

COMPRESSOR DIFERENCIAL DE QUADROS DE REFERÊNCIA SEM PERDAS PARA CODIFICADORES DE VÍDEO DE ALTA DEFINIÇÃO

GUILHERME POVALA; DIEISON SILVEIRA; LÍVIA AMARAL; BRUNO ZATT;
MARCELO PORTO; LUCIANO AGOSTINI

Grupo de Arquiteturas e Circuitos Integrados – Universidade Federal de Pelotas –
{gpovala,dssilveira,lldamaral,zatt,porto,agostini}@inf.ufpel.edu.br

1. INTRODUÇÃO

Com o aumento do interesse em dispositivos móveis que manipulam vídeos digitais, como *smartphones*, *tablets* e câmeras digitais, a preocupação com o consumo energético vem se tornando um ponto primordial a ser considerado. Esses dispositivos, que funcionam a base de baterias, devem tentar reduzir ao máximo o acesso à memória, uma vez que esses acessos consomem uma alta quantidade de energia. A redução do número de acessos à memória é uma das formas mais eficientes de reduzir o consumo de energia, e conseqüentemente, aumentando o tempo de vida da bateria.

Tendo em vista vídeos digitais em sua forma original, ou seja, a forma como são capturados, sem nenhuma compressão, seria inviável a quantidade de espaço demandado para o seu armazenamento. Portanto, pensando nessa redução de dados, padrões de codificação de vídeos foram desenvolvidos, como o HEVC (*High Efficiency Video Coding*) (ISO/IEC, 2013), padrão mais recente de codificação de vídeos, e o H.264/AVC (ISO/IEC, 2012), padrão atual de mercado e do SBTVD (Sistema Brasileiro de Televisão Digital). Esses padrões definem diversas técnicas, que usadas em conjunto, atingem uma alta taxa de compressão com uma perda em qualidade muitas vezes imperceptível aos olhos humanos.

Uma dessas técnicas definidas pelos padrões de codificação de vídeos é a estimação de movimento (*Motion Estimation* – ME). Essa técnica traz os maiores ganhos em compressão de um codificador de vídeo, porém, é a etapa mais complexa, exigindo uma grande demanda de tempo de processamento e acessos à memória. A ME é responsável por encontrar o melhor casamento entre o bloco atual, que está sendo codificado, com blocos candidatos de quadros previamente codificados (quadros de referência) em uma área de busca (*Search Area* – SA).

Esses acessos realizados na memória externa, principalmente dos quadros de referência, tem um alto consumo energético, que corresponde a cerca de 90% do consumo total da ME, representando assim, um gargalo no sistema como um todo (ZATT, 2011). Para tentar reduzir esse problema, existem duas principais soluções, uma é com o uso de memórias cache, e outra é pela compressão de quadros de referência. A primeira solução reduz apenas as operações de leitura e tem um alto custo de implementação. Já a segunda solução reduz tanto operações de leitura quanto de escrita na memória, com implementações que podem ser simples e de baixo custo computacional. A compressão de quadros de referência pode ser realizada com ou sem perdas de qualidade durante seu processo, porém, ao inserir perdas em qualidade, a qualidade do vídeo como um todo é prejudicada, o que não é desejável.

Levando tudo isso em consideração, este trabalho apresenta uma solução sem perdas e com baixo custo computacional para a redução de acessos à memória em codificadores de vídeos digitais. Essa solução é baseada na compressão de quadros de referência e se chama Compressor Diferencial de Quadros de Referência

(Differential Reference Frame Compressor – DFRC). O algoritmo proposto para essa solução utiliza codificação diferencial e de entropia, e é pensado especialmente para implementação em hardware.

2. METODOLOGIA

Para o desenvolvimento desse trabalho, foi realizado um estudo sobre a distribuição das amostras de vídeos digitais. Com esse estudo, foi possível perceber que a distribuição das amostras de luminância originais entre diferentes vídeos apresenta um comportamento variado, como pode ser visto na Figura 1 (a) e (b), fazendo com que uma codificação eficiente não possa ser aplicada. Porém, foi notado que as amostras desses vídeos tem uma grande redundância espacial, assim, ao ser aplicada uma codificação diferencial, os valores passam a se concentrar em torno de zero, como na Figura (c) e (d). Dessa forma, fazendo com que uma alta taxa de compressão possa ser obtida através de uma codificação de entropia eficiente. Além disso, essa concentração em torno de zero é obtida independente do comportamento da distribuição das amostras de um dado vídeo.

