

Algoritmo de Eliminações Sucessivas em Níveis baseado na Soma das Diferenças Transformadas Absolutas

Luiz Cancellier¹, Ismael Seidel¹ (coorientador), José Luís A. Güntzel¹ (orientador)

¹Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brazil

{luizhenriquecancellier,ismaelseidel}@gmail.com, j.guntzel@ufsc.com.br

Abstract. *The increasing resolutions combined with storage and processing limitations of mobile devices demand new compression techniques for video coding. Therefore, to achieve higher compression rates without compromising quality, the coding process becomes more and more complex. The most expensive step in video coding is the Motion Estimation, that consists of the search by a candidate block that minimizes a metric. This work proposes a new elimination criterion to Multilevel Successive Elimination Algorithm which is based on the Sum of Absolute Transformed Differences similarity metric. In the worst cases, the criterion eliminates 25% and 69% of evaluated candidates on Fractional and Integer Motion Estimation, respectively.*

Resumo. *O aumento das resoluções e as limitações impostas por dispositivos móveis tornaram necessária a adoção de novas técnicas de codificação de vídeo. Assim, visando alcançar taxas de compressão mais elevadas sem comprometer a qualidade, a codificação tem se tornado cada vez mais complexa. A etapa mais intensiva da codificação, do ponto de vista computacional, é a Estimção de Movimento (ME), a qual consiste na busca do bloco que minimize uma métrica. Este trabalho propõe um novo critério de eliminação de blocos para o Algoritmo de Eliminações Sucessivas em Níveis baseado na Soma das Diferenças Transformadas Absolutas. No pior caso, o critério eliminou 25% e 69% dos candidatos avaliados na ME Fracionária e Inteira, respectivamente.*

1. Introdução

Um vídeo é uma sequência de imagens, chamadas de quadros, apresentadas rapidamente no tempo. O armazenamento de todos os quadros de um vídeo não codificado é proibitivo devido ao grande volume de dados usados para sua representação [Agostini 2007]. Tal volume torna-se ainda maior com o aumento da resolução do vídeo. Dessa forma, torna-se necessário o desenvolvimento de novos padrões, como por exemplo o padrão estado da arte de Codificação de Vídeo de Alta Eficiência - *High Efficiency Video Coding* (HEVC) [ITU-T 2013].

Uma versão simplificada do fluxo de codificação é apresentada na Figura 1. Cada quadro não codificado, chamado de quadro original, é particionado em blocos chamados de originais. Para cada bloco original (Ori), a predição busca, em um conjunto de candidatos, por um bloco de referência (Ref) que minimize uma função de custo. A diferença entre o bloco original e a referência resulta no resíduo (Res), que será transformado (T) e

quantizado (Q). Na etapa de quantização, os elementos do bloco transformado são divididos por valores derivados do parâmetro de quantização. Quanto maior o parâmetro, mais os coeficientes serão reduzidos, diminuindo a informação que será usada para representar o bloco e também a qualidade do vídeo codificado. O bloco resultante de todo esse processo ainda passará pela codificação de entropia, onde é feita uma compressão sem perdas. Um bloco codificado é reconstruído com a quantização inversa (Q^{-1}), transformação inversa (T^{-1}) e o resultado é somado com o bloco de referência. Essa reconstrução irá realimentar a etapa de predição, adicionando mais um candidato ao *buffer*, o qual será usado para codificar os próximos blocos [Richardson 2003].

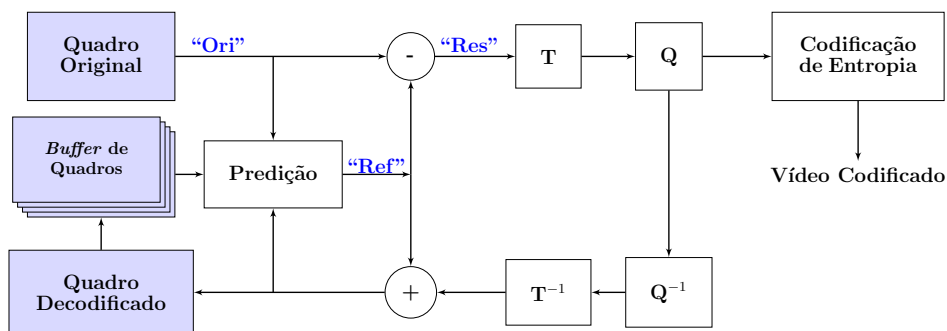


Figura 1. Diagrama simplificado do fluxo de codificação. Adaptado de: [Richardson 2003].

