EURASIP Journal on Advances
in Signal Processing

RESEARCH Open Access

Low-complexity enhancement VVC encoder for vehicular networks

Xiantao Jiang[1*], Wei Li[1] and Tian Song[2]

*Correspondence:
xtjiang@shmtu.edu.cn

[1] Department of Information Engineering, Shanghai Maritime University, No. 1550, Haigang Ave., Shanghai 201306, China
[2] Department of Electrical and Electronics Engineering, Tokushima University, 2-24 Shinkura-cho, Tokushima 770-8501, Japan

Abstract

In intelligent transportation systems, real-time video streaming via vehicle networks has been seen as a vital difficulty. The goal of this paper is to decrease the computational complexity of the versatile video coding (VVC) encoder for VANETs. In this paper, a low-complexity enhancement VVC encoder is designed for vehicular communication. First, a fast coding unit (CU) partitioning scheme based on CU texture features is proposed in VVC, which aims to decide the final type of CU partition by calculating CU texture complexity and gray-level co-occurrence matrix (GLCM). Second, to reduce the number of candidate prediction mode types in advance, a fast chroma intra-prediction mode optimization technique based on CU texture complexity aims to combine intra-prediction mode features. Moreover, the simulation outcomes demonstrate that the overall approach may substantially reduce encoding time, while the loss of coding efficiency is reasonably low. The encoding time can be reduced by up to 53.29% when compared to the VVC reference model, although the average BD rate is only raised by 1.26%. The suggested VVC encoder is also hardware-friendly and has a minimal level of complexity for video encoders used in connected vehicle applications.

Keywords: Vehicular networks, Versatile video coding, Low complexity, CU partitioning, Intra-prediction

1 Introduction

The quantity of road media has rapidly expanded with the quick growth of the transportation information sector, and the severity of the traffic safety situation has worsened. Vehicle Ad Hoc Networks (VANETs), as the infrastructure of Intelligent Transportation Systems (ITS), can provide multimedia communication between vehicles, thus playing an important role in accident warning, traffic safety, and passenger entertainment. Different sensor-equipped vehicles can exchange braking, network, and obstacle avoidance data. Sharing videos of traffic accident scenes might also facilitate rescue efforts and reduce traffic congestion. In the actual road environment, the dynamic changes in topology structure due to fast driving and the serious interference from terrain and objects on the communication performance of wireless signals between vehicle nodes have made high-quality real-time video transmission a difficult problem [1]. The primary goal of this effort is to create a video codec that enables real-time video transmission via VANETs for applications related to traffic safety.

In connected vehicle applications, due to the rapid movement of the vehicle or the movement of the photographed object, and the rapid change of the image in the encoding, the generated video image data packets will suddenly increase, and a burst of large capacity needs to be transmitted in the wireless network with limited bandwidth. Video data can cause network congestion and packet loss. Bandwidth restrictions and high loss rates, opportunity, connectedness, and mobility are the major issues that VANETs must contend with [2]. Bandwidth restriction is a hurdle for real-time video transmission over VANETs because video data in applications for road safety is resource-intensive [3, 4]. Additionally, streaming video over VANETs remains extremely difficult because of the low battery life of vehicle nodes. Essentially, low-complexity video encoders can speed up real-time video transmission and make low-latency video streaming possible. Therefore, it is crucial to develop efficient video encoders to convey real-time video with bandwidth constraints. Video encoders with high coding efficiency and low coding complexity are the core requirements of VANETs [5].

The goal of video coding technology is to maintain the quality of subjective perception while reducing the cost of video transmission. H.266/versatile video coding (VVC), the most recent video compression standard released by the Joint Video Expert Group (JVET) [6], has excellent coding performance. Under the same subjective visual perception quality, the performance of VVC can save about 50% bitrate on average compared with H.265/High Efficiency Video Coding (HEVC). VVC considers more video formats and content, aiming to provide more powerful compression performance and more flexible functions for existing and emerging video applications. Various techniques to improve coding performance have been proposed, including QTMT structure partition, intra-block copy (IBC), cross-component linear model (CCLM), geometric partitioning mode (GPM), etc. [7, 39, 40]. The introduction of new technology also brings high complexity. On average, the comprehensive complexity of VVC is 10 times higher than that of HEVC, as determined by the reference program VTM [8], which makes it difficult for VVC to be put into practical application. Reduced computational complexity in the coding process is, therefore, necessary while pursuing excellent compression performance.

Numerous improvements have been made recently to coding effectiveness. Even though several VVC optimization algorithms have produced positive advances in the past, the difficulty of quick CU division and chroma intra-prediction remains relatively high. This paper combines a fast CU partition scheme and an optimization algorithm for the candidate mode list of chroma intra-prediction to greatly reduce coding complexity. It has not been extensively researched from the point of view of hardware implementation how real-time coding complexity can be reduced by utilizing the available processing power on VANETs nodes. The main contributions of our work are as follows:

- We propose a low-complexity and hardware-friendly H.266/VVC encoder. The encoder can significantly reduce the coding complexity, thus meeting the low-latency requirement for video transmission in power-limited VANETs nodes. Compared with the anchored VTM-12.0, the encoding time is reduced by 53.29% with negligible encoding efficiency loss, and the proposed encoder is suitable for real-time video applications.

- A fast CU partitioning scheme is proposed. The texture feature of the block is determined by calculating the texture complexity and gray-level co-occurrence matrix (GLCM) of the coding unit, to determine the division type of the block in advance. Specifically, in addition to the traditionally used early termination method that directly skips all further partitions of the CU, we also use the GLCM to calculate texture features, thereby determining QT partitions, and terminating asymmetric rectangular partitions. Finally, combined with the complexity and direction of each block, the final partition is selected from the remaining 4 MT partitions.
- Based on CU texture properties, we provide a quick chroma prediction mode list improvement approach. Combined with the phenomenon that different intra-prediction modes show different prediction quality for video content, the mode that does not meet the conditions is skipped through different CU texture complexity to reduce the calculation of many RD costs.

The remainder of this essay is structured as follows. A short outline of the relevant work is given in Sect. 2. Background information is covered in Sect. 3. In Sect. 4, the fast CU splitting scheme and the chroma prediction candidate mode list optimization algorithm are presented. The experimental findings are given in Sect. 5. The paper is finally concluded in Sect. 6.

2 Related work

2.1 Low-complexity VVC encoder algorithm

The research on the optimization of the coding unit partition process has been widely carried out. Shen et al. [9] used CU texture characteristics to encode information from adjacent CUs and then skipped several CU sizes using texture uniformity and spatial correlation. The computational complexity of the partitioning operation is clearly reduced. By omitting pointless partition modes and intra-prediction processes, Yang et al. [10] created a low-complexity CU partition pipeline that is modeled as a multi-binary categorization procedure based on data analysis. By analyzing the edge types existing in each CU and its four sub-CUs, for each depth level, the CU will be determined as a partitioned CU, non-partitioned CU, or undefined CU, so as to determine the final CU partition mode [11]. Recently, learning-based CU partition theory has received more and more attention. Fu et al. [12] used the segmentation type and intra-prediction mode of the sub-CU as the input characteristics and skipped different segmentation modes in advance. In [13], Xu et al. terminated the CU partition process early by utilizing a specialized network structure and a hierarchical CU partition map, greatly reducing the computational complexity while achieving high prediction accuracy. In [14], a fast QTBT partitioning decision is introduced, and the computational complexity of coding unit partitioning is reduced using the CNN-based coding structure. Despite not requiring manually created features, CNN-based approaches require more computational work from the encoder due to complicated network computing and are unable to effectively utilize the significant spatial correlations of the video.

As an important part of intra-prediction, chroma intra-prediction has attracted more academic research on the optimization of chroma intra-prediction due to the CCLM technology proposed by VVC. In [15], two CCLM variants are proposed, which derive the linear model by using only left or upper adjacent samples. Zhang et al. [16] used a piecewise linear regression model to represent the linear correlation more accurately between channels. It is suggested to use an adaptive template selection strategy [17], so that the prediction of the Cr component can be combined with the Y component or a combination of Cr components. To further boost the prediction's precision, Zhang et al. [18] proposed a multi-hypothesis algorithm that combines the benefits of corner intra-prediction and CCLM intra-prediction.

