

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

JONAS DOS SANTOS SILVA

**IMPLEMENTAÇÃO DA COMPENSAÇÃO DE MOVIMENTO
EM VÍDEO ENTRELACADO NO TERMINAL DE ACESSO DO
SBTVD**

Porto Alegre

2013

JONAS DOS SANTOS SILVA

**IMPLEMENTAÇÃO DA COMPENSAÇÃO DE MOVIMENTO
EM VÍDEO ENTRELACADO NO TERMINAL DE ACESSO DO
SBTVD**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Engenharia de Computação – Micro e Nano Eletrônica.

ORIENTADOR: Altamiro Amadeu Susin

COORIENTADORA: Leticia Vieira Guimarães

Porto Alegre

2013

JONAS DOS SANTOS SILVA

**IMPLEMENTAÇÃO DA COMPENSAÇÃO DE
MOVIMENTO EM VÍDEO ENTRELACADO NO
TERMINAL DE ACESSO DO SBTVD**

Esta dissertação foi julgada adequada para a
obtenção do título de Mestre em Engenharia
Elétrica e aprovada em sua forma final pelo
Orientador e pela Banca Examinadora.

Orientador: _____

Prof. Dr. Altamiro Amadeu Susin, UFRGS

Banca Examinadora:

Prof. Dr. Sergio Bampi, UFRGS
Doutor pela Stanford University - Stanford, USA

Prof. Dr. Gilson Inácio Wirth, UFRGS
Doutor pela Universitaet Dortmund – Dortmund, Germany

Prof. Dr. Tiago Roberto Balen, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Coordenador do PPGEE:

Prof. Dr. João Manoel Gomes da Silva Junior

Porto Alegre, Agosto de 2013.

DEDICATÓRIA

Dedico este trabalho à minha tia Jane Maria Rosa da Silva e a minha avó Zoraides Rosa da Silva, pela dedicação em todo o desenvolvimento deste trabalho. Dedico a minha avó principalmente pelo apoio e preocupação em fornecer condições para minha dedicação exclusiva ao desenvolvimento deste trabalho. À minha tia por me dar suporte (inclusive financeiro) mas principalmente por acreditar sempre que eu seria capaz de vencer mais esta etapa de minha formação e apostar no meu sucesso.

AGRADECIMENTOS

Ao concluir mais uma importante etapa de meus estudos é imprescindível agradecer a todos que de alguma forma contribuíram para que esse desafio fosse superado.

Gostaria de começar agradecendo à minha família, grande responsável pela pessoa que me tornei e pelos valores que comigo carrego. Ao meu pai Francisco Carlos Rosa da Silva e minha mãe Sandra Maria Lemos dos Santos que sempre estiverem ao meu lado quando precisei de seu apoio. Aos meus irmãos Witor dos Santos Silva e Francini Santos Silva que, apesar da distância, sempre mantiveram uma relação de muito carinho. À minha avó Zoraides Rosa da Silva e a minha tia Jane Maria Rosa da Silva que sempre vibraram com cada conquista que alcancei e me forneceram todo o suporte para a realização deste trabalho. Meus familiares que mais atuaram no auxílio de minha formação, tanto para a graduação quanto para a pós-graduação, foram minha avó Zoraides e minha tia Jane.

Ao colega e amigo M. Sc. Rodolfo Barreto Canônico que me ajudou e incentivou durante a realização das disciplinas. Aos professores Dr. Gilson Inácio Wirth e Dr. Adalberto Schuck Júnior que contribuíram significativamente com minha base de conhecimentos. Ao Dr. Bruno Zatt que dedicou seu tempo para responder todas minhas dúvidas.

Aos meus colegas de trabalho do grupo de pesquisas LaPSI Eng. Bruno Policarpo Toledo Freitas, Eng Henrique Awoyama Klein e Dr. Marcelo Negreiros que sempre contribuíram com sua experiência e compartilharam informações nas áreas em que eu

não tinha domínio. Em especial ao Dr. André Borin Soares que sempre acompanhou e apoiou minha evolução e dedicou muito de seu tempo em meu auxílio, e assim, foi o principal colaborador no desenvolvimento do meu trabalho. Agradeço muito a todos, pois sem esse grupo forte e dedicado, seria muito mais difícil chegar aos resultados alcançados.

A minha coorientadora Dra. Leticia Vieira Guimarães, pelo apoio em todas as etapas do desenvolvimento do trabalho e por estar sempre presente nos momentos mais importantes. Também pela dedicação em me fornecer suporte, me auxiliar e orientar principalmente em momentos decisivos.

E ao meu orientador Dr. Altamiro Amadeu Susin que se mostrou um grande parceiro, estando sempre ao meu lado para encarar todas as dificuldades. Por se preocupar com meu trabalho, demonstrando-se sempre interessado e participativo em todas as etapas. E principalmente por acreditar em mim e me estimular mesmo nos momentos em que até mesmo eu tive dúvidas sobre o meu sucesso em superar este desafio.

Ao Programa de Pós-Graduação em Engenharia Elétrica, PPGEE, pela oportunidade de realização de trabalhos em minha área de pesquisa e pelo apoio nas situações de dificuldade.

Enfim, agradeço a todos que de alguma forma contribuíram, apoiaram ou acompanharam minha caminhada.

A todos, meu muito obrigado!

RESUMO

Uma sequência de vídeo pode ser adquirida de forma progressiva ou entrelaçada. No padrão de codificação de vídeo H.264/AVC os campos de uma imagem entrelaçada podem ser codificados em modo *frame* (campos *top* e *bottom* entrelaçados) ou em modo *field* (campos *top* e *bottom* agrupados separadamente). Quando a escolha é adaptativa para cada par de macro blocos a codificação é chamada de *Macroblock Adaptive Frame-Field* (MBAFF).

Inovações na predição inter-quadro do H.264/AVC contribuíram significativamente para a performance do padrão alcançar o dobro da taxa de compressão do seu antecessor (ITU, 1994), ao custo de um grande aumento de complexidade computacional do CODEC. Dentro da predição inter-quadro, o bloco de compensação de movimento (MC) é responsável pela reconstrução de um bloco de pixels. No decodificador apresentado em (BONATTO, 2012) está integrada uma solução em hardware para o MC que suporta a maior parte do conjunto de ferramentas do perfil *Main* do H.264/AVC. A compensação de movimento pode ser dividida em predição de vetores e processamento de amostras. No processamento de amostras é realizada a interpolação e a ponderação de amostras. O módulo de ponderação de amostras, ou predição ponderada, utiliza fatores de escala para escalonar as amostras na saída do MC. Isso é muito útil quando há esvanecimento no vídeo.

Inicialmente este trabalho apresenta um estudo do processo de compensação de movimento, segundo o padrão de codificação de vídeo H.264/AVC. São abordadas todas as ferramentas da predição inter-quadro, incluindo o tratamento de vídeo entrelaçado e todos os possíveis modos de codificação para o mesmo. A seguir é apresentada uma arquitetura em hardware para a predição ponderada do MC. Esta arquitetura atende o perfil *main* do H.264/AVC, que prevê a decodificação de imagens *frame*, *field* ou MBAFF. A arquitetura apresentada é baseada no compensador de movimento contido no decodificador apresentado em (BONATTO, 2012), que não tem suporte a predição ponderada e a vídeo entrelaçado. A arquitetura proposta é composta por dois módulos: *Scale Factor Prediction* (SFP) e *Weighted Samples Prediction* (WSP) .

A arquitetura foi desenvolvida em linguagem VHDL e a simulação temporal mostrou que a mesma pode decodificar imagens MBAFF em tempo real @60i. Dessa forma, tornando-se uma ferramenta muito útil ao desenvolvimento de sistemas de codificação e decodificação em HW. Não foi encontrada, na literatura atual, uma solução em hardware para compensação de movimento do padrão H.264/AVC com suporte a codificação MBAFF.

Palavras-chave: H264/AVC, MBAFF, Predição Inter-Quadro, Compensação de Movimento, Processamento de Amostras, Predição Ponderada, VHDL e Hardware.

ABSTRACT

A video sequence can be acquired in a progressive or interlaced mode. In the video coding H.264/AVC standard an interlaced picture can be encoded in frame mode (top and bottom fields interlaced) or field mode (top and bottom fields combined separately). When the choice for each pair of macro-blocks coding is adaptive, it is called Macroblock Adaptive Frame-Field (MBAFF).

The innovations in the inter-frame prediction of H.264/AVC contributed significantly to the performance of the standard that achieved twice the compression ratio of its predecessor (ITU, 1994), at the cost of a large increase in computational complexity of the CODEC. In the inter-frame prediction, the motion compensation (MC) module is responsible for the reconstruction of a pixel's block. In the decoder shown in (BONATTO 2012) an integrated hardware solution to the MC is included which can decode most of the H.264/AVC main profile tools. The motion compensation can be divided into motion vectors prediction and sample processing. In the sample processing part, samples interpolation and weighting are performed. The weighted samples prediction module uses scale factors to weight the samples for generating the output pixels. This is useful in video fading.

Initially, this work presents a study of the motion compensation process, according to the H.264/AVC standard. It covers all of inter-frame prediction tools, including all possible coding modes for interlaced video. A hardware architecture for the weighted samples prediction of MC is shown next. It is in compliance with the main profile of H.264/AVC standard, therefore it can decode frame, field and MBAFF pictures. The architecture presented is based on the motion compensator used in the (BONATTO, 2012) decoder, which does not support the weighted prediction and interlaced video. The purposed architecture is composed by two modules: Scale Factor Prediction (SFP) and Weighted Samples Prediction (WSP).

The hardware implementation was described using VHDL and the timing simulation has shown that it can decode MBAFF pictures in real time @60i. Therefore, this is an useful tool for hardware CODEC development. Similar hardware solution for H.264/AVC weighted prediction that supports MBAFF coding was not found in previous works.

Keywords: H264/AVC, MBAFF, Inter-frame Prediction, Motion Compensation, Samples Processing, Weighted Prediction, VHDL and Hardware.

SUMÁRIO

1 INTRODUÇÃO.....	16
2 H.264/AVC.....	20
2.1 INTRODUÇÃO AO H.264/AVC.....	20
2.2 PERFIS H.264/AVC.....	24
2.3 CODIFICADOR-DECODIFICADOR.....	26
2.4 COMPOSIÇÃO DE VÍDEO E CODIFICAÇÃO MBAFF.....	29
2.5 PREDIÇÃO INTER-QUADRO.....	32
2.5.1 Predição de Vetores de Movimento.....	37
2.5.2 Memória de Quadros de Referência.....	40
2.5.3 Processamento de Amostras.....	41
2.5.3.1 Interpolação de Amostras.....	42
2.5.3.2 Predição L0, Predição L1 e Bi-predição.....	45
2.5.3.3 Predição Ponderada.....	46
2.6 CONSIDERAÇÕES DO H.264.....	48
3 DECODIFICAÇÃO INTER-QUADRO.....	49
3.1 PREDITOR DE VETORES DE MOVIMENTO.....	51
3.1.1 Predição Padrão.....	52
3.1.2 Predição Direta.....	55
3.1.2.1 Predição Direta Espacial.....	55
3.1.2.2 Predição Direta Temporal.....	56
3.1.2.3 Identificação do Bloco Co-localizado	59
3.2 ACESSO À MEMÓRIA.....	61
3.3 INTERPOLAÇÃO DE AMOSTRAS.....	62
3.4 PREDIÇÃO PONDERADA.....	64
3.5 ARQUITETURAS MC.....	66
3.5.1 Arquitetura de Referência.....	68
4 ARQUITETURA PROPOSTA.....	72
4.1 SCALE FACTOR PREDICTION.....	74
4.1.1 Modo Explicit.....	74
4.1.2 Modo Implicit.....	77
4.2 WEIGHTED SAMPLES PREDICTION.....	78
4.3 RESULTADOS.....	79
5 CONCLUSÃO.....	83
REFERÊNCIAS	86
APÊNDICE A: Artigo.....	89
APÊNDICE B: Estudo Da Predição Inter-quadro.....	90

LISTA DE ILUSTRAÇÕES

Figura 1 Redundâncias Entre Amostras (RICHARDSON, 2003).	21
Figura 2 Sub-amostragem De Crominância (Sahafi, 2005).	22
Figura 3 MB Em Um Quadro De Um Vídeo H.264 Com Sub-amostragem 4:2:0. . .	23
Figura 4 Slices De Uma Imagem Em Um Vídeo H.264/AVC.	23
Figura 5 Perfis H.264/AVC (ZATT, 2008).....	25
Figura 6 Codificador H.264/AVC.....	26
Figura 7 Decodificador H.264/AVC.....	26
Figura 8 Composição Do Quadro E Modos De Codificação.	30
Figura 9 Exemplo De Quadro E Modos De Codificação.	30
Figura 10 Modos De Codificação Das Imagens No PAFF	31
Figura 11 Exemplo De Codificação MBAFF.	32
Figura 12 Ordem De Codificação.	32
Figura 13 Região De Referência.....	34
Figura 14 Particionamento De MB, Resolução De Luminância (ZATT, 2008).	35
Figura 15 Parte Fracionária De Vetores Para Sub-amostragem 4:2:0.	36
Figura 16 Região De Referência Para Extrapolação De Borda.....	37
Figura 17 Geração Do Motion Vector Difference (MVD).....	38
Figura 18 Vizinhança Do Macrobloco.....	39
Figura 19 Bloco Co-localizado.....	39
Figura 20 Referências Para Bi-predição (RICHARDSON, 2003).	41
Figura 21 Amostras De Luminância Half-pel (RICHARDSON, 2003).	42
Figura 22 Interpolação De Amostras De Luminância Half-pel (ITU, 2005).	44
Figura 23 Interpolação De Luminância Quarter-pel (RICHARDSON, 2003).....	45
Figura 24 Interpolação De Amostras De Crominância (RICHARDSON, 2003).....	45
Figura 25 Referências Para Bi-predição.....	46
Figura 26 Exemplo De Esvanecimento Em Vídeo.	47
Figura 27 Estrutura Típica De MC Para O Padrão H.264/AVC.	49
Figura 28 Vizinhança Para Macroblocos Particionados (RICHARDSON, 2003). . .	52
Figura 29 Vizinhança Utilizada No MVP (ZATT, 2006).	53
Figura 30 Vetores Para Predição Direta Temporal (ITU, 2005).....	59
Figura 31 Região De Referência Para Bloco 2x2 De Crominância.	63
Figura 32 Região De Referência Para Bloco 4x4 De Luminância.....	64
Figura 33 Estrutura Dos Arquivos Da Arquitetura MC E Test Bench.....	69
Figura 34 Arquitetura Do Compensador De Movimento (BONATTO, 2012).....	70
Figura 35 Arquitetura De Predição Ponderada Integrada Ao MC.....	73
Figura 36 Estrutura Dos Arquivos Da Arquitetura Proposta Integrada Ao MC.....	73
Figura 37 Modo Explicit Do Scale Factor Prediction (SFP).....	76
Figura 38 Modo Implicit Do Scale Factor Prediction (SFP).....	78
Figura 39 Módulo Weighted Samples Prediction (WSP).....	79

Figura 40	Simulação Do Módulo Scale Factor Prediction (SFP), Modo Explicit...	81
Figura 41	Simulação Do Módulo Scale Factor Prediction (SFP), Modo Implicit...	81
Figura 42	Simulação Do Módulo Weighted Samples Prediction (WSP).....	82

LISTA DE TABELAS

Tabela 1 Tipos de macrobloco para predição inter-quadro em slices P.....	50
Tabela 2 Tipos de macrobloco para predição inter-quadro em slices B.....	51
Tabela 3 Vizinhanças para MB tipo frame em imagem MBAFF.....	53
Tabela 4 Vizinhanças para MB tipo field em imagem MBAFF.....	54
Tabela 5 Modo padrão de predição do PMV.....	55
Tabela 6 MV para predição direta espacial.....	56
Tabela 7 Índice de referência RefIdxL0 para predição direta temporal.....	58
Tabela 8 Posição vertical yM para bloco co-localizado.....	60
Tabela 9 Localização da imagem de referência.....	62
Tabela 10 Identificação do modo de ponderação de amostras.....	65
Tabela 11 Fatores de escala para modo implícito do WP.....	66

LISTA DE ABREVIATURAS

AVC: *Advanced Video Coding*

CABAC: *Context-Based Adaptive Binary Arithmetic Coding*

CAVLC: *Context-Based Adaptive Variable Length Coding*

CODEC: Codificador-decodificador

DCT: *Discrete Cosine Transform*

FIFO: *First In First Out*

FIR: *Finite Impulse Response*

FPGA: *Field Programmable Gate Array*

FRExt: *Fidelity Range Extensions*

HCT: *High Correlation Transform*

HDTV: *High Definition Digital Television*

HP422-MoCHA: *High Profile 4:2:2 Motion Compensation Hardware*

Architecture

IEC: *International Engineering Consortium*

ISO: *International Standards Organization*

ITU: *International Telecommunication Union*

ITU-T: *ITU Telecommunication Standardization Sector*

JVT: *Joint Video Team*

L0: Lista 0

L1: Lista 1

LaPSI: Laboratório de Processamento de Sinais e Imagens

MB: *Macroblock*

mb_type: *Macroblock Type*

MBAFF: *Macroblock Adaptive Frame-Field*

MC: *Motion Compensation*

ME: *Motion Estimation*

MoCHA: *Motion Compensation Hardware Architecture*

MPEG: *Moving Picture Experts Group*

MV: *Motion Vector*

mvCol: *Motion Vector Co-located*

MVD: *Motion Vector Difference*

mvL0: *Motion Vector List0*

mvL1: *Motion Vector List1*

MVP: *Motion Vector Predictor*

PAFF: *Picture Adaptive Frame-Field*

PMV: *Predictive Motion Vector*

PPGEE: Programa de Pós-Graduação em Engenharia Elétrica

POC: *Picture Order Counter*

Q: Quantização Direta

QI: Quantização Inversa

QP: *Quantization Parameter*

RefIdx: *Reference Index*

RefIdxL0: *Reference Index List 0*

RefIdxL1: *Reference Index List 1*

RefPicCol: *Reference Picture Co-located*

RGB: *Red, Green, Blue*

SBTVD: Sistema Brasileiro de Televisão Digital

T: Transformada Direta

TI: Transformada Inversa

UFRGS: Universidade Federal do Rio Grande do Sul

VCEG: *Video Coding Experts Group*

VHDL: *VHSIC Hardware Description Language*

VHSIC: *Very High Speed Integrated Circuit*

YCbCr: *Luminance, Chrominance Blue, Chrominance Red*

WP: *Weighted Prediction*

1 INTRODUÇÃO

A busca constante para alcançar maiores resoluções de imagem resulta em um grande esforço de pesquisadores em aprimorar as técnicas de compressão de vídeos digitais. A compactação de vídeos pode ser empregada em inúmeras aplicações, por exemplo, com os recursos atuais, o armazenamento e a transmissão de vídeos em alta definição (HDTV) exigem alta capacidade de compressão para se tornarem viáveis.

O *Joint Video Team*, JVT (ITU, 2003), formado a partir da união de especialistas do ITU-T *Video Coding Experts Group* (VCEG) e ISO/IEC *Moving Picture Experts Group* (MPEG), alcançou seu principal objetivo ao desenvolver o padrão de codificação de vídeo H.264/AVC. Este padrão superou as taxas de compressão de todos padrões existentes até então, sendo capaz de dobrar a taxa de compressão do seu antecessor, o MPEG-2 (ITU, 2005).

Para alcançar esta alta capacidade de compressão foi necessário um grande aumento na complexidade computacional de um CODEC dedicado a este padrão. Este aumento de complexidade torna inviável, pelo menos na tecnologia atual, a utilização de um CODEC H.264/AVC completo implementado em software para aplicações comerciais. Um exemplo claro da necessidade do desenvolvimento de codificadores e decodificadores H.264/AVC em hardware está no crescente mercado de sistemas embarcados capazes de manipular vídeos (filmadoras digitais, tablets, smartphones, etc). Neste caso, questões como a quantidade de recursos de hardware utilizados e o consumo de energia são críticos. Portanto, para estas aplicações, a única alternativa é o desenvolvimento destes sistemas em hardware dedicado. Para o caso de SET-TOP BOX de TV digital, a necessidade da

modelagem em hardware fica evidente quando se almeja a decodificação em tempo real de vídeos com altas resoluções.

Somado a estes fatores, a adoção do H.264/AVC pelo Sistema Brasileiro de Televisão Digital (SBTVD) conseqüentemente aumentou o interesse da comunidade científica neste padrão. Em paralelo ao lançamento do SBTVD o governo brasileiro criou a Rede H.264, projeto que tem como objetivo desenvolver produtos para tratar da codificação dos sinais fonte para o SBTVD (ABNT, 2007). A UFRGS participa deste projeto desenvolvendo, em hardware, o codificador e o decodificador H.264/AVC, sendo que o decodificador é desenvolvido no Laboratório de Processamento de Sinais e Imagens (LaPSI). Atualmente o decodificador desenvolvido na UFRGS é capaz de tratar seqüências de vídeo do perfil *baseline* do H.264/AVC (BONATTO, 2012). Este perfil define um conjunto de ferramentas de codificação que não exigem um CODEC complexo, e portanto, é utilizado no SBTVD para aplicativos móveis. Para aplicações em que são necessárias altas taxas de compressão é utilizado o perfil *main*, que disponibiliza ferramentas mais complexas. Para a expansão ao perfil *main* do decodificador (BONATTO, 2012) deve agregar o suporte à: ponderação de amostras, codificação aritmética adaptativa ao contexto (CABAC) e à vídeos entrelaçados.

Um vídeo é denominado progressivo ou entrelaçado de acordo com sua aquisição. Em um vídeo progressivo os campos *top* e *bottom*, que são as linhas pares e ímpares de um quadro, são adquiridos no mesmo instante de tempo. Já nas sequências de vídeo entrelaçado os campos *top* e *bottom* são adquiridos em instantes de tempo distintos. No H.264/AVC, este conjunto de dados pode ser codificado em modo *frame* (campos entrelaçados – mesmo formato do vídeo na aquisição) ou modo *field* (campos agrupados separadamente). A escolha de cada modo de forma adaptativa para cada imagem da seqüência de vídeo é

chamada de *Picture Adaptive Frame-Field* (PAFF). Os quadros nos quais a escolha de modo *frame* ou *field* é adaptativa para cada par de macroblocos (um macrobloco corresponde à 16x16 pixels de luminância) são denominados como *Macroblock Adaptive Frame-Field* (MBAFF).

Este trabalho inicialmente apresenta um estudo do processo de decodificação inter-quadro, ou compensação de movimento, segundo o padrão de codificação de vídeo H.264/AVC. São descritas as ferramentas de codificação aplicáveis à predição inter-quadro e todos os processos envolvidos na decodificação utilizando este modo de predição. A descrição apresenta ênfase nas implicações da utilização de cada um dos modos de codificação disponíveis para vídeos entrelaçados.

Inserido no projeto Rede H.264, mais especificamente dedicado à integração com o decodificador desenvolvido no LaPSI (BONATTO, 2012), este trabalho apresenta a seguir uma solução em hardware para a ponderação de amostras do MC com suporte a vídeo entrelaçado codificado em MBAFF. A arquitetura de compensação de movimento do decodificador desenvolvido no LaPSI é utilizada como base para o desenvolvimento da arquitetura proposta. Este MC suporta a maior parte do conjunto de ferramentas do perfil *Main* do H.264/AVC, mas não tem suporte à ponderação de amostras e à decodificação de vídeos entrelaçados.

No capítulo 2 é apresentado o padrão de codificação de vídeo H.264/AVC, e de forma mais detalhada, são apresentadas a codificação de vídeo entrelaçado MBAFF e a predição inter-quadro. O capítulo 3 faz uma descrição do processo de decodificação realizado pela predição inter-quadro e apresenta o estado da arte das arquiteturas de compensação de movimento, com ênfase na arquitetura usada como base para o desenvolvimento deste trabalho.

A modelagem em hardware da ponderação de amostras e os resultados de simulação estão descritos e ilustrados no capítulo 4. O capítulo 5 expõe conclusões sobre o trabalho desenvolvido, incluindo discussões sobre ferramentas de codificação utilizadas no H.264/AVC, características das atuais arquiteturas de compensação de movimento e os aspectos relevantes da arquitetura desenvolvida.

2 H.264/AVC

O padrão de codificação de vídeo H.264/AVC (H.264/*Advanced Video Coding*) (ITU, 2003), foi desenvolvido de forma conjunta pelo grupo de especialistas em codificação de vídeo da *International Telecommunication Union (ITU-T)*, *Video Coding Experts Group (VCEG)*, e pelo grupo de especialistas em imagem e movimento da ISO/IEC, *Moving Picture Experts Group (MPEG)*. A cooperação destes dois grupos originou o JVT (*Joint Video Team*), que apresentou a definição do padrão em maio de 2003. A versão adotada pelo Sistema Brasileiro de Televisão Digital (ABNT, 2008) é regulamentada pela (ITU, 2005).

Neste capítulo são expostos inicialmente os aspectos mais relevantes do H.264/AVC, incluindo os perfis, a estrutura básica do codificador, a estrutura básica do decodificador, o entrelaçamento de vídeo e a codificação MBAFF. Mais detalhadamente é apresentado o processamento de amostras, que integra a compensação de movimento na predição inter-quadro.