Dentre estas etapas, a predição é uma das mais custosas, do ponto de vista computacional. Apenas a Estimação de Movimento - *Motion Estimation* (ME), que ocorre durante a predição, é responsável por cerca de 40% do tempo total de codificação do HEVC [Bossen et al. 2012]. A ME, ilustrada na Figura 2, explora a redundância de dados que existem em quadros temporalmente próximos.

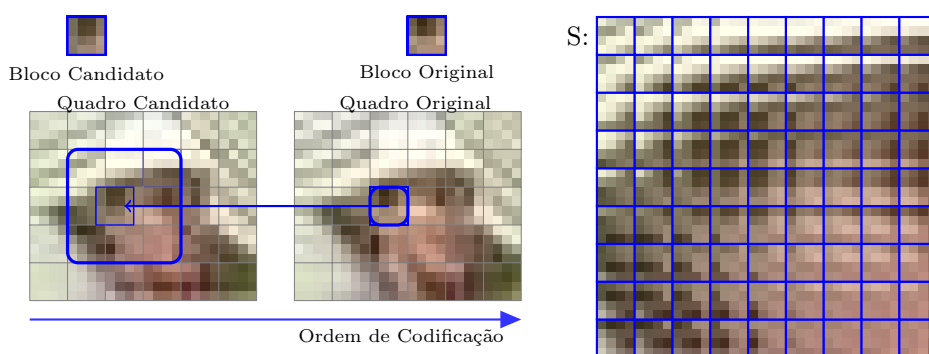


Figura 2. Exemplo do processo de ME. No quadro candidato está destacada a janela de busca e a seta aponta para o candidato que será tomado como referência para codificar o bloco original. O quadro à direita ilustra o espaço (S) dos possíveis candidatos na janela de busca. Fonte: [Seidel 2016].

O algoritmo que fornece o resultado ótimo na ME é a Busca Completa - *Fullsearch* (FS), que aplica a função de custo para todos os blocos candidatos no espaço de busca. Apesar do resultado ótimo, a busca exaustiva faz com que o tempo de execução seja muito elevado. Em função disso, diversos algoritmos rápidos foram propostos com o

objetivo de reduzir o número de candidatos avaliados, trocando o aumento de desempenho por resultados subótimos [Huang et al. 2006]. Visando acelerar a ME e ainda manter o resultado ótimo, Li e Salari propuseram o Algoritmo de Eliminações Sucessivas - *Successive Elimination Algorithm* (SEA) [Li and Salari 1995].

O SEA acelera a busca usando um critério de eliminação para descartar os candidatos impossíveis de serem tomados como referência. Posteriormente, foi feita uma generalização dessa técnica, estabelecendo diferentes níveis de eliminação [Gao et al. 2000]. Tal técnica foi usada no algoritmo chamado Algoritmo de Eliminações Sucessivas em Níveis - *Multilevel Successive Elimination Algorithm* (MSEA). Outros trabalhos foram desenvolvidos com base nos dois supracitados, mas todos se fundamentam no uso da métrica de similaridade Soma das Diferenças Absolutas - *Sum of Absolute Differences* (SAD) para o cálculo da função de custo [Zhu et al. 2005, Trudeau et al. 2015].

Uma métrica alternativa à SAD é a Soma das Diferenças Transformadas Absolutas - *Sum of Absolute Transformed Differences* (SATD), a qual melhora a relação entre a qualidade e a taxa de compressão do vídeo codificado [Wiegand and Schwarz 2016]. Um dos principais fatores que permitem à SATD apresentar melhor eficiência de codificação é a semelhança entre a transformada usada em seu cálculo e a Transformada Discreta dos Cossenos - *Discrete Cosine Transform* (DCT). Como pode ser observado na Figura 1, a DCT é aplicada após a predição, na etapa de transformação (T). Assim, escolher a referência com base em uma transformada que se aproxima da DCT reduz o erro gerado [Wiegand and Schwarz 2016]. Apesar da melhor eficiência de codificação, o cálculo da SATD é mais complexo que a SAD devido ao uso de uma etapa de transformação.