Different from previous methods, to reduce the coding complexity of the H.266/VVC encoder, this paper suggests a quick algorithm for CU division scheme and chroma intra-prediction optimization decision. In summary, the creation of video encoders that permit actual video streaming over VANETs is the main topic of this study. To better accommodate video transmission on VANETs, we created a hardware-friendly, low-complexity encoder. Additionally, the suggested encoder can obtain a greater trade-off between coding efficiency and computation cost than earlier efforts.

2.2 Video streaming in vehicular networks

Real-time video transmission through VANETs for information exchange will substantially improve the accuracy of information communication and driving safety. The most difficult components of video transmission via VANETs are low latency [19]. Recently, various ideas to address this issue have been put up. Reference [20] reduces video quality degradation caused by transmission impairments by combining techniques of intra-refresh video coding mode, frame segmentation, and quality of service (MAC) levels at media access control. The findings demonstrate that combining the suggested strategies greatly increases the video transmission's robustness for connected vehicle applications. To improve the user experience's quality and their perception of scalable video via VANET during network transmission, a cooperative vehicle to vehicle communication technique is suggested [21]. To increase the subjective quality of video, it is advised to use the selective added redundancy solution algorithm. The redundancy of video streams on VANET has also been researched concurrently [22]. Furthermore, a vehicle incentive and punishment method for video transmission on VANET was also proposed in Reference [23], which uses real domain and relative speed to optimize video transmission. Reference [24] proposed a delay, link quality, and link lifetime-aware routing protocol for video streaming to optimize video delay. To improve and optimize video communication, Marzouk et al. [25] proposed an advanced simulation toolset that integrates the EvalVid framework into NS-2 and adds a moving file generated from real-life maps using SUMO, which will make the toolset Confidence in measuring delivered video quality. With improvements, this toolkit enables academics studying networks to assess video streams using suggested network designs or protocols and to assess the video quality of their developed video encoding techniques using more plausible networks. Even though the issue of video streaming over VANETs has been studied in the past, the performance of video codecs that permit real-time video transmission is inadequate.

3 Technical background

VVC continues to use the traditional coding framework composed of six processes. Reduced spatial and temporal redundancy is achieved using intra- and inter-prediction modules. To cut down on visual repetition, transform and quantization modules are employed. Loop filtering is used to improve video quality and reduce deblocking and ringing. It should be noted that the improvement of VVC for the intra-prediction process introduces a lot of computational complexity, including the QTMT partition to adapt to different video contents and the chroma intra-prediction process, and so on. The recursive structure of coding blocks leads to a lot of redundant computations. The introduction of the chroma prediction candidate mode list greatly improves the efficiency of chrominance prediction, but there is still room for optimization in the computational complexity caused by the comparison and calculation of multiple prediction modes.

3.1 CU partitioning in VVC

The final CU size in VVC is established using a QTMT partition structure as opposed to the QT partition in HEVC. Quadtrees are first separated into the code tree unit (CTU). The multi-type tree (MTT) structure can then be used to further separate the four-leaf nodes. Four symmetrical partition modes are just a few of the five possible partition configurations of the QT node shown in Fig. 1. The BT structure splits the CU into two rectangular blocks with symmetrical areas, in contrast to the QT structure that divides the CU into four square sub-CUs, and TT structure divides CU according to the principle of asymmetry. The block needs to be forcibly divided until all of the sample points of the CU are inside the image boundary when the CU on the image boundary exceeds the bottom or right edge of the image. The processing of various video content is particularly friendly to this division structure. However, the complexity of the coding is necessarily increased by going through every potential partition combination in CTU one at a time.

The QTMT structure contains several predetermined restrictions to prevent overlapping partitions. For instance, the smallest QT partition is 16 × 16 and the largest MT partition is 32 × 32. The QT partition will not be allowed to acquire leaf nodes if MT splits the CU. It is not permitted to partition between sub-blocks using BT if the current CU uses TT partitioning. Additionally, treeing is not allowed when the length or height of the luminance encoding block is greater than 64 to adapt the pipeline design of 64 × 64 brightness blocks and 32 × 32 chroma blocks.

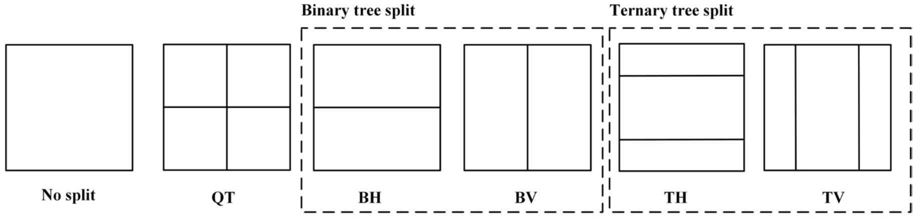


Fig. 1 Five different types available in the QTMT partitioning mechanism

Because the CTU can be any size, the block partition structure has the advantage that the codec can be easily optimized for various contents, applications, and devices. The recursive nature of coding blocks, however, results in a great deal of unnecessary processing. Although mobile vehicle video codecs for VANET must be more dependable and energy efficient, it is crucial to keep video encoder complexity low. The H.266/VVC encoder's redundancy computation needs to be significantly scaled back to handle real-time video transmission via VANET.

3.2 Chroma intra-prediction

The goal of chroma prediction technology is to enhance coding performance by utilizing component correlation. The luminance component has finished encoding to acquire the luminance reconstruction value before the CU chroma component is anticipated, and the obtained luminance data can be utilized in the chroma prediction process. Through luminance down sampling, CCLM is compatible with the spatial resolution of the chromaticity component. The chromaticity prediction value is then computed using the luminance reconstruction value and the linear model parameters derived from the original sample data. The CU texture features will obstruct the chroma prediction procedure. On the one hand, CCLM cannot effectively depict the association between complex components because of the uneven distribution of luminance and chrominance components. On the other hand, the processing likelihood for CU with complex texture is very low for other angle prediction modes in the list of chroma prediction candidate modes. Therefore, the issue of high computational redundancy in chroma prediction must be resolved.

Table 1 displays the chroma prediction modes' categories and method of derivation. 81, 82, and 83 are three cross-component linear prediction modes, while the remaining numbers are related angle prediction modes. The numbers in Table 1 correspond to different intra-prediction modes. One chrominance block corresponds to numerous luminance blocks because the luminance and chrominance components have independent block division structures. As a result, the intra-prediction mode covering the matching luminance blocks at the same place is directly inherited for the chroma prediction DM mode.

Table 1 Types and derivation of prediction models

Chroma prediction mode	**Luma intra-prediction mode**				
	0	**50**	**18**	**1**	**X ($0 \leq X \leq 66$)**
0	66	0	0	0	0
1	50	66	50	50	50
2	18	18	66	18	18
3	1	1	1	66	1
4	0	50	18	1	X
5	81	81	81	81	81
6	82	82	82	82	82
7	83	83	83	83	83

4 The proposed VVC encoder for vehicular networks

VVC introduces numerous new strategies that increase coding effectiveness but also significantly increase computing complexity, especially the CU partition and the chroma intra-prediction process. The CU partitioning involves a rate distortion optimization process for many blocks, resulting in a large amount of computational complexity. The process of selecting the optimal prediction mode for chroma intra-prediction lacks the analysis of the prediction mode and the texture features of the block, so there is a lot of computational redundancy. This part is problem-oriented, combined with the complexity and direction of the block, and creates a series of quick algorithms based on the QTMT and chroma prediction process. Section 4.1 is the fast CU division module, which combines the texture complexity of the block and the GLCM to determine the final block division type. Section 4.2 is the optimization of chroma intra-prediction mode selection, which determines the final chroma prediction mode selection by analyzing the characteristics between the block's texture complexity and the prediction mode. Section 4.3 presents the hardware architecture of the encoder, and the reduction in computational complexity significantly reduces the energy consumption of the hardware design.

4.1 Fast CU partitioning decision

The experiment was conducted using the pertinent reference program VTM-12.0 and the official documents produced by JVET [7]. Since the size of the CU differs due to the proportionate division structure of VTM, utilizing QTMT for blocks with various attributes can significantly increase coding efficiency. On the one hand, there are sizes of 64x64 partitions distributed, which is significantly fewer than the number of other modes [26, 27]. However, considering the direct decision for the partition of 64 × 64, a significant reduction in coding quality may result. Therefore, 32 × 32 blocks are the fundamental building blocks of the following algorithm, considering the efficiency of coding and computing complexity.