2.1 Introdução ao H.264/AVC

Ao analisar uma sequência de vídeo natural é possível perceber que há uma grande quantidade de informação redundante. Nos quadros do vídeo, é comum haver áreas de amostras (representações de pixels) com o mesmo valor ou com pequenas diferenças de magnitude, ou seja, há regiões com alta correlação entre as amostras, como exemplificado na Figura 1. A redundância espacial é tratada no H.264/AVC pela predição intra-quadro. Ao invés de preservar o valor de cada pixel, são mantidas informações que permitem

reconstruir parte de um quadro através dos valores das amostras de outra região do mesmo quadro.

Usualmente uma sequência de vídeo também possui regiões de pixels que estão presentes em diferentes quadros, portanto também há alta correlação entre os quadros do vídeo, Figura 1. Estas regiões não estão sempre na mesma posição espacial nos diferentes quadros. Por exemplo, um objeto em movimento em um vídeo pode ter a mesma representação em vários quadros temporalmente subsequentes, porém em cada quadro irá ocupar uma posição diferente. No H.264/AVC essa redundância temporal é tratada pela predição inter-quadro. Neste caso são preservadas informações que permitem reconstruir parte de um quadro através dos valores das amostras de uma região de outro quadro do vídeo.



Figura 1 Redundâncias Entre Amostras (RICHARDSON, 2003).

O sistema de visão humano é mais sensível a luminância do que a cor, por este motivo, é interessante representar estas informações de forma separada, o que não acontece no espaço de cores *Red Green Blue* (RGB). O H.264/AVC utiliza o espaço de cores YCbCr (MIANO, 1999), onde Y representa a luminância e Cb e Cr representam respectivamente as crominâncias *blue* e *red*. Desta forma o vídeo pode ser sub-amostrado visando preservar as informações mais sensíveis ao sistema de visão humano, portanto fica evidente que a sub-amostragem é outra ferramenta eficiente na redução de dados de uma sequência de vídeo. O H.264/AVC prevê a sub-amostragem de crominância 4:2:2, que define que as amostras de crominância Cb e Cr têm metade da resolução horizontal das amostras de luminância Y, e 4:2:0, onde as amostras de crominância têm a metade da resolução vertical e horizontal das amostras de luminância. A Figura 2 exemplifica estas duas sub-amostragens para um bloco de 4x4 amostras de luminância e respectivas amostras de crominância. A norma brasileira (ABNT, 2008) estabelece a utilização no SBTVD de sub-amostragem 4:2:0, portanto todo o conteúdo apresentado neste trabalho leva em consideração este tipo de sub-amostragem.

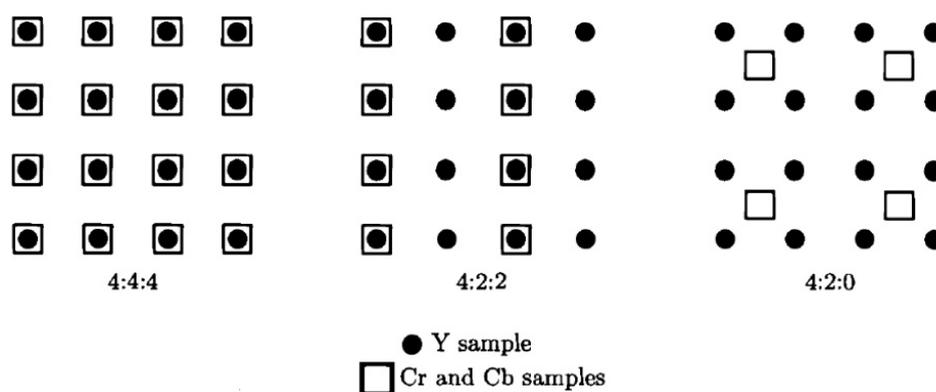


Figura 2 Sub-amostragem De Crominância (Sahafi, 2005).

Os quadros no H.264/AVC são codificados em macroblocos (MB) de 16x16 pixels de luminância (Y) e os pixels de crominância azul (Cb) e crominância vermelha (Cr)

associados. Para a sub-amostragem 4:2:0, um MB têm 16x16 Y, 8x8 Cb e 8x8 Cr (pixels), conforme Figura 3.

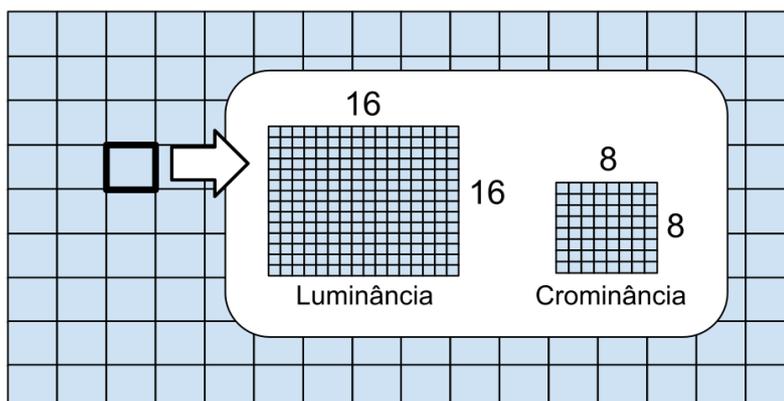


Figura 3 MB Em Um Quadro De Um Vídeo H.264 Com Sub-amostragem 4:2:0.

Macroblocos subsequentes, em relação à ordem de codificação de uma imagem, são agrupados em *slices*. Estes grupos são organizados em fatias, Figura 4, que podem abranger de apenas um a até todos macroblocos da imagem. Cada slice pode ser classificado em cinco diferentes tipos: I, P, B, SI e SP. Os slices SI e SP são definidos apenas no perfil *Extended* (os perfis do H.264/AVC serão abordados no item 2.2) que não foi adotado pelo SBTVD e, portanto, não é abordado neste trabalho.

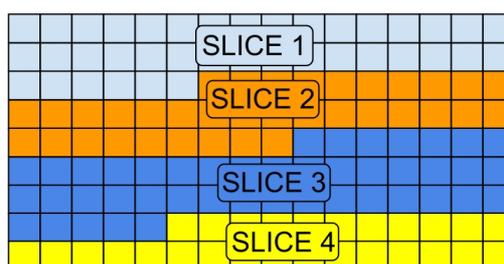


Figura 4 Slices De Uma Imagem Em Um Vídeo H.264/AVC.

Em slices do tipo I somente há macroblocos do tipo I. Slices do tipo P podem conter macroblocos tipo P e tipo I. Em slices B pode-se encontrar macroblocos I e B.

Macroblocos do tipo I utilizam codificação intra-quadro, que explora a redundância espacial dentro do slice, ou seja, utiliza amostras contidas no slice atual para inferir as amostras do MB corrente. Neste caso os macroblocos também podem ser codificados no modo I_PCM, onde os valores das amostras são enviados diretamente ao *bitstream*.

Os macroblocos tipo P sofrem o processo de predição inter-quadro, que usa amostras contidas em imagens previamente codificadas chamadas de imagens de referência. Desta forma, é explorada a redundância temporal que existe entre as imagens da sequência de vídeo. As imagens utilizadas como referência para estes macroblocos são organizadas em uma lista chamada de lista 0 (ITU, 2005).

Macroblocos B também utilizam a predição inter-quadro, no entanto, cada MB ou partição pode dispor de informações de até duas imagens de referência. Para isso, além da lista 0 (L0), há imagens de referência organizadas em outra lista, chamada de lista 1 (L1) (ITU, 2005). As estruturas das listas L0 e L1 são apresentadas no item 2.5.2 .

2.2 Perfis H.264/AVC

O padrão H.264/AVC é dividido em perfis que definem a utilização de grupos particulares de funções de codificação. Desta forma são especificadas quais ferramentas cada decodificador deve suportar. Em sua primeira versão, de maio de 2003 (ITU, 2003), foram definidos três perfis: *Baseline*, *Main* e *Extended*. O perfil *Baseline* é direcionado a dispositivos menos complexos, como comunicações em dispositivos móveis. Neste perfil há suporte a slices I e P (portanto tanto a predição intra-quadro quanto inter-quadro são permitidas) e codificação de comprimento variável adaptativa ao contexto (CAVLC). Há um grande acréscimo de complexidade para um CODEC com suporte ao perfil *Main* em

relação aos dedicados somente ao perfil *Baseline*. Aplicações comuns do perfil *Main* são: transmissão de televisão e armazenamento de vídeo. Neste perfil há suporte a vídeos entrelaçados, slices do tipo B, predição ponderada e codificação de entropia utilizando codificação aritmética adaptativa ao contexto (CABAC). O perfil *Extended* agrega funções ao perfil *Baseline* e é direcionado a aplicações de streaming de vídeo (PURI, 2004).

Aplicações profissionais que exigem vídeos de alta qualidade não foram contempladas nestes três perfis. Para atender estas necessidades, uma continuação do projeto JVT foi realizada para adicionar novas extensões ao padrão original. Estas extensões, chamadas de extensões para alcance de fidelidade (fidelity range extensions – FRExt), produziram quatro novos perfis chamados coletivamente de perfis High. Os conjuntos de ferramentas de codificação que cada perfil suportam são apresentados na Figura 5.

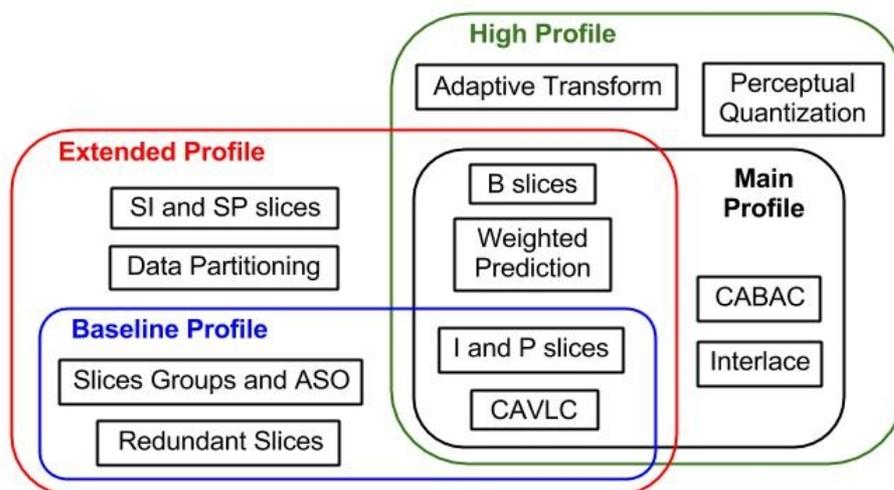


Figura 5 Perfis H.264/AVC (ZATT, 2008).

Os perfis *Extended* e *High* não foram adotados pelo SBTVD e, portanto, não são abordados neste trabalho.

2.3 Codificador-Decodificador

As Figuras 6 e 7 apresentam o diagrama de blocos simplificado de um codificador e um decodificador H.264/AVC respectivamente. Como ilustrado, tanto no codificador quanto no decodificador existe um chaveamento entre os dois tipos de predição de amostras, intra-quadro e inter-quadro. A predição intra-quadro, ou somente intra, foi uma das inovações do padrão, pois mesmo nos macroblocos do tipo I é realizada uma predição antes da aplicação da transformada (ZATT, 2008).

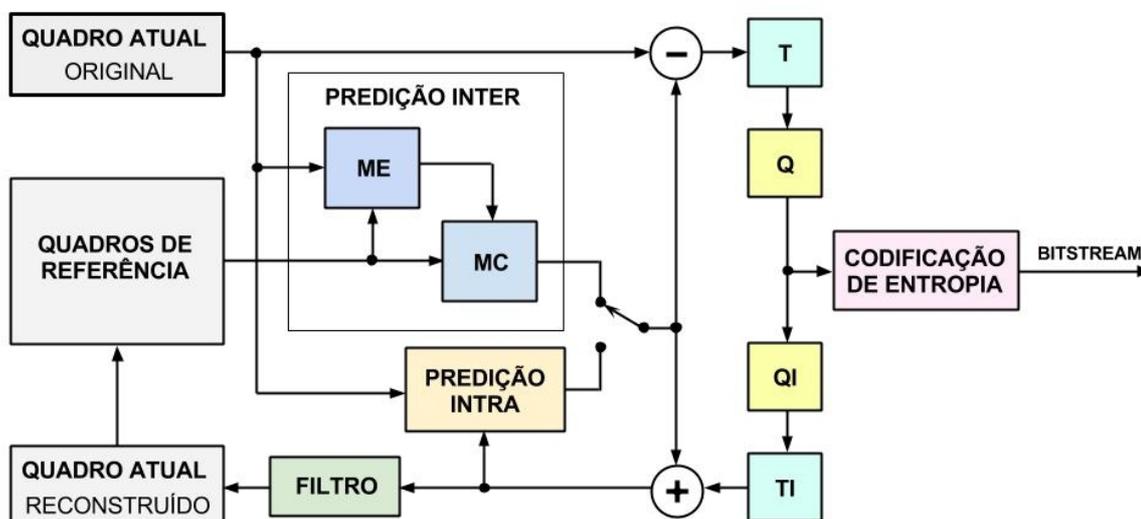


Figura 6 Codificador H.264/AVC.

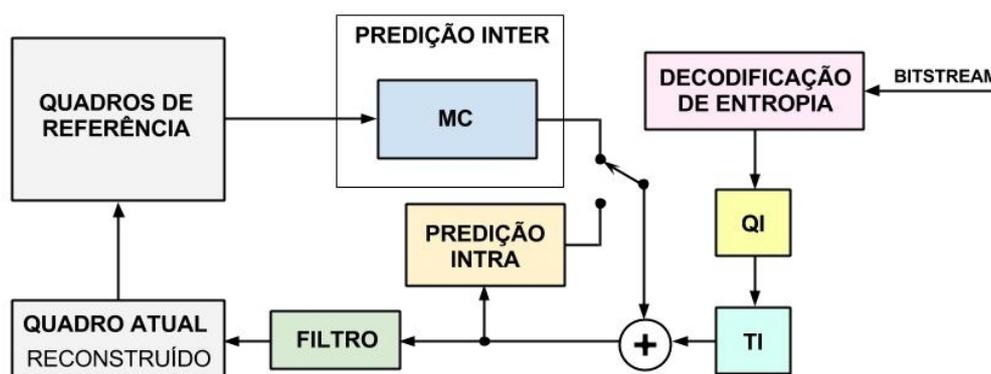


Figura 7 Decodificador H.264/AVC.

A predição inter-quadro pode ser dividida em dois módulos principais, *Motion Estimation* (ME) e *Motion Compensation* (MC). No codificador, o módulo de estimação de movimento (ME na figura 6) calcula vetores e índices que identificam a região, dentro de uma das imagens de referência, que mais se assemelha as amostras do MB atual. Com estas informações, o módulo de compensação de movimento (MC - Figuras 6 e 7) utiliza as amostras da região indicada para inferir as amostras do macrobloco atual.

As saídas das predições inter-quadro e intra-quadro são estimativas das amostras de um MB. No codificador, estas amostras estimadas são subtraídas das amostras na região correspondente da imagem original. A diferença é chamada de resíduo e é enviada à transformada direta (bloco T da Figura 6). A transformada utilizada no padrão H.264/AVC é uma aproximação inteira da DCT, e é conhecida na literatura como *high correlation transform* (HCT) (PURI, 2004). Esta aproximação impede um aumento excessivo de complexidade computacional (do CODEC H.264/AVC), que seria necessário para evitar erros de casamento entre o codificador e o decodificador com operações em ponto flutuante (MALVAR, 2003).

Os módulos de quantização (Q na Figura 6) e quantização inversa (QI nas Figuras 6 e 7) utilizam quantização escalar (RICHARDSON, 2003). Os cálculos realizados no módulo Q são diferentes para cada modo de predição e para cada tipo de amostra, croma ou luminância. Genericamente, as operações realizadas são: multiplicação por uma constante, soma do resultado com outra constante e por último um deslocamento controlado por uma terceira constante. Estas constantes são diretamente influenciadas por um parâmetro de entrada chamado QP, que é usado no codificador para controlar a taxa de compressão, e conseqüentemente a qualidade do vídeo.

Nos módulos de codificação e decodificação de entropia (Figuras 6 e 7), são utilizados 3 diferentes métodos: codificação de comprimento variável adaptativa ao contexto (CAVLC), *Exponential-Golomb* e codificação binária aritmética adaptativa ao contexto (CABAC). O *Exponential-Golomb*, ou somente *Exp-Golomb* (RICHARDSON, 2003), é usado para a compressão de cabeçalhos. Para os coeficientes quantizados, saída do bloco de quantização, pode-se definir para cada imagem o uso de CAVLC ou CABAC. A saída do módulo de codificação de entropia é assumida como sendo a saída do codificador, *bitstream* H.264/AVC. Portanto a decodificação de entropia é a primeira etapa a ser realizada no decodificador, conforme ilustrado na Figura 7. Todo processo de decodificação, após esta etapa, é realizado também pelo codificador, pois as predições inter-quadro e intra-quadro utilizam as amostras decodificadas do MB atual para codificar os macroblocos e imagens subsequentes. Desta forma, tanto no codificador como no decodificador, são aplicadas: a quantização inversa, a transformada inversa e a soma dos resíduos decodificados (resultado do bloco TI) com o resultado da predição inter ou intra. O resultado desta soma serve de realimentação à predição intra. Para a predição inter são utilizadas como referência as imagens de saída da decodificação, após a aplicação do filtro.

O filtro do H.264/AVC, conhecido como *deblocking filter*, por reduzir efeitos de bloco, atua nas bordas do macrobloco e das sub-partições 4x4. Suas operações são adaptativas conforme vários fatores: parâmetros de quantização do MB atual e dos vizinhos, tipo de codificação do MB atual, valores dos pixels do MB atual e dos vizinhos, e a magnitude dos vetores de movimento (MV). O resultado da filtragem é uma imagem reconstruída que é assumida como a saída do decodificador.

Cada imagem reconstruída pelo decodificador pode ser selecionada para compor o conteúdo da memória de quadros de referência, que é utilizada pela predição inter-quadro,

Figuras 6 e 7. Para isso cada imagem é rotulada como “*used as reference*” ou “*not used as reference*” conforme definido na codificação. As imagens da memória de quadros de referência são organizadas em duas listas, lista 0 (L0) e lista 1 (L1), e recebem denominações que identificam quais imagens devem permanecer na memória e quais as próximas a serem substituídas. O gerenciamento das listas será abordado no item 2.5.2.

Como o processo de decodificação deve ocorrer também dentro do codificador, aparentemente a codificação H.264/AVC é sempre mais complexa que a decodificação. No entanto, como somente é normatizado o conjunto de ferramentas que o decodificador H.264/AVC deve suportar (ITU, 2005), isto não é verdade. O codificador pode utilizar somente parte das ferramentas de codificação definidas para o padrão, e dessa forma reduzir consideravelmente sua complexidade, enquanto que na decodificação isso não é possível.

2.4 Composição de Vídeo e Codificação MBAFF

A (ITU, 2005) define que em uma sequência de vídeo cada quadro é formado pela composição de dois campos, linhas ímpares e linhas pares. Quando a aquisição de uma sequência de vídeo é progressiva, os dois campos do quadro são adquiridos no mesmo instante de tempo, ver quadro progressivo na Figura 8. No entanto, para a aquisição de vídeo entrelaçado, o quadro é composto por campos adquiridos em instantes de tempo distintos, conforme também pode ser visto na Figura 8.

No H.264/AVC os campos de um quadro podem ser codificados de duas formas diferentes, modo *field* e modo *frame*, conforme Figura 8 e exemplo da Figura 9. No modo *field*, são agrupadas separadamente as linhas pares e linhas ímpares em campos *top* e *bottom*, Figura 8. Para o modo *frame*, as linhas ímpares e pares permanecem entrelaçadas

conforme o formato na aquisição, Figura 8. Para vídeos progressivos somente é permitido o uso de codificação em modo *frame*, porém, os vídeos entrelaçados podem usar ambos os tipos.

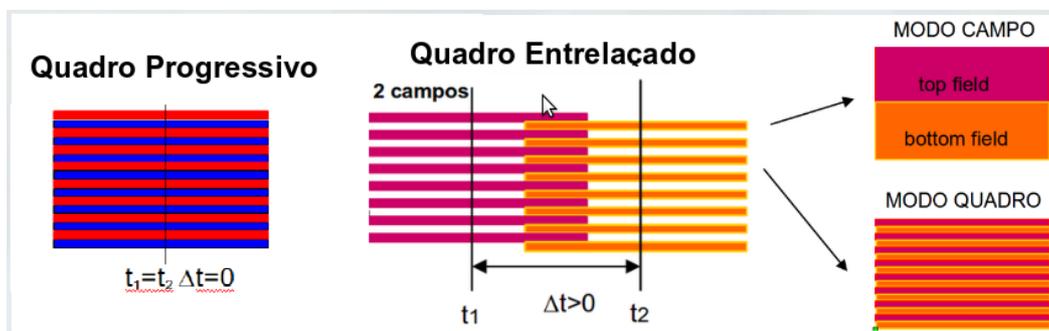


Figura 8 Composição Do Quadro E Modos De Codificação.

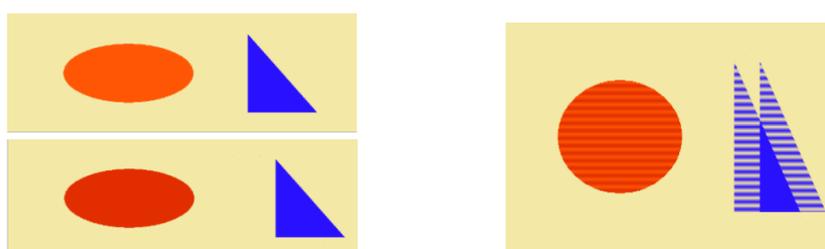


Figura 9 Exemplo De Quadro E Modos De Codificação.

Quando a escolha do tipo de composição para um quadro entrelaçado (modo campo ou modo quadro) é realizada para cada imagem da sequência, a codificação é chamada *Picture Adaptive Frame-Field* (PAFF), (quadros entrelaçados da Figura 9). No H.264/AVC um *frame* pode ainda ser codificado utilizando *Macroblock Adaptive Field Frame* (MBAFF), onde a escolha do modo de codificação dos campos é adaptativa para cada par de macroblocos, ver Figura 11. Portanto, neste tipo de codificação, cada par de macroblocos vizinhos verticais pode ser uma composição de campos entrelaçados (modo *frame*) ou simplesmente dois campos separados (modo *field*). Neste segundo caso o macrobloco

superior é o *top field* e o MB inferior é o *bottom field*. O SBTVD restringe a codificação dos quadros de vídeos entrelaçados somente em modo *field* e MBAFF.

Nos quadros de um vídeo (ou regiões dos quadros) em que não há movimento significativo na cena, a escolha da composição em modo *frame* é interessante pois os objetos da cena não sofrem deformação devido a aquisição entrelaçada. Porém quando há movimentos rápidos de objetos na cena, a aquisição de vídeo entrelaçado reduz a correlação entre os campos, como pode ser visto no triângulo das Figuras 10 (a) e 10 (b). Portanto, as regiões ou quadros onde há movimento são mais eficientemente comprimidas se os campos forem codificados separadamente, modo *field*. Isso justifica o uso de PAFF e principalmente o MBAFF, pois o modo *field* pode ser aplicado somente nestas regiões, tornando-as mais homogêneas (ver região do triângulo na Figura 11). O simples fato do uso da codificação MBAFF pode gerar um ganho de mais de 15% na capacidade de compressão do padrão H.264/AVC (PURI, 2004).



(a) Imagem em modo *field*

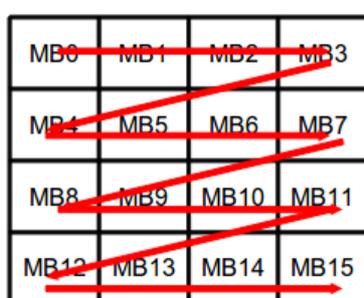
(b) Imagem em modo *frame*

Figura 10 Modos De Codificação Das Imagens No PAFF .

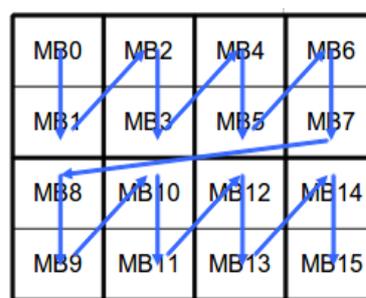
Além da seletividade dos modos de agrupamento dos campos, com MBAFF é alterada a ordem de codificação/decodificação dos macroblocos. Ao invés de aplicar a tradicional ordem de codificação raster scan à macroblocos, a mesma é aplicada à pares de macroblocos, conforme Figura 23.

0 e 1	2 e 3	4 e 5	6 e 7	8 e 9	10	12	14 e 15
0 e 1	2 e 3	4 e 5	6 e 7	8 e 9	11	13	14 e 15
16 e 17	18 e 19	20 e 21	22 e 23	24 e 25	26	28	30
16 e 17	18 e 19	20 e 21	22 e 23	24 e 25	27	29	32
32 e 33	34 e 35	36 e 37	38 e 39	40 e 41	42	44	46
32 e 33	34 e 35	36 e 37	38 e 39	40 e 41	43	45	47

Figura 11 Exemplo De Codificação MBAFF.



(a) *Raster scan* - macroblocos.



(b) *Raster scan* - pares de macroblocos.

