

AVALIAÇÃO DE ALGORITMOS DE ME PARA CODIFICAÇÃO DE VÍDEOS UHD NO PADRÃO HEVC

**GUILHERME PEGLOW; MATEUS NOREMBERG; CÁSSIO CRISTANI;
MARCELO PORTO; BRUNO ZATT; LUCIANO AGOSTINI**

*Universidade Federal de Pelotas
{gnpeglow, mwnoreMBERG, crCRISTANI, porto, zatt, agostini}@inf.ufpel.edu.br
Grupo de Arquiteturas e Circuitos Integrados*

1. INTRODUÇÃO

Um vídeo é uma sequência de imagens estáticas que, quando apresentadas em uma frequência aproximada de 30Hz, geram sensação de movimento. Hoje em dia, vídeos também são um popular meio de comunicação, com aplicações como vídeo conferências ou até mesmo vídeo chamadas pelo telefone celular. Vídeos digitais já representam grande parte do fluxo de dados na internet, é estimado que até 2017 os vídeos digitais sejam responsáveis por cerca de 90% do tráfego de dados na rede mundial de computadores (CISCO, 2012). Com a grande demanda do uso de vídeos, é necessário buscar novas tecnologias que facilitem o tráfego deste tipo de dados na rede. Neste contexto está inserida a codificação de vídeos, a qual permite reduzir a quantidade de informação necessária para armazenar e transmitir estes vídeos.

Exemplificando, um vídeo HD1080p (resolução 1920x1080) com 30 quadros por segundo e duração de 30 minutos, sem ser codificado, ocuparia aproximadamente 156GB de espaço em disco. Além disso, seria necessário cerca de 89Mbps no canal de comunicação para transmitir esse vídeo em tempo real, tornando difícil o tráfego desse arquivo via internet, por exemplo. O problema gerado pela quantidade de dados presente em vídeo é agravado ainda mais com o crescente aumento de resoluções suportados pelos dispositivos. Por exemplo, atualmente é possível encontrar no mercado televisores que suportem a resolução *Ultra High Definition 4K* (UHD4K – 3840x2160). Um vídeo UHD4K sem codificação, com 60 quadros por segundo e com duração de 30 minutos ocuparia cerca de 1,2 TB em disco, necessitando um canal de cerca de 705Mbps. Com estes dados é aparente a necessidade buscar novos métodos para comprimir vídeos possibilitando que esses dados sejam enviados mais facilmente.

O codificador de vídeo é responsável pela compressão de um vídeo. Esta ferramenta visa diminuir a quantidade de informação necessária para representar o vídeo ao passo que tenta manter a qualidade subjetiva do mesmo. O HEVC (*High Efficiency Video Coder*) (SULLIVAN, 2012) é o padrão estado-da-arte em codificação de vídeos, criado para substituir o H.264/AVC (WIEGAND, 2003). O HEVC apresenta uma complexidade computacional e taxa de compressão duas vezes maior que seu antecessor mantendo a qualidade de imagem. Além disso, o HEVC apresenta resultados de compressão eficientes para resoluções superiores ao HD1080p (resolução máxima suportada pelo H.264/AVC).

Dentro de um codificador de vídeo, o componente que representa o maior custo computacional é a estimação de movimento (*Motion Estimation – ME*), sendo responsável por até 80% da complexidade total do codificador. Desta forma, este módulo do codificador é o que apresenta o maior consumo energético. Com isso, é de suma importância buscar métodos de baixo custo computacional para realizar esta etapa da codificação de um vídeo. A estimação de movimento compara os pixels de um bloco de referência com os pixels de um bloco vizinho, procurando por

áreas similares ou idênticas. Quando o algoritmo de estimação de movimento encontra um bom casamento entre uma parte do quadro de referência e do quadro vizinho, um vetor de movimento é gerado. Como o bloco escolhido no quadro vizinho já foi codificado, o codificador só precisa do vetor de movimento e das diferenças entre os quadros comparados para reconstruir o bloco, mantendo a qualidade e reduzindo a informação transmitida. Quanto mais similares são os quadros comparados melhor é a compressão obtida.

Este trabalho visa encontrar uma melhor relação entre os eixos de complexidade computacional, qualidade de imagem e compressão com o uso do novo algoritmo para estimação de movimento apresentado a seguir.

2. METODOLOGIA

Neste trabalho são comparados o algoritmo presente no HEVC para estimação de movimento, o *Test Zone Search* (TZS), e o novo algoritmo de estimação de movimento, o *Random Radial Search* (RRS). O algoritmo TZS realiza um pré-processamento, onde é determinado se o melhor candidato é o bloco co-localizado (o centro da busca) ou se existem melhores candidatos em algum dos preditores, caso os preditores encontrem um bloco melhor que o bloco co-localizado. A janela de busca do TZS pode ser movida para uma nova área, ao redor do melhor candidato encontrado. Após a primeira busca, o algoritmo decide se a distância entre o centro da busca e o bom candidato é grande ou pequena, se for considerada grande, a busca *Raster* é realizada em seguida um refinamento é feito ao redor do bloco encontrado, a busca *Raster* apresenta características de um busca exaustiva, porém compara blocos entre uma distância pré-determinada, enquanto o refinamento usa uma máquina de busca em expansão com um formato geométrico, como um quadrado ou losango (ZHU,2000).

