Two-Step Predictive Models for Low-Complexity Multiple Transform Selection in VVC

Caroline Camargo [1], Daniel Palomino [1], Guilherme Correa [1]

[1] Video Technology Research Group (ViTech)
Graduate Program in Computing (PPGC)
Federal University of Pelotas (UFPel) – Pelotas, RS – Brazil

{caroline.sc, dpalomino, gcorrea}@inf.ufpel.edu.br

Abstract. *The Versatile Video Coding (VVC) standard achieves high compression efficiency but with significantly increased computational complexity. A major contributor is the Multiple Transform Selection (MTS) process, which exhaustively evaluates multiple transform combinations. This work proposes a machine learning approach using lightweight decision trees to replace the exhaustive MTS search. The models are trained with balanced sampling, feature selection, and hyperparameter optimization. Experiments in the All Intra configuration show an average encoding time reduction of 18.76% with only a 0.67% increase in BD-rate, reaching up to 30.90% time savings in the transform stage.*

1. Introduction

The Versatile Video Coding (VVC) standard is the successor to High Efficiency Video Coding (HEVC), achieving around 50% bitrate reduction while maintaining similar perceptual quality [Fraunhofer 2020]. These gains are enabled by advanced coding tools for partitioning, prediction, and transform processing, but significantly increase encoder complexity. VVC is estimated to be up to five times more complex than HEVC [Pakdaman et al. 2020], while its transform stage alone can be nine times more complex [Siqueira et al. 2020]. One of the main contributors to this complexity is the Multiple Transform Selection (MTS) mechanism. The MTS tool [Zhao et al. 2021] allows the encoder to select different transform types for horizontal and vertical directions, including DCT-II, DCT-VIII, and DST-VII. In explicit mode, the encoder evaluates multiple transform combinations selecting the one with the lowest rate-distortion cost [Hamidouche et al. 2022]. Although effective for compression, this exhaustive process substantially increases computational cost. This paper proposes an ML-based low-complexity MTS decision method for VVC. The exhaustive transform selection process is replaced by 14 low-complexity decision trees capable of predicting suitable transform candidates and skipping unnecessary evaluations. Experimental results show an average encoding time reduction of 18.76% with only a 0.67% BD-rate increase, demonstrating a favorable trade-off between complexity and coding efficiency.

2. Predictive Models Design

Some previous studies have investigated the behavior of the MTS tool, providing valuable insights into its operation and role in video encoding. In [Camargo et al. 2025c], one of the main analyses concerns the influence of block size on the selection rate of transform combinations. It has been observed that certain combinations, such as DST-VII×DST-VII, are predominantly used in smaller blocks, while their selection frequency decreases as block size increases. This pattern indicates that structural properties of the block, such as width, height, and aspect ratio, directly impact the transform decisions made by the encoder. These

This research was financed by CAPES – Finance Code 001, FAPERGS, and CNPq.

trends reveal complex relationships between block content, spatial structure, and transform efficiency, which can be explored through ML models capable of identifying subtle data correlations. Consequently, designing specialized models for each block size becomes a promising strategy, as each model can better capture the characteristics of its group.

Following this reasoning, the clustering strategy adopted in this work was based on the total block area (*width* $\times$ *height*), expressed as powers of two. This resulted in seven groups – *g16*, *g32*, *g64*, *g128*, *g256*, *g512*, and *g1024* – corresponding to areas of 2^N pixels, where $N \in [4, 10]$. These values correspond to the transform block sizes allowed in VVC. The clustering approach enhances prediction accuracy and reduces decision complexity within the transform module, leading to a more efficient and content-adaptive encoding process. According to the transform usage analyses of [Samir et al. 2024a], DCT-II$\times$DCT-II is the most frequently selected MTS combination, followed by DST-VII$\times$DST-VII, while others, like DCT-VIII$\times$DST-VII, DST-VII$\times$DST-VIII, and DCT-VIII$\times$DCT-VIII, occur much less often. Such insights are particularly valuable for predictive modeling, as they reveal that it is unnecessary to evaluate all possible transform pairs during decision making. By focusing on the most representative transforms, computational cost can be reduced without significantly compromising compression efficiency. However, improperly excluding key combinations such as DCT-II$\times$DCT-II and DST-VII$\times$DST-VII may lead to noticeable efficiency losses.