Figura 12 Ordem De Codificação.

2.5 Predição Inter-quadro

Ao analisar uma sequência de vídeo, é perceptível a presença de regiões de pixels que se repetem em diferentes quadros. Esta região pode estar na mesma posição espacial nos diferentes quadros, ou podem ainda apresentar deslocamentos de um quadro para outro. Para a compressão de dados é vantajoso, nestes casos, guardar apenas a informação do deslocamento desta região, ao invés dos valores das amostras em cada quadro da sequência. Esta é a principal função da predição inter-quadro, ou seja, tratar a correlação entre quadros, ou redundância temporal.

O H.264/AVC revelou inovações na predição inter-quadro que resultaram em um grande aumento de complexidade computacional, mas também em ganhos significativos

para a compressão. Neste modo de predição estão as principais fontes de ganhos do H.264/AVC em relação aos padrões de compressão de vídeo até então desenvolvidos (RICHARDSON, 2003).

No H.264/AVC as informações de deslocamento de pixels são manipuladas na forma de vetores e índices. A região indicada pelos índices e vetores é chamada de região de referência (também citada na literatura como *best match*, vide Figura 13). Esta região pode não representar necessariamente o real deslocamento de um objeto da cena de captura, mas deve ser a escolha que resulta nos resíduos de menor magnitude possível, ou seja, a região de maior correspondência com o bloco atual. Isto proporciona uma redução do número de bits que serão necessários para representar os resíduos. Evidentemente que a consumação deste objetivo dependendo da capacidade de busca do codificador. Esta busca por regiões é realizada pelo módulo de estimação de movimento (ME) nos quadros já codificados, chamados de quadros de referência. Segundo (PURI, 2004) o ME é o módulo que apresenta a maior complexidade computacional dentre todos os módulos do codificador.

Na decodificação, o módulo de compensação de movimento (MC) é responsável por reconstruir o bloco atual a partir das amostras da região de referência. Como mencionado, para identificar esta região são necessários os vetores e índices calculados na codificação, porém estas informações não são diretamente recebidas pelo decodificador. Há também um processo de predição para estes dados. Sendo assim o MC pode ser dividido em dois módulos principais: Preditor de Vetores de Movimento (MVP - *Motion Vector Predictor*) e Processamento de Amostras. Estes dois módulos serão detalhados nos itens 2.5.1 e 2.5.3, respectivamente. Para fins de implementação, há um terceiro módulo a ser considerado para o MC, o acesso à memória de quadros de referência, que será abordado no item 2.5.2.

Os dois módulos que constituem a predição inter-quadro (*Motion Estimation* - ME e *Motion Compensation* - MC) estão estreitamente relacionados e, portanto, as informações apresentadas neste capítulo são válidas para ambos os módulos.

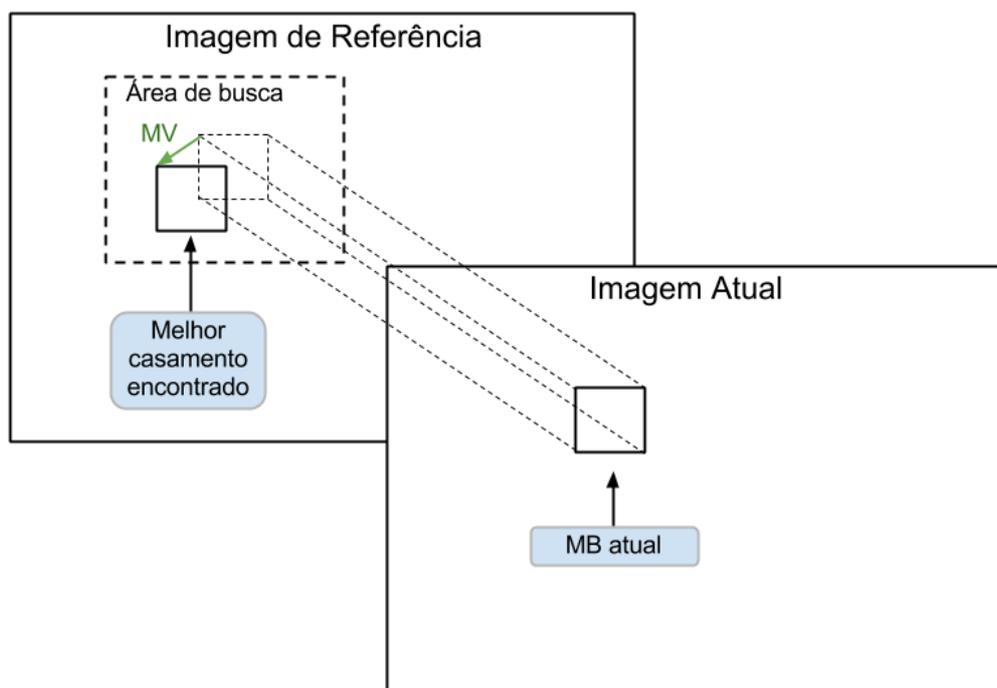


Figura 13 Região De Referência.

Quando é usada predição inter-quadro em *slices* P, ou seja, para macroblocos P, os índices fazem referência somente a imagens presentes na lista 0. Em macroblocos B, presentes em *slices* B, pode-se utilizar também imagens da lista 1. Neste caso é permitido usar imagens de referência da lista 0, lista 1 ou até mesmo uma imagem de cada lista para o mesmo MB. Quando as duas listas são usadas, um único bloco é reconstruído a partir de dois vetores de movimento e dois índices de referência, o que é chamado de bi-predição. Neste caso as duas regiões de referência passam pelo processo de interpolação de amostras, que será bordado no item 2.5.3.1, e o bloco corrente é o resultado da média entre as duas regiões já interpoladas.

No H.264/AVC os macroblocos podem ser divididos e sub-divididos em partições que são codificadas separadamente. Quando se utiliza codificação inter-quadro são permitidas divisões em forma de 16x8, 8x16 ou 8x8 pixels de luminância e os blocos 8x8 ainda podem ser sub-divididos em 8x4, 4x8 ou 4x4 pixels de luminância, Figura 14. Desta forma, para cada MB existem 259 possibilidades diferentes de particionamento. Se for escolhida a utilização de somente blocos 4x4, poderá haver 32 vetores de movimento para um único macrobloco. Isso não acontece com os índices de referência, pois são limitados a blocos 8x8, ou seja, partições menores que 8x8 compartilham o mesmo índice.

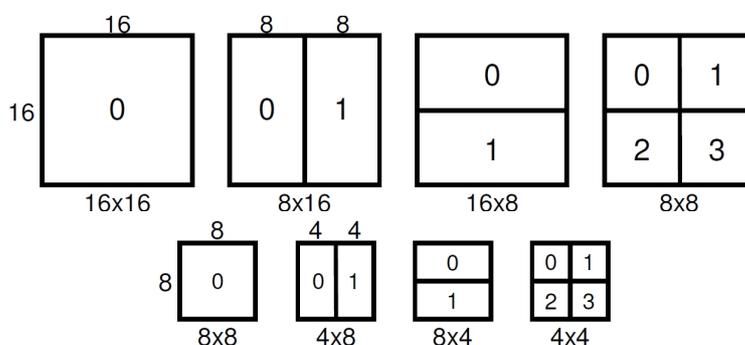


Figura 14 Particionamento De MB, Resolução De Luminância (ZATT, 2008).

Assim como as possibilidades de particionamento do MB, a precisão de 1/4 de pixel de luminância (e 1/8 de pixel de crominância) dos vetores de movimento (MV) foram inovações da predição inter-quadro do H.264/AVC. Vetores com precisão superior à distância entre pixels possibilitam um melhor casamento entre a predição do MB atual e a região do quadro de referência, resultando assim em uma redução na magnitude dos resíduos da codificação. Na Figura 15 há exemplos das magnitudes que a parte fracionária dos vetores de movimento podem assumir. Esta ilustração é direcionada para sub-amostragem 4:2:0, onde a menor magnitude de vetor de crominância é 1/8 da distância entre

as amostras inteiras, que corresponde espacialmente a $1/4$ da distância entre amostras inteiras de luminância.

O MV pode ainda apontar para fora da borda da imagem, neste caso as amostras das bordas são utilizados para completar a região de referência. A Figura 16 exemplifica esta extrapolação para o canto superior esquerdo de uma imagem. Neste caso, além de replicar as amostras das bordas, a amostra do canto superior esquerdo da imagem é utilizada para completar todo canto superior esquerdo da região de referência.

Na predição de vetores de movimento, que é abordada no item 2.4.1, o H.264/AVC inovou com a predição direta para *slices* B e o cálculo de vetores baseado em vetores vizinhos (RICHARDSON, 2003). No processamento de amostras, a bi-predição e a predição ponderada também foram inovações do padrão.

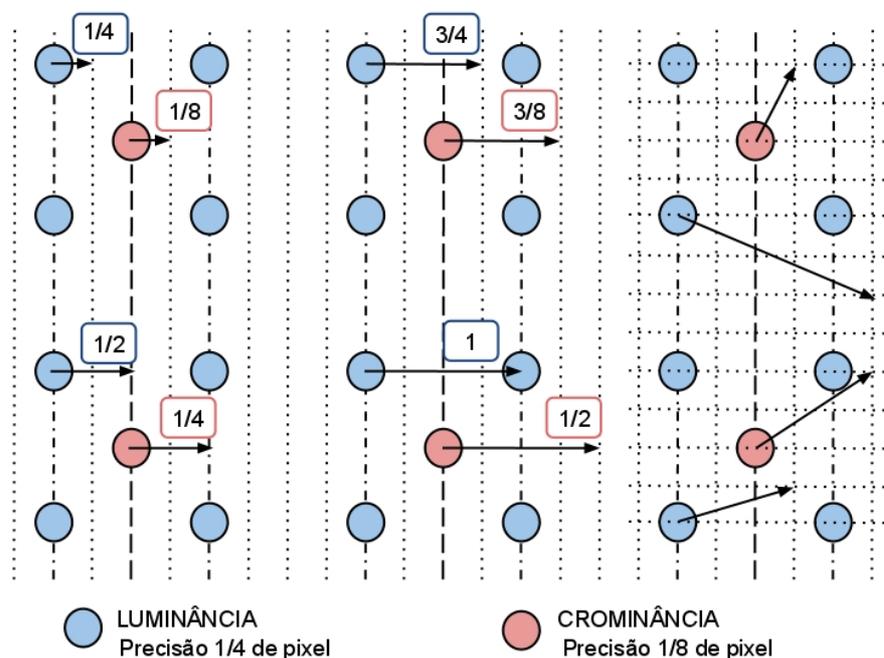


Figura 15 Parte Fracionária De Vetores Para Sub-amostragem 4:2:0.

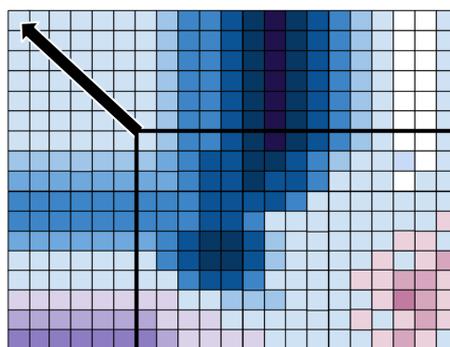


Figura 16 Região De Referência Para Extrapolação De Borda.

Os itens a seguir apresentam as ferramentas de codificação disponíveis no H.264/AVC para a codificação através da predição inter-quadro, com ênfase nas características mais relevantes à compensação de movimento.

2.5.1 Predição de Vetores de Movimento

Como intuito de reduzir a quantidade de dados necessária para transmitir ou armazenar os vetores de movimento e índices de referência, estes sofrem um processo de predição. Este processo é realizado pelo Preditor de Vetores de Movimento (MVP), que é parte integrante do módulo de Compensação de Movimento (MC). Os índices e vetores do MB atual são inferidos de vetores e índices de macroblocos já decodificados, e são chamados de *Predictive Motion Vectors* (PMV). No codificador o vetor de movimento (MV) calculado no ME, é subtraído do PMV, resultando no vetor de movimento diferencial (*Motion Vector Difference* - MVD), ver Figura 17. Estes vetores diferenciais são os únicos dados referentes aos vetores que são codificados no *bitstream*, ou seja, que estão disponíveis ao decodificador. Portanto, na decodificação, o processo de predição de vetores é realizado novamente e o vetor de movimento MV é reconstruído somando o MVD ao PMV.

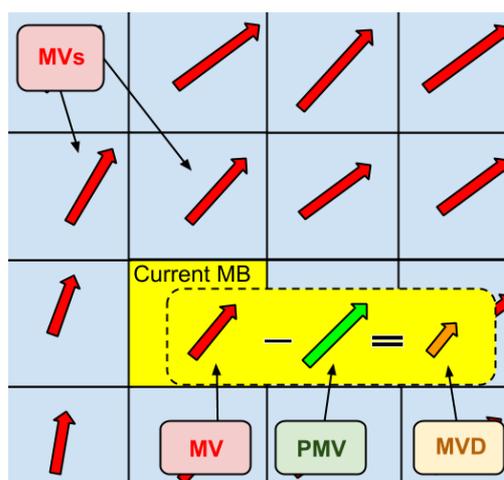


Figura 17 Geração Do *Motion Vector Difference* (MVD).

O PMV pode ser inferido de diversas formas, que basicamente podem ser resumidas em três modos: predição padrão, predição direta temporal e predição direta espacial. Na predição padrão são usados os vetores e índices dos macroblocos vizinhos “A”, “B”, “C”, e “D” (vide Figura 18) para inferir os vetores e índices do MB atual. De acordo com a disponibilidade dos macroblocos vizinhos e do tipo de particionamento, o PMV é assumido como sendo a mediana de três vetores vizinhos ou o MV de um único vizinho.

Para as predições direta espacial e direta temporal são considerados os vetores e índices do bloco co-localizado. No H.264/AVC é considerado como co-localizado o bloco da imagem com índice “0” na lista 1 que está na mesma posição espacial do macrobloco atual, Figura 19. A predição direta espacial, além de considerar os vetores do bloco co-localizado, e assim como a predição padrão, é baseada nos vetores da vizinhança do MB (“A”, “B”, “C”, e “D”). Já a predição direta temporal infere os índices e vetores do MB corrente usando como base somente os vetores e índices do bloco co-localizado. No caso do PMV, são ainda levados em conta fatores de ponderação derivados da distância entre os quadros de referência e o quadro atual.

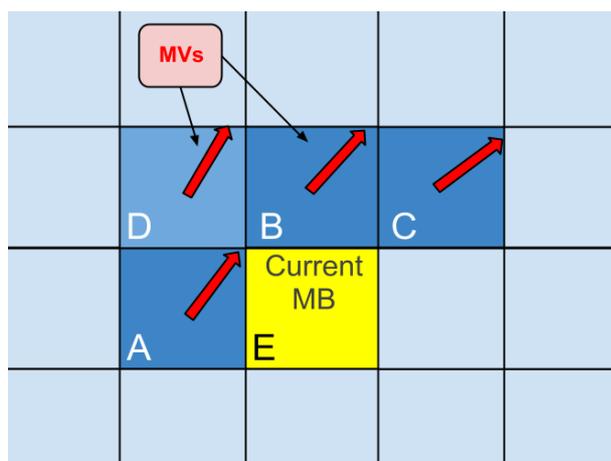


Figura 18 Vizinhança Do Macrobloco.

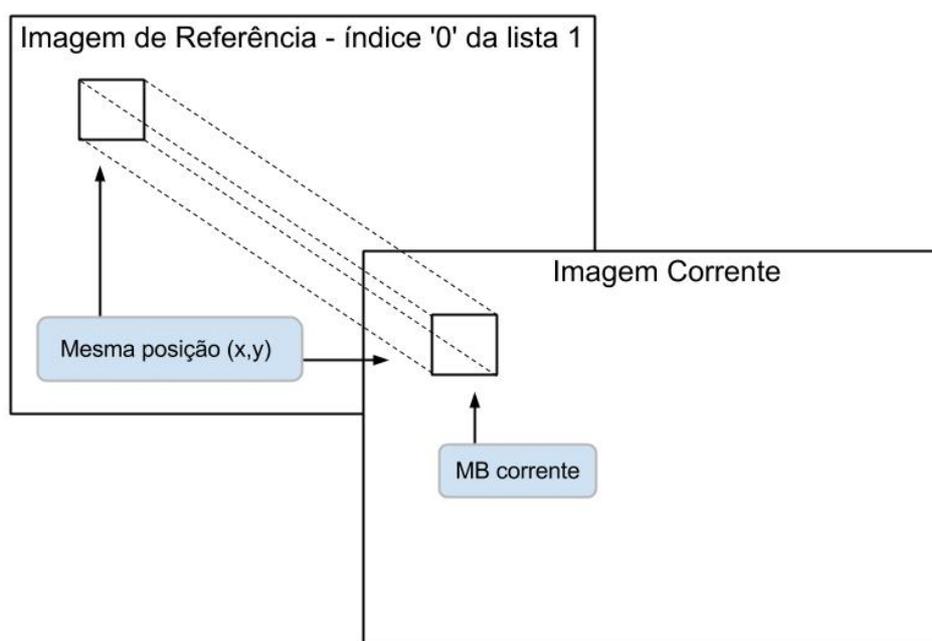


Figura 19 Bloco Co-localizado.

Tanto para a predição direta espacial quanto para a predição direta temporal não são codificados no *bitstream* os vetores de movimento diferenciais (MVD). Após a predição do PMV, este é assumido como sendo o próprio MV do bloco corrente.

Para alguns macroblocos, além do MVD, também não são codificados no *bitstream* os resíduos ou qualquer outra informação de codificação. Estes macroblocos, chamados de

skip, contribuem significativamente com a elevação da taxa de compressão do padrão. É comum constatar o uso de 50% a 70% destes macroblocos em *bitstreams* codificados no perfil *main* do H.264/AVC (KANNANGARA, 2005) (YOO, 2010). Independente do tipo de decodificação, macroblocos *skip* são tratados sempre como blocos 16x16 pixels de luminância, ou seja, MB sem particionamento.

2.5.2 Memória de Quadros de Referência

Levando em consideração o diagrama de blocos básico do CODEC H.264/AVC (apresentado nas Figuras 6 e 7), os dados de entrada do MC são originários de duas fontes. A primeira fonte é o parser, que trabalha em conjunto com a decodificação de entropia extraído do *bitstream* os parâmetros necessários à decodificação, e a segunda fonte é a memória de quadros de referência, que armazena os quadros previamente decodificados. Conforme Figuras 6 e 7, os quadros armazenados na memória são o produto final da decodificação, após a filtragem. Como citado anteriormente, a memória é organizada em duas listas distintas, lista 0 e lista 1, sendo que em *slices* P somente são utilizados os quadros da lista 0, e para *slices* B podem ser usadas ambas as listas, L0 e L1. A ordem de codificação dos quadros de uma sequência de vídeo H.264/AVC não necessita ser a ordem de exibição, portanto, os quadros das listas podem corresponder temporalmente a imagens do passado ou do futuro. No caso da bi-predição, um dos índices aponta para a lista 0 e o outro para a lista 1. Porém, em relação à ordem de exibição, as duas imagens referenciadas podem estar no passado, no futuro ou uma no passado e outra no futuro, Figura 20.

Ao longo da codificação/decodificação as listas de referência são atualizadas, a cada nova imagem com predição inter-quadro. Cada imagem decodificada é rotulada como “*used as reference*”, imagens que irão compor uma das listas na memória de referências, ou “*not*

used as reference”, imagens que somente permanecem em um buffer de saída. Para cada novo *slice*, pode ainda ocorrer um reordenamento, que é dependente de parâmetros enviados explicitamente pelo codificador. Cada quadro das listas é ainda identificado como: “*long term reference*” ou “*short term reference*”. Um quadro marcado como “*long term reference*” não é retirado da memória quando ocorre a atualização das listas. Esta marcação é utilizada quando o quadro deve servir de referência a imagens distantes no tempo. Já os quadros “*short term reference*” podem ser retirados das listas durante uma atualização, e, portanto, permanecem na memória por um período mais curto. As marcações “*short term reference*” e “*long term reference*” são alteradas durante a decodificação.

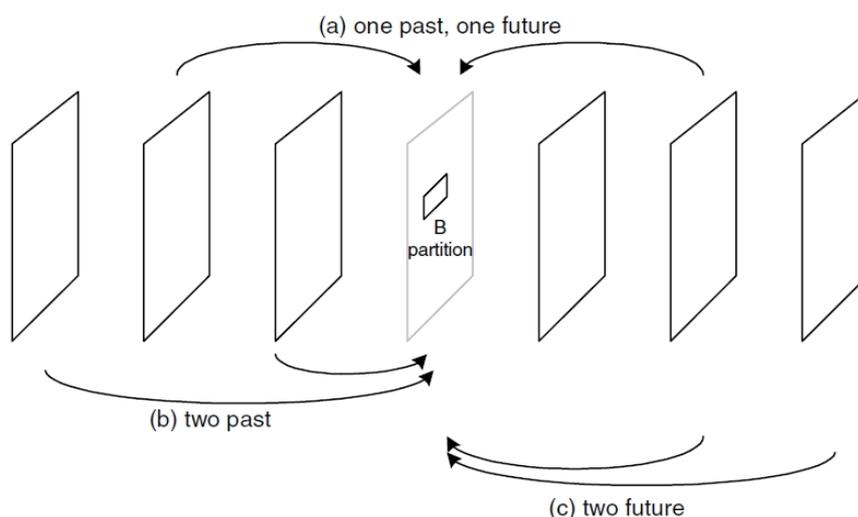


Figura 20 Referências Para Bi-predição (RICHARDSON, 2003).

2.5.3 Processamento de Amostras

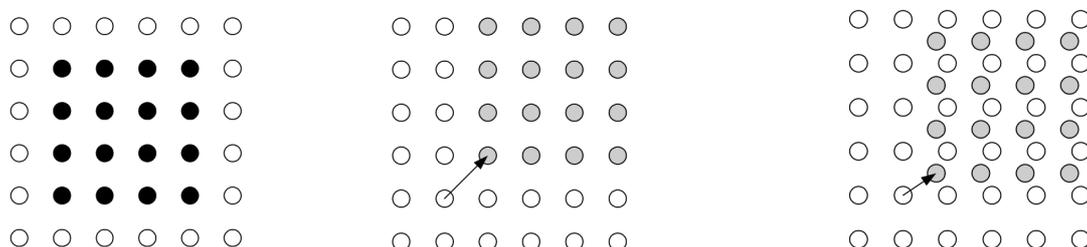
Este módulo é responsável por transformações nas amostras, da área de referência, que visam encontrar amostras para representar o MB corrente. Devido a quantidade de dados a ser processada, o processamento de amostras é o módulo crítico do compensador de movimento (MC). Para um vídeo de alta definição no formato 1920x1088, com sub-

amostragem 4:2:0 e uma taxa de 30 quadros por segundo, a taxa de saída pode ser superior a 94 milhões de amostras por segundo. Se para a codificação deste mesmo vídeo somente for aplicado predição inter-quadro, blocos 4x4 e bi-predição, o decodificador deve processar mais de 697,19 milhões de amostras por segundo.

O processamento de amostras pode ser dividido em: interpolação de amostras, bi-predição e predição ponderada. Estas ferramentas de codificação são abordadas no itens subsequentes.

2.5.3.1 Interpolação de Amostras

Levando em consideração os vetores de movimento com precisão superior a distância entre pixels, o MV pode apontar diretamente para as amostras da região de referência (vetores inteiros - vide Figura 21(b)) ou apontar para amostras que não existem, (posições fracionárias) Figura 21(c). A interpolação é a parte do processamento de amostras responsável por calcular o valor dos pixels quando os vetores são fracionários. Estes pixels são calculados com base nos valores inteiros das amostras da região de referência. A Figura 22 ilustra a região de referência necessária para cálculo de amostras fracionárias entre quatro amostras inteiras de luminância.



(a)Bloco Atual 4x4 (b)Amostras Para Vetor (1,-1) (c)Amostras Para Vetor (0.75, -0.5)
Figura 21 Amostras De Luminância *Half-pel* (RICHARDSON, 2003).

Quando o vetor tem precisão de 1/2 de pixel, por exemplo (-1 , 0.5), as amostras que serão calculadas são chamadas de *half-pel* (amostras rotuladas com letra minúscula na Figura 22). Para vetores com precisão de 1/4 de pixel, por exemplo (-1.25 , 0.5), as respectivas amostras são chamadas de *quarter-pel* (amostras rotuladas com letra minúscula e fundo branco na Figura 23). As amostras de luminância *half-pel* são obtidas através de um filtro FIR, que usa como entrada seis amostras inteiras ou outras seis amostras *half-pel* previamente calculadas, como no cálculo da amostra “b” exposto pela equação 1.

$$b = \text{round}((E - 5F + 20G + 20H - 5I + J)/32) \quad (1)$$

Como pode ser deduzido da própria nomenclatura, a função *round* realiza o arredondamento da equação. As posições das amostras de entrada “E”, “F”, “G”, “H”, “I”, “J”, assim como da amostra de saída “b”, estão ilustradas na Figura 22. Os pesos atribuídos a cada amostra são definidos pela (ITU, 2005).

