1.3TOPS H.264/AVC Single-Chip Encoder for HDTV Applications. In: International Solid-State Circuits Conference (ISSCC), IEEE, 2005. **Proceedings...** San Francisco: IEEE, 2005.

HUANG, H. et al. An Efficient Fast Inter Mode Decision Algorithm for H.264-AVC. In: International Conference on Communications and Mobile Computing (CMC), IEEE, 2009. **Proceedings...** Kunming: IEEE, 2009.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation H.262 (11/94)**: generic coding of moving pictures and associated audio information – part 2: video. [S.l.], 1994.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation H.263 v3 (11/00)**: video coding for low bit rate communication. [S.l.], 2000.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation H.264/AVC (05/03)**: advanced video coding for generic audiovisual services. [S.l.], 2003.

INTERNATIONAL TELECOMMUNICATION UNION. **JVT-L050**: H.264/AVC AVC Fidelity Range Extensions. [S.l.], 2004.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation H.264/AVC (03/05)**: advanced video coding for generic audiovisual services. [S.l.], 2005.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation H.264/AVC (11/07)**: advanced video coding for generic audiovisual services. [S.l.], 2007.

INTERNATIONAL TELECOMMUNICATION UNION. ITU-T Home. Disponível em: <www.itu.int/ITU-T/>. Acesso em: maio 2009.

INTERNATIONAL Organization for Standardization. MPEG - Moving Pictures Working Group. Disponível em: <<http://www.chiariglione.org/mpeg/>>. Acesso em: maio 2009.

JAIN, J.; JAIN, A. Displacement Measurement and Its Application in Interframe Image Coding. **IEEE Transactions on Communications**, [S.l.], v. 29, n. 12, p. 1799-1808, Dec. 1981.

JEON, Y.-I. et al. Fast Intra Mode Decision Algorithm Using Directional Gradients for H.264. In: International Congress on Image and Signal Processing, CISP, 2009. **Proceedings...** Tianjin: IEEE, 2009.

JOINT COLLABORATIVE TEAM ON VIDEO CODING. **JCT-VC Call for Proposals on Video Compression Technology**: VCEG-AM91, 2010.

KANNANGARA, C.; RICHARDSON, I. Computational Control of an H.264/AVC Encoder through Lagrangian Cost Function Estimation. In: International Workshop on Very Low Bitrate Video, 2005. **Proceedings...** [S.l.:s.n.], 2005.

KIM, Y.-H.; et al. Adaptive mode decision for H.264 encoder. **Electronic Letters**, v.40, n.19, p. 1172-1173, set. 2004a.

KIM, H.; et al. Low-complexity macroblock mode selection for H.264-AVC encoders. In: International Conference on Image Processing, ICIP, 2004. **Proceedings...** Singapura: IEEE, 2004b.

- KUHN, P. **Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation**. Boston: Kluwer Academic Publishers, 1999.
- KROUPIS, N. et al. A Modified Spiral Search Motion Estimation Algorithm and its Embedded System Implementation. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, ISCAS, 2005. **Proceedings...** [S.l.]: IEEE, 2005. p. 3347-3350.
- KUO, H.-C.; LIN, Y.-L. An H.264/AVC Full-Mode Intra-Frame Encoder for 1080HD Video. In: International Conference on Multimedia and Expo (ICME), IEEE, 2008. **Proceedings...** Hannover: 2008.
- LEE, J; JEON, B. Fast mode decision for H.264. In: International Conference on Multimedia and Expo, ICME, 2004. **Proceedings...** Taipei: ICME, 2004.
- LEE, Y.-M.; et al. An improved SATD-Based Intra Mode Decision Algorithm for H.264/AVC. In: International Conference on Acoustics, Speech, and Signal Processing, ICASSP, 2009. **Proceedings...** Taipei: ICASSP, 2009.
- LI, R. et al. A New Three-Step Search Algorithm for Block Motion Estimation. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v. 4, n. 4, p. 438-442, ago. 1994.
- LI, D.-W.; et al. A 61MHz 72K Gates 1280x720 30FPS H.264 Intra Encoder. In: International Conference on Acoustic, Speech and Signal Processing (ICASSP), IEEE, 2007. **Proceedings...** Honolulu : IEEE, 2007.
- LIU, P.; JIA, K. A Fast Mode Decision Scheme with Variable Block Sizes in H.264/AVC. In: International Conference on Intelligent Systems Design And Applications, ISDA, 2008. **Proceedings...** [S.l.]: IEEE, 2008. v. 3. p. 368 – 371.
- LIN, C.-C.; et al. A 160K Gates/4.5 KB SRAM H.264 Video Decoder for HDTV Applications. **IEEE Journal of Solid-State Circuits**, [S.l.], v.42, n.1, p. 170-182, jan. 2007.
- MA, W. et al. An Efficient Fast Mode Decision Algorithm Based on Motion Cost for H.264 Inter Prediction. In: International Symposium on Intelligent Information Technology Application, IITAW, 2008. **Proceedings...** [S.l.]: IEEE, 2008. p. 550 – 553.
- MENTOR GRAPHICS CORPORATION. **ModelSim SE User's Manual - Software Version 6.2d**. [S.l.], 2010. Disponível em: <www.model.com/support>. Acesso em: ago. 2010.
- PAN, L; HO, Y. A Fast Mode Decision Algorithm for H.264/AVC Intra Prediction. IEEE 2007 Workshop on Signal Processing Systems, **Proceedings...**, p. 704-709, 2007.
- PORTO, M., **Arquiteturas de Alto Desempenho e Baixo Custo em Hardware para a Estimaco de Movimento em Vdeos Digitais**. Dissertao (Mestrado) - Universidade Federal do Rio Grande do Sul. Programa de Ps-Graduao em Computao, Porto Alegre, RS, 2008.
- PURI, A. et al. **Video Coding Using the H.264/MPEG-4 AVC Compression Standard**. Elsevier Signal Processing: Image Communication, [S.l.], n. 19, p.793–849, 2004.
- RICHARDSON, I. **Video Codec Design – Developing Image and Video Compression Systems**. Chichester: John Wiley and Sons, 2002.

