Video Compression Based On Shots Reordering

G. R. Paiva, and L. E. Silva

***Abstract*— Over time, the video size has been increasing due to the substantial advancement of technologies related to video quality. For this reason, more video compression techniques have become necessary. This article presents a new method focusing on movies compression, where it is common to have similar shots interspersed by other shots in scenes. Taking advantage of compression technologies that use frames types Intra (I), Predictive (P) and Bi-predictive (B), this method reorders the shots in order to join similar shots reducing the need for I frames and improving utilization of frames P and B. For this, is proposed a method of automatic reordering of shots where they are identified and reordered by comparing the sum of the absolute difference of the sum of the values of the colors of each pixel between frames to detect if they are similar. With this, similar shots are kept in sequence and then the video is compressed. The proposed method presents a video file size reduction of up to 21.82% according to Full HD scenes testing using H.265/HEVC encoder.**

***Keywords*— Video compression, shots, reordering, movie, sum of absolute difference.**

I. INTRODUÇÃO

A QUALIDADE dos vídeos evoluiu rapidamente e junto com ela o espaço necessário para seu armazenamento também veio a aumentar, assim exigindo que as técnicas e tecnologias de compressão se adaptem para atender o mercado.

Imagens digitais consistem de um conjunto ordenado de pixels. A junção de várias imagens digitais exibidas rapidamente de forma continua constituem um vídeo. Essas imagens são denominadas frames quando em um vídeo. Um conjunto de frames com poucas diferenças entre si como pequenos movimentos de câmera ou de objetos ou com movimentos contínuos, é chamado de shot. A compressão de vídeo consiste em reduzir o espaço ocupado pelo vídeo ao diminuindo a redundância de informação entre os seus frames. Para comprimir um vídeo primeiramente deve-se categorizar os frames como Intra codificados ('I'), ou Inter codificados, que consiste de frames de Predição ('P') e frames Bidirecionais ('B'). Frames I são imagens chave para a compressão dos frames Inter codificados. Os frames P são gerados através de predição a partir de um frame anterior, sendo eles tanto frames I como frames P, capturando assim apenas as diferenças entre os dois frames. Frames B são gerados através da interpolação do frame anterior e posterior sendo eles tanto frames I como frames P. A escolha da quantidade de frames em cada categoria define o nível de compressão de um vídeo. Quanto mais frames I menor será a compressão [1].

A predição feita ao comprimir um vídeo consiste de basicamente dois processos, sendo eles estimativa de movimento e compensação de movimento. Para realizar a predição, cada frame é subdividido em vários blocos de pixels. Logo para cada bloco é verificado se existe um similar no frame posterior, averiguando se aquele bloco está na mesma posição ou em outra posição ao longo do frame. Esse procedimento é chamado de estimativa de movimento. A compensação de movimento consiste em mapear o deslocamento de cada bloco de pixels entre frames em vetores de deslocamento, e com isso evitar o reaparecimento de blocos semelhantes em frames próximos, culminando na redução do tamanho do arquivo.

Este artigo é composto de cinco seções. A seção II apresenta o referencial teórico. A seção III explica o método proposto. A seção IV expõe os resultados das experimentações assim como suas discussões. A seção V apresenta a conclusão do artigo.

II. REFERENCIAL TEÓRICO

Ao longo do tempo, vários métodos que visam melhorar a compressão de vídeo foram desenvolvidos pensando em casos específicos ou em novas abordagens, como a compensação de zoom e movimento horizontal da câmera [2]. A compressão baseada em segmentação, utiliza textura a partir do conteúdo de cenas, assim as partes consideradas como textura não são codificadas durante a compressão [3]. A compressão ciente de saliências em codificação baseada em área de interesse (*region-of-interest* (ROI)), consiste de reduzir artefatos de codificação saliente em partes fora da ROI do frame para manter a atenção do usuário na área de interesse [4].

Também foram desenvolvidos métodos voltados para nichos específicos de vídeos, como o algoritmo hibrido para imagens e vídeos biomédicos, onde é feito a transformada discreta do cosseno na transformada discreta de ondulação (*discrete wavelet transform),* preservando informações críticas da imagem ou vídeo [5]. A compressão de vídeos 3D baseados em *Hight Efficiency Video Coding* (HEVC), propõe um esquema de compressão de múltipla visualização que utiliza a ferramenta de codificação de visualização única do HEVC [6].

Também há o uso de características faciais para compressão adaptativa de vídeos para dispositivos mobile, como apresentado por Banerjee [7]. O método possui o foco em comprimir vídeos que possuem rostos humanos. Para isso são armazenados apenas os frames que apresentam características faciais distintas, assim os frames não armazenados são convertidos em meta-dados para a futura reconstrução em uma eventual descompressão.

