

Avaliação do potencial máximo de *speedup* usando *Tiles* para compressão de vídeo paralela segundo o padrão HEVC

Iago Storch, Daniel Palomino, Bruno Zatt, Luciano Agostini

Universidade Federal de Pelotas

Centro de Desenvolvimento Tecnológico, Centro de Engenharias

Grupo de Arquiteturas e Circuitos Integrados

{icstorch, dpalomino, zatt, agostini}@inf.ufpel.edu.br

Resumo—Este artigo apresenta uma análise da paralelização na compressão de vídeo de acordo com o padrão HEVC utilizando *Tiles*. Diferentes configurações de *Tiles* são avaliadas considerando duas variáveis principais: (1) *speedup* e (2) eficiência de compressão. Primeiro discutimos como as características do conteúdo podem influenciar na divisão da carga computacional utilizando a configuração de *Tiles* uniformes. Em seguida, vários vídeos são comprimidos utilizando diferentes configurações de *Tiles* para analisar o *speedup* resultante junto com a eficiência de compressão. Como conclusão, mostramos que existe potencial para melhorar em até 18,9% o *speedup* da compressão de vídeo utilizando o HEVC paralelizado através do uso de diferentes configurações de *Tiles* causando uma variação desprezível na eficiência de compressão.

Palavras-Chave—paralelismo, compressão, vídeo, *tiles*.

Abstract—This work presents an analysis of parallel HEVC video coding using *Tiles*. Different tiling configurations are evaluated considering two main variables: (1) *speedup* and (2) coding efficiency. First, we discuss how content properties can affect workload distribution using the uniform tiling configuration. Then, several sequences are encoded with different tiling configurations in order to analyze the resulted *speedup* and the coding efficiency. As conclusion, we show that there is potential to increase in up to 18.9% the *speedup* of parallel HEVC video coding by using different tiling configurations. Moreover, the coding efficiency effects are negligible.

Keywords—parallel video coding, *tiles*, HEVC.

I. INTRODUÇÃO

Devido aos últimos avanços tecnológicos, vídeos digitais estão se tornando cada vez mais comuns no dia a dia das pessoas. Muitos dispositivos e serviços já são capazes de reproduzir e transmitir vídeos de alta resolução. O padrão H.264 [1] era o estado-da-arte em compressão de vídeo, no entanto, com as novas resoluções *Ultra High Definition* (UHD - 4k e 8k) [2] um novo padrão de compressão de vídeo se fez necessário visto a necessidade de aumentar as taxas de compressão com esses novos vídeos de ultra definição. Nesse contexto, o padrão *High Efficiency Video Coding* (HEVC) foi desenvolvido com novas ferramentas de compressão mais eficientes. Essas ferramentas possibilitaram alcançar até 70% de ganho em compressão quando comparado com o H.264 [2]. Entretanto, esse aumento de eficiência de compressão veio ao custo de um grande aumento no esforço computacional [2].

Sendo assim, é importante explorar soluções que lidam com esse aumento de esforço.

Uma das maneiras de melhorar o desempenho computacional do processo de compressão de vídeo são as técnicas de compressão de vídeo paralela que já foram propostas nos últimos padrões de compressão de vídeo. Os *Slices* do padrão H.264 foram uma das primeiras ferramentas de paralelização utilizadas para compressão de vídeo. Entretanto, os *Slices* foram inicialmente propostos para lidar com problemas de sincronização de pacotes em transmissões de vídeo em rede, o que acaba gerando um *overhead* na quantidade de bits no cabeçalho de cada *Slice*. Além disso, não existe completa independência de dados entre os diferentes *Slices* o que impossibilita processamento completamente paralelo.

