

MSBSD: um Algoritmo para Decisão Rápida na Inter-Predição em Codificadores HEVC

Mateus Grellert¹, Felipe Sampaio², Luciano Agostini¹, Júlio C. B. Mattos¹

¹Grupo de Arquiteturas e Circuitos Integrados (GACI)
Universidade Federal de Pelotas (UFPEL)
Caixa Postal 354 – 96.001--970 – Pelotas – RS – Brasil

²Programa de Pós-Graduação em Computação
Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

{mgdsilva, agostini, julius}@inf.ufpel.edu.br, fmsampaio@inf.ufrgs.br

Abstract. *This paper presents the Motion Similarity Border Strength Decision (MSBSD) algorithm, which is a method to reduce the HEVC inter-prediction complexity through PU partition decision. The solution proposed in this work consists in avoiding several Motion Estimation searches during the inter-prediction in order to reduce execution time in the overall encoding process. The proposed method was implemented in the HM test model 5.1 and provides an execution time reduction of 37.11% on average with insignificant rate-distortion losses (0.006 dB quality drop and 0.948% bitrate increase in the average case). When compared other fast decision methods, MSBSD shows itself competitive, achieving results as good as those of related work.*

Resumo. *Este artigo apresenta o algoritmo Motion Similarity Border Strength Decision, um método para reduzir a complexidade da inter-predição do padrão emergente HEVC através de uma decisão heurística para as PUs. O MSBSD consiste em evitar diversas chamadas à Estimação de Movimento durante o processamento das PUs, reduzindo o tempo de execução da compressão. O método proposto foi implementado no HM 5.1 e atingiu uma redução de 37,11% em média no tempo de processamento, com perdas insignificantes em qualidade e compressão (perda de 0,006 dB na qualidade e aumento de bitrate de 0,948% no caso médio). Quando comparado com outros trabalhos, o MSBSD atingiu resultados tão bons quanto os demais.*

1. Introdução

Os avanços recentes em tecnologias de semicondutores permitiram uma série de melhorias nos sistemas digitais, expandindo de forma significativa os recursos computacionais presentes em um único chip. Aliado a isso, as redes de comunicação também avançaram a ponto de ser possível transferir um grande volume de dados em poucos segundos. Como consequência direta destas inovações, aplicações multimídia são atualmente suportadas por uma grande gama de dispositivos eletrônicos: de celulares e tablets a computadores pessoais. Quando foi lançado o padrão H.264/AVC [Wiegand et al, 2003], estado-da-arte em compressão de vídeo, apenas alguns dispositivos específicos suportavam vídeos de alta resolução. Todavia, o cenário atual é diferente, já que vídeos Full HD (1920x1080 pixels) são suportados por grande parte

dos dispositivos eletrônicos e resoluções ainda maiores começam a atrair a atenção do mercado, como a Ultra HD (7680x4320 pixels). Além disso, uma forte tendência em dispositivos móveis é prover mecanismos de codificação e decodificação para vídeos de alta resolução.

Para explorar melhor as características das novas mídias, um grupo composto por profissionais em codificação de vídeo da ITU-T e da ISO/IEC foi formado com o nome *Joint Collaborative Team on Video Coding* (JCT-VC) [JCT-VC, 2012]. O propósito do JCT-VC é desenvolver um novo padrão de codificação de vídeo com ferramentas aprimoradas para vídeos de alta resolução. Do trabalho colaborativo do JCT-VC está emergindo o padrão *High Efficiency Video Coding* (HEVC). O principal objetivo estabelecido para esse projeto de padronização é aumentar as taxas de compressão em 50% quando comparadas às do H.264/AVC, mantendo a mesma qualidade subjetiva da imagem [Bross et al, 2011]. Contudo, também foi estabelecido que a complexidade associada a esse padrão emergente não deve exceder a do H.264/AVC em mais do que um fator de três. A norma final para esse padrão está prevista para ser lançada em janeiro de 2013.

