REDUÇÃO DO TEMPO DO TEST ZONE SEARCH NA PREDIÇÃO INTER-QUADROS DO VVC USANDO APRENDIZADO DE MÁQUINA

RAMIRO VIANA[1]; MARCELO PORTO[2];
GUILHERME CORRÊA[3]; LUCIANO AGOSTINI[4]

[1]*Universidade Federal de Pelotas (UFPel) – rgsviana@inf.ufpel.edu.br*
[2]*Universidade Federal de Pelotas (UFPel) – porto@inf.ufpel.edu.br*
[3]*Universidade Federal de Pelotas (UFPel) – gcorrea@inf.ufpel.edu.br*
[4]*Universidade Federal de Pelotas (UFPel) – agostini@inf.ufpel.edu.br*

1. INTRODUÇÃO

Nos últimos anos houve um grande aumento na demanda por transmissão de vídeos digitais devido às diversas plataformas de *streaming* disponíveis para entretenimento, reuniões virtuais e trabalho remoto. Tudo isso levou a mais pesquisas sobre o processo de codificação de vídeo, de forma a comprimir o tamanho do arquivo digital do vídeo. Codificar um vídeo demanda muito tempo e esforço computacional devido às várias etapas necessárias a serem executadas por cada codificador. Essas etapas consistem em Predição Inter-Quadros, Predição Intra-Quadros, Transformadas Direta e Inversa, Quantização Direta e Inversa, Filtro de Laço e Codificador de Entropia (SALDANHA et al., 2020).

Em 2023, o codificador de vídeo estado da arte é o *Versatile Video Coding* (VVC) (CHEN et al., 2023). Apesar dos requisitos computacionais significativos, o VVC oferece taxas de compressão notáveis e supera outros codificadores disponíveis comercialmente em termos de eficiência de compressão. Ele consegue isso usando ferramentas e algoritmos avançados, resultando em compressões de vídeo altamente eficientes.

No VVC, os quadros do vídeo a serem codificados são primeiramente particionados em blocos denominados Unidades de Árvore de Codificação, do inglês *Coding Tree Units* (CTU), contendo um número de amostras variando de 32x32 a 128x128. Atuando como a raiz de uma estrutura Quadtree, cada CTU é subdividida em Unidades de Codificação, do inglês *Coding Units* (CU) a serem codificadas, com amostras variando de 4x4 a 128x128, sendo também utilizadas para melhor adequar o processo de codificação às características da imagem.

Um dos algoritmos mais importantes na Predição Inter-Quadros do VVC é o *Test Zone Search* (TZS), que é um algoritmo de busca que visa encontrar o melhor casamento de blocos em quadros de referência, aproveitando a entropia espacial e temporal em áreas vizinhas de um bloco candidato (MARTINS et al., 2017). No entanto, o TZS é uma das partes mais demoradas do processo de codificação de vídeo. No VVC, o TZS é aplicado em 12 número de amostras diferentes, variando de 16x16 a 128x128. O TZS executa quatro etapas: *Motion Vector Prediction*, *First Search*, *Raster Search* e *Refinement* (SANT'ANNA et al., 2021). Sendo que a etapa do *Motion Vector Prediction* sempre precisa ser executada pelo codificador (GONÇALVES et al., 2017).

Aprendizado de Máquina, do inglês *Machine Learning* (ML), se tornou uma das principais ferramentas para melhorar e otimizar codificadores de vídeo, dada a grande quantidade de dados que podem ser facilmente gerados e usados em muitos campos. O modelo de Aprendizado de Máquina mais simples é a Árvore de Decisão, do inglês *Decision Tree* (DT), porque é fácil de entender e simples de ser implementado, tornando-se o bloco de construção básico para ciência de dados.

Este trabalho apresenta um algoritmo para a Predição Inter-Quadros do VVC que seleciona instâncias onde as etapas do TZS devem ser ignoradas usando Aprendizado de Máquina de forma a reduzir o tempo total de codificação.

2. METODOLOGIA

Os experimentos foram realizados no *VVC Test Model* (VTM) versão 16.2 (SUEHRING, 2023), que é o software de referência para o VVC. Neste trabalho, a configuração *Random-Access* (RA) foi empregada em todos os experimentos, com foco na avaliação das etapas do *Test Zone Search*. Os Parâmetros de Quantização, do inglês *Quantization Parameters* (QP) utilizados foram 22, 27, 32 e 37, conforme especificado pelas Condições Comuns de Teste, do inglês *Commom Test Conditions* (CTCs) (BOSSEN et al., 2020). Todos os experimentos foram realizados em um servidor Intel Xeon CPU E5-2640 v3 @ 2,60 GHz, com oito núcleos e 32 GB de RAM.

