electronics

MDPI

Article

Fast Algorithm for CU Size Decision Based on Ensemble Clustering for Intra Coding of VVC 3D Video Depth Map

Wenjun Song, Guanxin Li and Qiuwen Zhang *

College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China; songwenjun@zzuli.edu.cn (W.S.); 332107040594@email.zzuli.edu.cn (G.L.)

* Correspondence: 2012032@zzuli.edu.cn; Tel.: +86-371-8660-9559

Abstract: As many new coding techniques and coding structures have been introduced to further improve the coding efficiency of depth maps in 3D video extensions, the coding complexity has been greatly increased. Fast algorithms are now needed to improve coding unit (CU) depth decisions as well as the coding pattern decision based on the coding. This paper presents an innovative machine learning-based approach aimed at mitigating the complexity associated with in-frame coding algorithms. We build different clustering models for different CU sizes to cluster CUs of the same size to decide their CU sizes. This is achieved by augmenting ensemble clustering through the expedited propagation of clustering similarities, considering CU with the same or similar texture complexity the same as for CU depth selection, which is informed by a comprehensive analysis of the original texture and its neighboring elements. The experimental findings demonstrate that the proposed scheme yields a substantial average reduction of 44.24% in the coding time. Remarkably, the corresponding Bjøntegaard delta bit rate (BDBR) increment observed for the synthetic view is a mere 0.26%.

Keywords: depth map; CU size decision; ensemble clustering

check for updates

Citation: Song, W.; Li, G.; Zhang, Q. Fast Algorithm for CU Size Decision Based on Ensemble Clustering for Intra Coding of VVC 3D Video Depth Map. *Electronics* **2023**, *12*, 3098. https://doi.org/10.3390/electronics12143098

Academic Editors: Radu Ciprian Bilcu and Ionut Schiopu

Received: 2 June 2023
Revised: 12 July 2023
Accepted: 13 July 2023
Published: 17 July 2023

1. Introduction

With the progressive maturation of the new-generation video coding technology versatile video coding (VVC), current coding technology research is mostly focused on two-dimensional video. For three-dimensional video, research is mostly based on 3D-HEVC, while the new, more efficient VVC technology is less used in three-dimensional video. Consequently, the primary emphasis of this paper is on the exploration and analysis of the new-generation video coding standard VVC, specifically within the domain of 3D video coding technology. This study undertook an exploration to optimize video encoding time and enhance the speed of 3D video encoding. The VVC standard introduced novel encoding techniques, notably leveraging quadtree multi-type tree (QTMT), which exhibits a significant improvement in encoding efficiency. Diverging from the quadrinomial tree (QT) partitioning structure employed in high-efficiency video coding (HEVC), the VVC standard introduces a multi-type tree (MTT) partitioning scheme, illustrated in Figure 1. The QTMT partition structure allows CU to have a square or rectangular shape, and QTMT allows further partitioning of coding tree units (CTUs) using binary tree (BT) or ternary tree (TT) on top of the QT partitioning of HEVC, resulting in a significant increase in coding complexity. Consequently, streamlining the QTMT-based CU partitioning process holds great potential for substantially curtailing the computational complexity inherent in VVC [1].

Encoding 3D video constitutes a considerably intricate system that entails concurrent encoding of multiple views. This necessitates augmented video transmission bandwidth and expanded video storage space. The video format is composed solely of a restricted number of texture videos and their associated depth maps. These texture videos and depth maps can be employed in synthesizing numerous virtual views through depth image-based rendering (DIBR) [2]. Although a greater number of views contributes to heightened immersion, it concurrently poses significant challenges concerning data storage and transmission [3].

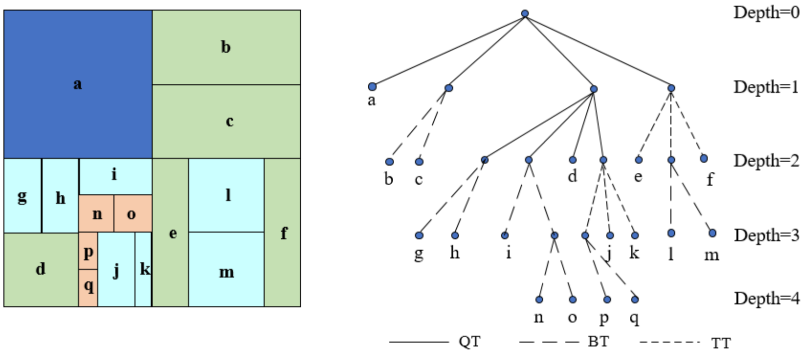


Figure 1. QTMT partitioning scheme in VVC.

The depth map serves as a means to capture and represent the geometric attributes of a given scene, utilizing grayscale to convey the spatial separation between the camera and the depicted objects. Notably distinct from texture frames, depth maps exhibit prominent characteristics by extensive homogeneous regions that are demarcated by sharp edges. In contrast, texture frames encompass intricate content comprising abrupt fluctuations in sample values. Although the depth map itself is not directly observable to the viewer, the intrinsic geometric information it encapsulates plays a pivotal role in the generation of synthetic views. The encoding process of the depth map necessitates meticulous preservation of its edge information, as the distortion of such details can result in the emergence of inaccuracies during the compositional fusion of foreground and background pixels in the synthesized view, thereby potentially altering the fundamental structure of the corresponding 3D video. Consequently, ensuring the fidelity of the encoded depth map's edge information emerges as a critical undertaking, thereby facilitating the provision of high-quality synthetic views. Particularly within homogeneous regions, coding techniques must be employed to accomplish this objective, given the human visual system's inherent limitation in discerning minute variations in depth values.

