IMPLEMENTAÇÃO DE MODELOS DE MACHINE LEARNING NA CODIFICAÇÃO INTRA-QUADRO DE VÍDEOS 360º NO PADRÃO DE CODIFICAÇÃO DE VÍDEO HEVC.

BERNARDO BELING[1]; IAGO STORCH[2]; DANIEL PALOMINO[1]

[1]*Universidade Federal de Pelotas – Video Technology Research Group (ViTech)*
{berbeling,dpalomino}@inf.ufpel.edu.br
[2]*Universidade Federal do Rio Grande do Sul – icstorch@inf.ufrgs.edu.br*

1. INTRODUÇÃO

Durante a pandemia do COVID-19, o uso de vídeos omnidirecionais, popularmente conhecidos como vídeos 360º, serviu como uma alternativa ao mercado decadente de turismo, de acordo com o (Acessibletourism, 2020). Essa alternativa se torna viável, pois vídeos 360º permitem que o usuário controle livremente o ponto de vista do vídeo enquanto está assistindo, gerando uma sensação de imersão ao ambiente. Porém, dentro do escopo de vídeos digitais existe a necessidade da compressão de vídeo, pois um vídeo digital em seu formato original necessita de uma grande quantidade de dados para ser representado. Por exemplo: um vídeo 360 de resolução 8K (8192×4096), possuindo três canais de cores por pixel representados por 1 byte, sendo apresentado à 60 quadros por segundo e com duração de 45s possui 271,79 GB de informação (8192×4096×60×3×45). Sendo assim, o processo de compressão desses vídeos se torna indispensável, pois busca reduzir a quantidade de dados necessários para representar o vídeo com o mínimo de perdas visuais possível.

A compressão de vídeo é realizada através de padrões de codificação que buscam explorar de forma efetiva redundâncias no sinal de vídeo, comumente encontradas entre pixels vizinhos e quadros vizinhos. Esse comportamento ocorre, pois um vídeo é uma sequência de imagens (quadros) apresentadas de forma sequencial a uma determinada taxa por segundo, com isso, regiões de um mesmo quadro e quadros vizinhos tendem a ser muito semelhantes. Por conseguinte, no padrão de codificação de vídeo *High Efficiency Video Coding* (HEVC) a compressão Intra-quadro é utilizada para explorar redundâncias espaciais, isto é, redundâncias dentro de um quadro. Para isso, o quadro é particionado em blocos de tamanho 64×64 pixels que são reduzidos de forma recursiva em sub-blocos até incorporarem *Predictions Units* (PUs), que podem possuir tamanho até 4×4 pixels. Em uma PU, é computada a escolha do melhor modo de predição para codificar a mesma, que no HEVC, consiste no teste de 35 modos através de uma etapa com alto custo computacional.

Considerando que padrões de codificação de vídeo como o HEVC realizam a compressão de vídeos 2D, visando possibilitar a compressão de vídeos omnidirecionais, a (JVET, 2018) propôs uma biblioteca contendo diversas projeções utilizadas para converter vídeos esféricos para vídeos bidimenionais. Dentre elas, a mais utilizada é a projeção Equiretangular (ERP), que mapeia coordenadas esféricas em coordenadas verticais e horizontais em um plano cartesiano. Porém, a projeção ERP acaba gerando distorções de imagem quando projeta regiões polares da esfera, sendo necessário um esticamento de pixels para preencher o plano em regiões com raio esférico menor. Em vista disso, em (STORCH, 2019) foi feita uma análise de características da predição Intra-quadro do padrão HEVC em vídeos 360 ERP, e foi observado uma tendência de seleção

de modos de predição DC/Planar (modo 0 e 1) e horizontais (modo 9, 10 e 11) em regiões distorcidas pela projeção ERP.

Sendo assim, considerando técnicas de *Machine Learning* (ML) que tem como objetivo identificar padrões baseado em dados de entrada (*Features*) para calcular uma predição sobre um dado de saída escolhido (*Label*), é possível elaborar e treinar um modelo que receba dados indicando essas características de distorção que geram a tendência de escolha dos modos de modo a reduzir a quantidade de modos de predição testados, reduzindo o custo computacional do processo de codificação de vídeos 360 ERP.

2. METODOLOGIA

Considerando o trabalho realizado pelos autores em (BELING, 2020), onde foi elaborada uma técnica que explora o particionamento de tamanhos de bloco baseada nas características da projeção ERP, o objetivo deste trabalho é incrementar a técnica fazendo uso de ML para reduzir a quantidade de modos de predição de PUs testados na etapa de *Rate Distortion Optimization* (RDO), que representa a etapa com maior custo computacional do HEVC. Para isso, baseado na tendência de seleção dos modos de predição DC, Planar e horizontais em regiões distorcidas pela projeção ERP, inicialmente foram pensados dois modelos de ML: um modelo que agrupa os modos DC e Planar (0 e 1) e um modelo que agrupa os modos horizontais (9, 10 e 11), cuja saída dos modelos é um booleano que caso seja validado, realiza a etapa de RDO somente sobre os modos agrupados, caso nenhum dos modelos seja validado, a etapa de RDO será executada com todos modos de predição escolhidos pelo codificador.