In the design of the predictive models proposed in this work, these characteristics guided the development of the decision-making mechanism. Rather than employing a single model, a set of models is used – one for each block group (g16–g1024). Moreover, instead of directly selecting a single transform pair, the strategy introduces greater flexibility by defining subsets of candidate pairs to be evaluated during encoding. This approach mitigates BD-rate penalties when a model's prediction is suboptimal. Specifically, two models are defined for each block group (Model A and Model B) to determine the transform pair to be tested. Figure 1 illustrates this process for an $N \times M$ block. When MTS is enabled, the models corresponding to the $N \times M$ block group are employed. Model A first decides whether to apply MTS, since in most cases the DCT-II$\times$DCT-II transform alone is sufficient. If Model A selects multiple transform combinations for testing, Model B then determines which ones should be evaluated – either a reduced list (DCT-II$\times$DCT-II and DST-VII$\times$DST-VII, the most frequently used) or all available combinations, following the standard process.

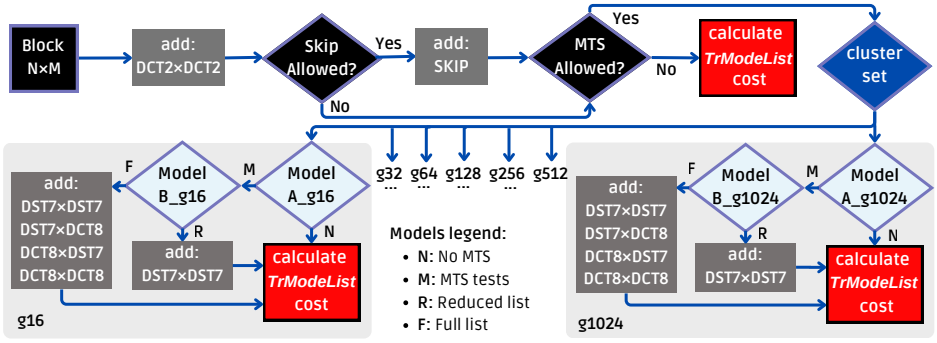

Figure 1. Proposed MTS execution explicit flow with the predictive models.

3. Data Acquisition and Model Training

Data for model training were collected directly from the encoding process, corresponding to the first pipeline phase shown in Fig. 2. For data collection, eight videos were used: *ToddlerFountain* (4K), *RollerCoaster* (4K), *Beauty* (4K), *CrowdRun* (1080p), *ParkScene* (1080p), *YachtRide* (1080p), *Vidyo3* (720p), and *Vidyo4* (720p) [Zhao et al.]. Each video was encoded using the VTM reference software version 23.0 [JVET 2024], following the

configurations recommended in the VVC Common Test Conditions [Bossen], with QP values of 22, 27, 32, and 37, considering all frames. For 4K videos at QPs 22 and 27, only 60 frames were encoded due to the high computational cost.

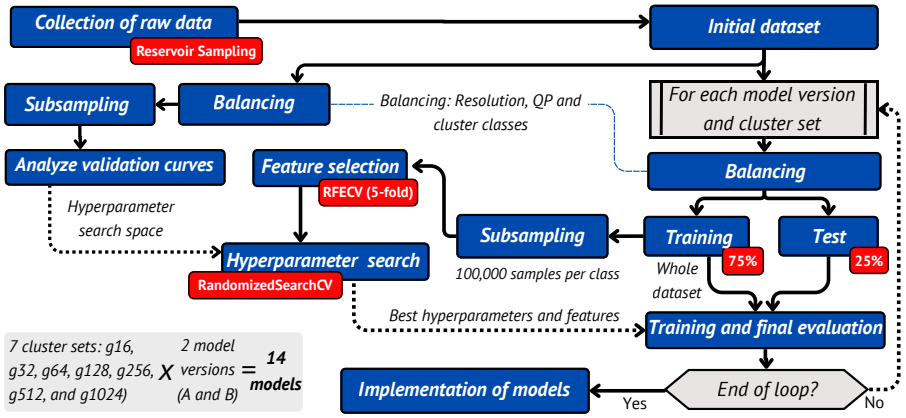

Figure 2. Pipeline of the model training process.

