IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 6 May 2024, accepted 30 May 2024, date of publication 3 June 2024, date of current version 11 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3408898

RESEARCH ARTICLE

CU Split Method Based on Adaptive CNN and Gradient Matrix for VVC 3D Video Depth Map

LINA SI, AOHUI YAN, AND QIUWEN ZHANG, (Member, IEEE)
College of Computer Science and Technology, Zhengzhou University of Light Industry, Zhengzhou 450002, China

Corresponding author: Aohui Yan (1002867167@qq.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61771432 and Grant 61302118, in part by the Basic Research Projects of Education Department of Henan under Grant 21zx003, in part by the Key Projects Natural Science Foundation of Henan under Grant 232300421150, in part by the Zhongyuan Science and Technology Innovation Leadership Program under Grant 244200510026, in part by the Scientific and Technological Project of Henan Province under Grant 232102211014 and Grant 232102211017, and in part by the Postgraduate Education Reform and Quality Improvement Project of Henan Province under Grant YJS2023JC08.

ABSTRACT In the digital age, people's demand for 3D videos is becoming increasingly strong. Nowadays, the 3D-HEVC video coding standard is far from meeting people's needs. Compared to HEVC, Versatile Video Coding (VVC) exhibits better encoding performance. Depth maps are a critical part of 3D video, but current research on VVC has only been limited to texture videos. Accordingly, to diminish the computational complicacy of depth map intra coding block division in VVC 3D video, a fast approach for VVC depth map coding based on texture characteristics and deep learning is presented in this paper. We first use gradient matrix to classify CUs into simple CUs, fuzzy CUs, and complex CUs. Simple CUs can terminate their partitioning process in advance, while fuzzy CUs use the original encoder algorithm. For complex CUs, we designed two adaptive CCNs that can serve multiple sizes of CUs. The first CNN model is used to determine whether a square CU performs quadtree partitioning or multi type tree partitioning. The second CNN model is used to determine whether a CU that definitely performs multi type tree partitioning performs horizontal or vertical tree partitioning. It is clear that the second model is complementary to the first. The experimental results indicate that this scheme can achieve an average reduction of 45.35% in coding time, while BDBR only increases by 0.23%, which is superior to existing technologies in reducing encoding complexity and ensuring encoding quality.

INDEX TERMS Depth map, early termination, VVC 3D video, CNN.

I. INTRODUCTION

Today, the advancement of video coding technology has ushered in a new era of higher quality multimedia videos [1], changing the way we experience visual media. From early black and white televisions to high-definition 2D video streams that dominate today's digital video, video encoding technology continues to improve to meet the growing needs of the general public and various industries. However, as our demand for immersive and engaging visual experiences continues to grow, the limitations of traditional 2D videos have become increasingly apparent. Its drawback is that it cannot convey the depth information present in real scenes [2], leading to the planarization of scenes in videos. It is not difficult to imagine that the lack of depth information makes it difficult for viewers to experience an immersive visual experience. Whether it is the daily entertainment needs of modern people or the needs of high-end technology, 3D videos play an important role, such as the added viewpoints and depth information in 3D videos, which are crucial in applications such as telemedicine, virtual tourism, and interactive games. Therefore, 3D video has gained the attention

The associate editor coordinating the review of this manuscript and approving it for publication was Massimo Cafaro.

and love of the public and many people in the technology field. The demand for more immersive visual experiences has driven the development of 3D video technology. Technically speaking, compared to 2D videos, 3D videos contain more viewpoints and corresponding depth information [3], which can effectively display the original real scene and create a more realistic and attractive viewing experience for viewers. However, at the same time, this also requires larger storage space to store 3D videos and faster data transmission speed to achieve the application of 3D videos. Compared with traditional 2D videos, the transmission and application of 3D videos pose greater challenges to video compression efficiency. Therefore, fast encoding of 3D videos has become one of the hotspots in current video research.

Traditional 2D videos cannot meet people's requirements for immersive visual experiences, to this end, in 2015, the 3D video encoding standard HEVC was jointly released by the Moving Picture Experts Group and the Video Coding Expert Group. 3D-HEVC employs some new coding tools especially for depth maps coding, such as depth modeling mode (DMM) [4], segment-wise DC coding (SDC) [5], depth intra skip (DIS) [6], and view synthesis optimization (VSO) [7], etc. These new coding techniques in 3D-HEVC improve the compression efficiency and bring the superior perceptual quality of synthesized views. With the rapid development of multimedia technology, in July 2020, the Joint Video Experts Team (JVET) released the VVC standard [8], which is committed to applying in application scenarios where previous generation standard HEVC cannot meet the requirements, such as 4K videos and 360-degree video. VVC defines 65 different basic intra prediction directions for brightness prediction blocks, which is equivalent to adding one direction between every two directions of 33 predicted directions in the HEVC frame. With the addition of Planar and DC modes, there are a total of 67 prediction modes. In addition, VVC adopts a new and more flexible partitioning structure, namely the Quad-tree with Nested Multi-type Tree (QTMT), which greatly increases coding efficiency. Although the structure of the VVC video encoding layer remains the traditional block-based hybrid video encoding mode, VVC provides multiple advanced video encoding tools, with a compression rate approximately double that of the previous HEVC standard.

