UM NOVO E EFICIENTE ALGORITMO RÁPIDO PARA A ESTIMAÇÃO DE MOVIMENTO EM VÍDEOS DE ALTA DEFINIÇÃO

DALL'OGLIO, Pargles[1]; CRISTANI, Cássio[1]; PORTO, Marcelo[2]; AGOSTINI, Luciano[3]

[1]UFPel – Curso de Ciência da Computação. Email: {pwdalloglio, crcristani@inf.ufpel.edu.br}
[2]UFPel – Centro de Artes. Email: porto@ufpel.edu.br
[3]UFPel – Centro de Desenvolvimento Tecnológico. Email: agostini@inf.ufpel.edu.br

1 INTRODUÇÃO

A demanda por vídeos de alta definição (*High Definition* - HD) vem crescendo intensamente, principalmente em aparelhos celulares, televisores e na internet. Para transmitir ou armazenar estes vídeos, torna-se necessário o estudo e o desenvolvimento de novos e eficientes algoritmos para a compressão de vídeos, para diminuir a quantidade de dados necessários para representar os vídeos, mas sem grandes perdas na qualidade das imagens geradas.

A estimação de movimento (EM) é a etapa que representa mais de 80% da complexidade computacional de um codificador de vídeo atual (PURI, 2004), no entanto, é a principal ferramenta responsável pela redução do tamanho de um vídeo.

Um vídeo digital é composto por uma sequência de imagens, denominadas quadros. Para gerar sensação de movimento, é necessário que aproximadamente 30 quadros sejam apresentados a cada segundo. Cada quadro é divido em blocos, estes são compostos por um conjunto de pixels. Em um vídeo *Full* HD, existem 1920x1080 pixels em cada quadro do vídeo.

Quadros vizinhos normalmente são muito similares e esta similaridade é denominada redundância temporal. A EM é a etapa responsável por identificar e reduzir a redundância temporal entre os quadros vizinhos. Para isso, a EM utiliza uma área de pesquisa em um quadro de referência (que foi anteriormente processado), aonde cada bloco do quadro atual é comparado com os blocos da área de pesquisa. Ao final, a EM indica o bloco com maior similaridade (melhor casamento) ao bloco atual, através de um vetor de movimento.

Um algoritmo de busca determina com que ordem os blocos serão comparados dentro da área de pesquisa. O algoritmo de busca completa (*Full Search* - FS) (RICHARDSON, 2003) é o algoritmo que apresenta a melhor qualidade, mas também possui o maior custo computacional, pois compara todos os blocos da área de pesquisa. Entre os algoritmos rápidos mais conhecidos, cita-se a busca em diamante (*Diamond Search* – DS) (ZHU, 2000), a busca em três passos (*Three Step Search* - TSS) (JING, 2004), a busca em quatro passos (*Four Step Search* - FSS) (TASDIZEN, 2009). Os algoritmos rápidos são baseados em heurísticas que visam convergir a busca mais rapidamente, assim, é possível atingir uma grande redução no custo computacional, porém, com perdas de qualidade.

A qualidade objetiva dos resultados da EM é avaliada através do PSNR (*Peak Signal-to-Noise Ratio*) (RICHARDSON, 2003). O PSNR é apresentado em decibéis (dB), em uma escala logarítmica. Assim, uma variação de 1dB no PSNR é de grande relevância.

O algoritmo DS é um dos algoritmos rápidos mais conhecidos e utilizados na literatura, pois apresenta bons resultados de qualidade e uma redução expressiva

do custo computacional. No entanto, conforme ilustrado na Figura 1, o algoritmo DS não mantém os bons resultados de qualidade com o aumento na definição dos vídeos. A heurística usada tende a concentrar os resultados no centro da área de pesquisa, ficando preso a mínimos locais. Isto faz com que as perdas em qualidade do DS, em relação ao algoritmo FS, aumentem de acordo com o aumento da resolução do vídeo. Os outros algoritmos rápidos apresentam esta mesma tendência.