Para o cálculo das amostras *quarter-pel*, além das amostras inteiras, são utilizadas também as amostras *half-pel* previamente calculadas. O cálculo compreende a média direta de amostras vizinhas, conforme Figura 23. A equação 2 exemplifica o cálculo das amostras *quarter-pel*.

$$a = \text{round}((G + b)/2) \quad (2)$$

No exemplo da equação 2 a amostra “a” é calculada a partir de uma amostra inteira “G” e um *half-pel* “b”, pois está espacialmente entre estas amostras, conforme Figura 23. Em outros casos podem ser usadas somente amostras *half-pel* para obter uma amostra *quarter-pel*, como é o caso das amostras “i”, “k”, “f”, “q”, “e”, “g”, “p” e “r”.

Para sub-amostragem 4:2:0 o uso de vetores de luminância com precisão de 1/4 de pixel requer que os vetores de crominância tenham precisão de 1/8 de pixel, conforme a

Figura 15. Portanto a interpolação das amostras de cromaticidade não ocorre conforme o processo descrito acima. Para qualquer vetor com precisão superior a pixels inteiros, é usado interpolação bilinear das quatro amostras inteiras ao redor da amostra fracionária, Figura 24. As amostras inteiras mais próximas da amostra “a” tem peso maior na interpolação, conforme equação 3.

$$a = \text{round}((A \cdot (8 - dx) \cdot (8 - dy) + dx \cdot B \cdot (8 - dy) + dy \cdot C \cdot (8 - dx) + dx \cdot dy \cdot D) / 64) \quad (3)$$

Na equação 3 as variáveis “A”, “B”, “C” e “D” são amostras inteiras da região de referência que estão ao redor da amostra “a”, a ser calculada. O peso aplicado a cada amostra inteira depende das distâncias “dx” e “dy”, dadas em pixels, conforme Figura 24.

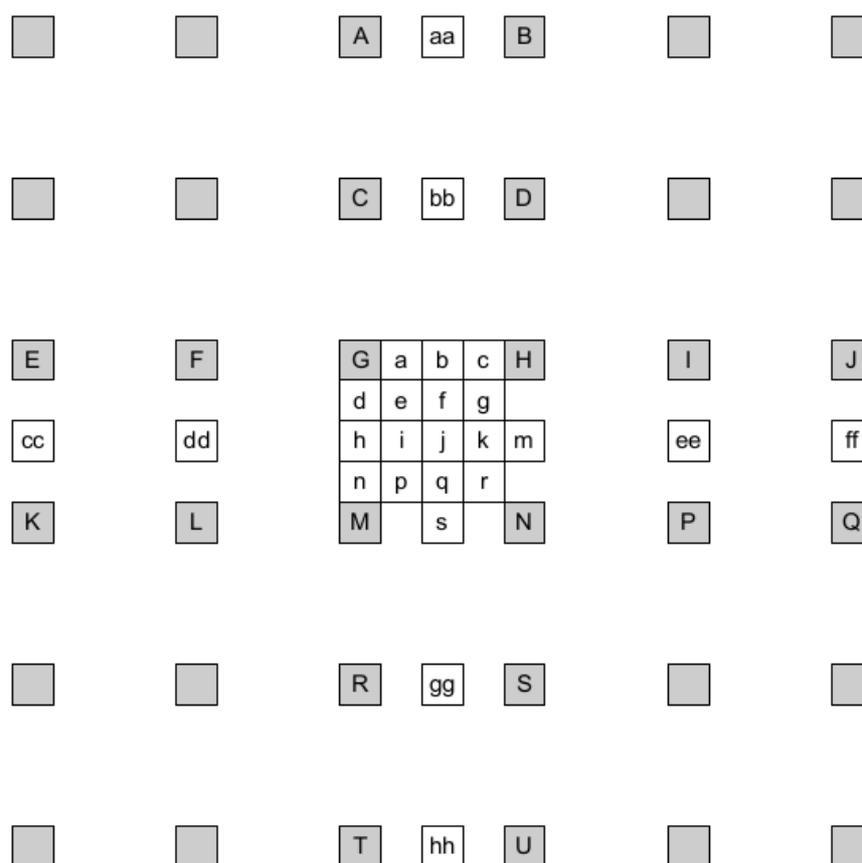


Figura 22 Interpolação De Amostras De Luminância *Half-pel* (ITU, 2005).

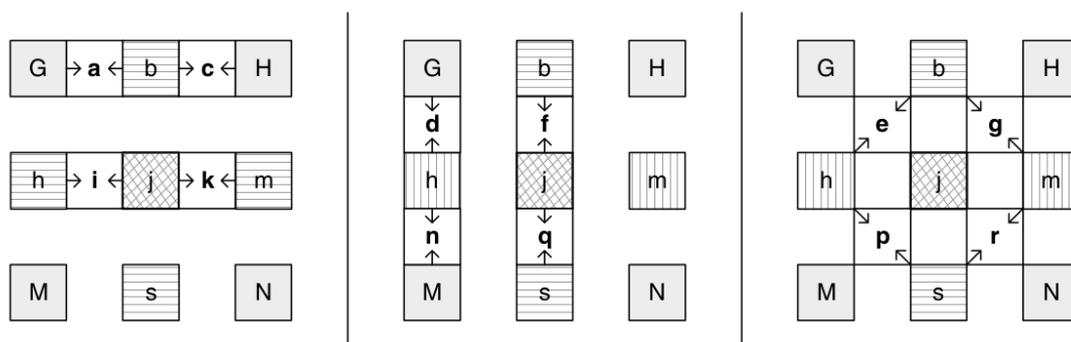


Figura 23 Interpolação De Luminância *Quarter-pel* (RICHARDSON, 2003).

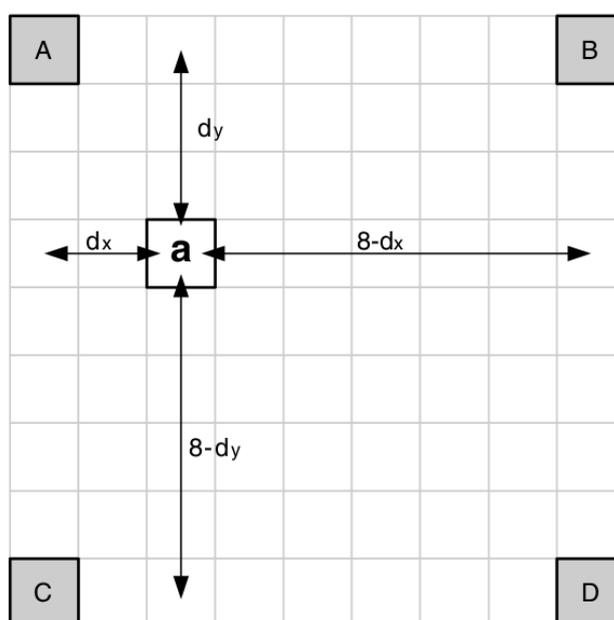


Figura 24 Interpolação De Amostras De Crominância (RICHARDSON, 2003).

2.5.3.2 Predição L0, Predição L1 e Bi-predição

As predições L0 e L1 se caracterizam pelo uso de somente uma das listas de referência (lista 0 ou lista 1 respectivamente). Na bi-predição são usadas as duas listas, o que implica no emprego de dois vetores e dois índices para cada MB ou partição. Portanto, há a indicação de duas regiões de referência. Cada índice deve necessariamente apontar, um para uma das imagens da lista 0 e o outro para uma imagem da lista 1, conforme Figura 25.

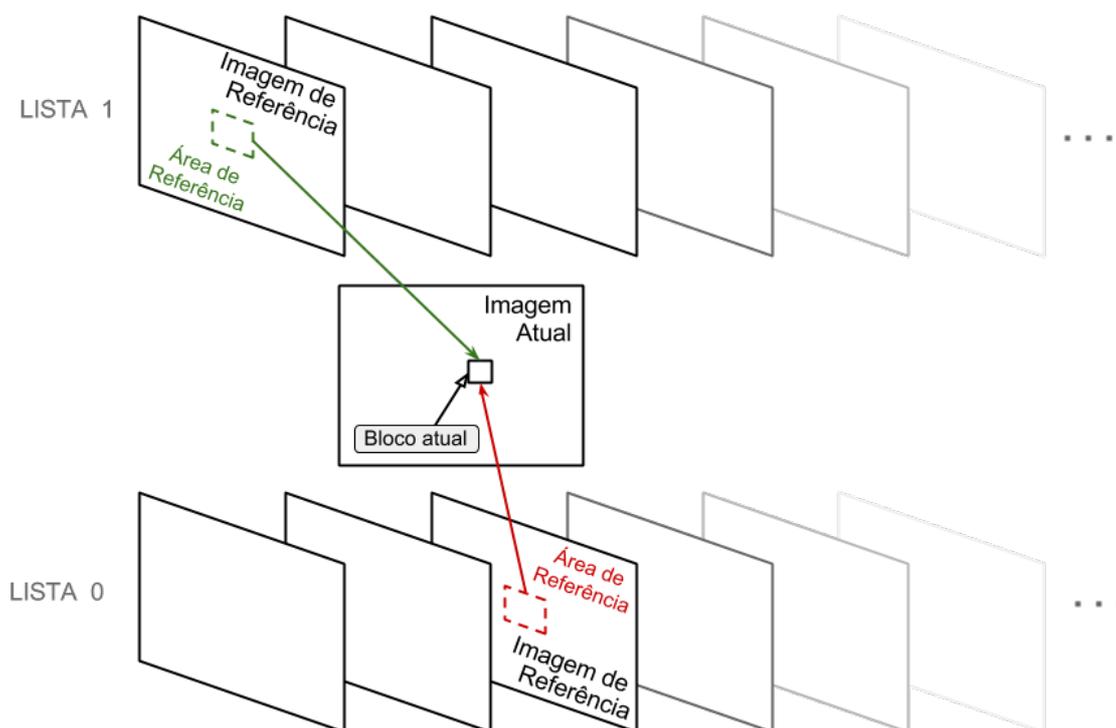


Figura 25 Referências Para Bi-predição.

Neste caso é realizada a média das amostras preditas referentes a cada imagem, ou seja, a média do resultado da interpolação das amostras de cada região de referência. Este tratamento adicional das amostras devido ao uso de bi-predição é comumente realizado no módulo de predição ponderada, apresentado a seguir.

2.5.3.3 Predição Ponderada

O processo de ponderação de amostras, ou módulo de predição ponderada, pode ser descrito como um escalonamento aplicado às amostras após a interpolação. Este processo é muito útil quando há esvanecimento na sequência de vídeo, vide exemplo da Figura 26. Para o cálculo dos valores das amostras ponderadas existe um fator de escala que é associado a cada quadro de referência, cada novo slice e para cada componente de cor. Os parâmetros que compõem este fator podem ser inferidos ou recebidos pelo decodificador

diretamente no *bitstream*, no modo implícito e no modo explícito respectivamente. Quando utilizado o modo implícito, o fator de escala é derivado com base na distância temporal do quadro de referência e o quadro corrente. Slices P e SP podem ser ponderados no modo explícito, enquanto que para slices B podem ser aplicados os dois modos.



Figura 26 Exemplo De Esvanecimento Em Vídeo.

A equação 4 define o cálculo das amostras ponderadas “ $pred(i,j)$ ”, para lista 0.

$$pred(i, j) = \left\lfloor \left((pred_0(i, j) * w_0 + 2^{\log_{WD}-1}) \gg \log_{WD} \right) + o_0 \right\rfloor \quad (4)$$

As amostras “ $pred_0(i,j)$ ” são o resultado da interpolação, “ w_0 ” é o fator de escala, “ o_0 ” é um ajuste aditivo e “ \log_{WD} ” é um valor entre 0 e 7 obtido do logaritmo de base 2 do denominador de todos os fatores de escala. O mesmo ocorre quando é utilizada a lista 1, conforme equação 5.

$$pred(i, j) = \left\lfloor \left((pred_1(i, j) * w_1 + 2^{\log_{WD}-1}) \gg \log_{WD} \right) + o_1 \right\rfloor \quad (5)$$

No caso da bi-predição, as amostras do bloco atual podem ser diretamente calculadas através da equação 6.

$$pred(i, j) = \left((pred_0(i, j) * w_0 + pred_1(i, j) * w_1 + 2^{\log_{WD}-1}) \gg (\log_{WD} + 1) \right) + \left((o_0 + o_1 + 1) \gg 1 \right) \quad (6)$$

2.6 Considerações do H.264

As ferramentas de codificação apresentadas neste capítulo contribuíram para o padrão H.264/AVC alcançar altas taxas de compressão. Este padrão pode reduzir em mais de 300 vezes a quantidade de dados adquiridos em uma sequência de vídeo (RICHARDSON, 2003). O capítulo 3, a seguir, faz uma descrição detalhada do processo de decodificação inter-quadro, e desta forma deixa explícito o tratamento necessário ao decodificador para o suporte às ferramentas apresentadas acima.

3 DECODIFICAÇÃO INTER-QUADRO

Como mencionado anteriormente, o processo de codificação inter-quadro é composto pelos módulos ME e MC, enquanto que a decodificação se resume somente à Compensação de Movimento. Para fins de implementação o módulo de compensação de movimento (MC) é dividido em três módulos principais: Preditor de Vetores de Movimento (MVP - *Motion Vector Predictor*), Acesso a Memória de Quadros de Referência e Processamento de Amostras. Devido a dependências no processo de decodificação, usualmente a estrutura da compensação de movimento é a composição destes três módulos combinados conforme Figura 27, tanto para o desenvolvimento em software quanto para hardware (ITU, 2003; PURI, 2004; AZEVEDO, 2006; ZATT, 2008).

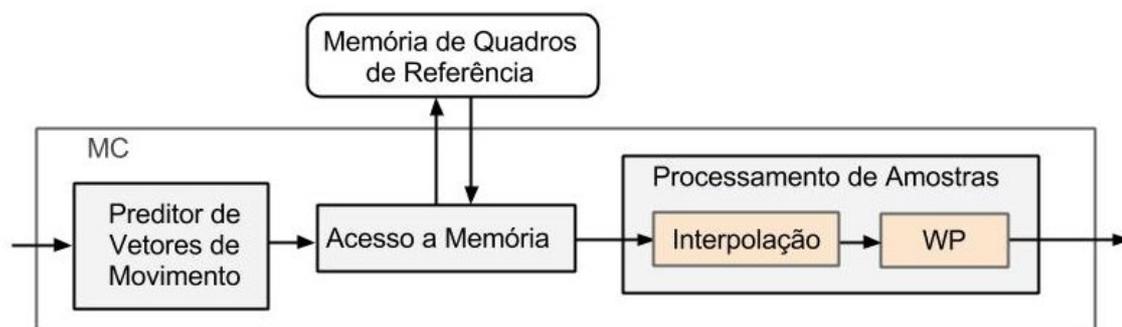


Figura 27 Estrutura Típica De MC Para O Padrão H.264/AVC.

O projeto de um decodificador H.264/AVC deve contar com um módulo de análise do *bitstream* (conhecido também como módulo de *parsing* ou simplesmente *parser*), onde são derivados uma série de parâmetros utilizados pelos componentes do decodificador. Esta é a primeira etapa da decodificação. Assim como os outros módulos, o MC depende dos parâmetros fornecidos pelo módulo de *parsing*. É através destes parâmetros que são obtidos: o tipo de codificação do quadro, o tipo de slice, tipo de codificação do MB, MVD, os fatores de escala para o modo explícito do WP, entre outros. Por exemplo, o parser

disponibiliza a informação do tipo de codificação do MB através de um elemento sintático chamado `mb_type`, conforme Tabelas 1 e 2. Portanto através do `mb_type` é possível saber o tipo de particionamento, quais listas devem ser utilizadas em cada partição e o tipo de decodificação a ser aplicado.

Tabela 1 Tipos de macrobloco para predição inter-quadro em *slices* P

<code>mb_type</code>	Nome do <code>mb_type</code>	Partição 0	Partição 1	Largura do bloco	Altura do bloco
0	P_L0_16x16	Pred_L0	-	16	16
1	P_L0_L0_16x8	Pred_L0	Pred_L0	16	8
2	P_L0_L0_8x16	Pred_L0	Pred_L0	8	16
3	P_8x8	-	-	8	8
4	P_8x8_ref0	-	-	8	8
inferred	P_Skip	Pred_L0	-	16	16

A maioria das arquiteturas de compensação de movimento encontradas na literatura processam todos os macroblocos em partições de 4x4 pixels de luminância, e consequentemente 2x2 de croma, independente se há particionamento ou não (AZEVEDO, 2006; YU, 2007; SHEN, 2009). Isso causa redundâncias no processo de decodificação, mas também resulta em uma considerável redução de complexidade da modelagem da predição inter-quadro, conforme será evidenciado nos próximos itens.

Baseado em um estudo da (ITU, 2003) ilustrado no Apêndice B, a seguir é apresentado o comportamento de cada um dos módulos que compõem o MC no processo de decodificação. São abordados todos os recursos de codificação da predição inter-quadro do H.264/AVC, incluindo a codificação MBAFF. Após, são expostas as principais características das arquiteturas de compensação de movimento encontradas na literatura, dando ênfase à arquitetura que é utilizada como base para o desenvolvimento do módulo de predição ponderada. Os itens a seguir levam em consideração que as informações

necessárias à predição inter-quadro já foram derivadas pelo parser e estão disponíveis ao MC, conforme descrito pela (ITU, 2003).

Tabela 2 Tipos de macrobloco para predição inter-quadro em *slices* B

mb_type	Nome do mb_type	Partição 0	Partição 1	Largura do bloco	Altura do bloco
0	B Direct 16x16	Direct	-	16	16
1	B L0 16x16	Pred L0	-	16	16
2	B L1 16x16	Pred L1	-	16	16
3	B Bi 16x16	BiPred	-	16	16
4	B L0 L0 16x8	Pred L0	Pred L0	16	8
5	B L0 L0 8x16	Pred L0	Pred L0	8	16
6	B L1 L1 16x8	Pred L1	Pred L1	16	8
7	B L1 L1 8x16	Pred L1	Pred L1	8	16
8	B L0 L1 16x8	Pred L0	Pred L1	16	8
9	B L0 L1 8x16	Pred L0	Pred L1	8	16
10	B L1 L0 16x8	Pred L1	Pred L0	16	8
11	B L1 L0 8x16	Pred L1	Pred L0	8	16
12	B L0 Bi 16x8	Pred L0	BiPred	16	8
13	B L0 Bi 8x16	Pred L0	BiPred	8	16
14	B L1 Bi 16x8	Pred L1	BiPred	16	8
15	B L1 Bi 8x16	Pred L1	BiPred	8	16
16	B Bi L0 16x8	BiPred	Pred L0	16	8
17	B Bi L0 8x16	BiPred	Pred L0	8	16
18	B Bi L1 16x8	BiPred	Pred L1	16	8
19	B Bi L1 8x16	BiPred	Pred L1	8	16
20	B Bi Bi 16x8	BiPred	BiPred	16	8
21	B Bi Bi 8x16	BiPred	BiPred	8	16
22	B 8x8	-	-	8	8
inferred	B_Skip	Direct	-	8	8

3.1 Preditor de Vetores de Movimento

A predição de vetores de movimento é dividida em predição padrão, predição direta espacial e predição direta temporal. Apesar desta separação alguns processos são compartilhados entre a predição direta espacial e a predição padrão. Entre a predição direta espacial e a predição direta temporal também há processo em comum, como é demonstrado nos itens a seguir.

3.1.1 Predição Padrão

O processo de predição de vetores começa com a identificação do tipo de MB ou partição. Para macroblocos do tipo P_L0_16x16, P_L0_L0_16x8 e P_L0_L0_8x16 os vetores são inferidos utilizando a predição padrão. Sendo assim, são utilizados os vetores de movimento dos macroblocos vizinhos “A”, “B”, “C” e “D” para calcular o PMV do bloco atual. Quando os macroblocos não são particionados, estes vizinhos são localizados conforme ilustração da Figura 18. Porém, para outros modos de codificação há um processo de busca dos blocos vizinhos, que podem ser partições de cada MB, como no exemplo da Figura 27.

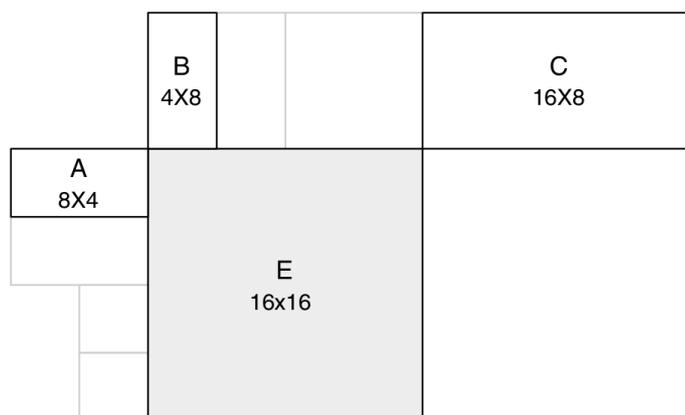


Figura 28 Vizinhança Para Macroblocos Particionados (RICHARDSON, 2003).

Para que estas informações estejam disponíveis, o MVP deve guardar os vetores em um buffer com capacidade de uma linha de macroblocos mais um. No caso das arquiteturas que trabalham sempre com blocos 4x4, para cada MB deve-se guardar os vetores dos blocos da borda inferior e da borda direita, conforme Figura 29.

O processo de busca de vizinhança torna-se mais complexo com o uso de MBAFF, pois, com a codificação em pares de macroblocos, cada vizinho pode ser um MB top ou bottom. Deve-se ainda levar em consideração o tipo de codificação de cada par de

macroblocos, quadro ou campo, este processo está resumido nas Tabelas 3 e 4. Além disso, com MBAFF, a capacidade do buffer que armazena vetores de macroblocos vizinhos deve ser duplicada.

Tabela 3 Vizinhanças para MB tipo frame em imagem MBAFF.

Vizinho	Identificador do par	Codificação do MB vizinho	Informação adicional	Atribuição
A	top	frame		A <= lateral top
		field	Índice do pixel é par Índice do pixel é ímpar	A <= lateral top A <= lateral bottom
	bottom	frame		A <= lateral bottom
		field		A <= lateral top A <= lateral bottom
	B	top	frame/field	
bottom		frame/field		B <= atual top
C	top	frame/field		C <= diagonal direita bottom
	bottom	Não disponível	Vizinho não disponível	Não há vizinhança
D	top	frame/field		D <= diagonal esquerda bottom
	bottom	frame	Vizinho de referência é o lateral	D <= superior top
		field	Vizinho de referência é o lateral	D <= superior bottom

Fonte: Klein, 2012.

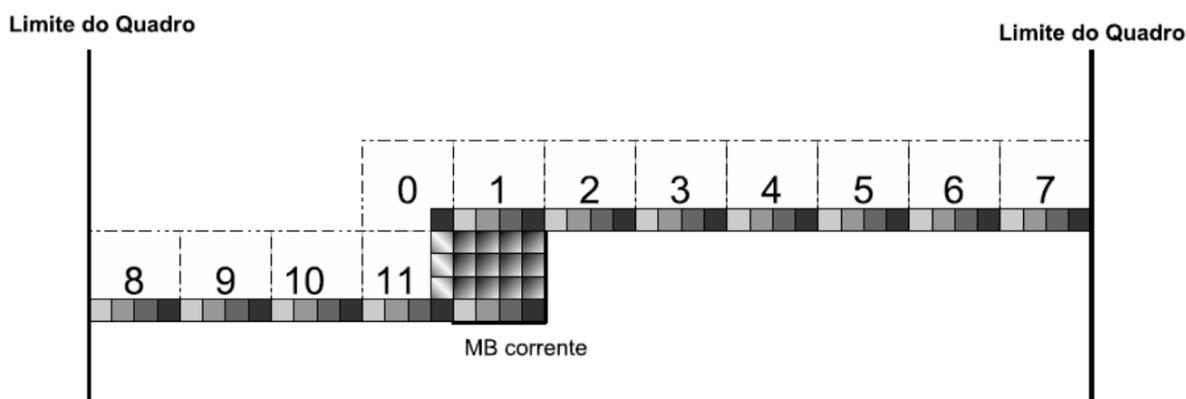


Figura 29 Vizinhança Utilizada No MVP (ZATT, 2006).

Após essa busca, o preditor de vetores pode assumir que o PMV do bloco atual é um dos vetores ou a mediana dos vetores dos macroblocos vizinhos, dependendo do tipo de partição do MB atual e da disponibilidade dos vetores. Como critério de seleção, ainda leva-

se em consideração se o índice de referência do bloco atual é igual ao índice do MB vizinho. Este procedimento está descrito na Tabela 5.

Tabela 4 Vizinhanças para MB tipo field em imagem MBAFF.

Vizinho	Identificador do par	Codificação do MB vizinho	Informação adicional	Atribuição
A	top	frame	Índice do pixel é menor que a metade da altura do MB	A <= lateral top
			Índice do pixel é maior ou igual a metade da altura do MB	A <= lateral bottom
		field		A <= lateral top
	bottom	frame	Índice do pixel é menor que a metade da altura do MB	A <= lateral top
			Índice do pixel é maior ou igual a metade da altura do MB	A <= lateral bottom
		field		A <= lateral bottom
B	top	frame		B <= superior bottom
		field		B <= superior top
	bottom	frame/field		B <= superior bottom
C	top	frame		C <= diagonal direita bottom
		field		C <= diagonal direita top
	bottom	frame/field		C <= diagonal direita bottom
D	top	frame		D <= diagonal esquerda
		field	vizinho de referência é o lateral	D <= superior top
	bottom	frame/field	vizinho de referência é o lateral	D <= superior bottom

Fonte: Klein, 2012.

