

Análise da magnitude dos valores salvos no buffer de transposição da transformada DCT-2D do codificador AV1

JELSON STOELBEN RODRIGUES; JONES WILLIAM GÖEBEL; MARCELO SCHIAVON PORTO

Universidade Federal de Pelotas – {jsrodrigues, jwgoebel, porto}@inf.ufpel.edu.br

1. INTRODUÇÃO

Vídeo digital é a principal forma de consumir, armazenar e transmitir vídeo.

Os codecs de vídeo são utilizados para lidar com vídeos digitais. Eles são responsáveis por codificar e decodificar o vídeo, definindo o formato de representação binária do vídeo (bitstream) e realizando a compressão. Isso é necessário porque o vídeo não comprimido utiliza muito espaço. Por exemplo, um vídeo de um segundo com resolução de 3840x2160 pixels (UHD 4K) com taxa de amostragem de 24 quadros por segundo e que utilize 10 bits para representar cada canal de cor (3 canais) utiliza 711MB de espaço. Um total de 41GB por minuto de vídeo.

Existem diferentes codecs de vídeo, como por exemplo o H264, H266/VVC, o AV1. O foco deste trabalho é o codificador AV1.

O AV1 é o formato de codificação introduzido em 2018 pelo consórcio de empresas AOMedia, formado por gigantes da tecnologia como Google, Apple, Microsoft, entre outras, com o objetivo de ser um padrão aberto e livre de royalties. O AV1 apresenta uma eficiência 30% (CHEN et al., 2020) superior que o seu antecessor, o VP9 da Google. O AV1 já está em uso em vários serviços de streaming, como o YouTube, Netflix e Vimeo.

Um codificador tenta reduzir as redundâncias presentes no vídeo para salvar menos informações para representar o vídeo mantendo alta qualidade (RICHARDSON, 2002).

Os codificadores de vídeo atuais funcionam quebrando o quadro que vai ser codificado em blocos menores, geralmente de tamanhos que vão de 4x4 até 128x128 pixels, e para cada bloco vai ser realizada uma predição. Caso esta predição utilize informações de blocos vizinhos neste mesmo quadro, é chamada de predição intra quadro. Caso a predição utilize informações de outros quadros, é chamada de predição inter quadros. O bloco predito vai então ser subtraído do bloco original que está sendo codificado para gerar o resíduo. O resíduo irá passar por uma etapa de transformada. A transformada converte o resíduo do domínio espacial para o domínio das frequências, de modo que a informação será o conjunto de amplitudes e frequências de uma onda cíclica, tipicamente seno ou cosseno, que somadas representam a diferença entre os blocos. O bloco transformado então será quantizado, os coeficientes do bloco serão divididos por um valor, chamado de coeficiente de quantização (CQ). A quantização divide os valores de forma que quando o resultado é fracionário, ele será arredondado para o inteiro mais próximo. Após a quantização os valores serão passados para o codificador de entropia. O codificador de entropia atribui símbolos para os valores de acordo com a frequência que eles aparecem. Valores que aparecem mais, têm símbolos menores e valores que aparecem menos têm símbolos maiores.

A transformada, no contexto de codificação de vídeo, é feita considerando duas dimensões, a largura e a altura. A transformada 2D é implementada utilizando um buffer de transposição (MERT et al., 2017), onde primeiramente é

feita a transformada em uma das duas dimensões e os valores são salvos no buffer e em sequência os valores do buffer são transformados na outra dimensão, a Figura 1 demonstra o processo da transformada DCT 2D.

