

Redução de Complexidade da Codificação de Nuvens de Pontos Dinâmicas no V-PCC com Uso de Aprendizado de Máquina

Eduardo de Figueiredo Costa; Gustavo Hübner Rehbein; Cristiano Flores dos Santos; Marcelo Schiavon Porto

*Universidade Federal de Pelotas - Video Technology Research Group
{edfcosta, ghrehbein, cfdsantos, porto}@inf.ufpel.edu.br*

1. INTRODUÇÃO

Com o desenvolvimento de novas tecnologias de aquisição e geração de modelos digitais 3D, a acessibilidade do público a essas tecnologias vêm aumentando, assim como a demanda por este tipo de conteúdo. Dentre as diferentes formas de representação da informação 3D, se destacam as nuvens de pontos. De forma geral, uma nuvem de pontos é formada por um conjunto de pontos no espaço tridimensional que representa uma cena ou objeto do mundo real. A cada ponto também podem ser atribuídas informações de aparência, como informações de cor (SANTOS, 2020).

Neste sentido, para tratar o volume de dados gerados por uma nuvem de pontos, em especial por uma nuvem do tipo dinâmica, foi proposto pelo Moving Pictures Experts Group (MPEG) o padrão Video-based Point Cloud Compression (V-PCC). Esse padrão propõe um fluxo de compressão de nuvens de pontos dinâmicas que utiliza um compressor de vídeos 2D, para isso, é necessária a planificação das informações 3D das nuvens de pontos. O processo prevê a geração de três subfluxos de vídeos 2D independentes: um subfluxo para as informações de geometria, um subfluxo para as informações de atributos, e um subfluxo para o mapa de ocupação. Os subfluxos de vídeos gerados pelas etapas de codificação do V-PCC são enviados para um codificador de vídeo 2D, o *High Efficiency Video Encoding* (HEVC), que possui o software de referência HM, que codifica cada um dos três fluxos de vídeo individualmente (PREDA, 2023).

Esta pesquisa tem como proposta o emprego de um modelo de aprendizado supervisionado de classificação que terá como objetivo redução de complexidade da etapa de particionamento de blocos no software HM, promovendo a interrupção antecipada do particionamento de blocos utilizando nuvens de pontos dinâmicas e o software de referência *Test Model Category 2* (TMC2) do padrão (V-PCC).

2. METODOLOGIA

A criação do dataset de treinamento do modelo conta com características de nuvens codificadas com a configuração *Random Access*, e todas suas configurações de taxa alvo, que vão de r1 à r5. Cada uma destas configurações foram utilizadas para nuvens de 11 bits. O rótulo de cada exemplo coletado para o dataset, indicará se o bloco foi selecionado como o melhor candidato dentre os blocos disponíveis (skip ou 1) ou não (no-skip ou 0).

Para cada substream, modelos especializados foram treinados considerando cada profundidade onde há possibilidade de decisão, ou seja, para os tamanhos de bloco 64x64, 32x32 e 16x16. Esta redução de tempo de codificação é feita somente para o subfluxo de geometria e de atributos de cor. O TMC2 na versão 22.1 foi utilizado nestes experimentos. O HM na versão 16.20 (ZAKHARCHENKO, 2019) foi utilizado para a compressão destes subfluxos de

ocupação, geometria e atributos. As características para a construção do modelo de aprendizado de máquina são as mesmas utilizadas em (CORREA, 2014).

Cada modelo foi treinado utilizando a linguagem Python e o algoritmo *DecisionTreeClassifier* contido na biblioteca *Scikit-Learn* (PEDREGOSA et al. 2011). Estes modelos foram avaliados utilizando a métrica *f1-score*. Foram adotados os hiperparâmetros de profundidade da árvore em 10 níveis e máximo de 1000 nós folha como base para comparação do modelo com busca de hiperparâmetros. Após isso, uma busca por hiperparâmetros foi realizada, a fim de encontrar hiperparâmetros mais otimizados. Foram feitas duas buscas aleatórias com o método *RandomizedSearchCV* (PEDREGOSA et al., 2011). O espaço de busca utilizado é apresentado na Tabela I. Adicionalmente, foi realizada uma segunda busca ajustando o espaço de busca, conforme detalhado na Tabela II.

