

## UMA ESTRATÉGIA PARA REDUÇÃO DO TEMPO DE CODIFICAÇÃO DE VÍDEOS NO PADRÃO HEVC

HENRIQUE MAICH; MATEUS MELO; JONES GÖEBEL; LUCIANO AGOSTINI;  
BRUNO ZATT; MARCELO PORTO

*Universidade Federal de Pelotas – Grupo de Arquiteturas e Circuitos Integrados  
{hdamaich, msdmelo, jwgoebel, agostini, zatt, porto}@inf.ufpel.edu.br*

### 1. INTRODUÇÃO

Atualmente, reprodução e gravação de vídeos digitais tem se tornado cada vez mais comum devido a popularização de dispositivos capazes de manipular eles como smartphones. Além da popularização, surgiu também uma demanda por maior qualidade de vídeos devido a novas tecnologias de reprodução como televisores/monitores em altas resoluções. Devido a essas novas resoluções o armazenamento de vídeos digitais se tornou um desafio para dispositivos móveis, como os próprios smartphones, pois vídeos digitais em alta resoluções demandam uma quantidade grande de dados para ser armazenado sem compressão.

Desta forma, um esforço pela comunidade em codificação de vídeo foi feito para criação de um conjunto de ferramentas para compressão de vídeos, chamado de padrões de codificação, onde o último criado foi o HEVC (*High Efficiency Video Coding*) (ITU, 2013). Este padrão tem como objetivo trabalhar com ultra alta resoluções (4K e 8K) de forma eficiente e busca ter maiores compressões para uma mesma qualidade que seus antecessores. Contudo, o esforço em obter altas taxas de compressão também trouxe uma alta complexidade computacional, assim, o padrão HEVC também se tornou mais complexo que os antecessores, criando um desafio para aplicações em dispositivos móveis, que apresentam restrições de taxa de processamento.

Todavia, os processadores embarcados nos dispositivos móveis recentes possuem GPUs integradas, onde é possível acelerar diferentes tipos de processo. Em geral, essas GPUs suportam a programação na linguagem OpenCL (KHRONOS, 2011), onde é possível programar algoritmos paralelos para processamento de dados. Desta forma, utilizando essas GPUs, é possível acelerar processos do codificador HEVC, reduzindo assim o tempo gasto para codificar os vídeos digitais. A Estimação de Movimento (*Motion Estimation – ME*) é a etapa que utiliza a maior fatia de tempo no processamento de vídeos no HEVC (PURNACHAND, 2012), onde ela busca explorar a redundância temporal entre os quadros dos vídeos. Essa exploração busca através de algoritmos de busca procurar em quadros já codificados informações para representar o quadro que está sendo codificado. O processo de busca dentro dos codificadores atuais é dado no nível de bloco, onde os quadros são divididos em blocos de diversos tamanhos, para uma busca mais refinada dependendo da textura do vídeo e usam um critério de seleção de melhor bloco baseado em resíduo e quantidade de dados.

No HEVC os blocos utilizados na ME podem assumir até 24 tamanhos diferentes, variando de 4x8 ou 8x4 até 64x64 (ITU, 2013), utilizando uma estrutura em árvore em níveis de tamanho de blocos. Também na ME, o critério para seleção do melhor bloco candidato mais utilizado é o SAD (*Sum of Absolutes Differences*), onde soma a diferença entre todos os pixels dos blocos testados e o bloco que está sendo codificado.

Este trabalho então busca através de configurações e algoritmos de estimação de movimento reduzir a complexidade dessas tarefas medindo o impacto em qualidade e compressão.

## 2. METODOLOGIA

Para o desenvolvimento do algoritmo de Estimação de Movimento paralelo, para processamento em GPU, foi utilizado a linguagem de programação OpenCL, pois diversos tipos de GPUs dão suporte a ela, dentre elas as disponíveis nos dispositivos moveis atuais. A programação de algoritmos eficientes para GPUs requer que os dados de entrada sejam disponibilizados na forma mais regular possível. Desta forma, o algoritmo de ME escolhido para desenvolvimento foi o *Full Search* (FS), algoritmo que busca dentro de uma região dos quadros passados todos os casamentos possíveis entre os blocos candidatos. Este algoritmo também foi escolhido porque sempre encontra o melhor bloco candidato, assim, melhorando a taxa de compressão e qualidade do vídeo codificado.

