Algoritmo de Eliminações Sucessivas baseado em Soma das Diferenças Transformadas Absolutas

Luiz Henrique De L. Cancellier[1]

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

l.h.cancellier@grad.ufsc.br

Resumo. *A Estimação de Movimento - Motion Estimation (ME) é a etapa computacionalmente mais intensiva da codificação de vídeo digital. Ela consiste na busca por um bloco que minimize uma métrica de similaridade para ser tomado como referência. Baseando-se no SEA, este trabalho propõe dois critérios de eliminação de cálculos da métrica de similaridade SATD. O primeiro critério, chamado de AFD, teve bom resultados na ME inteira, com 23% de candidatos eliminados no pior caso, enquanto segundo critério,chamado de MSATD, eliminou 24% e 69% 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 conforme as resoluções aumentam. Assim, ao adotarem-se resoluções como 2160p (3840×2160 pixels) e 4320p (7680×4320), a codificação de vídeos se torna mandatória.

Se por um lado a codificação de vídeo é necessária para reduzir o volume de dados, por outro este processo é computacionalmente intensivo [Bossen et al. 2012]. Dessa forma, é necessário otimizar a codificação visando atingir uma taxa de compressão aceitável, controlando-se as perdas na qualidade do vídeo e reduzindo o tempo gasto ao longo do processo.

O fluxo de codificação é apresentado na Figura 1. Cada quadro não codificado, que será chamado de quadro original, é particionado em pequenos blocos, chamados de blocos originais ("Ori"). Para cada bloco original, a etapa de predição realiza uma busca entre os blocos candidatos. O candidato que mais se assemelha ao original será tomado como referência ("Ref").

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 valores do bloco transformado serão reduzidos de acordo com o Parâmetro de Quantização - *Quantization Parameter* (QP) definido. Quanto maior o QP, mais os coeficientes serão reduzidos, ou até mesmo zerados. Dessa forma, menos informação será usada para representar o bloco e também será pior a qualidade do vídeo codificado. O bloco resultante de todo esse processo ainda será codificado por entropia, onde o dado é comprimido 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

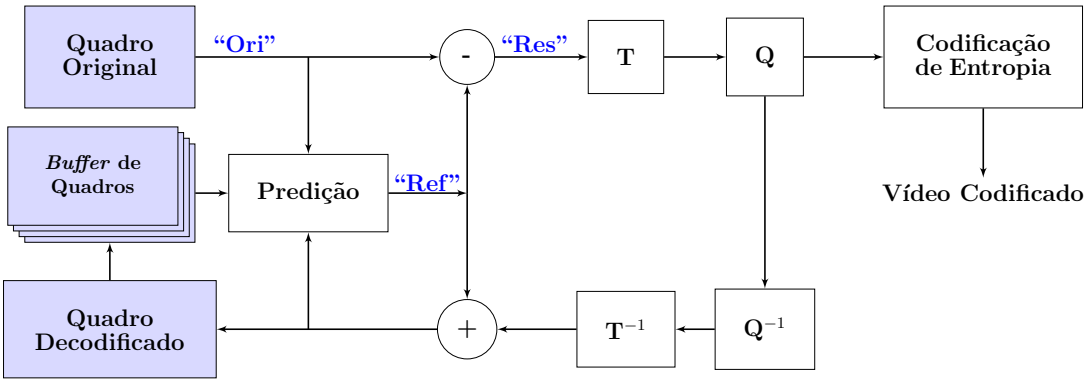

Figura 1. Diagrama simplificado do fluxo de codificação. Adaptado de: [Cancellier et al. 2015].

reconstrução irá realimentar a etapa de predição, adicionando ao *buffer* mais um candidato que será usado para codificar os próximos blocos [Richardson 2002].

A etapa de predição explora a redundância de informação entre blocos. Quando áreas de um mesmo quadro são muito semelhantes, tal característica é conhecida como redundância espacial. Também existe o conceito de redundância temporal, que ocorre quando quadros temporalmente próximos possuem áreas semelhantes [Shi and Sun 1999].

Para explorar a redundância temporal, é executado o processo de ME. O bloco referência é aquele que, dentre os possíveis candidatos em uma área de busca restrita e predeterminada no quadro candidato já codificado, apresenta maior semelhança com o original. Para determinar o quão semelhantes são dois blocos é usado um critério chamado de métrica de similaridade. A ME se resume então à busca pelo candidato que minimize o resultado da métrica de similaridade adotada.

