

## ALGORITMO DE ESTIMAÇÃO DE MOVIMENTO ADAPTATIVO AO CONTEXTO PARA VÍDEOS DIGITAIS DE ALTA RESOLUÇÃO

**CARDOSO, Gabriel<sup>1</sup>; AGOSTINI, Luciano<sup>2</sup>**

<sup>1</sup>UFPEL, Curso de Ciência da Computação.  
Email: glcardoso@inf.ufpel.edu.br

<sup>2</sup>UFPEL, Centro de Desenvolvimento Tecnológico.  
Email: agostini@inf.ufpel.edu.br

### 1 INTRODUÇÃO

Um vídeo é uma sequência de imagens estáticas que, para causar uma impressão de movimento, são amostradas a uma taxa de 24 a 30 imagens por segundo. Essas imagens possuem dois tipos de semelhanças, a semelhança entre posições do próprio quadro e as semelhanças entre regiões de quadros vizinhos. As semelhanças, por sua vez, geram redundância de informações. Essas redundâncias são exploradas pelos codificadores de vídeo, visando diminuir a quantidade de dados que deve ser armazenada ou transmitida para representar um vídeo.

A redundância quanto às semelhanças internas ao próprio quadro é chamada de redundância espacial. Já a redundância entre quadros vizinhos é chamada de redundância temporal e a técnica utilizada para reduzir esta redundância é a estimação de movimento (*Motion Estimation* - ME). A ME tem como função encontrar o bloco nos quadros vizinhos que mais se assemelha ao bloco atual em processamento (RICHARDSON, 2003), ou seja, a ME procura o melhor casamento entre um bloco do quadro que está sendo codificado com um ou mais quadros anteriormente processados. Para isto, a ME utiliza um algoritmo de busca que define como será feita a procura pelo bloco mais semelhante dentro da área de pesquisa. Para definir se foi encontrado o melhor casamento entre dois blocos é necessário definir uma métrica de distorção, que indique a semelhança entre os blocos. Neste trabalho, é usada a soma absoluta das diferenças (SAD) (RICHARDSON, 2003) como métrica de distorção. O SAD é a métrica mais usada pela comunidade da área de codificação de vídeo em função de sua simplicidade e de sua eficiência.

A ME é responsável por 80% da complexidade dos codificadores de vídeos atuais (CHENG, 2009) tornando-se, assim, uma partes mais importantes do codificador. A maior parte desta complexidade é função da o algoritmo de busca escolhido, portanto, o algoritmo de busca deverá ser escolhido mediante alguns critérios como a complexidade computacional e a qualidade dos resultados gerados. O algoritmo de busca mais conhecido é o *Full Search* (FS), que realiza uma busca completa do melhor casamento entre o bloco a ser codificado e os blocos candidatos no quadro anterior. Por esta razão, o FS gera resultados ótimos. Todos os demais algoritmos são considerados algoritmos sub-ótimos ou algoritmos rápidos, pois utilizam uma heurística para tentar encontrar o melhor casamento entre dois blocos, sem realizar todas as comparações possíveis. São exemplos de algoritmos rápidos os algoritmos *Diamond Search* (DS), *Three Step Search* (TSS), *One at Time Search* (OTS), entre outros (KHUN, 1999).

Com a popularização de aplicativos capazes de processar resoluções de vídeos em resolução elevada, como *High Definition* (HD) e *Full High Definition* (Full HD), os vídeos digitais passaram a ter mais qualidade, graças a um aumento na

quantidade de informações que devem ser guardadas para representar um vídeo. Isto ocasiona um aumento proporcional da complexidade do codificador e, conseqüentemente, da ME. Com isto, é necessário investigar novas técnicas para diminuir a complexidade dos algoritmos sem, no entanto, gerar um impacto expressivo na qualidade e na taxa de compressão. A perda na qualidade e na taxa de compressão são inevitáveis e estão presentes em algoritmos como DS, TSS e OTS, mas é importante que estas perdas sejam minimizadas. Neste artigo é apresentado um novo algoritmo, visando diminuir a complexidade do algoritmo *Full Search*, utilizando algoritmos rápidos de forma seletiva e gerando perdas mínimas na qualidade objetiva do vídeo.