Uma outra abordagem da compressão de vídeo é tomada para comprimir o banco de dados de simulação de oceano, que

G. R. Paiva, Universidade Federal de Alfenas, Alfenas, Minas Gerais, Brasil, gabriel.ribeiro.paiva@hotmail.com.

L. E. Silva, Universidade Federal de Alfenas, Alfenas, Minas Gerais, Brasil, luizedsilva@gmail.com.

suas imagens chegam a ocupar centenas de gigabytes. Utilizar compressão de vídeo para reduzir o espaço de armazenamento se mostra uma ótima solução como mostrado por Berres [8], podendo reduzir seu tamanho de 2 a 4 ordens de magnitude.

Trabalhos propondo otimização de métodos também são frequentes como o algoritmo de rápida decisão de modo baseado em gradiente para predição Intra em HEVC, onde direções de gradiente são calculadas e é gerado um histograma do modo gradiente para cada unidade de codificação. Assim, baseando-se na distribuição dos histogramas gerados é escolhido apenas uma pequena quantidade dos 35 modos de intra codificação propostos no HEVC para utilização nos processos de *rough mode decision* (RMD) e *rate-distortion optimization* (RDO) feitos pelo HEVC [9].

O conceito da detecção de transição de shot é algo já bem consolidado, onde se procura padrões em elementos de um frame com um outro. Alguns desses elementos são as cores, o histograma das cores, as bordas dos objetos ou a combinação desses elementos para se obter uma maior precisão [10].

Com o aumento da qualidade de vídeo, técnicas antigas precisaram ser aprimoradas, assim como novas técnicas vieram a ser necessárias. Com isso em mente, o formato de compressão H.265/HEVC veio a ser implementado baseado no seu predecessor, o H.264/ *Advanced Video Coding* (AVC), trazendo melhor compressão e suporte para novas tecnologias de vídeo [11].

III. MÉTODO PROPOSTO

Para obter uma melhor compressão é proposto reordenar o vídeo de forma que shots parecidas sejam comprimidas em sequência como ilustrado na Fig. 1. Com isso diminui-se a necessidade de frames I além de se obter uma melhor compressão em frame P e B por seus blocos serem transmitidos por mais frames através do vetor de deslocamento. Para isso deve-se identificar os shots, reordena-los e comprimi-los.

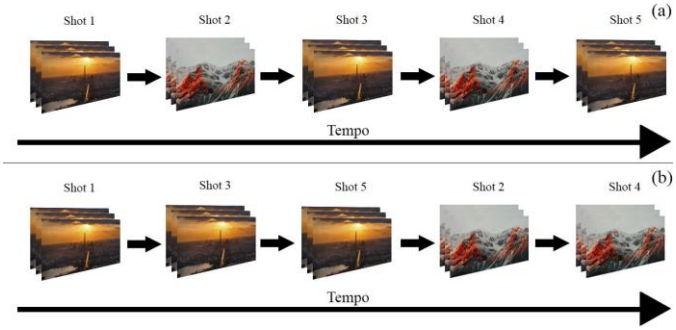


Figura 1. Exemplo de reordenação onde os shots 1, 3 e 5 são similares assim como os shots 2 e 4. (a) representa o vídeo antes da reordenação e (b) representa depois da reordenação.

A: Detecção de transição de shot

O método proposto utiliza a soma da diferença absoluta (SDA) devido a sua simplicidade computacional, pois a operação é muito utilizada.

Para detectar a transição de shots, faz-se a soma dos valores de cada cor de um pixel (vermelho, verde e azul), com cada cor correspondendo a um número de 0 a 255, e subtrai da soma dos valores de cada cor de um pixel na mesma posição no frame posterior, então retira-se o valor absoluto desta subtração. Essa operação se repete somando os resultados gerados até o valor de todos os pixels do frame serem calculados. Por fim, o resultado é divido pela quantidade de pixels multiplicada por 765 (soma dos valores máximo de cada cor), transformando o resultado em uma importância de 0 a 1, assim como descrito na equação (1).

$$s = \left(\sum_{n \le k}^{n} \left| \left(r'_n + g'_n + b'_n \right) - \left(r_n + g_n + b_n \right) \right| \right) / \left(k \times 765 \right) \quad (1)$$

Onde ‘k’ é o número de pixels por frame, ‘r’, ‘g’ e ‘b’ e ‘r’’, ‘g’’ e ‘b’’ são os valores das cores vermelho verde e azul de frames vizinhos respectivamente. Assim, obtém-se um valor ‘s’ para cada par de frames consecutivos.