Com o novo padrão HEVC, a ferramenta chamada *Tile* foi introduzida com o principal objetivo de permitir processamento paralelo completamente independente [3]. *Tiles* são unidades de compressão paralela que consistem em partições retangulares criadas a partir da divisão dos quadros que compõe o vídeo digital por bordas verticais e horizontais pré-definidas [3]. Cada *Tile* é composto por um conjunto de *Coding Tree Units* (CTUs - 64x64 pixels) e cada *Tile* é comprimido de forma completamente independente. Um vídeo pode ser particionado em *Tiles* de forma uniforme ou não-uniforme. Quando o particionamento uniforme é escolhido, o número de CTUs é dividido da maneira mais igualitária possível entre os *Tiles*. Já com o particionamento não uniforme, o compressor pode ser configurado para definir as posições específicas das bordas dos *Tiles* sem necessariamente considerar uma divisão igualitária. Enquanto que para grande parte das aplicações, dividir a quantidade de dados igualmente entre as diferentes *threads* significa divisão igualitária de carga computacional, na compressão de vídeo isso pode não ser verdade, pois existe uma relação direta entre o esforço computacional para realizar a compressão de uma determinada região do vídeo e as características do conteúdo do vídeo nessa região. Deste modo, é importante que exista uma avaliação do potencial máximo de *speedup* considerando a compressão de vídeo paralela usando as diferentes possibilidades de particionamento dos *Tiles* não uniformes.

Existem alguns trabalhos na literatura que também focam em compressão de vídeo paralela para melhorar a performance através de diferentes técnicas. Os autores em [4] apresentam

uma visão geral de como a ferramenta de *Tiles* funciona, comparando suas características e eficiência de compressão com a principal ferramenta de paralelização do padrão H.264, os *Slices*. Os trabalhos [5] e [6] propõem um *framework* de paralelização focando no processo de estimação de movimento. O maior objetivo é dividir a estimação de movimento em várias regiões, permitindo que cada região seja processada paralelamente e melhorando a eficiência da compressão paralela para ambientes com vários núcleos. O trabalho [7] propõe um algoritmo de particionamento de *Tiles* baseado no conteúdo do vídeo, onde o principal objetivo é encontrar diferentes localizações para as bordas dos *Tiles* de modo a minimizar as perdas de qualidade inerentes a utilização de *Tiles*. Uma técnica de paralelização utilizando *Tiles* é proposta em [8] para determinar o número ideal de núcleos para comprimir um dado conjunto de dados, maximizando a utilização de cada núcleo e minimizando o número de núcleos ativos. A maioria dos trabalhos propõem soluções para lidar com ambos os problemas que surgem com a utilização de *Tiles*: (1) *speedup* limitado pelo particionamento uniforme e (2) perda na eficiência de compressão devido a quebras de referências ocasionadas pelos particionamentos dos quadros. *No entanto, nenhum destes trabalhos avalia o potencial de speedup que pode ser proporcionado usando diferentes particionamentos de Tiles não uniformes na compressão paralela do padrão HEVC.*

Contribuição deste trabalho: Neste trabalho, nós apresentamos uma avaliação de diferentes particionamentos de *Tiles* e como isso afeta a divisão da carga computacional entre os diferentes *Tiles* na compressão de vídeo paralela segundo o padrão HEVC. O trabalho demonstra que existe potencial de aumento de *speedup* através da utilização de *Tiles* não uniformes. Além disso, a eficiência de compressão resultante desses diferentes particionamentos de *Tiles* também é apresentada.

Organização: este trabalho está organizado da seguinte maneira. A Seção II descreve a metodologia de avaliação utilizada nas simulações. As Seções III e IV apresentam a análise e discussão sobre os resultados obtidos com as simulações. Finalmente, a Seção V conclui o trabalho.

II. METODOLOGIA DE AVALIAÇÃO

Para realizar a avaliação da compressão de vídeo paralela com o HEVC proposta neste trabalho, a seguinte configuração de testes foi definida. Todas as simulações consistiram em comprimir 100 quadros dos doze vídeos especificados na Tabela I, e todas as simulações foram realizadas de acordo com as Condições Comuns de Teste especificadas pela *Joint Collaborative Team on Video Coding (JCT-VC)* [9].

A métrica de eficiência de paralelização utilizada foi o *speedup*, que representa o ganho de velocidade em ambiente paralelo em relação a um ambiente sequencial, ou seja, quão mais rápida uma tarefa é executada com processamento paralelo quando comparada a mesma tarefa sendo executada

sequencialmente. O *speedup* varia de **um** ao **número de Tiles** utilizados. Um representa o *speedup* mínimo, o caso em que toda a carga computacional está concentrada em um único Tile, enquanto que o número de *Tiles* representa o *speedup* ótimo, o caso em que a carga computacional está dividida igualmente entre os *Tiles*. A equação do *speedup* é descrita pela equação (1) onde **n** representa o número de *Tiles* utilizados e T_{tile_i} representa o tempo de compressão do *i*-ésimo *Tile*.