We combined the texture complexity of blocks with GLCM to create a feasible fast CU partitioning technique. Initially, RDO-based methods were used to partition blocks until each block was 32 × 32. The proposed algorithm consists of three steps, namely, Step A, Step B, and Step C:

Step A is based on early termination determined by texture, which first determines whether the texture complexity of the block is simple or complex, and the next step is to decide whether to segment it. By ending the further splitting of the CU, all five partition modes are bypassed, which saves a lot of time.

Step B is achieved by choosing QT based on GLCM. If the texture of the block is complex, then the next step is B. We calculate the GLCM of the block to determine the texture direction of the block. If the texture direction of the block is neither horizontal nor vertical, we select the QT partition. If that is the case, the remaining four partitions will not be used. This operation prematurely terminates all MT partitions.

Step C is achieved by choosing one partition from four candidates based on texture direction and complexity, which selects the final partition from one of the four MT structures based on the block's texture complexity level and texture direction. The three steps are interrelated and progressive, as shown in Fig. 2. The following content provides a detailed description of the three steps.

4.1.1 Early termination determined by texture

CU partitions are highly correlated with texture and motion complexity in video frames. A CU that is relatively complicated is likely to be divided into smaller CUs for more precise prediction, according to the appropriate CU partitioning in VVC. Likewise, for less complex CUs, mode decisions at small depths are not required. When CUs are smooth and simple, they tend to split into larger CUs. Alternatively, sections with intricate textures might be split up into smaller CU sizes. The opening frame of the traffic sequence is depicted in Fig. 2. It can be seen that whereas the characteristics of blocks C and D are quite rich, those of blocks A and B are smooth (Fig. 3).

First, we need to measure the texture of the block to determine the complexity of the CU. MAD (mean absolute deviation) can be used as a typical representation of texture complexity measurement due to its processing characteristics. MAD is calculated as follows:

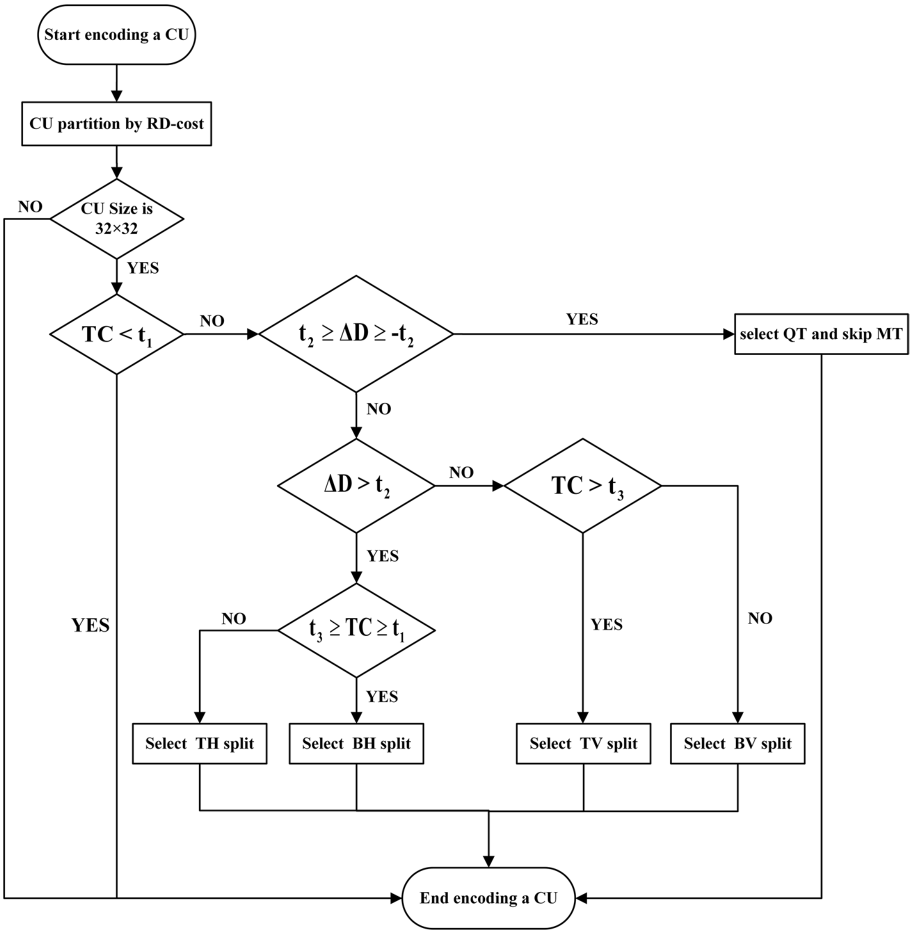


Fig. 2 Flowchart of fast CU partitioning algorithm

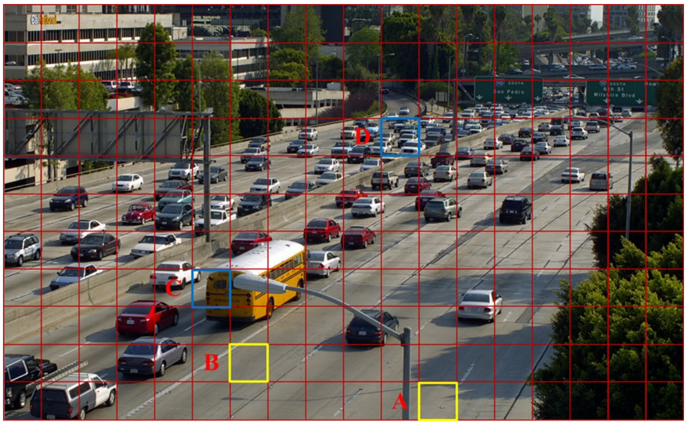


Fig. 3 The first frame of traffic sequence

$$\mathrm{MAD} = \frac{1}{\text{height} \times \text{width}} \sum_{j=1}^{\text{height}} \sum_{i=1}^{\text{width}} \mid P(i,j)\text{-mean} \mid, \tag{1}$$

where $P(i, j)$ is the luma value of the pixel at (i, j) and width and height are the corresponding width and height of the CU. The *mean* is the average of all sampled pixels.

To reduce the computational complexity, this paper uses the method of interval sampling to simplify the MAD calculation. This sampling technique can precisely quantify the texture complexity and cut the computation time in half. The formula used to calculate texture complexity (TC) is as follows:

$$\begin{aligned}\mathrm{TC} = \frac{2}{\text{height}\times\text{width}} &\left(\sum_{j=2n-1}^{\text{height}} \sum_{i=2n-1}^{\text{width}} \mid P(i,j) - \text{ mean } \mid \right. \\ &\left. + \sum_{j=2n}^{\text{height}} \sum_{i=2n}^{\text{width}} \mid P(i,j) - \text{ mean } \mid \right),\end{aligned} \tag{2}$$

where $P(i, j)$is the luma value of the pixel at (i, j) and width and height are the width and height of the CU, respectively.

CU is split into basic texture and complicated texture depending on the value of TC. When TC is lower than the set threshold t_1, the current CU is categorized as simple texture. When TC exceeds t_1, CU is classified as having complex texture. Rate distortion (RD), which is the equilibrium between the number of bits needed to encode an image block and the distortion these bits induce, is usually used to search for the best coding settings. When intra-prediction rate distortion optimization is carried out, RD cost is a good indicator of performance. The following is the RD cost function calculation formula:

$$J = D + \lambda \times R, \tag{3}$$

where D is the distortion, R is the number of bits needed to send the signaling parameter information and residual coefficient, and λ is the Lagrange multiplier factor to the intra-prediction stage.

