

4

Descrição dos Algoritmos de Estimação de Movimento

4.1

Introdução

A estimação de movimento, devido aos deslocamentos de objetos em uma cena ou da câmera, é realizada pelos algoritmos de estimação de movimentos, cujo objetivo é descrever, da melhor maneira e com a menor complexidade computacional possível, estes movimentos. O número de vetores de movimento a ser transmitido é fundamental para o valor da compressão final. Por esta razão os codificadores de vídeo não se utilizam de vetores de movimento pixel-a-pixel e sim bloco a bloco.

A transmissão de um vetor de movimento para cada pixel é indesejável e, em geral, desnecessária. Considerando que, normalmente, os elementos adjacentes de uma imagem estão submetidos ao mesmo movimento, alguns algoritmos de estimação de movimento dividem o quadro de imagem em blocos de pixels e, desta maneira, um único vetor é utilizado para representar o movimento de um bloco de pixels adjacentes. Este tipo de abordagem é chamado de casamento de blocos [24].

Os algoritmos que estarão em análise neste trabalho estão classificados como algoritmos de casamento de blocos, técnica esta que apresenta como principais fatores: bom desempenho, baixos esforços computacionais exigidos, além do fato de todos os padrões de compressão utilizarem este método.

4.2

Casamento de Blocos

Em algoritmos de casamento de blocos, um bloco ($N \times N$ pixels) de referência do quadro atual de uma sequência de imagens é comparado com blocos candidatos correspondentes dentro de uma janela de busca de tamanho $(2W + 1) \times (2W + 1)$ pixels no quadro anterior. Uma parte

ou todos os blocos do quadro anterior, dentro da área de busca, são confrontados com bloco de referência. Aquele que proporcionar o melhor casamento, ou seja, a menor distorção, fornecerá o vetor de movimento. Um vetor de movimento é transmitido para cada bloco de pixels do quadro atual.

Os principais parâmetros envolvidos nas técnicas de casamento de blocos são: o tamanho do bloco, a área de busca e os códigos utilizados para representar os vetores de movimento.

Este capítulo apresenta uma breve descrição de alguns dos principais algoritmos de estimação de movimentos por casamento de blocos publicados na literatura: FSA (*Full Search Algorithm*), UMHS (*Unsymmetrical-cross Multi-Hexagon-grid Search*) [28], 3SSA (*Three Step Search Algorithm*) [1], LOG-2D [4], N3SSA (*New Three Step Search Algorithm*) [21], 4SSA (*Four Step Search Algorithm*) [2], DSA (*Diamond Search Algorithm*) [8], UCBDSA (*Unrestricted Center-Biased Diamond Search Algorithm*) [26], CSA (*Cross Search Algorithm*) [7], IFA (*Integral Frames Algorithm*) [14], SSA (*Statistical Search Algorithm*) [25], AFSA (*Adaptive Full Search Algorithm*) [3], NNA (*Nearest Neighbors Algorithm*) [22], WUSA (*Winner Update Strategy Algorithm*) [31], SEA (*Successive Elimination Algorithm*) [5], ASEA (*Adaptive Successive Elimination Algorithm*) [6] e MSEA (*Modified Successive Elimination Algorithm*) [20].

4.2.1 FSA

Neste método, o bloco em análise do quadro atual de uma sequência de imagens é comparado com todos os blocos candidatos possíveis dentro da janela de busca de tamanho $(2W + 1) \times (2W + 1)$ pixels no quadro anterior. Com isto o número de testes de casamento, por bloco, é dado por:

$$Nt = (2W + 1)^2 \quad (4-1)$$

A vantagem do algoritmo FSA é o desempenho, ou seja, como todos os blocos candidatos possíveis dentro da janela de busca no quadro anterior são verificados sob o critério de mínima distorção, a possibilidade de se encontrar o erro mínimo global é maior que em outros algoritmos.

Na tentativa de diminuir o grande esforço computacional exigido pelo algoritmo FSA, diversos algoritmos de casamento de blocos de busca rápida têm sido propostos.