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

Além disso, nós utilizamos o *Bjontegaard Delta Rate (BD-Rate)* [10] como métrica para medir a eficiência de compressão. Todas as simulações foram realizadas utilizando o *HEVC Test Model (HM)*, na versão 16.0. Rotinas foram inseridas no código do HM para extrair o tempo de compressão de cada CTU e *Tile*, e adicionalmente, outras rotinas foram inseridas para permitir que os *Tiles* fossem redimensionados a cada novo quadro, algo que não é possível na implementação padrão do HM. Para todas as simulações neste trabalho, foram utilizados quatro *Tiles* (particionamento 2x2).

A análise apresentada neste trabalho foca nos resultados de *speedup* gerados pela utilização de *Tiles* no processo de compressão paralela de vídeos digitais. Deste modo, três conjuntos de simulações foram definidos para extrair os dados necessários para efetuar a avaliação da compressão de vídeo paralela usando *Tiles*.

A. Compressão sem Tiles

O primeiro conjunto de simulações é realizado com a ferramenta de *Tiles* desativada, ou seja, cada quadro é processado pelo HEVC de maneira sequencial. Sendo assim, é possível estabelecer uma referência para a taxa de bits, qualidade do vídeo e o tempo necessário para realizar a compressão. Além disso, o tempo de compressão de cada CTU é extraído e utilizado como métrica para a carga computacional associada a cada CTU. Os resultados extraídos da compressão sequencial são utilizados como referência para comparação com a utilização da compressão paralela usando *Tiles*.

B. Compressão usando Tiles uniformes

Para o segundo conjunto de simulações o software HM é configurado para utilizar a ferramenta de *Tiles*, particionando o quadro em *Tiles* de tamanhos uniformes, ou seja, com o número de CTUs igualmente divididos entre os *Tiles*. O tempo de compressão de cada *Tile* é utilizado para calcular o *speedup* resultante de tal divisão em relação à compressão sequencial. Além disso, essas simulações também são usadas como referência, já que proporcionaram informações importantes sobre a perda de eficiência de compressão que é inerente à utilização dos *Tiles*.

C. Compressão usando Tiles não uniformes

O terceiro conjunto de simulações também utiliza a ferramenta de *Tiles*, no entanto, neste caso a opção de *Tiles* não uniformes é considerada. O objetivo deste conjunto de simulações é extrair o particionamento que resulte no *speedup* máximo para cada quadro em termos de *speedup*, tornando possível a avaliação do potencial de aumento de *speedup* que pode ser proporcionado pela utilização de particionamentos diferentes do uniforme. Esses particionamentos que geram o *speedup* máximo foram obtidos a partir dos tempos de

TABELA I. VÍDEOS TESTADOS

Vídeo	Resolução	Vídeo	Resolução
<i>BasketballDrive</i>	1920x1080	<i>BQMall</i>	1920x1080
<i>BQTerrace</i>	1920x1080	<i>Kimono1</i>	1920x1080
<i>FourPeople</i>	1280x720	<i>KristenAndSa</i>	1280x720
<i>Johnny</i>	1280x720	<i>SlideShow</i>	1280x720
<i>BasketballDrill</i>	832x480	<i>PartyScene</i>	832x480
<i>BQMall</i>	832x480	<i>RaceHorses</i>	832x480

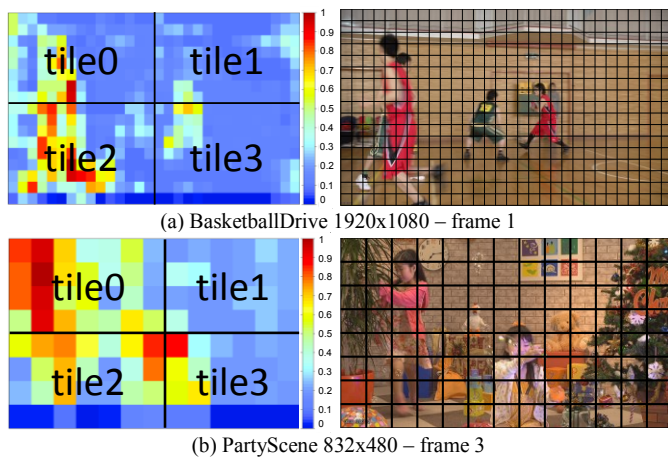


Fig. 1. Carga computacional por CTU.