Exclusivamente para macroblocos P_Skip, o PMV recebe valor zero caso o índice e o vetor do vizinho “A” ou do vizinho “B” estiverem indisponíveis ou forem iguais a zero. Além disso macroblocos P_Skip sempre usam como referência a imagem com índice zero da L0. As partições P_8x8 são outra exceção deste processo, neste caso um parâmetro chamado sub_mb_type indica o tipo de codificação de cada partição.

Tabela 5 Modo padrão de predição do PMV.

Particionamento	Partição atual	Comparação de índices de referência	Atribuição do vetor predito (PMV)
16x8	0	$RefIdx = RefIdx_B$	PMV \leq MV_B
		$RefIdx \neq RefIdx_B$	PMV \leq Mediana(A, B, C)
	1	$RefIdx = RefIdx_A$	PMV \leq MV_A
		$RefIdx \neq RefIdx_A$	PMV \leq Mediana(A, B, C)
8x16	0	$RefIdx = RefIdx_A$	PMV \leq MV_A
		$RefIdx \neq RefIdx_A$	PMV \leq Mediana(A, B, C)
	1	$RefIdx = RefIdx_C$	PMV \leq MV_C
		$RefIdx \neq RefIdx_C$	PMV \leq Mediana(A, B, C)
Outros casos	-	-	PMV \leq Mediana(A, B, C)

3.1.2 Predição Direta

Os vetores de macroblocos B_Skip, B_Direct_16x16 e B_Direct_8x8 são inferidos através da predição direta. Neste caso, a escolha da predição direta temporal ou espacial é realizada pelo codificador, que através de uma *flag* sinaliza ao decodificador qual dos dois modos deve ser aplicado. Para todos os outros macroblocos B é aplicada a predição padrão, descrita pela Tabela 5.

3.1.2.1 Predição Direta Espacial

A predição direta espacial também usa os vetores e índices dos macroblocos vizinhos para inferir os vetores e índices do MB atual. Portanto é realizada a busca pelos macroblocos descrita pelas Tabelas 3 e 4. Além disso são usadas informações do bloco co-localizado, que é obtido conforme a descrição do item 3.1.2.3.

Inicialmente são inferidos os índices de referência do bloco atual, que são assumidos como sendo os menores índices válidos dentre os índices das respectivas listas nos vizinhos “A”, “B” e “C”. Como exceção a esta regra, quando o valor dos dois índices derivados for

negativo, os índices do bloco atual recebem o valor zero. Logo após são preditos os vetores de movimento, conforme Tabela 6.

Tabela 6 MV para predição direta espacial.

Condição	Vetor de Movimento
O quadro com índice 0 na lista 1 for "used for short-term reference", o índice de referência do bloco co-localizado for igual a 0, a magnitude do vetor de movimento do bloco co-localizado e o índice de referência do bloco atual forem menores que 1	$MV \leq 0$
Caso contrário	É aplicado o processo de predição padrão, descrito na Tabela 5

3.1.2.2 Predição Direta Temporal

Na predição direta temporal também são utilizadas informações do bloco co-localizado, portanto, o primeiro passo da decodificação neste modo é a localização deste bloco, conforme item 3.1.2.3. O processo começa com a derivação dos índices de referência. O índice de referência do bloco atual para lista 0 (RefIdxL0) deve apontar para a mesma imagem de referência utilizada pelo bloco co-localizado (RefPicCol). Como as listas são dinâmicas, há a necessidade de preservar informações relativas à organização da lista 0 ao longo da decodificação. Para vídeos entrelaçados se deve ainda levar em consideração que a imagem de referência RefPicCol, a imagem co-localizada e a imagem atual, podem ser um campo, um quadro ou uma imagem MBAFF. Dependendo do tipo de codificação da imagem atual, da imagem co-localizada, do MB atual e do bloco co-localizado o valor do índice pode ainda sofrer modificações, conforme mostrado na Tabela 7. O índice para lista 1 é assumido como igual a zero, ou seja, aponta para a imagem do bloco co-localizado.

Ao contrário da predição direta espacial, na predição direta temporal o cálculo do vetor não leva em consideração os vizinhos espaciais do MB. Os vetores mvL0 e mvL1,

respectivamente para as listas L0 e L1, são obtidos através do próprio valor do vetor co-localizado ($mvCol$), com o auxílio de parâmetros derivados da distância temporal entre os quadros. O procedimento ocorre da seguinte forma:

- Se a imagem referenciada pelo índice $RefIdxL0$ está rotulada como “*long-term reference picture*”, o $mvL1$ é igual a zero. Para o $mvL0$ o seguinte se aplica:
 - Se o Mb atual está codificado como *frame* e o bloco co-localizado está codificado como *field*, deve-se dobrar a dimensão vertical do $mvCol$ e atribuir o vetor resultante ao $mvL0$.
 - Caso contrário, e se o Mb atual está codificado como *field* e o bloco co-localizado está codificado como *frame*, deve-se dividir a dimensão vertical do $mvCol$ e atribuir o vetor resultante ao $mvL0$.
- Caso contrário, o seguinte se aplica:
 - Se o Mb atual está codificado como *frame* e o bloco co-localizado está codificado como *field*, deve-se dobrar a dimensão vertical do $mvCol$ e aplicar o vetor resultante nas equações 7 e 8.

$$mvL0 = (DistScaleFactor * mvCol + 128) / 256 \quad (7)$$

$$mvL1 = mvL0 - mvCol \quad (8)$$

- Se o Mb atual está codificado como *field* e o bloco co-localizado está codificado como *frame*, deve-se dividir a dimensão vertical do $mvCol$ e aplicar o vetor resultante nas equações 9 e 10.

$$mvL0 = (DistScaleFactor * mvCol + 128) / 512 \quad (9)$$

$$mvL1 = mvL0 - mvCol \quad (10)$$

Tabela 7 Índice de referência RefIdxL0 para predição direta temporal.

Imagem atual	Imagem co-localizada	MB atual	MB co-localizado	Mesma paridade entre MB atual e campo de referência	Índice de referência para lista 0
Field	Field				RefIdxL0 <= índice da L0 corrente que aponta para RefPicCol
	Frame			Sim	RefIdx0 <= índice da L0 corrente que aponta para o campo da RefPicCol (este caso não é permitido no SBTVD)
				Não	-
	MBAFF		Frame	Sim	RefIdx0 <= índice da L0 corrente que aponta para o campo da RefPicCol
			Field	Não	-
	Frame	Field			
Frame					RefIdxL0 <= índice da L0 corrente que aponta para RefPicCol
MBAFF	Field	Frame			RefIdx0 <= índice da L0 corrente que aponta para o quadro que contém o RefPicCol
		Field		Sim	RefIdx <= 2 x (índice da L0 corrente que aponta para o quadro que contém o RefPicCol)
	Não			RefIdx <= 1 + 2 x (índice da L0 corrente que aponta para o quadro que contém o RefPicCol)	
	MBAFF	Frame	Frame		RefIdxL0 <= índice da L0 corrente que aponta para RefPicCol
			Field		RefIdx0 <= índice da L0 corrente que aponta para o quadro que contém o RefPicCol
	MBAFF	Field	Frame		RefIdx0 <= 2 x (índice da L0 corrente que aponta para RefPicCol)
			Field	Sim	RefIdx <= 2 x (índice da L0 corrente que aponta para o quadro que contém o RefPicCol)
		Não		RefIdx <= 1 + 2 x (índice da L0 corrente que aponta para o quadro que contém o RefPicCol)	

O “DistScaleFactor” é calculado a partir de t_b e t_d , parâmetros que representam respectivamente a distância temporal entre as imagens atual e RefPicCol, e a distância temporal entre a imagem co-localizada e a RefPicCol. A Figura 30 ilustra os vetores preditos e os parâmetros t_b e t_d .

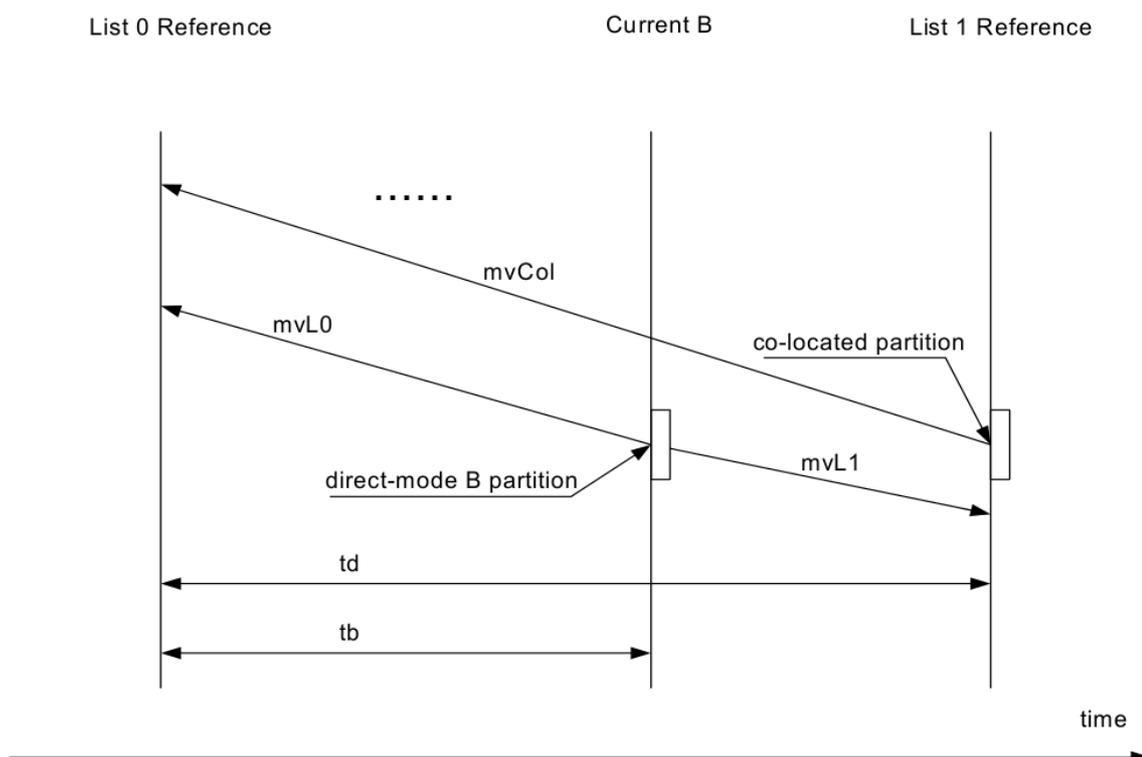


Figura 30 Vetores Para Predição Direta Temporal (ITU, 2005).

3.1.2.3 Identificação do Bloco Co-localizado

Este processo busca basicamente identificar a posição, o vetor $mvCol$ e o índice de referência $RefIdxCol$ do bloco co-localizado. Estas informações são essenciais às predições direta espacial e direta temporal. Quando as imagens da sequência de vídeo apresentam apenas codificação em modo *frame*, como ocorre com vídeos progressivos, o bloco co-localizado é a partição de MB com a mesma posição espacial da partição do macrobloco atual, conforme Figura 19. Porém para vídeos entrelaçados o uso de codificação *field* e

MBAFF para as imagens e o emprego de codificação tipo *field* ou *frame* para cada par de macroblocos exige um processo para localização do bloco co-localizado. Para isso deve ser encontrada a posição vertical y_M do bloco, conforme Tabela 8, enquanto que a posição horizontal permanece a mesma do MB corrente.

Tabela 8 Posição vertical y_M para bloco co-localizado.

Imagem atual	Imagem co-localizada	MB atual	MB co-localizado	y_M	Permitido no SBTVD	
Field	Field			y_{Col}	Sim	
	Frame			$(2 * y_{Col}) \% 16$	Não	
	MBAFF		Frame		$(2 * y_{Col}) \% 16$	Sim
		Field		y_{Col}	Sim	
Frame	Field			$8 * ((\text{Endereço do MB atual} / \text{Largura da imagem}) \% 2) + 4 * (y_{Col} / 8)$	Não	
	Frame			y_{Col}	Sim	
MBAFF	Field	Frame		$8 * (\text{Endereço do MB atual} \% 2) + 4 * (y_{Col} / 8)$	Sim	
		Field		y_{Col}	Sim	
	MBAFF	Frame	Frame		y_{Col}	Sim
			Field		$8 * (\text{Endereço do MB atual} \% 2) + 4 * (y_{Col} / 8)$	Sim
		Field	Frame		$(2 * y_{Col}) \% 16$	Sim
			Field		y_{Col}	Sim

A variável y_{Col} na Tabela 8 representa a posição vertical do bloco para os casos em que os dois tipos de codificação são iguais (do MB atual e do bloco co-localizado) assim como ocorre na decodificação de vídeos progressivos. Portanto y_{Col} representa a posição vertical do bloco corrente.

O vetor mv_{Col} e o índice $RefIdx_{Col}$ são derivados conforme segue:

- Se a *flag* de utilização da L0 estiver habilitada, para o bloco co-localizado, o mv_{Col} e o $RefIdx_{Col}$ recebem os valores mv_{L0} e $RefIdx_{L0}$ deste bloco.
- Se a *flag* de utilização da L0 não estiver habilitada e a *flag* de utilização da L1 estiver habilitada, o mv_{Col} e o $RefIdx_{Col}$ recebem os valores mv_{L1} e $RefIdx_{L1}$ do bloco co-localizado.

- Caso o MB co-localizado use codificação intra-quadro ou as duas *flags* de utilização das listas L1 e L0 estiverem desabilitadas, *mvCol* recebe zero e *RefIdxCol* recebe -1.

3.2 Acesso à Memória

Frequentemente, é atribuído ao módulo de acesso à memória a função do recorte da região de referência, ou seja, da localização desta região na imagem de referência. Esta é uma abordagem conveniente, pois torna possível disponibilizar as amostras da referência no formato adequado para a entrada da interpolação de amostras, que é o módulo crítico do fluxo de dados na decodificação inter-quadro. Conforme mencionado, na maioria das arquiteturas encontradas na literatura o processamento de amostras é realizado sempre em blocos de 4x4 pixels de luminância, e 2x2 pixels de crominância. Isso implica que, devido a características da interpolação (item 3.3) as regiões de referência são sempre blocos de 9x9 pixels para luminância e 3x3 pixels de crominância. Para a localização destes grupos de amostras o módulo de acesso à memória deve ser capaz de encontrar, no quadro indicado pelo índice de referência, os endereços das amostras apontadas pelo MV.

Nos casos de codificação de vídeo entrelaçado, cada imagem pode ser um par *field*, um *frame* ou MBAFF. Durante o processo de filtragem os campos de uma imagem MBAFF são reorganizados na forma de um *frame*, assim sendo, as imagens MBAFF de referência são armazenadas em formato *frame*. Portanto, a localização das posições de amostras da região de referência é realizada da mesma forma para vídeos progressivos e para vídeos entrelaçados. Para o cálculo da posição da região de referência, os componentes inteiros “x” e “y” dos vetores são somados à posição do bloco corrente e ao deslocamento referente à interpolação de quarto ou oitavo de pixel de luminância e crominância, respectivamente.

Com a posição da região de referência, o índice é utilizado para localizar o quadro de referência na respectiva lista. Porém, para vídeos entrelaçados, o processo de localização da imagem sofre alterações, conforme descrito na Tabela 9. Nesta tabela as listas de referência são representadas por um vetor chamado `RefPicList[]`. Como pode ser visto, a definição da imagem de referência depende do tipo de codificação das referências. Neste caso, o módulo de tratamento das listas de referência fica responsável por fornecer os quadros e campos conforme a descrição na coluna “Imagem Seleccionada”.

Tabela 9 Localização da imagem de referência.

Imagem atual	MB atual	RefIdx%2	Imagem Seleccionada
Field			O campo <code>RefPicList[RefIdx]</code> , ou o campo do quadro <code>RefPicList[RefIdx]</code>
Frame ou MBAFF	Frame		O quadro <code>RefPicList[RefIdx]</code> , ou o par de campos complementares <code>RefPicList[RefIdx]</code>
MBAFF	Field	0	O campo do quadro <code>RefPicList[RefIdx/2]</code> que possui a mesma paridade que o MB atual, ou o campo do par de campos complementares <code>RefPicList[RefIdx]</code> que possui a mesma paridade que o MB atual
		1	O campo do quadro <code>RefPicList[RefIdx/2]</code> que possui paridade oposta ao MB atual, ou o campo do par de campos complementares <code>RefPicList[RefIdx]</code> que possui paridade oposta ao MB atual

Para vídeos entrelaçados é preciso lembrar que, quando a imagem de referência utilizada a codificação *field*, o limite da dimensão vertical é a metade da respectiva dimensão de um quadro.

3.3 Interpolação de Amostras

O processo de interpolação de amostras, como apresentado anteriormente, visa inferir as amostras fracionárias para a luminância Y e as duas crominâncias Cb e Cr. Como

descrito no item 2.5.3.1, para encontrar as amostras fracionárias Y entre quatro amostras inteiras é necessário dispor de uma região com 6x6 amostras inteiras de luminância. Para as crominâncias apenas são suficientes as próprias quatro amostras inteiras Cb ou Cr. Portanto para obter um bloco 2x2 de amostras fracionárias Y, a região de referência deve ter dimensão 7x7, e, conseqüentemente, para 4x4 amostras fracionárias, a região de referência deve ser 9x9, conforme Figura 31. Para as crominâncias uma região de amostras fracionárias 2x2 é obtida a partir de uma região de referência de 3x3 pixels, Figura 32. Assim sendo, para as arquiteturas que trabalham com blocos de 4x4 amostras de luminância e sub-amostragem 4:2:0, o formato de entrada para as amostras deve ser 9x9 Y, 3x3 Cb e 3x3 Cr. Evidentemente, neste caso, a saída de amostras do interpolador é sempre 4x4 Y, 2x2 Cb e 2x2 Cr.

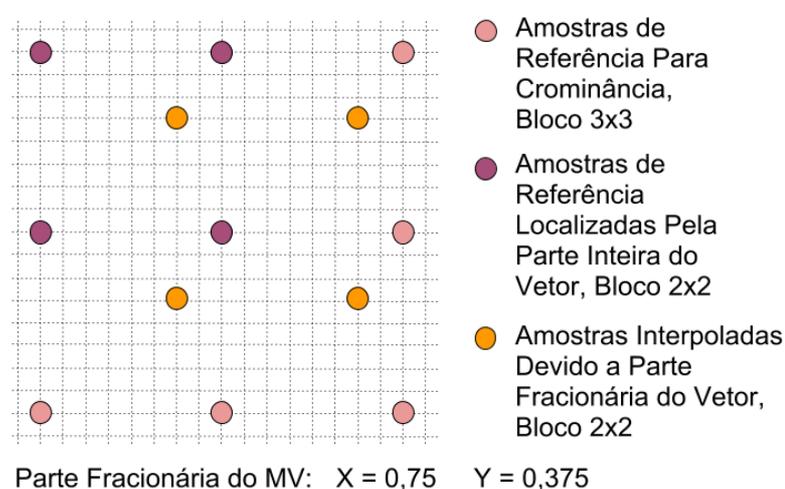


Figura 31 Região De Referência Para Bloco 2x2 De Crominância.

Os modos de codificação da imagem, *field*, *frame* ou MBAFF ou o modo de codificação de cada par de macroblocos, *field* ou *frame*, não possuem influência no processo de interpolação de amostras. Dessa forma o interpolador utilizado para a decodificação de vídeo progressivo pode ser usado para decodificar vídeos entrelaçados.

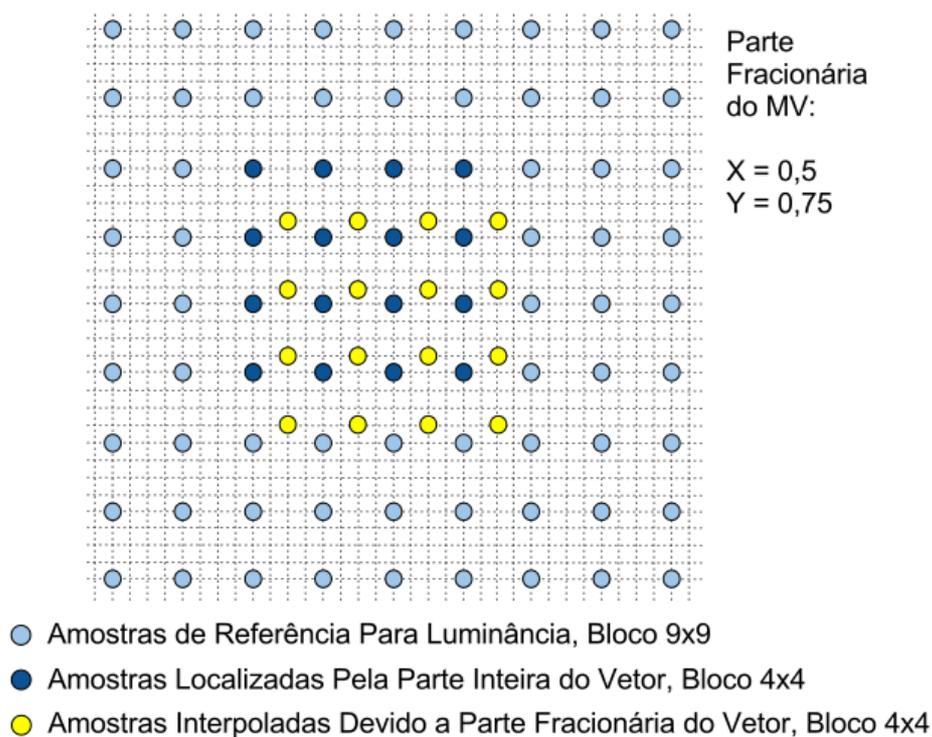


Figura 32 Região De Referência Para Bloco 4x4 De Luminância.

3.4 Predição Ponderada

O processo de predição ponderada (ou *Weighted Prediction* - WP) pode ser dividido em duas etapas distintas: predição dos fatores de escala e cálculo de escalonamento das amostras. A definição do fator de escala a ser aplicado depende da identificação do modo de ponderação de amostras do MB atual, padrão (implícito ou explícito). Para auxiliar na definição do modo utilizado, o WP recebe do parser uma *flag* chamada “weighted_pred” e uma variável chamada “weighted_bipred_idc”, conforme Tabela 10.

No modo padrão as amostras não são ponderadas, porém deve-se fazer a média entre as amostras interpoladas de cada lista para o caso de bi-predição. No modo explícito os fatores de escala são recebidos do codificador, porém a escolha do fator a ser aplicado ao MB corrente depende do tipo de codificação do mesmo. O parser disponibiliza os fatores na

forma de um vetor. Quando é usado MBAFF e o MB atual usa codificação *field*, o WP deve usar o fator que está na posição com índice igual a metade do valor do índice de referência de imagens RefIdx. Para as outras possibilidades de codificação, o WP deve usar o fator que está na posição com índice igual ao RefIdx. O mesmo acontece para o parâmetro “o”, que é utilizado como ajuste aditivo no cálculo de escalonamento de amostras. O “logWD” é o último parâmetro a ser considerado no cálculo de escalonamento de amostras, o valor deste parâmetro é obtido do logaritmo de base 2 do denominador de todos fatores de escala.

Tabela 10 Identificação do modo de ponderação de amostras.

Slice	weighted_pred	weighted_bipred_idc	Utilização das listas	Modo de Predição Ponderada
P	0	-	-	Padrão
	1	-	L0	Explícito
B	-	0	-	Padrão
	-	1		Explícito
	-	2	L0 ou L1	Padrão
			L0 e L1	Implícito

No modo implícito é atribuído valor zero aos fatores aditivos “o0” e “o1” e o parâmetro “logWD” recebe o valor 5, independentemente do tipo de codificação. Os fatores de escala “w0” e “w1” são inferidos de acordo com a distância temporal entre o quadro atual e os quadros de referência. Para isso é calculado o “DistScaleFactor” conforme equação 11.

$$DistScaleFactor = (32 + tb * (16384 + ValorAbsoluto(td/2))) / td / 64 \quad (11)$$

Os parâmetros td e tb são os mesmos utilizados pela predição direta temporal do MVP. Estes podem ser obtidos das diferenças entre o POC (*Picture Order Counter* – contador de imagens que leva em consideração a codificação PAFF) das imagens atual (currPicOrField), referência da lista 0 (pic0) e referência da lista 1 (pic1). As imagens de

referência *pic0* e *pic1* devem ser assumidas conforme descrito na Tabela 9, do módulo de acesso a memória. A variável *currPicOrField* é obtida da seguinte forma:

- Se a imagem atual não está codificada em modo *field* e o MB atual está codificado em modo *field*, a *currPicOrField* é o campo do *frame* atual que possui a mesma paridade que o MB atual.
- Caso contrário, a *currPicOrField* é o próprio quadro atual.

Com base nestas definições, os fatores de escala são adquiridos conforme condição apresentada na Tabela 11.

Tabela 11 Fatores de escala para modo implícito do WP.

