

## INVESTIGAÇÃO DE ALGORITMOS RÁPIDOS MULTIPONTO PARA A ESTIMAÇÃO DE MOVIMENTO DE VÍDEOS EM ALTA RESOLUÇÃO

SIEDLER, Gabriel; NOBLE, Diego  
Grupo de Arquiteturas e Circuitos Integrados  
Universidade Federal de Pelotas

PORTO, Marcelo; AGOSTINI, Luciano  
Grupo de Arquiteturas e Circuitos Integrados  
Universidade Federal de Pelotas

### 1 INTRODUÇÃO

Um vídeo é uma sequência de imagens estáticas que, para causar a sensação de movimento, devem ser apresentadas em uma taxa de 24 a 30 quadros por segundo, o que faz com que os quadros vizinhos temporalmente sejam muito similares. Essa semelhança, por sua vez, gera uma redundância, chamada de redundância temporal, que pode ser explorada pelo codificador de vídeo para reduzir consideravelmente o tamanho final do vídeo (RICHARDSON, 2003).

Para explorar essa redundância, a estimação de movimento (ME – *Motion Estimation*) tem como função - dado um bloco do quadro atual (o bloco que está sendo codificado no momento) e uma área de pesquisa de um quadro anterior, já codificado - encontrar o bloco na área de pesquisa que mais se assemelha a esse bloco atual (RICHARDSON, 2003). Para isso, a ME utiliza um algoritmo de busca que define como será realizada a busca pelo bloco mais semelhante dentro da área de pesquisa. Também é necessária uma métrica de distorção, para definir a semelhança entre blocos. Neste trabalho usaremos o SAD (*Sum of Absolute Differences*) (RICHARDSON, 2003) como métrica de distorção.

A busca é realizada dentro da área de pesquisa e quando o melhor casamento, ou seja, quando a menor diferença entre os blocos for encontrada, um vetor de movimento é gerado, indicando o deslocamento deste bloco dentro da área de pesquisa. A utilização dos vetores de movimento reduz significativamente o tamanho do vídeo codificado. Na representação final, cada bloco de pixels (que pode ter tamanhos variados, desde 4x4 pixels até 16x16 pixels), é substituído por um vetor de movimento de duas dimensões.

A ME é responsável por 80% da complexidade dos codificadores de vídeo atuais (CHENG, 2009), se tornando, assim, a parte mais importante do codificador. Portanto, o algoritmo utilizado para realizar essa busca deve ser escolhido através de alguns critérios, como complexidade computacional e qualidade resultante. A complexidade computacional pode ser medida através do número de cálculos necessários para gerar o vetor de movimento e a qualidade é medida através do PSNR (*Peak to Signal-to-Noise Ratio*) e do percentual de redução do resíduo (PRR), comparado ao simples resíduo diferencial do vídeo sem ME. O PSNR é medido em decibéis (dB), e quanto maior for o seu valor, maior a qualidade objetiva do vídeo (RICHARDSON, 2003).

O algoritmo de busca mais conhecido é o *Full Search* (FS), que consiste em uma busca exaustiva em todas as posições da área de busca e,

portanto, é um algoritmo que gera resultados ótimos, pois sempre encontrará o melhor vetor de movimento para uma dada área de busca. Todos os demais algoritmos são considerados algoritmos sub-ótimos ou algoritmos rápidos, pois utilizam uma heurística para a aceleração do cálculo dos vetores de movimento, como os algoritmos *Diamond Search* (DS), *Three Step Search* (TSS) e *One At A Time Search* (OTS), entre outros (KUHN, 1999).

Porém, de acordo com o avanço da tecnologia, os vídeos digitais passaram a ser representados em altas resoluções (HD – *High Definition*) possuindo uma grande quantidade de pixels, o que aumenta proporcionalmente a complexidade da ME. Tamanha complexidade exige o uso de algoritmos rápidos, quando são consideradas aplicações para a codificação de vídeo em tempo real. Algoritmos rápidos tendem a escolher mínimos locais como sendo o melhor resultado entre o casamento de blocos. Isto ocorre, pois os algoritmos rápidos usam heurísticas para acelerar o processo de busca, logo, a condição de parada do algoritmo pode ser atendida antes que o algoritmo convirja para a região da área de busca onde se encontra o verdadeiro melhor casamento.