We statistically observed the CU brightness samples in the test video sequence and calculated the TC of each CU to judge the texture complexity. The RD cost of a CU that has been divided into blocks is denoted as $J - partition$, and the RD cost of a CU that has not been divided into blocks is denoted as $J - no\ partition$. In the sequence for data statistics, the TC values of the CU are divided into two groups according to the pre-division and post-division. The texture complexity of CU is judged according to the probability density of TC, as shown in Fig. 4. The figure shows that when TC is in the small range fluctuation area of 20, the probability density changes of the first group and the second group are most obvious. Therefore, we set t_1 to 20. When the size of CU meets 32×32, if the TC value is less than 20, the texture of the current block is judged as simple, and the segmentation operation is skipped directly. On the contrary, we consider the texture of the current block to be complex and perform additional segmentation operations, as shown in steps B and C below.

4.1.2 Choosing QT based on GLCM

Through the above method, we can easily distinguish whether the texture of the block is simple or complex. In this part, we further analyze the texture direction information of the coding unit by calculating the GLCM [28, 38]. GLCM can display useful information about image gray levels regarding direction, interval, and dynamic area, making it a common tool for measuring texture characteristics [29]. The GLCM method accurately describes the texture features of the encoding unit based on its own characteristics. Therefore, we consider using GLCM to analyze the direction information of coding units.

First, when the size of the coding unit meets the above premise, calculate the current GLCM, and calculate the COR parameters along the horizontal direction (CORH) and vertical direction (CORV) as

$$\mathrm{COR} = \frac{\sum_{a,b}[P(a,b)]\mu_x\mu_y}{\sigma_x\sigma_y}, \tag{4}$$

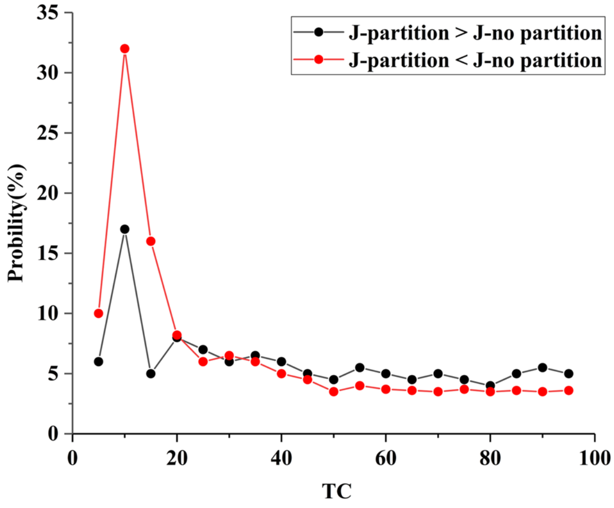


Fig. 4 Probability density of TC

where $P(a, b)$ stands for the GLCM of the current CU and other a and b combinations indicate the GLCM of various nearby angles and intervals. The mean is represented by μ_x and μ_y, while the standard deviation is shown by σ_x and σ_y.

The texture orientation of the existing CU is established by computing the difference ΔD from CORH and CORV and comparing it with the threshold t_2, which is done as

$$\Delta D = \text{COR H-CORV}, \tag{5}$$

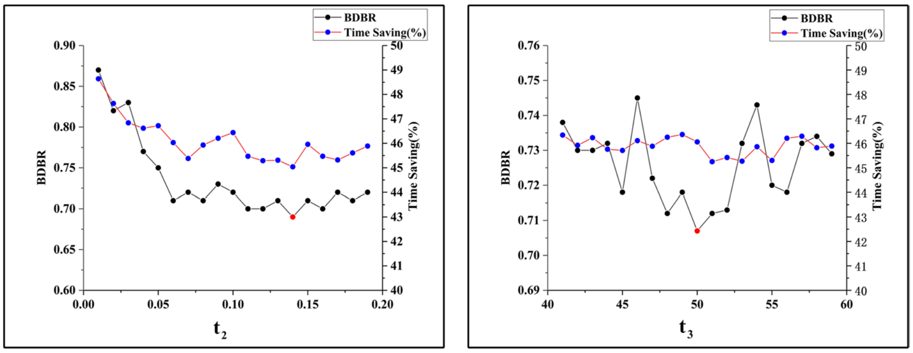
$$\text{direction } = \begin{cases} \text{hor}, & \text{if } \Delta D > t_2, \\ \text{ver}, & \text{if } \Delta D < -t_2, \\ \text{uncertain}, & \text{otherwise.} \end{cases} \tag{6}$$

According to the results of comparing the difference ΔD of CORH and CORV with the threshold value t_2, three cases are classified. If $\Delta D > t_2$, the texture direction of the current CU is determined to be horizontal; if $\Delta D < -t_2$, the texture direction of the current CU is determined to be vertical; if $t_2 \geq \Delta D \geq -t_2$, the texture of the current CU may appear symmetrical or intricate irregular texture. QT partitioning is suitable for texture symmetry or complex and irregular texture. Therefore, when $t_2 \geq \Delta D \geq -t_2$ this occurs, the CU is directly divided by QT, and no MT partition is processed. If step A signifies the ending of every partition structure, then step B can be viewed as the termination of every MT partition structure, in which rectangular partitions can be directly ignored.

We considered BDBR and time savings when choosing the t_2 threshold. A line graph of BDBR and time savings as a function t_2 is shown in Fig. 5a. The probability of CU being split by QT cannot be too high, which will cause the predictor not to enter step C, so t_2 cannot be set too high. To ensure good performance, choose a value where the BDBR and Time saving are considered comprehensively. Based on observations on training results, we set the t_2 threshold to 0.12.

4.1.3 Choosing one partition from four candidates based on texture direction and complexity

For improved prediction performance, sub-blocks are used to separate areas with different textures, as these sub-blocks exhibit distinct texture differences within the initially divided blocks. If the 32 × 32 CU does not meet the conditions outlined in step A and step B, we comprehensively consider both the texture complexity and texture direction



(a) (b)

Fig. 5 Results of two thresholds' training regarding time savings and BDBR. **a** Threshold t_2. **b** Threshold t_3

of the block to determine the final division. This decision involves selecting one of the four MT divisions (BH, BV, TH, TV) as the final partition. These four division modes are accurately categorized based on both texture complexity and texture direction.

According to Eq. 6, the texture direction is divided into horizontal texture and vertical texture by the relationship between the threshold t_2 and ΔD. If $\Delta D > t_2$, the texture of the block is horizontal, and the division mode of the block excludes BV and TV division. If $\Delta D < -t_2$the texture of the block is vertical, then the partition mode of the block excludes BH and TH partitions. Next, we further grade the texture complexity of the block based on determining whether the texture complexity of the block is simple or complex in step A. If the texture complexity of the block is complex, the threshold value t_3 is further used to classify the texture complexity of the block into relatively complex and very complex. If $t_1 \leq TC \leq t_3$, the texture complexity of the block is relatively complex. If $TC > t_3$, the texture complexity of the block is very complex. By dividing the texture of the block into relatively complex and very complex, the final selection can be made in the remaining two partition modes (BV and TV, BH and TH) to determine the final partition.

If the two-level division of texture complexity is directly determined by the RD cost according to the selection method in step A, there will be a certain error. Therefore, we comprehensively consider the two index factors of BDBR and time saving to determine the threshold t_3. A line graph of BD rate and time savings as a function of t_3 is shown in Fig. 5b. It can be seen in the figure that BD rate fluctuates with the change of t_3, while the change of time saving is not drastic. Therefore, we choose t_3 to correspond to BDBR, which is 50. The final selection is shown:

$$\begin{cases} \mathrm{BV}, & \text{if } \Delta D < -t_2 \wedge t_1 \leqslant \mathrm{TC} \leqslant t_3, \\ \mathrm{TV}, & \text{if } \Delta D < -t_2 \wedge \mathrm{TC} > t_3, \\ \mathrm{BH}, & \text{if } \Delta D > t_2 \wedge t_1 \leqslant \mathrm{TC} \leqslant t_3, \\ \mathrm{TH}, & \text{if } \Delta D > t_2 \wedge \mathrm{TC} > t_3. \end{cases} \tag{7}$$

It is important to note that the three steps involved in block partitioning correspond to three thresholds: t_1, t_2, and t_3. The selection of t_1 determines whether the encoding unit is considered complex and is based on the probability density of the tested sequence. t_2 is the threshold that determines the direction of the encoding unit, and the optimal value is selected based on factors such as BD rate and time savings. Similarly, t_3 determines the final partitioning method by considering the complexity and direction of the encoding unit and is also chosen as the optimal value based on factors like BD rate and time savings. When selecting experimental sequences, we adhere to the principle of robustness and reliability, taking into account the dynamic, static, texture, and color information of the video content to ensure a certain degree of reliability in threshold selection.