Condição	Fatores de escala
Se (diferença entre o POC da <i>pic1</i> e o POC da <i>pic0</i> for igual a zero) OU (pelo menos uma das imagens <i>pic0</i> ou <i>pic1</i> estiver marcada na lista como "used for long-term reference") OU ($-64 > \text{DistScaleFactor}/4$) OU ($128 < \text{DistScaleFactor}/4$)	$w0 \leq 32$ $w1 \leq 32$
Caso contrário	$w0 \leq 64 - (\text{DistScaleFactor}/4)$ $w1 \leq \text{DistScaleFactor}/4$

O processo para o cálculo de escalonamento de amostras não é alterado conforme os tipos de codificação *frame* ou *field*. Portanto, o procedimento apresentado no item 2.4.3.2 é válido para a codificação MBAFF. Dessa forma, as amostras ponderadas podem ser obtidas através das equações (4), (5) e (6).

3.5 Arquiteturas MC

A seguir são citadas algumas arquiteturas de compensação de movimento encontradas na literatura. São realizadas breves descrições relacionadas à implementação da predição ponderada, suporte a vídeos entrelaçados e à codificação MBAFF.

Em (NA, 2007) é apresentado um decodificador H.264/AVC com otimização do consumo de potência, e que é destinado a aplicações móveis. Não há suporte às ferramentas do perfil main do H.264, e portanto o MC não possui predição ponderada e suporte a vídeos entrelaçados. Um compensador de movimento foi implementado por (WANG, 2005), mas da mesma forma, apenas estão disponíveis ferramentas do perfil *baseline*. Consequentemente este MC também não possui ponderação de amostras e não é capaz de tratar vídeos entrelaçados.

São apresentados modelos para a predição ponderada do H.264/AVC em (JILL, 2004) e (TSANG, 2010). Estes modelos visam a derivação dos parâmetros realizada pelo módulo de estimação de movimento (ME), ou seja, são aplicados somente à codificação de vídeo.

Em (TSAI, 2005), (CHEN, 2006) e (LI, 2007) são apresentadas otimizações da largura de banda dos dados processados pelo MC. No compensador de movimento do CODEC (CHEN, 2006) está previsto a ponderação de amostras, mas nos três casos não há suporte à entrelaçamento de vídeo, e consequentemente estes não estão habilitados a decodificar quadros MBAFF.

O decodificador apresentado em (LIN, 2007) possui um módulo de predição ponderada. Porém este trabalha somente com o modo explícito. Em (TANIZAWA, 2012) é proposto um novo método para a ponderação de amostras, em que o modo implícito é aplicado quando é utilizada somente uma das listas e, além dos slices B, a predição ponderada pode ser aplicada a slices P. Este método não atende a especificação da norma do padrão H.264/AVC (ITU, 2005) e, evidentemente, também não é aplicável ao SBTVD.

As arquiteturas MC em hardware apresentadas por (CHEN, 2008) e (SHEN, 2009) foram construídas visando a integração a um codificador e a um decodificador, respectivamente. Em ambas não é possível empregar vídeos entrelaçados.

Em (MATSUI, 2011) é descrita a arquitetura em hardware de um codificador com suporte a entrelaçamento, codificação de imagens *field* e codificação MBAFF. A implementação do módulo de estimação de movimento está descrita de forma detalhada. Porém, o módulo de compensação de movimento não está caracterizado.

Desta forma destaca-se que não foi encontrada na literatura atual uma arquitetura em hardware para a compensação de movimento do H.264/AVC com suporte à vídeo entrelaçado e a codificação MBAFF. A maioria das implementações de MC com suporte a estas ferramentas foram desenvolvidas em software, como em (YANMEI, 2004) e (SHAO, 2012).

3.5.1 Arquitetura de Referência

O Laboratório de Processamento de Sinais e Imagens (LaPSI) da Universidade Federal do Rio Grande do Sul (UFRGS) desenvolveu, em hardware, um decodificador para o padrão H.264/AVC que é capaz de decodificar bitstreams do perfil *baseline*, e possui suporte a algumas ferramentas de codificação do perfil *main*. Dentre estas ferramentas pode-se citar: a codificação aritmética adaptativa baseada em contexto (CABAC), a codificação MBAFF para macroblocos I, e a compensação de movimento com predição direta espacial e bi-predição (BONATTO, 2012). Para o completo suporte ao perfil *main* do H.264/AVC, o MC deste decodificador deve ainda contar com um módulo dedicado à ponderação de amostras, e estar apto a tratar vídeos entrelaçados, incluindo a codificação de imagens *field* e MBAFF.

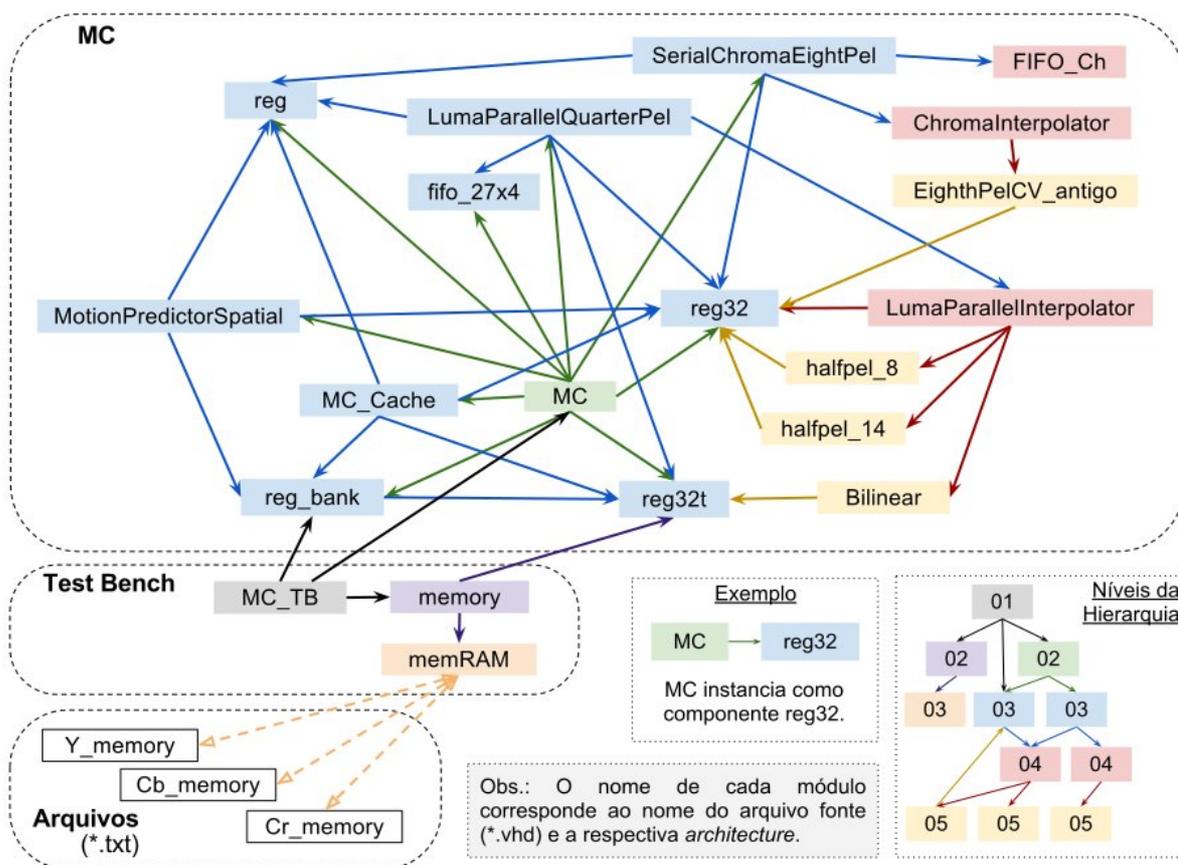


Figura 33 Estrutura Dos Arquivos Da Arquitetura MC E Test Bench.

Este compensador de movimento é capaz de decodificar vídeos de alta definição (HDTV) com formato de até 1920x1080, a 30 quadros por segundo. Neste módulo não está incluído o gerenciamento das listas de referência e as operações de alimentação e retirada de quadros do buffer de quadros de referência, esta tarefa é delegada ao controlador de memória. A descrição deste MC está distribuída em arquivos conforme Figura 33, totalizando cerca de 13.500 linhas de código VHDL. A arquitetura está ilustrada na Figura 34.

Como o algoritmo para predição de vetores de movimento se mostra extremamente sequencial, a descrição de predição de vetores de movimento do MC apresentado em (BONATTO, 2012) é baseada em máquinas de estados finitos (FSM). Há suporte para os

três modos de predição: padrão, direta espacial e direta temporal. Entre a predição direta espacial e a predição padrão há um compartilhamento de parte do hardware, relativo ao processo de derivação de vetores baseado nos macroblocos vizinhos. Neste preditor está previsto também o tratamento de codificação bi-preditiva.

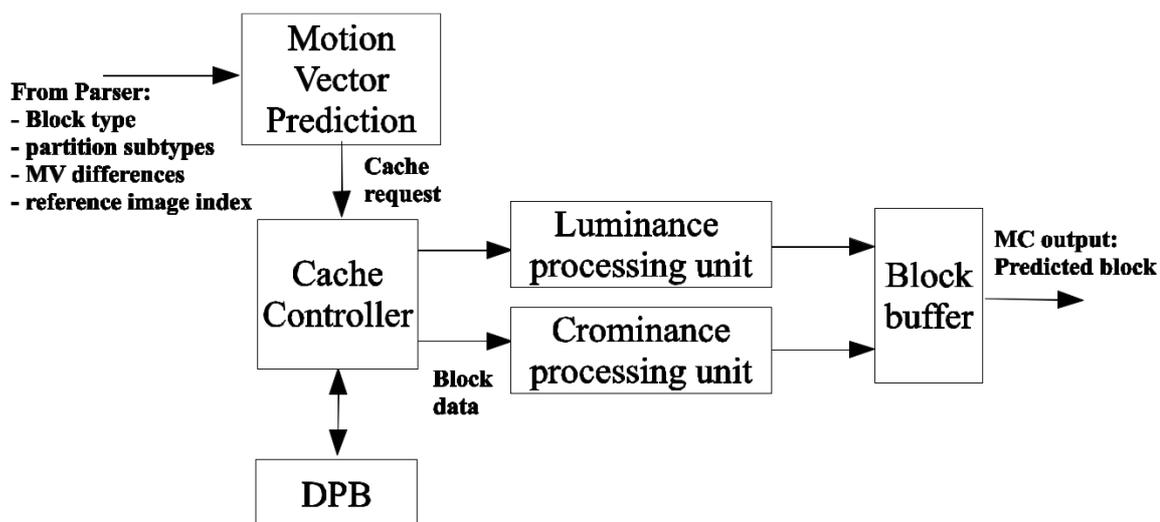


Figura 34 Arquitetura Do Compensador De Movimento (BONATTO, 2012).

O módulo de acesso à memória trabalha de forma conjunta com a memória cache, que serve como buffer para sincronização da arquitetura e como interface para a interpolação de amostras (vide *Cache Controller* da Figura 32). A cache é organizada em uma estrutura tridimensional, duas dimensões espaciais, horizontal e vertical, e uma dimensão para diferentes imagens. Existem 32 conjuntos de 16x40, linhas e colunas respectivamente, para as amostras de luminância, enquanto que para a croma são 32 conjuntos de 16 linhas e 20 colunas. As amostras na cache são armazenadas no formato adequado ao envio para o processamento das amostras.

O processamento de amostras se resume à interpolação para 1/4 e 1/8 de pixels e a média entre as amostras de referência para a bi-predição. Portanto não há suporte para a

predição ponderada. O interpolador é baseado em filtros unidimensionais e projetado para o processamento de blocos 4x4 Y, 2x2 Cb e 2x2 Cr.

Como cada módulo apresenta latências variáveis, no controle da arquitetura desse MC há um conjunto de máquinas de estados que sincronizam o fluxo de dados entre o preditor de vetores, a memória cache, o processamento das amostras e os buffers de saída. Na saída do preditor de vetores, buffers armazenam os vetores de movimento, índices de referência e flags de utilização das listas. Um controle de acesso à cache lê esses dados, e após a definição da região de referência, disponibiliza esta região ao módulo de processamento de amostras. Ainda há um buffer localizado depois do processamento de amostras, que visa serializar a saída do MC.

4 ARQUITETURA PROPOSTA

Este capítulo apresenta uma arquitetura em hardware para a predição ponderada (WP) do padrão de codificação de vídeo H.264/AVC. A arquitetura é capaz de tratar todas as possíveis configurações de codificação do perfil main, incluindo os diferentes tipos de composição de imagens e macroblocos aplicados a vídeos entrelaçados. Há suporte aos dois modos de predição ponderada: explícito e implícito.

Para o desenvolvimento deste trabalho foi utilizada como base a arquitetura de compensação de movimento do decodificador apresentado em (BONATTO, 2012), descrita como “Arquitetura de Referência” no capítulo 3. Como o interpolador deste MC trabalha com blocos 4x4, a arquitetura de predição ponderada (WP) proposta deve trabalhar também sempre com blocos 4x4 Y, 2x2 Cb e 2x2 Cr. Neste caso, em cada ciclo do processamento de amostras, o WP recebe 4 amostras de luminância e duas amostras de cada crominância, que são ponderadas em paralelo.

A arquitetura está descrita em dois módulos distintos: *scale factor prediction* (SFP) e *weighted samples prediction* (WSP), Figura 35. O SFP é responsável por encontrar os parâmetros necessários para o cálculo de escalonamento das amostras, e para isso estão disponíveis os dois modos de derivação, *explicit* e *implicit*. O cálculo de escalonamento é realizado pelo WSP. Os itens 4.1 e 4.2 expõem mais detalhes sobre cada um dos módulos.

Na arquitetura do MC o módulo “LumaParallelQuarterPel” faz o controle do processamento de amostras de luminância, que neste caso é basicamente o interpolador descrito no “LumaParallelInterpolador”, conforme Figura 33. Neste módulo de controle foi integrada parte do SFP (derivação no modo *explicit*) e a seleção dos modos *explicit* ou

implicit. Além disso o WSP, descrito no módulo “WeightSamplesPred”, está ligado ao “LumaParallelQuarterPel” para a ponderação das amostras de luminância, Figura 36.

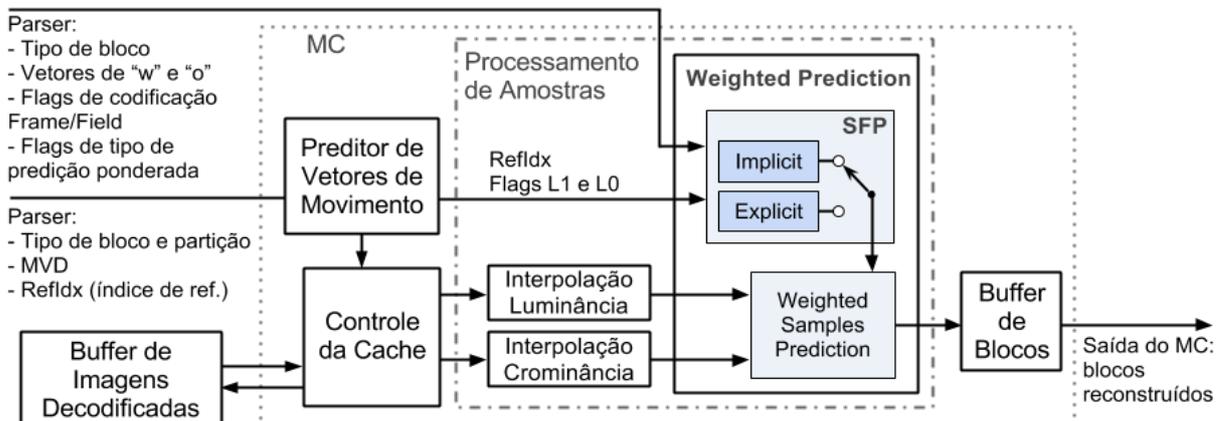


Figura 35 Arquitetura De Predição Ponderada Integrada Ao MC.

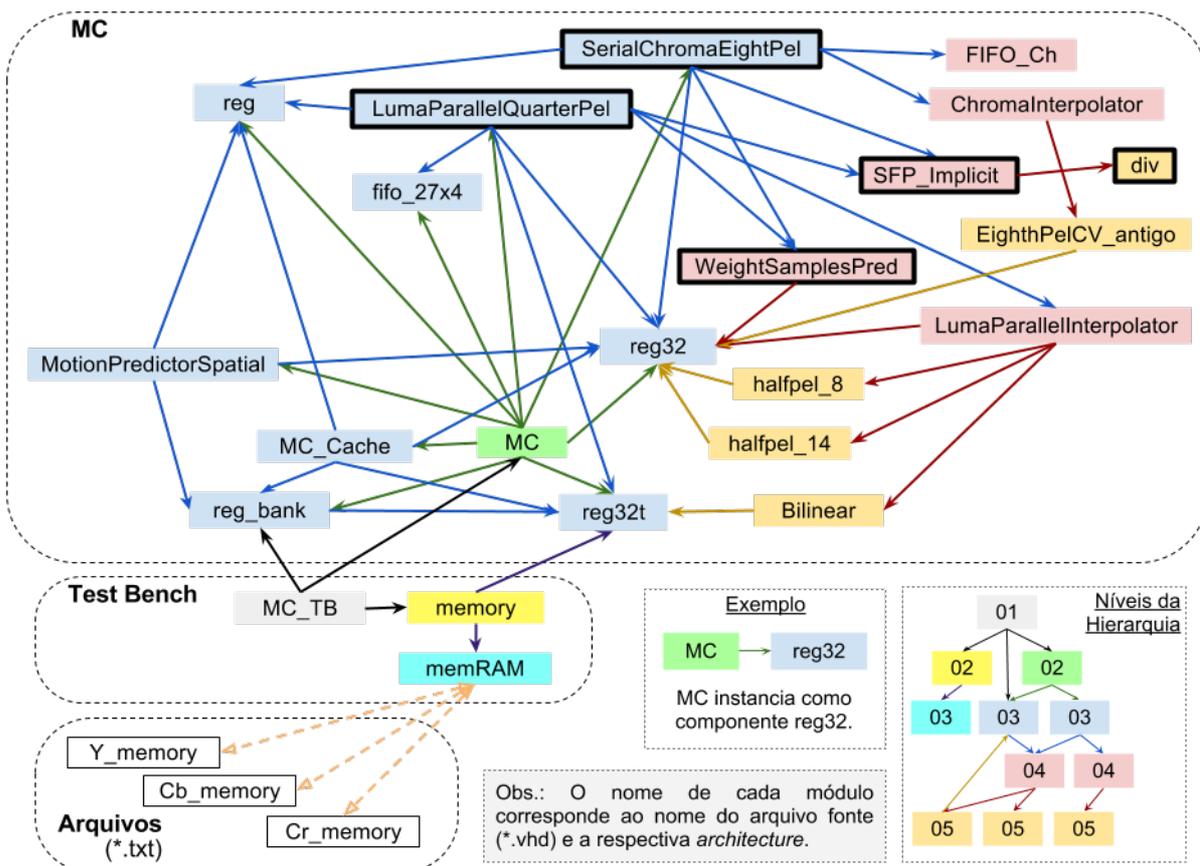


Figura 36 Estrutura Dos Arquivos Da Arquitetura Proposta Integrada Ao MC.

O mesmo processo foi realizado com o módulo de processamento de amostras de cromaticidade “SerialChromaEighPel”, como também pode ser visto na Figura 36. O processo de derivação no modo *implicit* é realizado em um módulo separado, chamado “SFP_Implicit”.

4.1 *Scale Factor Prediction*

Como as informações necessárias para a derivação dos parâmetros são provenientes do *parser* e do preditor de vetores (vide Figura 35) o *scale factor prediction* (SFP) trabalha em paralelo com a interpolação de amostras. Desta forma, quando as amostras estão disponíveis na saída do interpolador o WP está pronto para realizar o processo de escalonamento, *weighted samples prediction* (WSP).

4.1.1 *Modo Explicit*

Para a derivação dos parâmetros no modo *explicit* são recebidas as seguintes informações:

- Vetores de pesos e vetores de *offset* para cada componente de cor YCbCr e para cada lista, L0 e L1.
- Índices de referência das listas L0 e L1.
- Logaritmo de base 2 do denominador dos pesos e flags dos modos de codificação de vídeo entrelaçado.

O código abaixo mostra a interface do módulo “LumaParallelQuarterPel”, no qual está integrado o modo explícito do SFP. Os sinais com “#” são utilizados pelo SFP e os sinais com “&” são usados no WSP.

```

ENTITY LumaParallelQuarterPel IS
  GENERIC (  datawidth : natural := 8;
            Bff_Size  : natural := 24);
  PORT (
#    mb_adaptive_frame_field_flag      : IN std_logic;
#    mb_field_decoding_flag            : IN std_logic;
    BiPred                             : IN std_logic;
#    weighted_bipred_idc                : IN std_logic_vector(1 downto 0);
#    weighted_pred_flag                : IN std_logic;
#    WP_add_inf_In                      : IN std_logic_vector(44 downto 0);
&   WP_samples_In                     : IN std_logic_vector(31 downto 0);
#    RefIdxL0, RefIdxL1                : IN std_logic_vector(7 downto 0);
#    PredFlagL0, PredFlagL1           : IN std_logic;
#    POC, POC_L0, POC_L1              : IN std_logic_vector(datawidth -1 downto 0);
#    LongTermL0, LongTermL1           : IN std_logic;
    TopFieldOrderCnt                   : IN std_logic_vector(datawidth -1 downto 0);
    BottomFieldOrderCnt                 : IN std_logic_vector(datawidth -1 downto 0);
    InLine_0, InLine_1, InLine_2       : IN std_logic_vector(datawidth -1 downto 0);
    InLine_3, InLine_4, InLine_5       : IN std_logic_vector(datawidth -1 downto 0);
    InLine_6, InLine_7, InLine_8       : IN std_logic_vector(datawidth -1 downto 0);
    Img_type                            : IN std_logic;
    slice_type                          : IN std_logic_vector(2 downto 0);
    mb_type_In                          : IN std_logic_vector(5 downto 0);
    new_mbp                             : IN std_logic;
    dX_l0, dY_l0, dX_l1, dY_l1         : IN std_logic_vector(1 downto 0);
#    luma_weight_l0, luma_weight_l1    : IN ARRAY_256_BYTES;
#    luma_offset_l0, luma_offset_l1    : IN ARRAY_256_BYTES;
#    luma_log2_weigth_denom            : IN std_logic_vector(2 downto 0);
#    W_l0, o_l0, W_l1, o_l1           : IN std_logic_vector(datawidth -1 downto 0);
    LWd                                 : IN std_logic_vector(2 downto 0);
#&   WE, CLK, rst                     : IN std_logic;
#    ImgParReqL0, ImgParReqL0         : OUT std_logic;
#    RefIdxReqL0, RefIdxReqL1        : OUT std_logic_vector(7 downto 0);
    Bff_Full, validy, Avail             : OUT std_logic;
&   OutLine_0, OutLine_1             : OUT std_logic_vector(datawidth -1 downto 0)
&   OutLine_2, OutLine_3             : OUT std_logic_vector(datawidth -1 downto 0)
  );
END LumaParallelQuarterPel;

```

Neste modo os fatores de escala “w0” e “w1” são obtidos através dos vetores de pesos fornecidos pelo parser. A escolha do fator é realizada da seguinte forma:

- Se o MB for *field*, os índices de referência RefIdxL0 e RefIdxL1 são usados como índice de cada vetor.
- Caso contrário é utilizado a metade do valor de cada RefIdx como índice dos vetores.

Para determinar o tipo de codificação do MB atual são lidas das flags “mb_adaptive_frame_field_flag” e “mb_field_decoding_flag”. Quando as duas flags estão habilitadas o MB corrente é *field*, conforme Figura 37.

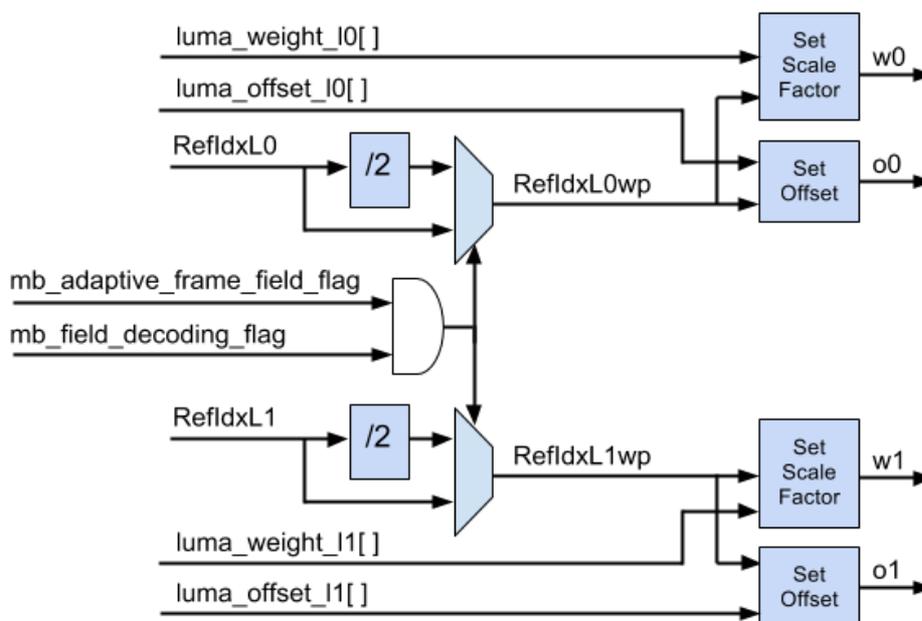


Figura 37 Modo Explicit Do Scale Factor Prediction (SFP).