Within the context of H.266/VVC-based 3D video, this research endeavor presents a novel proposition encompassing an ensemble clustering methodology for the purpose of effectively addressing the intricacies associated with depth map coding. We consider depth map CU coding as a clustering problem and use the augmented ensemble clustering by fast propagation of clustering similarity proposed in [4] to solve this problem. The texture information and similarity function of the depth map CU are used to construct the base clusters, and then the base clustering results are aggregated and unified by the consensus function. Creating different clusters of CUs with the same or similar pixel points allows us to quickly determine the different types of multiple CUs at the same time, which facilitates the next step of CU partitioning. Our method effectively improves the coding efficiency of VVC-based 3D video depth map intra-frame coding, shortens the coding time, and replaces the complicated intra-frame coding calculation process.

The subsequent sections of this paper can be summarized as follows. Section 2 provides a comprehensive overview of recent advancements pertaining to the reduction in complexity in both VVC and 3D-HEVC frameworks. Section 3 proposes a fast CU partitioning algorithm and introduces the high-correlation feature. Section 4 describes the combined experimental results. Section 5 concludes the paper.

2. Related Works

2.1. Status of H.266/VVC Research for 2D Video

In the two algorithms proposed in the literature [5], the initial stage involves conducting pruning aimed at filtering out redundant partitioning patterns through the utilization of an algorithm. Its primary objective is to efficiently identify and eliminate superfluous patterns. The subsequent phase encompasses predictive termination, wherein a meticulous selection process is employed to identify the most suitable model from the obtained outcomes, ultimately facilitating an early conclusion of predictive partitioning. In the literature [6], video content features are counted based on three aspects: CU pixel gradient, block mean and block variance. These features are used in conjunction with CU depth and prediction patterns of temporally and spatially adjacent CUs for fast CU segmentation decisions and fast prediction. In the relevant literature [7], an adaptive approach is introduced whereby an intra-frame prediction pattern adjacent to a neighboring coding unit (CU) is leveraged to dynamically incorporate selection candidates into the rate-distortion optimization (RDO) process. Similarly, another scholarly work [8] presents a hierarchical convolutional neural network (CNN) architecture, serving as an alternative to the computationally intensive brute force method employed in the search for RDCost. Within the existing literature, a scholarly work [9] introduces an enhanced internal subdivision algorithm specifically tailored for partitioning CUs into sub-CUs based on diverse directional considerations. This algorithm further facilitates the identification of the optimal division pattern. A hierarchical CNN model incorporating early termination principles is proposed in [10], specifically for the prediction of CU partitions pertaining to 64 × 64 CUs. This model effectively bypasses the necessity for redundant partitions, thereby enhancing the overall efficiency of the process. Ref. [11] employs deep convolutional networks to extract spatiotemporal coding features by integrating reference features obtained from diverse convolutional kernels. These fused features are instrumental in determining the coding depth within a given frame. Notably, candidate partitioning patterns predicated on probabilistic models and spatiotemporal coherence are strategically employed to discern the most optimal encoding depth. The literature [12] determines the CU division direction as well as binary and ternary divisions in advance based on the directional gradient, which leads to a fast selection of CU division. In the literature [13], an early decision method is proposed based on TT characteristics, which effectively reduces the complexity of TT. One study [14] designed a fast algorithm based on a random forest classifier model and texture region features. Another study [15] proposes a lightweight adjustable QTBT partitioning scheme based on a machine learning approach using a random forest classifier to determine the partitioning method of coded blocks.

2.2. Current Research Status and Dynamic Analysis of the Complexity of Three-Dimensional Video Coding (3D-HEVC)

Ref. [16] proposed an algorithm for the early determination of motion and parallax vectors. This algorithm employs a statistical analysis-based approach to adaptively reduce the number of spatial and temporal motion vector candidates specifically for prediction units characterized by merged patterns. By implementing this reduction strategy, the algorithm effectively mitigates the complexity associated with texture view coding. Ref. [17] proposed a rapid coding scheme, leveraging the inherent properties of video content to expedite the compression process in 3D-HEVC. The algorithm capitalizes on the spatial and temporal correlations within the video content to analyze the characteristics of the CTUs. By leveraging these correlations, the algorithm selectively bypasses unnecessary coding patterns, effectively reducing computational complexity while maintaining satisfactory compression quality. In the pertinent literature [18], the early decision-making process for CU sizes is approached as a clustering problem. Subsequently, three distinct clustering models are devised specifically for CUs of sizes 64 × 64, 32 × 32, and 16 × 16, with the aim of determining if further splitting is necessary at an early stage. Notably, a similarity distance metric is introduced into the early CU size decision to ensure a balanced consideration between time-saving and coding efficiency, thereby allowing users to make informed choices. One study [19] proposes the use of effective homogeneity determination to improve previously proposed depth map PU size decisions. Other research [20] is based on a decision tree early skipping method for fixed-size CU. Another study [21] proposes an intercomponent tool that uses links to save runtime through the joint encoding of quadtrees. A fast decision-making method based on decision trees for depth graph coding was proposed in [22]. The method uses data mining and machine learning to associate encoder contextual attributes and construct a set of decision trees. One study [23] introduces two methods to reduce the complexity of 3D-HEVC when encoding blocks to be encoded, and then goes on to form a hybrid complexity reduction scheme using a dual-mode prediction method, motion information of the base texture view, and rate distortion cost information of the encoded blocks. The method proposed in [24] can adaptively use the tree block ground complexity classification to analyze the tree block encoding complexity model, thus performing fast pattern size determination and adaptive motion search range. Another study [25] applied depth modeling model 1 in depth map coding to save coding time.

3. Proposed Early CU Size Decision Algorithm

Traditionally, the decision-making process for coding unit sizes is tackled through supervised machine learning techniques. Clustering, an unsupervised machine learning approach, involves partitioning a dataset into one or multiple clusters based on shared similarity characteristics among the data points. A CU of a certain size is divided into different clusters and distinguished according to the CU in different clusters for fast coding.

