5
Algoritmo UMHS Adaptativo ao Movimento

5.1
Introdução

Neste capítulo descrevemos o algoritmo proposto nesta dissertação de Mestrado, o qual foi nomeado como AUMHS (*Adaptive Unsymmetrical-cross Multi-Hexagon-grid Search*). Este algoritmo traz modificações ao algoritmo original (UMHexagonS), no sentido de atuar nas seqüências de vídeo explorando a correlação de movimento em macroblocos espacialmente vizinhos e possibilitando o ajuste do número de posições de busca de acordo com o movimento detectado.

O algoritmo AUMHexagonS traz como principais introduções:

- Utilização da medida de movimento do algoritmo AFSA;
- Alteração dos parâmetros do UMHexagonS, de acordo com esta medida de movimento;
- Alteração dos parâmetros do codificador H.264/AVC, de acordo com esta medida de movimento;

Os resultados alcançados mostram que ainda é possível reduzir a complexidade dos codificadores de vídeo, tanto com relação a complexidade inerente aos algoritmos de estimação de movimento, quanto com relação à utilização otimizada de outros parâmetros de codificação. Apesar da significativa redução de complexidade observada com o AUMHexagonS, não foi observada nenhuma redução de qualidade das imagens reconstruídas, tanto em termos subjetivos quanto objetivos.

A seguir, apresentamos o algoritmo AUMHexagonS. Primeiramente descrevemos a implementação da medida de movimento do algoritmo AFSA no codificador do padrão H.264/AVC versão JM11.0. Posteriormente descrevemos cada um dos passos do UMHexagonS mostrando como estes foram implementados pelo JVT no JM11.0 e como foram alterados no

AUMHexagonS a partir da medida de movimento. A seguir apresentamos como alguns parâmetros do codificador foram alterados a partir da medida de movimento implementada.

5.2 Algoritmo AUMHexagonS

5.2.1 Implementação da Medida de Movimento

A medida de movimento do algoritmo AFSA foi descrita no Capítulo 4 e nesta seção apresentamos como foi a abordagem utilizada para a implementação no codificador H.264/AVC, através do código disponibilizado pelo JVT na versão JM11.0.

Para demonstrarmos a abordagem de implementação descreveremos passo a passo a comportamento do algoritmo em questão:

- A medida de movimento foi implementada em cima dos quadros P (tanto para a inicialização quanto para a atualização dos limiares T1 e T2), possibilitando a utilização de seqüências com estrutura IPPP, IBPB ou IBBP (neste trabalho as simulações foram realizadas com a estrutura IBBP de acordo com a recomendação do documento VCEG-N81 [29]);

- O primeiro quadro de uma seqüência com estrutura IBBP é um quadro I, o qual é codificado através das técnicas e parâmetros intra-quadros;

- Os limiares T1 e T2 são inicializados no primeiro quadro P, através de uma lista de SAD's referentes a cada macrobloco do quadro P. Ex.: Em uma imagem de resolução 176x144 onde esta é particionada em macroblocos de 16x16 tem-se 99 macroblocos os quais formarão uma lista decrescente de SAD's;

- A partir desses macroblocos, calcula-se o número de vetores de movimento que tiveram distorção maior que W/2, em qualquer dimensão, classificando-os como *na*;

- Da mesma forma, calcula-se o número de vetores de movimento que tiveram distorção maior que W/4 e menor que W/2, em qualquer dimensão, classificando-os como *nb*;

- A partir desses números o limiar T1 será o *na-ésimo* valor da lista decrescente de SAD's e o limiar T2 será o *nb-ésimo* valor da mesma lista;

- Dessa maneira os valores de SAD que recaírem acima do valor de T1 serão classificados como alto movimento, entre T1 e T2 serão classificados como médio movimento e abaixo de T2 como baixo movimento;

- Após o primeiro quadro I e o primeiro quadro P são codificados dois quadros B's, ainda sem utilização da medida de movimento;

- No 2º quadro P classificamos o movimento de cada macrobloco (alto, médio ou baixo movimento) de acordo com o quadro P anterior e codificamos cada macrobloco com os parâmetros previamente ajustados para cada tipo de movimento;

- Após o 2º quadro P, a mesma classificação é utilizada para a codificação dos próximos dois quadros B's;

- Após esta inicialização os limiares T1 e T2 são atualizados de acordo com as fórmulas apresentadas no Capítulo 4.

5.2.2 Alterações no UMHexagonS

O algoritmo UMHexagonS é utilizado pelo codificador do padrão H.264/AVC disponibilizado pelo orgão JVT (*Joint Video Team*) para a realização da estimação de movimento em pixel inteiro.