O RRS realiza uma busca usando a estratégia pseudoaleatória e divide a busca em uma série de níveis, onde a quantidade de níveis na busca é pré-definida, para os resultados mostrados a seguir neste trabalho foram usados 4 e 10 níveis de busca, indicados no nome do algoritmo como mostrados nas tabelas por exemplo RRS 10 realiza 10 níveis de busca. O algoritmo realiza a busca dentro de uma área de 32x32 pixels, para os 3 primeiros níveis de busca é realizada uma busca num formato de losango, para que seja alcançada uma densidade maior de pontos na área central da busca. Após os primeiros passos o algoritmo realiza um sorteio de pontos para realizar as comparações, o número de pontos sorteados é definido por $Nível * MultFact$ onde *Nível* é o nível atual da busca e *MultFact* é o fator de multiplicação. Para evitar que pontos sejam sorteados fora da área desejada, a janela de busca é limitada, o limite dessa área é dado por sua vez é dividido em um limite inferior e um superior. O limite inferior é dado por $2^{nível-1}$ enquanto o limite superior é dado por $2^{nível}$, assim garantindo que os blocos sorteados estarão sempre dentro da área desejada, a busca expande até que o tamanho da área de busca seja atingido.

O RRS procura entregar bons resultados de compressão e qualidade através de uma estratégia de baixo custo computacional e resistente a mínimos locais, que são responsáveis pela perda de desempenho de grande parte dos algoritmos rápidos de estimação de movimento. Esta perda de desempenho ocorre uma vez que estes não conseguem lidar com os mínimos locais como uma informação redundante. Os mínimos locais também tendem a ser mais frequentes em resoluções maiores, como visto em (PORTO, 2012). É também de muita importância buscar por estratégias que sejam de fácil implementação em hardware,

como o uso da estratégia pseudoaleatória apresentada pelo RRS, tendo em mente os desafios apresentados pela resolução UHD 4K.

Para a realização dos testes que geraram os dados apresentados a seguir neste trabalho foram usadas 4 sequencias de vídeo UHD 4K, *Cactus*, *Foreman*, *Coastguard* e *News* (ELEMENTAL,2014). Foram codificados os 100 primeiros quadros dessas sequencias de vídeo nas condições comuns de teste (BOSSSEN,2012), com o software de referencia HEVC 13.0(HEVC,2014), no modo de baixo atraso e no perfil de alta eficiência, em quatro diferentes parâmetros de quantização (QP) 22, 27, 32, 37. Os resultados são apresentados nos eixos de Bitrate, que é a quantidade de bits transferida por segundo apresentada em Mbps. O eixo PSNR (*Peak Signal Noise Ratio*) (KUHN, 1999) representa a qualidade de imagem obtida no vídeo. E por fim o eixo dado em Número de Blocos Comparados (*Compared Candidate Block - CCB*), apresentado na escala de 10^9 .

3. RESULTADOS E DISCUSSÃO

São apresentados nesta seção resultados comparando os algoritmos apresentados anteriormente. A Tabela 1 apresenta os resultados de bitrate, qualidade e número de blocos candidatos comparados para os algoritmos RRS e TZS. É possível perceber na Tabela 1 que o algoritmo RRS apresenta um custo computacional estável para as diferentes sequencias testadas. Além disso, o custo apresentado pelo algoritmo é baixo quando comparado ao custo computacional apresentado pelo TZS, tanto para Nível=10 e Nível=4. Entretanto, o TZS apresenta melhores resultados para os eixos de qualidade e compressão, as maiores diferenças estão presentes em vídeos de baixa movimentação como *Cactus* e *News*, onde ocorre a maior perda de desempenho pelo RRS.

