

# LIGHT GRADIENT BOOST MACHINE (LIGHTGBM): UM ALGORITMO RÁPIDO DE PREVISÃO DE PARTIÇÃO INTRA QT DO VVC

*Samuel Freitas Oliveira*  
Universidade Federal de Pelotas (UFPel)  
Video Technology Research Group (ViTech)  
Pelotas, Brasil  
sfoliveira@inf.ufpel.edu.br

## ABSTRACT

For the development of VVC a series of new video techniques was adopted, such as quad-tree plus multi-type tree (QTMT) of the step unit (CU) partition. efficiency in the evaluation over its predecessor the H.265 being up to 50% more efficient along with the efficiency gain increased drastically being measured the complexity can be verified by measuring through the VTM Reference software a non-intra-mode VVC derivative complexity is in average 18 times longer than HEVC, reaching in some cases up to 27 times longer than H.265. According to the QTMT structure is responsible for more than 97% of the obtained time, since the CTU in the H.265 is 64x64 whereas in VVC it is extended to 128x128 In this paper, a Machine Learning Support Vector Machine complex model is proposed as a solution to minimize the complexity of the QTMT based on the Coding Unit (CU) to predict the direction of the CU division focused on whether it is QT or not using the LumaQT dataset which it has. features of the current CU to train the model. Experimental results have not yet been obtained.

**Index Terms**— VVC, SVM, QTMT, Data Mining, Machine Learning

## 1. INTRODUÇÃO

A transmissão de vídeos e streaming atualmente são responsáveis por mais de 70% do tráfego na internet [1]. Segundo o relatório da Cisco, o impacto dos vídeos no

consumo de banda é alto devido à introdução do streaming de vídeo em Ultra-Alta Definição(UHD) ou 4K. Essa tecnologia tem esse efeito porque a taxa de bits para vídeos 4K consome entre 15 a 18 Mbps é mais que o dobro da taxa de bits de vídeo HD e nove vezes mais que a taxa de bits de vídeo de definição padrão (SD) [2].

O consumo de vídeos demanda muita banda, especialmente agora durante a pandemia do COVID-19, onde foi observado o crescente tráfego de dados usado para videoconferência, jogos e aprendizado online forçando plataformas como a Netflix e o YouTube a limitar a qualidade do vídeo na Europa [4]; para diminuir o consumo de banda é necessário a utilização de algoritmos para reduzir a quantidade de bits necessárias para a representação de informações visuais, a área responsável por isso é denominada de codificação de vídeos.

Com o a demanda áudio visual cada vez mais exigente, especialmente o surgimento da tecnologia de vídeo Ultra-High Definition (UHD), a quantidade de dados tremendamente crescente de vídeo UHD sobrecarrega o sistema de armazenamento e transmissão.

Porém segundo a CISCO a demanda por compressão de vídeo de alto desempenho tende a continuar o crescendo. Como visto acima o aumento de largura de banda e armazenamento no futuro cada vez mais demandará dos padrões atuais mais adotados, como o VP9 e o HEVC projetados em 2013 com a necessidade de técnicas de compressão de vídeo com melhor compressão eficiência [3].

Após a detecção da necessidade de um novo padrão de compressão de vídeo que suprisse as limitações que o HEVC não é capaz de lidar VVC com o objetivo de obter um desempenho significativo melhor em relação ao seu antecessor. Em 2015 foi formada uma equipe com a ITU-T VCEG e ISO / IEC MPEG dessa fusão nasceu o grupo Joint Video Exploration Team (JVET ) proposto a explorar métodos de melhorar a eficiência da codificação após estudos iniciais se mostrarem promissores (estudos demonstram até 30%, redução da taxa de bits em comparação com HEVC). Após os resultados obtidos pela JVET mostrarem promissores foi feita uma chamada Conjunta formal de Propostas (CfP ) , a JVET então foi