Os ajustes aditivos “o0” e “o1” são obtidos com o mesmo procedimento dos fatores de escala. Neste caso a normatização do H.264/AVC prevê ainda um escalonamento dependente do número de bits usado para cada amostra de pixel. Porém, para as características do padrão utilizadas no SBTVD as amostras são sempre representadas por 8 bits, e neste caso o fator para este escalonamento é igual a 1.

Os valores de LogWD para cada componente de cor são atribuídos diretamente a partir dos valores de “luma_log2_weight_denom”, “chromaCb_log2_weight_denom” e “chromaCr_log2_weight_denom”.

4.1.2 Modo *Implicit*

O modo *implicit* do *scale factor prediction* esta descrito em um módulo chamado “SFP_implicit”, Figura 36. A interface deste módulo é apresentada abaixo.

```

ENTITY SFP_implicit IS
  PORT (
    mb_adaptive_frame_field_flag : IN std_logic;
    mb_field_decoding_flag       : IN std_logic;
    POC, POC0, POC1              : IN std_logic_vector(16 downto 0);
    RefIdxL0, RefIdxL1           : IN std_logic_vector(7 downto 0);
    LongTermL0, LongTermL1      : IN std_logic;
    CLK, rst, start              : IN std_logic;
    Valid                         : OUT std_logic;
    ImgParReqL0, ImgParReqL1     : OUT std_logic;
    RefIdxReqL0, RefIdxReqL1     : OUT std_logic_vector(7 downto 0);
    WeightL0, WeightL1          : OUT std_logic_vector(7 downto 0);
    OffsetL0, OffsetL1          : OUT std_logic_vector(7 downto 0);
    LogWD                         : OUT std_logic_vector(2 downto 0)
  );
END SFP_implicit;

```

No modo *implicit* os ajustes aditivos “o0” e “o1” são assumidos como sendo iguais a zero e o logaritmo LogWD é igual a 5. Os fatores de escala “w0” e “w1” são calculados a partir do valor do DistScaleFactor, que é derivado dos parâmetros “tb” e “td”, conforme item 2.5.3.3. Como o restante da arquitetura do MC não possui suporte a entrelaçamento, mais especificamente o MVP, não é possível utilizar os valores de “tb” e “td” calculados na predição direta temporal. O cálculo destes parâmetros é realizado conforme equações 12 e 13.

$$tb = POC(currPicOrField) - POC(pic0) \quad (12)$$

$$td = POC(pic1) - POC(pic0) \quad (13)$$

As imagens atual (currPicOrField), referência da lista 0 (pic0) e referência da lista 1 (pic1) são identificadas conforme o procedimento descrito no capítulo 3, item 3.4. A Figura

38 ilustra a arquitetura do módulo de previsão em modo implícito. O bloco “div” é um divisor de N bits que foi reutilizado da arquitetura de MC apresentada em (ZATT, 2006).

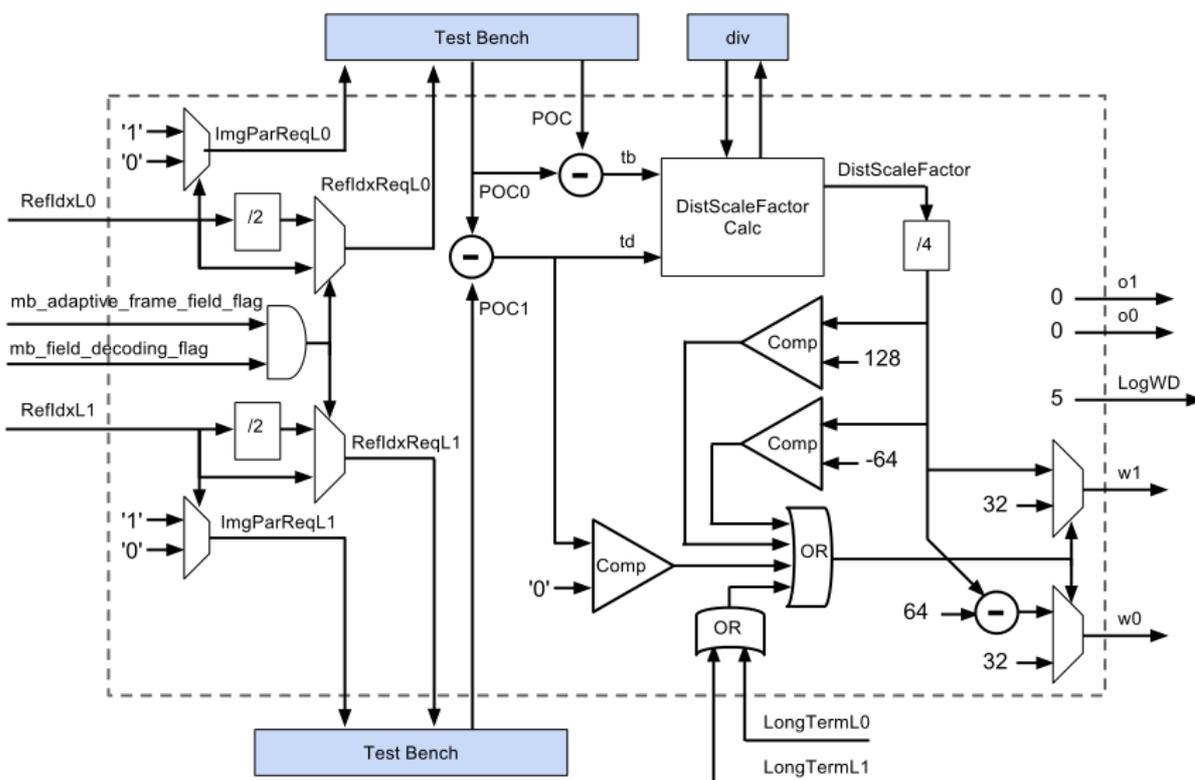


Figura 38 Modo *Implicit Do Scale Factor Prediction* (SFP).

4.2 *Weighted Samples Prediction*

No *Weighted Samples Prediction* (WSP) as amostras são ponderadas conforme os parâmetros obtidos no SFP, offset “o0” e “o1”, “LogWD” e os fatores de escala “w0” e “w1”. O processo é realizado conforme equações 4 e 5, e sua descrição está em um módulo chamado “WeighSamplesPred”. O código abaixo descreve a interface, e a Figura 39 ilustra a estrutura do módulo WSP.

```
ENTITY WeighSamplesPred IS
  PORT (
    SamplesIn      : IN std_logic_vector(7 downto 0);
    CLK, WE       : IN std_logic;
    W, o          : IN std_logic_vector(7 downto 0);
```

```

log2WD      : IN std_logic_vector(2 downto 0);
SamplesOut  : OUT std_logic_vector(16 downto 0)
);
END WeighSamplesPred;

```

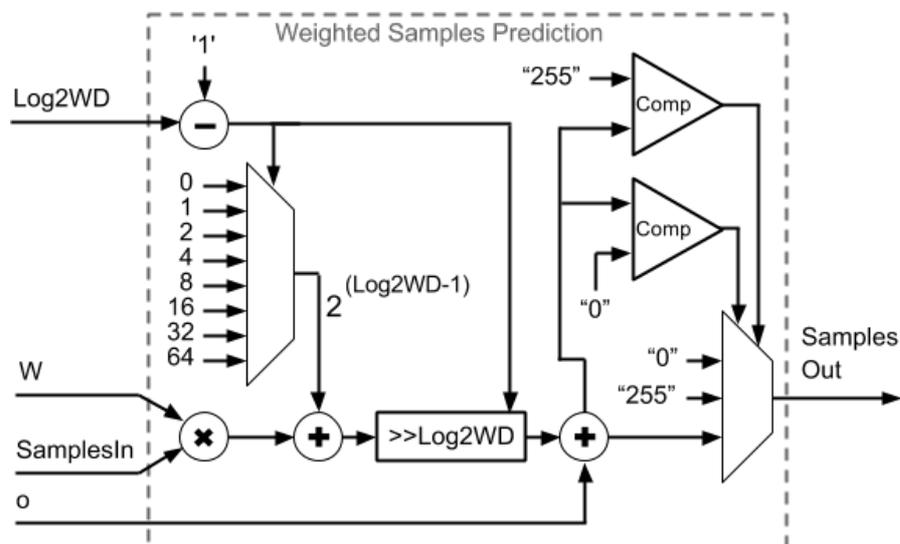


Figura 39 Módulo *Weighted Samples Prediction* (WSP).

No MC, no qual foi integrado os módulos do WP, há suporte para bi-predição. As amostras das regiões de referência de cada lista são processadas pela mesma estrutura de interpolação, primeiro amostras de uma lista e depois da outra. Estas amostras são armazenadas em registradores, com capacidade para um bloco 5x4, 5 ciclos de interpolação. A partir do 5º ciclo, estes registradores já possuem amostras das duas listas e então é realizada a média entre as mesmas. Este procedimento foi mantido e portanto a bi-predição não está sendo tratada pelo módulo de predição ponderada apresentado neste trabalho.

4.3 Resultados

O hardware foi descrito em VHDL no ambiente Xilinx ISE (XILINX, 2012) e as simulações da arquitetura foram realizadas no ambiente ModelSIM (MENTOR, 2012).

Como fonte de dados para as simulações foram utilizados arquivos de *test bench*. Inicialmente cada módulo foi simulado isoladamente, usando um *test bench* dedicado para o WSP e outro para o SFP. Desta forma foi validado o funcionamento de cada componente. Em uma segunda etapa, os módulos foram integrados a arquitetura de compensação de movimento (BONATTO, 2012). Para a combinação dos dois componentes, o WSP e o SFP (que formam a arquitetura de predição ponderada WP), foram realizadas simulações a partir de um *test bench* aplicado a toda a arquitetura do MC. Este *test bench* possui aproximadamente 1500 linhas de código VHDL. Para vídeos progressivos foram extraídas ainda informações do software de referência (JVT, 2009). Para vídeos entrelaçados foram usados somente vetores de dados descritos no *test bench*.

O módulo *Scale Factor Prediction* (SFP) foi simulado em duas etapas distintas, uma com foco no modo *explicit* e outra para o modo *implicit*. A Figura 40 apresenta a simulação da arquitetura SFP para um MB tipo P com predição ponderada em modo *explicit*. Nesta simulação é possível ver que o índice de referência RefIdxL0 é dividido (ver sinal interno RefIdxL0wp) a partir da transição do *mb_field_decoding_flag* para '1', que indica que o próximo MB é codificado em *field*.

Na Figura 41 está ilustrada a simulação do SFP para um MB tipo B com predição ponderada em modo *implicit*. Conforme citado anteriormente, os valores de “o0” e “o1” (offsetL0 e offsetL1 na Figura 41) são sempre iguais a zero e LogWD é sempre igual a 5. Os fatores de escala “w0” e “w1” (weightL0 e weightL1 na Figura 41) sofrem uma transição, respectivamente de 32 para 64 e 32 para 0, devido a diferença entre o POC das duas imagens de referência do MB subsequente não ser igual a zero.

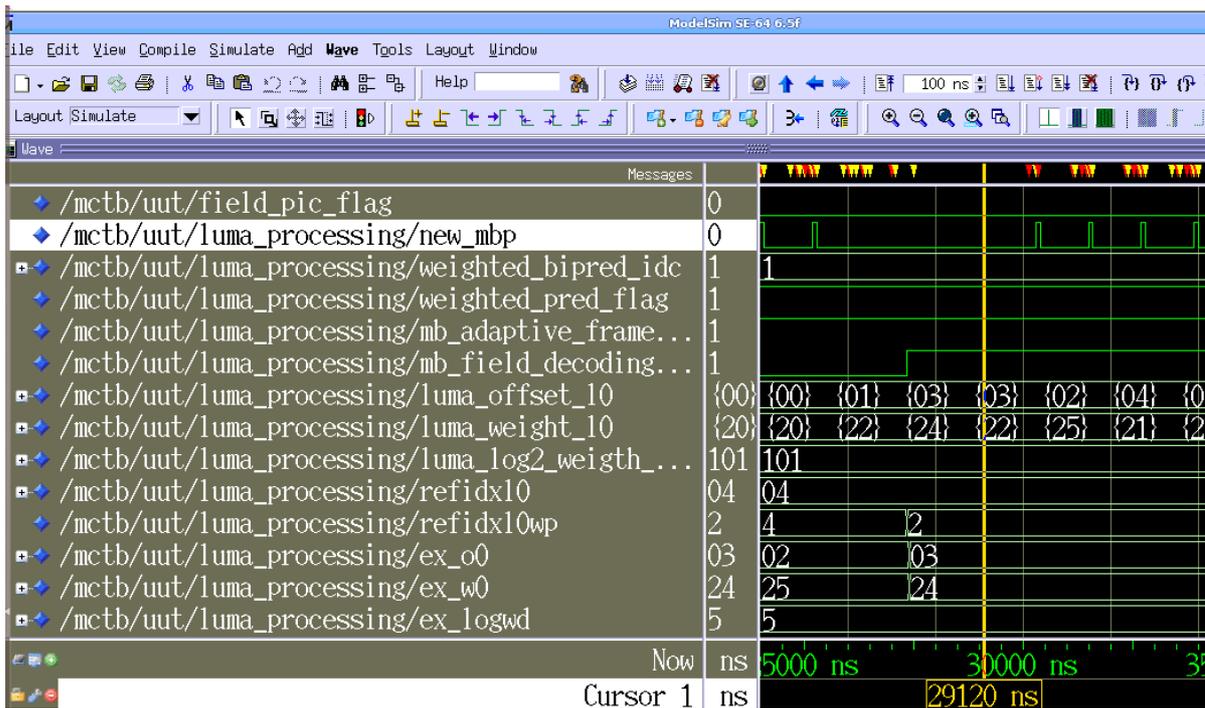


Figura 40 Simulação Do Módulo *Scale Factor Prediction* (SFP), Modo *Explicit*.

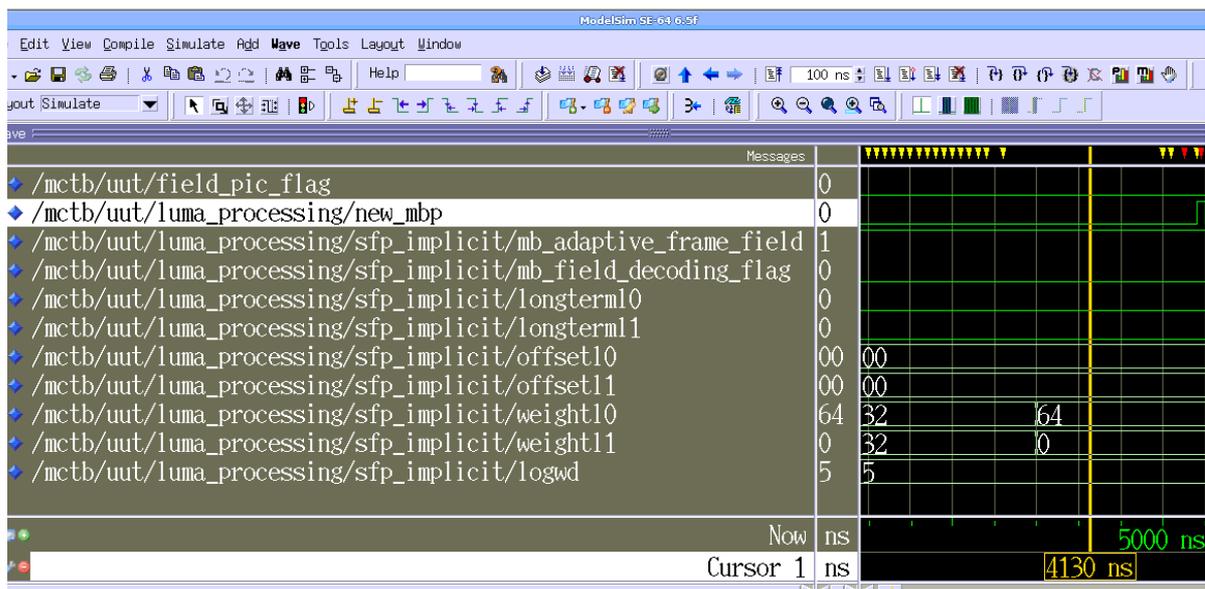


Figura 41 Simulação Do Módulo *Scale Factor Prediction* (SFP), Modo *Implicit*.

Uma simulação para o módulo *Weighted Samples Prediction* (WSP) é apresentada na Figura 42. Como este módulo é utilizado tanto para amostras da lista 0 quanto para amostras da lista 1, os parâmetros de entrada estão identificados simplesmente como

“log2wd”, “o”, e “w”. O “SamplesIn” é a entrada de amostras e o “SamplesOut” é a saída de amostras previstas.

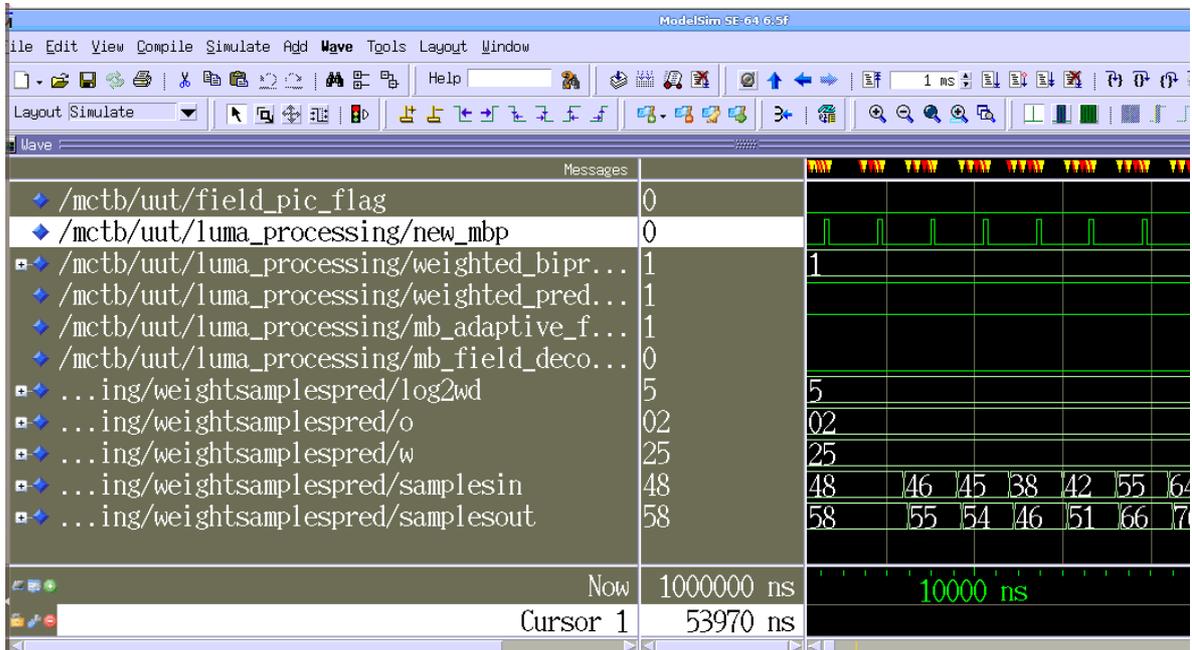


Figura 42 Simulação Do Módulo *Weighted Samples Prediction* (WSP).

5 CONCLUSÃO

Este documento apresentou um estudo da decodificação Inter-quadro, o qual pode ser utilizado como referência à implementações do módulo MC, e uma modelagem em hardware para predição ponderada da compensação de movimento do H.264/AVC. Neste estudo foram abordadas todas as ferramentas e modos de decodificação previstos pela normatização deste padrão (ITU, 2005) que atendem as especificações do SBTVD (ABNT, 2008).

A grande maioria dos trabalhos encontrados na literatura que envolvem a compensação de movimento do H.264/AVC, sequer abordam o tratamento de vídeos entrelaçados, ou ainda a codificação MBAFF. Há muitos trabalhos que se propõem a realizar otimizações em arquiteturas ou parte de arquiteturas de MC. Porém, não foi encontrada uma arquitetura para compensação de movimento com suporte a todas ferramentas de codificação do perfil *main* do H.264/AVC.

O preditor de vetores de movimento do H.264/AVC teve um acréscimo significativo de complexidade comparado ao seu antecessor o MPEG-2, apesar da simplificação realizada na predição direta temporal. O módulo de processamento de amostras também se tornou mais complexo, por exemplo, o uso de vetores fracionários na interpolação de amostras faz com que este seja o processo crítico da compensação de movimento. Apesar dos inúmeros avanços apresentados por este padrão, foram identificadas algumas deficiências. Por exemplo na predição ponderada o fator de escala somente é adaptativo por MB para o modo *implicit*, ou seja, somente para a bi-predição. Desta forma, quando e utilizada a predição através de apenas uma das listas não é possível aplicar um fator de

escala diferente para algumas regiões da imagem, o que poderia significar uma redução da magnitude dos resíduos.

A modelagem da arquitetura de ponderação de amostras proposta alcançou os três principais objetivos, chegar a uma implementação funcional, tratar todas as possibilidades de decodificação previstas pelo H.264/AVC para este módulo e manter a taxa de decodificação de quadros da arquitetura de MC na qual foi integrada. Como o processo de interpolação de amostras não sofre influência dos possíveis tipos de composição de campos, pode-se dizer que toda arquitetura de processamento de amostras está apta a decodificar vídeos entrelaçados com qualquer tipo de codificação. Um artigo que apresenta a arquitetura construída foi descrito, conforme Apêndice A.

Como trabalhos futuros, pode ser implementado o suporte aos diversos tipos de composição de campos para os módulos de predição de vetores e acesso a memória, da arquitetura usada como base neste trabalho (BONATTO, 2012). No preditor de vetores este aperfeiçoamento deve dar atenção especial à localização dos vizinhos espaciais (“A”, “B”, “C” e “D”) e do bloco co-localizado. No acesso à memória deve-se principalmente criar um processo de reconstrução das referências para os casos em que o MB atual é *field* e a referência é *frame*, ou vice-versa. Ainda, neste MC, podem ser realizadas aprimoramentos em cada módulo. Por exemplo, o interpolador de amostras processa as amostras na bi-predição de forma serial. Se estas amostras forem interpoladas em paralelo a arquitetura pode contribuir para a redução do *delay* de decodificação, ou simplesmente reduzir a frequência de trabalho do chip.

REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 15602-1**: Televisão Digital Terrestre – Codificação de Vídeo, Áudio e Multiplexação. Parte 1: Codificação de Vídeo. Rio de Janeiro, 2007.

AZEVEDO FILHO, A. P. **MoCHA**: Arquitetura Dedicada para a Compensação de Movimento em Decodificadores de Vídeo de Alta Definição, Seguindo o Padrão H.264. 2006. 120 f. Dissertação (Mestrado em Engenharia) – Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2006.

BONATTO, A. C. et al. Hardware Decoding Architecture for H.264/AVC Digital Video Standard. In: M. Akshaya; N. Zafar; S. Zafar (Org.).. **An Introductory Guide to Image and Video Processing**. 1ed. Hong Kong: iConcept Press, 2012. p. 1-22.

CHEN, Y. H. et al. Efficient Architecture Design of Motion-Compensated Temporal Filtering/Motion Compensated Prediction Engine. **IEEE Transactions on Circuits and Systems for Video Technology**, [S. l.], v. 18, n. 1, p. 98 - 109, Jan. 2008.

CHEN, T. et al. Hardware Architecture Design of an H.264/AVC Video Codec. In: ASIA AND SOUTH PACIFIC DESIGN AUTOMATION CONFERENCE (ASP-DAC), 11., 2006. Yokohama. **Proceedings...** Piscataway: IEEE, 2006. p. 750-757.

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.264 (03/05)**: advanced video coding for generic audiovisual services. [S.l.], 2005.

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

JILL, M. B. Weighted prediction in the H.264/MPEG AVC video coding standard. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS). 2004, Vancouver. **Proceedings...** [S.l.]: IEEE, 2004. p. 789-792.

JOINT VIDEO TEAM. **JM, versão 18.0**. Disponível em: <<http://iphome.hhi.de/suehring/tml/>>. Acesso em: 10 dez. 2011.

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. Sardinia. **Proceedings...** [S.l.:s.n.], 2005. p. 4.

KLEIN, H. A. **Implantação do Suporte a Vídeo Entrelaçado no Módulo de Predição Intra-quadros para o SBTVD**. 2012. 66 f. Trabalho de Conclusão de Curso (Engenharia de Computação - Graduação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.

LI, Y. et al. Memory Cache Based Motion Compensation Architecture for HDTV H.264/AVC Decoder. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, 2007. New Orleans. **Proceedings...** New Orleans: IEEE, 2007. p. 2906-2909.

LIN, C. C. 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.

MALVAR, H. et al. Low-Complexity Transform and Quantization in H.264/AVC. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v. 13, n. 7, p. 598-603, Jul. 2003.

MATSUI, H. et al. An H.264 Full HD 60i Double Speed Encoder IP Supporting Both MBAFF and Field-Pic Structure. In: International Symposium on VLSI Design, Automation and Test (VLSI-DAT), 2011. Hsinchu. **Proceedings...** Piscataway: IEEE, 2011. p. 1-4.