Contudo, o FS também o algoritmo com maior complexidade devido a alta quantidade de blocos candidatos comparados. Desta forma, para reduzir a complexidade do algoritmo, sem prejudicar a regularidade, o número de tamanhos de blocos testados foi limitado a apenas 4 entre os 24 tamanhos. Dentre os possíveis tamanhos, foram escolhidos os blocos quadráticos: 8x8, 16x16, 32x32 e 64x64, pois estes tamanhos mantém a maior regularidade e também são os blocos base dentre os níveis da árvore. Além disso, trabalhos anteriores verificaram que estes são os blocos mais selecionados pelo HEVC (MAICH, 2014).

Além desta otimização dos tamanhos de blocos, o algoritmo desenvolvido utiliza uma estratégia de acumulação para reduzir o número de SADs calculados, onde apenas os SADs de blocos 4x4 são calculados, sendo que os demais tamanhos são gerados a partir da associação de tamanhos de blocos. O tamanho de 4x4 foi utilizado como menor grão para calcular os SADs, pois a partir dele é possível gerar qualquer tamanho de bloco definido pelo HEVC através da estratégia de associação. Outra otimização realizada foi a redução de quadros utilizados como referência pela ME. O padrão HEVC usa, como padrão, quatro quadros de referência na ME, no entanto, neste trabalho o número de quadros de referência foi limitado a apenas um. O quadro de referência utilizado pelo algoritmo sempre é o ultimo quadro codificado, pois normalmente este é o mais similar ao quadro que está sendo codificado, assim encontrando melhores casamentos.

Para validar o algoritmo desenvolvido o software de referência (HM – HEVC Model) do HEVC foi utilizado, onde o algoritmo desenvolvido substituiu o *Test Zone Search*, usado como base para experimentos com o HM. O objetivo dos experimentos foi verificar o impacto na qualidade e compressão dos vídeos utilizando a métrica BD-Rate (BJONTEGAARD, 2008) que avalia a variação de compressão para uma mesma qualidade de vídeo e o impacto das otimizações no tempo de execução. Desta forma, dois experimentos foram realizados, onde o primeiro buscou encontrar o impacto em utilizar apenas um quadro de referência para a ME, e o segundo buscou avaliar o impacto das modificações na HM.

Para a execução dos experimentos foram utilizadas 7 sequencias de vídeos, baseados nas sequencias recomendada para teste pelos desenvolvedores do padrão HEVC (BOSSEN, 2012). Foram escolhidas as sequências de maior resolução, sendo 2 em resolução WQXGA (2560x1600 pixels) e 5 Full HD

(1920x1080 pixels). A escolha de sequências de alta resolução se justifica, pois, como o processo de codificação nessas resoluções é mais complexo, a aceleração deste processo acaba por ser mais relevante.

### 3. RESULTADOS E DISCUSSÃO

Ambos experimentos realizados por esse trabalho foram executados em um computador com processador Intel Core i7-3770 @ 3.4 GHz e uma GPU NVidia GeForce GT 630 com suporte à programação OpenCL 1.1. A versão utilizada do software de referência do padrão HEVC foi o HM 16.0.

O resultado do primeiro experimento verificou o impacto em tempo e BD-Rate ao se utilizar apenas um quadro de referência, onde podemos verificar os resultados na Tabela 1.

Tabela 1 - Variação de BD-Rate e tempo em se usar apenas um quadro.

Vídeo	Resolução	BD-Rate (%)	Redução do Tempo (%)
<b>Traffic</b>	WQXGA (2560x1600 pixels)	37,22	40,55
<b>PeopleOnStreet</b>		4,00	44,72
<b>Kimono</b>		5,37	50,08
<b>ParkScene</b>		21,92	41,34
<b>Cactus</b>		25,42	41,03
<b>BQTerrace</b>		63,17	34,12
<b>BasketballDrive</b>		8,71	47,19
<b>Média</b>		23,69	42,72

Nos resultados apresentados pela tabela podemos notar que o experimento um obteve uma alta taxa de redução do tempo de processamento para a maioria dos vídeos selecionados, obtendo resultados acima dos 40%. A exceção é o vídeo BQTerrace que obteve menor variação de tempo, devido ao fato do vídeo ter muita variação de textura, onde a redução do número de quadros de referência acabou reduzindo muito a possibilidade de encontrar um bom casamento, desta forma, exigindo um esforço maior para a compressão. Pelos mesmos motivos, é possível perceber que BQTerrace e Traffic foram os vídeos que tiveram a maior degradação da compressão para uma mesma qualidade (BD-Rate).