4.2 Selection of the best chroma intra-prediction mode

For chroma mode coding, there are eight intra-modes, including five traditional intra-modes and three CCLM modes. Intra-prediction modes exhibit different efficiencies based on various CU texture characteristics [30, 31]. This section proposes an improved algorithm for selecting the chroma intra-prediction mode. By analyzing the texture complexity of the coding unit and considering the characteristics of the intra-prediction mode, it is determined whether the chroma component prediction candidate mode should be skipped to save encoding time.

In addition to CCLM processing region characteristics, the smooth region typically has excellent linear correlation between components. On the other hand, it is ineffective to forecast regions with complex textures using simple linear regression. This demonstrates that when dealing with simple texture CU, the original three simple linear models (CCLM, CCLM A, and CCLM L) perform better. We can see that the intra-prediction modes have higher prediction effectiveness for simple textures from the overview of the association between texture attributes and intra-prediction modes in [32]. Prediction redundancy can be decreased by the interaction between CU texture features and prediction mode. In conclusion, for simple textures, five angle predictions modes are more effective. The derive mode (DM) differs from the preceding seven prediction modes, directly duplicates the prediction mode of the relevant luminance block among the eight-chrominance frame prediction candidate modes, as indicated in Table 1. Along with the DM mode, the method may effectively take advantage of the linkages between the remaining seven modes and the CU texture, and by bypassing processing, it can streamline the chroma intra-prediction process. The texture complexity judgment method is as described in step A, TC is obtained by Eq. 2, and finally, the complexity of the block is divided into complicated and simple according to the threshold t_1. If TC > 20, the texture complexity of the CU is complex. If TC < 20, the texture complexity of the block is simple. The algorithm is described as follows:

1. Obtain the current CU's texture complexity value TC.
2. When the CU texture is found to be complex, three cross-component prediction modes and five angle prediction modes in the candidate mode list are skipped, and DM is immediately selected as the best mode for chromaticity prediction.
3. Otherwise, the CU texture is determined to be simple, and the original prediction process is performed.

The flowchart of the algorithm is shown in Fig. 6.

The chrominance intra-prediction candidate mode and CU texture complexity are strongly related to the prediction mode described above. Except for the DM mode, out of the eight possible prediction modes, the other seven are more efficient for processing CUs with simple textures. Therefore, the RD cost calculation of the candidate modes is often carried out if the current CU is categorized as having a basic texture. The inefficient modes of the candidate mode list will be skipped if the current CU is not classified as having a complex texture; instead, DM will be chosen as the best mode for chrominance prediction.

In essence, the basic chroma intra-prediction algorithm is supplemented by a decision step. The TC value of the current CU is established after considering the chroma intra-prediction candidate mode. If the first prediction process proceeds as planned, move on to the following chroma intra-prediction and use the final DM mode right away.

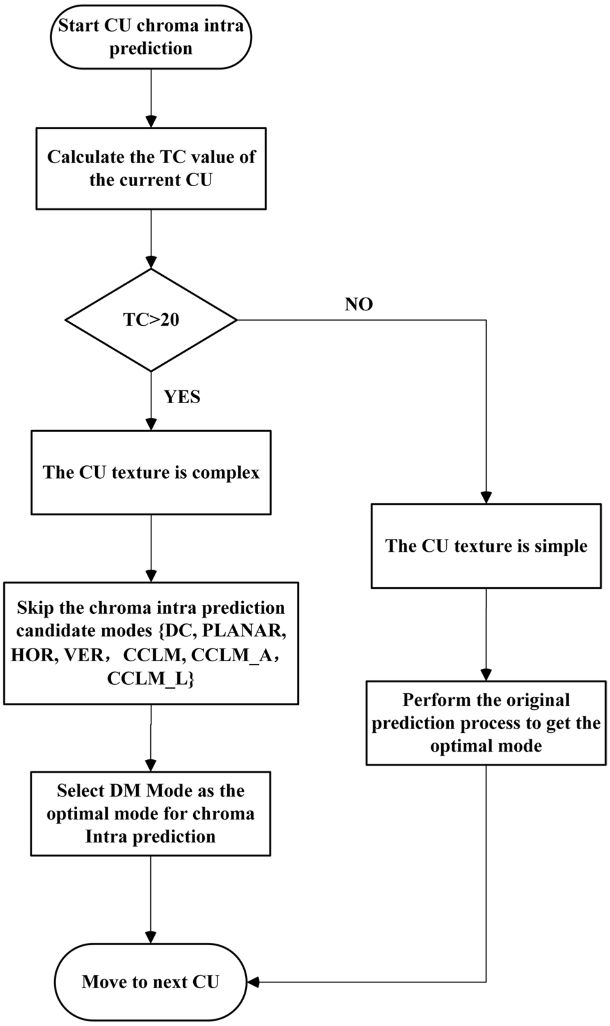

Start CU chroma intra prediction
Calculate the TC value of the current CU
TC>20
NO
YES
The CU texture is complex
The CU texture is simple
Skip the chroma intra prediction candidate modes {DC, PLANAR, HOR, VER，CCLM, CCLM_A，CCLM_L}
Perform the original prediction process to get the optimal mode
Select DM Mode as the optimal mode for chroma Intra prediction
Move to next CU

Fig. 6 Flowchart of the chroma intra-prediction optimization algorithm

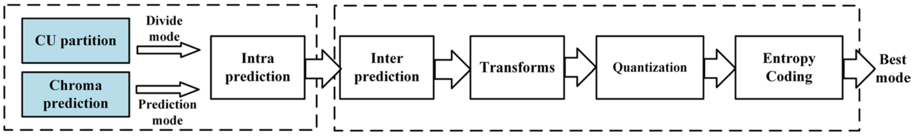


Fig. 7 Mode decision process of CU partition and chroma prediction

4.3 Encoder hardware architecture

The H.266/VVC core design is depicted in Fig. 7, along with the partition mode and prediction mode decisions. By using this architecture, computational complexity is reduced through fast CU partitioning and chroma intra-prediction optimization. Reducing encoder complexity is essential because video codecs for moving vehicle video codecs in VANETs must be more reliable and energy efficient. Additionally, the suggested H.266/VVC encoder, which is hardware-friendly and has low complexity, can significantly increase the dependability of VANETs video codecs. The energy consumption of the hardware design can be greatly decreased due to the high rate of complexity reduction.

5 Experimental results

To assess the efficacy of the suggested H.266/VVC encoder for VANETs' low complexity and hardware friendliness, we implemented the proposed algorithm using the VVC test model version 12.0 (VTM-12.0) to demonstrate experimental results. The proposed algorithm is primarily designed to reduce the complexity of intra-prediction. Given this focus, using an all-intra-configuration is sufficient to validate the performance of our algorithm. It is noted that P and B configurations are typically associated with inter-frame prediction techniques, and these configurations are not directly applicable to our algorithm's scenario. There are 22 standard test sequences, which are divided into 6 types of resolution sequences. According to the JVET Common Test Conditions (CTC) configuration [33], the encoding parameters were set at QP = 22, 27, 32, and 37, respectively. According to Reference [34], Bjontegarrd Delta Bit Rate (BDBR) and Bjontegarrd Delta PSNR (BDPSNR) were used to assess coding efficiency. Time saving (TS) is a measure of computational complexity reduction. The BDBR of all sequences is averaged to indicate the overall encoding quality. Fast techniques typically enhance BDBR. The encoding quality decreases with increasing BDBR value. The complexity reduction of the encoder is assessed using the average TS for all test sequences. Larger TS values indicate more time savings, and fast algorithms are efficient enough to reduce computational complexity. The shortened time is represented by the negative value of TS in the table, and the larger the value, the better the effect. The lost PSNR is represented by the negative BDPSNR in the table, and the larger the absolute number, the worse the effect. Equation 8 displays the TS calculation formula:

$$\mathrm{TS} = \frac{1}{4} \times \sum_{i=1}^{4} \frac{\mathrm{Time}_{\mathrm{VTM12.0}}\left(\mathrm{QP}_i\right) - \mathrm{Time}_{\ \mathrm{pro}}\left(\mathrm{QP}_i\right)}{\mathrm{Time}_{\ \mathrm{VTM12.0}}\left(\mathrm{QP}_i\right)}, \tag{8}$$

where $\mathrm{Time}_{\mathrm{VTM12.0}}\left(\mathrm{QP}_i\right)$ and $\mathrm{Time}_{\mathrm{pro}}\left(\mathrm{QP}_i\right)$ represent the encoding time using VTM-12.0 and the proposed algorithm under different QPs (Fig. 8).