O algoritmo mais conhecido e trivial para ME é a Busca Completa - *Fullsearch* (FS), que aplica a métrica de similaridade para todos os blocos candidatos da janela de busca. A busca intensiva na FS fornece resultado ótimo, porém seu custo computacional é muito elevado. Em função disso, diversos algoritmos rápidos foram propostos com objetivo de reduzir o número de candidatos avaliados, trocando o aumento de desempenho por resultados subótimos [Huang et al. 2006]. Por outro lado, existem algoritmos como o Algoritmo de Eliminações Sucessivas - *Successive Elimination Algorithm* (SEA) [Li and Salari 1995], que utilizam propriedades da métrica de similaridade para eliminar candidatos impossíveis mediante o uso de cálculos mais simples. Um candidato é dito impossível quando sabe-se antes do cálculo da métrica de similaridade que ele não será tomado como referência. Dessa forma, é possível aplicar essas técnicas de eliminação para acelerar o FS e ainda manter os resultados ótimos.

As métricas de similaridade mais usadas em codificação de vídeo são a Soma das Diferenças Absolutas - *Sum of Absolute Differences* (SAD), a Soma das Diferenças Quadráticas - *Sum of Squared Differences* (SSD) e a Soma das Diferenças Transformadas Absolutas - *Sum of Absolute Transformed Differences* (SATD) [Richardson 2003]. Por permitir resultados aceitáveis de qualidade de codificação e ainda ter cálculo bastante simples, a métrica SAD é a mais utilizada.

A SATD, por sua vez, apresenta cálculo mais complexo que a SAD, computando uma transformada Hadamard sobre a matriz de diferenças. Com relação à eficiência de codificação, os resultados obtidos com o uso da SATD são melhores que as demais métricas citadas [Dominges et al. 2011]. Um dos principais fatores que permitem à SATD apresentar melhor eficiência de codificação é que a transformada utilizada em seu cálculo está correlacionada com a Transformada Discreta dos Cossenos - *Discrete Cosine Transform* (DCT) [Zhu and Xiong 2009]. A DCT é aplicada no processo de transformação, que ocorre após a ME, sobre a matriz de diferenças entre o bloco original e o referência. Desta forma, escolher a referência com base numa transformada que se aproxima da DCT reduz o erro gerado [Dominges et al. 2011].

Apesar de apresentar melhores resultados na eficiência de codificação, o uso da SATD em um processo de ME é proibitivo por conta do alto custo para computar a transformada Hadamard. Para reduzir tal custo, serão desenvolvidas técnicas baseadas no SEA para evitar a computação completa da transformada.

O restante deste trabalho está organizado da seguinte forma. Na Sessão 2 será apresentado em detalhes a métrica SATD. Na Sessão 3 serão formalizados os dois critérios de eliminação de candidatos para o cálculo de SATD e os resultados obtidos serão apresentados e avaliados na Sessão 4. Por fim, as conclusões serão apresentadas na Sessão 5.

2. Soma das Diferenças Transformadas Absolutas

Para definir o cálculo da métrica de similaridade SATD, inicialmente é necessário computar a diferença entre os blocos original e candidato, como apresenta a Equação 1. Após computar as diferenças, é preciso aplicar uma transformação sobre ela. Para isso, a SATD usa a transformada Hadamard (Equação 2), que faz uso das matrizes de Hadamard (H).

$$D_{2^n \times 2^n} = Ori_{2^n \times 2^n} - Can_{2^n \times 2^n} \tag{1}$$

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

A matriz Hadamard pode ser obtida recursivamente, como apresenta a Equação 3. O símbolo "$\otimes$" representa o produto de Kronecker [Weisstein 2009], que para esse caso pode ser expandido como mostra a Equação 4. Esta última equação será importante para provar as técnicas de eliminação propostas.

$$H_{2^n \times 2^n} = \begin{cases} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & \text{se } n = 1 \\ \\ \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes H_{2^{n-1} \times 2^{n-1}} & \text{se } n > 1 \end{cases} \tag{3}$$

$$H_{2^n \times 2^n} = \begin{bmatrix} 1 \times H_{2^{n-1} \times 2^{n-1}} & 1 \times H_{2^{n-1} \times 2^{n-1}} \\ 1 \times H_{2^{n-1} \times 2^{n-1}} & -1 \times H_{2^{n-1} \times 2^{n-1}} \end{bmatrix} \tag{4}$$