To ensure a balanced and representative dataset for each combination of transform type, block size, QP, and video, the *Reservoir Sampling* algorithm was employed [Vitter 1985]. A separate reservoir was maintained for each combination, ensuring diverse sampling. This approach enabled balanced supervised training and improved representation of less frequent transform cases. The decision tree models were trained using the *DecisionTreeClassifier* algorithm implemented in the *scikit-learn* library [Pedregosa et al. 2011], which provides tools for feature selection, hyperparameter optimization, and model evaluation. For the initial search of optimal configurations, a subset of 100,000 samples was used. Feature selection employed *Recursive Feature Elimination with Cross-Validation* (RFECV), and hyperparameter tuning used *RandomizedSearchCV* [Pedregosa et al. 2011]. A preliminary validation curve analysis was also performed to define appropriate search spaces and avoid overfitting or underfitting. During this process, multiple intermediate versions of the models were trained, all subjected to a balancing process to obtain classes with uniform sample distributions.

The resulting features and hyperparameters were defined separately for each block group and model configuration. The selected features include both basic block information, such as size and coding tree depth, and encoding-related data, including residual statistics. Residual-based features were consistently selected across most groups, while structural and partitioning features were more relevant for larger block sizes. In addition, VVC-specific characteristics were considered, including intra prediction modes, Matrix-based Intra Prediction (MIP), Intra Subpartition (ISP) flags, Multiple Reference Line (MRL) index, and partitioning history represented by the encoded split series. The hyperparameter configuration was also similar across groups, with most models using shallow-to-medium depth decision trees, with maximum depths ranging from 17 to 23 and up to 460 leaf nodes, while adopting either the Gini or entropy criteria.

After determining the best configurations, the final training of each model used 75% of the dataset, reserving 25% exclusively for evaluation, as shown in Fig. 2. This test set was not used in earlier stages. Accuracy results for each model and group are shown in Table 1. Accuracy results show consistent performance across block groups, with Model A around 0.80 and Model B slightly lower (0.71–0.77).

Table 1. Accuracy results for each model and block group

	Accuracy						
Model	**g16**	**g32**	**g64**	**g128**	**g256**	**g512**	**g1024**
A	0.80	0.79	0.80	0.79	0.80	0.80	0.81
B	0.73	0.72	0.71	0.72	0.72	0.73	0.77

4. Results and Discussion

The last stage of the pipeline depicted in Fig. 2 consisted of deploying the models into the VTM reference software. The models were integrated into VTM version 23.0 [JVET 2024]. Since the models were trained exclusively with intra blocks, the MTS flag was set to 1, enabling explicit MTS only for these blocks, the focus of the proposed optimization. The LFNST tool was disabled, as it is not applied to blocks with explicit MTS and lies outside this work's scope. This configuration aligns with methodologies adopted in previous studies [Wang et al. 2022, Camargo et al. 2025a, Camargo et al. 2025b]. In addition to the complete solution, Models A and B were evaluated separately. All test videos from the VVC CTC set were encoded using the first 60 frames of each sequence, with QP values of 22, 27, 32, and 37 [Bossen]. Table 2 presents the Bjontegaard Delta-rate (BD-rate) [Bjøntegaard 2001] and Time Saving (TS) obtained for the proposed models under the All Intra configuration. The table presents the average values of the results by class and the overall average across all sequences. The column "Model A+B" indicates the combined operation of both models, representing the scenario with the highest potential for time reduction. The proposed approach led to an increase of only 0.67% in BD-rate, indicating a minimal loss in compression efficiency. In contrast, the total encoding time was reduced by 18.76%, a significant improvement for the overall process. Considering only the transform step, the average encoding time reduction reached 30.90% when both models are employed.