O algoritmo UMHexagonS é chamado de híbrido pois utiliza quatro passos com diferentes padrões de busca, sendo:

- Predição do Ponto Inicial de Busca;
- Busca Assimétrica Cruzada;
- Busca com Hexágonos Irregulares;
- Busca baseada em Hexágonos Estendidos;

A Figura 5.1 [27] apresenta os quatro passos citados acima:

A seguir detalhamos cada um dos quatro passos do algoritmo UMHexagonS:

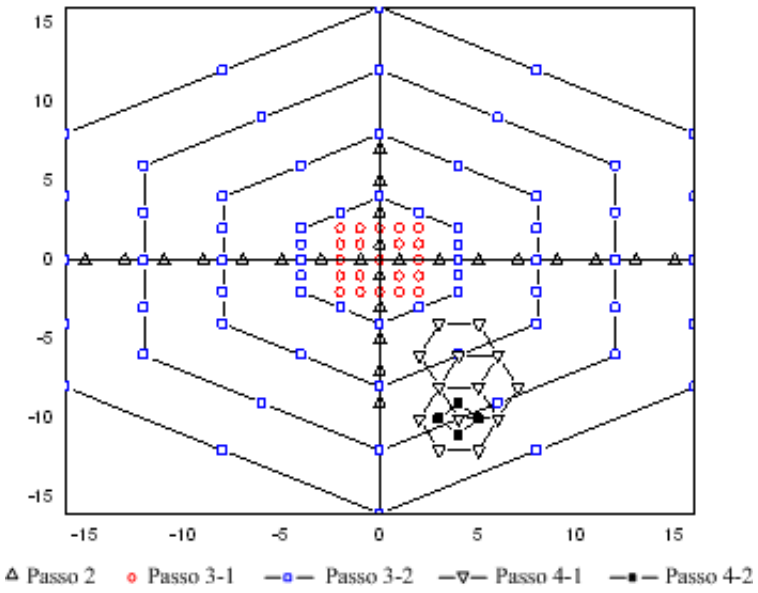

Figura 5.1: Processo de busca do algoritmo UMHexagonS, com W=16

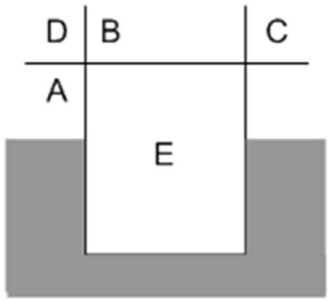

Figura 5.2: Blocos usados para cálculo do preditor mediano

5.2.3 Predição do Ponto Inicial de Busca

Conceito Teórico do UMHexagonS

A Figura 5.2 [27] mostra o preditor mediano usado no algoritmo através de vetores de movimento espacialmente vizinhos ao macrobloco corrente, onde mediana é definida como o valor que se situa no meio do grupo em análise, separando-o em 50% de valores inferiores e 50% de superiores.

Este preditor é calculado através do valor mediano entre os macroblocos vizinhos a esquerda, superior e superior direito do macrobloco corrente.

$$pred\_mv = median(Mv\_A, Mv\_B, Mv\_C) \tag{5-1}$$

A partir deste cálculo, o preditor mediano passa a ser mais um

candidato a vetor de movimento juntamente com os vetores de movimento dos blocos A, B e C. Entende-se como vetor candidato aquele que fornecerá uma posição para o cálculo de distorção.

O vetor de fornecer a menor distorção será o escolhido, mas para isso algumas regras são aplicadas antes da escolha do vetor de movimento que será usado para a predição:

1) Quando o bloco A estiver fora das dimensões da imagem, o vetor (0,0) é escolhido para substituir o vetor candidato do bloco A;

2) Quando o bloco C estiver fora das dimensões da imagem, o vetor do bloco D é utilizado para substituir o vetor candidato do bloco C;

3) Quando os blocos B e C estiverem fora das dimensões da imagem, os respectivos vetores de movimento são substituídos pelo vetor de movimento do bloco A;

Além desses candidatos, outros são introduzidos quando da busca do macrobloco que produzirá a menor distorção, da seguinte maneira:

A Figura 5.3[9] mostra que existem sete modos de predição definidos pelo JVT. A busca pelo modo que irá produzir a menor distorção acontece do modo 1 (16x16) ao modo 7 (4x4). Sendo assim, o algoritmo UmHexagonS utiliza como candidatos os vetores dos modos anteriores ao modo em questão, da seguinte maneira:

Para o modo 1 são utilizados os seguintes candidatos: co-situado (0,0); mv_A; Mv_B; Mv_C e o preditor mediano.