The scenarios in this work have been specifically choose. The creation of a video codec that permits real-time video transmission via VANETs for applications relating to traffic safety is the main objective of this effort. Different spatiotemporal characteristics and frame rates are present in the CTC test sequences. Additionally, "DaylightRoad2" and "BQTerrace" (both depicted in Fig. 9) are examined in this work together with other video sequences of traffic scenarios.

Fig. 8 Traffic scenario. **a** The sequence of "DaylightRoad2." **b** The sequence of "BQTerrace"

Table 2 presents the single assessment results of the two strategies, Fast CU Partition (FCP) and Chroma Prediction Optimization (CPO), and the comprehensive evaluation results of the combination of the two strategies. FCP can achieve about 44.71% average encoding time savings and BDBR improves by about 0.84%. It can also be observed that the overall sequence obtains a consistent gain, with a minimum of 34.47% for "BQSquare" and a maximum of 53.21% for "BQTerrace." For high-resolution sequences, such as "BQTerrace" and "Kimono," the encoding time reduction using FCP is particularly high. As far as the CPO strategy is concerned, the average encoding time is reduced by 12.63% with a maximum of 16.54% in "PartyScene" and a minimum of 9.07% in "RaceHorses." Furthermore, the average BDPSNR decreased by 0.029 dB, while the average 0.31% increase in BDBR was not significant. It is discovered that using the suggested FCP and CPO algorithms together reduces coding execution time by an average of 53.29%, with a maximum value (Kimono) of 59.47% and a minimum value of 42.65% (RaceHorses). "Kimono" saves more time than other test videos because of fewer moving pictures and higher texture quality. The average BDBR of "BQSquare" increased by 1.26% and the minimum value was 0.67%, and the average BDPSNR of "KristenAndSara" decreased by 0.093 dB and the minimum value was 0.061dB. Therefore, according to the characteristics of the test video, the computational complexity of the encoding procedure can be greatly reduced by the suggested approach. In addition, the proposed method can achieve a good balance between computational complexity and coding efficiency.

The data from the prior fast algorithm [26] on QTMT and intra-partitioning based on depth feature fusion and probability estimate FCP algorithm [35] are compiled in Table 3. It should be noted that the VTM-12.0 configuration file disables ISP, MIP, and LFNST, which can have an impact on the encoding procedure. We employed the measurement indicated in [36, 37], which is used to compare TS/BDBR, to intuitively assess the performance of various approaches. We simply compare the available data with our data because, according to CTC, the test findings provided in [35] are incomplete. According to the comparison findings, the average BDBR and time savings for the FQDD method are 1.63% and 49.27%, respectively. The average BDBR and time savings for the DFFPE method are 1.40% and 55.59%, respectively. The proposed method introduces a fast-encoding unit partitioning scheme based on CU (Coding Unit) texture features. The primary objective is to determine the final type of CU partitioning through the calculation of CU texture complexity and grayscale co-occurrence matrix. The experimental results reveal that our FCP approach achieves an average encoding time savings of approximately 44.71%, while the BDBR sees only a modest improvement of about 0.84%. Moreover, in the context of autonomous driving in vehicular networks, video transmission latency consists of two primary components: video compression latency and video communication latency. Our proposed method yields significant savings, approximately 45%, in video encoding time, which directly contributes to reducing the latency introduced by video compression. However, it is important to note that the discussion in this paper is limited to the compression aspect of latency.

The effectiveness of each stage of our FCP algorithm is displayed in Table 4. We combine Steps A and B while evaluating TS because Step A's data must be extracted, stored, and integrated with Step B. The time required by this process will roughly

Table 2 Performance of FCP and CPO algorithm

Class	Sequence	FCP			CPO			FCP+CPO		
		BDBR (%)	BDPSNR	TS (%)	BDBR (%)	BDPSNR	TS (%)	BDBR (%)	BDPSNR	TS (%)
A1	Tango2	1.12	− 0.049	− 41.61	0.27	− 0.027	− 14.45	1.49	− 0.082	− 54.51
	FoodMarket4	0.53	− 0.072	− 46.43	0.32	− 0.034	− 15.17	0.93	− 0.117	− 59.04
	Campfire	0.85	− 0.056	− 42.77	0.33	− 0.028	− 14.59	1.26	− 0.085	− 54.62
A2	CatRobot	0.96	− 0.045	− 48.02	0.25	− 0.035	− 13.98	1.28	− 0.079	− 58.63
	DaylightRoad2	0.47	− 0.033	− 41.16	0.29	− 0.029	− 14.46	0.77	− 0.081	− 54.77
	ParkRunning3	0.83	− 0.112	− 49.46	0.32	− 0.032	− 12.61	1.38	− 0.073	− 58.91
B	BasketballDrive	0.79	− 0.062	− 39.63	0.27	− 0.029	− 14.02	1.15	− 0.104	− 49.28
	BQTerrace	1.08	− 0.074	− 53.21	0.35	− 0.027	− 15.27	1.58	− 0.096	− 58.76
	Cactus	1.35	− 0.104	− 43.08	0.44	− 0.029	− 13.39	1.87	− 0.131	− 54.93
	Kimono	1.04	− 0.043	− 51.27	0.28	− 0.023	− 9.41	1.43	− 0.076	− 59.47
	ParkScene	0.57	− 0.139	− 45.79	0.26	− 0.026	− 11.93	0.94	− 0.152	− 52.28
C	BasketballDrill	1.12	− 0.038	− 47.18	0.34	− 0.031	− 14.84	1.56	− 0.075	− 56.15
	BQMall	0.67	− 0.047	− 50.06	0.30	− 0.028	− 11.43	1.17	− 0.089	− 58.34
	PartyScene	0.94	− 0.052	− 38.22	0.28	− 0.027	− 16.54	1.22	− 0.081	− 45.57
	RaceHorsesC	1.22	− 0.047	− 50.23	0.41	− 0.030	− 10.23	1.71	− 0.093	− 51.49
D	BasketballPass	0.63	− 0.083	− 44.37	0.39	− 0.021	− 11.07	1.23	− 0.128	− 49.41
	BlowingBubbles	0.71	− 0.071	− 39.81	0.31	− 0.031	− 11.12	1.16	− 0.105	− 47.52
	BQSquare	0.36	− 0.062	− 34.47	0.28	− 0.029	− 12.46	0.67	− 0.091	− 43.24
	RaceHorses	0.54	− 0.042	− 36.37	0.25	− 0.035	− 9.07	0.89	− 0.074	− 42.65
E	FourPeople	0.95	− 0.034	− 46.26	0.29	− 0.034	− 11.75	1.35	− 0.087	− 54.19
	Johnny	1.06	− 0.053	− 49.39	0.32	− 0.029	− 9.62	1.47	− 0.094	− 56.57
	KristenAndSara	0.78	− 0.035	− 44.85	0.26	− 0.023	− 10.39	1.17	− 0.061	− 52.07
Average		0.84	− 0.062	− 44.71	0.31	− 0.029	− 12.63	1.26	− 0.093	− 53.29