Primeiramente, foi realizada uma análise de tempo de cada etapa do TZS, então as *features* foram extraídas do VTM, e por fim, implementou-se o Aprendizado de Máquina no VTM utilizando as *features* para otimizar o TZS.

Considerando as etapas do TZS apresentadas anteriormente, seu consumo de tempo é analisado com base no processo de codificação dos vídeos selecionados. Para esta análise, foram codificados os primeiros 40 quadros de 23 sequências de vídeo diferentes, de acordo como definido pelas CTCs. O VTM foi modificado apenas para inclusão de contadores de tempo. Estas sequências foram: três da classe A1 (*Tango2*, *FoodMarket4* e *Campfire*), três da classe A2 (*CatRobot*, *DaylightRoad2* e *ParkRunning3*), cinco de classe B (*MarketPlace*, *RitualDance*, *Cactus*, *BasketballDrive* e *BQTerrace*), quatro da classe C (*BasketballDrill*, *BQMall*, *PartyScene* e *RaceHorsesC*), quatro da classe D (*BasketballPass*, *BQSquare*, *BlowingBubbles* e *RaceHorses*) e quatro da classe F (*SlideEditing*, *SlideShow*, *BasketballDrillText* e *ArenaOfValor*). Os resultados médios desta análise estão apresentados na Figura 1.

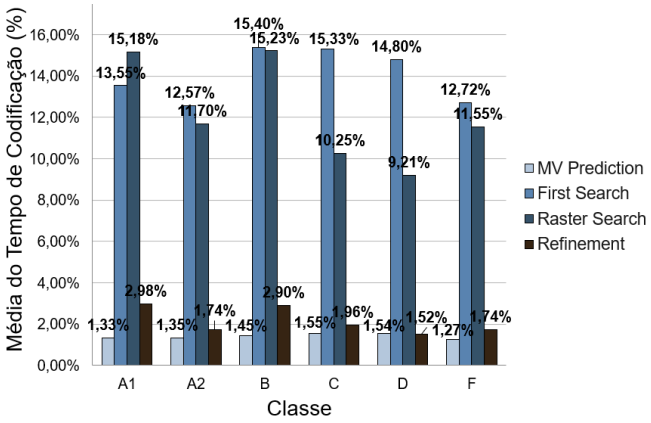

Figura 1: Tempo médio de cada etapa TZS para diferentes classes das CTCs em relação ao tempo total de codificação.

Analisando a Figura 1, é possível observar que o *First Search* e o *Raster Search* são as etapas mais demoradas dentro do algoritmo TZS. Além disso, também é possível verificar que o TZS como um todo ocupa uma alta porcentagem do tempo total de codificação do vídeo.

Para extrair as *features*, o codificador VTM foi executado normalmente, evitando processamentos adicionais para obtenção desses dados. Foi empregado

um conjunto de 10 sequências de vídeo com resoluções variadas: quatro CIF (*Highway, Foreman, Container* e *Coastguard*), duas HD (*Vidyo4* e *KristenAndSara*), duas Full HD (*Netflix_TunnelFlag e Rush_field_cuts*) e duas Ultra HD (*Beauty* e *Lips*). Os primeiros 20 quadros de cada sequência foram codificados quatro vezes, uma vez para cada QP (22, 27, 32 e 37). Três *datasets* foram usados para obter essas sequências de vídeo: UVG, NETVC e JVET.

As *features* foram extraídas da CU atual e das CUs previamente codificadas (pai, esquerda e vizinhos acima). Essas *features* são: Vetor de Movimento (valores *x* e *y*), QP, posição da CU dentro do quadro e precisão do Vetor de Movimento. Foram criados 12 conjuntos de dados de *features*, um para cada tamanho de CU suportado pelo TZS.

Um modelo de Árvore de Decisão foi treinado para cada um dos 12 diferentes tamanhos de CUs. Para isso, o conjunto de dados foi adequadamente balanceado e os modelos foram treinados usando a biblioteca *Python Scikit-learn* de maneira offline. Isso significa que um conjunto de dados fixo foi obtido antes da realização dos testes e permaneceu inalterado durante o processo de treinamento. A profundidade máxima das árvores foi definida em sete níveis.

Para cada tamanho de CU, a Árvore de Decisão específica é utilizada e define, para o bloco que está sendo processado, se devem ser executadas as quatro etapas do TZS ou se apenas a primeira deve ser aplicada, pulando as três últimas, como mostrado na Figura 2.

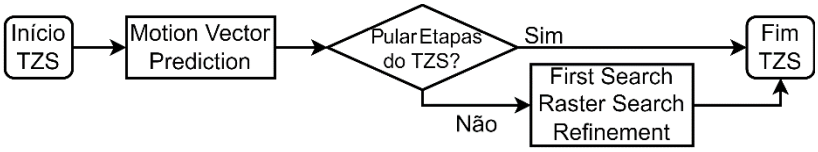

Figura 2: Fluxograma do método proposto.