MENTOR GRAPHICS. **ModelSim, versão 6.6a**. 2010. Disponível em: <<http://www.model.com>>. Acesso em: 20 mar. 2012.

MIANO, J. **Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP**. New York: Addison-Wesley, 1999.

NA, S. et al. 1.8mW, Hybrid-Pipelined H.264/AVC Decoder For Mobile Devices. In: IEEE INTERNATIONAL ASIAN SOLID-STATE CIRCUITS CONFERENCE (A-SSCC), 2007. Jeju. **Proceedings...** New York City: IEEE, 2007. p. 192-195.

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. **H.264 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 Frase University, Burnaby, 2005.

SHAO, L.; YUN, H. MBAFFF, An Adaptive Interpolation To AVS Video. In: IEEE INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING COMMUNICATION AND COMPUTING (ICSPCC), 2012, Hong Kong. **Proceedings...** Piscataway: IEEE, 2012. p. 341 - 344.

SHEN, D. Y.; TSAI, T. H. A 4X4-Block Level Pipeline and Bandwidth Optimized Motion Compensation Hardware Design for H.264/AVC Decoder. In: IEEE INTERNATIONAL CONFERENCE ON MULTIMEDIA AND EXPO (ICME), 2009, New York. **Proceedings...** Piscataway: IEEE, 2009. p. 1106-1109.

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, 27., 2004, Denver. **Proceedings...** [S.l.:s.n.], 2004. pages 1-21.

TANIZAWA, A. Multi-Directional Implicit Weighted Prediction Based on Image Characteristics of Reference Pictures for Inter Coding. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 2012, Orlando. **Proceedings...** [S.l.]: IEEE, 2012. p. 1545-1548.

TSAI, C. Y.; et al. Bandwidth Optimized Motion Compensation Hardware Design for H.264/AVC HDTV Decoder. In: 48TH MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS, 2005, Covington. **Proceedings...** Piscataway: IEEE, 2005. p. 1199-1202.

TSANG, S.-H.; CHAN, Y.-L. H.264 Video Coding With Multiple Weighted Prediction Models. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 2010, Hong Kong. **Proceedings...** [S.l.]: IEEE, 2010. p. 2069-2072.

WANG, S.-Z. et al. A New Motion Compensation Design for H.264/AVC Decoder. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS), 2005, Kobe. **Proceedings...** [S.l.]: IEEE, 2005. p.4558-4561.

YANMEI, Q.; GUOPING L.; A Fast MBAFF Mode Prediction Strategy for H.264/AVC. In: INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING (ICSP), 2., 2004, Beijing. **Proceedings...** [S.l.]: IEEE, 2004. p. 1195 - 1198.

YOO, K. et al. Hardware Design of Motion Data Decoding Process for H.264/AVC. **Elsevier Signal Processing: Image Communication**, [S.l.], n. 25, p.208–223, 2010.

XILINX INC. **ISE Project Navigator, versão 10.1.** Disponível em: <<http://www.xilinx.com/products/design-tools/ise-design-suite/index.htm>>. Acesso em: 27 maio 2012.

ZATT, B. **Modelagem de Hardware para Codificação de Vídeo e Arquitetura de Compensação de Movimento Segundo o Padrão H.264/AVC.** 2008. 121 f. Dissertação (Mestrado em Engenharia) – Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2008.

ZATT, B. **Preditor de Vetores de Movimento em Hardware Segundo o Padrão H.264/AVC.** 2006. 73 f. Trabalho de Conclusão de Curso (Engenharia de Computação - Graduação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2006.

APÊNDICE A: Artigo

Arquitetura em Hardware de Predição Ponderada da Compensação de Movimento do
Padrão H.264/AVC

Hardware Implementation of the H.264-AVC Weighted Prediction Motion Compensation

Jonas S. Silva, André B. Soares, Altamiro A. Susin

Electrical Engineering Department
Federal University of Rio Grande do Sul
Porto Alegre, Brazil

E-mail: {jonas.santos, andre.borin, altamiro.susin}@ufrgs.br

Leticia V. Guimarães

Digital Systems Engineering
State University of Rio Grande do Sul
Porto Alegre, Brazil

E-mail: leticia-guimaraes@uergs.edu.br

Abstract – Motion compensation is a powerful compression tool of the H.264 video encoding standard. The many available features of the standard are freely chosen by the encoder. The decoder, however, must be able to decode all of them. The most common shortcoming is related to the interlaced video and the MBAFF (Macro-block Adaptive Frame-Field) fields composition. The reason is the complexity of the algorithm and the corresponding hardware needed to the implementation. This paper proposes a hardware architecture for weighted prediction of motion compensation with support for all H.264/AVC main profile tools, including interlaced video encoded with MBAFF. The Weighted Prediction (WP) was decomposed in two main processes, the Scale Factor Prediction (SFP) and the Weighted Samples Prediction (WSP), and are presented in this paper.

I. INTRODUCTION

On a video capture each individual picture can be acquired by progressive or interlaced scanning. In the video coding standard H.264/AVC [1] fields of an interlaced picture can be encoded in frame mode (top and bottom fields interlaced) or field mode (top and bottom fields coded separately). When the choice between frame mode and field mode is adaptive for each macro-block pair the coding is called Macro-block Adaptive Frame-Field (MBAFF). The MBAFF coded video imposes more computational complexity to all H.264 decoder modules specially on Motion Compensation (MC) module that is responsible to reconstruct the blocks based on previously decoded ones.

The MC can be divided into motion vectors prediction and samples processing. In the sample processing two tasks are performed: samples interpolation and weighting. The weighted prediction (WP) module uses scale factors to weight the samples before MC output. This is particularly useful in video fades.

This paper presents a hardware architecture for the weighted samples prediction of MC. It is in compliance with the main profile of H.264/AVC standard. Therefore it can decode field and MBAFF pictures. For this purpose two modules were built: Scale Factor Prediction (SFP) and Weighted Samples Prediction (WSP). The SFP aims to predict the scale factors while WSP calculates weighted samples. The architecture was designed to upgrade the motion compensator designed for the decoder presented in [2], which was designed for Brazilian digital television system, SBTVD [3].

II. RELATED WORKS

The MC hardware architectures presented in [4][5][6][7] and [8] are not able to decode interlaced video. The weighted prediction module in decoder presented by [9] works only with explicitly mode. In [10] is proposed a new method for implicit mode of weighted prediction which is not in accordance with the H.264/AVC. The Matsui describes in [11] an architecture of a hardware encoder that supports interlacing, however the motion compensation module is not featured. We did not find any similar hardware solution in the literature.

III. THE H.264 STANDARD

The H.264/AVC standard is divided into profiles that define the use of particular groups of encoding functions. This work is directed to the main profile that uses the interlaced video coding MBAFF and weighted prediction in motion compensation.

A. Motion Compensation

The motion compensation (MC) module uses motion vectors (MV) and indexes to identify a region of samples in a previously decoded image. The samples of current macro-block (MB) are inferred from samples of this region. The MB can be divided into 16x8, 8x16, 8x8, 8x4, 4x8 or 4x4 blocks. Motion vectors are accurate to 1/4 for luma samples and 1/8 for chroma, and may point out of the frame. The MC can use bi-prediction and weighted prediction [1].

B. Macro-block Adaptive Frame-Field

A frame in a sequence of interlaced video consists of the combination of top and bottom fields acquired at different times. These fields can be coded in frame or field mode, as Fig 1. For the MBAFF the choice of frame or field is adaptive for each pair of macro-blocks, as shown in Fig. 2. Thus it is possible to encode each frame region in a different mode. For example, it can consider the correlation between the fields. In order to increase the compression rate, using MBAFF the order coding change from traditional macro-block raster scan to macro-blocks pairs raster scan.

C. Weighted Prediction

The weighted prediction (WP) process is a scaling applied to the samples at the output of MC decoding process. This is very useful when there are fades in the video sequence. The WP can be divided into two distinct steps: *scale factor prediction* (SFP) and *weighted samples*

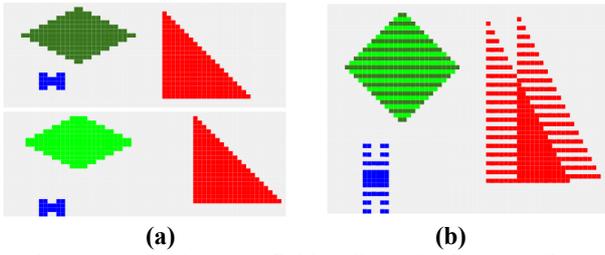


Figure 1. Interlace: (a) field coding, (b) frame coding.

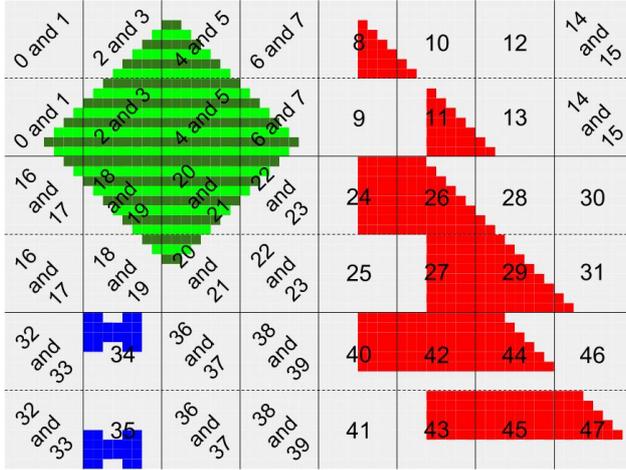


Figure 2. Example of a MBAFF picture structure.

prediction (WSP). The WSP calculates the new samples values "pred(i,j)" by

$$\text{pred}(i, j) = ((\text{pred}(i, j) * w + 2^{\{\log WD - 1\}}) \gg \log WD) + o \quad (1)$$

The SFP predict "w", "o" and "logWD". These parameters are associated to each reference frame, each new slice and for each color component. The definition depends on the WP coding type of current MB, default, implicit or explicit. In default mode the samples remain unaffected. The parameters are received from the encoder on explicit mode. For implicit mode "o" and "logWD" are set to a fixed value and the scaling factor "w" is derived based on the "tb" and "td". These represent the temporal distance between the current frame and the reference frames, as shown Fig. 3.

IV. WEIGHTED PREDICTION ARCHITECTURE

This section presents an weighted prediction hardware architecture for H.264/AVC standard. This proposed WP architecture is integrated into the motion compensation module of decoder presented by [2], as shown in Fig. 4. The architecture is able to handle all possible main profile encoding settings that includes MBAFF coding for interlaced video.

In H.264/AVC decoder, the parser module extracts from the bitstream all required parameters for the decoding modules. The parser and the Motion Vector Prediction supply the inputs to the SFP, as shown in Fig. 4. The samples interpolation process and the SFP are performed in parallel. Hence, all parameters are available at the end of luminance interpolation and chrominance interpolation process.

The WSP input samples are provided by the interpolation module. Since this MC interpolation works on 4x4 blocks and it uses 4:2:0 sampling, the WSP receives four luma samples in each processing cycle and four chroma samples in each two processing cycles. Color components are weighted in parallel.

A. Scale Factor Prediction

The SFP predicts three parameters: the base 2 logarithm of the denominator for all luma weighting factors "logWD", the additive offset "o" and the scale factor "w". This

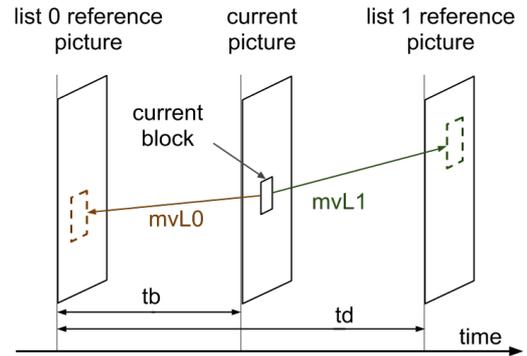


Figure 3. Temporal distance between frames, tb and td.

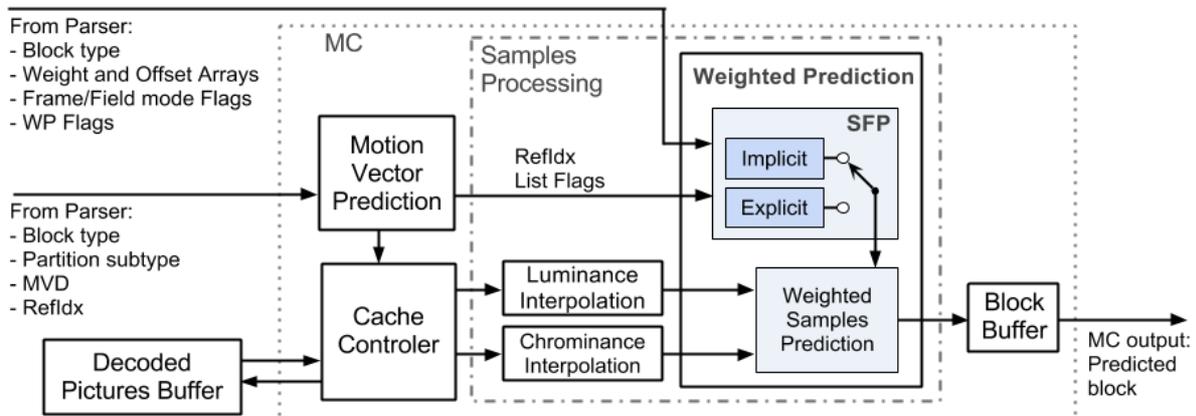


Figure 4. Weighted Prediction integrated into MC architecture [2].

module receives from parser one flag called "weighted_pred" and a variable called "weighted_bipred_idc" to define the derivation type, as shown Table I.

On explicit mode the LogWD values for each color component are received from the parser. The scale factor "w" and offset "o" are obtained through the arrays also provided by the parser. The choice of the factor and the offset are performed as follows:

- If the current MB is a field, the reference index RefIdx is used as the index of each array.
- Otherwise, it is used half the value of each reference index RefIdx.

This is performed for each reference list, Fig 5 show this process architecture for list 0. The scale factor "w" applied to list 0 is named "w0" and it is "w1" for list 1. The same applies to offset "o", "o0" for list 0 and "o1" for list1.

For implicit mode, the offsets "o0" and "o1" are assigned with 0 and the parameter "logWD" is set to 5. The scale factors are estimated based on the "DistScaleFactor" parameter. It is given by

$$\text{DistScaleFactor} = (32 + \text{tb} * (16384 + \text{Abs}(\text{td}/2)) / \text{td}) / 64 \quad (2)$$

where "Abs" is the absolute value operation. The parameters "td" and "tb" are given by

$$\text{tb} = \text{POC}(\text{currPic}) - \text{POC}(\text{pic0}) \quad (3)$$

$$\text{td} = \text{POC}(\text{pic1}) - \text{POC}(\text{pic0}) \quad (4)$$

The Picture Order Count (POC) is computed out of the WP architecture and the picture "currPic" is determined as follows:

- If the current picture "currPic" is not coded in field mode and the current MB is encoded as field, the "currPic" is the field of the current picture that has the same parity (top or bottom) as the current MB.
- Otherwise, "currPic" is the current frame.

The list 0 reference picture "pic0" and the list 1 reference picture "pic1" can be acquired as described in Table II. The scale factors "w0" and "w1" are acquired by following condition:

- If $(\text{POC}(\text{pic1}) - \text{POC}(\text{pic0}) = 0)$ or (at least one of pic0 and pic1 is "used for long-term reference") or $((\text{DistScaleFactor} / 4) < -64)$ or $((\text{DistScaleFactor} / 4) > 128)$, then $w_0 = 32$ and $w_1 = 32$.

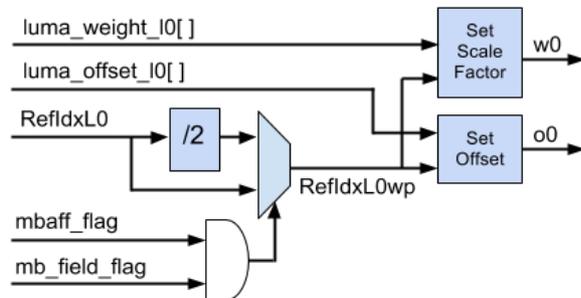


Figure 5. Scale factor prediction explicit mode architecture.

- Otherwise, $w_1 = \text{DistScaleFactor}/4$ and $w_0 = 64 - w_1$.

The pictures "pic0" and "pic1" are identified by calculated indexes "RefIdxReqL0" and "RefIdxReqL1", as Fig. 6.

B. Weighted Samples Prediction

In the Weighted Samples Prediction module the samples are weighted according to the parameters obtained in the SFP. These are offsets "o0" and "o1", "LogWD" and the scale factors "w0" and "w1". The weighted samples "pred(i,j)" are calculated by equation 1. For bi-prediction this is applied for samples from each list. Fig. 7 shows the architecture of the WSP process.

V. VALIDATION

The WP hardware architecture was described in VHDL on Xilinx ISE design suite [12]. The modules were independent and integrated simulated. The simulations have been run into ModelSim software environment [13]. The input data for progressive videos was extracted from reference software JM 18.0 [14]. For interlaced video were used data vectors described in a VHDL test bench file. These data were based on informations provided by CodecVisa 4.21 analyzer [15]. The simulations shown that the integration of Weighted Prediction in MC architecture did not change the MC processing cycles. The synthesis results for the Virtex 5 "XC5VLX110T" [16] are shown in Table III. Both architectures performed to 98,6 MHz.

TABLE I. WP DECODING MODES.

Slice	weighted_pred	weighted_bipred_idc	Reference List	WP Mode
P	0	-	-	Default
	1	-	L0	Explicit
B	-	0	-	Default
	-	1	-	Explicit
	-	2	L0 or L1	Default
	-		L0 and L1	Implicit

TABLE II. RECOGNITION OF REFERENCE PICTURE.

Current Picture	Current MB	Reference Pictures pic0 or pic1
Field	Field	The field of RefPicList[RefIdx]
Frame or MBAFF	Frame	The frame RefPicList[RefIdx]
MBAFF	Field	The field of frame RefPicList[RefIdx/2] that has the same parity as the current MB (for RefIdx%2 = 0)
		The field of frame RefPicList[RefIdx/2] that has opposite parity of the current MB (for RefIdx%2 = 1)

TABLE III. SYNTHESIS RESULTS.

	Slice Registers	Slice LUTs	Block RAM (36Kb)
MC [2]	9415	11266	13
Weighted Prediction integrated in MC	9708	11731	13

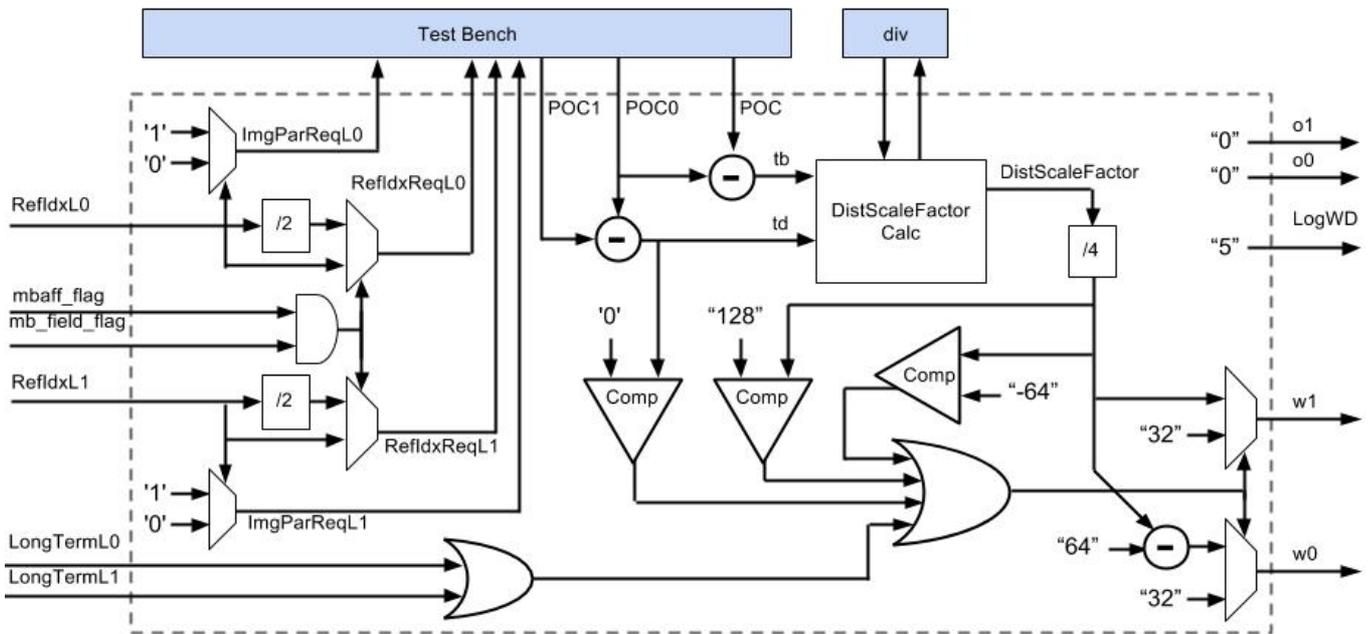


Figure 6. Scale factor prediction implicit mode architecture.

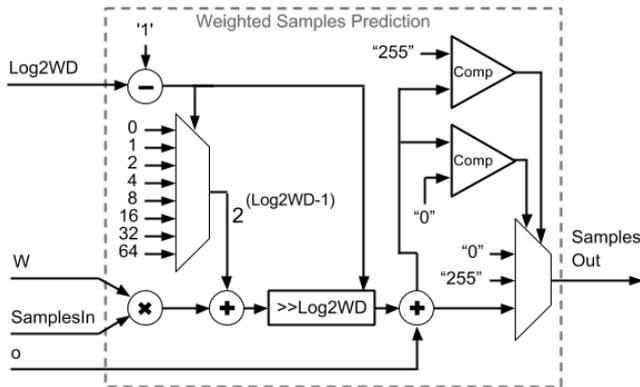


Figure 7. Weighted Samples Prediction Architecture.

VI. CONCLUSION

This paper presented a weighted prediction hardware architecture for the H.264/AVC main profile. Thereby it is able to decode interlaced video with any coding type. It can be field, frame or MBAFF coding. We did not find any similar hardware solution in the literature.

The modules were independent and integrated validated. The integration of the WP architecture in motion compensation architecture [2] did not change the rate of this MC operation.

REFERENCES

- [1] ITU-T Recommendation H.264 – Advanced video coding for generic audiovisual services, Video Coding Experts Group, Mar. 2005.
- [2] A. Bonatto, et al, “Hardware Decoding Architecture for H.264/AVC Digital Video Standard,” An Introductory Guide to Image and Video Processing. 1ed. Hong Kong: iConcept Press, 2012.
- [3] ABNT 15602-1:2007 Televisão Digital Terrestre – Codificação de Vídeo, Áudio e Multiplexação. Parte 1: Codificação de vídeo. Rio de Janeiro, Brasil, 2007.
- [4] C. Y. Tsai, T. C. Chen, T. W. Chen, and L. G. Chen, “Bandwidth optimized motion compensation hardware design for H.264/AVC HDTV decode,” 48th Midwest Symposium on Circuits and Systems, 2005.
- [5] T. Chen, C. J. Lian, and L. G. Chen, “Hardware Architecture Design of an H.264/AVC Video Codec,” Asia and South Pacific Design Automation Conference, pp. 750-757, 2006.
- [6] Y. Li, Y. Qu, Y. He, “Memory Cache Based Motion Compensation Architecture for HDTV H.264/AVC Decode,” IEEE International Symposium on Circuits and Systems, pp. 2906-2909, 2007.
- [7] Y. H. Chen, et al, “Efficient architecture design of motion-compensated temporal filtering/motion compensated prediction engine,” IEEE Transactions on Circuits and Systems for Video Technology, v. 18, n. 1, pp. 98 - 109, 2008.
- [8] D. Y. Shen, T. H. Tsai, “A 4x4-block level pipeline and bandwidth optimized motion compensation hardware design for H.264/AVC decoder,” IEEE International Conference on Multimedia and Expo, pp. 1106-1109, 2009.
- [9] C. C. Lin, et al, “A 160k gates/4.5 KB SRAM H.264 video decoder for HDTV applications,” IEEE Journal of Solid-State Circuits, v. 42, n. 1, pp. 170-182, 2007.
- [10] A. Tanizawa, “A multi-directional implicit weighted prediction based on image characteristics of reference pictures for inter coding,” IEEE International Conference on Image Processing, pp. 1545-1548, 2012.
- [11] H. Matsui, et al, “An H.264 Full HD 60i double speed encoder IP supporting both MBAFF and field-pic structure,” International Symposium on VLSI Design, Automation and Test, pp. 1-4, 2011.
- [12] <http://www.xilinx.com/products/design-tools/ise-design-suite>. Accessed 10 August 2012.
- [13] <http://www.mentor.com/products/fv/modelsim>. Accessed 15 September 2012.
- [14] <http://iphome.hhi.de/suehring/tml>. Accessed 20 June 2012.
- [15] <http://www.codecian.com/>. Accessed 12 June 2013.
- [16] http://www.xilinx.com/support/index.html/content/xilinx/en/supportNav/silicon_devices/fpga/virtex-5. Accessed 12 August 2012.

APÊNDICE B: Estudo da Predição Inter-quadro

Estudo da Predição Inter-quadro da Norma ITU-T Recommendation
H.264 (03/2005)