Tabela 1: Resultados para os algoritmos RRS 4, RRS 10 e TZS

Sequencia		QP22			QP27			QP32			QP37		
		Bitrate (Mbps)	PSNR (dB)	CCB ($\times 10^9$)	Bitrate (Mbps)	PSNR (dB)	CCB ($\times 10^9$)	Bitrate (Mbps)	PSNR (dB)	CCB ($\times 10^9$)	Bitrate (Mbps)	PSNR (dB)	CCB ($\times 10^9$)
<i>Cactus</i>	RRS 4	34544	48,019	0,90	18776	46,563	0,89	9985	44,576	0,89	6685	42,050	0,88
	RRS 10	34517	48,020	1,96	18753	46,565	1,94	9959	44,577	1,93	6675	42,060	1,92
	TZS	33677	48,022	4,33	18099	46,578	3,45	9510	44,616	2,73	6372	42,109	2,12
<i>Coastguard</i>	RRS 4	141841	43,022	1,08	51168	39,727	0,97	20567	37,727	0,91	11346	36,182	0,89
	RRS 10	141841	43,022	2,34	51149	39,727	2,12	20549	37,728	1,97	11344	36,184	1,94
	TZS	141797	43,023	8,01	51079	39,726	6,63	20483	37,727	5,29	11285	36,177	4,35
<i>Foreman</i>	RRS 4	52746	46,061	0,92	24511	44,210	0,90	12013	42,028	0,89	7549	39,653	0,89
	RRS 10	52696	46,060	2,01	24485	44,208	1,96	11989	42,028	1,94	7535	39,659	1,93
	TZS	51929	46,059	5,20	23771	44,208	4,19	11530	42,039	3,34	7271	39,683	2,59
<i>News</i>	RRS 4	31287	51,827	0,91	18028	48,720	0,90	10017	45,197	0,89	6811	41,991	0,89
	RRS 10	31292	51,826	1,97	18022	48,720	1,96	10014	45,205	1,94	6810	41,991	1,93
	TZS	31156	51,832	4,44	17931	48,719	3,76	9957	45,204	3,12	6769	41,992	2,58
<i>Média</i>	RRS 4	65104	47,233	0,95	28121	44,805	0,92	13145	42,382	0,90	8098	39,969	0,89
	RRS 10	65087	47,232	2,07	28102	44,805	1,99	13128	42,384	1,95	8091	39,974	1,93
	TZS	64640	47,234	5,49	27720	44,808	4,51	12870	42,397	3,62	7924	39,990	2,91

4. CONCLUSÕES

Com o trabalho apresentado é possível concluir que novos métodos para codificar vídeos UHD 4K são necessários, uma vez que os meios para trafegar esses dados não suportariam tamanha quantidade de dados. Além disso, verificou-se que as estratégias adotadas necessitam apresentar baixo custo computacional e serem de fácil implementação em hardware, isto porque que a estimação de movimento apresenta o maior custo computacional dentro de um codificador de vídeo e conseqüentemente, o maior consumo energético.

O algoritmo RRS apresenta de fato baixo custo computacional, porém, devido a característica de sua busca, que apenas sorteia blocos novos independente de ter encontrado candidatos bons ou não, o algoritmo acaba perdendo desempenho para vídeos de baixa movimentação, uma vez que uma área de alto potencial pode deixar de ser explorada. Uma possível solução seria refinamento ao algoritmo, ou seja, uma vez que o algoritmo encontrasse um bom candidato, o mesmo deveria ser capaz de explorar a área ao redor do bloco encontrado.

Mesmo com o aumento do número de sorteios, o algoritmo RRS não obteve melhoras significativas para os casos de sequencias de baixa movimentação. O algoritmo TZS, que apresenta busca refinada, é capaz de obter melhores resultados gerais do que o RRS, perdendo apenas no eixo da complexidade, o que pode ser também explicado pela estratégia do TZS. Tendo em mente que o uso da busca *Raster* apresenta um alto custo computacional, e que a possibilidade da movimentação da janela de busca também pode gerar perdas de desempenho do algoritmo.

5. REFERÊNCIAS BIBLIOGRÁFICAS

CISCO. **Cisco Visual Networking Index: Forecast and Methodology**, 2012–2017. Cisco Public, 2013.

KUHN, P.; **Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation**. Boston: Kluwer Academic Publishers, 1999.

SULLIVAN, G.J.; Ohm, J.; Woo-Jin Han; Wiegand, T., Overview of the High Efficiency Video Coding (HEVC) Standard, *Circuits and Systems for Video Technology*, **IEEE Transactions on**, vol.22, no.12, pp.1649,1668, Dec. 2012.

WIEGAND, T.; SULLIVAN, G.; BJONTEGAARD, G; e LUTHRA, A.; Overview of the H.264/AVC video coding standard. *Circuits and Systems for Video Technology*, **IEEE Transactions on**, vol.13, no.7, pp.560-576, Jul. 2003.

BOSSEN, F., Common test conditions and software reference configurations. **Doc. JCTVC-C500**, JCT-VC of ITU-T SG16 WP3, 2012.

ZHU, S., MA, K., A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation. In: **IEEE Transactions on Image Processing**, Vol. 9, No. 2, pp. 287-290, 2000.

PORTO, M.S. **Desenvolvimento Algorítmico e Arquitetural para a Estimação de Movimento na Compressão de Vídeo de Alta Definição**. Fevereiro de 2012. Tese (Doutorado em Ciência da Computação) – Curso de Pós-Graduação em Ciência da Computação – Universidade Federal do Rio Grande do Sul.

(2014) Elemental Technologies. Disponível em: <http://www.elementaltechnologies.com>

(2014) HEVC Reference Software. Disponível em: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/