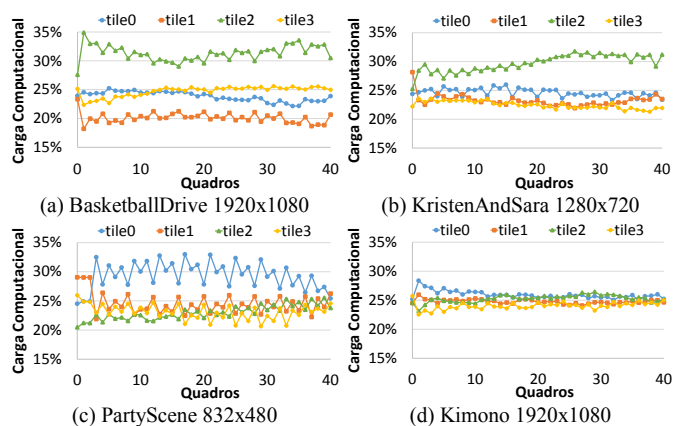
compressão por CTU extraídos do primeiro conjunto de simulações (compressão sem *tiles*). Usando tais tempos, é possível estimar o tempo de compressão para cada *Tile* de todos os possíveis particionamentos 2x2.

Por fim, todos os vídeos são comprimidos novamente com a configuração de *Tiles* não uniformes, onde as bordas dos *Tiles* são definidas de modo a aumentar o *speedup*. Esse tipo de compressão só é possível devido às rotinas que foram inseridas no software HM para que fosse possível redimensionar os *Tiles* a cada quadro. Sendo assim, é possível comparar a compressão usando *Tiles* que são definidos dinamicamente para cada quadro com uma compressão onde todos os quadros usam *Tiles* uniformes, tanto em relação ao *speedup* quanto à eficiência de compressão.

III. ANÁLISE DA COMPRESSÃO PARALELA USANDO *TILES* UNIFORMES

Considerando a metodologia proposta, um conjunto de resultados é extraído para avaliar a compressão de vídeo paralela utilizando *Tiles* segundo o padrão HEVC. Como mencionado anteriormente, uma divisão igualitária de informação no contexto da compressão de vídeo não é o suficiente para proporcionar máximo desempenho de *speedup* quando se considera processamento paralelo. Como o processo de compressão de vídeo explora redundâncias espaciais e temporais, áreas com alta textura e movimento aumentam o esforço computacional do processo de compressão daquela região específica, enquanto que áreas homogêneas e estáticas demandam menos esforço computacional. Sendo assim, para avaliar como a carga computacional está dividida entre os *Tiles* primeiro é necessário avaliar como a carga computacional está distribuída pela imagem.

Para ilustrar essa avaliação, vídeos foram comprimidos considerando a metodologia de compressão sem *Tiles*. Os mapas de calor da Fig. 1 mostram a carga computacional associada a cada CTU em um quadro, sendo que cada quadrado representa o tempo de compressão de cada CTU e as cores estão normalizadas de modo que o quadrado com o vermelho mais escuro representa a CTU mais custosa em termos de processamento. Tanto a Fig. 1(a) quanto a Fig. 1(b) revelam que essas CTUs mais custosas estão agrupadas, no entanto, esses agrupamentos não se dividem igualmente pela imagem. Quando é feito um particionamento uniforme 2x2 na Fig. 1(a), é possível notar que o *tile2* possui um grande conjunto das CTUs com maior carga computacional, enquanto que o *tile1*, por exemplo, possui CTUs menos custosas. Isso

Fig. 2. Divisão da carga computacional entre os *Tiles*.