Apesar de o padrão HEVC ainda estar em desenvolvimento, uma lista extensa de documentos é lançada após cada reunião do JCT-VC. Essa lista inclui as atualizações na norma em construção, descrições detalhadas e análises de novas ferramentas, entre outros. Além disso, o software de referência (conhecido como HEVC Test Model – HM) também é publicamente disponibilizado para download, então é possível realizar análises de compressão, qualidade e complexidade à medida que o HEVC é desenvolvido. Em termos de compressão, os resultados atuais são promissores, indicando que a taxa de 50% na compressão deve ser atingida. Por outro lado, a complexidade continua sendo um problema, já que a última versão do HM (5.1) [McCann et al, 2011] já inclui técnicas de compressão que são muito mais complexas que as do H.264/AVC. Isso gera forte preocupação quando aplicações de tempo real, como transmissões ao vivo, são consideradas. A alta complexidade é também um fator crítico para dispositivos móveis baseados em bateria, pois nesse caso uma grande quantidade de energia é necessária para realizar tarefas de codificação e decodificação. Mais que isso, a complexidade do H.264/AVC ainda é um problema nas tecnologias atuais, e vários projetos de pesquisa buscam soluções para esse problema, portanto, fica ainda mais claro que a complexidade do HEVC é um problema real que deve ser tratado de forma cuidadosa pelo JCT-VC e por outros trabalhos paralelos desenvolvidos na academia.

As estruturas de dados hierárquicas definidas no padrão HEVC podem ser destacadas como uma das principais causas da sua elevada complexidade. Quadros são agora particionados em diversas *Treeblocks*, que são estruturas fixas bastante análogas aos *Macroblocos* dos padrões anteriores. No H.264/AVC, um *Macrobloco* tem tamanho fixo de 16x16 amostras, podendo ser particionado em blocos de tamanho 16x8, 8x16, 8x4, 4x8 e 4x4 durante a predição inter-quadros [Wiegand et al, 2003], mas nenhuma estrutura de dados adicional foi definida nesse padrão. Em contraste ao H.264/AVC, *Treeblocks* têm tamanhos maiores (até 64x64 amostras) e podem ser recursivamente divididas em Unidades de Codificação (*Coding Units* - CUs), formando uma estrutura de árvore quaternária, o que garante uma estrutura de codificação muito mais flexível. A raiz da árvore quaternária corresponde à maior CU, que é particionada recursivamente até chegar às folhas, as quais representam as menores CUs possíveis. Um exemplo

dessa árvore é ilustrado na Figura 1, explicitando a profundidade na árvore para cada tamanho de CU.

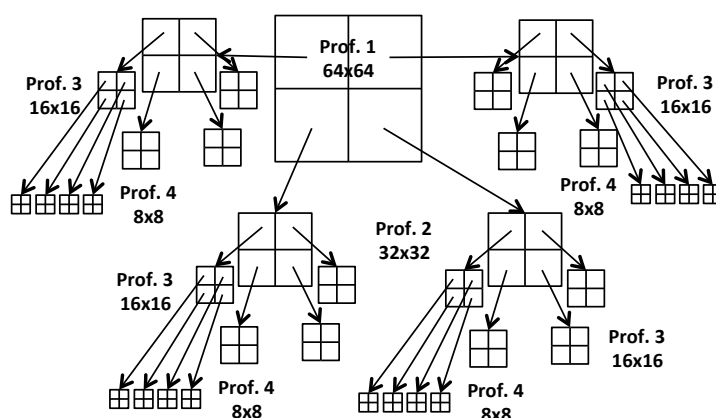


Figura 1. Exemplo de uma Treeblock dividida em uma árvore quaternária

A decisão em nível de CU envolve descobrir qual nodo da árvore quaternária é o mais adequado para codificar as amostras da Treeblock que está sendo processada. Na versão atual do HM (5.1) essa decisão é realizada através da técnica *Rate-Distortion Optimization* (RDO), a qual avalia o bitrate (medida de taxa de compressão) e o PSNR (medida de qualidade objetiva) produzidos por todas as configurações necessárias. Em outras palavras, a predição, a codificação residual e a codificação de entropia são realizadas em cada nodo da árvore quaternária. Durante a etapa de predição, cada CU é novamente dividida em Unidades de Predição (*Prediction Units* - PU), e o custo RD (custo que resume PSNR e bitrate a um único valor) deve ser avaliado para os três modos de predição disponíveis: skip, intra-quadro e inter-quadros.

A maior parte dos ganhos de compressão vem da predição inter-quadros, já que esse processo explora redundâncias temporais, e tais redundâncias são muito frequentes em altas taxas de quadros (normalmente são necessários 30 quadros por segundo para transmitir a ilusão de movimento). No entanto, a inter-predição também introduz a maior parte da complexidade computacional (entre 70% e 96%) associada ao processo de compressão. Tal complexidade deriva da Estimação de Movimento (EM), que é a principal ferramenta deste processo. O HEVC define que a inter-predição deve ser realizada para cada partição de PU quando quadros do tipo preditivo (P) ou bi-preditivo (B) são codificados. Quadros P e B ocorrem com muito mais frequência que os do tipo intra (I), portanto, soluções que reduzem o número de chamadas à EM causam uma grande redução na complexidade total da codificação.