Por fim, tomando os elementos $t_{i,j} \in T(D_{2^n \times 2^n})$, a SATD pode ser definida como mostra a Equação 5.

$$SATD(Ori_{2^n \times 2^n}, Can_{2^n \times 2^n}) = \frac{1}{2^{n-1}} \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} |t_{i,j}| \tag{5}$$

3. Critérios de Eliminação

O SEA se fundamenta na propriedade de subaditividade do módulo (Equação 6) para propor um critério mais simples de similaridade para eliminar candidatos impossíveis, ou seja, candidatos que garantidamente não serão tomados como referência.

$$|a| + |b| \geq |a + b| \tag{6}$$

Aplicando a propriedade de subaditividade do módulo na SATD, obtém-se a Equação 7. Da forma como foi apresentada, essa última equação mostra um critério de eliminação que ainda necessita da transformação dos elementos, que é justamente a operação que torna a SATD tão custosa. Para evitar a transformação completa, foram propostos dois critérios de eliminação que também baseadas na propriedade de subaditividade do módulo.

$$\frac{1}{2^{n-1}} \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} |t_{i,j}| \geq \frac{1}{2^{n-1}} | \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} t_{i,j}| \tag{7}$$

3.1. Primeira Diferença Absoluta

Partindo da soma dos elementos $t_{i,j}$, deseja-se definir um critério de eliminação que possa ser computado de forma mais eficiente. Para isso, será tomada como hipótese a Equação 8, onde nenhuma transformação precisa ser computada. Para provar esse resultado, será usado o método de prova por indução matemática.

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

Inicialmente será demonstrado que a propriedade é válida para o passo base. Tomando n, tal que $n = 1$, a transformada é aplicada sobre D como segue:

$$\begin{aligned} T(D_{2\times2}) &= \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \times \begin{bmatrix} d_{1,1} & d_{1,2} \\ d_{2,1} & d_{2,2} \end{bmatrix} \times \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ &= \begin{bmatrix} (+d_{1,1} + d_{2,1} + d_{1,2} + d_{2,2}) & (+d_{1,1} + d_{2,1} - d_{1,2} - d_{2,2}) \\ (+d_{1,1} - d_{2,1} + d_{1,2} - d_{2,2}) & (+d_{1,1} - d_{2,1} - d_{1,2} + d_{2,2}) \end{bmatrix} \end{aligned} \tag{9}$$

Ao manipular $T(D_{2\times2})$ na Equação 8, como apresentado em 10, é possível observar que todos os elementos, exceto $d_{1,1}$, se anulam, resultando na Equação 11. Constata-se então que a propriedade definida inicialmente é de fato válida para o caso base.

$$\frac{1}{2^0}|\sum_{i=1}^{2^1}\sum_{j=1}^{2^1} t_{i,j}| = |(+d_{1,1}+d_{2,1}+d_{1,2}+d_{2,2})+(+d_{1,1}+d_{2,1}-d_{1,2}-d_{2,2}) \\ +(+d_{1,1}-d_{2,1}+d_{1,2}-d_{2,2})+(+d_{1,1}-d_{2,1}-d_{1,2}+d_{2,2})| \tag{10}$$

$$|\sum_{i=1}^{2^1}\sum_{j=1}^{2^1} t_{i,j}| = 4 \times |d_{1,1}| \tag{11}$$

No passo indutivo será assumido que, para um valor k arbitrário, a propriedade definida na Equação 8 é válida. Por fim, é necessário provar que a propriedade também se aplica a $n = k + 1$. As matrizes envolvidas no cálculo serão particionadas como mostra a Equação 12 e as multiplicações da transformada serão feitas por particionamento. Essa multiplicação é feita de forma semelhante à usual, porém os fatores são partições das matrizes[Rowland 2006]. Após aplicar a transformação, obtém-se a Equação 13.