Currently, 3D videos generally use the popular Multi view Video Plus Depth (MVD) format, which requires encoding and transmitting texture and depth maps from three different viewpoints. Therefore, compared to 2D videos, this encoding method will bring a huge amount of data. Depth maps are actually grayscale images, where the grayscale values represent the distance between the object and the camera in the scene [9]. However, the color, pattern, and brightness of the object itself are not reflected in depth map. Therefore, it has less edge information than texture maps. For example, in depth maps, edges are generated by the relative distance difference between the entity and the background in the scene, while in the corresponding texture map, this edge usually also appears unless the color information of both, that is, the entity and its surrounding background, are completely consistent [10]. In texture maps, an entity or background contains a lot of texture information, while in depth maps, the same object or background is often very flat and smooth, with smaller textures. The texture differences between the two types of images are reflected in Figure 1. There are two contrasting texture regions in the depth map. For the segmentation of coding units in the depth map, the segmentation depth in the flat area is often lower, while in the sharp edge area, the encoder tends to choose small size blocks for encoding. Based on this, the traditional CU partitioning method for texture maps is not suitable for depth maps [11]. In VVC 3D Video depth map coding, CU partitioning adopts the QTMT partitioning structure. In the original algorithm of the intra encoding unit, regardless of whether the encoded region is flat or complex, all possible partitioning methods need to be traversed. Due to the new QTMT partitioning structure, compared to 3D-HEVC depth map encoding, this will result in greater computational complexity. Currently, there are a large number of fast encoding algorithms applied to VVC 2D Video, but few are targeted at 3D videos. Therefore, designing a low complexity coding unit partitioning approach for VVC 3D Video depth map based on the texture characteristics of depth maps to replace or partially replace the original algorithm will greatly reduce encoding time and facilitate the further development of 3D video.

(a) (b)

FIGURE 1. A texture and depth map in the Poznan_Street video sequence.

In response to the problem of high computational complicacy in depth map intra frame encoding under QTMT partitioning technology, this paper aims to accelerate the decision-making process of depth map CU partitioning. To this end, we propose an algorithm that combines traditional manual methods and deep learning methods to accelerate depth map encoding. It can be summarized as the following two points. (1) Use gradient matrix to divide CU into simple CU, fuzzy CU, and complex CU. Simple CU can terminate its partitioning process in advance, thus avoiding some unnecessary Rate-Distortion Optimization (RDO) processes. Fuzzy CU uses the original encoder algorithm. For complex CU, we designed two adaptive CCNs to accelerate its partitioning process. (2) We propose two adaptive Convolutional Neural Network (CNN)s that can serve multiple sizes of CUs. The first CNN model is used to determine whether CUs perform quadtree partitioning or multi type tree partitioning on square CUs. The second CNN model is used to determine whether CU performs horizontal or vertical

tree partitioning for CU that definitely performs multi type tree partitioning, thus replacing some of the original RDO calculations and achieving the purpose of this paper.

The structure of the following four chapters in the article is as follows. Chapter 2 introduces some methods for accelerating 3D-HEVC encoding and VVC 2D texture map encoding. Chapter 3 provides a detailed description of the method proposed in this article. Chapter 4 presents the experimental results of our algorithm. Chapter 5 summarizes this paper.

II. RELATED WORKS

Nowadays, a number of approaches have been proposed to accelerate the block partition for 3D-HEVC and VVC 2D Video.

A. APPROACHES FOR 3D-HEVC

There are currently many mature methods available to accelerate 3D-HEVC block partitioning. Li et al. [12] used an unsupervised learning method to accelerate the CU partitioning process of depth maps. The algorithm regards the decision of CU division as a clustering problem, and only uses a feature, namely RD cost. In general, the method uses three K-means clustering methods to determine 64 × 64, 32 × 32 and 16 × 16 size CU continues to be divided or terminated in advance. In addition, a variable similarity distance is introduced so that users can flexibly adjust the balance between video quality and complicacy reduction. Hamout and Elyousf [13] proposed a clustering algorithm to diminish the computational complicacy of CU partitioning within depth map frames. The method needs to calculate the sample of simplified mass center (ASMCV) and the variance, and compare them with four thresholds to construct a CU size decision model. In [14], the authors use deep learning methods to speed up the decision-making process of 3D-HEVC depth map CU quadtree partitioning patterns. Firstly, the authors constructed a database of depth map partitioning information and ultimately used this data to create a Multi Deep Convolutional Neural Network (MD-CNN) model. This model can directly predict the partitioning mode of CU without requiring time-consuming RDO process, thus achieving the goal of reducing encoding complexity. Reference [15] proposed a method to simultaneously accelerate the encoding speed of texture and depth maps. The core of this algorithm is to disguise the self-learning residual model as a binary classifier to determine in advance whether the coding units in texture and depth maps need to be partitioned. Specifically, the algorithm treats residual signals as features of each coding unit, extracts residual signals from each CU, and ultimately constructs the self-learning residual model using intra feature learning. In [16], Chen et al. found that the division depth of coding units in flat regions in depth maps is often not very deep, and DMM modes are often not selected. Therefore, if it is known in advance that CU will not continue to partition and DMM modes will not be selected, unnecessary RDO calculations and DMM calculations can be skipped, thus reducing the encoding complexity. The authors use sum of gradient to calculate the flatness of CU. Due to the characteristics of depth maps, most coding units have high flatness. Therefore, this algorithm improves encoding efficiency while ensuring encoding quality.