Como apresentado, se por um lado a codificação de vídeo é necessária para reduzir o volume de dados, por outro, este processo é computacionalmente intensivo. Dessa forma, é necessário balancear o compromisso entre taxa de compressão, qualidade e tempo de codificação. Ora, ao se adotar o uso de SATD na ME, consegue-se melhorar o compromisso entre taxa de compressão e qualidade, ao custo de aumento no tempo de codificação. Posto isso, o objetivo deste trabalho é acelerar o processo de ME baseada em SATD e garantir o resultado ótimo. Para isso, primeiramente definimos um critério de eliminação inspirado no SEA, publicado em [Seidel et al. 2016]. Tal como no SEA, a possibilidade de definir múltiplos níveis de eliminação foi desconsiderada. Neste trabalho, feito com base no MSEA, é proposto um novo critério que generaliza a definição anterior para múltiplos níveis, reduzindo o número de candidatos que terão sua SATD calculada.

O restante deste trabalho está organizado da seguinte forma. Na Seção 2 será apresentada a métrica SATD e um primeiro critério de eliminação definido por Seidel [Seidel et al. 2016]. O critério de eliminação proposto neste trabalho será apresentado na Seção 3. O método utilizado para a avaliação da efetividade da proposta é apresentado na Seção 4. Os resultados, por sua vez, serão apresentados e avaliados na Seção 5. Por fim, as conclusões e perspectivas de trabalhos futuros serão apresentadas na Seção 6.

2. Fundamentação

O cálculo da SATD inicia com a subtração do bloco original pelo candidato (Equação 1). Sobre a matriz resultante, é realizada uma transformada de Hadamard (Equação 2) [Agaian et al. 2011], que faz uso de uma matriz de Hadamard (H).

$$D_{2^n \times 2^n} = Ori_{2^n \times 2^n} - Can_{2^n \times 2^n} \quad (1)$$

$$T(D_{2^n \times 2^n}) = H_{2^n \times 2^n} \times D_{2^n \times 2^n} \times H_{2^n \times 2^n} \quad (2)$$

A matriz Hadamard pode ser obtida recursivamente para blocos quadrados de tamanho 2^n , tal que $n \in \mathbb{N}^*$, como apresenta a Equação 3.

$$H_{2^n \times 2^n} = \begin{cases} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & , \text{ se } n = 1 \\ \begin{bmatrix} H_{2^{n-1} \times 2^{n-1}} & H_{2^{n-1} \times 2^{n-1}} \\ H_{2^{n-1} \times 2^{n-1}} & -H_{2^{n-1} \times 2^{n-1}} \end{bmatrix} & , \text{ caso contrário} \end{cases} \quad (3)$$

Tomando os elementos $t_{i,j}$ em $T(D_{2^n \times 2^n})$, a SATD pode ser definida como mostra a Equação 4.

$$\text{SATD}(Ori_{2^n \times 2^n}, Can_{2^n \times 2^n}) = \frac{1}{2^{n-1}} \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} |t_{i,j}| \quad (4)$$

2.1. Primeiro Critério de Eliminação

O SEA se fundamenta na propriedade de subaditividade do módulo (Equação 5) [Schechter 1996] para propor um critério mais simples do que a SAD para eliminar candidatos impossíveis.

$$|a| + |b| \geq |a + b| \quad (5)$$

Aplicando a subaditividade do módulo na SATD obtém-se a Equação 6, na qual o módulo do somatório dos elementos transformados pode ser usado como critério de eliminação. Se durante a avaliação de um candidato a Equação 7 for verdadeira, então ele pode ser descartado, pois, pela Equação 6, sua SATD também será garantidamente maior ou igual à SATD mínima ($satd_{min}$). Da forma como foi apresentado, o cálculo do critério é ineficiente, pois ainda necessita transformar os elementos.

$$\frac{1}{2^{n-1}} \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} |t_{i,j}| \geq \frac{1}{2^{n-1}} \left| \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} t_{i,j} \right| \quad (6)$$

$$\frac{1}{2^{n-1}} \left| \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} t_{i,j} \right| \geq satd_{min} \quad (7)$$