Esse artigo apresenta o algoritmo *Motion Similarity Border Strength Decision* (MSBSD), um método que reduz o tempo de codificação através do uso de heurísticas que reduzem o número de partições de PU que são aplicadas à predição inter-quadros, reduzindo, portanto, o número de chamadas à EM necessárias dentro de cada nodo da árvore quaternária. Para decidir quais partições de PU serão efetivamente avaliadas, duas heurísticas de decisão foram utilizadas: *Motion Vector Merge* (MVM), para as partições simétricas, e *Border Strength* (BS), para as assimétricas. Uma partição de cada tipo é selecionada por essas decisões, então a EM é chamada para ambas, e a que atingir o menor custo RD é escolhida.

O MSBSD foi implementado no HM 5.1, e os resultados de custo RD, bem como resultados de bitrate e tempo de processamento, foram obtidos. De acordo com os resultados, o MSBSD é capaz de atingir uma redução no tempo de codificação de 37,11% com perda insignificante de qualidade (0,006 dB) e pequeno aumento no bitrate (0,948% no caso médio). Além disso, o algoritmo MSBSD pode ser facilmente agregado a outras decisões rápidas que trabalham com as outras estruturas de codificação do HEVC, como as CUs.

Todas as etapas deste projeto foram desenvolvidas pelo primeiro autor deste artigo, aluno de graduação da Universidade Federal de Pelotas e bolsista de iniciação científica junto ao Grupo de Arquiteturas e Circuitos Integrados. O trabalho foi orientado pelos demais autores. As etapas envolvendo elaboração de heurísticas, implementações no software de referência HM, as diversas análises elaboradas e, por fim, a extração dos resultados foram realizadas pelo primeiro autor. É importante salientar que a solução desenvolvida e apresentada neste artigo está estreitamente conectada a um dos cinco Grandes Desafios da Pesquisa em Computação no Brasil, definidos pela SBC no ano de 2006 [SBC, 2012]. O foco deste trabalho é o desenvolvimento de soluções de complexidade para a compressão no padrão emergente HEVC, contribuindo diretamente para o desafio “Gestão da Informação em grandes volumes de dados multimídia distribuídos”, uma vez que uma das questões essenciais em sistemas multimídia é justamente comprimir a informação de vídeo com taxas elevadas, com perdas mínimas de qualidade e processamento em tempo real.

O restante do artigo está organizado como segue: a seção 2 discute o processo de predição inter-quadros, incluindo análises de complexidade, uso de partições de PUs e algoritmos de busca; a seção 3 explica detalhadamente o algoritmo MSBSD, que é a principal contribuição deste trabalho; na seção 4, resultados de PSNR, bitrate e complexidade são expostos e discutidos; e a seção 5 finalmente apresenta as conclusões e trabalhos futuros.

2. Predição Inter-Quadros no HEVC

Como mencionado anteriormente, Treeblocks são áreas de até 64x64 amostras do quadro sendo codificado que podem ser recursivamente divididas em CUs menores, com tamanho mínimo de 8x8 amostras cada, formando uma estrutura de árvore quaternária, e cada nodo desta árvore representa uma possível CU. Quando o quadro sendo codificado é do tipo P ou B, ocorre a predição inter-quadro.

No padrão HEVC, uma CU pode ser ainda mais particionada em PUs durante a etapa de predição. Inicialmente, quatro tipos de partições de PU foram definidas. Esse conjunto é composto por partições de movimento simétricas (PMS) quadradas e retangulares: $2N \times 2N$, $2N \times N$, $N \times 2N$ e $N \times N$ [Bross et al, 2011], onde $2N$ é equivalente à dimensão da CU.

Desde a primeira reunião do JCT-VC, quatro partições de movimento assimétricas (PMA) adicionais foram propostas [McCann et al, 2010], mas elas só foram incluídas de fato na versão 3.4 do HM. De acordo com [McCann et al, 2010], o uso de PMA pode aumentar a eficiência da codificação, pois padrões irregulares de imagem podem ocorrer e as PMS não representariam uma solução adequada. A Figura 2 ilustra as partições de movimento simétricas (a) e assimétricas (b) possíveis para uma CU de tamanho 64x64.

