ESTIMAÇÃO DE MOVIMENTO AFFINE BASEADA EM APRENDIZADO DE MÁQUINA PARA REDUZIR O TEMPO DE CODIFICAÇÃO DO VVC

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

De acordo com o *Global Internet Phenomena Report 2024*, os principais provedores de conteúdo continuaram a dominar o tráfego de dados, com grandes empresas como Google, Facebook e Netflix sendo responsáveis por 65% do tráfego de internet fixa e 68% do tráfego móvel (MARWAHA, 2024). Essa participação é atribuída principalmente ao consumo extensivo de vídeos digitais em serviços de *streaming* e mídias sociais. Devido à alta complexidade envolvida na transmissão e no armazenamento desses vídeos, o desenvolvimento de codificadores eficientes tornou-se imperativo. Esses codificadores são projetados para comprimir dados de forma eficiente, mantendo a qualidade da imagem. Mas isso exige um esforço computacional considerável, desta forma, impulsionando um aumento na pesquisa sobre o processo de codificação de vídeo.

O *Versatile Video Coding* (VVC) é a tecnologia de codificação de vídeo mais avançada atualmente. Apesar de suas exigências computacionais substanciais, o VVC se destaca por suas taxas de compressão excepcionais, superando outros codificadores comerciais em eficiência, graças às suas ferramentas e algoritmos avançados que permitem uma compressão de vídeo altamente eficiente (VIANA et al., 2025). A Predição Inter-Quadros do VVC, responsável pela utilização da redundância temporal para melhor predizer os quadros sendo codificados, incorpora ferramentas-chave para otimizar a codificação de vídeo, explorando a redundância temporal, como a Estimação de Movimento *Affine*.

A Estimação de Movimento *Affine* (AME) é crucial para aprimorar a Predição Inter-Quadros, estimando movimentos complexos, como redimensionamento, rotação e cisalhamento. No entanto, seu uso aumenta significativamente as demandas computacionais, resultando em um tempo total de codificação maior no processo de predição. O VVC emprega dois estágios de AME para mapear esses movimentos: o modelo de 4-Parâmetros para movimentos simples, como redimensionamento e rotação, e o modelo de 6-Parâmetros para movimentos mais complexos, como ajuste de proporção de tela e cisalhamento. A AME utiliza 12 tamanhos de Blocos de Codificação, do inglês *Coding Blocks* (CBs), entre 16x16 e 128x128. (VIANA et al., 2024).

O aumento do esforço computacional exigido pelos codificadores de vídeo atuais impulsionou o desenvolvimento de novas soluções para reduzir esse esforço com baixo impacto na eficiência da codificação. Desta forma, o Aprendizado de Máquina, do inglês *Machine Learning* (ML), se tornou uma ferramenta crucial. Entre os modelos de ML, as Árvores de Decisão, do inglês *Decision Trees* (DT) se destacam por sua simplicidade e facilidade de interpretação, tornando-se uma ferramenta muito importante na ciência de dados.

Este trabalho propõe uma aceleração da Predição Inter-Quadros do VVC, com foco na AME, utilizando 12 Árvores de Decisão, uma para cada tamanho de CB, para reduzir a complexidade computacional e o tempo total de codificação, sem comprometer significativamente a eficiência de codificação.

2. METODOLOGIA

Os experimentos foram realizados no *software* de referência do VVC: *VVC Test Model* (VTM) versão 16.2 (SUEHRING, 2023), usando a configuração *Random-Access*, com foco específico na avaliação e aceleração da AME. 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 com dois Intel Xeon CPU E5-2640 v3 @ 2,60 GHz de oito núcleos e 128 GB de RAM.

A princípio, foi realizada uma análise de tempo de cada etapa da AME. Então as *features* foram extraídas do VTM para a AME, seguida pela implementação do Aprendizado de Máquina no VTM utilizando as *features* para otimizar a AME.

Considerando os modos da AME apresentados anteriormente, seu consumo de tempo foi analisado com base no processo de codificação dos vídeos selecionados. Para esta análise, foram codificados os primeiros 64 quadros de 23 sequências de vídeo diferentes, 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.