RICHARDSON, I. **H.264/AVC and MPEG-4 Video Compression – Video Coding for Next-Generation Multimedia**. Chichester: John Wiley and Sons, 2003.

SAHAFI, L. **Context-Based Complexity Reduction Applied to H.264/AVC Video Compression**. 2005. 70 f. Thesis (Master in Applied Science) – School of Engineering Science, Simon Fraser University, Canada.

SHEN, L. et al. Fast Inter Mode Decision Using Spatial Property of Motion Field. **IEEE Transactions on Multimedia**, [S.l.], v. 10, n. 6, p. 1208 – 1214, out. 2008.

SHI, Y.; SUN, H. **Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms and Standards**. Boca Raton: CRC Press, 1999.

SUH, K.; PARK, S.; CHO, H. An Efficient Hardware Architecture of Intra Prediction and TQ/IQIT Module for H.264 Encoder. **ETRI Journal**, [S.l.], v. 27, n. 5, p. 511-524, out. 2005.

SUHRING, K. H.264/AVC Reference Software. **In: Fraunhofer Heinrich-Hertz-Institute**. Disponível em: <<http://iphome.hhi.de/suehring/tml/download/>>. Acesso em: mai. 2010.

SULLIVAN, G.; WIEGAND, T. Rate-Distortion Optimization for Video Compression. **IEEE Signal Processing Magazine**, [S.l.], v. 15, p. 74-90, nov. 1998.

SULLIVAN, G. et al. The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions. In: Conference on Applications of Digital Image Processing, SPIE, 2004. **Proceedings...** Denver: SPIE, 2004.

SUN, Y. et al. SATD-based Intra Mode Decision for H.264/AVC Video Coding. In: International Conference on Multimedia and Expo, ICME, 2008, **Proceedings...** Hannover: ICME, 2008.

SYNOPSYS INC. **Synopsys: Predictable Success**. Disponível em: <www.synopsys.com>. Acesso em: ago. 2010.

TOURAPIS, M. et al. Fast Motion Estimation Using Circular Zonal Search. In: Visual Communications and Image Processing. **Proceedings...** San Jose: [s.n.], 1999. v. 2, p. 1496-1504.

TSAI, A.-C. et al. Effective Subblock-Based and Pixel-Based Fast Direction Detections for H.264 Intra Prediction. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v. 18, n. 7, p. 975-982, jul. 2008.

WANG, H.-M. et al. Fast H.264 Inter Mode Decision Based on Inter and Intra Block Conditions. In: International Symposium on Circuits and Systems, ISCAS, 2007. **Proceedings...** [S.l.]: IEEE, 2007. p. 3647-3650.

WIEGAND, T. et al. Rate-Constrained Coder Control and Comparison of Video Coding Standards. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v. 13, n. 7, p. 688-703, jul. 2008.

WU, D. et al. Fast Intermode Decision in H.264/AVC Video Coding. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v. 15, n. 6, p. 953 – 958, jul. 2005.

XILINX INC. **Xilinx: The Programmable Logic Company**. Disponível em: <www.xilinx.com>. Acesso em: ago. 2010.

YI, X.; LING, N. Rapid Block-Matching Motion Estimation Using Modified Diamond Search Algorithm. In: International Symposium on Circuits and Systems, ISCAS, 2005. **Proceedings...** [S.l.]: IEEE, 2005. p. 5489-5492.

YU, A. C. et al. Efficient block size selection algorithm for inter-frame coding in H.264/AVC/MPEG-4 AVC. In: International Conference on Acoustics, Speech, and Signal Processing, ICASSP, 2004. **Proceedings...** Montreal: ICASSP, 2004.

YU, A. et al. Fast Inter-Mode Selection in the H.264/AVC Standard Using a Hierarchical Decision Process. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v. 18, n. 2, p. 186 – 195, fev. 2008.

ZHAN, B. et al. A Fast Intra/Inter Mode Decision Algorithm of H.264/AVC for Real-Time Applications. In: International Conference on Communications, ICC, 2008. **Proceedings...** Beijing: ICC, 2008.