B. APPROACHES FOR VVC 2D VIDEO

The emergence of the new partitioning structure, QTMT, has led researchers to invest in the research of new fast block partitioning algorithms for coding units. Currently, there are many efficient research literatures to improve the intra frame encoding efficiency of VVC 2D texture maps. Two solutions were presented in reference [17]. Specifically, the first algorithm: select three important features and use them to trains the five Support Vector Machine (SVM) models for CUs of sizes from 8 × 16 to 32 × 32 respectively, which can adaptively skip the unnecessary division direction of the CU in advance. The second algorithm: use the Laplace operator to determine the general direction in order to skip some intra modes in advance. Zhao et al. [18] presented a fast-partitioning algorithm based on the edge features of coding units. First, coding unit was divided into CU of complex region and CU of simple region according to whether it contained edges. For the former, the authors calculated their edge feature value and compared it with the two preset thresholds, so as to skip the impossible partitioning direction in advance. For the latter, whether to end the division process in advance is determined according to the difference between their division depth and the division depth of adjacent CUs. Wang et al. [19] used machine learning approaches to speed up the decision-making process of coding unit size. Specifically, the authors constructed two extra tree models, which were executed in series. The algorithm process involved calculating some relevant characteristic values of the CU and inputting this information into the first extra tree model to determine the current direction of CU partitioning. The second extra tree model was able to predict whether the CU would be partitioned into a binary tree or a ternary tree, thus simplifying the encoding process and greatly accelerating the encoding process. Unlike previous fast algorithms for intra and inter frame coding, in general, intra frame coding is aimed at reducing spatial redundancy, while inter frame coding is aimed at reducing temporal redundancy. This difference often leads to different methods used for both. Kuang et al. [20] used the same approach to diminish the complexity of intra and inter frame prediction coding, utilizing several historical information, namely traversed partition modes. This method can skip RDO calculation process of some partition modes, thereby improving coding efficiency. In response to the time-consuming problem of using QTMT partition structure to partition coding units in VVC encoders, Amna et al. [21] presented a swift CU size decision strategy. The authors constructed a lightweight neural network (LNN) model to determine whether the current CU needs ternary tree partition or binary tree partition. This method requires calculating several important feature values of the current CU

and inputting them into the LNN model, which can save the encoder a lot of computation process of traversing the ternary tree partition mode and effectively reduce encoding time.

The methods mentioned above that use traditional manual methods, machine learning, and deep learning all work well in reducing coding complexity, but all of these algorithms are applied to 3D-HEVC and VVC 2D video. At present, there are few papers on VVC 3D video in the field of scientific research, and we believe that with the continuous popularity of VVC video coding and people's demand for higher quality 3D video, the research on VVC 3D video will eventually become hot in the near future.

III. PROPOSED ALGORITHM

In recent years, emerging deep learning technologies such as CNN have played a significant role in many fields, many scholars have used CNNs to accelerate CU division and mode selection in the field of video coding research and have achieved amazing results [22]. In this paper, we present a CU partitioning method based on gradient matrix and adaptive CNN. Firstly, we introduce a coding unit classifier based on texture complexity to divide CUs into three categories, namely simple, fuzzy, complex. Specifically, if the texture complexity (TC) value calculated by the CU classifier of the coding unit is less than T1, it is classified as simple class, and the partition process is terminated; if the TC is larger than the T1 but less than T2, the CU is considered as fuzzy CU that will perform primitive RDO calculation to decide on the division mode; for the third case, CUs are classified as complex CUs, such CUs will be fed into subsequent CNNs for further division determination. Specifically, the first CNN model is used to determine whether a square CU performs quadtree partitioning or multi type tree partitioning. The second CNN model is used to determine whether a CU that definitely performs multi type tree partitioning performs horizontal or vertical tree partitioning. Figure 2 shows the overall algorithm flowchart. The following sections describe the complexity criterion used to classify CUs, the adaptive CNN structure, and their training process.

A. CU CLASSIFICATION MODEL BASED ON GRADIENT MATRIX

Depth maps represent the distance of objects in a scene from the camera's viewpoint, and are characterized by a large area of flat and a small number of distinct shape edge areas. Considering the depth map feature, this paper proposes to use gradient matrix to calculate the texture complexity of CUs and classify CUs: homogeneous CU, fuzzy CU and complex CU. Among them, homogeneous CU means that there is no need to continue to divide into smaller CUs for encoding, and the current size is the optimal coding size; complex CU refers to the need to continue to divide into smaller size CUs for encoding.

In depth map encoding, the pixel values of flat areas change very little or nothing, and on the contrary, the pixel values of the edge areas of the shape change dramatically [23].