Este trabalho apresenta um algoritmo rápido alternativo que visa melhorar os resultados da busca para vídeos digitais em alta definição, considerando a aplicação de várias buscas em paralelo, em regiões diferentes da área de busca.

## 2 METODOLOGIA

Os algoritmos rápidos de ME possuem algumas características em comum. Em geral, estes algoritmos seguem algum padrão de busca, iniciando no centro da área de pesquisa. Este padrão é repetido em  $n$  iterações, sempre em direção ao menor resultado de diferença (menor SAD). A alternativa discutida neste trabalho é, além de instanciar o algoritmo no centro da área de busca, fazer quatro outras instanciações do mesmo algoritmo em torno deste centro, dispostos em diferentes setores, como demonstrado na **Erro! Fonte de referência não encontrada.** Esta abordagem foi batizada de multiponto, e pode ser aplicada a qualquer algoritmo rápido de ME.

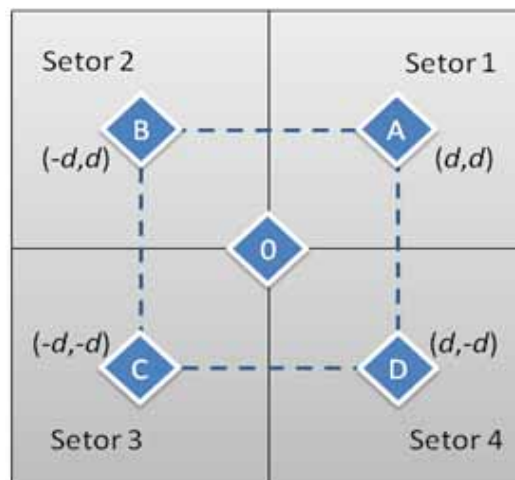


Figura 1. Algoritmos dispostos em setores na área de busca

A abordagem multiponto proposta visa diminuir a ocorrência de vetores em mínimos locais. Visto que, com a adição destes algoritmos em vários setores, mais alternativas de possíveis blocos aparecerão em cada setor, não dependendo, assim, do caminho escolhido apenas pelo algoritmo central. Além disso, este método é completamente paralelizável, onde as cinco buscas podem operar independentemente uma das outras. Desta forma, o algoritmo multiponto pode operar na mesma média de tempo de processamento do algoritmo original.

Porém, para obter melhores resultados com essa técnica, é necessário determinar a distância do centro (distância  $d$  na Figura 1) a qual os algoritmos serão aplicados. Se a distância  $d$  for muito pequena, haverá um desperdício de cálculos, pois, mais de uma busca pode acabar convergindo para o mesmo ponto. Por outro lado, caso a distância  $d$  seja muito grande, haverá um espaço grande não explorado entre a busca central e as buscas periféricas e, portanto, blocos potencialmente melhores podem ser desperdiçados. Além disso, o valor ótimo da distância  $d$  pode variar consideravelmente de acordo com as características do vídeo a ser codificado. Vídeos com pouco movimento tendem a gerar vetores pequenos, ou seja, mais próximos do centro da área de pesquisa, enquanto um vídeo com grande quantidade de movimento tem comportamento oposto.

### 3 RESULTADOS

O algoritmo de busca desenvolvido neste trabalho foi a versão multiponto do *Diamond Search*, chamado de MPDS - *Multi-Point Diamond Search*. Este algoritmo foi implementado em um software planejado para esta finalidade, na linguagem C. O algoritmo foi aplicado a 10 vídeos em alta definição (*Full HD – Full High-Definition*), com resolução de 1920x1080 pixels.

A Figura 2(a) apresenta os resultados médios percentuais de escolha dos vetores de movimento entre os cinco setores do algoritmo. Em média, 55% dos vetores são encontrados nos setores A, B, C e D e 45% dos vetores são encontrados no setor central. Isto demonstra que a maior parte dos vetores é encontrada nos setores periféricos, demonstrando a eficiência do algoritmo multiponto.