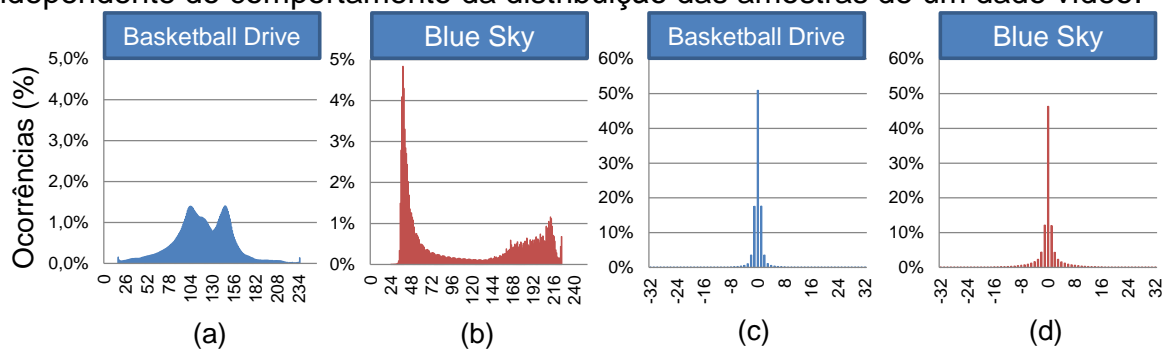


Figura 1 – Análise da Distribuição das Amostras de Luminância Originais (a), (b) e dos Valores Residuais Após Aplicação da Codificação Diferencial (c), (d).

A partir dessas informações, foi desenvolvido um compressor de quadros de referência que leva em conta a distribuição das amostras ao se utilizar uma codificação diferencial. A Figura 2 mostra o fluxo da codificação de um bloco de um quadro do vídeo sendo codificado utilizando o algoritmo proposto.

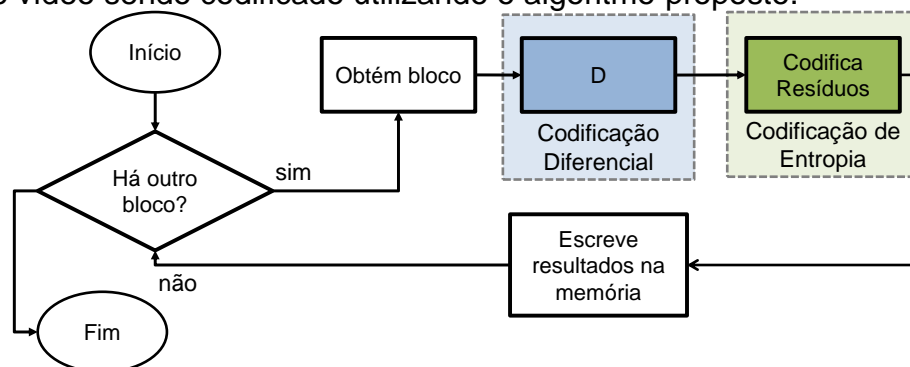


Figura 2 – Codificador do Compressor Diferencial de Quadros de Referência.

Como visto na Figura 2, a compressão é feita bloco a bloco. Neste bloco, é aplicada uma codificação diferencial e uma codificação de entropia. O processo de codificação diferencial é dado pela equação (1):

$$D[i][j] = \begin{cases} B[i][j], & i \wedge j = 0 \\ B[i][j] - B[i-1][j], & i \neq 0 \wedge j = 0 \\ B[i][j] - B[i][j-1], & i \wedge j \neq 0 \end{cases} \quad (1)$$

A equação (1) é responsável por fazer a codificação diferencial de um bloco. As variáveis i e j são, respectivamente, a linha e a coluna do bloco, B é o bloco original e D é o bloco de resíduos gerados. A codificação diferencial é definida para três casos. No primeiro caso, a amostra é copiada para o bloco de resíduos. No segundo caso, o resíduo é calculado pela diferença entre a linha atual e a linha anterior. E o último caso, o resíduo é dado pela diferença entre coluna atual e a coluna anterior.

Os resíduos obtidos na etapa de codificação diferencial são codificados através da codificação de entropia, que nesse trabalho utiliza uma tabela estática de Huffman com códigos previamente calculados para um conjunto de vídeos com diferentes características. Assim, ao passar pela codificação de entropia, temos o bloco codificado e pronto para ser armazenado na memória.

O processo de decodificação de um bloco é realizado de forma inversa ao da codificação. A primeira etapa a ser realizada é decodificar o bloco previamente codificado. A decodificação do bloco de resíduos é realizada da seguinte forma: (i) os primeiros oito *bits* representam a amostra original que não foi codificada; após, (ii) é feita a leitura *bit a bit* onde, caso o primeiro *bit* não esteja na tabela de códigos, o próximo *bit* é concatenado a este e o processo por procurar na tabela continua até que um código válido é encontrado na tabela.

Após o bloco ser decodificado, a codificação diferencial inversa é executada, que é o processo inverso da codificação diferencial e é definida pela equação (2):

$$ID[i][j] = \begin{cases} D[i][j], & i \wedge j = 0 \\ D[i][j] + ID[i-1][j], & i \neq 0 \wedge j = 0 \\ D[i][j] + ID[i][j-1], & i \wedge j \neq 0 \end{cases} \quad (2)$$

A equação (2) é responsável por fazer o processo inverso da equação (1). Desta forma, após ser aplicada a codificação diferencial inversa, o bloco original é obtido, sem apresentar perdas durante o processo.