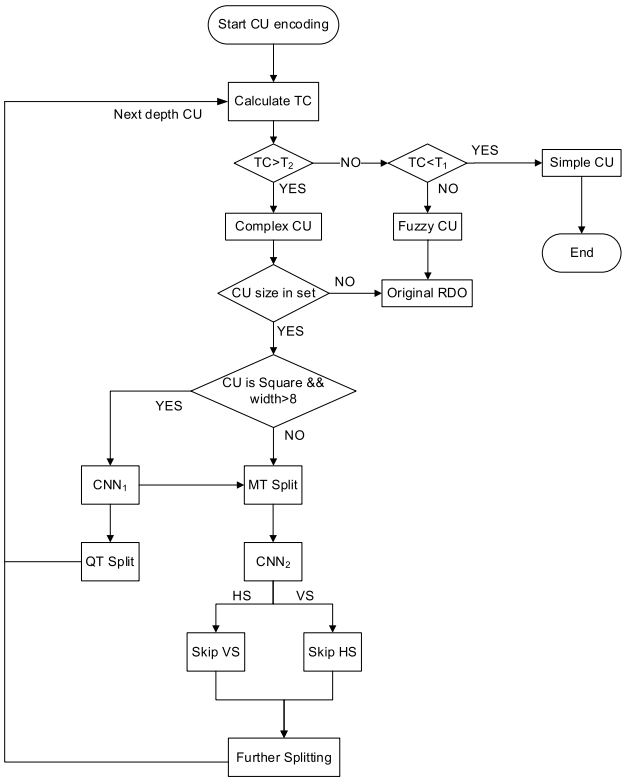


FIGURE 2. Overall algorithm flowchart.

To evaluate the texture complicacy of the depth map coding block, in this paper, we use gradient matrix as a tool to calculate the texture complicacy of coding units. The specific details of the gradient matrix are as follows: for a CU with height H and width W, it contains H × W grayscale values, among which there are (H-2) × (W-2) non edge grayscale values. These non-edge grayscale values can form a matrix of 3 × 3 size with their surrounding grayscale values. In this article, we refer to this matrix as the 3 × 3 matrix of non-edge pixels, and its shape is shown in Figure 3. Each such matrix will form an element of the final gradient matrix, and each element actually represents the texture complexity of the local area where its matrix is located. It calculates the sum of the gradients of the pixel in four directions, which are 0°, 90°, 45°, and 135°, respectively. Their calculation formulas are as follows:

$$(i,j)_{0^\circ} = |p\,(i-1,j) - p\,(i+1,j)| \tag{1}$$

$$(i,j)_{90^\circ} = |p\,(i,j-1) - p\,(i,j+1)| \tag{2}$$

$$(i,j)_{45^\circ} = |p\,(i+1,j-1) - p\,(i-1,j+1)| \tag{3}$$

$$(i,j)_{135^\circ} = |p\,(i-1,j-1) - p\,(i+1,j+1)| \tag{4}$$

We set the coordinates of each element of the gradient matrix as (a, b) and use $p\,(a,b)$ to represent the values of elements in the gradient matrix, whose values are the sum of gradients in four directions. Its calculation formula is as follows:

$$p\,(a,b) = (a,b)_{0^\circ} + (a,b)_{90^\circ} + (a,b)_{45^\circ} + (a,b)_{135^\circ} \tag{5}$$

P(i-1, j-1)	P(i, j-1)	P(i+1, j-1)
P(i-1, j)	P(i, j)	P(i+1, j)
P(i-1, j+1)	P(i, j+1)	P(i+1, j+1)

FIGURE 3. Matrix with size of 3 × 3 composed of non-edge pixels.

Each depth map coding unit has a unique corresponding gradient matrix. If the length and width of a CU are H and W, respectively, then the size of its gradient matrix is (H-2) × (W-2). We can easily infer that the pixel values in the pixel matrix of the coding units in the flat area of the depth map are basically the same, which results in most of the elements in the gradient matrix being almost zero. On the contrary, for the coding units in the sharp area of the depth map, the element values in the pixel matrix differ greatly, leading to a larger element value in the gradient matrix. Figure 4 shows the pixel matrix and its gradient matrix of a depth map coding unit with simple and complex textures, confirming this conclusion.

In the study of CU partitioning in video encoding, texture complexity often plays a decisive role in the depth of coding unit partitioning [24]. CUs in regions with strong texture homogeneity are likely to end the partitioning process earlier, while CUs in regions with more complex textures are likely to continue their partitioning process. In 3D video encoding, this rule still holds true for both texture and depth videos [25]. Figure 5 shows the coding unit partitioning result of a depth map in the Newspaper video sequence, showing the relation between CU texture features and partitioning. And the gradient matrix we introduce can better reflect the texture complicacy of the depth map coding unit, we give the texture complexity (TC) calculation method according to the gradient matrix of CU. By comparing the magnitude relationship of TC with the preset threshold, we divide CU into simple CU, fuzzy CU, and complex CU. The calculation formula for TC is as follows:

$$TC = \sum_{a=1}^{W-2} \sum_{b=1}^{H-2} p\,(a, b) \times \frac{1}{(W-2) \times (H-2)} \qquad (6)$$