Em primeiro momento, visando uma abordagem inicial e experimental, os esforços se deram em implementar o modelo que valida os modos DC/Planar. Para tal fim, foi necessário realizar uma análise sobre quais *Features* seriam ideais para construir o modelo. Sendo assim, as *Features* escolhidas no modelo DC/Planar foram: posição vertical (1) e horizontal (2) da PU codificada; tamanho da PU codificada (3); RD-Cost do melhor modo de predição escolhido para a PU (4), que representa uma estimativa menos complexa do RDO; Variância (5), que consiste no cálculo de homogeneidade da textura da PU; RD-Cost do melhor modo de predição da PU acima (6), à esquerda (7), esquerda acima (8) e direita acima (9) da PU codificada; Lista de *Most Probable Modes* (MPM) do próprio codificador (10), que indica três modos mais prováveis de serem selecionados; Lista de *Rough Mode Decision* (RMD) que lista de forma ordenada os modos mais prováveis baseado no valor de RD-Cost (11); o menor RD-Cost entre os modos 0 e 1 (12); e por fim, o melhor modo de predição escolhido para a PU codificada (13). Em vista das *Features* escolhidas, o modo de predição escolhido (*feature* 13) é utilizado como *Label* do modelo, i.e., será a informação que o modelo irá realizar a predição, baseado nos dados de entrada.

Para gerar o *dataset* de entrada do modelo DC/Planar, as *Features* propostas foram extraídas da codificação Intra-quadro do vídeo 360 ERP *Aerial City*, utilizando o *software* de referência *HEVC Test Model* 16.16 (HM-16.16) junto com a biblioteca 360Lib 5.0. Na construção do modelo proposto, foi utilizado o módulo *Scikit Learning* que dispõe de diversas ferramentas e algoritmos de ML implementados na linguagem de programação *Python*. Em seguida, após a obtenção dos dados foi realizada uma engenharia de atributos sobre a *Feature* 10 e 11, a fim de facilitar a interpretação das mesmas na etapa de treinamento, pois ambas *Features* consistem em listas que possuem modos de predição sem

qualquer ordem de grandeza entre si. A biblioteca (em *Python*) Pandas foi usada nessa etapa, cujo objetivo foi transformar as listas em um valor que indique um peso sobre a distribuição dos modos DC/Planar nas mesmas. A transformação proposta na lista de MPM (*feature* 10), foi quantificar a ocorrência dos modos DC e Planar, pois quanto maior for a ocorrência dos mesmos na lista, maior a chance de um dos modos ser escolhido como melhor modo. Sendo assim, a *feature* 10 pode assumir valores entre 0 e 2. A seguir, como a lista de RMD (*feature* 11) ordena os modos mais prováveis a partir da primeira posição da lista, o peso é atribuído considerando o menor índice entre os modos desejados subtraído do tamanho da lista, gerando um valor inverso à grandeza original. Por fim, o *Label* do modelo foi atribuído como um valor booleano, sendo Verdadeiro caso o melhor modo de predição da PU seja DC ou Planar, e Falso caso contrário.

Visando treinar o modelo, o *dataset* foi subdividido em dados de teste, constituindo 20% dos dados, e dados de treinamento, constituindo 80% dos dados. O método de aprendizagem dos dados utilizado foi o *Decision Tree Classifier*, com profundidade máxima de 20 nós e critério de entropia para decisão de divisões dos nós. A fim de avaliar a acurácia do modelo, os métodos de avaliação utilizados foram o *Accuracy score* e o *Cross Validation Score* (K-FOLD) de 20 partições. Tendo em vista que o HM-16.16 utiliza a linguagem de programação C++, foi utilizada a biblioteca M2CGEN para transpilar o modelo treinado de *Python* para a linguagem do codificador.

Considerando que a técnica proposta está em seu estágio inicial, os testes experimentais visam analisar o impacto das *Features* escolhidas na acurácia do modelo treinado e a relevância das mesmas no processo de aprendizado, além de observar o impacto das ferramentas e configurações utilizadas sobre o tempo de codificação de vídeo. Sendo assim, a avaliação da performance do modelo DC/Planar se deu sobre a comparação de tempo entre a codificação Intra-quadro dos 5 primeiros quadros do vídeo *Aerial City* utilizando o HM-16.16 com e sem o modelo proposto implementado.