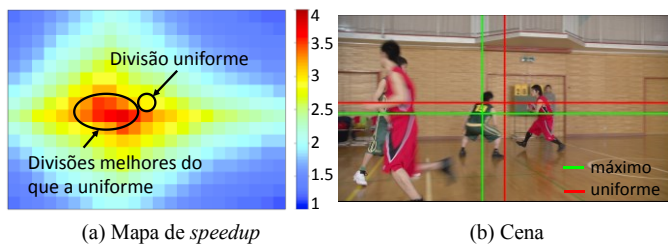
acontece, pois no *tile2* existe uma região com maior presença de detalhes (alta textura) e também maior movimentação (o jogador está se movendo na cena). Similarmente, em (b) o *tile0* é o que requer o maior esforço computacional. Sendo assim, é possível perceber que a utilização do particionamento de *Tiles* uniformes 2x2 nem sempre vai explorar ao máximo o paralelismo, já que mesmo que as CTUs estejam divididas igualmente entre os *Tiles*, o tempo que cada *Tile* leva para ser comprimido varia com as características do vídeo.

Considerando as informações extraídas dos três conjuntos de simulações definidos na Seção II, é possível estabelecer uma ligação entre o particionamento dos *Tiles* e como isso afeta a eficiência do processamento paralelo. A primeira análise comparou a divisão da carga computacional entre os *Tiles* utilizando o particionamento uniforme 2x2. A fim de fazer tal análise, o tempo gasto na compressão de cada *Tile* é comparado ao tempo total para a compressão do quadro, assim é possível extrair a carga computacional relativa de cada *Tile*, como é mostrado na Fig. 2.

Observando os gráficos na Fig. 2 é possível concluir que o particionamento uniforme nem sempre dividirá o quadro da maneira mais igualitária em termos de carga computacional. Esse particionamento pode acabar gerando um desbalanceamento de carga computacional entre os *Tiles* devido às diferentes propriedades presentes em cada região do vídeo, tais como movimento e textura, que influenciam diretamente no esforço computacional associado à compressão de cada uma dessas regiões. Ainda baseado na Fig. 2, é possível perceber que alguns *Tiles* levam um tempo consideravelmente maior do que outros para serem processados. Considerando o vídeo *BasketballDrive*, por exemplo, o *tile2* leva 1,92 vezes mais tempo do que o *tile1* para alguns quadros, e além disso, as linhas se cruzando mostram que os *Tiles* que levam mais tempo para serem processados podem não ser os mesmos durante a compressão de todo o vídeo. Isso acontece devido à frequente movimentação dos objetos pela cena e pode acontecer de maneira ainda mais brusca quando vídeos que possuem frequente troca de cena são considerados. Esse desbalanceamento da carga computacional entre os *Tiles* pode acabar limitando o *speedup* de vídeos comprimidos segundo o padrão HEVC.

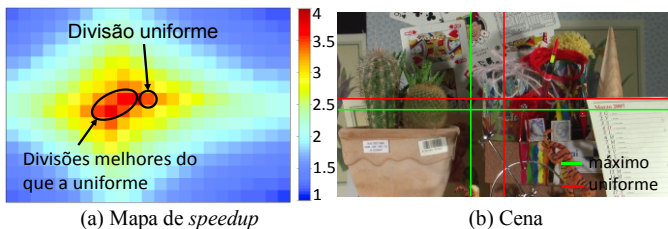
IV. ANÁLISE DO POTENCIAL DE *SPEEDUP* USANDO *TILES* NÃO UNIFORMES

O principal objetivo deste trabalho é avaliar o potencial de *speedup* na compressão de vídeo paralela do HEVC utilizando *Tiles*. A principal pergunta que a avaliação proposta neste

(a) Mapa de *speedup*

(b) Cena

Fig. 3. BasketballDrive 1920x1080 – frame 1.

(a) Mapa de *speedup*

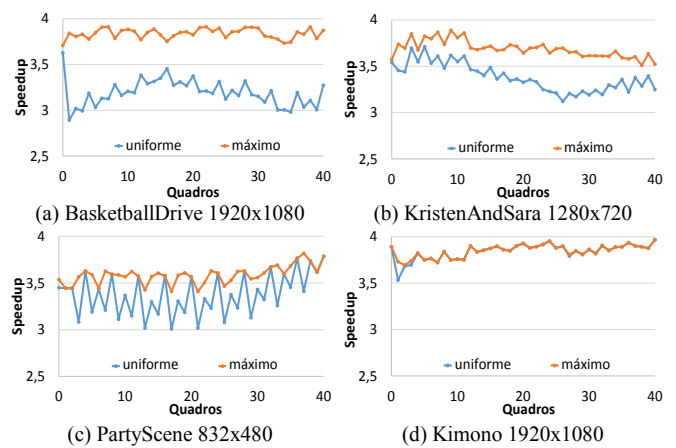
(b) Cena

Fig. 4. Cactus 1920x1080 – frame 5.