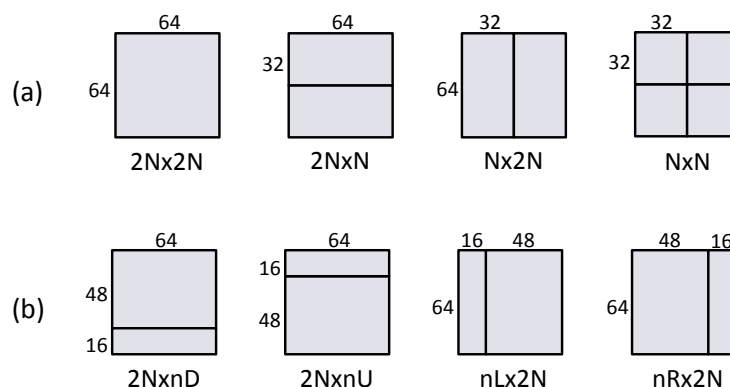


Figura 2. Partições de movimento (a) simétricas e (b) assimétricas

Quando a predição inter-quadros é processada, a EM é chamada para cada partição de PU possível. Essa ferramenta consiste em tentar descobrir para onde um objeto se deslocou entre um quadro e outro do vídeo. De forma geral, isso se resume em dividir o quadro sendo processado em blocos e, para cada bloco, encontrar a região mais semelhante a ele dentro de um quadro previamente codificado (chamado quadro de referência). Quando o melhor casamento é encontrado, ao invés de reproduzir todas as amostras do bloco, um vetor de movimento que aponta para o melhor casamento no quadro de referência é gerado. No contexto do HEVC, esses blocos são representados pelas PUs, portanto esse processo é aplicado para todas as partições de PU possíveis dentro de cada CU da árvore quaternária.

A busca pelo melhor casamento na EM é realizada de acordo com um algoritmo de busca, e o HM implementa dois deles: (1) *Full Search*, que realiza uma pesquisa exaustiva dentro de uma área do quadro de referência, comparando todos os candidatos possíveis; e (2) *TZ Search* (TZS), um algoritmo de busca que é 23 vezes mais rápido que o FS e um pouco menos eficiente em termos de qualidade [Wiegand et al, 2003]. Devido à caracterização força bruta do FS, ele sempre produz os melhores vetores de movimento possíveis, mas sua complexidade computacional também é muito elevada.

2.1. Análise de Complexidade do HM

A Figura 3 apresenta gráficos representando o tempo de execução gasto na Estimação de Movimento, comparado ao tempo das demais ferramentas de codificação do HEVC no HM 5.1. Dois vídeos comumente utilizados pela comunidade que trabalha com o HEVC foram utilizados: BQTerrace e Vidyo.

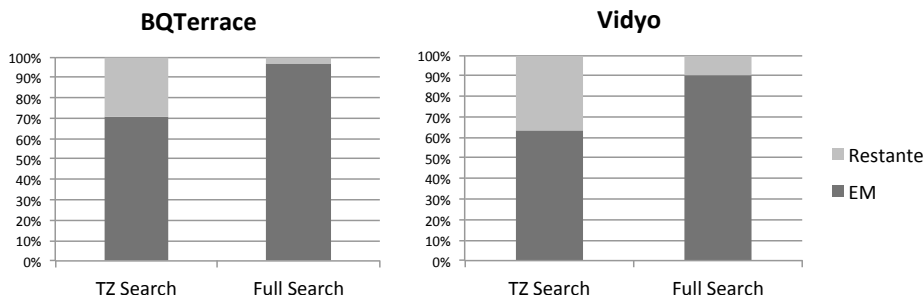


Fig. 3. Tempo de execução no HM usando os algoritmos de busca FS e TZS

Os dados do gráfico mostram que a Estimação de Movimento, a tarefa principal da inter-predição, ocupa mais de 96% do tempo de compressão quando o algoritmo *Full Search* é utilizado. Além disso, mesmo quando o *TZ Search* realiza a busca, a EM ainda ocupa mais de 70% do tempo total de processamento.