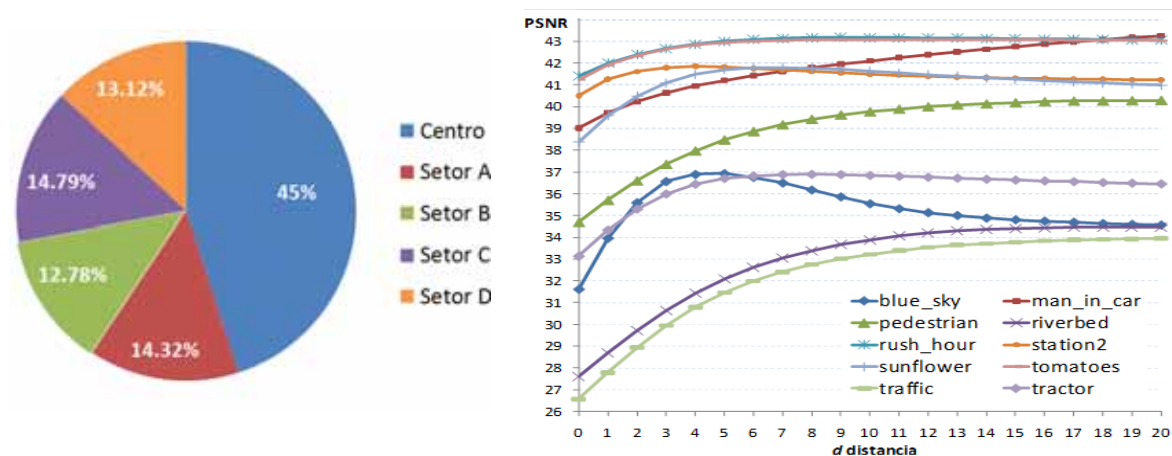


Figura 2. (a) Média de vetores escolhidos por setor e (b) variação de PSNR em relação à distância

Para aproveitar melhor o algoritmo setorizado, foram testadas várias distâncias  $d$  destes setores em relação ao centro. Foi observado que a melhor

média para estes 10 vídeos testados foi obtida para  $d = 17$ . A Figura 2(b) apresenta a curva de ganho PSNR para cada um dos 10 vídeos avaliados, considerando a variação da distância  $d$  de 0 (algoritmo original) até 20.

A Tabela 1 apresenta os resultados médios de qualidade (PSNR e PRR) e custo computacional (em número de cálculos de SAD) para os algoritmos FS, DS e MPDS. Os resultados do MPDS apresentados consideram a distância  $d = 17$ . O algoritmo MPDS apresenta um ganho de PSNR de aproximadamente 4dB, em relação ao algoritmo DS original, que é um ganho muito expressivo. Em comparação com o algoritmo FS, o MPDS só perde para a versão com área de busca de 64x64, onde o custo computacional é cerca de 28 vezes maior.

Tabela 1. Comparação de resultados com algoritmos conhecidos

Algoritmo	PSNR (dB)	PRR (%)	Nº de Cálculos x 10 <sup>6</sup>
DS	35,41	73,77	9,28
<b>MPDS</b>	<b>39,13</b>	<b>82,64</b>	<b>54,67</b>
FS16	33,62	32,48	70,09
FS32	37,94	79,77	348,78
FS64	40,78	85,02	1.543,17

#### 4 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, foi apresentada uma nova abordagem para a estimação de movimento que visa melhorar a qualidade objetiva de vídeos *Full HD* codificados. O algoritmo DS foi utilizado em uma versão multiponto, chamada MPDS - *Multi-Point Diamond Search*. O MPDS foi testado em 10 sequências de vídeo *Full HD*. Seus resultados foram comparados com o algoritmo DS original e com o algoritmo FS. Os resultados mostram que essa versão multi-ponto pode atingir um ganho médio de 3,72dB, em comparação com a sua versão original. O MPDS também possui resultados melhores que o algoritmo ótimo FS para uma área de busca de 32x32, com uma redução do custo computacional de 6,4 vezes, tendo assim, um melhor equilíbrio entre custo computacional e qualidade.

Como trabalho futuro, pretende-se adaptar outros algoritmos para a forma multiponto, a fim de permitir uma melhor avaliação da proposta e então, conduzir a comparação com algoritmos encontrados na literatura.

#### 5 REFERÊNCIAS

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

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

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