trabalho busca responder é se existem outras maneiras de configurar os *Tiles* no HEVC de modo a aumentar o desempenho da compressão de vídeo paralela. Para responder a tal pergunta, o *speedup* de todos os possíveis particionamentos de *Tiles* 2x2 foi extraído. O tempo gasto em cada CTU, extraído no primeiro conjunto de simulações (compressão sem *Tiles*), é utilizada novamente já que esses dados possibilitaram o cálculo da distribuição da carga computacional ao longo do quadro, e isso possibilita estimar qual é o *speedup* associado a cada um dos possíveis particionamentos 2x2. A Fig. 3 e a Fig. 4 mostram o resultado dessa avaliação, onde cada quadrado representa o *speedup* calculado referente a cada intersecção de um possível particionamento 2x2, ou seja, a intersecção entre uma borda vertical e uma borda horizontal. A cor de cada quadrado está normalizada de modo que o vermelho mais escuro representa o *speedup* ótimo na compressão de um quadro, enquanto que o azul mais escuro representa o *speedup* mínimo. Como foram utilizados quatro *Tiles*, o *speedup* ótimo também é quatro. A Fig. 3(b) e a Fig. 4(b) representam a cena real do vídeo, sendo que as bordas vermelhas representam o particionamento uniforme enquanto que as bordas verdes representam o particionamento que resulta no *speedup* máximo.

Analisando a Fig. 3 e a Fig. 4 é possível perceber que o particionamento uniforme não resulta no melhor *speedup*. Além disso, é possível notar que existe um bom espaço de soluções onde os particionamentos não uniformes resultam em um *speedup* melhor do que o uniforme. Esse espaço de soluções tem potencial de ser explorado para melhorar a eficiência da compressão de vídeo paralela do HEVC.

Uma outra avaliação foi realizada comparando os resultados de *speedup* entre o particionamento uniforme e o particionamento não uniforme que resulta no *speedup* máximo. A Fig. 5 mostra essa avaliação onde a variação de *speedup* para quatro vídeos é apresentada. A linha laranja representa o *speedup* máximo para cada quadro, gerado a partir do melhor particionamento não uniforme, enquanto que a linha azul representa o *speedup* do particionamento uniforme. É visível que o particionamento uniforme raramente é o melhor particionamento possível, visto que as linhas não se tocam com muita frequência para a maioria dos vídeos. Inclusive, o melhor particionamento é muito melhor do que o particionamento uniforme para boa parte do vídeo, podendo aumentar o *speedup* em até 27%, como ocorre no frame 50 do vídeo *BasketballDrive*. Por outro lado, para vídeos como

Fig. 5. *Speedup* máximo versus uniforme.

Kimono, onde a distribuição da carga computacional é naturalmente bem distribuída, o particionamento uniforme já é bom o suficiente para gerar resultados de *speedup* próximos do máximo, coincidindo com o melhor particionamento por boa parte do vídeo. Outro aspecto a ser observado é que tanto o particionamento uniforme quanto o não uniforme apresentam uma alta variação temporal entre os quadros. Isso ocorre devido à alta variação do conteúdo entre os quadros (textura e movimento), já que existe movimento constante nas cenas.

Por fim, um sumário com todos os resultados em termos de *speedup* médio é mostrado na Tabela II. Os resultados são apresentados tanto para o particionamento uniforme quanto para o particionamento não uniforme que proporciona o máximo *speedup*, e também a variação de *speedup* entre esses dois particionamentos.

A partir da Tabela II é possível notar que a maior variação de *speedup* médio alcançou 18,9% para o vídeo *BasketballDrive*. Isso ocorre porque *BasketballDrive* é um vídeo onde as regiões de alta textura e alta movimentação não estão divididas de maneira uniforme ao longo do quadro. Além disso, é possível observar que o desvio padrão do *speedup* médio dos vídeos diminuiu de 0,19x para 0,12x, mostrando que os resultados podem ser mais consistentes quando se utiliza o particionamento que visa o melhor *speedup*. Por outro lado, existem casos (*Kimono* e *Johnny*) em que o *speedup* máximo é apenas 0,26% maior do que o *speedup* atingido pelo particionamento uniforme. Nestes casos, o particionamento uniforme já é bom o suficiente em

TABELA II. RESULTADOS DE *SPEEDUP*.

