

## AVALIAÇÃO DO PARTICIONAMENTO DE QUADROS USANDO TILES PARA A COMPRESSÃO DE VÍDEO PARALELA SEGUNDO O PADRÃO HEVC

IAGO STORCH; DANIEL PALOMINO; LUCIANO AGOSTINI

Universidade Federal de Pelotas

Grupo de Arquiteturas e Circuitos Integrados  
*{icstorch, dpalomino, agostini}@inf.ufpel.edu.br*

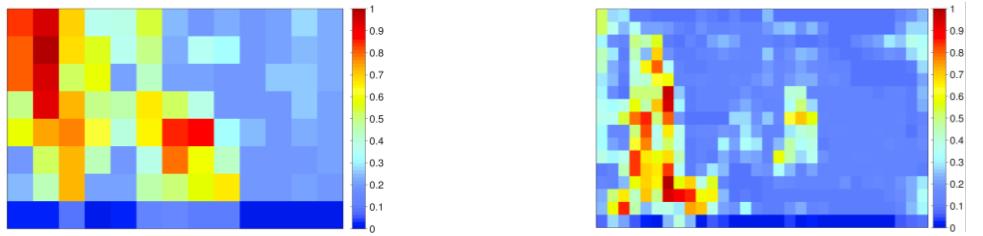
### 1. INTRODUÇÃO

Com os avanços tecnológicos, vídeos de alta resolução se tornaram algo comum no nosso cotidiano. Vários dispositivos e serviços são capazes de reproduzir/transmitir vídeos de alta definição HD720p (1280x720 pixels) e HD1080p (1920x1080 pixels). No entanto, mesmo os vídeos de resolução menor, quando estão no seu formato original são muito grandes para serem manipulados ou transmitidos pelas tecnologias disponíveis e nesse cenário surge a compressão de vídeo.

A compressão de vídeo é realizada através da exploração de várias redundâncias de dados existentes nos vídeos digitais, e esse processo tem uma alta complexidade computacional (Sullivan et al, 2012). O padrão H.264/AVC era o que existia de melhor entre as tecnologias de compressão, entretanto, com a crescente demanda por altas resoluções, o grupo *Joint Collaborative Team on Video Coding (JCT-VC)* propôs o padrão *High Efficiency Video Coding (HEVC)*, que incorporou novas ferramentas de compressão e através disso foi capaz de reduzir de 36% a 70% a taxa de bits quando comparado com o padrão anterior (OHM, J. et al, 2012). Entretanto, a complexidade computacional do novo padrão aumentou em relação ao H.264/AVC (OHM, J. et al, 2012).

Uma possível solução para resolver o problema do aumento da complexidade computacional gerada pelas novas ferramentas de compressão é o processamento paralelo. Atualmente, múltiplas unidades de processamento estão cada vez mais populares e presentes em qualquer tipo de dispositivo (NVIDIA, 2010). A ideia principal do processamento paralelo é dividir uma dada tarefa em **N** sub-tarefas, para que estas possam ser processadas simultaneamente pelas unidades de processamento, e assim diminuir o tempo de execução da tarefa. Dentre as novas ferramentas do padrão HEVC está o *Tile* que é utilizado para explorar esse cenário de processamento paralelo popularizado. Os *tiles* criam arranjos retangulares formados pelas unidades básicas de codificação, as *Coding Units (CUs)*, de modo que cada arranjo possa ser processado independentemente, e assim, reduzir o tempo de compressão proporcionalmente ao número de *tiles* (CHI, C. et al, 2012).

Entretanto, quando o processamento em questão é a compressão de vídeo, dividir a quantidade de dados igualmente não significa reduzir a carga de trabalho na mesma proporção. Como dito anteriormente, a compressão é feita através da exploração de semelhanças dentro do vídeo, logo, áreas com muita textura e movimento geram uma carga de trabalho maior em relação a áreas homogêneas e estáticas. A Figura 1 mostra a divisão de carga de trabalho para os vídeos *PartyScene* (832x480 pixels) e *BasketballDrive* (1920x1080 pixels), onde cada um dos retângulos representa a carga de trabalho (normalizada) para uma determinada CU. É possível notar que para ambos os vídeos existem áreas em que há uma grande concentração da carga de trabalho, enquanto em outras essa concentração é reduzida. Considerando um contexto onde esses quadros seriam divididos (por exemplo, em quatro partes iguais) para serem processados em paralelo, é possível notar que a divisão de carga de trabalho não estaria igualmente representada em cada uma das partes do vídeo.