## 2 METODOLOGIA (MATERIAL E MÉTODOS)

Os algoritmos rápidos de ME possuem algumas características em comum. Em suma, estes algoritmos possuem um padrão de busca, que vai ser repetido por  $n$  iterações, até que a condição de parada seja satisfeita. Este padrão de busca visa uma diminuição na complexidade dos algoritmos mantendo uma qualidade próxima da ótima. O problema dessas heurísticas é que, por fazerem uma melhor escolha local, elas não levam em consideração a área de pesquisa total, fazendo que elas possam cair em mínimos locais. Estes mínimos locais são áreas da área de pesquisa que possuem um casamento bom com o bloco que está sendo codificado, porém não é o melhor. Assim é gerada a perda da qualidade objetiva do vídeo codificado.

Normalmente, os algoritmos de busca para ME se preocupam somente com a redundância temporal que há entre os quadros e acabam ignorando que os quadros anteriores também possuem uma redundância espacial. O objetivo do algoritmo proposto neste trabalho é usar a redundância espacial da área de pesquisa como um meio de diminuir a complexidade do algoritmo *Full Search*, utilizando algoritmos rápidos para as áreas homogêneas. Estas áreas homogêneas possuem uma quantidade muito menor de mínimos locais, então o uso do algoritmo rápido não irá gerar perdas significativas. O algoritmo utilizado para essa abordagem será o DS, que apresenta um dos melhores resultados dentre os algoritmos rápidos. O problema recai em como decidir qual algoritmo será usado sem aumentar excessivamente a complexidade do algoritmo. Para isto, foram utilizadas duas abordagens diferentes para tentar definir se a área de pesquisa é homogênea. Estas abordagens tratam da soma das diferenças absolutas entre alguns pontos da área de pesquisa. Os pontos selecionados nessas duas abordagens estão apresentados na Figura 1.

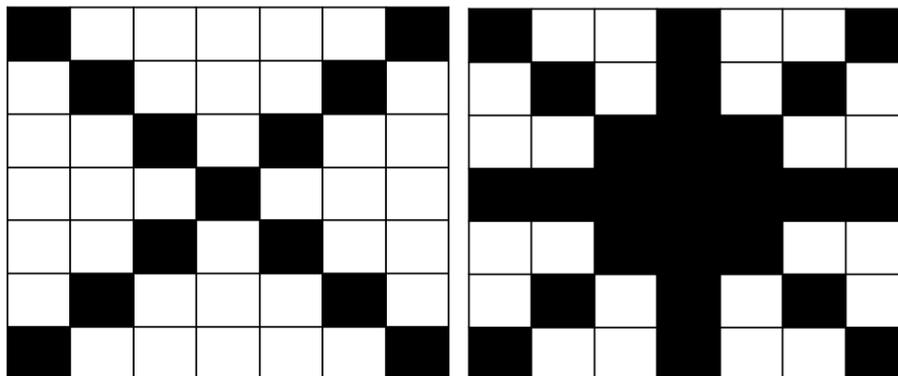


Figura 1 – Formas de definição da homogeneidade da área de pesquisa (a) usando as diagonais e (b) usando um formato asterisco.

Após realizada a soma das diferenças absolutas da área é possível obter um grau quantitativo de homogeneidade desta área. Assim é possível criar um critério para definir quando a área é homogênea ou não. Para isto foram realizados testes tentando definir qual seria o melhor grau quantitativo de homogeneidade e se há somente um grau para diferentes tamanhos de áreas de pesquisa.

Os testes de qualidade da abordagem proposta foram realizados através de um algoritmo que foi implementado em um software planejado para tal finalidade na linguagem C++. Foram executados testes com três vídeos de alta resolução para tamanhos de bloco de 8x8 pixels e uma área de pesquisa de 32x32 blocos para 100 quadros por vídeo.

### 3 RESULTADOS E DISCUSSÃO

Para estipular qual a faixa de valores de SAD seria adequada para determinar o grau quantitativo de homogeneidade onde o uso do *Diamond Search* é suficiente, algumas avaliações foram realizadas. A Tabela 1 mostra os resultados obtidos pela abordagem demonstrada na Figura 1(a) com o limiar de 100 e 150 no grau de homogeneidade (valor de SAD) para o uso do *Diamond Search*. O *Peak-to-Signal-Ratio* (PSNR) é uma métrica utilizada para analisar a qualidade objetiva do vídeo. O PSNR é medido em decibéis (dB) e quanto maior for o seu valor, maior a qualidade objetiva do vídeo (RICHARDSON, 2003). O número de cálculos efetuados foi medido em números de cálculos de SAD.