$$T(D_{2^n\times 2^n}) = H_{2^{k+1}\times 2^{k+1}} \times D_{2^{k+1}\times 2^{k+1}} \times H_{2^{k+1}\times 2^{k+1}} \\ = \left[\begin{array}{c|c} H_{2^k\times 2^k} & H_{2^k\times 2^k} \\ \hline H_{2^k\times 2^k} & -H_{2^k\times 2^k} \end{array}\right] \times \left[\begin{array}{c|c} (D_{2^k\times 2^k})_{1,1} & (D_{2^k\times 2^k})_{1,2} \\ \hline (D_{2^k\times 2^k})_{2,1} & (D_{2^k\times 2^k})_{2,2} \end{array}\right] \times \left[\begin{array}{c|c} H_{2^k\times 2^k} & H_{2^k\times 2^k} \\ \hline H_{2^k\times 2^k} & -H_{2^k\times 2^k} \end{array}\right] \tag{12}$$

$$\left[\begin{array}{c|c} T(D_{1,1})+T(D_{2,1})+T(D_{1,2})+T(D_{2,2}) & T(D_{1,1})+T(D_{2,1})-T(D_{1,2})-T(D_{2,2}) \\ \hline T(D_{1,1})-T(D_{2,1})+T(D_{1,2})-T(D_{2,2}) & T(D_{1,1})-T(D_{2,1})-T(D_{1,2})+T(D_{2,2}) \end{array}\right] \tag{13}$$

É interessante observar que o padrão de sinais do caso base se repete entre as partições da matriz transformada. Ao somar as partições, todas se anulam exceto $T(D_{1,1})$. O resultado do módulo da soma dos elementos de uma matriz de diferenças transformadas de tamanho $2^k \times 2^k$ foi assumido no início do passo indutivo e será substituído em 7, como mostra a Equação 14. Como $n = k + 1$, essa última equação pode ser reescrita de forma que seja possível concluir que a hipótese é válida.

$$\frac{1}{2^k}|\sum_{i=1}^{2^n}\sum_{j=1}^{2^n} t_{i,j}| = \frac{(2^{k+1})^2}{2^k}|d_{1,1}| \tag{14}$$

Utilizando a Equação 8, cálculos de SATD podem ser poupados usando um critério de seleção baseado apenas no primeiro elemento da matriz de diferenças. Esse critério será chamado de Primeira Diferença Absoluta - *Absolute First Difference* (AFD) e é definido na Equação 15. Como na ME deseja-se encontrar o candidato que minimiza o valor de SATD, se um bloco sendo avaliado apresentar AFD maior ou igual à SATD do melhor candidato selecionado até aquele momento, esse bloco pode ser descartado. Isso é possível pois garantidamente a SATD do bloco descartado também será maior que

$$AFD(Ori_{2^n\times 2^n}, Can_{2^n\times 2^n}) = \frac{(2^n)^2}{2^{n-1}}|d_{1,1}| \tag{15}$$

3.2. Eliminações Sucessivas em Níveis

Uma técnica mais genérica de eliminação faz sucessivos particionamentos da matriz de diferenças para eliminar um número maior de candidatos enquanto se aproxima do cálculo final da métrica de similaridade. Esse critério é definido para computar o Algoritmo de Eliminações Sucessivas em Níveis - *Multilevel Successive Elimination Algorithm* (MSEA) [Gao et al. 2000] e também pode ser aplicado à SATD.

No al ser dividido em l níveis, tal que $0 \leq l < n$ para blocos quadrados de tamanho 2^n. A métrica $MSATD_l$ é definida como mostra a Equação 16, onde (o, p) indexa um elemento da partição (i, j) da matriz das diferenças transformadas.

$$\mathrm{MSATD}_l = \frac{1}{2^{n-1}} \sum_{i=1}^{2^l} \sum_{j=1}^{2^l} | \sum_{o=1}^{2^n/2^l} \sum_{p=1}^{2^n/2^l} \Big(T(D_{i,j}) \Big)_{o,p} | \tag{16}$$

O cálculo do nível $l = 0$ é simplesmente o AFD. Para demonstrar o cálculo do nível 1 será usado como base as diferenças transformadas para blocos de tamanho 2^n, onde $n = k+1$, que é dada pela Equação 13. Neste caso a MSATD_1 é dada pela equação:

$$\begin{aligned}
\mathrm{MSATD}_{l=1} = & \frac{1}{2^k} | \sum_{o=1}^{2^n/2^l} \sum_{p=1}^{2^n/2^l} \Big(+ T(D_{1,1}) + T(D_{2,1}) + T(D_{1,2}) + T(D_{2,2}) \Big)_{o,p} | \\
& + \frac{1}{2^k} | \sum_{o=1}^{2^n/2^l} \sum_{p=1}^{2^n/2^l} \Big(+ T(D_{1,1}) + T(D_{2,1}) - T(D_{1,2}) - T(D_{2,2}) \Big)_{o,p} | \\
& + \frac{1}{2^k} | \sum_{o=1}^{2^n/2^l} \sum_{p=1}^{2^n/2^l} \Big(+ T(D_{1,1}) - T(D_{2,1}) + T(D_{1,2}) - T(D_{2,2}) \Big)_{o,p} | \\
& + \frac{1}{2^k} | \sum_{o=1}^{2^n/2^l} \sum_{p=1}^{2^n/2^l} \Big(+ T(D_{1,1}) - T(D_{2,1}) - T(D_{1,2}) + T(D_{2,2}) \Big)_{o,p} |
\end{aligned} \tag{17}$$