(a) *PartyScene* - quadro 3

(b) *BasketballDrive* - quadro 1

Figura 1. Carga de trabalho para cada CU dos vídeos.

O objetivo deste trabalho é avaliar diferentes possibilidades de particionamento dos quadros a serem comprimidos por um compressor HEVC e seu impacto na divisão de carga de trabalho entre os *tiles* que serão processados. Com uma divisão de carga de trabalho mais homogênea é possível realizar a compressão paralela de um vídeo em menos tempo.

## 2. METODOLOGIA

Este trabalho visa avaliar diferentes particionamentos de quadros a serem comprimidos segundo o padrão HEVC e verificar quais particionamentos geram melhores distribuições de carga de trabalho entre os *tiles*, de modo a otimizar o desempenho quando processados num ambiente paralelo. Para fazer tais avaliações, foram considerados dois conjuntos de vídeos: o Conjunto A indica os vídeos de resolução 832x480, e o Conjunto B os de resolução 1920x1080. A Figura 2 mostra um fluxograma onde as etapas utilizadas para realizar a avaliação proposta neste trabalho estão detalhadas.

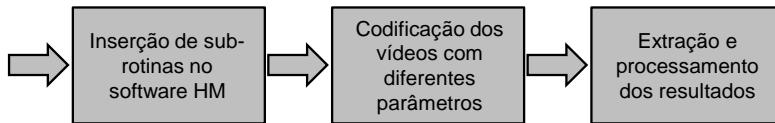
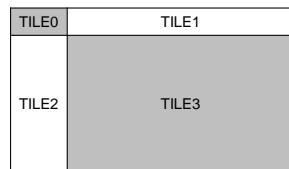


Figura 2. Fluxograma da metodologia

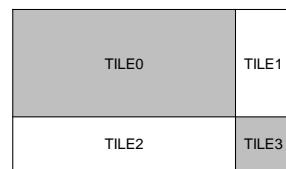
Primeiramente, utilizamos o *HEVC Test Model (HM)*, na versão 16.0 (JCT-VC, 2013), que é um software que implementa as ferramentas de compressão do padrão HEVC. Foram inseridas algumas sub-rotinas dentro do software HM para a medição do tempo de compressão de cada *tile*. Em seguida, foi criado um *script* utilizando a linguagem de programação *python* para executar o HM modificado tantas vezes quanto necessárias. Neste trabalho o particionamento que será avaliado será o 2x2 com quatro *tiles*. A divisão dos *tiles* foi alterada de todas as formas possíveis, e foram executados 60 quadros de sete vídeos diferentes, incluindo resoluções 832x480 (Conj. A) e 1920x1080 (Conj. B). Assim, obtivemos uma vasta gama de resultados, a partir de um diversificado conjunto de dados. A Figura 3 mostra três tipos de particionamento 2x2 onde o particionamento (a) é o uniforme (partições com o mesmo tamanho) e os particionamentos (b) e (c) são exemplos de não uniformes.



(a) Uniforme



(b) Não Uniforme



(c) Não Uniforme

Figura 3. Diferentes particionamentos em tiles 2x2.