Tabela 1 - Primeiro espaço de busca.

	Hyperparâmetros	Espaço de Busca	Total de Valores
a	Criterion	{'gini', 'entropy'}	2
b	min_samples_split	[25, 10025, passo 25]	400
c	min_samples_leaf	[1, 100, passo 10]	10
d	max_depth	[2, 18, passo 1]	16
e	max_leaf_nodes	[25, 1000, passo 25]	39
f	max_features	[1, 12, passo 1]	11
Total de Combinações			54.912.000

Tabela II - Segundo espaço de busca.

	Hyperparâmetros	Espaço de Busca	Total de Valores
a	Criterion	{'gini', 'entropy'}	2
b	min_samples_split	[25, 10125, passo 100]	101
c	min_samples_leaf	[1, 1050, passo 50]	21
d	max_depth	[2, 15, passo 1]	13
e	max_leaf_nodes	[25, 2075, passo 50]	41
f	max_features	[1, 12, passo 1]	11
Total de combinações			24.870.846

Tabela III - Hiperparâmetros utilizados para treinamento para cada *dataset*.

Dataset	a	b	c	d	e	f	F1-score
geometry_16x16	125	11	325	11	16	entropy	0,92
geometry_32x32	--	--	1000	--	10	--	0,87
geometry_64x64	--	--	1000	--	10	--	0,83
attribute_16x16	725	101	1625	11	10	entropy	0,92
attribute_32x32	425	31	625	8	11	gini	0,89
attribute_64x64	150	61	375	5	14	gini	0,89

A Tabela III contém a seleção de hiperparâmetros utilizados para treinamento individual de cada um dos modelos. A avaliação para seleção de um novo conjunto de hiperparâmetros seguiu como critério a alteração de *F1-score* e

número de nós. No entanto, se houve uma redução significativa no *F1-score*, ou manteve o *F1-score* aumentando a quantidade de nós na árvore de decisão, foram mantidos os hiperparâmetros base, sendo eles 10 de profundidade máxima e máximo de 1000 nós folhas, como é o caso de geometria 32x32 e 64x64.

3.RESULTADOS E DISCUSSÃO

Nesta etapa foram realizadas duas rodadas de codificação. A primeira rodada foi feita utilizando o TMC2, chamado de V-PCC-Base. No V-PCC-Base é utilizado o software HM original, ou seja, sem o modelo de aprendizado de máquina implementado. Este experimento de codificação foi feito para coleta de tempo de codificação e eficiência de compressão. Na segunda rodada de experimentos, foi utilizada uma implementação modificada do TMC2, onde foram usados apenas os modelos treinados com as configurações base. Esta versão será chamada de V-PCC-Treinado. Após este experimento, também foram realizados experimentos com os modelos treinados com ajustes de hiperparâmetros para avaliar o impacto dos ajustes, esta versão será chamada de V-PCC-Ajustado.

Nesta pesquisa, a eficiência de compressão do TMC2 foi avaliado pela métrica BD-Rate. As codificações foram feitas com as configurações de taxa de bits (r1 à r5). A métrica BD-Rate utiliza o PSNR resultante das métricas de avaliação de geometria, D1 e D2 e de atributos de cor, Y, Cb e Cr (WANG et al., 2024). A Tabela IV apresenta os resultados de BD-Rate para o V-PCC-Treinado e V-PCC-Ajustado. O V-PCC-Ajustado apresenta resultados muito melhores em relação ao V-PCC-Treinado, reduzindo a perda de eficiência de compressão das nuvens em 5,83 e 5,44 pontos percentuais para geometria, e para a luminância, o modelo com ajustes foi 2,41 pontos percentuais mais eficiente.

Tabela IV - BD-Rate do V-PCC Treinado e Ajustado para as sequências de nuvens de pontos.

