Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica
Programa de Pós-Graduação em Engenharia Elétrica

Projeto de Pesquisa

Algoritmo de Busca Adaptativo baseado em

Jean Felipe Fonseca de Oliveira

Campina Grande – PB
Abril de 2012

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica
Programa de Pós-Graduação em Engenharia Elétrica

Algoritmo de Busca Adaptativo baseado em

Jean Felipe Fonseca de Oliveira

Relatório de Projeto de Pesquisa submetido à Coordenação do Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica da Universidade Federal de Campina Grande como requisito necessário para a obtenção do grau de Doutor em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Comunicações

Marcelo Sampaio de Alencar
Orientador

Campina Grande – PB, Maio de 2011

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Nos dias atuais, as aplicações multimídia tem se tornado mais flexíveis e mais poderosas com a evolução dos semicondutores e o desenvolvimento de novos métodos de processamento de sinais digitais. Devido à limitação da largura de banda do canal e dos rigorosos requisitos de reprodução de vídeo em tempo real, a codificação é um processo indispensável para muitas aplicações de comunicação visual que requerem taxas de compressão muito alta. A grande quantidade de correlação temporal entre quadros adjacentes em uma seqüência de vídeo, também chamada de redundância temporal, deve ser devidamente identificada e eliminada para garantir essas taxas de compressão [2].

Com o aumento na popularidade das comunicações de vídeo, a qualidade da experiência do usuário passam a ser uma das preocupações mais importantes na concepção e avaliação de sistemas multimídia [**?**].

Em uma cadeia de transmissão de vídeo, vários fatores influenciam e prejudicam a qualidade da imagem exibida resultante. Um desses fatores é o próprio algoritmo de codificação de fonte. Como consequência da codificação com perdas, uma degradação visível da qualidade do vídeo pode ser observada [4].

Capítulo 2

Revisão

O objetivo da estimação de movimento é a redução de redundância temporal entre quadros causada pela correlação de objetos em movimento. No entanto, a estimação e codificação de vetores de movimento devem ser apropriados aos custos computacionais e taxas de bits de acordo com as perspectivas de cada sistemas de compressão. Dessa forma, é muito importante a relação entre a precisão da estimação de movimento e simplicidade dos campos de vetores de descrição [1].

A abordagem mais popular é a de reduzir o número de locais de pesquisa utilizando o pressuposto da superfície de erro unimodal em que o erro da busca diminui monotonicamente quando a posição de melhor busca se aproxima do ponto ótimo global. No entanto, esta hipótese nãoé geralmente satisfeita, resultando em um erro de predição, conhecido como erro de mínimo local.

A estimação de movimento ocupa de 60% a 90% do tempo computacional de todo codificador variando das configurações mais simples para as configurações mais complexas, respectivamente. O artigo [**?**] define e implementa uma técnica de estimação de movimento fracionada em que todo processo ocupa apenas metade do tempo computacional em que um codificador padrão executaria essa tarefa.

Uma solução para a redução da quantidade de esforço computacional é usar os dados da sequência de vídeo no domínio codificado. Abordagens de processamento de vídeo no domínio codificado são recentes e menos exploradas em comparação com o processamento no domínio de pixel. Um exemplo de um algoritmo simples e rápido para a detecção de alterações de domínio codificado é apresentado por [**?**].

O processamento no domínio codificado evita a decodificação e reconstrução completa do vídeo, o qual fornece um potencial para processamento em tempo real de fluxos de vídeo múltiplos, por exemplo. O processamento no domínio codificado tem também a vantagem de extrair dados do fluxo de vídeo, que são gerados utilizando os dados originais não comprimidos, os que não estarão disponíveis durante o processamento do fluxo decodificado [3].

2.1 O Codificador H.264 AVC

2.2 A Estimação de movimento baseada em blocos

Os algoritmos de busca de blocos são a técnica mais popular usada na estimação de movimento de codificadores de vídeo. Em geral, esses algoritmos se baseiam na divisão do quadro de luminância em macroblocos não sobrepostos de tamanho N×N que, por sua vez, são comparados com o macrobloco correspondente e seus vizinhos adjacentes no quadro de referência para criar um vetor que estipula a sua movimentação, encontrando o macrobloco correspondente do mesmo tamanho $N \times N$ na área de busca no quadro de referência. A posição do macrobloco correspondente no quadro de referência dá o vetor de movimento (MV) do macrobloco corrente, como mostrado na Figura 1. Este vetor de movimento é constituído das coordenadas (x, y) do canto esquerdo-superior do macrobloco corrente representando as coordenadas iniciais do vetor e das coordenadas (x, y) do canto esquerdo-superior do macrobloco do quadro de referência. Estas coordenadas podem ser positivas ou negativas. Um valor positivo significa um movimento para a direita e/ou um movimento descendente e um valor negativo significa um movimento para a esquerda e/ou movimento ascendente.