O HM implementa uma otimização que evita a inter-predição para partições NxN nas CUs que não estão na profundidade máxima da árvore, i.e, nas CUs de menor tamanho possível, as quais serão chamadas de CUs folha neste trabalho. Isso evita cálculos redundantes, pois uma partição NxN em uma CU de profundidade p é equivalente a uma partição 2Nx2N em uma CU na profundidade $p+1$. Além disso, as PMA não são avaliadas nas CUs folha. Portanto, a EM é chamada quatro vezes em CUs folha e sete vezes nas demais.

2.2. Análise da Ocorrência de Partições de PU na Inter-Predição

As partições escolhidas pela técnica RDO considerando cada tamanho de UC foram obtidas para a análise da ocorrência de partições de PU na inter-predição. Considerando todas as dimensões possíveis (64, 32, 16 e 8), toda partição escolhida foi contabilizada. É importante ter conhecimento da partição mais utilizada com a técnica RDO, pois ela pode ter um peso maior no desenvolvimento de heurísticas. A Tabela 1 mostra as porcentagens médias das partições de PU escolhidas para três sequências Full HD: BasketballDrive, BQTerrace e Cactus.

Tabela 1. Porcentagens médias do uso de partições de PU

Tamanho CU	2Nx2N	2NxN	Nx2N	NxN	2NxN _U	2NxN _D	nLx2N	nRx2N
64	73,02	5,76	6,96	0,00	3,71	2,93	4,17	3,45
32	82,44	2,19	2,33	0,00	1,84	1,39	1,88	1,43
16	89,48	0,85	0,88	0,00	0,59	0,50	0,64	0,46
8	92,52	0,40	0,45	0,00	-	-	-	-

Analisando a tabela, percebe-se que as partições 2Nx2N são majoritariamente escolhidas, especialmente em CUs menores. Isso significa que essas partições devem ser priorizadas em heurísticas de decisão para aumentar a taxa de acertos. As partições NxN sequer foram escolhidas para esses vídeos, o que mostra que esse tipo de partição deve ter o menor peso em decisões heurísticas. Ademais, as PMS retangulares e as PMA tiveram uma ocorrência bastante baixa.

3. O Algoritmo MSBSD

Como explicado anteriormente, a EM é chamada de três a sete vezes em cada nodo da árvore quaternária de acordo com a técnica RDO. Essa característica é extremamente indesejada em função da elevada complexidade computacional que isso acarreta. Portanto, soluções que objetivam reduzir o número de chamadas à EM são muito importantes para tornar factíveis as tarefas de codificação e de decodificação HEVC em tempo real. O algoritmo *Motion Similarity Border Strength Decision* (MSBSD) descrito neste trabalho foi desenvolvido com tal objetivo em mente. Para atingir esse objetivo, uma heurística é aplicada para decidir as partições de UP superiores à NxN que serão processadas pela EM. Duas decisões são consideradas: (1) *Motion Vector Merge* (MVM), para as PMS; e (2) *Border Strength* (BS), para as PMA. Uma visão geral do algoritmo MSBSD é ilustrada na Figura 4, e cada uma das decisões empregadas será explicada nas próximas subseções.

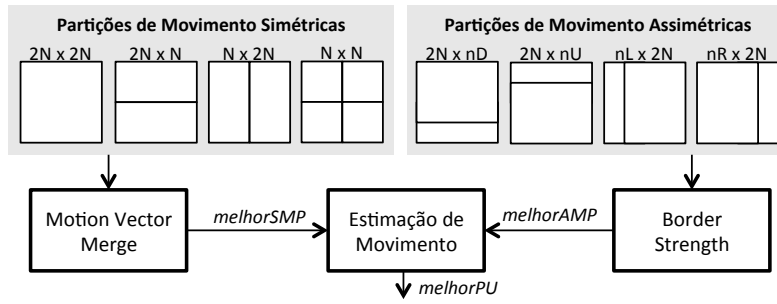


Fig. 4. Visão geral do algoritmo MSBSD

3.1. Motion Vector Merge

Esse algoritmo é usado para decidir de forma antecipada qual PMS deve atingir o menor custo RD. Ele foi inicialmente proposto pelos autores deste artigo em [Sampaio et al, 2012], quando somente as PMS eram consideradas na inter-predição. Quando certas condições são satisfeitas, partições $N \times N$ são combinadas em partições $2N \times N$, $N \times 2N$ ou $2N \times 2N$, sem a necessidade de realizar a EM para cada uma das partições maiores. Se as condições não são satisfeitas, então $N \times N$ é selecionada como a melhor PMS e nenhuma chamada adicional à EM é necessária para esse tipo de partição. É importante salientar que as partições $2N \times 2N$ devem ter prioridade caso haja empate, pois essas partições, como mostrado na análise da seção anterior, são as mais escolhidas pela técnica de RDO. Além disso, partições maiores também aumentam os ganhos de compressão. Isso é explicado pelo fato de as PUs maiores necessitarem de menos informação de controle por pixel, como vetores de movimento e índices de quadros de referência.