renomeada para Equipe Conjunta de Especialistas em Vídeo (também abreviado para JVET) com a tarefa de desenvolver um novo padrão de codificação de vídeo em 2018 depois da CfP bem sucedida onde todas as submissões feitas por mais de 30 organizações da qual todas obtiveram resultados subjetivos melhores que o HEVC nas três categorias de conteúdo de vídeo: alcance dinâmico padrão (SDR), HDR e vídeo 360°. Em 2018 ainda o novo padrão de codificação de vídeo começa a ser desenvolvido cujo qual, foi denominado Versatile Video Coding (VVC) sendo finalizado em junho de 2020. É registrado como ITU-T Recommendation H.266 | ISO / IEC 23090 -3 [4]. O JVC começou a considerar padrão de codificação de vídeo que suprisse as necessidades do HEVC. Ainda em 2015 foi proposto o VVC com o objetivo de obter um desempenho significativo melhor em relação ao seu antecessor o HEVC. O VVC é considerado o próximo padrão de codificação de vídeo [5].

O padrão de compressão de vídeo VVC foi projetado para transmitir e armazenar vídeo digital com eficiência com uma variedade de requisitos em taxa de bits, qualidade de imagem, atraso, acessibilidade aleatória, complexidade e assim por diante obtendo um alto desempenho na compressão de vídeos não somente para as tecnologias atuais, mas sim, para todas as tecnologias emergentes relacionadas a vídeos e aplicativos como: vídeos com Ultra Alta Definição, sendo capaz de lidar com resoluções 4k, 8k ou até mesmo superiores, alta faixa dinâmica (HDR) e ampla gama de cores, aplicativos de compartilhamento de tela de computadores e jogos online, vídeo 360° para realidade imersiva e aumentada, streaming, serviços de conversação em tempo real, transmissão ao vivo [5].

Para o desenvolvimento do VVC foi adotada uma série de novas técnicas de codificação de vídeo como o quad-tree plus multi-type tree (QTMT) da partição da unidade de codificação (CU), a intra-predição dependente da posição, o movimento afim previsão de compensação e assim por diante. Com a introdução de todas essas técnicas no VVC obtiveram ótimos resultados de eficiência na codificação sobre seu antecessor o H.265 sendo até 50% mais eficiente, porém junto com o ganho da eficiência aumentou dramaticamente sendo medida a complexidade podendo ser constatada medindo através do software de Referência. VTM a complexidade de codificação de VVC no intra-modo é em média 18 vezes maior do que o HEVC chegando em alguns casos até a ser 27 vezes mais demorado que o H.265. Segundo [6] a estrutura QTMT é responsável por mais de 97% do tempo de codificação, já que a CTU no H.265 é 64x64 já no VVC é estendido para 128x128 [6][7]. Neste artigo é proposto como solução um modelo de machine learning Support Vector Machine visando minimizar a

complexidade do QTMT baseado na Coding Unit (CU) para prever a direção da divisão CU focado se ela é Quad-Tree (QT) ou não usando o dataset LumaQT transformado-o em um problema de classificação.

## 2. CONCEITOS BÁSICOS

O VVC introduz uma série de novas técnicas, como a nova estrutura de Unidade de Codificação quadtree com árvore multi-tipo aninhada (QTMT) estrutura de codificação, cuja qual, apresenta novos tamanhos de blocos e modos de divisão tornando o QTMT estrutura de codificação a técnica mais eficaz, porém é também extremamente demorada, devido sua alta complexidade [8] [9].

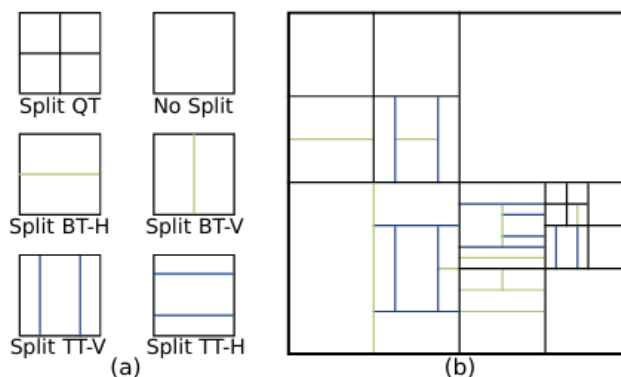


Figura 1: Coding Tree Unit (CTU) particionamento no VVC (a) Tipos de divisão do VVC (b) Exemplo de particionamento no VVC

