

DESENVOLVIMENTO DE ALGORITMO PARA A COMPRESSÃO DE QUADROS DE REFERÊNCIA EM CODIFICADORES DE VÍDEOS DIGITAIS

SILVEIRA, Dieison¹; AGOSTINI, Luciano²

¹ UFPel, Curso de Ciência da Computação.
Email: dssilveira@inf.ufpel.edu.br

² UFPel, Centro de Desenvolvimento Tecnológico.
Email: agostini@inf.ufpel.edu.br

1 INTRODUÇÃO

A codificação de vídeos é de extrema importância nos dias atuais, por conta da grande quantidade de informação presente em vídeos de alta resolução e a quantidade de aparelhos eletrônicos que manipulam este tipo de mídia, como *smartphones*, *tablets*, celulares, televisores digitais, câmeras, filmadoras digitais portáteis, entre muitos outros.

Os vídeos necessitam de uma grande quantidade de dados para serem representados, mas eles possuem uma importante característica: apresentam elevado grau de redundância (RICHARDSON, 2002). Isto significa que grande parte da enorme quantidade de dados necessários na representação do vídeo é dispensável. O objetivo da compressão de vídeo é a máxima eliminação possível desses dados desnecessários, para conseguir representar o vídeo digital com um número de bits muito menor do que o original. No entanto, o grau de complexidade do processo que leva à compressão de um vídeo faz com que o desenvolvimento de arquiteturas em hardware para esse fim seja uma tarefa igualmente complexa e desafiadora.

O padrão de codificação de vídeo H.264/AVC é um padrão consolidado no mercado e foi adotado pelo Sistema Brasileiro de TV Digital, que hoje está em quase todas as regiões do país. O padrão H.264/AVC aumentou significativamente a taxa de compressão de vídeo, para uma mesma qualidade do vídeo, quando comparado a padrões anteriores. Isso se deve ao fato do padrão introduzir o uso de blocos de tamanhos variáveis, predição intra-quadro realizada no domínio espacial, utilização de múltiplos quadros de referência para a predição inter-quadros, entre outros (AGOSTINI, 2007).

O codificador H.264/AVC, quando implementado em hardware, utiliza quantidades expressivas de acessos à memória externa para realizar a codificação. Sendo assim, o projeto do codificador deve considerar que um dos maiores gargalos deste tipo de sistema está nesta comunicação entre a memória externa e as unidades de processamento (AGOSTINI, 2007). Isso ocorre porque os quadros codificados são armazenados em memória externa para serem utilizados como referência na codificação. A comunicação com a memória é um gargalo porque, além da memória ser mais lenta do que as unidades de processamento, os acessos à memória também demandam um elevado consumo de energia.

Muitas soluções que visam reduzir, os problemas que existem na comunicação entre memória externa e o núcleo de processamento dos codificadores de vídeo podem ser encontradas na literatura. Tais soluções são baseadas em duas abordagens principais: (1) redução de largura de banda de memória externa, através de estratégias de reutilização de dados usando *caches*; e (2) redução de largura de

banda através da compressão de quadros de referência antes de serem armazenados na memória externa. A principal vantagem da compressão de quadros de referência, quando comparado com a outra abordagem, é que a reutilização de dados reduz apenas o número de acessos de leitura, enquanto que a compressão de quadros de referência reduz o número de acessos de leitura e de escrita.

Este trabalho apresenta uma solução, com base na compressão de quadros de referência, que é capaz de reduzir, em média, mais de 30% do volume de dados que é armazenado na memória externa, conseqüentemente, diminuindo o número de operações de leitura e escrita necessárias para acessar esses dados. O processo de compressão é realizado sem qualquer perda de informação, evitando assim a degradação da qualidade do vídeo. Esta é uma vantagem deste trabalho em relação a outros trabalhos da literatura.

2 METODOLOGIA (MATERIAL E MÉTODOS)

O algoritmo proposto neste trabalho é uma adaptação do algoritmo de Huffman (SALOMON, 2007). O algoritmo de Huffman é uma técnica de compressão de dados bastante conhecida, a técnica baseia-se em métodos estatísticos e é amplamente utilizada em compressores de imagem e texto (SALOMON, 2007).