Para os outros modos (do 2 ao 7) são usados como candidatos o co-situado (0,0); o preditor mediano e os vetores de movimento dos modos anteriores (exemplo: modo 5 ou 6 é o modo anterior ao modo 7 e o modo 4 é anterior aos modos 5 ou 6).

Implementação do UMHexagonS no JM11.0

A implementação desse passo pelo JVT foi feita de acordo com o conceito teórico descrito acima.

Abordagem Utilizada pelo AUMHexagonS no JM11.0

A implementação desse passo no algoritmo AUMHexagonS foi feita de acordo com o conceito teórico descrito acima.

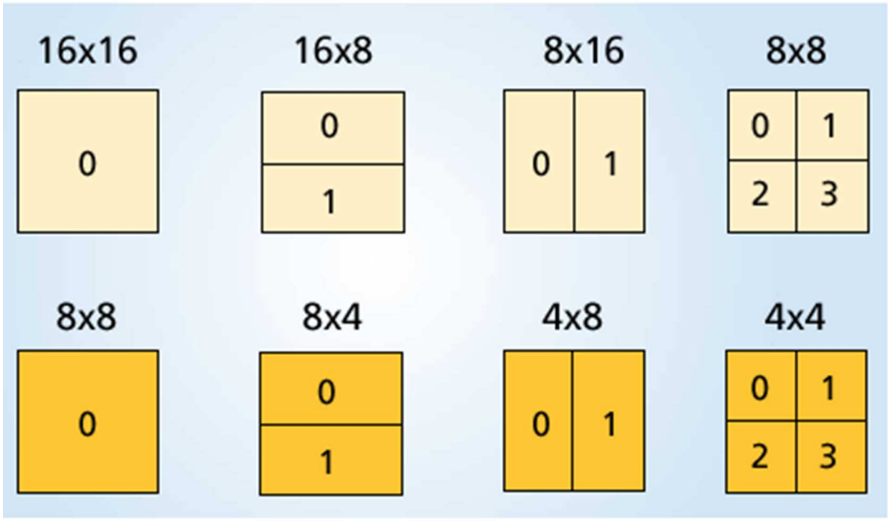

Figura 5.3: Tipos de blocos para estimação de movimento no padrão H.264

5.2.4 Busca Assimétrica Cruzada

Baseando-se no fato de que o movimento de uma cena é sempre mais intenso na direção horizontal que na vertical, o vetor de movimento ótimo pode ser obtido por uma busca assimétrica cruzada

O passo 2 da Figura 5.1 [27] mostra uma cruz assimétrica com tamanho horizontal igual a W e tamanho vertical igual a W/2 com espaçamento de duas unidades entre cada ponto da cruz assimétrica. O ponto que apresentar a menor distorção será o ponto de partida para o próximo passo.

Implementação do UMHexagonS no JM11.0

A implementação desse passo pelo JVT foi feita de acordo com o conceito teórico descrito acima.

Abordagem Utilizada pelo AUMHexagonS no JM11.0

As primeiras alterações foram introduzidas neste passo, de acordo com a classificação dada pela medida de movimento, sendo:

- Para baixo movimento: Alterou-se os valores da cruz assimétrica para W/2 na horizontal e para W/4 na vertical;

- Para médio movimento: Alterou-se os valores da cruz assimétrica para W na horizontal e para W/4 na vertical;

- Para alto movimento: Conservou-se os valores da cruz assimétrica para W na horizontal e para W/2 na vertical;

5.2.5 Busca com Hexágonos Irregulares

Conceito Teórico do UMHexagonS

Este passo é composto por duas buscas: primeiramente realiza-se uma busca exaustiva em torno do centro com W=2, como pode ser visto na Figura 5.1 [27] através do passo 3-1.

Essa estratégia baseia-se no fato de que a busca assimétrica cruzada fornecerá um acurado ponto de partida para posteriormente o passo que utiliza hexágonos irregulares trabalhe principalmente os casos em que há movimentos grandes e irregulares nas cenas de vídeo.

O algoritmo UMHexagonS utiliza o padrão de busca de um hexágono de 16 pontos (16HP - *Hexagon Pattern*) considerando que o movimento de uma cena é mais intenso na direção horizontal que na vertical, já que através desse padrão de busca tem-se mais pontos de busca na horizontal que na vertical.