V-PCC-Treinado					
Sequência	D1(%)	D2(%)	Y(%)	Cb(%)	Cr(%)
basketball	5,65	5,98	3,25	-1,44	1,59
dancer	6,25	6,49	2,3	-1,54	2,45
exercise	4,59	4,85	1,86	1,14	-0,88
model	5,02	5,45	4,82	0,37	1,56
Média	5,38	5,69	3,06	-0,37	1,18
V-PCC-Ajustado					
Sequência	D1(%)	D2(%)	Y(%)	Cb(%)	Cr(%)
basketball	0,21	0,15	0,84	-0,09	-0,94
dancer	0,26	0,25	0,51	-2,24	1,07
exercise	0,11	0,07	0,46	2,48	1,03
model	0,36	0,36	1,25	-0,27	-0,04
Média	0,24	0,21	0,77	-0,03	0,28

Os resultados para geometria e atributos, são apresentados em Tabela V, indicando a redução do tempo de codificação dos 32 quadros codificados para as quatro sequências utilizadas na avaliação do modelo implantado, onde cada sequência possui a média das cinco configurações de taxa de bits, apresentando estes resultados em porcentagem de redução de tempo de codificação. Os dois modelos preditivos de árvore de decisão implantados no TMC2 apresentaram um

desempenho satisfatório para a redução de tempo na codificação, apresentando um comportamento decrescente de redução de tempo, conforme a taxa de bits aumenta. O V-PCC-Treinado obteve a maior redução de tempo de codificação tanto para geometria quanto para atributos em comparação ao V-PCC-Ajustado.

Tabela V - Redução de tempo de codificação de geometria e de atributo do V-PCC-Treinado e V-PCC-Ajustado.

Sequência	Treinado		Ajustado	
	Geom (%)	Atrib (%)	Geom (%)	Atrib (%)
basketball	37,21	47,79	29,5	54,77
dancer	38,5	42,31	27,45	40,37
exercise	40,44	53,23	32,01	50,75
model	39,24	38,74	29,32	34,87
Média	38,85	45,51	29,57	45,19

4. CONCLUSÕES

Esta pesquisa apresenta uma solução para redução do tempo de codificação na compressão de nuvens de pontos dinâmicas utilizando árvores de decisão. Um *dataset* foi criado com dados extraídos de nuvens de pontos para o treinamento de modelos preditivos com o objetivo de auxiliar na decisão do particionamento de blocos das etapas de codificação de vídeo do V-PCC. Os resultados obtidos demonstram uma redução média de 29,57% para substreams de geometria e 45,19% para substreams de atributo em relação ao V-PCC-Base, com um impacto na eficiência de codificação de 0,21% para geometria e 0,77% para atributos.

Os modelos com ajustes de hiperparâmetros foram capazes de ajustar o algoritmo, obtendo resultados finais no software de referência amplamente superiores. Indicando uma melhor generalização dos dados e árvores de decisão mais eficientes. Esses resultados destacam a viabilidade e o impacto substancial do uso de aprendizado de máquina para otimizar processos de compressão de dados em cenários de alto desempenho utilizando árvores de decisão, que possuem baixo custo computacional e resolvem o problema apresentado de maneira eficiente e a importância dos hiperparâmetros para ajuste do algoritmo.

5. REFERÊNCIAS BIBLIOGRÁFICAS

SANTOS, C. F. dos. Compressão de nuvens de pontos dinâmicas: uma abordagem eficiente para a etapa de predição. 2020. 152p. Tese de Doutorado (Curso de Ciência da Computação) — Universidade Federal de Pelotas, Pelotas.

PREDÁ, M. Common Test Conditions for V3C and V-PCC. [S.l.]: ISO/IEC JTC 1/SC 29/WG 11, 2023.

ZAKHARCHENKO, V. V-PCC codec description. Document ISO/IEC JTC1/SC29/WG11 N, [S.l.], v.18487, 2019.

Corrêa, Guilherme Ribeiro. Computational Complexity Reduction and Scaling for High Efficiency Video Encoders. Diss. Universidade de Coimbra (Portugal), 2014.

Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.

Wang, Yihan, et al. "Fast Video-Based Point Cloud Compression Based on Early Termination and Transformer Model." IEEE Transactions on Emerging Topics in Computational Intelligence (2024).