3.1. Ensemble Clustering

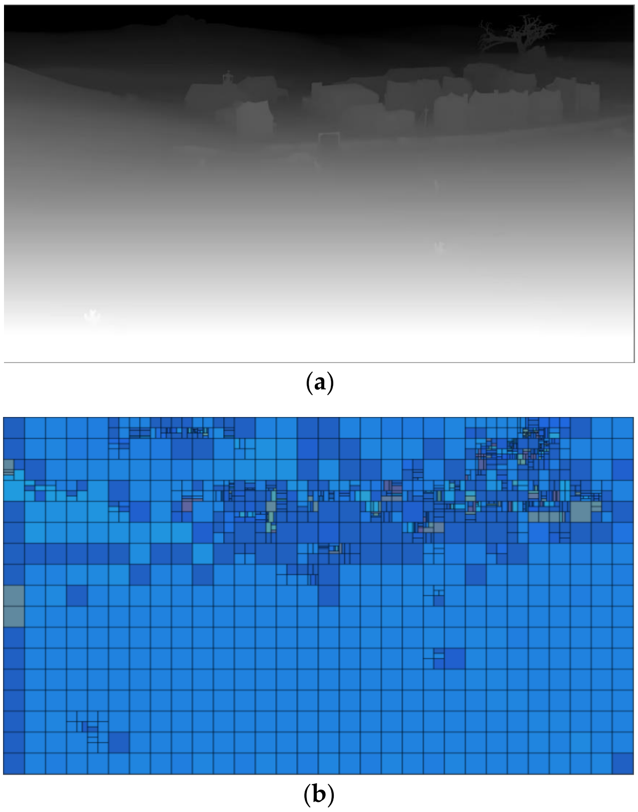
When dealing with complex data structures, indistinct boundaries, aspherical distributions, and high dimensionality, individual clustering algorithms often struggle to yield optimal clustering results. Moreover, different algorithms applied to the same dataset may produce varying clustering performances. Each clustering algorithm possesses its own strengths and weaknesses, and no single algorithm can universally address all data distributions and application scenarios. Selecting an appropriate clustering algorithm for a given problem, particularly in the absence of prior knowledge, can be challenging. Furthermore, even when employing a specific algorithm, determining the optimal parameters for the clustering task can prove arduous. Ensemble clustering algorithms provide a compelling strategy by harnessing a consistency function to proficiently integrate multiple base clustering algorithms. This approach aims to achieve improved results with enhanced robustness. The ensemble clustering algorithm has two phases. The first phase is to generate a set of base clusters with differences so that the clustering of the dataset does not appear homogeneous when there is diversity among the base cluster members. The second phase is to use a combination strategy to improve the accuracy of the clustering ensemble results.

Within ensemble clustering, the fundamental information lies in direct co-occurrence relationships among objects. Fred and Jain [26] introduced the notion of a co-association matrix to capture these relationships, representing the frequency of occurrences of two objects from multiple basic clusters within the same cluster. However, the conventional co-association matrix considers only direct co-occurrence relationships, overlooking the inclusion of comprehensive information from indirect connections within the dataset. When two objects are present within the same cluster in the basic clusters, they are deemed directly connected. On the other hand, if two objects are situated in different clusters and these clusters possess direct or indirect links to each other, the objects are considered indirectly connected.

Most ensemble clustering algorithms study object-level ensemble information and are unable to explore higher-level information in multiple base cluster ensembles. Multi-scale indirect links in base clustering are usually ignored, which may negatively affect the robustness of their consistent clustering performance. Addressing the need to enhance consistency clustering performance by incorporating higher-level ensemble information and integrating multi-scale direct and indirect connections, we have opted for an ensemble clustering method that employs fast propagation of clustering similarity through random walk. This approach serves as a solution for the early CU size decision problem. By leveraging the ensemble clustering approach, we delve into the abundant information available in the ensemble at the basic clustering level, utilizing multi-scale ensemble techniques and mapping strategies to analyze clustered objects. This enables us to effectively and efficiently tackle the aforementioned challenging problems. Karl Peason (1905) first introduced the term “random walk” to create a mathematical model of random trajectories, which is relevant to our daily life. For example, the diffusion process of colored liquid drops in water, the propagation of gas molecules in the air, and the trajectory of molecules from one place to another can be regarded as random walk models. Random wandering represents a dynamic process that moves from one node to another with a certain transfer probability each time it moves, with randomness and uncertainty. Since each time it moves with a certain transfer probability, the wanderer keeps repeating the process and eventually obtains a probability distribution. Based on the random wanderer model and improved models, a major breakthrough has been made in solving the problems of node ordering, clustering, classification, graph theory, and topological correlation, etc.

3.2. Feature Selection

The relationship between CU block size and its content is widely recognized, and it is observed that depth maps often contain numerous flat regions comprising pixels with similar or even identical values. As a result, the utilization of larger blocks in the depth map is significantly more probable than the adoption of smaller blocks. Figure 2 shows the depth map and CU division of the “GhostTownFly” video frame. We can see that the depth map has a large number of areas at the bottom and a small number of areas at the top as smooth areas, corresponding to a CU size of 64 × 64 at the corresponding position in Figure 2b, and the division method is not used. Only the middle of the figure is in the edge position, and the CU in the edge region in the corresponding Figure 2b continues to be divided; the details are gradually highlighted. Therefore, using an exhaustive partition size search method will generate a lot of coding time to check unnecessarily smaller CUs, which leads to a large computational load. Therefore, it is necessary to terminate CU partitioning early.



(**a**)

(**b**)

Figure 2. (**a**) Video sequence “GhostTownFly” video frame depth map; (**b**) its CU division method.