O somatório dos elementos de $T(D_{i,j})$ é conhecido e corresponde ao AFD extraído da matriz $D_{i,j}$. Para fins de simplificação, os elementos $d_{i,j}$ da Equação 18 não serão aqueles referentes a matriz D, mas sim ao elemento $(1, 1)$ relativo à partição (i, j) da matriz D. Reescrevendo a Equação 17 obtém-se:

$$\begin{aligned}
\mathrm{MSATD}_{l=1} = \frac{(2^{n-l})^2}{2^k} (& |(d_{1,1} + d_{2,1} + d_{1,2} + d_{2,2})| + |(d_{1,1} + d_{2,1} - d_{1,2} - d_{2,2})| \\
& + |(d_{1,1} - d_{2,1} + d_{1,2} - d_{2,2})| + |(d_{1,1} - d_{2,1} - d_{1,2} + d_{2,2})|)
\end{aligned} \tag{18}$$

A MSATD_1 é semelhante ao cálculo da $SATD_{2\times2}$ e isso permite que os elementos que compõem as partições do cálculo da MSATD_1 sejam escritas na forma de uma transformada Hadamard, como mostra a Equação 19. Para fins de simplificação, a matriz formada pelos elementos na posição $(1, 1)$ relativa a cada partição será chamada de

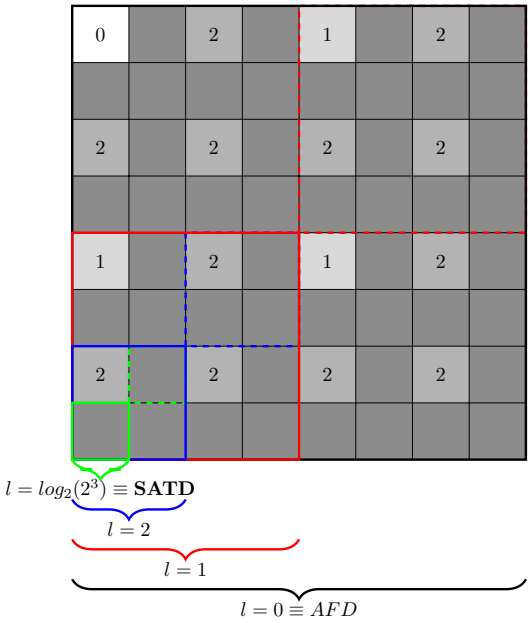

Figura 2. Níveis de particionamento de um bloco 8×8 no algoritmo MSEA-SATD. As posições marcadas com valores menores ou iguais a l compõem a matriz FD_M usada na métrica MSATD_l.

FD_M e as constantes do somatório serão colocadas em evidência. Apesar de ser apresentada apenas para o nível um de particionamento, a Equação 19 é genérica o suficiente para ser aplicada para n e l quaisquer.

$$\mathrm{MSATD}_l = \frac{(2^{n-l})^2}{2^{n-1}} \sum_{i=1}^{2^l} \sum_{j=1}^{2^l} \left| \left(H_{2^l \times 2^l} \times FD\_M_{2^l \times 2^l} \times H_{2^l \times 2^l} \right)_{i,j} \right| \tag{19}$$

A Figura 2 ilustra como é feito o particionamento e seleção de elementos para a transformação em cada nível É importante observar que a computação de MSATD_l tende a se tornar mais precisa e tão complexa quanto a SATD à medida que l aumenta. Entretanto, mesmo computando diretamente o último nível do critério de eliminação, ainda são usadas apenas 25% de operações em comparação com o cálculo completo da SATD.

