ANÁLISE DO PARTICIONAMENTO DE SUPERBLOCOS DO FORMATO DE VÍDEO AV1

ALEX BORGES; ISIS BENDER; BRUNO ZATT;
MARCELO PORTO; GUILHERME CORREA
Video Technology Research Group (ViTech)
Programa de Pós-Graduação em Computação (PPGC)
Centro de Desenvolvimento Tecnológico (CDTec) - Universidade Federal de Pelotas (UFPel)
{amborges;idbender;zatt; porto;gcorrea}@inf.ufpel.edu.br

1. INTRODUÇÃO

A demanda por vídeos de maior resolução e sendo transmitidos pela internet vêm crescendo nos últimos anos (CLEMENT, 2019). A codificação de vídeos é um processo obrigatório hoje em dia. Contudo o *High Efficiency Video Coding* (BROSS, 2013), mesmo sendo o estado-da-arte na área, não ganhou adesão por parte do mercado, devido ao seu complexo e elevado custo de pagamentos de royalties (OZER, 2018). Com isso, o consórcio *Alliance for Open Media* (AOMEDIA, 2015) providenciou um novo codificador de vídeo livre de royalties e com maior eficiência de codificação que o HEVC, o *AOMedia Video 1* (CHEN, 2020). Um artigo recente demonstra que o AV1 é em média 27,7% superior ao HEVC para codificar vídeos (ZHAO, 2020). Recentemente, HAN (2020) disponibilizou o primeiro documento oficial que descreve o codificador AV1 em alto nível e de forma mais completa. Os trabalhos de CHEN (2020) e RIVAZ (2018) abordam vários aspectos do AV1 de forma mais superficial.

De forma geral, o AV1 segue o mesmo fluxo de codificação híbrida utilizado por outros codificadores modernos (HAN, 2020). Antecipando as diversas etapas de codificação, há a divisão do quadro em regiões, denominadas superblocos (SB). Cada SB possui uma estrutura que se subdivide em blocos. Blocos grandes são mais bem aproveitados em regiões homogêneas, pois permitem representar com poucos bits uma grande quantidade de píxeis repetidos; já as regiões heterogêneas são geralmente codificadas com blocos pequenos, pois resultam em baixo resíduo e alta fidelidade com a imagem original. Essa estrutura de subdivisão é chamada de árvore de particionamentos, sendo categorizada por dez tipos de particionamento e seis níveis de profundidade no formato AV1. Os tipos de particionamento definem as combinações dos blocos possíveis: quadrática, binária, ternárias e quaternária. Essas combinações estão presentes em seis níveis profundidade, partindo de um bloco quadrático de 128×128 (nível zero) píxeis até um bloco quadrático de 4×4 (nível cinco).

HAN (2020) descreve bem o processo de particionamento de um superbloco; contudo, não é conhecido nenhum artigo que aborde a complexidade (ou custo computacional) ou a eficiência de codificação resultante do uso desta estrutura no formato AV1. Esse tipo de informação auxilia os pesquisadores a descobrir pontos críticos para dedicar esforços de pesquisa. Objetivando sanar essa vacância de informação, ao menos em parte, este trabalho visa apresentar uma análise da eficiência de codificação do codificador de vídeo AV1 com foco específico na árvore de particionamento.

2. METODOLOGIA

O *libaom* (AOMEDIA, 2018) é o software de referência do formato de codificação de vídeo AV1. O *libaom* implementa a árvore de particionamentos de

SB, que é executada durante a codificação do vídeo. O software permite 18 configurações distintas, sumarizadas na Tabela 1.

O tamanho padrão do superbloco do AV1 é de 128×128. Contudo, é possível reduzi-lo para 64×64 (conf. 01). Também é possível desabilitar as divisões ternárias e quaternárias (conf. 03 e 04), assim como desabilitar qualquer tipo de particionamento não quadrático (conf. 02). Os níveis de profundidade permitidos da árvore do AV1 também podem ser controlados, identificando-se os níveis máximos e mínimos. Para tanto, deve-se respeitar duas regras básicas: primeiro, os níveis de profundidade devem ser contíguos; segundo, a subtração entre o nível mínimo e nível máximo deve ser no mínimo um. Dessa forma, há um total de 13 variações dos níveis de profundidade da árvore do AV1 (conf. 05 a 18). É importante mencionar que todas essas configurações são variações da configuração recomendada para o *libaom* (conf. 00), conforme CHEN (2020).