A Figura 1 (a) mostrada acima introduz as divisões suportadas pelo VVC, na qual existem quatro tipos de divisão na estrutura do QTMT, Binary Tree Horizontal (BT-H) e Binary Tree Vertical (BT-V), Ternary Tree Horizontal (TT-H) e Ternary Tree Vertical (TT-V) [3,5] [10][12]. Como Quad-Tree (QT) cujo o qual divide a CU em quatro sub-CU iguais. O VVC diferentemente do HEVC permite realizar a divisão de forma retangular para CU com Binary-Tree (BT) e Ternary-Tree (TT). O BT divide uma CU em duas sub-CU enquanto o TT divide uma CU em três sub-CU com a proporção 1: 2: 1 Tanto o BT e TT podem dividir uma CU horizontal ou verticalmente [4][11]. Já a Figura. 1 (b) apresenta as divisões de uma CTU processada por VTM Rate-Distortion Optimization (RDO).

O processo RDO seleciona a partição CTU cuja qual gera o menor custo (RD) para cada CU através de uma busca exaustiva onde testa todas possibilidades possíveis (exceto as limitações impostas pelo VTM i.e. quando a partição é BT ou TT é selecionada em um CU, os modos de partição QT não é mais permitido [10][11]) e calcula o custo

da Taxa de Distorção (RD) para cada CU procurando minimizar a função de custo.[4, 3][10][11]. Na configuração, AI, o VTM força a primeira divisão da CTU a ser um QT. O consumo de tempo da estrutura QTMT ultrapassa 90%, e o consumo de tempo médio de todas as classes testadas é de 93,84%. A parte mais demorada no VVC é a estrutura do QTMT nos testes realizados em [8]. Portanto, para diminuir a complexidade do processo de codificação, o processo de QTMT deve ser otimizado. Na estrutura de codificação QTMT, a partição CU possui mais modos de divisão o que gera melhores resultados, porém, a complexidade da codificação é bastante aumentada devido à otimização da distorção da taxa (RDO)[1]. Então é necessário otimizar o processo QTMT de maneira a manter o desempenho de taxa de distorção de taxar(RD) enquanto tenta diminuir a complexidade do VVC o que se torna um desafio imenso.

Agora uma rápida introdução a uma máquina de vetor de suporte (SVM) é um dos algoritmos clássicos de machine learning de aprendizado supervisionado largamente usado para resolver problemas de classificação na maioria dos casos. Por exemplo, um SVM pode aprender a reconhecer atividades fraudulentas de cartão de crédito examinando centenas ou milhares de relatórios de atividades fraudulentas e não fraudulentas de cartão de crédito[17]. Nesse trabalho será usado para determinar se a divisão será QT ou não QT.

### 3. TRABALHOS RELACIONADOS

Como visto o QTMT é um problema que possui uma alta complexidade o que o torna um grande gargalo para a adoção do padrão VVC atrasando o seu desenvolvimento e implantação, principalmente em plataformas embarcadas e streaming. Visando reduzir a complexidade computacional dos codificadores, diversas técnicas se propõem a reduzir os candidatos intra-modo testados, ou técnicas de Machine learning (ML) para prever um conjunto reduzido de modos intra prováveis. Outras técnicas reduzem a complexidade do processo de codificação, concentrando-se no esquema de particionamento e testando um número reduzido de configurações de partição. [https://ieeexplore.ieee.org/abstract/document/8826595] No [13] como solução é proposto uma decisão rápida de tamanho de CU com base em textura e um algoritmo de decisão intra-modo. Comparado com o software de referência VVC mais recente VTM-4.2, o algoritmo proposto pode atingir aproximadamente 46% de economia de tempo de codificação em média com apenas 0,91% de aumento de BD-RATE ou 0,046 de redução de BD-PSNR.

Já o autor [14] visando reduzir a complexidade de VVC, desenvolveu um algoritmo de decisão de tamanho de (CU) baseado SVM. Os resultados experimentais mostram

que o algoritmo proposto pode economizar significativamente o tempo de codificação em 51,01% com um ligeiro aumento da taxa de bits delta Bjontegaard.