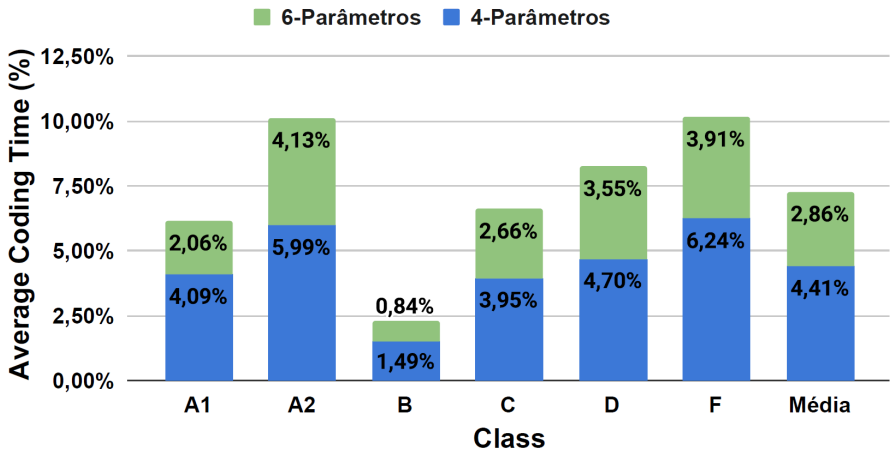

Figura 1: Tempo médio usado por cada etapa da AME em relação ao tempo total de codificação do VVC, considerando diferentes classes de vídeos.

A Figura 1 mostra que os dois modos da AME combinados, representam em média 7,27 % do tempo total de codificação.

A aceleração do VVC proposta neste trabalho usa 12 modelos de Árvore de Decisão para pular seletivamente todo o processo da Estimação de Movimento *Affine* sob condições específicas, conforme ilustrado na Figura 2 com as decisões das DTs destacadas em azul.

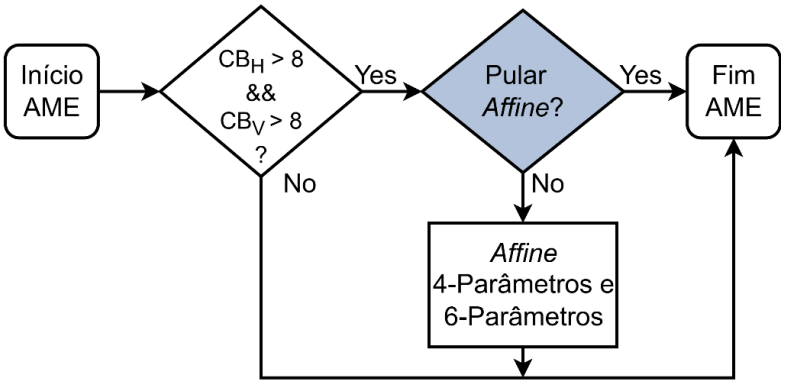

Figura 2: Fluxograma do método proposto.

Para extrair *features* relevantes, o codificador VTM foi executado normalmente com contagens adicionais de tempo de estágios específicos e a extração de dados adicionais. Foi utilizado um novo conjunto de 10 sequências de vídeo com diferentes resoluções: quatro vídeos em *Common Intermediate Format* (CIF) (*Highway*, *Foreman*, *Container* e *Coastguard*), dois vídeos em HD (*KristenAndSara* e *Vidyo4*), dois vídeos em Full HD (*Netflix_TunnelFlag* e *rush_field_cuts*) e dois vídeos em Ultra HD (*Beauty* e *Lips*). Os primeiros 20 quadros de cada sequência foram codificados para extrair as *features*, e cada sequência foi codificada quatro vezes, uma para cada QP. O processo de treinamento de cada Árvore de Decisão é mostrado na Figura 3.