Já em um segundo experimento testou o quanto foi possível reduzir utilizando o algoritmo de estimativa de movimento acelerado, usando como base o primeiro experimento. O resultado desse experimento é mostrado na Tabela 2.

Tabela 2 - Variação de BD-Rate e tempo da estratégia proposta pelo artigo.

Vídeo	Resolução	BD-Rate (%)	Redução do Tempo (%)
<b>Traffic</b>	WQXGA (2560x1600 pixels)	1,21	2,42
<b>PeopleOnStreet</b>		2,07	1,22
<b>Kimono</b>		1,00	-2,39
<b>ParkScene</b>		1,05	-1,92
<b>Cactus</b>		1,55	2,49
<b>BQTerrace</b>		1,34	1,70
<b>BasketballDrive</b>		2,00	3,27
<b>Média</b>		1,46	0,97

Nesta tabela podemos verificar que a aceleração do algoritmo trouxe ainda mais redução do tempo de codificação, porém quase insignificativa comparado com o primeiro experimento, porém destaca-se ainda mais perdas em compressão para

uma mesma qualidade. Desta forma, esse resultado indica que o somente a modificação em reduzir os quadros de referência seja o suficiente para obter uma redução eficiente no tempo de codificação, pois está tem uma melhor troca entre perdas na compressão, versus redução do tempo de codificação. Porém, como o segundo experimento é um resultado inicial talvez a redução do tempo de codificação possa ter sido mascarada entre processos alternativos executados pelo sistema operacional no momento da execução do experimento.

#### 4. CONCLUSÕES

Este trabalho desenvolveu uma estratégia para redução do tempo de codificação de vídeos do padrão HEVC, devido ao alto tempo necessário para processar um vídeo em alta resoluções nos processadores comerciais. A estratégia desenvolvida foi composta em uma redução da quantidade de quadros de referência, redução do número de tamanhos de blocos para ME e o desenvolvimento de um algoritmo de ME utilizando uma abordagem paralela para processamento em GPU. Para o desenvolvimento do algoritmo foi utilizado a linguagem OpenCL, devido à alta compatibilidade com diversas placas.

Além disso, esse trabalho avaliou, através do software de referência, o impacto da estratégia desenvolvida, onde foi verificado uma alta redução do tempo de codificação, porém, com perdas na compressão do vídeo para uma mesma qualidade.

Como trabalho futuro, será melhor explorado a possibilidade de paralização do algoritmo utilizando as GPU, avaliando melhores conjunto de modificações para se obter menores perdas em compressão e mantendo a redução de tempo de codificação. Para isso, novos experimentos, como 2, 3 ou 4 quadros de referência serão realizados, reduzindo assim as perdas gerada por essa modificação e aumentando o potencial de ganho de aceleração do algoritmo desenvolvido.

#### 5. REFERÊNCIAS BIBLIOGRÁFICAS

- BJONTEGAARD, G. Improvements of the BD-PSNR model, **VCEG-AI11**, July 2008.
- BOSSEN, F. **Common Test Conditions and Software Reference Configurations**. Outubro, 2012. Acessado em 1 out. 2013. Online. Disponível em: [http://phenix.it-udparis.eu/jct/doc\\_end\\_user/current\\_document.php?id=6469](http://phenix.it-udparis.eu/jct/doc_end_user/current_document.php?id=6469)
- ITU - International Telecommunication Union. **Recommendation ITU-T H.265: High efficiency video coding**. Abril, 2013. Acessado em 1 out. 2013. Online. Disponível em: <http://www.itu.int/rec/T-REC-H.265-201304-I>
- KHRONOS. **The OpenCL Specification. Revision**. 2011. ed. 44.
- MAICH, Henrique, et al. **HEVC Fractional Motion Estimation complexity reduction for real-time applications**. IEEE 5th Latin American Symposium on Circuits and Systems (LASCAS), 2014.
- PURNACHAND, N.; ALVES, Luis Nero; NAVARRO, Antonio. **Fast motion estimation algorithm for HEVC**. IEEE International Conference on Consumer Electronics-Berlin (ICCE-Berlin). 2012. p. 34-37.