Tradicionalmente, Huffman é implementado usando árvores de dados, que são criadas de acordo com regras específicas. Através destas regras, os padrões de dados que ocorrem com mais frequência são codificados com uma quantidade menor de bits. Para que se consiga fazer isso, o algoritmo requer uma análise de duas passagens: na primeira passagem, os cálculos estatísticos são executados; no segundo, o algoritmo constrói uma árvore com base nos resultados obtidos no passo anterior. Este tipo de aplicação não é adequado para um codificador de vídeo em hardware, onde desempenho é uma restrição crítica. Portanto, as implementações de hardware de um codificador de Huffman são normalmente adaptadas usando uma tabela estática de Huffman, assim, a etapa de análise estatística não é realizada de forma dinâmica, sendo realizada de forma estática antes da codificação, considerando valores estatísticos médios para vídeos típicos.

Este trabalho propõe uma solução adaptativa ao contexto, utilizando oito tabelas estáticas de Huffman para comprimir os quadros de referência. Cada tabela está especializada para uma série de características diferentes dos quadros, de modo que uma estratégia de decisão deve ser aplicada.

Para este trabalho, cada tabela contém os códigos para cada valor de amostra de luminância. Apenas amostras de luminância são usadas porque nos padrões de vídeo atuais, apenas estas amostras são usadas como referência e, portanto, armazenadas na memória externa. Uma vez que as amostras de luminância são tipicamente representadas com oito bits por amostra, cada tabela tem 256 entradas, uma para cada valor de luminância possível. Os valores para cada tabela foram obtidos separadamente e correspondem à ocorrência média de cada amostra de luminância, de acordo com uma análise com 10 sequência de vídeos com resolução 1080p (1920x1080 pixels), as sequências utilizadas apresentam movimento e características de iluminação diferentes, deixando a solução mais robusta.

A média das amostras é calculada para cada bloco e, então, verifica-se em qual intervalo esta média se encontra, assim, a partir do intervalo selecionado, é

determinada a tabela de codificação. A Tab. 1 apresenta a relação das oito tabelas com seus respectivos intervalos de valores.

Tabela 1 – Relação entre tabela e intervalo de valores

Tabelas	Intervalo da média	Tabelas	Intervalo da média
Tabela 1	0 – 31	Tabela 5	128 – 159
Tabela 2	32 – 63	Tabela 6	160 – 191
Tabela 3	64 – 95	Tabela 7	192 – 223
Tabela 4	96 – 127	Tabela 8	224 – 255

A Fig. 1 apresenta o fluxograma do algoritmo de compressão. O processo de codificação inicia com a busca de um bloco 4x4 e calcula o valor da média das amostras. Com base na média resultante, a tabela mais adequada é usada para comprimir o bloco. Depois disso, o bloco comprimido é armazenado na memória externa e o mesmo processo é aplicado para os próximos blocos do quadro.