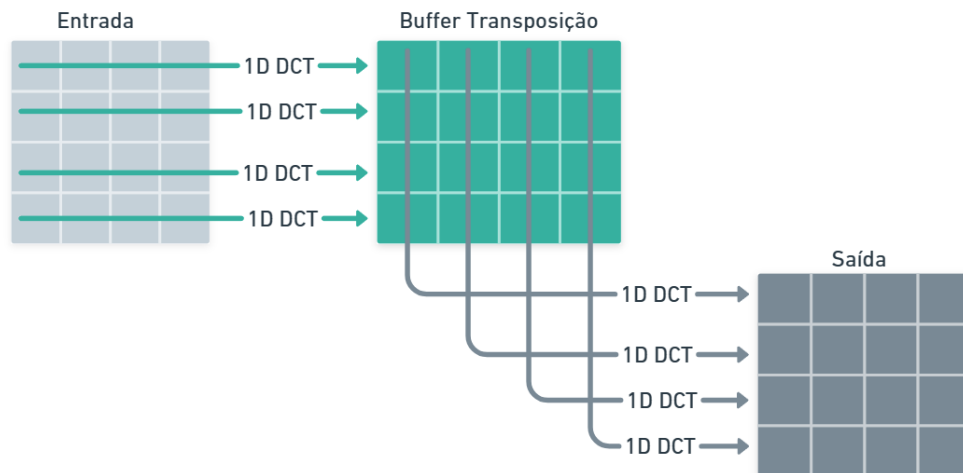


Figura 1 - Transformação de um bloco pela DCT 2D

O AV1 transforma blocos de tamanhos que variam de 4x4 até 64x64, portanto o buffer deve suportar o maior tamanho. Um buffer de 64x64 ocupa uma área considerável de hardware. Reduções no tamanho deste buffer se traduzem em menor consumo de energia em hardware. Portanto, o objetivo deste trabalho é realizar uma análise da magnitude dos valores salvos no buffer de transposição, visando diminuir a área ocupada pelo buffer, sem perder qualidade.

2. METODOLOGIA

O AV1 possui quatro diferentes kernels de transformada, a DCT (*Discrete Cosine Transform*), a ADST (*Asymetric Discrete Sine Transform*), a FLIPADST (*Flipped Asymetric Discrete Sine Transform*) e a IDTX (*Identity*). Por questão de espaço, para este trabalho a análise foi realizada apenas no contexto da DCT, por ser a transformada mais importante, já que o kernel DCT é o único que processa todos os tamanhos de bloco da transformada, os outros kernels são limitados a no máximo 32x32 (IDTX), 16x16 (ADST e FLIPADST).

A implementação da transformada DCT no AV1 utiliza uma representação inteira, com o codificador utilizando uma precisão nos coeficientes que varia de 10 até 13 bits. O decodificador por sua vez utiliza 12 bits de precisão na transformada.

Para coletar as magnitudes dos valores que são salvos no buffer, foi utilizado o software de referência do AV1, o *libaom* na versão 3.6.1, com a hash do git *1aa9bbbe*. O software foi modificado para capturar os maiores e menores valores que são salvos no buffer por tamanho de bloco. A captura dos valores se deu na função *fwd_txfm2d_c* do arquivo *av1/encoder/av1_fwd_txfm2d.c*. Além disso, foi utilizada a maior precisão para todos os coeficientes (13 bits) da transformada.

O software modificado foi então experimentado na sequência de vídeos *objective-2-slow*, recomendada pelo documento da IETF-NETVC para teste de codificadores de vídeo (DAEDE et al., 2020). Todos os vídeos são de resolução UHD 4K, com 10 bits de profundidade de cor e possuem 60 quadros cada. Foi utilizado o coeficiente de quantização 20 para codificação dos vídeos.

Os valores que foram extraídos do codificador foram o tamanho do bloco e o maior e menor valor que foi obtido durante a codificação de um vídeo para cada posição do buffer para aquele tamanho de bloco. Então os valores de todos os vídeos foram combinados para se obter os maiores e menores valores globais por tamanho de bloco.

Os valores foram então convertidos para o número de bits necessários para representar o valor de acordo com a seguinte fórmula $teto(\log_2(|valor|)) + 1$, onde a função $teto(x)$ retorna o menor inteiro maior ou igual a x .

3. RESULTADOS E DISCUSSÃO

Na Tabela 1 estão representados apenas os blocos quadrados, por questão de espaço disponível neste artigo. É possível observar que o número total de bits necessários para realizar a transposição dos blocos quadrados varia de oito a 15 bits. Sendo que nos blocos de 64X64 é onde existe a maior variação, isto porque como o número de coeficientes é muito grande, a amplitude para as frequências maiores é mais baixa e por consequência menos bits são necessários para representá-los.