A Figura 5 mostra o algoritmo MVM e a heurística aplicada na sua decisão. Para cada CU, a EM é inicialmente chamada para as partições $N \times N$. Com os vetores produzidos pela EM, a heurística MVM é aplicada para decidir quais partições serão agrupadas. As condições definidas pela heurística são baseadas na similaridade dos vetores de movimento entre PUs vizinhas, isto é, se todas as quatro partições $N \times N$ produziram o mesmo vetor, então $2N \times 2N$ é considerada a melhor partição e a EM é chamada para essa partição somente. Se restrições para $2N \times 2N$ não são satisfeitas, as PMS retangulares são consideradas da mesma forma, ou seja, se duas PUs vizinhas produziram o mesmo vetor de movimento, então a partição correspondente é selecionada e a EM é chamada para produzir o melhor vetor para esta partição.

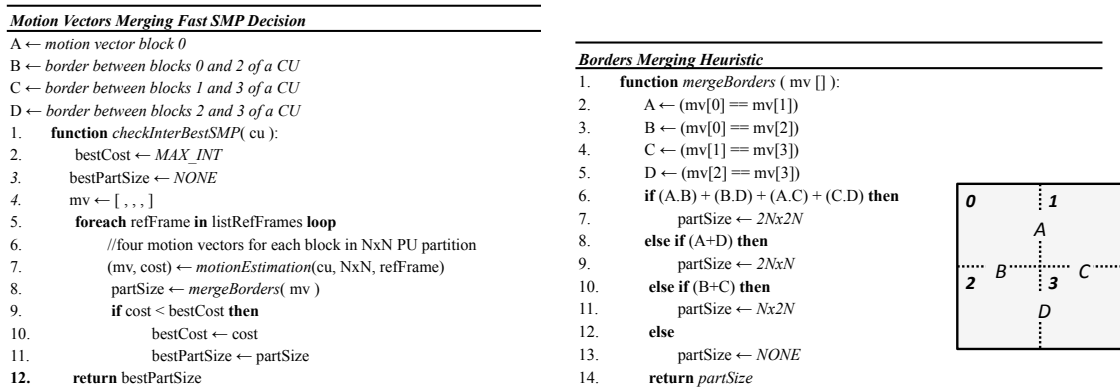


Fig. 5. Algoritmo MVM e sua heurística para decidir a melhor PMS

3.2. Border Strength

Essa métrica foi proposta em [Corrêa et al, 2010] e adaptada nesse trabalho para o contexto do HEVC. Ela é utilizada para decidir qual PMA é mais provavelmente a melhor entre as quatro possíveis. O algoritmo *Border Strength* identifica regiões com bordas em uma determinada área (nesse contexto, dentro de uma CU). A Equação (1) é utilizada para calcular a força de cada borda. O cálculo consiste em acumular as diferenças absolutas entre um determinado grupo de amostras dentro de uma CU.

$$BS = \sum_{i=0}^{CUsize} \sum_{k=0}^{CUsize/8} |p_{(i,k)} - q_{(i,k)}| \quad (1)$$

Após todas as forças de borda serem calculadas, outro cálculo é aplicado, a fim de decidir qual borda é a mais forte. Cada PMA é mapeada para um dos quatro resultados de força de borda: (1) borda superior para partições 2NxnU; (2) borda inferior para 2NxnD; (3) borda esquerda, nLx2N; e (4) borda direita para as partições nRx2N. O algoritmo *Border Strength* é mostrado na Figura 6.