Additionally, Table 2 shows the individual performance of each model. In the Model A column, results correspond to encoding with only Model A, responsible for deciding whether additional transform combinations should be evaluated. Unlike the original encoder, which tests all transform pairs when MTS is enabled, Model A performs this selectively and intelligently, reducing total encoding time by 11.27% with only a 0.63% BD-rate increase, indicating that most compression efficiency loss originates from the use of this model. For Model B, the results are also promising. The total encoding time decreased by 11.35% on average, with a negligible BD-rate increase of 0.15%, making it an excellent option for reducing encoding time with minimal compression loss. This occurs because Model B limits evaluations to the most common transform pairs (DCT-II×DCT-II and DST-VII×DST-VII), reducing errors without losing flexibility.

Table 2. Average BD-rate (%) and Time Saving (%) per class (All Intra)

Class	**Models A+B**		**Model A**		**Model B**	
	BD-rate	**TS**	**BD-rate**	**TS**	**BD-rate**	**TS**
A1	0.90	24.58	0.95	17.21	0.17	14.88
A2	0.73	20.58	0.62	12.73	0.22	11.95
B	0.80	20.65	0.75	13.08	0.19	11.97
C	0.56	16.00	0.49	8.11	0.16	10.41
D	0.48	15.96	0.44	8.06	0.14	10.13
E	0.78	18.88	0.74	11.64	0.13	11.01
F	0.46	14.67	0.43	8.06	0.08	9.08
Average	0.67	18.76	0.63	11.27	0.15	11.35

Besides the main analysis under the All Intra configuration, additional experiments using Random Access configuration were conducted to assess the generalization of the models under different conditions. In this configuration, the proposed method achieves a 29.74%

time reduction in the transform step when processing intra-predicted blocks, exhibiting behavior consistent with previous results. However, since the models are applied only to intra-predicted blocks – which represent a minority in the Random Access configuration – the overall encoding time reduction is 3.53%, with a 0.39% increase in BD-rate. These results highlight the potential of extending the approach to inter-predicted blocks in future work, enabling further acceleration in Random Access scenarios.

Finally, an additional experiment enabled only the DCT-II×DCT-II transform. This setup was used to assess whether removing transform variability could outperform the proposed models and to estimate the maximum possible time savings and BD-rate impact. The result, presented in Table 3, shows that, although this configuration reduces encoding time by 43.05%, it significantly increases the BD-rate, more than twice the value obtained with the proposed approach. Table 3 also compares this work with related ML approaches for reducing transform stage cost. Direct comparisons are difficult due to methodological differences: [Samir et al. 2024b] and [Saldanha et al. 2022] use older VTM versions (14 and 10), while this study uses VTM 23; [Camargo et al. 2025a] and [Samir et al. 2024b] report only Random Access results, and only [Camargo et al. 2025b] and [Saldanha et al. 2022] evaluate All Intra. Also, BD-rate values are not reported in [Samir et al. 2024b]. To provide a broader perspective, the BD-rate/TS trade-off is presented in Table 3, where lower values represent better balance between compression efficiency and time. Although [Camargo et al. 2025b] reports a 30.32% time reduction (versus 18.76% in this work), its BD-rate increase was higher (0.89% vs. 0.67%), leading to a less favorable trade-off. It is also noteworthy that [Camargo et al. 2025b] trained models only on 32×32 blocks, without specialization for different block groups. Model B achieved the best balance between encoding time reduction and compression efficiency, with the lowest BD-rate impact, making it ideal for scenarios needing minimal loss and significant time savings.