Tabela 1 - Faixa de largura de bits observados para os tamanhos de bloco quadrados

Tamanho de Bloco	Menor	Maior
4x4	12	15
8x8	11	15
16x16	11	15
32x32	8	14
64x64	8	15

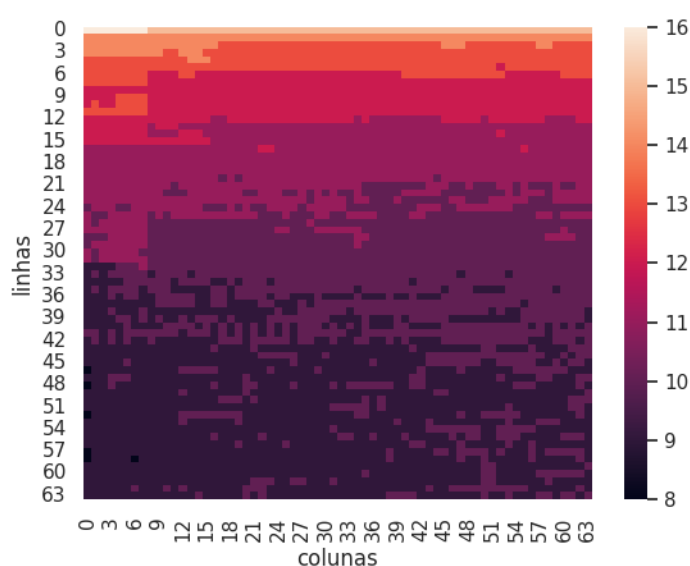


Figura 2 - Número de bits necessários por posição do buffer

O buffer que considera o maior número de bits observado por posição considerando todos os tamanhos de bloco sobrepostos é representado na Figura 2. Observa-se que no canto superior esquerdo são necessários mais bits (16 bits), enquanto que conforme as linhas vão descendo o número de bits necessários diminui (8 bits). Isto se dá porque no canto superior esquerdo fica o componente DC da transformada, que é o mais importante, uma vez que ele representa a média de todos os valores de entrada (KHAYAM, 2003). Conforme as linhas e as colunas aumentam, a frequência representada pelo coeficiente é maior, porém, a amplitude é menor, o que demanda uma menor largura de bits.

Para representar o buffer da Figura 2, são necessários 42770 bits. Isto representa uma redução de 34,74% em relação a um buffer que utilize 16 bits para todas as posições. Estes resultados demonstram que o projeto em hardware deste buffer pode ter ganhos em redução de área e de consumo de energia na mesma ordem, caso sejam implementado utilizando os valores máximo de bits indicados neste estudo.

4. CONCLUSÕES

Foi possível observar que os valores que são salvos no buffer de transposição de transformada 2D DCT no codificador AV1 não necessitam de todos os 16 bits em todas as posições. Sendo que é possível realizar a transformada sem perdas com o buffer possuindo uma redução no número de bits necessários de 34,74% ao considerar um buffer que suporte todos os tamanhos de bloco.

Como trabalhos futuros, pode ser feito o desenvolvimento de hardware para a transformada DCT utilizando os resultados obtidos neste trabalho, ou ainda pode ser avaliado o uso da mesma ideia em diferentes partes do codificador, como a predição inter quadros, ou a predição intra quadros para reduzir o número de bits necessários nestes módulos do codificador.

5. REFERÊNCIAS BIBLIOGRÁFICAS

MERT, Ahmet Can; KALALI, Ercan; HAMZAOGLU, Ilker. High performance 2D transform hardware for future video coding. **IEEE Transactions on Consumer Electronics**, v. 63, n. 2, p. 117-125, 2017.

CHEN, Yue et al. An overview of coding tools in AV1: The first video codec from the alliance for open media. **APSIPA Transactions on Signal and Information Processing**, v. 9, p. e6, 2020.

DAEDE, Thomas; NORKIN, Andrey; BRAILOVSKIY, Ilya. **Video Codec Testing and Quality Measurement**. [S.l.]: Internet Engineering Task Force, 2020. 23 p. (Internet-Draft). Disponível em: <https://datatracker.ietf.org/doc/draft-ietf-netvc-testing/09/>. Acesso em: 09 set. 2023.

RICHARDSON, Iain E. **Video codec design: developing image and video compression systems**. John Wiley & Sons, 2002.

KHAYAM, Syed Ali. The discrete cosine transform (DCT): theory and application. **Michigan State University**, v. 114, n. 1, p. 31, 2003.