Esses vetores de movimento serão usados no decodificador para predizer um novo quadro a partir do quadro de referência. Esse processo é chamado de compensação de movimento e está ilustrado no Figura 2. A métrica usada é geralmente determinada usando uma das medidas de distorção de blocos (BDM – *Block Distortion Measure*) como a diferença média absoluta (MAD – *Mean Absolute Difference*), a soma das diferenças absolutas (SAD – *Sum of Absolute Differences*) ou erro médio quadrático (MSE – *Mean Squared Error*). O macrobloco com o menor custo, retornado por uma dessas métricas, é considerado o correspondente ao macrobloco corrente.

2.3 Medidas de distorção de blocos (BDM – *Block Distortion Measure*)

Para cada desses critérios um bloco de tamanho $N \times N$ é considerado. O valor do *pixel* na coordenada (n_1, n_2) no quadro k é dado por $S(n_1, n_2, k)$ em que $0 \leq n_1, n_2 \leq N-1$. O quadro k representa o quadro corrente, assim como o bloco de *pixels* descrito acima representa o bloco corrente. Nas subseções 2.3.1, 2.3.2, 2.3.3 são descritas três medidas de distorção de bloco empregadas em codificadores de vídeo.

2.3.1 Erro Médio Quadrático

Considerando $(k-l)$ como os quadros de referência passados, para $(l > 0)$, no processo de estimação de movimento, o erro médio quadrático para um bloco de $N \times N$ *pixels* é dado por 2.1 [2] [**?**].

$$MSE(i,j) = \frac{1}{N^2} \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} [S(n_1,n_2,k) - S(n_1+i, n_2+j, k)]^2 \tag{2.1}$$

O significado físico da equação 2.1 deve ser bem compreendida. Considera-se um bloco de pixels de tamanho $N \times N$ no quadro de referência a um deslocamento de (i,j), em que i e j são números inteiros em relação à posição do bloco candidato no quadro corrente.

O erro médio médio quadrático é calculado para cada posição (i,j) de deslocamento dentro de um intervalo de pesquisa especificado na imagem de referência e o deslocamento, que dá o menor valor de MSE, é designado como vetor de deslocamento que é comumente conhecido como vetor de movimento e é dado por 2.2.

$$[d_1, d_2] = \min_{i,j} [MSE(i,j)] \tag{2.2}$$

O cálcudo erro médio quadrático requer o processamento de N^2 subtrações, N^2 multiplicações e (N^2-1) adições para cada bloco candidato em cada posição da busca. Isto é computacionalmente dispendioso e um critério mais simples, como definido na subseção 2.3.2 é muitas vezes preferido sobre o critério MSE.

2.3.2 Diferença Absoluta Média

Como o critério MSE, a diferença média absoluta (MAD – *Mean Absolute Difference*) também faz com que os valores de erro sejam sempre positivos, mas em vez de somar as diferenças de quadrados, as diferenças absolutas é que serão somadas. A medida das diferença média absoluta, a MAD, é definida em 2.6 [**?**].

$$MAD(i,j) = \frac{1}{N^2} \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} [S(n_1,n_2,k) - S(n_1+i, n_2+j, k)] \tag{2.3}$$

O vetor de movimento é determinado de forma semelhante à medida MSE, que usa Fórmula **??**, e é definada por 2.4.

$$[d_1, d_2] = \min_{i,j} [MSE(i,j)] \tag{2.4}$$

O critério MAD requer o processamento de N^2 subtrações de valores absolutos e N^2 adições para cada bloco candidato em cada posição de busca determinada pelo algoritmo de estimação de movimento. A ausência de multiplicações faz desse critério, uma opção

computacionalamente menos dispendiosa, facilitando assim possíveis implementações em *hardware*.