Além disso, testes foram realizados para decidir qual o melhor tamanho de bloco a ser utilizado pelo algoritmo. Com isso, foi observado que blocos de tamanho maior ou igual a 32x32 amostras, independente do tamanho do bloco escolhido, a taxa de compressão não varia. Assim, para uma melhor futura integração com um sistema de reuso de dados, o tamanho de bloco 32x32 foi escolhido.

Ainda, como os resíduos gerados pela codificação diferencial se concentram em torno de zero, quanto mais distante de zero, menor suas ocorrências, como pode ser visto na Figura 1 (c) e (d). Desta forma, é utilizada uma pequena tabela de Huffman para os resíduos [-32, 32]. Para os resíduos que se encontram fora deste intervalo, é utilizado um código de exceção, seguido do valor da amostra original, mantendo assim, uma codificação sem perdas de informações.

3. RESULTADOS E DISCUSSÃO

Para avaliar o algoritmo proposto, 14 sequências de vídeo (XIPH, 2013), com resolução HD 1080p, foram avaliadas. As sequências de vídeo reconstruídas foram geradas com o software de referência do HEVC, o HM (HM, 2014), considerando o parâmetro de quantização (*Quantization Parameter* – QP) no valor 32. A versão utilizada foi a HM-12.0 com o algoritmo TZ Search.

A Tabela 1 contém os resultados dos experimentos realizados para o algoritmo DRFC. Para esses resultados, apenas o componente de luminância das sequências de vídeo foram utilizadas, visto que, geralmente, apenas esse componente é utilizado pela ME em codificadores de vídeos de trabalho em tempo real.

Tabela 1 – Taxa de Compressão (TC) do Algoritmo DRFC.

Sequência de Vídeo	TC (%)	Sequência de Vídeo	TC (%)
Blue Sky	54.69	Tennis	65.99
In to tree	70.42	Tractor	59.45
Pedestrian area	64.39	BQTerrace	47.06
Sunflower	55.95	Cactus	53.62
Kimono	61.57	Riverbed	57.76
Rolling Tomatoes	76.21	Basketball Drive	61.63
Station2	60.07	Man in car	80.36
Taxa de Compressão Média (%)		62.08	

Como pode ser observado na Tabela 1, a taxa de compressão varia de acordo com a sequência de vídeo. Isso acontece devido ao fato de que vídeos são compostos de forma diferente. Alguns vídeos têm muitas áreas que são compostas por áreas homogêneas, como na sequência *Man In Car*. Isso facilita a codificação, assim, atingindo uma taxa de compressão alta. Porém, como na sequência *BQTerrace*, devido a alta ocorrência de texturas, a taxa de compressão é menor. Além disso, a taxa de compressão obtida para cada sequência de vídeo também indica a redução da largura de banda com comunicação com a memória externa atingida pelo algoritmo proposto.

4. CONCLUSÕES

Este trabalho apresentou uma abordagem para a redução da largura de banda de memória em codificadores de vídeos, utilizando compressão sem perdas dos quadros de referência. O algoritmo se chama Compressor Diferencial de Quadros de Referência (DRFC) e realiza codificação diferencial sobre blocos de tamanho 32x32 para preparar os dados para uma codificação de Huffman estática e eficiente.

Uma pequena tabela estática de códigos Huffman com 65 posições é utilizada para codificar os resíduos provenientes da codificação diferencial, atingindo uma taxa de compressão média superior a 62% para vídeos HD 1080p, apresentando consequentemente, a mesma redução na largura de banda de memória. O DRFC utilizada uma abordagem de fácil implementação em hardware, com baixo custo computacional, especialmente concebida para aplicações em tempo real e em dispositivos móveis, que exigem baixo consumo de energia. Além disso, esta solução é totalmente compatível com os padrões estado da arte de codificação de vídeo, como o H.264/AVC e o HEVC.

5. REFERÊNCIAS BIBLIOGRÁFICAS

HEVC Reference Software (HM). Acessado em Janeiro de 2014. Disponível em: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-12.0/.

ISO/IEC 14496-10: **Coding of audio-visual objects - Part 10: Advanced Video Coding**. Relatório Técnico, 2012.

ISO/IEC 23008-2: **High efficiency coding and media delivery in heterogeneous environments - Part 2: High efficiency video coding**. Relatório Técnico, 2013.

Xiph Test media. Acessado em Janeiro de 2014. Disponível em: <http://media.xiph.org/video/derf/>.

ZATT, B.; SHAFIQUE, M.; SAMPAIO, F.; AGOSTINI, L.; BAMPI, S.; HENKEL, J. Run-time adaptive energy-aware motion and disparity estimation in multiview video coding. **ACM Digital Automation Conference**, New York, p. 1026-1031, 2011.