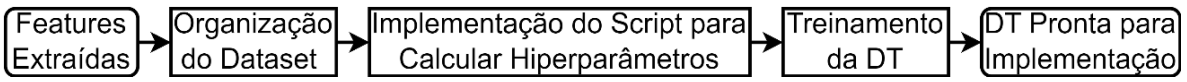

Figura 3: Fluxograma do processo para criação e treinamento de uma DT.

Os hiperparâmetros encontrados e utilizados foram: *Criterion* (*entropy* ou *gini*), *Min Samples Split*, *Min Samples Leaf*, *Max Features*, *Max Depth* e *Max Leaf Nodes*.

3. RESULTADOS E DISCUSSÃO

Os primeiros 64 quadros das 23 sequências de vídeo usadas anteriormente para a análise da AME foram novamente codificados quatro vezes, uma vez para cada QP, mas desta vez usando o VTM 16.2 desenvolvido neste trabalho, com a AME acelerada, com temporizadores adicionais para cronometrar a AME otimizada com Árvores de Decisão. É importante realçar que as sequências das CTCs utilizadas nestes experimentos são completamente diferentes daquelas usadas durante o processo de treinamento.

A avaliação dos resultados dos experimentos usou as métricas Redução de Tempo (TR) e Redução de Tempo AME (TR AME), além da métrica *Bjøntegaard Delta-BitRate* (BD-BR) para avaliar a eficiência de codificação. 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 AME	BD-BR
Média da Classe A1	5,65%	58,41%	0,25%
Média da Classe A2	7,30%	58,10%	1,18%
Média da Classe B	3,91%	65,56%	0,40%
Média da Classe C	4,76%	64,64%	0,42%
Média da Classe D	7,03%	73,67%	0,72%
Média da Classe F	5,48%	55,76%	0,17%
Média Geral	5,54%	63,20%	0,50%

4. CONCLUSÕES

Este trabalho apresentou uma solução baseada em Aprendizado de Máquina para reduzir o esforço computacional da ferramenta AME no VVC. A solução utiliza modelos de Árvores de Decisão para pular seletivamente todo o processo da AME. No total, 12 Árvores de Decisão foram treinadas, uma para cada tamanho de CB suportado pela AME no VVC.

O *software* VTM acelerado alcançou uma redução média de 5,54% no tempo total de codificação do VVC. Especificamente, reduziu o tempo médio de processamento da AME em 63,20%, com uma perda média de eficiência correspondente de 0,50% em BD-BR. Este trabalho alcançou uma boa redução no tempo de codificação, mantendo uma baixa perda de eficiência de codificação.

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.

MARWAHA, S. **Sandvine's 2024 Global Internet Phenomena Report: Global Internet Usage Continues to Grow**. AppLogic Networks, EUA, 09 mar. 2024. Especiais. Acessado em 07 abr. 2025. Online. Disponível em: https://www.applogicnetworks.com/blog/sandvines-2024-global-internet-phenomena-report-global-internet-usage-continues-to-grow

SUEHRING, K. **VTM-16.2**. JVET, 23 mai. 2022. Acessado em 08 fev. 2025. Online. Disponível em: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/releases/VTM-16.2.

VIANA, R.; FERREIRA, R.; LOOSE, M.; PORTO, M.; CORRÊA, G.; AGOSTINI, L. A Hardware-Friendly Acceleration of VVC Affine Motion Estimation Using Decision Trees**. IEEE 37TH SYMPOSIUM ON INTEGRATED CIRCUITS AND SYSTEMS DESIGN (SBCCI)**, João Pessoa, Brasil, 2024. **Anais**. . . 2024. p.1–5.

VIANA, R.; LOOSE, CONCEIÇÃO, R.; M.; PORTO, M.; CORRÊA, G.; AGOSTINI, L. Fast VVC Test Zone Search and Affine Motion Estimation Using Machine Learning**. IEEE 16TH LATIN AMERICA SYMPOSIUM ON CIRCUITS AND SYSTEMS (LASCAS)**, Bento Gonçalves, Brasil, 2025. **Anais**. . . 2025. p.1–5.