75	75	75	75	75	75	75	75
75	75	75	75	75	75	75	75
75	75	75	75	75	75	75	75
75	75	75	75	75	75	75	75
75	75	75	75	75	75	75	75
75	75	75	75	75	75	75	75
75	75	75	75	75	75	75	75
75	75	75	75	75	75	75	75

→

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

(a)

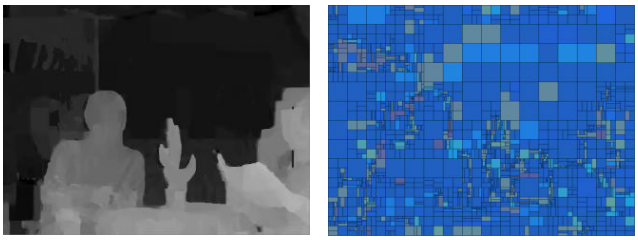
17	17	19	20	26	35	43	55
17	17	19	20	25	35	42	54
17	17	19	19	26	35	41	52
17	17	19	19	26	36	43	50
21	21	19	17	29	37	44	52
30	38	35	33	40	45	47	52
32	38	38	32	41	45	49	53
42	56	54	52	52	52	52	58

→

6	8	21	46	49	58
6	7	21	49	50	51
10	4	26	56	49	39
52	55	54	63	48	40
52	56	61	57	38	35
55	62	53	51	32	26

(b)

FIGURE 4. Pixel matrix and gradient matrix of depth map CU (a) Pixel matrix and gradient matrix of simple CU (b) Pixel matrix and gradient matrix of complex CU.



(a) (b)

FIGURE 5. A frame image in the newspaper video sequence and its CU partitioning results.

Obviously, the setting of the two thresholds T_1 and T_2 is particularly crucial in this algorithm. Similar to the threshold setting in manual algorithms aimed at reducing video encoding complexity, our threshold setting needs to minimize encoding time as much as possible while ensuring minimal loss of video quality. To maintain the effectiveness of the threshold, we use the method of training frames to dynamically adjust the threshold, so that it can better serve the algorithm and classify CUs into three categories. Specifically, we define the first frame of the video sequence as the training frame, and the encoding of the first frame image adopts the anchor encoder algorithm. After encoding the first frame, we obtain two ideal thresholds T_1 and T_2. Considering the temporal correlation of video encoding, the threshold is updated every 30 frames to ensure that the algorithm remains effective. Through this method, we get two thresholds, T_1 and T_2. If the texture complexity (TC) value calculated by the CU

classifier for the coding unit is less than T_1, it is classified as simple class, and the partition process is terminated. If the value of TC is between T_1 and T_2, CUs will execute the original encoder encoding to ensure that the algorithm hardly affects the encoding quality. If TC is greater than T_2, the proposed adaptive CNNs are utilized to speed up the QTMT segmentation process. This algorithm can avoid unnecessary pattern calculations, thus achieving the goal of this paper, and can remove some training samples from our proposed adaptive CNNs, thereby improving its accuracy.

B. ADAPTIVE CNN STRUCTURE

Because new partition module named Multi-Type Tree (MTT) is added in VVC, which greatly increases coding efficiency at the expense of enormous complicacy. At the same time, changes in the rules for dividing code units will also bring about changes in the dividing results. In VVC, there are 16 CU sizes [26] (Contains squares and rectangles) that can continue to be divided compared to the 3 CU sizes (Square only, 64 × 64, 32 × 32, 16 × 16) of HEVC. Therefore, some algorithmic ideas for designing three models separately for the three CU sizes of HEVC will not work well in VVC 3D Video depth map intra coding. Authors in [27] have presented a CNN network model with variable pooling layer size to ensure that the final output feature map size is consistent. Based on the structural characteristics of this CNN network model, we design two CNNs to accelerate the QTMT division of depth maps intra-frame coding in VVC 3D video. Consequently, the proposed method can replace part of RDO, which is significant for reducing depth map intra coding complexity.

Specifically, the biggest highlight of the adaptive convolutional neural network proposed in this paper is that it can serve a variety of coding units with different sizes. Its specific performance is that CUs with different sizes are input into the adaptive CNN, and before they reach the full connection layer, they will become input blocks with a size of 4 × 4. Therefore, we can only use this model to predict the division mode of these CUs, which benefits from the clever design of the pooling layer. The following introduces the input layer, convolutional layer, pooling layer, fully connected layer, and output layer of the proposed adaptive CNN one by one. Our model includes an input layer, where the input is the residual block of the coding unit rather than the original block, as the former can better reflect its partitioning information. Convolutional layer 1 has 64 convolution kernels, with a size of 5 × 5 and a step size of 1. The same filling mode is used to keep the size of the feature map unchanged. Adaptive pooling layer 1 is the max pooling layer, and its size varies adaptively based on the size of the input block. The function of this pooling layer is to reduce the input block by half when its width or length is greater than or equal to 32. The only difference from convolution layer 1 is that the kernel size of convolution layers 2 and 3 is 3 × 3, and they still use the same filling method. The structure and function of adaptive pooling layer 2 are basically the same as the first pooling layer. The function of this pooling layer is to reduce the input block by half when its width or length is greater than or equal to 16. Pooling layer 3 is the max pooling layer with a size of 2 × 2. The fully connected layer has 64 neurons, and to raise the accuracy of the model, it includes encoding quantization parameters and the height and width of the input CU. The output layer of the entire model uses the SoftMax activation function to output the predicted probabilities of two classification results. Figure 6 and Table 1 respectively show the brief process and detailed structure of the model.