4. Resultados

As técnicas propostas foram implementadas no código de referência do padrão de Codificação de Vídeo de Alta Eficiência - *High Efficiency Video Coding* (HEVC) [Sullivan et al. 2012, ITU-T 2013]. Os testes foram feitos com base na Condições Comuns de Teste - *Common Test Conditions* (CTC) [Bossen 2012], usando o arquivo de configuração "*Low-delay P-High efficiency*". Exceto por dois vídeos com 10 bits por pixel, todas as outras 22 sequências de vídeo foram executadas usando os quatro QPs indicados (22, 27, 32, 37). Neste artigo, serão apresentados apenas os resultados de pior caso, que ocorreram com o uso do QP 22.

Foram feitos experimentos nas duas etapas da ME. A primeira etapa, chamada de Estimação de Movimento Inteira - *Integer Motion Estimation* (IME), consiste numa busca pelos candidatos de um quadro já codificado. A segunda etapa, chamada de Estimação de Movimento Fracionária - *Fractional Motion Estimation* (FME), consiste em um refinamento onde são gerados e avaliados novos candidatos gerados a partir do candidato selecionado na IME.

4.1. Estimação de Movimento Fracionária

A Figura 4.1 apresenta o percentual acumulado de eliminações em cada nível do algoritmo MSEA-SATD para cada QP. Os gráficos avaliam apenas candidatos que não tiveram sua transformada completa calculada. Aqueles compostos por blocos 4×4 são considerados até o nível 1 enquanto as eliminações no nível 2 consideram apenas candidatos compostos por blocos 8×8.

É possível perceber que, em QPs mais baixos, a técnica AFD (Nível 0) apresenta um baixo percentual de eliminações. À medida que o QP aumenta, a técnica de eliminação se torna mais efetiva e também mais imprevisível, uma vez que o percentual de eliminações varia para cada vídeo avaliado.

O método MSEA-SATD apresenta um bom resultado no último nível de eliminação. No contexto da FME, onde os candidatos possuem valores muito próximos, o último nível 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 em longas sequências de quadros. Pelos resultados obtidos, há um indício de que a métrica AFD é uma técnica bastante efetiva para esse nicho de aplicação.

4.2. Estimação de Movimento Inteira

Na IME, diferente do resultado observado na FME, o uso da AFD (nível 0) apresenta uma boa taxa de eliminação. Isso ocorre pois 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 uma métrica que usa apenas três operações aritméticas.

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

5. Conclusão

Neste trabalho foram apresentadas duas técnicas de eliminação de candidatos na ME que faz uso da métricas SATD na busca pelo bloco de referência. A primeira técnica, inspirada no SEA, deu origem ao critério de eliminação chamado de AFD, que possui cálculo bastante simples e se provou eficiente na IME, com aproximadamente 23% de eliminações no pior cenário.

Foi demonstrado que a técnica usada no MSEA também pode ser aplicada a métrica SATD. Isso resultou na definição do critério de eliminação em níveis MSATD_l.

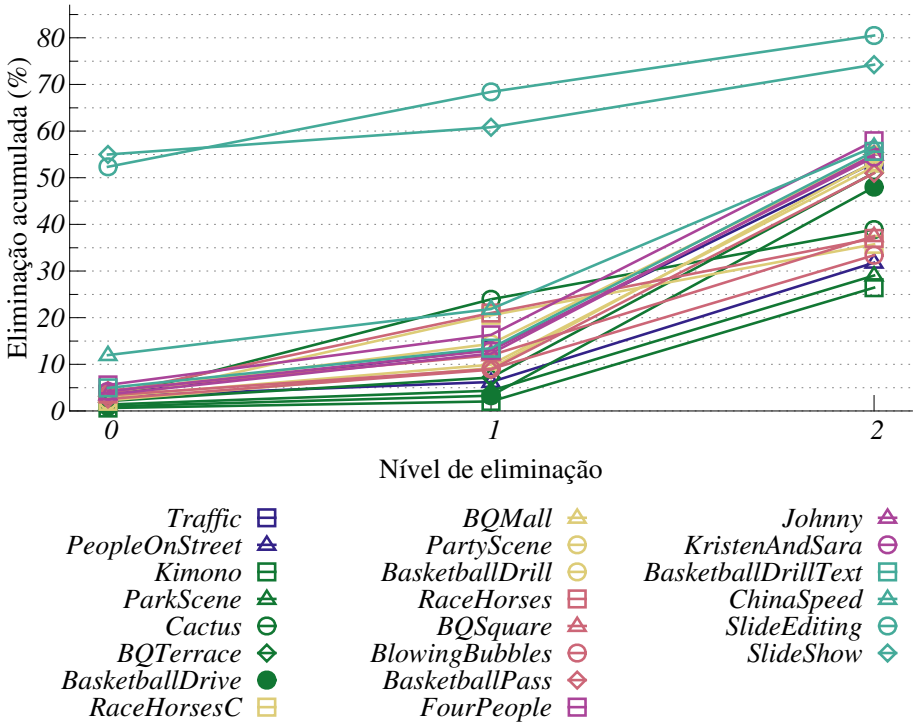

Figura 3. Percentual acumulado de eliminações de candidatos em cada nível na FME usando o QP 22.