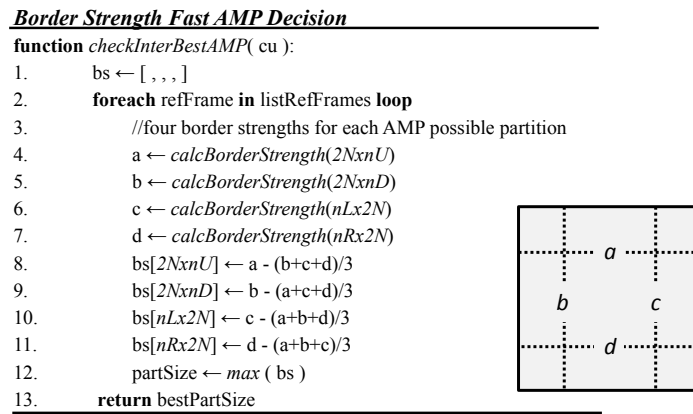


Fig. 6. Algoritmo *Border Strength* utilizado para decidir a melhor PMA

Depois de ter as partições decididas pelo MVM e pelo BS, uma comparação final é realizada entre os dois custos RD atingidos e a partição de PU para a CU atual é finalmente decidida.

O MSBSD diminui drasticamente o número de chamadas à EM no processo de codificação. Mesmo que o HM implemente as otimizações mencionadas nas seções anteriores, o número de chamadas à EM usando a técnica RDO é ainda maior em qualquer situação. A Tabela 2 demonstra esta afirmação, mostrando as partições de PU avaliadas usando a técnica RDO e o algoritmo MSBSD.

Tabela 2. Partições avaliadas considerando RDO e MSBSD

Partição	RDO	MSBSD
Simétrica	3 - 4	1 - 2
Assimétrica	0 - 4	1
Total	4 - 7	2 - 3

4. Resultados

Os resultados utilizados como referência nas comparações realizadas neste trabalho foram extraídos do HM 5.1 utilizando a técnica RDO. Diversos vídeos foram testados e

os resultados para dois deles são mostrados na Figura 7, através da curva PSNR vs. bitrate. Os pontos no gráfico correspondem a diferentes Parâmetros de Quantização (Quantization Parameters – QPs). De forma geral, esse parâmetro controla a força utilizada na etapa de quantização. Quanto mais forte for a quantização, maiores são as taxas de compressão, porém as perdas em qualidade também aumentam.

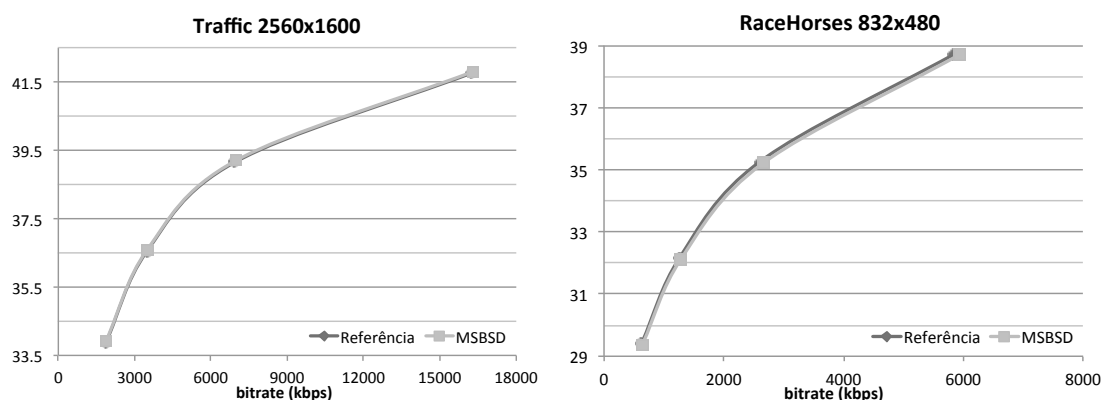


Fig. 7. Gráficos de PSNR vs. bitrate para duas sequências

Como pode ser observado na Figura 7, existe uma diferença insignificante entre os resultados obtidos com a técnica RDO e com o algoritmo MSBSD. Para o vídeo RaceHorses a diferença entre os resultados é mais visível. Isso acontece porque este vídeo possui bastante movimento, o que diminui as chances das heurísticas acertarem na sua decisão. Por outro lado, para o vídeo Traffic, os resultados foram praticamente os mesmos e as linhas traçadas são quase que totalmente sobrepostas.