Esses hexágonos de 16 pontos podem ser observados através do passo 3-2 da Figura 5.1 [27]. Sendo assim, grades de hexágonos irregulares são construídas com escalas diferentes, variando de 1 a W/4, sendo que o processo de busca se inicia do hexágono interior para o exterior.

Da mesma maneira, o melhor vetor de movimento derivado deste passo será o ponto de partida para o passo posterior.

Implementação do UMHexagonS no JM11.0

Com relação ao passo 3-1 o codificador JM11.0 utilizou uma busca exaustiva de 5x5, enquanto que para o passo 3-2 utilizou hexágonos de 16 pontos.

Abordagem Utilizada pelo AUMHexagonS no JM11.0

Para os passos 3-1 e 3-2 foram introduzidas as seguintes modificações:

- Para baixo movimento: Alterou-se os valores da busca exaustiva para um quadrado de 3x3 pontos e para hexágonos de 4 pontos;

- Para médio movimento: Alterou-se os valores da busca exaustiva para um quadrado de 4x4 pontos e para hexágonos de 8 pontos;

- Para alto movimento: Conservou-se os valores da busca exaustiva para um quadrado de 5x5 pontos e para hexágonos de 16 pontos;

5.2.6 Busca baseada em Hexágonos Estendidos

Conceito Teórico do UMHexagonS

A busca realizada pelo passo anterior pode fornecer vetores de movimento com boa precisão, porém com acuracidade diferente entre eles, dependendo da distância entre o centro da janela de busca e o ponto de busca. Por exemplo, quando o vetor de movimento fornecido pelo passo anterior localiza-se na área concêntrica exterior, o resultado da busca apresenta uma acuracidade baixa necessitando de um refinamento.

Para os casos em que o modo de predição que está sendo usado é pequeno, como no modo 7, a estratégia de busca a ser utilizada pode ser apenas não utilizar os passos 2 e 3, já que o vetor de movimento predito no modo 7 é suficientemente acurado.

O algoritmo UMHexagonS utiliza a busca com hexágonos estendidos como seu algoritmo de busca para refinamento, onde a vantagem é que a procura continua até que o ponto de menor distorção seja o centro do recém formado hexágono, como mostra a Figura 5.1 [27] através do passo 4.

Implementação do UMHexagonS no JM11.0

Com relação ao passo 4-1 o codificador JM11.0 utilizou um número de seis hexágonos estendidos, enquanto que para o passo 4-2 utilizou 4 hexágonos para o refinamento.

Abordagem Utilizada pelo AUMHexagonS no JM11.0

Para os passos 4-1 e 4-2 foram introduzidas as seguintes modificações:

- Para baixo movimento: Alterou-se os valores do passo 4-1 para um hexágono estendido e para o passo 4-2 para um hexágono de refinamento;

- Para médio movimento: Alterou-se os valores do passo 4-1 para três hexágonos estendidos e para o passo 4-2 para dois hexágonos de refinamento;

- Para alto movimento: Conservou-se os valores do passo 4-1 para seis hexágonos estendidos e para o passo 4-2 para quatro hexágonos de refinamento;

5.2.7 Outras Técnicas Utilizadas pelo UMHexagonS

Conceito Teórico do UMHexagonS

O algoritmo UMHexagonS também utiliza outras técnicas como uma busca para pixel fracionário e uma técnica para terminação forçada, as quais não fazem parte do escopo proposto para o algoritmo AUMHexagonS.

Porém, as técnicas desenvolvidas e implementadas no algoritmo AUMHexagonS para pixel inteiro podem ser utilizadas para pixel fracionário. Os resultados podem ser ainda melhores em termos de redução da complexidade computacional. No Capítulo 7 recomendamos estas etapas na seção sobre trabalhos futuros.

5.2.8 Alterações nos Parâmetros do Codificador H.264/AVC

Ainda aproveitando os benefícios de se utilizar uma medida de movimento, utilizou-se a classificação das cenas de uma seqüência de vídeo em alto, médio ou baixo movimento para otimizar a utilização dos parâmetros de codificação do H.264/AVC.

Nesse sentido alteramos os modos de predição da seguinte maneira:

- Para baixo e médio movimento: Passou-se a utilizar apenas os modos 16x16, 8x16 e 16x8. Dessa forma foi possível manter a qualidade em termos de PSNR e reduzir a carga computacional;

- Para alto movimento: Conservou-se os sete modos de predição, mantendo-se dessa forma a carga computacional e evitando perdas de qualidade em cenas com grande quantidade de movimento;