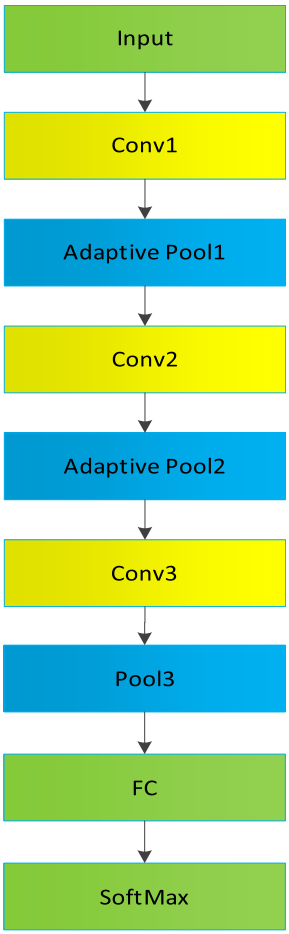


FIGURE 6. The presented adaptive CNN architecture.

C. CNNs TRAINING

This article uses official 3D video sequences to train and test the presented model. And the training data of both CNN models are all from the data classified as complex CU by the first algorithm proposed in this article, to raise the prediction accuracy of the CNN model, and to make the two sub algorithms complement each other, which can promote our overall algorithm to realize the goal of significantly reducing encoding complicacy without affecting video quality. The official 3D video sequence is shown in Table 2. It should be pointed out that two CNNs are complementary to each other.

TABLE 1. Details of the presented adaptive CNN structure.

Type	Architecture
Input	Residual Block: Width × Height × 1
Conv1	Filter: 5 × 5 × 64, Stride:1, Pad:2, ReLU
Adaptive Pool1	Max, Size: ((Width ≥ 32) + 1) × ((Height ≥ 32) + 1)
Conv2	Filter: 3 × 3 × 64, Stride:1, Pad:1, ReLU
Adaptive Pool2	Max, Size: ((Width ≥ 16) + 1) × ((Height ≥ 16) + 1)
Conv3	Filter: 3 × 3 × 64, Stride:1, Pad:1, ReLU
Pool3	Max, Size: 2 × 2
FC	64, include: QP, Width, Height; ReLU
SoftMax	2

To forecast the division mode of a square CU, our proposed adaptive CNN sub algorithm first uses the first CNN to determine whether it is QT partitioned. If its optimal partition mode is QT partitioned, predicting whether it is horizontally or vertically partitioned is meaningless and wastes encoding time. Similarly, predicting whether to perform QT partitioning on a rectangular CU is meaningless and a waste of coding time. In summary, if the current CU is square, it must be input into the first CNN model first. If the prediction result is MT partition, the residual block of this CU must be input into the second CNN model. Otherwise, the second model will not be utilized. If the current CU is a rectangle, it is directly input into the second CNN model. Therefore, in order to achieve high prediction accuracy, appropriate steps must also be followed when training the dataset. Square CUs must be filtered by the first CNN model before being used as training data for the second CNN model. The two proposed CNN models have the same network structure and use the same optimization algorithm and loss function during their training process. The optimization algorithm uses the Stochastic Gradient Descent (SGD) algorithm, and the loss function uses the commonly used cross-entropy loss function for binary classification prediction. The formula is as follows:

$$loss = \sum_{i=1}^{n} [\mathrm{m}_i \log(\hat{m}) + (1 - m_i) \log(1 - \hat{m})] \quad (7)$$

where m_i is the true value of the sample, $\hat{m}$ is the predicted value output by the model.

During the training process, for the first CNN model, if the division mode of the CU is QT division, it is marked "1,0"; otherwise, it is marked "0,1". Similarly, for the second CNN model, if the division mode of the CU is horizontal division, it is marked "1,0"; otherwise, it is marked "0,1". After the model training is completed, we can apply the adaptive CNN solution to seven block sizes, thus avoiding unnecessary partition pattern traversal for many coding units.

IV. EXPERIMENTAL RESULTS

The experimental results of the algorithm presented in this article are presented in this section. The depth map video sequence is from the official 3D video sequence as shown in Table 2. It contains eight video sequences, two resolutions, and has multiple video features. It should be pointed out that our proposed algorithm only applies to the division

TABLE 2. Official 3D video sequence.