Table 3 Coding performance of the proposed algorithm compared to FQPD and DFFPE

Class	Sequence	FQPD [26]			DFFPE [35]			Proposed FCP		
		BDBR (%)	TS (%)	TS/BDBR	BDBR (%)	TS (%)	TS/BDBR	BDBR (%)	TS (%)	TS/BDBR
A1	Tango2	0.00	0.00	0.00	1.33	− 67.02	− 50.39	1.12	− 41.61	− 37.15
	FoodMarket4	0.00	0.00	0.00	0.97	− 53.17	− 54.81	0.53	− 46.43	− 87.60
	Campfire	0.00	0.00	0.00	1.56	− 57.32	− 36.74	0.85	− 42.77	− 50.32
A2	CatRobot	0.00	0.00	0.00	1.63	− 63.18	− 38.76	0.96	− 48.02	− 50.02
	DaylightRoad2	0.00	0.00	0.00	1.23	− 62.88	− 51.12	0.47	− 41.16	− 87.58
	ParkRunning3	0.00	0.00	0.00	0.88	− 59.52	− 67.64	0.83	− 49.46	− 59.60
B	BasketballDrive	3.28	− 59.35	− 18.09	1.53	− 60.35	− 39.44	0.79	− 39.63	− 50.17
	BQTerrace	1.08	− 45.30	− 41.94	1.16	− 56.19	− 48.44	1.08	− 53.21	− 49.27
	Cactus	1.84	− 52.44	− 28.50	1.78	− 62.98	− 35.38	1.35	− 43.08	− 31.91
	Kimono	1.93	− 59.51	− 30.83	0.93	− 67.04	− 72.09	1.04	− 51.27	− 49.30
	ParkScene	1.26	− 51.84	− 41.14	1.47	− 59.66	− 40.59	0.57	− 45.79	− 80.33
C	BasketballDrill	1.82	− 48.48	− 26.64	1.99	− 48.91	− 24.58	1.12	− 47.18	− 42.13
	BQMall	1.87	− 52.47	− 28.06	2.02	− 51.22	− 25.36	0.67	− 50.06	− 74.72
	PartyScene	0.26	− 38.62	− 148.54	0.87	− 49.86	− 57.31	0.94	− 38.22	− 40.66
	RaceHorsesC	0.88	− 49.05	− 55.74	1.27	− 49.98	− 39.35	1.22	− 50.23	− 41.17
D	BasketballPass	1.95	− 47.70	− 24.46	1.54	− 43.62	− 28.32	0.63	− 44.37	− 70.43
	BlowingBubbles	0.47	− 40.35	− 85.85	0.91	− 39.74	− 43.67	0.71	− 39.81	− 56.07
	BQSquare	0.19	− 31.95	− 168.16	0.79	− 45.31	− 57.35	0.36	− 34.47	− 95.75
	RaceHorses	0.54	− 41.69	− 77.20	1.09	− 48.93	− 44.89	0.54	− 36.37	− 67.35
E	FourPeople	2.70	− 57.57	− 21.32	1.97	− 58.45	− 29.67	0.95	− 46.26	− 48.70
	Johnny	3.22	− 56.88	− 17.66	2.05	− 59.37	− 28.96	1.06	− 49.39	− 46.59
	KristenAndSara	2.78	− 55.11	− 19.82	1.90	− 58.21	− 30.64	0.78	− 44.85	− 57.50
Average		1.63	− 49.27	− 30.23	1.40	− 55.59	− 39.71	0.84	− 44.71	− 53.23

Table 4 Performance of steps A, B, and C in the fast CU partition algorithm

Class	BDBR(%)			TS(%)	
	Step A	Step B	Step C	Step A &B	Step C
A1	0.14	0.44	0.25	− 30.97	− 12.63
A2	0.12	0.51	0.12	− 33.73	− 12.48
B	0.08	0.61	0.28	− 35.41	− 11.19
C	0.07	0.54	0.38	− 35.85	− 10.57
D	0.04	0.33	0.19	− 32.27	− 6.49
E	0.14	0.45	0.34	− 28.65	− 18.18
Average	0.06	0.57	0.21	− 34.23	− 10.48

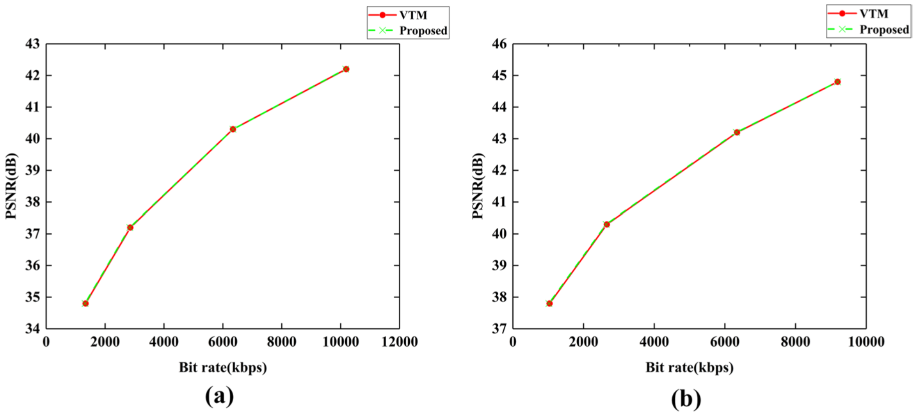


Fig. 9 The RD curve of the proposed method. **a** RD of "DaylightRoad2," **b** RD of "BQTerrace"

equal the time saved by step A; hence, step A alone did not significantly change performance. We can observe from the table that steps A and B contribute 34.23% to TS, while BDBR increases by 0.06% and 0.57%, respectively. Step C saves a further 10.48% encoding time at the cost of a 0.21% increase in BDBR. It should be noted that while each average value in this table was created by averaging the data from each sequence, the results may not equal the average value for each class.

The rate distortion (RD) curve can be used to indicate the objective evaluation of video quality. Get the PSNR/bitrate data to match the RD curve at QP = 22, 27, 32, and 37 in accordance with the specifications of the general test set CTC. In this study, bit rate and PSNR are used to evaluate the objective video quality. Therefore, the bit rate increases higher as the prediction accuracy of the existing fast method decreases. Figure 9 displays the RD performance comparison between the proposed approach and VTM-12.0 in the test video. We test and compare the video sequences "DaylightRoad2" and "BQTerrace" of the traffic scene, and the corresponding RD curves are shown in Fig. 9a, b. In comparison with VTM-12.0, it is seen that the suggested strategy can deliver consistent RD performance.

6 Conclusions

This paper introduces the CU partitioning algorithm of VVC, and a chroma intra-prediction optimization approach is proposed to achieve a low-complexity and hardware-friendly H.266/VVC encoder for VANETs. Based on the CU texture complexity and texture direction, CU division can skip unnecessary processes to reduce coding complexity. In addition, the chroma intra-prediction candidate mode list is optimized by combining the relationship between CU texture complexity and prediction modes, thereby further reducing computational complexity. Experimental results show that the method can decrease the average coding complexity of H.266/VVC encoders by up to 53.29% for VANETs, while the scheme has a reasonable loss of coding efficiency.

Acknowledgements
None.

Author contributions
XTJ devised the study plan and led the writing of the article. WL conducted the experiment and collected the data. TS conducted the analysis. All authors read and approved the final manuscript.

Funding
This work was sponsored by Shanghai Pujiang Program (22PJD028).

Availability of data and materials
The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 1 July 2023 Accepted: 20 November 2023
Published online: 27 November 2023

References

1. M. Vafaei, A. Khademzadeh, M.A. Pourmina, A new QoS adaptive multi-path routing for video streaming in urban VANETs integrating ant colony optimization algorithm and fuzzy logic. Wirel. Pers. Commun. **118**(4), 2539–2572 (2021)
2. B. Yu, C. Xu, Vehicular ad-hoc networks: an information-centric perspective. ZTE Commun. **118**(4), 2539–2572 (2021)
3. A. Torres, P. Piñol, C. T. Calafate, J.-C. Cano, P. Manzoni, Evaluating H. 265 real-time video flooding quality in highway V2V environments, in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)* (IEEE, 2014), pp. 2716–2721
4. R.F. Shaibani, A.T. Zahary, Survey of context-aware video transmission over vehicular ad-hoc networks (VANETs). EAI Endorsed Trans. Mob. Commun. Appl. **4**(15), e4–e4 (2018)
5. Z. Pan, L. Chen, X. Sun, Low complexity HEVC encoder for visual sensor networks. Sensors **15**(12), 30115–30125 (2015)
6. B. Bross et al., Overview of the versatile video coding (VVC) standard and its applications. IEEE Trans. Circuits Syst. Video Technol. **31**(10), 3736–3764 (2021)
7. J. Chen, Y. Ye, S. Kim, Algorithm description for versatile video coding and test model 12 (VTM 12), Document: JVET-U2002-v1, Brussels, 6–15 January (2021). https://jvet-experts.org/
8. Y.-J. Choi, D.-S. Jun, W.-S. Cheong, B.-G. Kim, Design of efficient perspective affine motion estimation/compensation for versatile video coding (VVC) standard. Electronics **8**(9), 993 (2019)
9. L. Shen, Z. Zhang, Z. Liu, Effective CU size decision for HEVC intracoding. IEEE Trans. Image Process. **23**(10), 4232–4241 (2014)
10. H. Yang, L. Shen, X. Dong, Q. Ding, P. An, G. Jiang, Low-complexity CTU partition structure decision and fast intra mode decision for versatile video coding. IEEE Trans. Circuits Syst. Video Technol. **30**(6), 1668–1682 (2019)
11. S. Huade, L. Fan, C. Huanbang, A fast CU size decision algorithm based on adaptive depth selection for HEVC encoder, in *Proceedings of the International Conference on Audio, Language and Image Processing* (2014), pp. 143–146
12. T. Fu, H. Zhang, F. Mu, H. Chen, Fast CU partitioning algorithm for H. 266/VVC intra-frame coding, in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)* (2019), pp. 55–60