4.2.2 3SSA

O algoritmo 3SSA [1], normalmente é utilizado para uma janela de busca com $W = 7$ pixels e, para o primeiro passo, as buscas são realizadas somente para as seguintes posições: $(-4,-4)$; $(0,-4)$; $(4,-4)$; $(-4,0)$; $(0,0)$; $(4,0)$; $(-4,4)$; $(0,4)$; $(4,4)$. Um segundo passo é executado com um novo procedimento de busca utilizando-se o mínimo local encontrado no primeiro passo como centro de um novo diagrama de busca. Nesse passo as seguintes posições são avaliadas $(-2,-2)$; $(0,-2)$; $(2,-2)$; $(-2,0)$; $(0,0)$; $(2,0)$; $(-2,2)$; $(0,2)$; $(2,2)$. Finalmente, a exemplo do segundo passo, um terceiro e último passo, identifica o vetor de movimento utilizando-se mais uma vez do mínimo local encontrado no passo anterior, como centro de um último diagrama de busca, ou seja: $(-1,-1)$; $(0,-1)$; $(1,-1)$; $(-1,0)$; $(0,0)$; $(1,0)$; $(-1,1)$; $(0,1)$; $(1,1)$.

4.2.3 LOG-2D

Este algoritmo é baseado na premissa de que a distorção cresce monotonicamente à medida que nos afastamos do ponto de menor distorção [4]. O objetivo desse algoritmo, a exemplo dos demais, é determinar o bloco do quadro anterior que apresenta o melhor casamento com o bloco em análise do quadro atual. Isto é feito através de diversas etapas, sempre procurando determinar a direção de mínima distorção.

Cada passo do algoritmo consiste de procura em cinco localizações em forma de cruz. Este procedimento continua até que o plano de procura se reduza a um tamanho 3×3 . A distância entre os pontos de busca é reduzida se o mínimo de distorção coincidir com o centro do diagrama de busca ou com o limite da área de busca. No passo final, todas as nove posições são examinadas e a posição correspondente ao mínimo é a direção de mínima distorção.

4.2.4 N3SSA, 4SSA, DSA, UCBDSA, CSA, IFA e SSA

Esses algoritmos têm uma característica em comum aos algoritmos 3SS e LOG-2D que reside no fato de recaírem rapidamente no erro local mínimo resultando em uma significativa perda na acuracidade de estimação e, por conseguinte, no desempenho de compressão.

Em particular, o algoritmo SSA apresenta uma técnica de interesse na sub-amostragem dos macroblocos para o cálculo da medida de custo, porém mantendo o número de posições de procura a fim de não recair no erro local mínimo.

4.2.5 AFSA

Este algoritmo explora a correlação de movimento espacial dos blocos vizinhos e determina a origem da procura, ajustando o tamanho da janela de busca de acordo com os diferentes conteúdos de movimento dos blocos em análise [3].

A idéia básica deste algoritmo consiste em ajustar a origem de busca para cada bloco em análise através dos vetores de movimento dos blocos adjacentes à esquerda, esquerda superior e superior juntamente com o vetor $v = (0,0)$ (bloco co-situado do quadro anterior). O vetor (m,n) que minimiza a medida de distância Função Distorção Erro Absoluto Médio (EAM) aponta para a localização no quadro anterior a ser utilizada como origem da área de busca. Note que (m,n) representa uma predição do vetor de movimento do bloco de referência e identifica um bloco candidato inicial ao casamento. Cada bloco de referência é classificado em baixo, médio e alto movimento e a janela de busca é ajustada em $W/4$, $W/2$ e W , respectivamente. O algoritmo FSA é, então, aplicado.

A grande redução do número médio de procuras por quadro é obtida considerando-se que no algoritmo FSA são examinados

$$(2W + 1)^2 \quad (4-2)$$

pontos no quadro anterior para cada bloco de referência, e que somente um número pequenos de blocos possui muito movimento para a maioria das seqüências de imagens utilizadas.