Video Sequences	Resolution	3-Views Input	Frame Rate	Frames to Be Encoded
Balloons	1024 × 768	1-3-5	30	300
Kendo		1-3-5	30	300
Newspaper		2-4-6	30	300
GT_Fly	1920×1088	9-5-1	25	250
Poznan_Hall2		7-6-5	25	200
Poznan_street		5-4-3	25	250
Undo_dancer		1-5-9	25	250
Shark		1-5-9	30	300

of 3D video depth map coding units and does not include texture maps. The software and hardware environment for the experiment is as follows: the CPU is Intel (R) Core (TM) i7-10700 K, the RAM is 32 GB, and the GPU is RTX 3070. The original encoder is VTM 10.0, with an encoding configuration of All intra and QP (depth) set to 34, 39, 42, and 45. We use two performance metrics to demonstrate the effectiveness of the method, namely Bjøntegaard delta bit rate (BDBR) [28] and time reduction (TR). BDBR represents the rate savings under a certain objective encoding quality, a negative value indicates an improvement in encoding performance, and a positive value indicates a loss in encoding quality. The change in coding complicacy is represented by the proportion of encoding time saved, and its expression is:

$$TR = \frac{T_{anc} - T_{pro}}{T_{anc}} \times 100\% \quad (8)$$

where T_{pro} represents the time required to encode depth maps using the proposed approach, and T_{anc} represents the time required to encode depth maps using anchoring algorithms.

A. ALGORITHM PERFORMANCE ANALYSIS

The proposed approach includes two sub algorithms, namely the CUs classification algorithm based on gradient matrix and the adaptive CNN fast coding unit partitioning algorithm. Both contribute to reducing the complicacy of intra frame depth map CU division in VVC 3D videos. The first sub algorithm mainly utilizes the texture characteristics of 3D video depth maps, namely the presence of large flat areas and some sharp areas, focusing on canceling the rate distortion cost calculation process of coding units in flat areas, thereby significantly reducing encoding time. The second sub algorithm addresses the phenomenon of diverse sizes of CUs in the QTMT partitioning mode, and constructs two adaptive CNNs to solve the problem of input CU sizes being different. This efficiently predicts the partitioning mode of CUs, eliminates the rate distortion cost calculation process of unnecessary partitioning modes of CUs, and accelerates the encoding speed.

To test the specific performance of the two sub algorithms, we examine their performance when acting alone. Specifically, we first use the gradient matrix to classify each coding unit. For simple CUs, the RDO process can be skipped, while

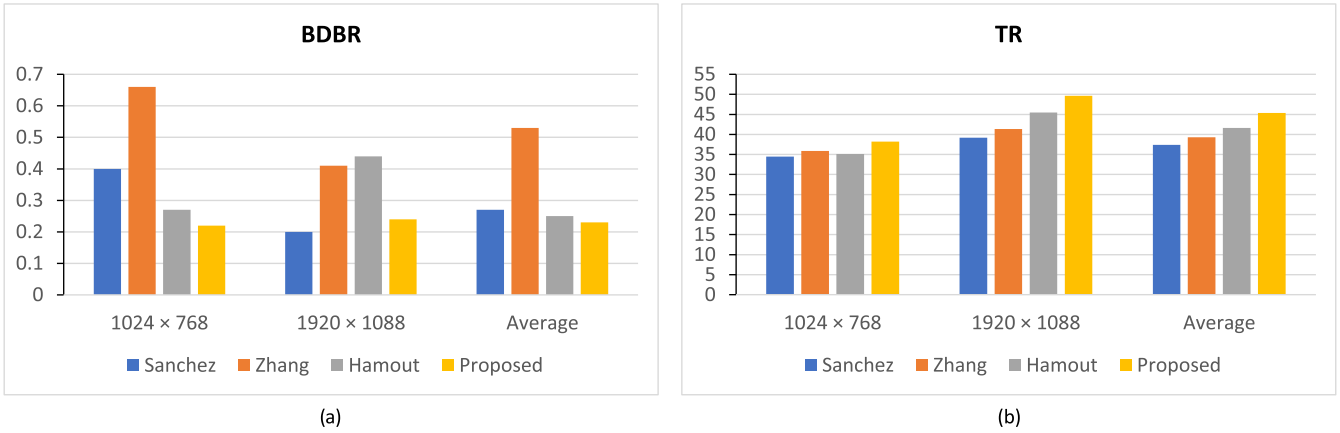


FIGURE 7. Performance comparison of four methods at two resolutions (a) Encoding loss (b) Time reduction.

for other types of CUs, the original encoder algorithm needs to be used for partitioning decisions. This way, we obtain the performance of the first sub algorithm. We changed the training data of the adaptive CNN to all CUs and used this model to predict whether the square CU will undergo QT partitioning and whether the rectangular CU will skip a certain direction of partitioning mode. This way, we obtained the performance of the second sub algorithm. Obviously, using the second sub algorithm alone will reduce its expected effect. The overall algorithm and the performance of the two sub algorithms are shown in Table 3.