Diversos trabalhos que propõem soluções para a complexidade na codificação de vídeo podem ser encontrados na literatura, porém poucos apresentam soluções de complexidade para o HEVC especificamente. Trabalhos voltados a outros padrões não representam uma comparação justa, pois as soluções não podem ser diretamente importadas para o HEVC, visto que este padrão emergente possui estruturas de codificação mais complexas. O trabalho de [Corrêa et al, 2011] propõe um módulo de controle de complexidade no HM para diminuir o tempo de codificação, focando aplicações com restrições de energia. Se a bateria do dispositivo estiver com carga baixa, o processo de codificação desce poucos níveis na árvore quaternária, pulando a avaliação em CUs menores. [Leng et al, 2011] propõe um algoritmo de decisão rápida para as CUs. A aceleração é alcançada desconSIDERANDO o processamento de algumas CUs, baseando-se em informações de quadros anteriores e resultados de codificação das CUs vizinhas. Os resultados destes trabalhos comparados aos resultados do MSBSD encontram-se na Tabela 3.

Tabela 3. Comparação com trabalhos relacionados

Trabalho	MSBSD	Corrêa et al, 2010	Leng et al, 2011
Estrutura de Codificação	PU	CU	CU
Δ PSNR (média)	-0,006 dB	-0,01 dB	-0,04 dB
Δ Bitrate (média)	0,948%	1,26%	-0,01%
Δ Tempo de Execução	-37,11%	-38%	-44%

A comparação deixa claro que a solução proposta por este trabalho atingiu resultados expressivos de redução de complexidade. O MSBSD apresenta a menor

degradação de qualidade entre os três trabalhos, com um aumento pequeno no bitrate. Além disso, o algoritmo MSBSD pode ser estendido para outras estruturas de codificação, de forma a diminuir ainda mais o tempo de codificação.

5. Conclusões

Este artigo apresentou o algoritmo Motion Similarity Border Strength Decision, cujo objetivo é diminuir a complexidade no processo de codificação do padrão emergente HEVC. Para atingir esse objetivo, foram desenvolvidas soluções heurísticas que decidem de forma antecipada as PUs avaliadas pela inter-predição. Os resultados mostram redução significativa no tempo de codificação (37,11% em média), com perdas inexpressivas de qualidade e bitrate (0,006 dB e 0,948% respectivamente). Quando comparado com outras soluções propostas na literatura, o MSBSD provou ser uma solução eficiente e competitiva, atingindo resultados tão expressivos quanto os demais trabalhos. Além disso, os cálculos introduzidos pelo MSBSD na codificação são triviais, o que facilita futuras implementações em hardware.

Como trabalhos futuros, pretende-se estudar heurísticas rápidas para outras estruturas de codificação, gerando um algoritmo capaz de controlar todas as etapas de decisão presentes na codificação HEVC. Além disso, soluções em hardware deste algoritmo também são consideradas.

Referências

- Bross, B., Han, W.-J., Sullivan, G. J. et al (2011). "High Efficiency Video Coding (HEVC) text specification Working Draft 5," Reunião JCT-VC em Genebra, Suíça.
- Corrêa, G., Palomino, D. et al (2010). "Homogeneity and Distortion-Based Intra Mode Decision Architecture for H.264/AVC," In *ICECS*.
- Corrêa, G. et al (2011) "Complexity Control of High Efficiency Video Encoders for Power-Constrained Devices," In *IEEE TCE*, vol. 57, pág. 1866-1874.
- JCT-VC (2012). "JCT-VC Group", <http://www.itu.int/en/ITU-T/studygroups/com16/video/Pages/jctvc.aspx>, março.
- Leng, J., Sun, L., Ikenaga, T., e Sakaida, S. (2011) "Content Based Hierarchical Fast Coding Unit Decision Algorithm for HEVC," In *CMSP*, pág.56-59.
- McCann, K. et al (2010). "Video coding technology proposal by Samsung (and BBC)," In *Reunião JCT-VC* em Dresden, Alemanha.
- McCann, K., Bross, B. et al (2011). "High Efficiency Video Coding (HEVC) Test Model 5 (HM 5) Encoder Description," In *Reunião JCT-VC* em Genebra, Suíça.
- Sampaio, F., Grellert, M., Bampi, S., Mattos, J. C. B e Agostini, L. (2012). "Motion Vectors Merging: A Low Complexity Prediction Unit Decision Heuristic For the Inter-Prediction of HEVC Encoders" In *ICME*. (aceito para publicação)
- SBC (2012). "Grandes Desafios da Pesquisa em Computação no Brasil – 2006 – 2016", <http://www.sbc.org.br/>, março.
- Wiegand, T., Sullivan, G. J., Bjontegaard, G. e Luthra, A. (2003). "Overview of the H.264/AVC video coding standard," In *IEEE TCSVT*, vol. 13, pág. 560–576.