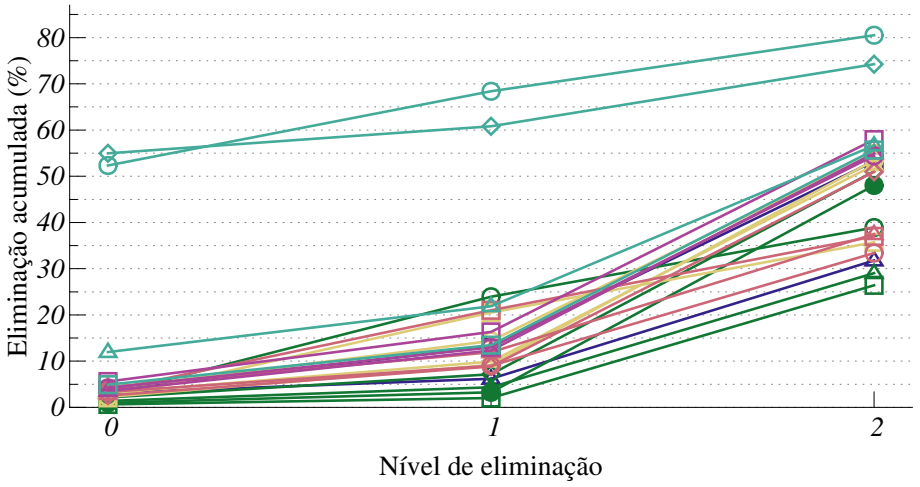

Figura 4. Percentual acumulado de eliminações de candidatos em cada nível na IME usando o QP 22.

Esse critério demonstrou ser bastante eficiente, principalmente no terceiro nível. Ele eliminou, no pior caso, aproximadamente 24% e 69% de candidatos na FME e IME, respectivamente. Além de apresentar uma boa taxa de eliminações no terceiro nível, ele computa apenas 25% das operações necessárias para calcular a SATD.

Referências

Agostini, L. V. (2007). Desenvolvimento de arquiteturas de alto desempenho dedicadas à compressão de vídeo segundo o padrão h.264/avc. Tese de doutorado., UFRJ.

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. Circuits Syst. Video Technol.*, 22(12):1685–1696.

Cancellier, L. H., Bräscher, A. B., Seidel, Ismael, G. J. L., and Agostini, L. (2015). Exploring optimized hadamard methods to design energy-efficient satd architectures. In *Journal of Integrated Circuits and Systems*, JICS, pages 113–112. SBC.

Dominges, Jr, J. S., Possani, V. N., Silveira, D. S., da Rosa Jr, L. S., and Agostini, L. V. (2011). High throughput 4x4 and 8x8 SATD similarity criteria architectures for video coding applications. In *2011 VII Designer Forum (DF)*, page 115. Citeseer.

Gao, X., Duanmu, C., and Zou, C. (2000). A multilevel successive elimination algorithm for block matching motion estimation. *Image Processing, IEEE Transactions on*, 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, Genebra.

Li, W. and Salari, E. (1995). Successive elimination algorithm for motion estimation. *IEEE Trans. on Image Processing*, 4(1):105–107.

Richardson, I. E. G. (2002). *Video codec design: developing image and video compression systems*. John Wiley and Sons.

Richardson, I. E. G. (2003). *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons Inc.

Rowland, T. (2006). Block matrix.

Shi, Y. and Sun, H. (1999). *Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards*. Image Processing Series. Taylor & Francis.

Sullivan, G., Ohm, J., Han, W.-J., and Wiegand, T. (2012). Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668.

Weisstein, E. W. (2009). Kronecker product.

Zhu, C. and Xiong, B. (2009). Transform-exempted calculation of sum of absolute hadamard transformed differences. *IEEE Trans. Circuits Syst. Video Technol.*, 19(8):1183–1188.