Table 3. Comparison with Related Work (All Intra)

Solution	BD-rate (%)	TS (%)	BD/TS
Ours - Models A+B	0.67	18.76	0.036
Ours - Model A	0.63	11.27	0.056
Ours - Model B	0.15	11.35	0.013
DCT-II×DCT-II only	1.58	43.05	0.037
[Camargo et al. 2025a]	1.16	7.98	0.090
[Camargo et al. 2025b]	0.89	30.32	0.038
[Saldanha et al. 2022]	0.21	5.00	0.042
[Samir et al. 2024b]	–	9.05	–

5. Conclusion

The proposed approach introduced a machine learning-based strategy to accelerate the MTS process in the VVC encoder. By integrating lightweight decision tree models trained for specific block sizes through a robust training methodology – including balanced sampling, feature selection, and hyperparameter optimization – the method replaces the exhaustive evaluation of transform combinations with predictive decisions, significantly reducing computational demand. Experimental results demonstrated an average 18.76% reduction in total encoding time with only a 0.67% increase in BD-rate, while the transform stage alone achieved a 30.90% time saving. Compared to related works, the proposed method provided a superior balance between compression efficiency and complexity reduction, confirming the effectiveness of predictive modeling for practical VVC acceleration.

References

Bjøntegaard, G. (2001). Calculation of average psnr differences between rd-curves.

Bossen, F. VTM common test conditions and software reference configuration for SDR video. https://jvetexperts.org/doc_end_user/current_document.php?id=10545.

Camargo, C., Silveira, B., and Correa, G. (2025a). Computational cost analysis and reduction of vvc multiple transform selection. In *2025 IEEE 16th Latin America Symposium on Circuits and Systems (LASCAS)*, volume 1, pages 1–5.

Camargo, C., Silveira, B., Zatt, B., and Correa, G. (2025b). Machine learning-driven multiple transform selection for low-complexity vvc encoding. In *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5.

Camargo, C. S., Silveira, B., and Correa, G. (2025c). A comprehensive analysis of the multiple transform selection tool in versatile video coding. *Journal of Integrated Circuits and Systems*, 20(2).

Fraunhofer (2020). Fraunhofer hhi is proud to present the new state-of-the-art in global video coding: H.266/VVC brings video transmission to new speed. https://newsletter.fraunhofer.de/-viewonline2/17386/465/11/14SHcBTt/V44RELLZBp/1.

Hamidouche, W., Biatek, T., Abdoli, M., François, F., Radosavljević, M., Menard, D., and Raulet, M. (2022). Versatile video coding standard: A review from coding tools to consumers deployment. *IEEE Consumer Electronics Magazine*, 11(5):10–24.

JVET (2024). VVC software VTM, VTM-23.0.

Pakdaman, F., Adelimanesh, M. A., Gabbouj, M., and Hashemi, M. R. (2020). Complexity analysis of next-generation VVC encoding and decoding. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3134–3138.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12(null):2825–2830.

Saldanha, M., Sanchez, G., Marcon, C., and Agostini, L. (2022). Fast transform decision scheme for VVC intra-frame prediction using decision trees. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1948–1952.

Samir, S., Damak, T., Saumard, M., Ben Ayed, M. A., Jridi, M., and Masmoudi, N. (2024a). Performance analysis of mts on the vvc encoder. *Signal, Image and Video Processing*, 18(2):1671–1681.

Samir, S., Saumard, M., Damak, T., Jridi, M., and Ben Ayed, M. A. (2024b). Random forest based fast mts algorithm for vvc encoder. In *International Conference on Service-Oriented Computing*, pages 287–298. Springer.

Siqueira, Í., Correa, G., and Grellert, M. (2020). Rate-distortion and complexity comparison of hevc and vvc video encoders. In *2020 IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS)*, pages 1–4. IEEE.

Vitter, J. S. (1985). Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1).

Wang, Z., Wang, J., Yang, J., Luo, C., Liang, F., and Huang, K. (2022). A fast transform algorithm for VVC intra coding. In *2022 11th International Conference on Communications, Circuits and Systems (ICCCAS)*, pages 237–240.

Zhao, X., Chen, J., Choi, K., Cock, J. D., Grange, A., He, Y., Helle, P., Ye, Y., and Zhang, L. AOM common test conditions v2.0. https://aomedia.org/docs/CWG-B075o_AV2_CTC_v2.pdf.

Zhao, X., Kim, S.-H., Zhao, Y., Egilmez, H. E., Koo, M., Liu, S., Lainema, J., and Karczewicz, M. (2021). Transform Coding in the VVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3878–3890.