Vídeo	Uniforme [x]	<i>Speedup</i> máximo [x]	Variação
1920x1080			
<i>BasketballDrive</i>	3,22	3,83	18,90%
<i>Cactus</i>	3,39	3,73	10,03%
<i>BQTerrace</i>	3,53	3,77	6,80%
<i>Kimono</i>	3,83	3,84	0,26%
1280x720			
<i>FourPeople</i>	3,40	3,59	5,59%
<i>Johnny</i>	3,78	3,79	0,26%
<i>SlideShow</i>	3,73	3,75	0,54%
<i>KristenAndSara</i>	3,65	3,75	2,74%
832x480			
<i>RaceHorses</i>	3,38	3,44	1,78%
<i>PartyScene</i>	3,34	3,60	7,78%
<i>BasketballDrill</i>	3,45	3,60	4,35%
<i>BQMall</i>	3,50	3,55	1,43%

TABELA III. AUMENTO DE BD-RATE.

Vídeo	Uniforme	Speedup máximo	Varição
1920x1080			
<i>BasketballDrive</i>	0,85%	0,81%	-0,04
<i>Cactus</i>	0,34%	0,32%	-0,02
<i>BQTerrace</i>	0,63%	0,61%	-0,02
<i>Kimono</i>	0,89%	0,85%	-0,04
1280x720			
<i>FourPeople</i>	2,01%	2,12%	+0,11
<i>Johnny</i>	3,05%	3,05%	0,00
<i>SlideShow</i>	3,70%	4,01%	+0,31
<i>KristenAndSara</i>	2,39%	2,36%	-0,03
832x480			
<i>RaceHorses</i>	0,81%	0,71%	-0,10
<i>PartyScene</i>	0,44%	0,38%	-0,06
<i>BasketballDrill</i>	0,98%	1,10%	+0,12
<i>BQMall</i>	1,24%	1,29%	+0,05
Média	1,44%	1,47%	+0,03

termos de *speedup* já que os vídeos já possuem uma distribuição de conteúdo uniforme. Esses resultados discrepantes advêm da natureza distinta dos vídeos digitais, com objetos se movendo pela cena, com a câmera se movimentando e também com áreas altamente detalhadas. Outra observação interessante a respeito da Tabela II é que em nenhum vídeo, o particionamento uniforme coincide com o que gera o *speedup* máximo em todos os quadros. Em média, o particionamento não uniforme que proporciona *speedup* máximo é 5,04% melhor do que o particionamento uniforme. Isso pode ser um indicio de que na maioria dos vídeos existe espaço para encontrar soluções que resultem em maior *speedup* do que simplesmente usar o particionamento uniforme, e consequentemente melhorar o desempenho da compressão de vídeo paralela.

O uso de *Tiles*, entretanto, afeta negativamente a eficiência de compressão. Isso ocorre devido às quebras de contexto entre regiões vizinhas separadas pelas bordas verticais e horizontais [7]. Considerando isso, uma avaliação foi realizada para medir a eficiência de compressão (em termos de BD-Rate [10]) resultante do particionamento uniforme e do particionamento que proporciona *speedup* máximo. O objetivo desta análise é saber se alterar o particionamento uniforme para um particionamento que proporcione melhores resultados de *speedup* gera impacto significativo na eficiência de compressão. Os resultados são apresentados na Tabela III onde os valores em percentual representam o aumento de BD-Rate (valores maiores são piores) da compressão paralela usando *Tiles* (particionamento uniforme e particionamento com *speedup* máximo) em comparação com a compressão sequencial. A coluna de variação na Tabela III mostra a diferença de BD-Rate entre o particionamento uniforme e o particionamento não uniforme com *speedup* máximo.