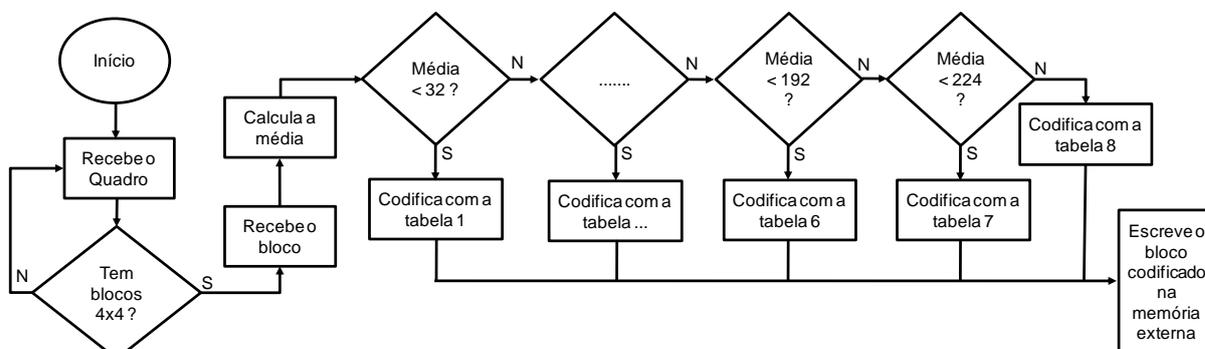


Figura 1 – Fluxograma do processo de codificação utilizando 8 tabelas estáticas

A fase de decodificação consiste em ler o quadro codificado na memória de quadros de referência, aplicando o processo inverso da fase de codificação. O quadro é interpretado como uma lista de palavras de 32 bits. A tradução é realizada utilizando uma estratégia gulosa, ou seja, que consome os bits da lista, lendo uma palavra por vez. Os três primeiros bits da palavra são usados para selecionar a tabela correta. Os bits restantes são processados utilizando esta tabela. Se a sequência de bits é totalmente consumida e a tabela não atingiu o seu fim, a próxima palavra é lida a partir da memória externa e a decodificação continua até que a extremidade da tabela seja atingida.

3 RESULTADOS E DISCUSSÃO

A Tab. 2 apresenta os ganhos obtidos através da técnica desenvolvida. Estes resultados de compressão foram obtidos utilizando oito vídeos de alta definição, com diferentes aspectos de movimento e iluminação. As taxas de compressão alcançadas por cada sequência indicam também a redução de largura de banda de memória externa obtida.

Como está apresentado na Tab. 2, uma variação de taxa de compressão ocorre entre as diferentes sequências de vídeo. Isto é explicado pelo fato de que algumas sequências apresentam quantidade maior de áreas homogêneas, o que favorece a compressão obtida com esta solução.

Tabela 2 – Resultados da média de compressão dos vídeos

Vídeo	Taxa de compressão (%)	Vídeo	Taxa de compressão (%)
Tennis	31,71	Kimono	31,17
Rolling Tomatoes	31,27	In To Trees	30,55
Sunflower	31,30	Station 2	29,20
Tractor	27,50	Pedestrian Area	30,33
Média de compressão		30,38%	

A taxa de compressão média ficou em 30,38%, o desvio padrão ficou em 1,3%, indicando que a compressão dos vídeos atinge boas taxas, independente do vídeo utilizado. Entre estas oito sequências, quatro delas (In To Trees, Tennis, Rolling Tomatoes e Tractor) não foram utilizadas na análise que foi realizada para gerar as tabelas. Isto foi feito de forma intencional, a fim de provar que esta solução atinge boas taxas de compressão para quaisquer sequências de vídeo.

Todos os vídeos utilizados são de alta definição, com resolução de 1920x1080 pixels. O valor médio de compressão foi maior que 30%. Isso significa que o tempo necessário para a estimação de movimento realizar os acessos à memória pode ser reduzido em 30%.

4 CONCLUSÃO

Este trabalho apresentou uma solução para a redução da largura de banda de memória em codificadores de vídeos digitais, através da compressão de quadros de referência. O método apresentado realiza a compressão sem degradação do vídeo, utilizando, para isto, oito tabelas de códigos estáticos e blocos com tamanho de 4x4 amostras. A decisão de qual tabela será usada é simples e baseia-se no valor médio de cada bloco 4x4. A solução atinge uma redução de largura de banda de memória média de mais de 30%. Essa largura de banda é reduzida para operações de leitura e escrita. Este método é inspirado no algoritmo de codificação Huffman e é totalmente compatível com o padrão de codificação H.264/AVC.

Como trabalho futuro, a solução proposta será implementada usando uma linguagem de descrição de hardware, como VHDL. Esta descrição será sintetizada para um FPGA e seu desempenho será avaliado.

5 REFERÊNCIAS

RICHARDSON, I. **Video Codec Design Developing Image and Video Compression Systems**. Wiley, 2002.

SALOMON, David. **Data Compression - The Complete Reference**. London: Springer, 2007.

AGOSTINI, Luciano. **Desenvolvimento de Arquiteturas de Alto Desempenho Dedicadas à Compressão de Vídeo Segundo o Padrão H.264/AVC**. Agosto de 2007. Doutorado em Computação, UFRGS. Porto Alegre, agosto de 2007.