Ao realizar a soma de todos os elementos transformados, é possível reescrever o critério de eliminação como mostra a Equação 8. Esta é usada como critério de eliminação para o algoritmo SEA baseado em SATD. Esse critério foi chamado de Primeira Diferença Absoluta - *Absolute First Difference* (AFD) [Seidel et al. 2016].

$$\frac{1}{2^{n-1}} \left| \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} t_{i,j} \right| = \frac{(2^n)^2}{2^{n-1}} |d_{1,1}| \quad (8)$$

Para ilustrar este resultado, será usada como base uma matriz transformada de tamanho 2×2 . Ao aplicar a transformação sobre a matriz das diferenças, obtém-se a Equação 9. É possível perceber que, quando todos os elementos são somados, todas as diferenças se cancelam, exceto $d_{1,1}$, que será utilizada na AFD.

$$\begin{aligned}
 T(D_{2 \times 2}) &= \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \times \begin{bmatrix} d_{1,1} & d_{1,2} \\ d_{2,1} & d_{2,2} \end{bmatrix} \times \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\
 &= \begin{bmatrix} (+d_{1,1} + \cancel{d_{2,1}} + \cancel{d_{1,2}} + \cancel{d_{2,2}}) & (+d_{1,1} + \cancel{d_{2,1}} - \cancel{d_{1,2}} - \cancel{d_{2,2}}) \\ (+d_{1,1} - \cancel{d_{2,1}} + \cancel{d_{1,2}} - \cancel{d_{2,2}}) & (+d_{1,1} - \cancel{d_{2,1}} - \cancel{d_{1,2}} + \cancel{d_{2,2}}) \end{bmatrix}
 \end{aligned} \tag{9}$$

3. Critério Proposto

Um bloco quadrado de tamanho 2^n pode ser dividido em l níveis, tal que $0 \leq l < n$. O critério $MSATD_l$, proposto neste trabalho, pode ser definido como na Equação 10, na qual (o, p) indexa um elemento da partição (i, j) de tamanho $2^n/2^l$ da matriz das diferenças transformadas. É importante observar que, da forma como foi definida, a Equação 10 mostra uma abordagem ingênua, dado que calcula toda a transformação da matriz de diferenças.

$$MSATD_l = \frac{1}{2^{n-1}} \sum_{i=1}^{2^l} \sum_{j=1}^{2^l} \left| \sum_{o=1}^{2^n/2^l} \sum_{p=1}^{2^n/2^l} \left(T(D)_{i,j} \right)_{o,p} \right| \tag{10}$$

O cálculo do nível $l = 0$ é simplesmente a AFD. Para demonstrar o cálculo do nível 1, serão utilizadas como base as diferenças transformadas para blocos de tamanho 2^n . Neste caso, a $MSATD_1$ é dada pela Equação 11. As transformadas menores, de tamanho $2^n/2^l$, foram obtidas utilizando a multiplicação de matrizes particionadas [Liu et al. 2011].

$$\begin{aligned}
 MSATD_{l=1} &= \frac{1}{2^{n-1}} \left| \sum_{o=1}^{2^n/2^l} \sum_{p=1}^{2^n/2^l} \left(+ T(D_{1,1})_{\frac{2^n}{2^l} \times \frac{2^n}{2^l}} + T(D_{2,1}) + T(D_{1,2}) + T(D_{2,2}) \right)_{o,p} \right| \\
 &+ \frac{1}{2^{n-1}} \left| \sum_{o=1}^{2^n/2^l} \sum_{p=1}^{2^n/2^l} \left(+ T(D_{1,1}) + T(D_{2,1}) - T(D_{1,2}) - T(D_{2,2}) \right)_{o,p} \right| \\
 &+ \frac{1}{2^{n-1}} \left| \sum_{o=1}^{2^n/2^l} \sum_{p=1}^{2^n/2^l} \left(+ T(D_{1,1}) - T(D_{2,1}) + T(D_{1,2}) - T(D_{2,2}) \right)_{o,p} \right| \\
 &+ \frac{1}{2^{n-1}} \left| \sum_{o=1}^{2^n/2^l} \sum_{p=1}^{2^n/2^l} \left(+ T(D_{1,1}) - T(D_{2,1}) - T(D_{1,2}) + T(D_{2,2}) \right)_{o,p} \right|
 \end{aligned} \tag{11}$$