To enhance the performance of clustering analysis, it is crucial to devise a method for discerning whether the present CU corresponds to a flat region, while also eliminating irrelevant and redundant features. In this regard, we have opted to employ the Roberts gradient sum as a feature. The Roberts operator, also referred to as the cross-differencing algorithm, utilizes local differencing calculations to detect edge lines. The Roberts gradient can be defined as follows:

$$\nabla_x f = f(i,j) - f(i+1,j+1) \tag{1}$$

$$\nabla_y f = f(i,j+1) - f(i+1,j) \tag{2}$$

$$R(i,j) = |\nabla_x f| + \left|\nabla_y f\right| \tag{3}$$

A small pixel gradient sum in the current CU indicates that the depth values in this CU are almost the same, indicating that the pixel difference in this region is not large, so this CU should be terminated early for division. The current CU gradient sum is defined as:

$$S = \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} R(i,j) \tag{4}$$

The height and width of the current CU are denoted by *H* and *W*, respectively, while *R(i, j)* represents the gradient value of the pixel at coordinates (*i*, *j*) within the current CU.

It is important to note that distinct thresholds are applied to different CU sizes, and even for the same CU, varying quantization parameters (QPs) result in diverse content details. Consequently, the gradient sum varies with different CU sizes and QPs.

3.3. Clustering for Early CU Size Decision

Ensemble clustering involves the integration of multiple base clusters to achieve a more consistent and improved clustering outcome. Consider a dataset $X = \{x_1, \ldots, x_n\}$ consisting of n objects, where x_i represents the ith data object. Let $\Pi = \{\pi^1, \ldots, \pi^m\}$ represent a set of M basic clusters derived from the dataset, with $\pi^m = \left\{C_1^m, \ldots, C_{n^m}^m\right\}$ representing the mth cluster. For clarity of expression, we denote the set of all clusters as $\mathcal{C} = \left\{C_1, \ldots, C_{N_C}\right\}$, where C_i is the ith cluster

In CTU, the CU depth is divided into {0, 1, 2, 3, 4}. The optimal CU size in the coding region can be its own size, i.e., the CU texture is smooth or simple enough not to need further division, or it can be divided into smaller CUs, i.e., the CU texture is complex at the edge position. We set up several different clustering models for 64 × 64, 32 × 32, 16 × 16, and 8 × 8 coding regions for analysis. If the image pixels in the same-size CU are all very similar, they can be considered as one cluster, and the CUs in this cluster do not need to be further split. Instead, further division of the current CU is required to determine the optimal CU size.

Formally, the Jaccard [27] coefficient between two sets, denoted as C_x and C_y, is calculated as follows:

$$Jaccard\left(C_x, C_y\right) = \frac{\left|C_x \cap C_y\right|}{\left|C_x \cup C_y\right|} \tag{5}$$

In this graph, each cluster serves as a node, and the Jaccard coefficient is defined as follows:

$$\mathcal{G} = (\mathcal{V}, \varepsilon) \tag{6}$$

Here, $\mathcal{V}$ = C represents the set of nodes in graph $\mathcal{G}$, while ε represents the set of edges. The weights of the edges connecting two nodes are defined as follows:

$$e_{xy} = Jaccard\left(C_x, C_y\right) \tag{7}$$

Once the initial similarity graph is constructed, additional multi-scale information is incorporated to augment the clustering similarity. To determine the probability of transferring from one node to another by random walk, the transfer probability matrix P is constructed:

$$P_{xy} = \begin{cases} \frac{e_{xy}}{\sum_{C_k \neq C_x} e_{xk}}, & if\ x \neq y \\ 0, & if\ x = y \end{cases} \tag{8}$$

The probability from node C_x to C_y, denoted as P_{xy}, is proportional to the edge weights connecting them. By employing random walk trajectories with various step sizes, the clustering similarity is further refined. Subsequently, a new similarity measure is obtained for each pair of nodes by considering the similarity observed during the nodes' random walk.

The new similarity matrix between all clusters is represented as follows:

$$\mathrm{Z} = \{\mathrm{z}_{ij}\}_{N_C \times N_C} \tag{9}$$

The cosine similarity is used here. The cosine similarity can be expressed in terms of Z_{ij}.

In the context of ensemble clustering, the traditional co-association matrix utilizes a data structure to represent object similarity. The pairwise relationship of the mth base cluster can be represented by the connectivity matrix. The definition of this similarity is as follows:

$$A^m = \left\{a_{ij}^m\right\}_{N \times N} \tag{10}$$

$$a_{ij}^m = \begin{cases} 1,\ if\ Cls^m(x_i) = Cls^m(x_j), \\ 0, \qquad\qquad otherwise, \end{cases} \tag{11}$$

Cls^m*(x*$_i$*)* denotes the class cluster to which x_i belongs in the base clustering member π^m.

$$A = \frac{1}{M} \sum_{m=1}^{M} A^m \tag{12}$$

The above equation is the traditional covariance matrix of the whole system.

While the traditional covariance matrix captures the frequency of two objects appearing together in multiple base clusters, it treats each cluster as an independent entity. Although it leverages object co-occurrence relationships to extract cluster information, this approach limits the potential for enhancing object connectivity through the utilization of rich information available for further refinement. For this purpose, we use the enhanced connectivity matrix (ECA) matrix to extract similarities in clustering on multiple scales.

The ECA for each individual base cluster is constructed by utilizing the clustering similarity matrix.

$$B^m = \left\{b_{ij}^m\right\}_{N \times N} \tag{13}$$

$$b_{ij}^m = \begin{cases} 1,\ if\, Cls^m(x_i) = Cls^m(x_j) \\ z_{uv},\ if\, Cls^m(x_i) \neq Cls^m(x_j) \end{cases} \tag{14}$$

By constructing the enhanced connectivity matrix for each basic cluster, the resulting covariance matrix encompasses the entire system. The definitions are as follows:

$$B = \frac{1}{M} \sum_{m=1}^{M} B^m \tag{15}$$

The ECA matrix can be used for any consistency function based on the concatenation matrix.