A introdução do QTMT traz desempenho de codificação superior ao custo de um grande consumo de tempo. No artigo [15] é desenvolvido um algoritmo de intrapartição rápido baseado em variância e operador de Sobel. Quando detectado uma textura considerada suave é encerrada a divisão posterior da CU. O operador Sobel é usado para extrair recursos de gradiente para decidir se deve dividir esta, CU por QT, encerrando assim outras partições MT.

Logo após é aplicado um novo método para selecionar apenas uma partição de cinco partições QTMT. Comparando com a âncora VTM-7.0, o método economiza o tempo de codificação em 49,27% em média ao custo de apenas 1,63% de aumento de BDBR. No [16] os autores propõem um algoritmo de intrapartição rápido baseado em variância e gradiente para resolver o problema de partição retangular em VVC. Primeiro, a divisão adicional de áreas lisas é encerrada. Então, a partição QT é escolhida dependendo dos recursos de gradiente extraídos pelo operador Sobel. Finalmente, uma partição de cinco partições QTMT possíveis é escolhida diretamente pelo cálculo da variância de sub-CU. Os Resultados experimentais mostram que o método desenvolvido economizar em média 53,17% do tempo de codificação com apenas 1,62% de aumento de BDBR e 0,09 dB de perda de BDPSNR em comparação com a âncora VTM4.0.

### 4. DESENVOLVIMENTO

O trabalho desenvolvido pode ser dividido em três partes ou fases cujas as quais foram as seguintes:

- 1 Coleta de dados
- 2 Desenvolvimento do SVM
- 3 Treinamento do Modelo

Na 1 fase foi necessário a coleta de todos os dados para gerar o dataset “LumaQT” para isso foi extraída um conjunto de 1021668 linhas com 20 features da atual CU dos vídeos de referências cujo o qual foi gentilmente cedido pelo pesquisador Mario saldanha.

O dataset possui 20 features cujas as quais são as seguintes: 'qp', 'width', 'currCost', 'currDistortion', 'currIntraMode', 'var', 'maxVarQT', 'diffVarQT', 'diff\_var\_hor', 'diff\_var\_ver', 'Gx', 'Gy', 'ratioGxGy', 'norm\_gradient', 'ispMode', 'avg\_neigh\_QT', 'neigh\_higher\_QT', 'avg\_neigh\_MTT', 'neigh\_higher\_MTT', 'isQT'.

O dataset LumaQT segundo o pesquisador Mário Saldanha: “Possui informações de textura global consideram recursos relacionados às amostras de CU atuais, como largura e altura da CU atual, área, proporção de bloco,

variância (var), gradientes horizontais (Gx) e verticais (Gy) com base no operador Sobel. Gx dividido por Gy (razão GxGy), e a soma de Gx e Gy dividida pela área do bloco (gradiente da norma).

As informações de textura local referem-se a recursos obtidos em regiões menores da CU atual, como a diferença absoluta de variâncias em quatro sub-trimestres (diffVarQT), variância máxima em quatro sub-trimestres (maxVarQT), a diferença absoluta entre as variações de superior e inferior regiões da CU (diffVarHor), e a diferença absoluta entre as regiões esquerda e direita da UC (diffVarVer).

As informações de contexto incluem recursos de CU vizinhos à esquerda, acima, à esquerda e acima à direita, como QT médio (neighAvgQT) e níveis de profundidade de MTT (neighAvgMTT) em CU vizinhos e número de CU vizinhos com QT (neighHigherQT) e MTT (neighHigherMTT) níveis de profundidade superiores à CU atual.

Uma vez que nenhuma divisão é avaliada primeiro do que as divisões QT, BT e TT, podemos considerar vários atributos de codificação obtidos com o tamanho da CU atual para decidir os tipos de divisão. As informações de codificação compreendem atributos de codificação relacionados à CU atual avaliada sem tipo de divisão, como QP, custo RD (currCost), distorção (currDistortion), corrente QT (QTD), BT (BTD), MTT (MTTD) e QTMT (QTMTD) níveis de profundidade, melhor modo de predição intra (currIntraMode). Índice MRL (mrlIdx). Índice LFNST (lfnstIdx), modo ISP (ispMode) e sinalizador MTS (mtsFlag) [artigo ainda não publicado].