2.3.3 Correspondência na Contagem dos *Pixels*

Nesse critério, os *pixels* de um bloco candidato B são comparados com os *pixels* correspondentes do bloco com deslocamento (i, j) no quadro de referência e aqueles que são menores do que um limiar especificado são contados. A contagem para a comparação e o deslocamento (i, j) para os quais a contagem é máxima corresponde as coordenadas do vetor de movimento e é expressa em um função binária $count(n_1, n_2) \forall (n_1, n_2) \in B$ definida em **??**.

$$count(n_1, n_2) = \begin{cases} 1, & \text{se } |S(n_1, n_2, k) - S(n_1 + i, n_2 + j, k - l)| \leq \theta \\ 0, & \text{caso contrário} \end{cases} \tag{2.5}$$

em que θ é um limiar pré-determinado. A correspodência na contagem de *pixels* para um deslocamento (i, j) é dada pelo valor acumulado da equação 2.5 é em definida por **??**

$$MPC(i, j) = \sum_{n_1=0}^{N-1} count(n_1, n_2) \tag{2.6}$$

e as respectivas coordenadas finais do vetor de movimento são definadas por 2.7

$$[d_1, d_2] = \min_{i,j} [MPC(i, j)] \tag{2.7}$$

2.4 Os algoritmos de estimação de movimento implementados no codificador H.264 JM18.2

2.4.1 Busca Exaustiva

Na estimação de movimento por busca exaustiva (FS – *Full Search*) [**?**] a busca pelo bloco candidato é exaustivamente realizada em todas as posições possíveis dentro de um raio de busca w pré-determinado no quadro de referência. A Figura X ilustra esse princípio.

Um bloco de $N \times N$ *pixels* é considerado dentro do quadro candidato como mostrado na Figura X na coordenada (r, s). Considera-se então uma janela de pesquisa tendo um raio de busca $\pm w$ em ambas direções no quadro de referência. Para cada posição de busca $(2w + 1)^2$, o bloco candidato é comparado com um bloco de tamanho $N \times N$ de acordo com o critério de distorção utilizado e o bloco com menor distorção, juntamente com o vetor de movimento, são determinados depois que todas as $(2w + 1)^2$ posições forem avaliadas.

A busca exaustiva tem um resultado ótimo, desde de que o intervalo de busca seja corretamente definido, garantindo assim a determinação do deslocamento exato do bloco candidato. No entanto, esse processo exige processamentos elevados, muitas vezes impeditivos para determinadas aplicações, principalmente quando aplicado a codificação de vídeo em tempo real ou vídeos com altas resoluções com duas ou mais visões. Para cada posição de busca são exigidos $O(N^2)$ processamentos (adições, subtrações, multiplicações, etc) e uma vez que existem $(2w+1)^2$ posições, o número de processamentos considerando os critérios de distorção descrito na seção 2.3 são apresentados na Tabela 2.1.

Tabela 2.1: Complexidade computacional da busca exaustiva

Criterio de Distorção	**Número de processamentos**		
	Adições/Subtrações	Multiplicações	Comparações
MSE	$(2N^2-1)(2w+1)^2$	$2N^2(2w+1)^2$	$(2w+1)^2$
MAD	$(2N^2-1)(2w+1)^2$		$(2w+1)^2$
MPC	$(2N^2-1)(2w+1)^2$		$2N^2(2w+1)^2$

Para exemplificar os valores da Tabela 2.1, dado um valor $w = 7$ *pixels*, considerando que a melhor busca encontra-se com um deslocamento menor igual a 7 *pixels* com relação a posição no bloco corrente, será necessário realizar o procedimento de busca em $15\times15 = 255$ ($N\times N$) posições. Para implementações que demandem menos processamento, alguma alternativas com menos posições de busca e com resultados semelhantes foram desenvolvidas e são exploradas na maioria dos codificadores disponíveis atualmente. Entretando, deve-se enfatizar que essas técnicas de pesquisa rápida por não realizarem a busca exaustiva na área de busca podem, no máximo, obterem resultados sub-ótimos.

2.4.2 Fast Full Search

2.4.3 UMHEX

2.4.4 UMHEXSIMPLE

2.4.5 EPZS

Referências Bibliográficas

[1] Multimedia Communications and Slovak Republic. The fast search motion estimation algorithms and their errors. *Informatica*, 8(3):26–30, 2008.

[2] B Kamolrat, W A C Fernando, M Mrak, and A Kondoz. 3D motion estimation for depth image coding in 3D video coding, 2009.

[3] Krzysztof Szczerba, Sø ren Forchhammer, Jesper Stottrup-Andersen, and Peder Tanderup Eybye. Fast Compressed Domain Motion Detection in H.264 Video Streams for Video Surveillance Applications. *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 478–483, 2009.

[4] Tobias Wolff, Hsin-han Ho, John M Foley, and Sanjit K Mitra. H . 264 CODING ARTIFACTS AND THEIR RELATION TO PERCEIVED ANNOYANCE. *Signal Processing*, (Eusipco):1–5, 2006.