Com os tempos de compressão para cada *tile*, foi avaliado como a divisão dos tiles alterou o tempo de compressão dos mesmos, e quais divisões resultaram numa divisão da carga de trabalho mais homogênea. Para isso, foi

utilizada uma métrica chamada de *fator de SpeedUp* que foi aplicado a cada uma das possíveis divisões. Essa métrica mede, em uma escala de 1 ao número de *tiles*, o quanto próximo da divisão ideal está uma divisão em questão. A divisão ideal é aquela onde a carga de trabalho é dividida igualmente entre os *tiles*, e a pior divisão é aquela em que todo o trabalho está concentrado num único *tile*. A equação 1 mostra como o *SpeedUp* é calculado onde  $n$  representa o número de *tiles* e o  $T_{tile_i}$  é o tempo de compressão do  $i$ -ésimo *tile*.

$$SpeedUp = \frac{\sum_{i=0}^{i=n-1} T_{tile_i}}{\max(T_{tile_0}, \dots, T_{tile_n})} \quad (1)$$

Para o processamento dos resultados foi utilizada uma planilha eletrônica com os tempos de compressão para todos os casos avaliados. Gráficos de linha foram gerados para visualizar os resultados dos diferentes testes, e em seguida, foi desenvolvido um software, na linguagem de programação *python*, para processar os tempos de compressão e calcular o *SpeedUp* para cada uma das divisões de quadro. Por fim, com o auxílio do software MATLAB, esses *SpeedUp*s foram mapeados para gráficos de calor, que indicam as divisões mais eficientes.

### 3. RESULTADOS E DISCUSSÃO

Para facilitar o entendimento dos resultados obtidos com este trabalho, foram gerados alguns gráficos a partir de três avaliações de diferentes características do processamento. A primeira delas teve como objetivo criar uma referência de como a carga de trabalho está distribuída entre as unidades de processamento quando os *tiles* são divididos uniformemente. A (a) Figura 4 mostra a relação entre a carga de trabalho de um determinado *tile* e a carga total do quadro para o particionamento uniforme.

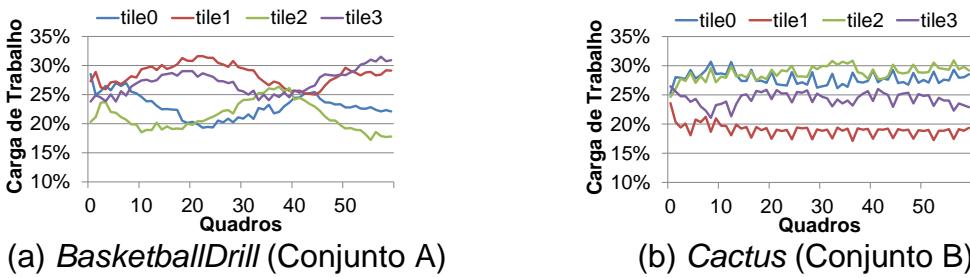
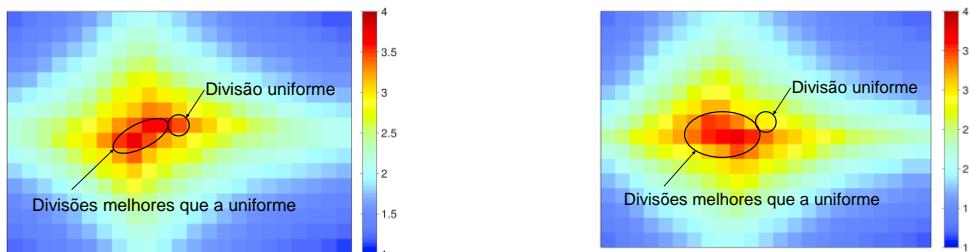


Figura 4. Carga de trabalho relativa por *tile*

Analizando os gráficos, é possível notar que ao aplicarmos uma divisão uniforme, alguns *tiles* levam uma quantidade de tempo substancialmente maior que os demais para serem comprimidos, e esses *tiles* podem, inclusive, se alterar ao longo do vídeo, criando assim uma característica prejudicial ao processamento paralelo, pois para que um novo quadro seja processado, todos os *tiles* do quadro anterior precisam ter sido comprimidos.