For coded frames, we divide them into their corresponding datasets according to 64 × 64, 32 × 32, 16 × 16, and 8 × 8, for example, $D_{64\times64}$ = {n_1, n_2, ... , n_m}, which can be divided into two clusters by the above similarity metric process. One of the clusters denotes no division of CU, which is denoted as C_n. The other cluster denotes division of CU, which is denoted as C_s. Similarly, the other N × N CU is similar to the clustering process. Typically, as the sample data approach the cluster center, the likelihood of being assigned to a particular cluster increases. Figure 3 illustrates the flowchart of the algorithm.

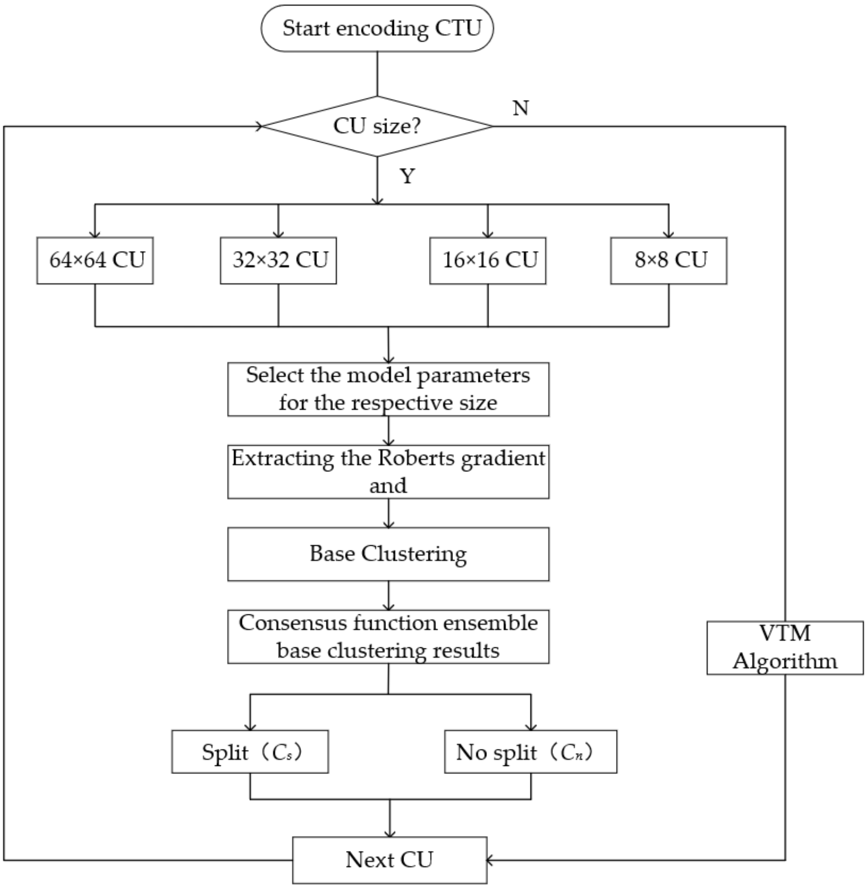

Start encoding CTU
CU size?
N
Y
64×64 CU
32×32 CU
16×16 CU
8×8 CU
Select the model parameters for the respective size
Extracting the Roberts gradient and
Base Clustering
Consensus function ensemble base clustering results
VTM Algorithm
Split（Cₛ）
No split（Cₙ）
Next CU

Figure 3. Algorithm flowchart.

3.4. Consensus Functions

The proposed ensemble clustering framework utilizes the consensus function to derive the final consistent clusters. Initialize the data as:

$$R^{(0)} = \left\{R_1^{(0)}, \ldots, R_n^{(0)}\right\} \tag{16}$$

where $R_i^{(0)} = \{x_i\}$ denotes the ith initial region and the initial region set similarity matrix is defined as:

$$S^{(0)} = \left\{s_{ij}^{(0)}\right\}_{N \times N} \tag{17}$$

$$s_{ij}^{(0)} = b_{ij} \tag{18}$$

Having obtained the initial set of regions and their respective similarity matrices, an iterative process of region merging is initiated. Starting with an initial number of regions N, after N − 1 iterations, all objects are merged into a root region. Here, we choose to perform N − 2 iterations, iterate the initial number of regions N to the number of regions 2, cluster all Cus of a certain size into two clustering results, and finally output C_S and C_n.

4. Experimental Results

To test the proposed method for intra-frame coding of VVC depth maps in 3D video, we used VTM10.0 to test it. We used the commonly used Common Test Conditions file. The dataset consists of eight test sequences, each encoded with different quantization parameters. Simulations were conducted for each test sequence using both the three-view case and random access configurations. The coding efficiency is evaluated by measuring the Bjøntegaard delta bit rate (BDBR) of the synthetic view and the time saved during depth map coding. The time savings for depth map coding (*Ts*) are calculated using the following formula:

$$T_S = \frac{T_0 - T_1}{T_0} \tag{19}$$

In the given context, "T_0" represents the total encoding time of the original method, while "T_1" represents the total encoding time of the proposed method.

The experimental results are given in Table 1. These show that the algorithm saves 44.24% of time on average from the full-frame intra-configuration results, and the BDBR increases only by 0.26%. The minimum time saving is 35.33% for the video sequence "Kendo" and the maximum time saving is 51.58% for the video sequence "GTFly". The small amount of BDBR added kept the encoding performance largely unchanged, but the encoding time was greatly reduced. In the case of random access configurations, the algorithm demonstrates reduced coding time without significantly compromising RD performance across all tested sequences. The proposed depth map intra-frame prediction method effectively reduces unnecessary computational costs, leading to time savings in video coding through the utilization of a more efficient prediction model.

Table 1. Experimental results of the proposed algorithm in the case of full intra-frame configuration and random access configuration.