Logo define-se um valor ‘t’ correspondente a um limite, onde se ‘s’ for maior que ‘t’, então os dois frames pertencem a shots diferentes.

Esse procedimento gera um resultado altamente satisfatório identificando mudanças de shots do tipo corte (onde um frame pertence ao um shot e o frame seguinte pertence a outro shot) com precisão.

B: Reordenação

Para reordenar as cenas é gerado duas listas, uma com o frame inicial e outra com o frame final de cada shot.

Inicialmente adiciona-se na lista de frames reordenados o número do primeiro e último frame do primeiro shot, e ao mesmo tempo os retira das listas de frame inicial e final de cada shot. Em seguida compara-se, utilizando o mesmo método proposta para a detecção de transição de shot, o ultimo frame do primeiro shot com os frames iniciais dos shots seguintes, até se obter um valor de ‘s’ menor que o ‘t’ definido. Deste modo, adiciona-se o frame inicial e final do shot à lista de frames reordenados e passa a usar o frame final da última cena comparada para comparar as cenas seguintes.

Esse processo ocorre até não haver mais elementos para se comparar na lista de frames iniciais, como ilustrado na Fig. 2. Assim, retira-se os frames iniciais e final desses shots das listas de frames iniciais e finais de shot respectivamente e reinicia o processo até as listas de frames iniciais e finais estarem vazias.

C: Corte e concatenação dos shots

Recorta-se do vídeo os shots onde o primeiro shot inicia no frame inicial 1 e termina do frame final 1 e assim sucessivamente até recortar todos os shots do vídeo e assim os concatena de acordo com a ordem da lista de shots reordenadas. Em seguida o vídeo é comprimido retirando o limite de intervalo de frames I e a obrigatoriedade de eles existirem no início das cenas, com o intuito de reduzir ao máximo a quantidade de frames I no vídeo e, portanto, melhorar a compressão dos frames Inter codificados.

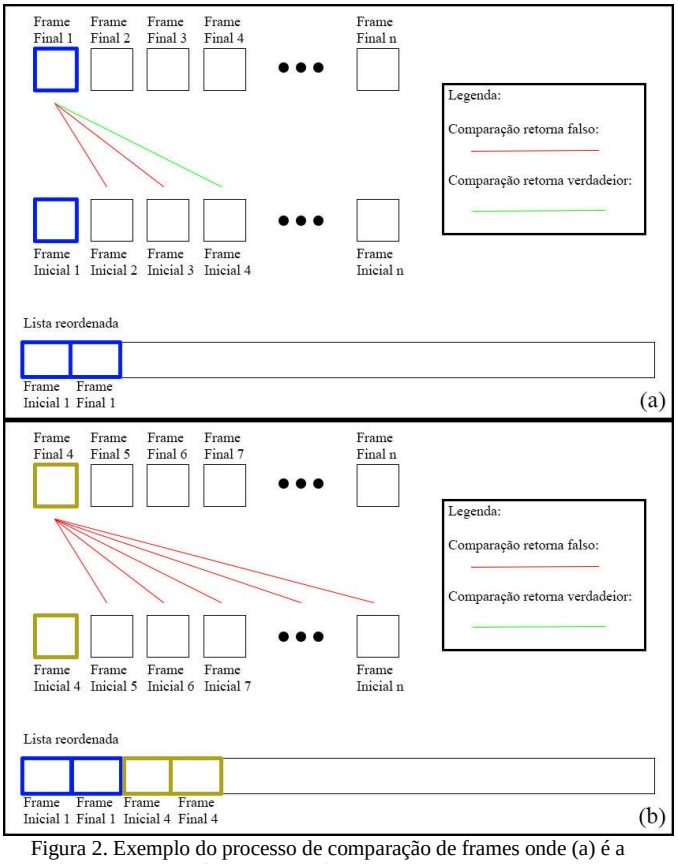


Figura 2. Exemplo do processo de comparação de frames onde (a) é a primeira iteração e (b) é a segunda e última iteração.

IV. EXPERIMENTAÇÃO E DISCUSSÃO

Para verificar a eficiência e o ganho de espaço do método proposto foram utilizadas quatro cenas conforme a Tabela I. As cenas foram separadas em duas categorias sendo elas: diálogo, que consiste de shots com câmera estática ou com pouco movimento, contendo as cenas *Split* e *Hatefull*; e ação, onde os shots possuem muita movimentação de objetos e câmera, abrangendo as cenas *John* e *Assassins*.