A segunda fase se concentrou em desenvolver o modelo de machine learning SVM na linguagem de C++ usando a biblioteca Libsvm como entrada do algoritmo todas as features do dataset “LumaQT” exceto a última coluna cuja qual determina se é QT ou não foram usadas. O algoritmo rápido SVM intra-frame VVC é baseado na versão VTM 13.0. Para os testes foram usadas as 4 sequências de vídeos eles são: Cactus, RaceHorseC, BasketballPass, FourPeople; todos os testes são realizados de acordo com JVET. O desempenho do modelo desenvolvido é medido por Bjontegaard Delta Bit Rate (BDBR), Bjontegaard Delta PSNR e economia de tempo são calculados usando o original plataforma VTM como benchmark,

## 5. RESULTADOS

O algoritmo proposto baseado no Lightgbm aplicado no VVC um algoritmo rápido de previsão de partição intra QT no VVC. Todas as sequências de teste foram testadas de

acordo com a documentação disponibilizada pelo JEV.T. O Lightgbm foi desenvolvido em python usando a API do SciKit-Learn e pandas. O modelo foi treinado com o dataset ‘Luma\_QT’ cujo o qual foi dividido em dois conjuntos o treino e teste, cujo o treino possui 766251 e o teste 255417 obtendo uma precisão de 80.40%. amostras, posteriormente o modelo treinado foi convertido e adaptado para C++ usando a ferramenta m2cgen 0.9.0 para ser integrado ao VTM o código se encontra disponível em: <https://github.com/freliver/PVC> 3. A plataforma de teste foi o sistema operacional Ubuntu 18.04.4 com 16GB de memória e processador AMD Ryzen 5 3400G 3.7GHz. No VTM os testes foram feitos sobre os 3 vídeos disponibilizados pela disciplina: BasketballPass, FourPeople e o RaceHorseC cada um com 25 fps.

Os resultados obtidos foram no basketball bitrate 108.0160 e PSNR 34.0254. O FourPeople bitrate 967.3920 e PSNR 39.0235 e o HorseRaceC bit rate 482.7840 e PSNR 31.4166.

Os resultados aqui obtidos foram de cunho inicial para realizar previsão da direção da partição de CU através das features extraídas do atual CU para determinar se é QT ou não porém parece promissor. Como trabalho futuro a finalização da implementação do SVM no Libsvm realizando os testes posteriormente nos servidores do GACI, atualizando a literatura com resultados atuais baseados no VTM 13.0 cujo qual pretendesse obter resultados promissores.

## REFERÊNCIAS

- [1] [https://www.cisco.com/c/dam/en\\_us/about/annual-report/2017-annual-report-full.PDF](https://www.cisco.com/c/dam/en_us/about/annual-report/2017-annual-report-full.PDF)
- [2] Y. Wang et al., "The High-Level Syntax of the Versatile Video Coding (VVC) Standard", *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1-1, 2021. Available: 10.1109/tcsvt.2021.3070860 [Accessed 20 June 2021].
- [3] B. Bross, J. Chen, J. Ohm, G. Sullivan and Y. Wang, "Developments in International Video Coding Standardization After AVC, With an Overview of Versatile Video Coding (VVC)", *Proceedings of the IEEE*, pp. 1-31, 2021. Available: 10.1109/jproc.2020.3043399 [Accessed 19 June 2021].
- [4] F. JCT-VC, "Versatile Video Coding (VVC) | JVET", *Jvet.hhi.fraunhofer.de*, 2021. [Online]. Available: <https://jvet.hhi.fraunhofer.de/>. [Accessed: 19- June- 2021].
- [5] A.B. Smith, C.D. Jones, and E.F. Roberts, "Article Title," *Journal*, Publisher, Location, pp. 1-10, Date.
- [6] T. Li, M. Xu, R. Tang, Y. Chen and Q. Xing, "DeepQTMT: A Deep Learning Approach for Fast QTMT-Based CU Partition of