Sequence	All Intra Case		Random Access Case	
	BDBR (%)	T_S (%)	BDBR (%)	T_S (%)
Balloons	0.32	36.18	0.18	3.94
Kendo	0.48	35.33	0.07	3.82
Newspaper1	0.12	48.07	0.13	5.14
GTFly	0.15	51.58	0.04	6.07
PonznanHall2	0.35	50.01	0.08	5.76
PonznanStreet	0.36	49.74	0.33	6.75
UndoDancer	0.08	47.51	0.13	4.36
Shark	0.23	35.47	0.11	7.02
Average	0.26	44.24	0.13	5.36

Table 2 shows that the texture complexity presented by the intra-frame CU differs for different depth map QPs, which in turn reflects the different reduction times of the algorithm for their CU division.

Table 2. Reduction rate of coding time for different QPs.

Depth Map QPs	Overall Time Reduction	
	All Intra Case	Random Access Case
34	36.59%	3.85%
39	37.80%	3.94%
42	42.47%	4.59%
45	51.01%	5.76%

As shown in Table 3, we compare the experimental results of the proposed method with other related work schemes, e.g., fast decision-making for 3D-HEVC depth map coding units using two-dimensional entropy and variance [28]. The proposed method achieved a greater reduction in encoding time than [28], the increments of BDBR were also significantly lower than those in [28], with some advantages indicating better encoding performance for the proposed algorithm compared to the method presented in [28]. The proposed method exhibits superior time efficiency compared to [28] for depth map intra-frame encoding [29]. Analysis of depth map tree block features using Bayesian decision rules and correlation of neighboring tree blocks. Comparing the performance to the experimental results in [29], it exhibits similar scaling time during encoding. However, the increment in BDBR achieved by the proposed method is smaller than that reported in [29], rendering the proposed algorithm more efficient for encoding. In [30], tensor features were extracted to represent the uniformity of the CU, and then the extracted features were analyzed to find a suitable threshold. Again, the time-saving rate and the increased BDBR in the results of the proposed algorithm are better than the experimental results in [30].

Table 3. Comparison of coding performance with other related jobs.

Sequences	Yao [28]		D [29]		Bakkouri [30]		Proposed	
	BDBR (%)	*Ts* (%)	BDBR (%)	*Ts* (%)	BDBR (%)	*Ts* (%)	BDBR (%)	*Ts* (%)
Balloons	0.20	32.60	1.02	51.70	0.21	38.24	0.32	36.18
Kendo	1.00	36.50	1.09	52.20	0.31	37.54	0.48	35.33
Newspaper	0.90	24.40	1.21	49.80	0.55	35.01	0.12	48.07
GTFly	0.40	31.30	0.93	52.80	0.41	40.28	0.15	51.58
PonznanHall2	−1.70	51.10	0.65	57.10	0.21	35.92	0.35	50.01
PonznanStreet	0.30	32.90	0.85	53.40	0.25	35.01	0.36	49.74
UndoDancer	1.10	43.70	0.25	45.90	0.25	34.14	0.08	47.51
Shark	1.50	41.40	0.65	46.50	0.42	40.89	0.23	35.47
Average	0.46	36.74	0.83	51.2	0.33	37.13	0.26	44.24

Figure 4 visualizes the comparison with the experimental results in [28] using a single video sequence as the basic unit and time saving as a metric. The proposed algorithm still performs relatively well in terms of time saving with BDBR lower than [28]. The video sequences "Newspaper" and "GTFly" have more obvious time-saving advantages, and most of them also perform better among different kinds of video sequences.

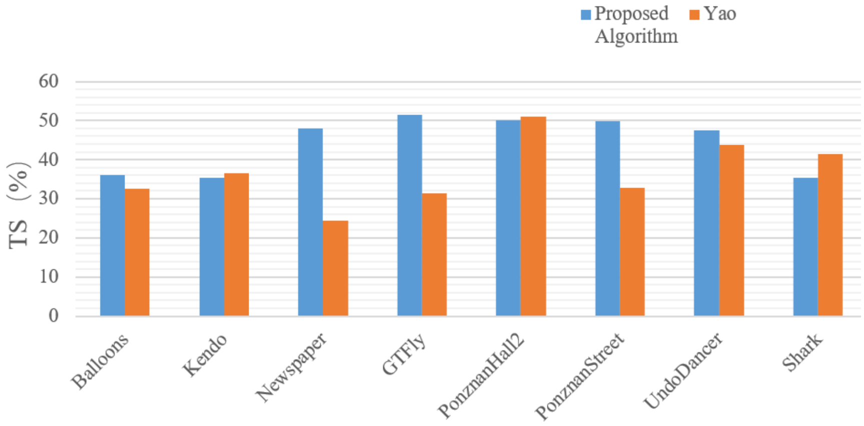
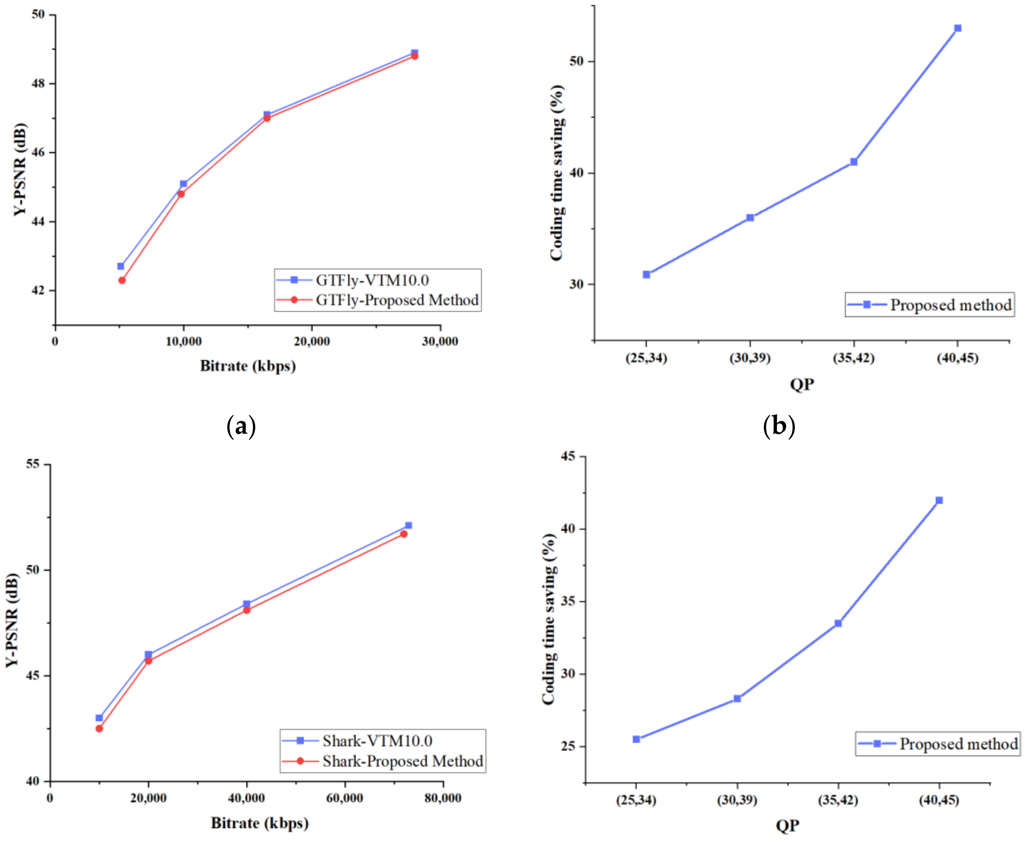