Considerando-se que não é uma boa idéia fixar a área de procura para todos os blocos do quadro, uma vez que sempre haverá partes distintas de imagens com diferentes graus de movimento envolvidos, uma solução consiste em ajustar o tamanho da janela de busca de acordo com a quantidade de movimento contida em cada bloco. Desta maneira, pode ser economizada bastante complexidade computacional.

O valor encontrado para a função distorção EAM, na origem de busca, reflete o grau de movimento para um determinado bloco em questão. Valores altos de EAM indicam uma grande mudança entre o quadro presente e o

quadro passado e que existe uma grande possibilidade do bloco possuir muito movimento.

- Se $EAM > T1$ = Bloco com muito movimento;
- Se $T2 < EAM \leq T1$ = Bloco com médio movimento;
- Se $EAM \leq T2$ = Bloco com pouco movimento.

onde, T1 e T2 são dois valores de limiar.

Os valores de T1 e T2 são ajustados no início de cada sequência de imagens de acordo com a lista de EAM's calculadas para cada origem de busca e posteriormente são ajustados quadro a quadro de acordo com as expressões a seguir.

Para o quadro n, o algoritmo calculará [3]:

- $NH(n)$ = número de blocos com muito movimento;
- $NM(n)$ = número de blocos com médio movimento;
- $N1(n)$ = número de blocos com movimento maior que $W/2$;
- $N2(n)$ = número de blocos com movimento entre $W/4$ e $W/2$;
- $PH(n) = N1(n)/NH(n) \times 100\%$;
- $PM(n) = N2(n)/NM(n) \times 100\%$;

$$T1^{n+1} = T1^n - \lfloor |PH(n) - PH(n-1)/20\%| \rfloor \text{sgn } PH(n) - PH(n-1) \quad (4-3)$$

$$T2^{n+1} = T2^n - \lfloor |PM(n) - PM(n-1)/20\%| \rfloor \text{sgn } PM(n) - PM(n-1) \quad (4-4)$$

onde, $\lfloor x \rfloor$ representa o maior inteiro menor que x , $\text{sgn}(f) = -1; f < 0$ e $\text{sgn}(f) = +1; f > 0$

4.2.6 NNA, WUpA e UMHS

Os algoritmos citados nesta seção ajustam a origem de procura de cada bloco de acordo com os vetores de movimento dos blocos adjacentes vizinhos. Em particular, o algoritmo WUpA utiliza um interessante critério de descarte alcançando bons resultados em termos de redução de complexidade, mantendo a solução ótima alcançada pelo FSA.

O algoritmo UMHS é o algoritmo que foi adotado pelo JVT para ser utilizado pelos codificadores de referência do padrão H.264/AVC. Este algoritmo apresenta grande redução da complexidade, quando comparado ao FSA, mantendo a qualidade em termos de PSNR. Este algoritmo será descrito em detalhes no Capítulo 5, já que este foi a referência utilizada para a criação do algoritmo proposto neste trabalho.

4.2.7 SEA

A idéia básica deste algoritmo consiste em se obter a melhor estimativa dos vetores de movimento através de sucessivas eliminações de posições de procura dentro da janela de busca, e com isso diminuir o número necessário de avaliações de casamento, tornando menor o esforço computacional necessário [5].

Considerando-se blocos de tamanho $N \times N$ pixels, janela de busca de $(2W + 1) \times (2W + 1)$ pixels, que $f(i,j,t)$ é a intensidade de um pixel com coordenada (i,j) no quadro "t", que a função EAM é utilizada para avaliar o casamento entre dois blocos e ainda considerando que o casamento seja realizado com o quadro "t - 1" anterior, temos:

$$Casamento = F(i, j, t) - F(i - x, j - y, t - 1) \quad (4-5)$$

Para mostrar a equação final do algoritmo em análise, que permitirá reduzir o número médio de procuras por quadro, utiliza-se as seguintes inequações matemáticas:

$$||a| - |b|| \leq |a - b| \quad (4-6)$$

$$|a| - |b| \leq |a - b| \quad (4-7)$$

Utilizando-se estas inequações e a expressão de casamento, é fácil mostrar que:

$$R - M(x, y) \leq EAM(x, y) \quad (4-8)$$

$$M(x, y) - R \leq EAM(x, y) \quad (4-9)$$

onde R representa a soma de normas dos pixels do bloco de referência, $M(x, y)$ é a soma de normas dos pixels de um determinado bloco candidato ao casamento com vetor de movimento (x, y) , e $EAM(x, y)$ representa a Função Distorção Erro Absoluto Médio para o vetor de movimento (x, y) .