TABELA I
INFORMAÇÕES DAS CENAS

Cenas	Duração	Número de frames	Frames por segundo	Tamanho
Split	04:03.58	5840	23.98	1920x800
Hatefull	08:21.04	12013	23.98	1920x704
Assassins	05:14.06	7530	23.98	1920x800
John	07:37.50	10969	23.98	1920x800

Como o foco do método é a compressão do vídeo, todos os exemplos estão com suas faixas de áudio removidas para não haver interferências no resultado.

Todos as compressões foram feitas utilizando a implementação do padrão de compressão H.265/HEVC, x265 [12].

Uma das métricas usada nas experimentações é o *Constant Rate Factor* (CRF), sendo uma variável que representa o padrão de qualidade do vídeo após a compressão com *bitrate* variável presente na implementação x265, onde seu valor está entre 0 e 51, sendo 0 sem perda de dados e 51 com alta perda de dados.

A métrica de redução de tamanho descrito pela equação (2) utiliza o vídeo reordenado comprimido pelos mesmos parâmetros do seu divisor, sendo um deles o CRF.

$$\left[1-\left(\frac{tamanho\ do\ vídeo\ reordenado\ (bytes)}{tamanho\ do\ vídeo\ (bytes)}\right)\right]\times 100 \qquad (2)$$

Na Tabela II são comparados os tamanhos dos arquivos das cenas reordenadas e comprimidas com CRF 20, que representa uma boa fidelidade visual ao vídeo original. Pode-se observar que o método apresenta uma considerável redução do tamanho do arquivo de video em todas as cenas.

TABELA II
COMPARAÇÃO DAS CENAS ANTES E PÓS REORDENAÇÃO

Cenas	Tamanho da cena reordenada (bytes)	Tamanho da cena não reordenada (bytes)	Redução de tamanho
Split	29.182.815	37.328.997	21,82%
Hatefull	54.018.518	64.633.820	16,42%
Assassins	111.193.842	127.505.845	12,79%
John	109.735.154	123.337.410	11,02%

Para verificar se a qualidade da compressão influencia no ganho de espaço após a reordenação, as cenas foram comprimidas com parâmetro CRF 10, 20 e 30 respectivamente e foi calculado sua redução de tamanho como mostrado na Tabela III. Quando comprimido com maior perda de dados (CRF 30), a reordenação deixa de ser tão eficiente, pois os quadros passam a ser mais facilmente comprimidos, de tal modo a compressão sem reordenação também tem um bom resultado. Quando comprimido com pouca redução de dados (CRF 10), os quadros passam a ser pouco comprimidos, causando a perda de eficiência do método.

TABELA III
COMPARAÇÃO DAS CENAS EM RELAÇÃO AO CRF

Cenas	Melhoria com CRF 10	Melhoria com CRF 20	Melhoria com CRF 30
Split	6,23%	21,82%	7,73%
Hatefull	14,15%	16,42%	6,24%
Assassins	9,14%	12,79%	9,06%
John	5,57%	11,02%	5,22%

Com o intuito de averiguar a redução de tamanho causada exclusivamente pela reordenação, sem levar em consideração o impacto do aumento do intervalo limite de frames I, foi feita a compressão das cenas sem o limite de intervalo, assim como acontece com as cenas reordenas com CRF 20, cujo resultado representado na Tabela IV demonstra que a reordenação representa pelo menos 75% do ganho na compressão das cenas experimentadas.

Pode-se constatar a partir da Tabela V, que utiliza os vídeos comprimidos com CRF 20, que cenas de diálogo apresentam uma maior redução de espaço ocupado do que cenas de ação. Isso ocorre pois em cenas de diálogo, os shots intercalados são

geralmente muito parecidos, assim resultando em uma melhor compressão de frames P e B, já em cenas de ação os shots intercalados em sua maioria possuem muita movimentação, o que causa dificuldades em melhor comprimir os frames Inter codificados.

TABELA IV
REDUÇÃO OBTIDA PELA REORDENAÇÃO

Cenas	Tamanho da cena reordenada (bytes)	Tamanho da cena não reordenada (bytes)	Redução de tamanho
Split	29.182.815	34.907.713	16,40%
Hatefull	54.018.518	62.877.663	14,09%
Assassins	111.193.842	123.953.863	10,29%
John	109.735.154	121.126.221	9,40%

TABELA V
COMPARAÇÃO ENTRE CATEGORIAS

Categoria	Redução média de tamanho
Diálogo	19,12%
Ação	11,9%

A Tabela VI, cujo os vídeos foram comprimidos com CRF 20, demonstra que a eficiência do método está majoritariamente ligada ao desempenho da compressão dos frames Inter codificados e não na quantidade de frames P ou B, que representam juntos um aumento de menos de 3% do total de frames por cena.