A partir da Tabela III pode ser observado que existe um aumento médio de BD-Rate de 1,44% quando o particionamento uniforme é utilizado, enquanto que o particionamento com *speedup* máximo leva a um aumento médio de BD-Rate de 1,47%. Comparando esses resultados, é possível observar que a diferença de BD-Rate é mínima (0,03% em média) entre os dois particionamentos. Além disso,

para alguns vídeos os resultados de BD-Rate para o particionamento com *speedup* máximo são melhores do que os resultados para o particionamento uniforme. Isso significa que não existe variação significativa de BD-Rate relacionada ao uso de *Tiles* não uniformes em comparação com *Tiles* uniformes. Os resultados extraídos da Tabela II junto com os resultados da Tabela III mostram informações importantes relacionadas a como diferentes particionamentos não uniformes tem grande potencial de melhorar o *speedup* da compressão de vídeo paralela usando *Tiles*, mantendo a eficiência de compressão.

V. CONCLUSÕES

Este trabalho apresentou uma avaliação da compressão de vídeo paralela utilizando a ferramenta de *Tiles* no HEVC. Foram avaliados *speedup* e eficiência de compressão para mostrar o potencial de melhorias quando os *Tiles* são utilizados para a compressão de vídeo paralela. Com os resultados apresentados nós mostramos que a compressão de vídeo paralela utilizando *Tiles* tem um comportamento desbalanceado em termos de divisão da carga computacional entre os *Tiles*. Além disso, existe espaço para melhorar a eficiência de compressão paralela com impacto insignificante na eficiência final da compressão. Com as condições de teste e vídeos avaliados, nós mostramos que existe espaço para melhorar em média 5,04% o *speedup*, sendo que esta melhoria pode atingir até 18,9% em alguns casos. Em termos de BD-Rate, houve um aumento médio de 0,03% em comparação ao particionamento uniforme, sendo que o aumento não ultrapassou 0,31% em nenhum caso. De modo geral, os resultados mostram que existem desafios para realizar compressão de vídeo paralela utilizando *Tiles* eficientemente, já que a divisão uniforme padrão não considera a natural variabilidade de conteúdo dos vídeos.

VI. REFERÊNCIAS

- [1] H. Kalva, "The H.264 Video Coding Standard", IEEE MultiMedia, v. 13, n. 4, pp. 86-90, 2006.
- [2] J. Ohm, G. Sullivan, H. Schwarz, T. Tan, T. Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards – Including High Efficiency Video Coding (HEVC)", IEEE Transactions on Circuits and Systems for Video Technology, v. 22, n. 12, pp. 1669-1684, 2012.
- [3] C. Chi, M. Alvarez-Mesa, B. Juurlink, G. Clare, F. Henry, S. Pateux, T. Schierl, "Parallel Scalability and Efficiency of HEVC Parallelization Approaches", IEEE Transactions on Circuits and Systems for Video Technology, v. 22, n. 12, pp. 1827-1838, 2012.
- [4] K. Misra, A. Segall, M. Horowitz, S. Xu, A. Fuldseth, M. Zhou, "An Overview of Tiles in HEVC", in Selected Topics in IEEE Journal of Signal Processing, v. 7, n.6, pp. 969-977, 2013.
- [5] C. Yan, Y. Zhang, F. Dai, L. Li, "Highly Parallel Framework for HEVC Motion Estimation on Many-Core platform", in Data Compression Conference, pp. 63-72, 2013.
- [6] C. Yan, Y. Zhang, J. Xu, F. Dai, J. Zhang, Q. Dai, F. Wu, "Highly Parallel Framework for HEVC Motion Estimation on Many-Core Processors", in IEEE Transactions on Circuits and Systems for Video Technology, v. 24, n. 12, pp. 2077-2089, 2014.
- [7] C. Blumenberg, D. Palomino, S. Bampi, B. Zatt, "Adaptive Content-Based Tile Partitioning Algorithm for the HEVC Standard," in Picture Coding Symposium, pp.185-188, 2013.
- [8] M. Shafique, M.U.K. Khan, J. Henkel, "Power Efficient and Workload Balanced Tiling for Parallelized High Efficiency Video Coding", in International Conference on Image Processing, 2014.
- [9] F. Bossen, Common Test Conditions and Software Reference Configurations, document JCTVC-H1100, JCT-VC, Feb. 2012.
- [10] G. Bjøntegaard, "Calculation of average PSNR differences between RDcurves", Technical Report VCEG-M33, ITU-T SG16/Q6, Austin.