Figure 4. Comparison of the time-saving rate of the algorithm in this paper with that of the literature [28].

In Figure 5, the RD curves of the two video sequences and the time-saving curves at different QPs are given. From the figure, we can see that the RD performance of the algorithm in this paper is almost the same as that of VTM10.0, and the average time-saving rate is also good. When encoding the same video sequence with different QPs, the larger the QP of the video sequence is, the more obvious the time-saving effect is.



(**a**) (**b**)

(**c**) (**d**)

Figure 5. Experimental results of video sequences under different quantization parameters (QPs): (**a**) RD curve of "GTFly", (**b**) time-saving rate curve of "GTFly", (**c**) RD curve of "Shark", (**d**) time-saving rate curve of "Shark".

5. Conclusions

We propose an ensemble clustering approach to make fast CU decisions for different sizes of CUs, thus speeding up the rate distortion optimization cost process and partially replacing the algorithm in the encoder, saving coding time and improving coding efficiency. In the clustering process, we used the gradient sum to characterize the data, designed different clustering models for different sizes of CUs, used the similarity measure to measure whether the CUs had similar characteristics in the case of the same-size CU, and finally integrated the clustering results through the consensus function to obtain the final results of the division of CUs. On average, the proposed algorithm achieved a time saving of 44.24%. Additionally, the Bjøntegaard delta bit rate (BDBR) for the synthetic view showed a negligible increase of merely 0.26%.