3. RESULTADOS E DISCUSSÃO

Os primeiros 40 quadros das 23 sequências de vídeo usadas anteriormente para a análise TZS foram novamente codificados quatro vezes, uma vez para cada QP, mas desta vez usando o VTM 16.2 modificado com temporizadores adicionais para cronometrar cada etapa TZS e o algoritmo do TZS otimizado desenvolvido neste trabalho com Árvores de Decisão. A destacar que o conjunto de vídeos usados no treinamento é totalmente distinto dos vídeos utilizados na avaliação.

A avaliação dos resultados dos experimentos usou a métrica *Bjøntegaard Delta-BitRate* (BD-BR) para avaliar a eficiência de codificação, além das métricas Redução de Tempo (TR) e Redução de Tempo TZS (TR TZS). Os resultados médios para todas as sequências e todos os QPs estão apresentados na Tabela 1, organizados por Classe das CTCs.

Tabela 1: Resultados da redução de tempo e eficiência de codificação.

Classe das Sequências de Vídeo	TR	TR TZS	BD-BR
Média da Classe A1	27,43%	89,96%	0,13%
Média da Classe A2	23,80%	89,67%	0,52%
Média da Classe B	24,57%	87,97%	0,31%
Média da Classe C	21,73%	86,48%	0,53%
Média da Classe D	16,87%	83,64%	0,64%
Média da Classe F	17,12%	84,14%	0,59%
Média Geral	21,92%	86,98%	0,45%

4. CONCLUSÕES

Este trabalho apresentou uma solução para reduzir o tempo total de codificação usando Aprendizado de Máquina no algoritmo *Test Zone Search*.

A solução proposta foi implementada com o uso de Árvores de Decisão e conseguiu uma redução média de 21,92% do tempo total de codificação e redução de 86,98% do tempo de processamento do TZS tendo uma perda de eficiência de 0,45% em BD-BR. Esse resultado foi considerado muito positivo, pois indica que para codificar o vídeo de mesma qualidade, a solução apresentada nesse trabalho irá necessitar de apenas 0,45% mais bits, enquanto consegue reduzir em quase 22% o tempo total de processamento do codificador.

5. REFERÊNCIAS BIBLIOGRÁFICAS

BOSSEN, F.; BOYCE, J.; SUEHRING, K.; LI, X.; VADIM, S. **VTM Common Test Conditions and Software Reference Configurations for SDR Vídeo.** Document WG 05 MPEG Joint Video Coding Team(s) with ITU-T SG 16 JVET-T2010.ed., out. 2020. Online.

CHEN, B.; WANG, Z; LI, B; WANG, S.; YE, Y. Compact Temporal Trajectory Representation for Talking Face Video Compression. **IEEE Transactions on Circuits and Systems for Video Technology**, p.1–1, 2023.

GONÇALVES, P.; CORRÊA, G.; PORTO, M.; ZATT, B.; AGOSTINI, L. Multiple Early-Termination Scheme for TZ Search Algorithm Based on Data Mining and Decision Trees. **IEEE 19TH INTERNATIONAL WORKSHOP ON MULTIMEDIA SIGNAL PROCESSING (MMSP)**, Luton, Reino Unido, 2017. **Anais. . .** 2017. p.1–6.

MARTINS, A; PENNY, W.; WEBER, M.; PALOMINO, D.; MATTOS, J.; PORTO, M.; AGOSTINI, L.; ZATT, B. Cache Memory Energy Efficiency Exploration for the HEVC Motion Estimation. In: **VII BRAZILIAN SYMPOSIUM ON COMPUTING SYSTEMS ENGINEERING (SBESC)**, Curitiba, PR, Brazil, 2017. **Anais. . .** 2017. p.31–38.

SALDANHA, M.; CORRÊA, M.; CORRÊA, G.; PALOMINO, D.; PORTO, M.; ZATT, B.; AGOSTINI, L. An Overview of Dedicated Hardware Designs for State-of-the- Art AV1 and H. 266/VVC Video Codecs. In: **IEEE INTERNATIONAL CONFERENCE ON ELECTRONICS, CIRCUITS AND SYSTEMS (ICECS)**, Glasgow, Reino Unido, 2020. **Anais. . .** 2020. p.1–4.

SANT'ANNA, G. B.; CANCELLIER, L. H.; SEIDEL, I.; GRELLERT, M.; GUNTZEL, J. L. Relying on a Rate Constraint to Reduce Motion Estimation Complexity. In: **ICASSP 2021 - 2021 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING (ICASSP)**, Toronto, Canadá, 2021. **Anais. . .** 2021. p.1560–1564.

SUEHRING, K. **VTM-16.2**. JVET, 23 maio 2022. Acessado em 28 abril 2023. Online. Disponível em: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/releases/VTM-16.2.