Assumindo que tenhamos obtido $EAM(m, n)$ para um bloco inicial candidato a casamento, com vetor de movimento (m, n) , fica claro que a procura por outros blocos candidatos que apresentem melhor casamento (menor EAM) deverá ser realizada somente para aqueles que apresentarem vetor de movimento (x, y) e atenderem à seguinte condição:

$$EAM(x, y) \leq EAM(m, n) \quad (4-10)$$

Utilizando-se a inequação anterior, podemos reescrever (4-8) e (4-9) da seguinte maneira:

$$R - M(x, y) \leq EAM(m, n) \quad (4-11)$$

$$M(x, y) - R \leq EAM(m, n) \quad (4-12)$$

Finalmente, de (4-11) e (4-12), temos:

$$R - EAM(m, n) \leq M(x, y) \leq R + EAM(m, n) \quad (4-13)$$

A inequação (4-13) é o resultado de maior importância utilizado pelo algoritmo SEA. Esta expressão mostra que a procura para se obter o melhor casamento (menor EAM), só será realizada nos blocos que apresentarem soma de normas que a satisfaçam.

4.2.8 ASEA

O algoritmo ASEA utiliza-se da mesma estratégia (correlação de movimento espacial dos blocos vizinhos) empregada pelo algoritmo AFSA para aumentar a eficiência da predição inicial do vetor de movimento dos esquemas de eliminação sucessiva.

4.2.9 MSEA

O algoritmo MSEA é uma modificação do algoritmo SEA, que mantém a solução ótima para busca do erro mínimo global e reduz ainda mais a complexidade do FSA quando comparado ao SEA original.

Este algoritmo introduz novos critérios de descarte dos macroblocos, antes do cálculo da medida de custo. Estes critérios ajudam na pré-seleção dos macroblocos a serem utilizados para o cálculo da medida de custo.

Como inovação, este algoritmo traz a incorporação de critérios para descarte dos blocos. Novos critérios de descarte são inclusos e executados antes do cálculo de distorção, sendo estes critérios organizados em ordem crescente de complexidade.

Além disso, é realizada a pré-ordenação dos blocos antes do cálculo de distorção, onde a soma das normas apontará para os blocos que sofrerão o cálculo de distorção.

Em termos de resultado, o algoritmo MSEA obtém os mesmos valores de PSNR que o SEA e o FSA, alcançando redução no tempo de processamento em torno de 60 vezes, quando comparado ao FSA, e em torno de 48 vezes, quando comparado ao SEA [5].

4.2.10 Algoritmos de interesse

Após esta breve análise, definiram-se como algoritmos de interesse para esta dissertação de Mestrado, os seguintes algoritmos:

- FSA (*Full Search Algorithm*)
- AFSA (*Adaptive Full Search Algorithm*)
- UMHS (*Unsymmetrical-cross Multi-Hexagon-grid Search*)

As escolhas acima tiveram como base as seguintes premissas:

- Aplicar técnicas que se aproximem do erro mínimo global, alcançando qualidade, em termos de PSNR, sempre maior ou igual as dos algoritmos FSA e UMHS.
- Implementar técnicas de medida de movimento com o objetivo de utilizar os parâmetros corretos para cada tipo de movimento.
- Utilizar os algoritmos implementados no codificador de referência do padrão H.264/AVC disponibilizado pelo órgão JVT (*Joint Video Team*).

- Reduzir a complexidade computacional quando comparado aos algoritmos FSA e UMHS.
- Desenvolver algoritmo que possa ser utilizado para a aplicação de TV Digital.