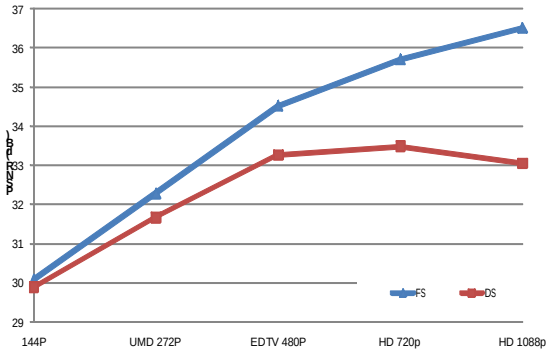

Figura 1 – Curva de qualidade entre FS e DS para diferentes resoluções.

Este trabalho apresenta um novo algoritmo rápido para estimação de movimento em vídeos de alta definição, chamado *Random Diamond Search* (RDS). O algoritmo RS é baseado em uma heurística de busca aleatória, permitindo que blocos candidatos mais distantes sejam avaliados. Além disto, o algoritmo RS utiliza uma heurística multiponto, garantindo a avaliação da região central da área de pesquisa e refinamento final dos resultados encontrados em cada ponto de busca. Assim, o RS é mais resistente a mínimos locais e apresenta ganhos de qualidade expressivos quando comparado ao DS.

2 METODOLOGIA (MATERIAL E MÉTODOS)

O algoritmo *Random Diamond Search* (RDS), proposto neste trabalho, foi desenvolvido na linguagem de programação C e avaliado para um conjunto de 10 sequências de teste Full HD.

O algoritmo RDS é composto por três etapas principais. Na primeira etapa, são sorteados *N* blocos candidatos, divididos igualmente entre os quatro quadrantes da área de pesquisa. Testes foram realizados e a diferença na qualidade de um vídeo entre vários sorteios com o mesmo *N* é desprezível, portanto, essa diferença não será considerada nesse trabalho. A segunda etapa é responsável pela execução do algoritmo DS em cada um dos quatro melhores pontos de cada quadrante, obtidos na etapa anterior. Paralelamente à segunda etapa, uma instância do DS é executada na posição central da área de pesquisa, garantindo bons resultados para vídeos de baixa movimentação. A terceira e última etapa compara os resultados obtidos pelo algoritmo DS aplicado aos melhores pontos de cada quadrante e o DS central, gerando um vetor de movimento para o melhor resultado alcançado.

Nas diversas avaliações realizadas, foram utilizadas 10 sequências de teste (XIPH.ORG, 2011), de modo a não favorecer qualquer particularidade, como baixa ou alta movimentação no vídeo. As sequências usadas são amplamente utilizadas pela comunidade da área de compressão de vídeos.

Utilizou-se a quantidade de blocos candidatos calculados (BCC) como métrica de complexidade e o PSNR (dB) para avaliar a qualidade dos vídeos.

O número de blocos candidatos sorteados (*N*) e a área de pesquisa influenciam diretamente na qualidade e na complexidade do vídeo codificado. Áreas de busca maiores podem conduzir a maiores valores de PSNR, contudo, exigem um maior valor de *N* de modo a explorar esta ampla área. O aumento no *N* produz resultados com melhor qualidade, porém, exigem maior número de comparações. Para determinar o valor ideal do *N* para cada tamanho da área de pesquisa, diversos testes foram realizados analisando a variação destes parâmetros.

A Figura 2a e 2b apresenta as curvas de qualidade e complexidade do algoritmo RDS, considerando *N* como 32, 48, 64 e 80 e as seguintes áreas de pesquisa (*ranges*): 32, 40, 48, 56 e 64. Blocos de tamanho 16x16 pixels foram utilizados para a realização dos testes.

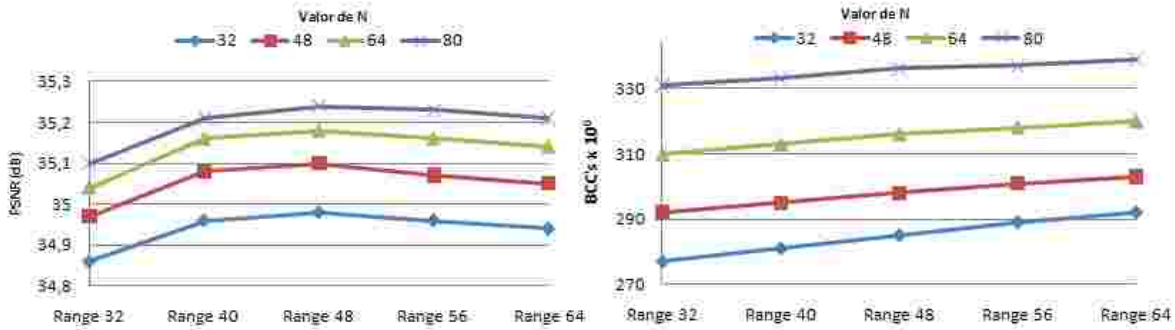

Figura 2 Avaliação dos impactos da variação no valor de N (a) na qualidade (b) na complexidade.