Tabela 1 – Resultados do algoritmo proposto usando padrão da Figura 1(a).

Algoritmo	PSNR Médio	Cálculos de SAD (x10 <sup>9</sup> )	Modo de Decisão (x10 <sup>6</sup> )
<i>Full Search</i>	39,13	3,52	0
Proposto (100)	39,06	3,26	0,32
Proposto (150)	37,86	1,76	0,32

Como é possível perceber na Tabela 1, o algoritmo atinge uma redução de 8% na complexidade computacional com uma perda de somente 0,07 dB quando o limiar do grau de homogeneidade para o uso do *Diamond Search* está em 100. Sendo assim, este um valor é bem próximo do limite onde existe ganho computacional sem perdas no PSNR. Entretanto, o algoritmo obtém taxas maiores de redução na complexidade computacional sem obter perdas expressivas de PSNR, como é possível observar, quando o limiar está em 150. Neste caso, é possível obter uma redução da complexidade computacional de até 45% com uma perda no PSNR de 1,27 dB.

A Tabela 2 demonstra os resultados obtidos nesta avaliação quando os limiares do grau de homogeneidade de 150 e 250 são usados para a abordagem apresentada na Figura 1(b). Neste caso, os valores de limiar foram alterados em relação aos resultados da Tabela 1 porque a abordagem da Figura 1(b) utiliza mais amostras na comparação do que a abordagem da Figura 1(a) e, portanto, o valor acumulado das diferenças é maior.

Com a abordagem da Figura 1(b) e um limiar de 150, é possível obter uma redução pequena no número de cálculos de SAD se comparada com a abordagem anterior, porém, sem gerar perdas no PSNR do vídeo quando comparado com o *Full Search*. Resultados mais expressivos nesta abordagem

podem ser obtidos quando são utilizados limiares maiores, como 250, onde o algoritmo é capaz de atingir uma redução de 37% na complexidade computacional, com uma perda no PSNR de somente 0,415 dB.

Tabela 2 – Resultados do algoritmo proposto usando o padrão da Figura 1(b).

Algoritmo	PSNR Médio	Número de SAD ( $\times 10^9$ )	Modo de Decisão ( $\times 10^6$ )
Full Search	39,13	3,52	0
Proposto(150)	39,13	3,48	0,64
Proposto(250)	38,72	2,19	0,64

## 4 CONCLUSÃO

Neste trabalho, foi apresentado um novo algoritmo de estimação de movimento que considera, também, a redundância temporal da área de pesquisa. Assim, é possível o uso seletivo de algoritmos rápidos (*Diamond Search*) para regiões homogêneas e do *Full Search* para regiões heterogêneas. Os resultados obtidos foram comparados com o *Full Search* e demonstram que é possível reduzir em até 37% a complexidade da ME, com uma perda de somente 0,415 dB na qualidade objetiva do vídeo (PSNR). Deste modo, o algoritmo desenvolvido possui um melhor compromisso entre qualidade e custo computacional do que o *Full Search*. Com isso, é possível demonstrar que é possível reduzir a complexidade computacional da ME com uma perda mínima e aceitável na qualidade.

Como trabalhos futuros, estão planejados mais testes, na tentativa de analisar o comportamento do algoritmo sobre outros cenários, como a mudança da área de pesquisa, o tamanho de bloco, o grau quantitativo de homogeneidade e diferentes resoluções.

## 5 REFERÊNCIAS

RICHARDON, Iain. **H.264 and MPEG-4 Video Compression – Video Coding for Next Generation Multimedia**. Chichester: John Wiley&Sons, 2003.

KUHN, P. **Algorithms, Complexity, Analysis and VLSI Architectures for MPEG-4 Motion Estimation**, Springer, 1999.

CHENG, Y. CHEN, Z. and CHANG, P. “An H.264 Spation-Temporal Hierarchical Fast Motion Estimation Algorithm for High-Definition Video”. **IEEE International Symposium on Circuits and Systems**, ISCAS, p. 880 - 883, 2009.