A fim de possibilitar os resultados para esses experimentos, foram utilizados onze vídeos de resoluções 1920×1080 e 4096×2160 píxeis, recomendados por DAEDE (2020): *Crowd Run*, *Guitar HDR Amazon*, *Netflix Crosswalk, Netflix Square and Timelapse, Netflix Tunnel Flag, Netflix Bar Scene, Netflix Boxing Practice, Netflix Dancers, Netflix Ritual Dance, Netflix Toddler Fountain* e *Street HDR Amazon*. Os vídeos foram codificados usando a versão 2.0 do *libaom* (*hash code* a5e3f02b1) em um servidor com Debian 4.19.118-2 instalado, possuindo Intel Xeon E5-4650 v3 com oito núcleos de 2.10GHz e com 512GB de memória.

3. RESULTADOS E DISCUSSÃO

A Tabela 2 sumariza os resultados médios obtidos com as configurações para todos os vídeos citados. Nesta tabela, Bjøntegaard Delta Rate (colunas "BDR") é a métrica que indica o percentual de aumento da taxa de bits da configuração observada, em relação à conf. 00, para uma mesma qualidade de imagem (BJØNTEGAARD, 2001). Já o Impacto de Tempo de Processamento (ITP) informa o percentual do custo computacional que essa configuração tem na execução do *libaom*, ou seja, quanto a mais de tempo de processamento há no *libaom* quando determinada configuração está habilitada.

Permitir que o *libaom* codifique com apenas dois níveis de profundidade da

Tabela 1. Configurações da árvore de particionamento com o software *libaom*

Conf.	Comando	Conf.	Comando	Conf.	Comando
00	--verbose --psnr --lag-in-frames=19 --passes=2 --frame-parallel=0 --tile-columns=0 --cpu-used=0 --threads=1 --kf-min-dist=1000 --kf-max-dist=1000 --end-usage=q --cq-level={20, 32, 43, 55}				
01	--enable-rect-partitions=0	07	--min-partition-size=16 --max-partition-size=128	13	--min-partition-size=16 --max-partition-size=32
02	--enable-ab-partitions=0	08	--min-partition-size=8 --max-partition-size=128	14	--min-partition-size=8 --max-partition-size=32
03	--enable-1to4-partitions=0	09	--min-partition-size=32 --max-partition-size=64	15	--min-partition-size=4 --max-partition-size=32
04	--sb-size=64	10	--min-partition-size=16 --max-partition-size=64	16	--min-partition-size=8 --max-partition-size=16
05	--min-partition-size=64 --max-partition-size=128	11	--min-partition-size=8 --max-partition-size=64	17	--min-partition-size=4 --max-partition-size=16
06	--min-partition-size=32 --max-partition-size=128	12	--min-partition-size=4 --max-partition-size=64	18	--min-partition-size=4 --max-partition-size=8

Tabela 2. Média dos resultados obtidos com todas as configurações testadas

Conf.	BDR	ITP	Conf.	BDR	ITP	Conf.	BDR	ITP
01	3.7712	33.92	07	5.3355	19.40	13	11.3392	45.91
02	0.2809	7.02	08	0.2506	5.36	14	6.1271	32.14
03	0.3088	6.85	09	21.8884	52.16	15	5.8899	30.90
04	0.8622	10.64	10	6.2660	29.80	16	22.0947	59.62
05	65.2027	64.72	11	1.1159	13.44	17	21.7981	57.70
06	20.8918	40.72	12	0.8691	12.22	18	77.9409	78.80

árvore de particionamentos gera os melhores ITP (respectivamente, conf. 18 e 05). Ambas configurações representam extremos: enquanto a conf. 18 apenas utiliza blocos pequenos (8×8 e 4×4), a conf. 05 utiliza somente os dois primeiros níveis de profundidade (blocos 128×128 e 64×64). Contudo, por essas configurações explorarem o melhor de cada área (ou bitrate ou qualidade), falham por não proverem flexibilidade ao codificador, para que se adapte aos diferentes tipos de conteúdo. Isso pode ser bem expresso com o aumento expressivo do BDR, em ambos os casos com mais de 65%.