- Intra-Mode VVC", *IEEE Transactions on Image Processing*, vol. 30, pp. 5377-5390, 2021. Available: 10.1109/tip.2021.3083447 [Accessed 19 June 2021].
- [7] G. Tang, M. Jing, X. Zeng and Y. Fan, "Adaptive CU Split Decision with Pooling-variable CNN for VVC Intra Encoding", *2019 IEEE Visual Communications and Image Processing (VCIP)*, 2019. Available: 10.1109/vcip47243.2019.8965679 [Accessed 19 June 2021].
- [8] T. Amestoy, A. Mercat, W. Hamidouche, C. Bergeron and D. Menard, "Random Forest Oriented Fast QTBT Frame Partitioning", *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019. Available: 10.1109/icassp.2019.8683413 [Accessed 15 June 2021].
- [9] T. Amestoy, A. Mercat, W. Hamidouche, D. Menard and C. Bergeron, "Tunable VVC Frame Partitioning Based on Lightweight Machine Learning", *IEEE Transactions on Image Processing*, vol. 29, pp. 1313-1328, 2020. Available: 10.1109/tip.2019.2938670 [Accessed 15 June 2021].
- [10] A. Tissier, W. Hamidouche, J. Vanne, F. Galpin and D. Menard, "CNN Oriented Complexity Reduction Of VVC Intra Encoder", *2020 IEEE International Conference on Image Processing (ICIP)*, 2020. Available: 10.1109/icip40778.2020.9190797 [Accessed 10 June 2021].
- [11] F. Chen, Y. Ren, Z. Peng, G. Jiang and X. Cui, "A fast CU size decision algorithm for VVC intra prediction based on support vector machine", *Multimedia Tools and Applications*, vol. 79, no. 37-38, pp. 27923-27939, 2020. Available: 10.1007/s11042-020-09401-8 [Accessed 20 June 2021].
- [12] A. Tissier, W. Hamidouche, J. Vanne, F. Galpin and D. Menard, "CNN Oriented Complexity Reduction Of VVC Intra Encoder", *2020 IEEE International Conference on Image Processing (ICIP)*, 2020. Available: 10.1109/icip40778.2020.9190797 [Accessed 15 June 2021].
- [12] Jones, C.D., A.B. Smith, and E.F. Roberts, *Book Title*, Publisher, Location, Date.
- [13] J. Chen, H. Sun, J. Katto, X. Zeng and Y. Fan, "Fast QTMT Partition Decision Algorithm in VVC Intra Coding based on Variance and Gradient", *2019 IEEE Visual Communications and Image Processing (VCIP)*, 2019. Available: 10.1109/vcip47243.2019.8965674 [Accessed 18 June 2021].
- [14] [5]Y. Fan, J. Chen, H. Sun, J. Katto and M. Jing, "A Fast QTMT Partition Decision Strategy for VVC Intra Prediction", *IEEE Access*, vol. 8, pp. 107900-107911, 2020. Available: 10.1109/access.2020.3000565 [Accessed 30 May 2021].
- [15] J. Cao, N. Tang, J. Wang and F. Liang, "Texture-Based Fast CU Size Decision and Intra Mode Decision Algorithm for VVC", *MultiMedia Modeling*, pp. 739-751, 2019. Available: 10.1007/978-3-030-37731-1\_60 [Accessed 10 June 2021].
- [16] F. Chen, Y. Ren, Z. Peng, G. Jiang and X. Cui, "A fast CU size decision algorithm for VVC intra prediction based on support vector machine", *Multimedia Tools and Applications*, vol. 79, no. 37-38, pp. 27923-27939, 2020. Available: 10.1007/s11042-020-09401-8 [Accessed 20 June 2021].
- [17] W. Noble, "What is a support vector machine?", *Nature Biotechnology*, vol. 24, no. 12, pp. 1565-1567, 2006. Available: 10.1038/nbt1206-1565 [Accessed 10 June 2021].