O somatório dos elementos da matriz transformada, $T(D_{i,j})$, já foi demonstrado no critério AFD (Equação 8) e corresponde ao primeiro elemento extraído da partição $D_{i,j}$. É possível reparar também que o padrão de sinais dos elementos é o mesmo que o de uma transformada Hadamard de tamanho 2 (Equação 3). Para os próximos níveis, o mesmo procedimento de multiplicação por particionamento pode ser aplicado, resultando na transformação de partições menores com o mesmo padrão de sinais de uma matriz Hadamard maior. Assim, o critério de eliminação apresentado na Equação 10 pode ser eficientemente calculado conforme apresentado na Equação 12, na qual FD é a matriz composta pela primeira diferença de cada partição. A Figura 3 ilustra como é feito o particionamento e seleção de elementos para a transformação de cada nível.

$$MSATD_l = \frac{(2^{n-l})^2}{2^{n-1}} \sum_{i=1}^{2^l} \sum_{j=1}^{2^l} \left| \left(H_{2^l \times 2^l} \times FD_{2^l \times 2^l} \times H_{2^l \times 2^l} \right)_{i,j} \right| \tag{12}$$

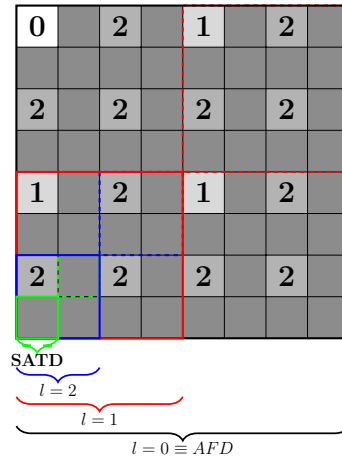


Figura 3. Níveis de particionamento de um bloco 8×8 no algoritmo MSEA-SATD. As posições marcadas com valores menores ou iguais a l compõem a matriz FD usada na métrica $MSATD_l$.

O Algoritmo 1 mostra como o critério de eliminação é aplicado durante a busca. As entradas são um bloco original (Ori) e o espaço de busca (S). Para cada candidato em S , são avaliados os níveis do critério de eliminação (linhas 1 a 8). Se em algum momento o valor do critério for maior que $satd_{min}$, a avaliação para a métrica SATD do candidato atual não é calculada. Se o candidato passar pela avaliação do critério (linha 9), então sua SATD é calculada (linha 10) e se ele for um candidato melhor (linha 11), então a referência e o valor de $satd_{min}$ são atualizados (linhas 12 e 13).

Algoritmo 1: MSEA baseado em SATD

Entrada: Ori, S
Saída: Ref

```

1  $satd_{min} \leftarrow \infty$ ;
2 para cada  $Can \in S$  faça
3   para  $l = 0$  até  $n - 1$  faça
4      $msatd \leftarrow MSATD_l(Ori, Can)$ ;
5     se  $msatd \geq satd_{min}$  então
6       para;
7     fim
8   fim
9   se  $msatd < satd_{min}$  então
10     $satd \leftarrow SATD(Ori, Can)$ ;
11    se  $satd < satd_{min}$  então
12       $satd_{min} \leftarrow satd$ ;
13       $Ref \leftarrow Can$ ;
14    fim
15  fim
16 fim

```

3.1. Análise do Número de Operações

A Tabela 1 apresenta o custo para computar a AFD, $MSATD_{n-1}$ e SATD. É possível perceber que a métrica AFD apresenta um custo baixo e independente do tamanho do bloco. Em contrapartida, a computação de $MSATD_l$ se torna mais complexa à medida que l aumenta. Ainda assim, no último nível ($l = n - 1$) a $MSATD_l$ necessita de apenas cerca de 25% das operações usadas para computar a SATD. Dessa forma, mesmo calculando diretamente o último nível, apesar de consumir mais operações para eliminar candidatos que já seriam impossíveis em níveis anteriores, a computação de $MSATD_{n-1}$ ainda é eficiente em relação à métrica SATD.