Em seguida, com base nos tempos de compressão por *tile*, foi possível calcular o *SpeedUp* para todas as possíveis divisões, e a partir destes dados, foram gerados gráficos de calor para visualizar o comportamento do *SpeedUp* para divisões ao longo do quadro. A Figura 5 mostra esses gráficos onde cada uma das células representa o *SpeedUp* para um ponto de corte, ou seja, uma interseção entre o corte horizontal e o corte vertical (ver Figura 3).

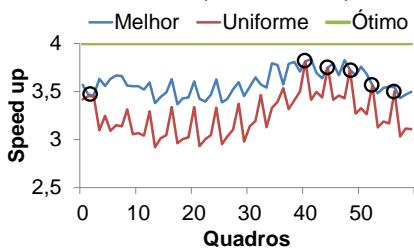
É possível notar que a região com as melhores divisões não se encontram necessariamente no centro do quadro, ou seja, existem divisões não uniformes que apresentam valores de *SpeedUp* maiores do que a divisão uniforme. Isso significa que é possível atingir melhores resultados buscando divisões que não sejam a uniforme.



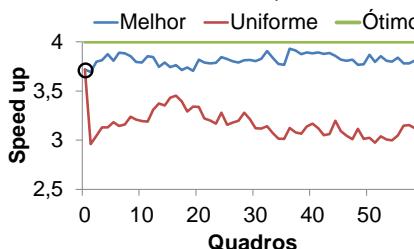
(a) *Cactus* (Conj. B - 5º quadro) (b) *BasketballDrive* (Conj. B - 2º quadro)

Figura 5. Variações do *SpeedUp* ao longo do quadro

Para ter uma visão mais completa em relação ao potencial das divisões não uniformes em relação a uniforme, a Figura 6 mostra dois gráficos que apresentam o *SpeedUp* que seria ótimo (linha verde), o melhor *SpeedUp* para um determinado vídeo (linha azul) e o *SpeedUp* da divisão uniforme (linha vermelha).



(a) *PartyScene* (Conjunto A)



(b) *BasketballDrive* (Conjunto B)

Figura 6. *SpeedUps* das divisões ótimo, melhor e uniforme

A divisão uniforme raramente é a melhor divisão possível (círculos pretos), e para a maioria dos quadros a divisão uniforme tem um desempenho inferior à melhor divisão possível, chegando a ter um *SpeedUp* 22% menor. Além disso, é possível notar que existe um espaço considerável entre a divisão uniforme e a melhor divisão. Esse espaço pode ser explorado através de técnicas que busquem divisões melhores que a uniforme em termos de *SpeedUp*.

## 4. CONCLUSÕES

Este trabalho apresentou uma análise de como a utilização da ferramenta de *tiles* pode influenciar no emprego de processamento paralelo para compressão de vídeo. Com os resultados da avaliação é possível concluir que o modo como a ferramenta divide os quadros na versão padrão do software não é a mais eficiente em termos de *SpeedUp*. Sendo assim, nós propomos que se busquem soluções para melhorar a eficiência da utilização de *tiles* através de divisões mais adequadas e assim melhorar o desempenho da compressão de vídeo paralela.

## 5. REFERÊNCIAS BIBLIOGRÁFICAS

- Sullivan et al. Overview of the High Efficiency Video Coding (HEVC) Standard. **IEEE Transactions on Circuits and Systems for Video Technology**. [S.I.:s.n], v. 22 n. 12, p.1649-1668, 2012.
- OHM, J. et al. Comparison of the Coding Efficiency of Video Coding Standards - Including High Efficiency Video Coding (HEVC). **IEEE Transactions on Circuits and Systems for Video Technology**. [S.I.:s.n.], v. 22 n. 12, p.1669-1684, 2012.
- NVIDIA. **The Benefits of Multiple CPU Cores in Mobile Devices**. [S.I.]. 2010.
- CHI, C. et al. Parallel Scalability and Efficiency of HEVC Parallelization Approaches. **IEEE Transactions on Circuits and Systems for Video Technology**. [S.I.:s.n.], v. 22 n. 12, p. 1827-1838, 2012.
- JCT-VC. High Efficiency Video Coding text specification draft 10, doc. JCTVC-L1003, Genebra, Suíça, 2013.