13. M. Xu, T. Li, Z. Wang, X. Deng, R. Yang, Z. Guan, Reducing complexity of HEVC: A deep learning approach. IEEE Trans. Image Process. **27**(10), 5044–5059 (2018)
14. F. Galpin, F. Racapé, S. Jaiswal, P. Bordes, F. Léannec, E. François, CNN-based driving of block partitioning for intra slices encoding, in *Proceedings of the Data Compression Conference (DCC)* (2019), pp. 162–171
15. X. Zhang, C. Gisquet, E. Francois, F. Zou, O.C. Au, Chroma intra prediction based on inter-channel correlation for HEVC. IEEE Trans. Image Process. **23**(1), 274–286 (2013)
16. K. Zhang, J. Chen, L. Zhang, X. Li, M. Karczewicz, Multi-model based cross-component linear model chroma intra-prediction for video coding, in *2017 IEEE Visual Communications and Image Processing (VCIP)* (IEEE, 2017), pp. 1–4
17. T. Zhang, X. Fan, D. Zhao, W. Gao, Improving chroma intra prediction for HEVC, in *Proceedings of the IEEE International Conference on Multimedia & Expo Workshops (ICMEW)* (2016), pp. 1–6
18. K. Zhang, J. Chen, L. Zhang, X. Li, M. Karczewicz, Enhanced cross-component linear model for chroma intra-prediction in video coding. IEEE Trans. Image Process. **27**(8), 3983–3997 (2018)
19. P. Sharma, A. Kaul, M. Garg, Performance analysis of video streaming applications over VANETs. Int. J. Comput. Appl. **112**(14) (2015)
20. P.P.G. Abenza, M.P. Malumbres, P.P. Peral, O.López-Granado, Evaluating the use of QoS for video delivery in vehicular networks, in *Proceedings of the 29th International Conference on Computer Communications and Networks (ICCCN)* (2020), pp. 1–9
21. E. Yaacoub, F. Filali, A. Abu-Dayya, QoE enhancement of SVC video streaming over vehicular networks using cooperative LTE/802.11 p communications. IEEE J. Sel. Top. Signal Process. **9**(1), 37–49 (2014)
22. C. Rezende, A. Boukerche, M. Almulla, A.A. Loureiro, The selective use of redundancy for video streaming over Vehicular Ad Hoc Networks. Comput. Netw. **81**, 43–62 (2015)
23. W.S.M. Yousef, M.R.H. Arshad, A. Zahary, Vehicle rewarding for video transmission over VANETS using real neighborhood and relative velocity (RNRV). J. Theor. Appl. Inf. Technol. **95**(2) (2017)
24. E.B. Smida, S.G. Fantar, H. Youssef, Video streaming forwarding in a smart city's VANET, in *Proceedings of the 11th Conference on Service-Oriented Computing and Applications (SOCA)* (2018), pp. 1–8
25. H. Marzouk, A. Badri, A. Sahel, A. K. Belbachir, A. Baghdad, New simulation model for video transmissions in VANETs using EVALVID framework, in *Proceedings of the 7th Mediterranean Congress of Telecommunications (CMT)* (2019), pp. 1–5
26. Y. Fan, H. Sun, J. Katto, J. Ming, A fast QTMT partition decision strategy for VVC intra prediction. IEEE Access **8**, 107900–107911 (2020)
27. X. Zhou, G. Shi, Z. Duan, Visual saliency-based fast intracoding algorithm for high efficiency video coding. J. Electron. Imaging **26**(1), 013019 (2017)
28. Y. Shen, S. Zhang, C. Yang, Image texture based fast CU size selection algorithm for HEVC intra coding, in *Proceedings of the IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)* (2014), pp. 363–367
29. P. Mohanaiah, P. Sathyanarayana, L. GuruKumar, Image texture feature extraction using GLCM approach. Int. J. Sci. Res. Publ. **3**(5), 1–5 (2013)
30. J. Chen, V. Seregin, Chroma intra prediction by scaled luma samples using integer operations, JCTVC-C206(2010)
31. F. Duanmu, Z. Ma, Y. Wang, Fast mode and partition decision using machine learning for intra-frame coding in HEVC screen content coding extension. IEEE J. Emerg. Sel. Top. Circuits Syst. **6**(4), 517–531 (2016)
32. R. Ghaznavi-Youvalari, J. Lainema, Joint cross-component linear model for chroma intra prediction, in *Proceedings of the IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)* (2020), pp. 1–5
33. F. Bossen, J. Boyce, X. Suehing, X. Li, V. Seregin, JVET common test conditions and software reference configurations for SDR video, Document: JVET-N1010, Geneva, 19–27 March (2019). https://jvet-experts.org/
34. G. Bjontegaard, Calculation of average PSNR difference between RD-curves. in *13th VCEG-M33 Meeting, Austin, Texas* (2001)
35. T. Zhao, Y. Huang, W. Feng, Y. Xu, Efficient VVC intra prediction based on deep feature fusion and probability estimation. arXiv:2205.03587v1 (2022)
36. G. Tang, M. Jing, X. Zeng, Y. Fan, Adaptive CU split decision with pooling-variable CNN for VVC intra encoding, in *Proceedings of the IEEE Visual Communications and Image Processing (VCIP)* (2019), pp. 1–4
37. W. Li, X. Jiang, J. Jin, T. Song, F.R. Yu, Saliency-enabled coding unit partitioning and quantization control for versatile video coding. Information **13**(8), 394 (2022)
38. O. Akbulut, A new perspective on decolorization: feature-preserving decolorization. SIViP **15**(3), 645–653 (2021)
39. Q. Li, H. Meng, Y. Li, Texture-based fast QTMT partition algorithm in VVC intra coding. SIViP **17**(4), 1581–1589 (2023)
40. O. Akbulut, M.Z. Konyar, Improved intra-subpartition coding mode for versatile video coding. SIViP **16**(5), 1363–1368 (2022)

Publisher's Note

Xiantao Jiang (Member, IEEE) received an M.E. degree from Shanghai Maritime University (SMU) in 2012 and the dual Ph.D. degrees from Tongji University of China and Tokushima University of Japan in 2016. After that, he joined the College of Information Engineering at Shanghai Maritime University as an Associate Professor. He is currently a postdoc in Carleton University, Ottawa, Canada. His research interests include multimedia, deep learning, and Internet of vehicles.

Wei Li received the B.S. degree in Information Management and Information Systems from Information

Engineering College of Fuyang Normal University, Fuyang, China, in 2019. He is currently working toward the M.S. degree in Computer Science and Technology at the School of Information Engineering, Shanghai Maritime University, Shanghai, China. His research interests include video coding.

Tian Song (Member, IEEE) received his B.E. degree from Dalian University of Technology, China, in 1995 and his M.E. and Dr.E. degrees from Osaka University in 2001 and 2004, respectively. He joined Tokushima University in 2004 as an Assistant Professor. Presently, he is an Associate Professor of the Department of Electrical and Electronic Engineering, Graduate School of Advanced Technology and Science, Tokushima University. He is a member of IEICE and IEEE. His current research interests include video coding algorithms.