Tabela 1. Custo de operações para computar as métricas AFD, $MSATD_{n-1}$ e SATD. Os valores apresentados para transformação e somatório se referem ao número de somas realizadas. O produto por potências de dois é substituído por deslocamentos binários.

Métrica	Diferença	Transformação	Módulo	Somatório	Deslocamento
<i>AFD</i>	1	0	1	0	1
$MSATD_{n-1}$	$\frac{1}{4}(2^n)^2$	$\frac{1}{4}(2^n)^2((2^n)^2 - 4)$	$\frac{1}{4}(2^n)^2$	$\frac{1}{4}(2^n)^2 - 1$	1
<i>SATD</i>	$(2^n)^2$	$(2^n)^2((2^n)^2 - 1)$	$(2^n)^2$	$(2^n)^2 - 1$	1

4. Método

A técnica proposta foi implementada no código de referência do HEVC, o Modelo de Teste do HEVC - *HEVC Test Model* (HM)¹. Os testes foram feitos com base nas Condições Comuns de Teste - *Common Test Conditions* (CTC) [Bossen 2012], usando o arquivo de configuração “*Low-delay P-High efficiency*”. Todas as outras 22 sequências de vídeo com 8 bits por pixel foram codificadas usando os quatro parâmetros de quantização indicados (22, 27, 32 e 37). Neste artigo, serão apresentados apenas os resultados de pior caso, que ocorreram usando o parâmetro de quantização 22.

Foram feitos experimentos em duas etapas da ME. A primeira etapa, chamada de Estimção de Movimento Inteira - *Integer Motion Estimation* (IME), consiste numa busca pelos candidatos de um quadro já codificado. A segunda etapa, chamada de Estimção de Movimento Fracionária - *Fractional Motion Estimation* (FME), faz um refinamento no qual são gerados e avaliados novos candidatos a partir do candidato selecionado na IME.

5. Resultados

A Figura 5 apresenta o percentual acumulado de eliminações em cada nível do algoritmo MSEA. No contexto da FME, é possível perceber que o critério AFD (nível 0) apresenta um baixo percentual de eliminações. Como na FME os candidatos possuem valores muito próximos, o último nível do MSEA tende a ser o único com algum impacto significativo. Aplicando diretamente o nível 2, ainda seriam eliminados, no pior caso, aproximadamente 25% dos candidatos.

Em duas sequências de vídeo, o nível 0 apresentou um percentual de eliminação acima de 50%. Tanto o “*SlideEditing*” quanto o “*SlideShow*” são vídeos atípicos, com pouca movimentação. Pelos resultados obtidos, há um indício de que o critério AFD é uma técnica efetiva para esse nicho de aplicação.

¹Versão 16.6. Disponível em: “https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware”