3. RESULTADOS E DISCUSSÃO

Seguindo a metodologia descrita anteriormente, ao realizar os testes de acurácia do modelo treinado, foi observado que o método *Accuracy score* obteve 95,38% de acurácia, enquanto o método K-FOLD obteve 84,24%. Tal discrepância foi observada, pois o método *Accuracy Score* utiliza o *dataset* na forma como foi distribuído (20% dados de teste e 80% dados de treinamento), o que pode gerar uma divisão muito otimista ou pessimista dos dados resultando em acurácias tendenciosas. Por outro lado, o K-FOLD contorna esse problema dividindo o *dataset* em K sub-partições pré-determinadas, visando “embaralhar” os dados enquanto segue a mesma distribuição de dados. Sendo assim, levamos em conta a acurácia obtida com o K-FOLD, visto que representa um valor mais adequado para cenários reais. Seguindo, a *Figura 1* apresenta um gráfico que mostra a relevância das *Features* para o processo de treinamento, onde é possível ver que quatro *Features* apresentaram valor de relevância superior a 10%, sendo elas o RD-Cost do melhor modo (4) com 21%, o peso da lista de MPM (10) com 10,58%, o peso da lista de RMD (11) com 49,72% e o menor RD-Cost dos modos DC/Planar (12) com 15,68%. Portanto, visto que o restante das *Features* beirou 1% de relevância, o gráfico indica um bom aproveitamento de *Features* que fazem uso de valores de RD-Cost. Por fim, fazendo uso do modelo DC/Planar no HM-16.16 com as configurações propostas, resultou em 146,89

segundos de codificação, enquanto sem o modelo foram 125,7 segundos. Portanto, inicialmente o uso do modelo gerou um aumento de 16,85% no tempo de codificação, cujo resultado não pode ser explicado até o momento, pois requer uma análise mais detalhada sobre toda implementação e configuração da técnica.

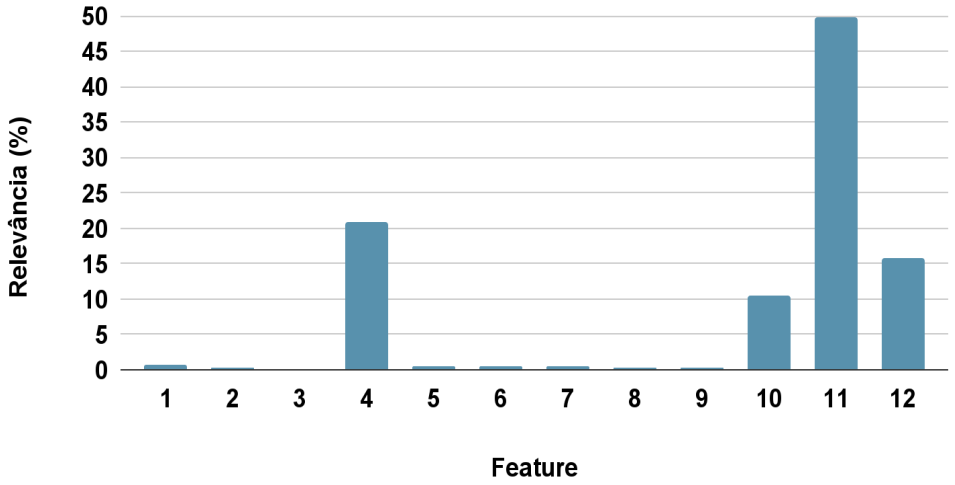

Figura 1. Gráfico de relevância das Features propostas

4. CONCLUSÕES

Apesar dos resultados preliminares apontarem um aumento no tempo de codificação quando o modelo é utilizado, era de objetivo dos autores ter um contato introdutório com a implementação da técnica, possibilitando investigações e aperfeiçoamentos da mesma. Por outro lado, foram obtidos valores satisfatórios de acurácia e relevância com as *Features* propostas, o que indica um bom processo de engenharia de atributos e também indica o potencial da aplicação de outros modelos na técnica. Vale ressaltar que, quando concluída, esta será a primeira técnica na literatura a explorar tanto o particionamento de tamanhos de bloco quanto a escolha de modos de predição na predição Intra-quadro do padrão HEVC.

5. REFERÊNCIAS BIBLIOGRÁFICAS

ACCESSIBLETOURISM, **COVID-19 and opportunities for VR based tourism economy.** 15 jun. 2020. Acessado em 4 jul. 2021. Online. Disponível em: https://www.accessibletourism.org/?i=enat.en.news.2176

YE, Y.; ALSHINA, E.; BOYCE, J. JVET-E1003: Algorithm descriptions of projection format conversion and video quality metrics in 360Lib. In: **Joint Video Exploration Team ITU-T SG16 WP3 ISO/IECJTC 1/SC 29/WG 11 5th Meet**, Geneva, 2017.

STORCH, I.; CRUZ, L.; AGOSTINI, L.; ZATT, B.; PALOMINO, D. The Impacts of Equirectangular 360-degrees Videos in the Intra-Frame Prediction of HEVC. **Journal of Integrated Circuits and Systems**, Online, v.14, n.1, 2019

BELING, B.; STORCH, I.; AGOSTINI, L.; ZATT, B.; BAMPI, S.; PALOMINO, D. ERP-Based CTU Splitting Early Termination for Intra Prediction of 360 videos, ***2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)***, 2020, pp. 359-362.