Author Contributions: Conceptualization, W.S. and G.L.; methodology, W.S.; validation, W.S., Q.Z. and G.L.; formal analysis, G.L.; investigation, G.L.; resources, Q.Z.; data curation, G.L.; writing—original draft, G.L.; writing—review and editing, W.S.; visualization, W.S.; supervision, Q.Z.; project administration, Q.Z.; funding acquisition, Q.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (61771432 and 61302118), the Basic Research Projects of the Education Department of Henan (21zx003 and 20A880004), the Key Projects Natural Science Foundation of Henan (232300421150), the Scientific and Technological Project of Henan Province (232102211014), and the Postgraduate Education Reform and Quality Improvement Project of Henan Province (YJS2021KC12, YJS2023JC08, and YJS2022AL034).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pan, Z.; Zhang, P.; Peng, B.; Ling, N.; Lei, J. A CNN-Based Fast Inter Coding Method for VVC. *IEEE Signal Process. Lett.* **2021**, *28*, 1260–1264. [CrossRef]
2. Gu, K.; Qiao, J.; Lee, S.; Liu, H.; Lin, W.; Le Callet, P. Multiscale Natural Scene Statistical Analysis for No-Reference Quality Evaluation of DIBR-Synthesized Views. *IEEE Trans. Broadcast.* **2020**, *66*, 127–139. [CrossRef]
3. Liu, C.; Jia, K.; Liu, P. Fast Depth Intra Coding Based on Depth Edge Classification Network in 3D-HEVC. *IEEE Trans. Broadcast.* **2022**, *68*, 97–109. [CrossRef]
4. Huang, D.; Wang, C.-D.; Peng, H.; Lai, J.; Kwoh, C.-K. Enhanced Ensemble Clustering via Fast Propagation of Cluster-Wise Similarities. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 508–520. [CrossRef]
5. Dong, X.; Shen, L.; Yu, M.; Yang, H. Fast Intra Mode Decision Algorithm for Versatile Video Coding. *IEEE Trans. Multimed.* **2022**, *24*, 400–414. [CrossRef]
6. Yang, Z.; Shao, Q.; Guo, S. Fast Coding Algorithm for HEVC Based on Video Contents. *IET Image Process.* **2017**, *11*, 343–351. [CrossRef]
7. Zhou, M.; Wei, X.; Jia, W.; Kwong, S. Joint Decision Tree and Visual Feature Rate Control Optimization for VVC UHD Coding. *IEEE Trans. Image Process.* **2023**, *32*, 219–234. [CrossRef]
8. Xu, M.; Li, T.; Wang, Z.; Deng, X.; Yang, R.; Guan, Z. Reducing Complexity of HEVC: A Deep Learning Approach. *IEEE Trans. Image Process.* **2018**, *27*, 5044–5059. [CrossRef]
9. Akbulut, O.; Konyar, M.Z. Improved Intra-Subpartition Coding Mode for Versatile Video Coding. *SIViP* **2022**, *16*, 1363–1368. [CrossRef]
10. Abdallah, B.; Belghith, F.; BenAyed, M.A.; Masmoudi, N. Low-Complexity QTMT Partition Based on Deep Neural Network for Versatile Video Coding. *SIViP* **2021**, *15*, 1153–1160. [CrossRef]
11. Zhao, T.; Huang, Y.; Feng, W.; Xu, Y.; Kwong, S. Efficient VVC Intra Prediction Based on Deep Feature Fusion and Probability Estimation. *IEEE Trans. Multimed.* **2022**, 1–11. [CrossRef]
12. Cui, J.; Zhang, T.; Gu, C.; Zhang, X.; Ma, S. Gradient-Based Early Termination of CU Partition in VVC Intra Coding. In Proceedings of the 2020 Data Compression Conference (DCC), Snowbird, Utah, USA, 24–27 March 2020; IEEE: Snowbird, UT, USA, 2020; pp. 103–112.
13. Park, S.-H.; Kang, J.-W. Context-Based Ternary Tree Decision Method in Versatile Video Coding for Fast Intra Coding. *IEEE Access* **2019**, *7*, 172597–172605. [CrossRef]
14. Zhang, Q.; Wang, Y.; Huang, L.; Jiang, B. Fast CU Partition and Intra Mode Decision Method for H.266/VVC. *IEEE Access* **2020**, *8*, 117539–117550. [CrossRef]
15. Amestoy, T.; Mercat, A.; Hamidouche, W.; Menard, D.; Bergeron, C. Tunable VVC Frame Partitioning Based on Lightweight Machine Learning. *IEEE Trans. Image Process.* **2020**, *29*, 1313–1328. [CrossRef]
16. Pan, Z.; Yi, X.; Chen, L. Motion and Disparity Vectors Early Determination for Texture Video in 3D-HEVC. *Multimed. Tools Appl.* **2020**, *79*, 4297–4314. [CrossRef]
17. Zhang, Q.; Wang, Y.; Huang, L.; Wei, T.; Su, R. Fast Coding Scheme for Low Complexity 3D-HEVC Based on Video Content Property. *Multimed. Tools Appl.* **2021**, *80*, 25909–25925. [CrossRef]
18. Li, Y. Tunable Early CU Size Decision for Depth Map Intra Coding in 3D-HEVC Using Unsupervised Learning. *Digit. Signal Process.* **2022**, *123*, 103448. [CrossRef]
19. Hamout, H.; Elyousfi, A. A Computation Complexity Reduction of the Size Decision Algorithm in 3D-HEVC Depth Map Intracoding. *Adv. Multimed.* **2022**, *2022*, 1–12. [CrossRef]
20. Fu, C.-H.; Chen, H.; Chan, Y.-L.; Tsang, S.-H.; Hong, H.; Zhu, X. Fast Depth Intra Coding Based on Decision Tree in 3D-HEVC. *IEEE Access* **2019**, *7*, 173138–173147. [CrossRef]
21. Mora, E.G.; Jung, J.; Cagnazzo, M.; Pesquet-Popescu, B. Initialization, Limitation, and Predictive Coding of the Depth and Texture Quadtree in 3D-HEVC. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *24*, 1554–1565. [CrossRef]
22. Moura, C.; Saldanha, M.; Sanchez, G.; Marcon, C.; Porto, M.; Agostini, L. Fast Intra Mode Decision for 3D-HEVC Depth Map Coding Using Decision Trees. In Proceedings of the 2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Glasgow, UK, 23–25 November 2020; IEEE: Glasgow, UK, 2020; pp. 1–4.
23. Tohidypour, H.R.; Pourazad, M.T.; Nasiopoulos, P. Online-Learning-Based Complexity Reduction Scheme for 3D-HEVC. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 1870–1883. [CrossRef]
24. Zhang, Q.; Huang, K.; Wang, X.; Jiang, B.; Gan, Y. Efficient Multiview Video plus Depth Coding for 3D-HEVC Based on Complexity Classification of the Treeblock. *J. Real-Time Image Proc.* **2019**, *16*, 1909–1926. [CrossRef]
25. Saldanha, M.; Zatt, B.; Porto, M.; Agostini, L.; Sanchez, G. Solutions for DMM-1 Complexity Reduction in 3D-HEVC Based on Gradient Calculation. In Proceedings of the 2016 IEEE 7th Latin American Symposium on Circuits & Systems (LASCAS), Florianopolis, Brazil, 28 February–2 March 2016; IEEE: Florianopolis, Brazil, 2016; pp. 211–214.

26. Fred, A.L.N.; Jain, A.K. Combining Multiple Clusterings Using Evidence Accumulation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 835–850. [CrossRef]
27. Levandowsky, M.; Winter, D. Distance between sets. *Nature* **1971**, *234*, 34–35. [CrossRef]
28. Yao, W.; Huang, H.; Wu, Z.; Zhang, H. Intra-Frame Fast Coding Algorithm for 3D-HEVC Depth Map Based on Two-Dimensional Entropy and Variance. In Proceedings of the 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 March 2021; IEEE: Chongqing, China, 2021; pp. 889–894.
29. Zou, D.; Dai, P.; Zhang, Q. Fast Depth Map Coding Based on Bayesian Decision Theorem for 3D-HEVC. *IEEE Access* **2022**, *10*, 51120–51127. [CrossRef]
30. Bakkouri, S.; Elyousfi, A. Machine Learning-Based Fast CU Size Decision Algorithm for 3D-HEVC Inter-Coding. *J. Real-Time Image Proc.* **2021**, *18*, 983–995. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.