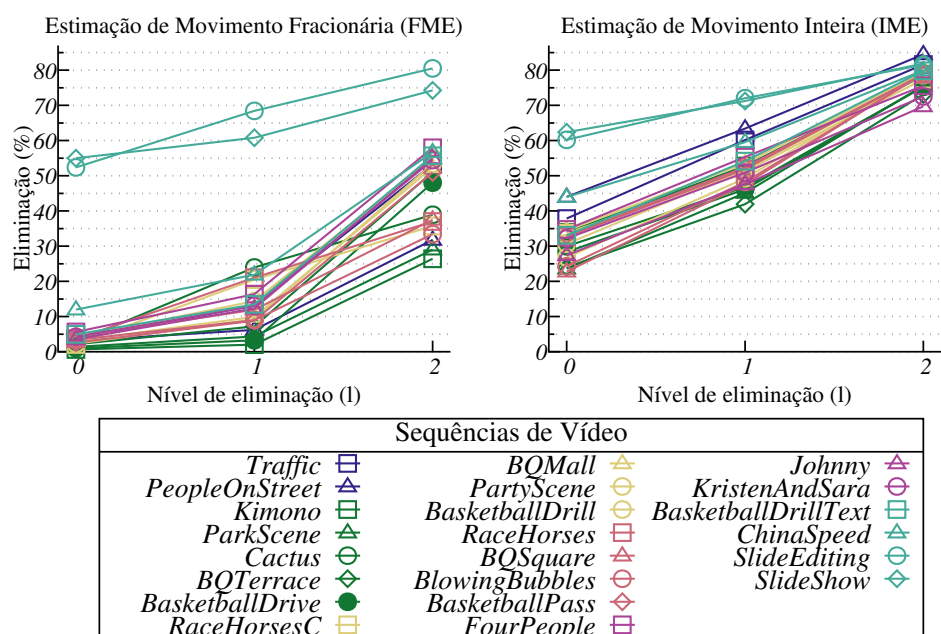


Figura 4. Percentual acumulado de eliminações de candidatos em cada nível na FME e IME.

Na IME, diferente do resultado observado na FME, o uso da AFD apresenta um elevado percentual de eliminação. Isso ocorre porque são avaliados candidatos pouco similares, onde eventualmente um bloco com baixo valor de SATD é encontrado e os outros serão eliminados já nos primeiros níveis. No pior caso, aproximadamente 22,75% dos candidatos foram eliminados, valor considerável para um critério que usa apenas três operações aritméticas.

O uso do algoritmo MSEA também é efetivo. No último nível, as sequências convergem para um percentual de eliminação acima de 65%. Novamente, observa-se a possibilidade de computar apenas o terceiro nível como critério de eliminação.

5.1. Comportamento por Tamanho de Bloco

No HEVC, blocos não quadrados ou maiores que 8×8 são computados como o somatório da SATD de partições de tamanho 8×8 ou 4×4 . Adicionando o somatório das partições em ambos os lados das equações, todas as propriedades definidas nas seções anteriores são mantidas. A Tabela 2 apresenta os resultados de eliminação por tamanho de bloco.

QP	Eliminação (%)		
	4×4	8×8	Total
22	24,02	6,40	30,42
27	26,46	7,46	33,93
32	29,63	9,56	39,18
37	32,84	12,37	45,21

Tabela 2. Percentual médio de eliminações de candidatos compostos por blocos 4×4 e 8×8 usando o critério AFD.

Nos experimentos realizados, foi possível observar que o critério de eliminação AFD foi mais efetivo para eliminar blocos compostos por 4×4 do que blocos compostos

por 8×8 . Para entender tal efeito, é necessário observar o comportamento dos critérios de eliminação e da SATD. Para isso será tomada como base a matriz de diferenças que resulta no maior valor possível de SATD. Tal matriz, que será chamada de D' , foi obtida com base na propriedade definida na Equação 13 [Agaian et al. 2011], onde I é a matriz identidade. Tomando D' como mostra a Equação 14, onde v é o maior valor que o pixel pode assumir, a propriedade pode ser aplicada na transformação, resultando na Equação 15.

$$H_{2^n \times 2^n} \times H_{2^n \times 2^n} = 2^n \times I_{2^n \times 2^n} \quad (13)$$

$$D'_{2^n \times 2^n} = v \times H_{2^n \times 2^n} \quad (14)$$

$$T(D'_{2^n \times 2^n}) = H_{2^n \times 2^n} \times (v \times H_{2^n \times 2^n}) \times H_{2^n \times 2^n} = 2^n \times v \times H_{2^n \times 2^n} \quad (15)$$

Através da Equação 15, os resultados de AFD, $MSATD_{n-1}$ e SATD para a matriz D' são dados pelas equações abaixo:

$$AFD = 2 \times 2^n v \quad (16)$$

$$MSATD_{l=n-1} = (2^n)^2 v \quad (17)$$

$$SATD = 2 \times (2^n)^2 v \quad (18)$$

É possível reparar que o comportamento da AFD tem crescimento linear com relação ao tamanho do bloco, enquanto o comportamento da métrica SATD é quadrático. Isso justifica o maior percentual de eliminações para os blocos menores, uma vez que a diferença entre as métricas também será menor. Quanto ao critério de eliminação $MSATD_{n-1}$, é possível perceber que ele também apresenta o comportamento quadrático, justificando assim o maior percentual de eliminação de candidatos no último nível.