TABELA VI
AUMENTO DE FRAMES P E B

Cenas	Número frames P	Número frames P (reordenado)	Número frames B	Número frames B (reordenado)
Split	1498	1502	4283	4337
Hatefull	2994	2984	8923	9028
Assassins	2850	2938	4465	4591
John	3623	3721	7152	7247

V. CONCLUSÃO

Neste artigo foi proposto um novo método de compressão de vídeo, com foco em filmes. A partir das experimentações, foi constatado que a reordenação do vídeo para aproximar shots similares possui uma boa eficácia na redução do tamanho do arquivo de video. A redução está ligada a qualidade da compressão, onde deve-se ter um equilíbrio na redução de dados para extrair melhor resultado do método e ligada a quantidade de movimentos dos shots, onde quanto menos movimento melhor a compressão. Também se conclui que a redução de tamanho ocorre em sua maior parte em razão da reordenação onde se obtém um maior desempenho de compressão dos frames Inter codificados e não à diminuição de frames I diretamente.

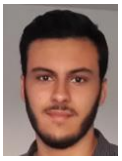
Este trabalho foi focado na compressão, assim sendo necessário novas abordagens na descompressão para o seu uso em aplicações que exigem exibição em tempo real, como o *streaming*.

AGRADECIMENTOS

Agradeço ao professor Dr. Luiz Eduardo da Silva, da Universidade Federal de Alfenas, pelos conhecimentos e competências que facilitaram a pesquisa e desenvolvimento do trabalho.

REFERÊNCIAS

[1] D. Mitrovic, "Video Compression," University of Edinburgh, 2012.
[2] Y. T. Tse, and R. L. Baker, "Global zoom/pan estimation and compensation for video compression," *Acoustics, Speech, and Signal Processing International Conference on* IEEE, pp. 2725-2728, 1991.
[3] M. Bosch, F. Zhu, and E. J. Delp, "Segmentation-based video compression using texture and motion models," IEEE *Journal of Selected Topics in Signal Processing*, vol. 5, pp. 1366-1377, Nov. 2011.
[4] H. Hadizadeh, and I. V. Bajic, "Saliency-Aware Video Compression," IEEE *Transactions on Image Processing*, vol. 23, no. 1, pp. 19-33, Jan. 2014.
[5] S. Shrestha, and K. Wahid, "Hybrid DWT-DCT Algorithm for Biomedical Image and Video Compression Applications," *10th International Conference on Information Science, Signal Processing and their Applications on* IEEE, pp. 280-283, 2010.
[6] G. V. Wallendael, S. V. Leuven, J. D. Cock, F. Bruls, and R. V. Walle, "3D Video Compression Based on High Efficiency Video Coding," IEEE *Transactions on Consumer Electronics*, vol. 58, no. 1, pp. 137-145, Mar. 2012.
[7] R. Banerjee, G. Chikara, V. Naik, A. V. Subramanyam, and K. Dey, "Use of Facial Landmarks for Adaptive Compression of Videos on Mobile Devices," *Communication Systems & Networks* (COMSNETS), *10th International Conference on*. IEEE, pp. 320-327, 2018.
[8] A. S. Berres, T. L. Turton, M. Petersen, D. H. Rogers, and J. P. Ahrens, "Video Compression for Ocean Simulation Image Databases," *Workshop on Visualisation in Environmental Sciences* (EnvirVis), 2017.
[9] W. Jiang, H. Ma, and Y. Chen, "Gradient Based Fast Mode Decision Algorithm for Intra Prediction in HEVC," *Consumer Electronics, Communications and Networks* (CECNet), *2nd International Conference on* IEEE, pp. 1836-1840, 2012.
[10] C. Cotsaces, N. Nikolaidis, and I. Pitas, "Video shot boundary detection and condensed representation: a review," IEEE *signal processing magazine*, vol. 23, no. 2, pp. 28-37, 2006.
[11] V. Sze, M. Budagavi, and G. J. Sullivan, "High Efficiency Video Coding (HEVC): Algorithms and Architectures", Springer, 2014.
[12] MulticoreWave Inc., x265 Encoder. Avaliable in: <http://x265.org/>. Access Date: 17/07/2018. 2018.



Gabriel Ribeiro Paiva cursa Ciência da Computação na Universidade Federal de Alfenas, Alfenas-MG, Brasil. Possui experiência em desenvolvimento de software, banco de dados, aplicações web, desktop e mobile e tem interesse no tema de inteligência artificial.