As configurações que apresentam grandes impactos na execução do libaom seguem a mesma premissa do parágrafo anterior, configurações que habilitam apenas blocos menores que 32×32 (confs. 16 e 17) ou maiores que 16×16 (confs. 06 e 09). A habilitação desses casos expressa 40% ou mais de ITP, justificando-se por evitar a perda da eficiência de codificação em mais de 20% de BDR. Outra configuração que merece ser mencionada é a conf. 01, onde blocos não-quadráticos são desabilitados. Nesta configuração, há um significativo ITP de quase 34%, havendo pouca perda de BDR (3.77%). Considerando as devidas proporções, é correto afirmar que trabalhos de aceleração da codificação do AV1 que envolverem modos de decisão rápidas de particionamentos quadráticos e não-quadráticos podem apresentar um ganho no tempo de execução do *libaom* considerável, podendo haver pouca perda de eficiência de codificação.

As demais configurações não apresentam resultados significativos no impacto do tempo de processamento em relação ao custo da perda de eficiência de codificação. Portanto, não serão apresentados neste trabalho com maiores detalhes.

4. CONCLUSÕES

Neste trabalho foi apresentado uma breve análise de complexidade e da eficiência de codificação do uso de diferentes configurações de particionamento de superbloco no codificador de vídeo segundo o formato AV1, através do *libaom*. Com esse software é possível manipular a árvore de particionamentos de 18 formas diferentes, comparando-as com a codificação padrão para o AV1. Os resultados mostram que é importante manter uma variedade de tamanhos de blocos habilitados na estrutura de particionamento do AV1. Os resultados obtidos indicam que estratégias que foquem na escolha rápida de blocos, principalmente os não-quadráticos, têm potencial de apresentar bons resultados de redução de complexidade sem afetar significativamente a eficiência de codificação.

5. AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001, FAPERGS e CNPq.

6. REFERÊNCIAS BIBLIOGRÁFICAS

CLEMENT, J.. **Distribution of global downstream internet traffic as of October 2018, by category**. Statista, 13 fev. 2019. Acessado em 12 ago. 2020. Online. Disponível em: https://www.statista.com/statistics/271735/internet-traffic-share-by-category-worldwide/.

BROSS, B; HAN, W.J; OHM, J.R.; SULLIVAN, G. J.; WANG, Y.K.; WIEGAND, T.. JCTVC-L1003: High Efficiency Video Coding (HEVC): text specification draft 10 (for FDIS and Consent). In: **12TH JCT-VC MEETING,** Geneva, 2013.

OZER, J.. **HEVC Advance Cuts Content Fees on Streaming**. Streaming Media, 13 mar. 2018. Online. Acessado em 12 ago. 2020. Disponível em https://www.streamingmedia.com/Articles/News/Online-Video-News/HEVC-Advance-Cuts-Content-Fees-on-Streaming-123828.aspx.

AOMEDIA. **Alliance for Open Media**. 2015. Online. Acessado em 08 set. 2020. Disponível em: http://aomedia.org/.

CHEN, Y.; 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**, Cambridge, v.9, 2020. DOI: 10.1017/ATSIP.2020.2.

ZHAO, X.; LIU, S.; ZHAO, L.; XU, X.; ZHU, B.; LI, X.. A comparative study of HEVC, VVC, VP9, AV1 and AVS3 video codecs. **Proceedings on SPIE 11510, Applications of Digital Image Processing XLIII**, 1151011, 21 ago. 2020. DOI: 10.1117/12.2570003

HAN, J.; et. al. A Technical Overview of AV1. arXiv, 13 ago. 2020. Online. Acessado em 20 ago. 2020. Disponível em https://arxiv.org/abs/2008.06091.

RIVAZ, P.; HAUGHTON, J.. **AV1 Bitstream & Decoding Process Specification**. Alliance for Open Media, 2018. Online. Acessado em 6 jul. 2020. Disponível em https://aomediacodec.github.io/av1-spec/av1-spec.pdf.

AOMEDIA. **AV1 Codec Library**. Alliance for Open Media, 2018. Online. Acessado em 6 jul. 2020. Disponível em https://aomedia.googlesource.com/aom/.

DAEDE, T.; NORKIN, A.; BRAILOVSKIY, I.. **Video Codec Testing and Quality Measurement: draft-ietf-netvc-testing-09**. Network Working Group, 31 jan. 2020. Online. Acessado em 22 jul. 2020. Disponível em https://tools.ietf.org/html/draft-ietf-netvc-testing-09.

BJØNTEGAARD, G.. VCEG-M33: Calculation of Average PSNR Differences between RD curves. In: **VIDEO CODING EXPERTS GROUP (VCEG)**, Austin, 2001.