Através da Figura 2a é possível observar que entre as áreas 32 e 48 ocorre um acréscimo na qualidade, enquanto que a partir da área 48 até 64 ocorre uma perda em qualidade. Observando a Figura 2b percebe-se um acréscimo continuo de complexidade para todos os valores de *N*. A partir de tais observações, tomou-se uma área de 48 e um *N* igual a 48 para a geração dos resultados finais de qualidade e complexidade do algoritmo RDS.

3 RESULTADOS E DISCUSSÃO

Os resultados médios de qualidade e complexidade do algoritmo RDS estão apresentados na Tabela 1. A Tabela 1 também apresenta os resultados para o algoritmo FS e mais quatro algoritmos rápidos, MPDS, DS, FSS e TSS. O algoritmo RDS obteve o melhor resultado de qualidade em relação aos demais algoritmos rápidos avaliados. Na comparação com o algoritmo MPDS, um ganho de 0,36dB foi obtido, além de uma redução em relação à complexidade.

Tabela 1 – Comparação de qualidade e custo do algoritmo desenvolvido

Algoritmo	PSNR (dB)	BCC's x 10^6
FS	35,89	14.662,59
RDS	**35,11**	**298,13**
MPDS	34,75	307,51
DS	33,02	48,06
FSS	32,39	58,03
TSS	30,94	43,50

O algoritmo RDS obteve ainda ganhos de 2,09dB, 2,72dB e 4,17dB em relação aos algoritmos DS, FSS e TSS, respectivamente. Nestes casos, os ganhos

expressivos em qualidade também refletiram em um aumento significativo na complexidade. No entanto, se comparado com o algoritmo FS, a redução na complexidade é superior a 49 vezes.

O algoritmo RDS apresenta o resultado de qualidade mais próximo ao algoritmo FS, dentre todos os algoritmos comparados, com uma diferença de apenas 0,78 dB.

4 CONCLUSÃO

Este trabalho apresentou um novo algoritmo para estimação de movimento em vídeos de alta definição, chamado *Random Diamond Search* (RDS). O RDS é muito menos suscetível a mínimos locais, ocasionando maior ganho em qualidade no processo de codificação em relação aos demais algoritmos rápidos, especialmente em vídeos de alta resolução.

O algoritmo RDS foi desenvolvido em linguagem C e avaliado para um conjunto de 10 sequências de teste *Full* HD. O RDS obteve um PSNR apenas 0,78dB inferior ao FS, com uma redução superior a 49 vezes no número de cálculos. O RDS obteve os melhores resultados de qualidade entre todos os algoritmos rápidos investigados. Este expressivo ganho de qualidade também causou uma esperada ampliação no custo computacional.

Estes resultados são especialmente relevantes no cenário atual da área de vídeo digital, já que vídeos digitais de elevada resolução, como *Full* HD, estão cada vez mais presentes nas aplicações atuais, como televisões, tocadores de blu-ray, celulares, câmeras digitais, entre outras.

5 REFERÊNCIAS

JING, X. et al. An efficient three-step search algorithm for Block motion Estimation. **IEEE Transactions on Multimidea**, 2004.

PURI, A. et al. Video Coding Using the H.264/MPEG-4 AVC Compression Standard. **Elsevier Signal Processing: Image Communication**, 2004.

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

TASDIZEN, O. et al. Dynamically variable motion estimation algorithm and dynamically reconfigurable hardware for its implementation. **IEEE Transactions on Consumer Electronics**, 2009

XIPH.ORG. **Test Media.** Disponível em: <http://media.xiph.org/video/derf/>, Acesso em: 12 agosto de 2011.

ZHU, S. et al. A new Diamond Search Algorithm for Fast Block-Matching Motion Estimation. **IEEE Transactions on Image Processing**, 2000.