6. Conclusões e Trabalhos Futuros

Neste trabalho foi proposto um novo critério de eliminação, chamado $MSATD_l$, que generaliza o SEA baseado em SATD para múltiplos níveis. Este critério demonstrou ser efetivo, principalmente no terceiro nível. No pior caso, foram eliminados aproximadamente 24% dos candidatos avaliados na FME. Quando aplicado na IME, os resultados foram ainda melhores, eliminando 69% dos candidatos. Além de apresentar melhor taxa de eliminações no terceiro nível, o $MSATD_l$ computa apenas 25% das operações necessárias para calcular a SATD, demonstrando ser não somente efetivo como também eficiente.

Trabalhos futuros envolvem o estudo da relação entre diferentes níveis do critério $MSATD_l$. Um exemplo é reusar as operações das transformadas de níveis menores como cálculo parcial dos maiores, evitando o recálculo de operações. Também deve ser feita a avaliação de desempenho ao usar apenas o último nível de eliminação, visando verificar o impacto de usar a operação mais custosa, mesmo quando o último nível não fosse necessário. Além das pesquisas com o critério proposto neste trabalho, também é possível buscar novos critérios baseados em diferentes construções da matriz de Hadamard, bem como outras formas de particionamento além da proposta original do MSEA.

Referências

- Agaian, S., Sarukhanyan, H., Egiazarian, K., and Astola, J. (2011). *Hadamard Transforms*. SPIE Press Monograph Vol. PM207. SPIE Press.
- Agostini, L. V. (2007). Desenvolvimento de arquiteturas de alto desempenho dedicadas à compressão de vídeo segundo o padrão H.264/AVC. Tese de doutorado, UFRGS.
- Bossen, F. (2012). Common test conditions and software reference configurations. Document JCTVC-K1100, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Shanghai.
- Bossen, F., Bross, B., Suhring, K., and Flynn, D. (2012). HEVC complexity and implementation analysis. *IEEE Trans. on Circ. Syst. for Video Technology*, 22(12):1685–1696.
- Gao, X., Duanmu, C., and Zou, C. (2000). A multilevel successive elimination algorithm for block matching motion estimation. *IEEE Trans. on Image Process.*, 9(3):501–504.
- Huang, Y.-W., Chen, C.-Y., Tsai, C.-H., Shen, C.-F., and Chen, L.-G. (2006). Survey on block matching motion estimation algorithms and architectures with new results. *J. VLSI Signal Process. Syst.*, 42(3):297–320.
- ITU-T (2013). Recommendation ITU-T H.265: High efficiency video coding. Recommendation H.265, International Telecommunication Union, Geneva.
- Li, W. and Salari, E. (1995). Successive elimination algorithm for motion estimation. *IEEE Trans. on Circ. Syst. for Video Technology*, 4(1):105–107.
- Liu, J., Chen, X., Fan, Y., and Zeng, X. (2011). A full-mode FME VLSI architecture based on 8x8/4x4 adaptive Hadamard transform for QFHD H.264/AVC encoder. In *2011 IEEE/IFIP 19th International Conference on VLSI and System-on-Chip*, pages 434–439.
- Richardson, I. E. G. (2003). *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons Inc.
- Schechter, E. (1996). *Handbook of Analysis and Its Foundations*. Elsevier Science.
- Seidel, I. (2016). *Exploiting SATD Properties to Reduce Energy in Video Coding*. Exame de qualificação (doutorado), UFSC, Florianópolis-SC.
- Seidel, I., Cancellier, L., Güntzel, J. L., and Agostini, L. (2016). Rate-constrained successive elimination of Hadamard-based SATDs. In *IEEE International Conference on Image Process. (ICIP)*, pages 2395–2399.
- Trudeau, L., Coulombe, S., and Desrosiers, C. (2015). An adaptive search ordering for rate-constrained successive elimination algorithms. In *IEEE International Conference on Image Process. (ICIP)*, pages 207–211.
- Wiegand, T. and Schwarz, H. (2016). Video coding: Part II of fundamentals of source and video coding. *Foundations and Trends in Signal Processing*, 10(1–3):1–346.
- Zhu, C., Qi, W.-S., and Ser, W. (2005). Predictive fine granularity successive elimination for fast optimal block-matching motion estimation. *IEEE Trans. on Image Process.*, 14(2):213–221.