We can conclude from Table 3 that both of our proposed sub algorithms can diminish the complicacy of depth map encoding and maintain almost no loss of encoding quality. Specifically, the first CUs classification algorithm based on gradient matrix can diminish encoding time by 33.71% on average, while the encoding quality loss of this sub algorithm is only 0.18%. This indicates that it can effectively identify simple CUs and terminate their partitioning decision process in advance, thus avoiding unnecessary RDO calculation processes for simple CUs. This makes the original encoder only need to make partitioning mode decisions for other types of CUs. The second fast CU partitioning method based on adaptive CNN can diminish encoding time by 28.88% on average, while the encoding quality loss of this sub algorithm is only 0.25%. This indicates that the presented CNN model can be effectively applied to various sizes of CUs and successfully predict their possible partitioning directions, reducing the burden of encoder computation rate distortion cost. Specifically, when using the first sub algorithm alone, it showed the best performance in the Poznan_street video sequences, with a reduction of 48.18% in encoding time compared to the original encoder. This is because the video sequence has a large flat area, which contains a large number of simple CUs, thus leveraging the advantages of this algorithm. The second sub algorithm showed the best performance in Shark video sequences with relatively few flat regions, reducing encoding complexity by 34.71%. In summary, the two sub algorithms efficiently diminish the encoding time of VVC 3D video depth maps while ensuring that the encoding quality is not significantly affected.

TABLE 3. Individual performance and overall performance of two sub algorithms.

Sequence	Gradient Matrix		Adaptive CNNs		Overall	
	BDBR (%)	TR (%)	BDBR (%)	TR (%)	BDBR (%)	TR (%)
Balloons	0.13	21.25	0.24	25.19	0.16	36.32
Kendo	0.20	23.43	0.25	24.35	0.24	37.49
Newspaper	0.31	26.38	0.31	24.42	0.26	40.74
GT_Fly	0.19	36.49	0.36	33.57	0.27	45.67
Poznan_Hall2	0.14	43.16	0.22	29.13	0.21	55.83
Poznan_street	0.15	48.18	0.26	28.14	0.22	50.29
Undo_dancer	0.16	34.46	0.19	31.56	0.21	53.17
Shark	0.13	36.29	0.20	34.71	0.30	43.28
1024 × 768	0.21	23.69	0.27	24.65	0.22	38.18
1920 × 1088	0.15	39.72	0.25	31.42	0.24	49.65
Average	0.18	33.71	0.25	28.88	0.23	45.35

The last two columns of Table 3 show the performance of the overall method. As our first and second sub algorithms can complement each other, the former filters out sample data that fits its function and can train models with higher prediction accuracy, while the latter acts on complex CUs that the first sub algorithm cannot handle. The two complement each other, so the overall algorithm's performance is much higher than that of a single sub algorithm. The eight test video sequences used in this article have two resolutions, 1024 × 768 and 1920 × 1088, respectively. The method proposed in the five video sequences with a resolution of 1920 × 1088 can perform better, reducing coding time by 49.65% on average, and encoding loss is only negligible by 0.24%. Overall, the average experimental results of 8 video sequences indicate that the overall method can diminish coding time by 45.35%, while the BDBR increase is only 0.23%, achieving the goal of reducing the complicacy of VVC 3D video depth map encoding.

TABLE 4. The experimental results of the proposed approach compared to the other three approaches.

Sequence	Sanchez [29]		Zhang [30]		Hamout [31]		Proposed	
	BDBR (%)	TR (%)	BDBR (%)	TR (%)	BDBR (%)	TR (%)	BDBR (%)	TR (%)
Balloons	0.36	34.1	0.67	35.49	0.37	32.31	0.16	36.32
Kendo	0.37	33.9	0.43	38.23	0.50	34.29	0.24	37.49
Newspaper	0.46	35.4	0.89	33.85	0.17	38.71	0.26	40.74
GT_Fly	0.12	40.6	0.21	43.14	0.08	49.24	0.27	45.67
Poznan_Hall2	0.43	38.8	0.75	48.33	0.13	46.03	0.21	55.83
Poznan_street	0.22	41.7	0.29	39.67	0.50	50.73	0.22	50.29
Undo_dancer	0.12	38.5	0.49	41.57	0.12	50.67	0.21	53.17
Shark	0.11	36.2	0.33	34.16	0.09	30.75	0.30	43.28
Average	0.27	37.4	0.53	39.31	0.25	41.59	0.23	45.35

B. PERFORMANCE COMPARISON

To indicate the advantages of the proposed approach, we will compare its performance with three advanced methods. Table 4 presents in detail the experimental results of three methods and the method presented in this paper on eight video sequences. Reference [29] provides a scheme that includes three fast pattern decision algorithms for depth maps, achieving 37.4% coding time savings at a cost of 0.27% coding quality loss. Compared with this algorithm, our presented method can diminish coding time by 7.95% on average. To alleviate the huge computational complexity problem induced by the introduction of new encoding technology in 3D-HEVC, [30] provides two methods to quicken the intra frame mode decision-making process of depth maps, achieving a reduction of 39.31% in coding time, but an increase of 0.53% in BDBR. Compared with this, our algorithm achieves more encoding time reduction at the cost of smaller encoding quality loss, demonstrating the superiority of our method. Reference [31] utilized the clustering method to accelerate the encoding speed of 3D video depth maps, achieving good results in ensuring encoding quality, just like the method proposed in this paper. The BDBR increase was only 0.25% (the proposed algorithm was 0.23%), while the method proposed in this paper provided an additional 3.76% reduction in encoding complexity compared to the method proposed in [31]. Moreover, in the Shark video sequences, the proposed method realizes a 14.53% reduction in encoding time compared to the literature method. Figure 7 shows the performance of four schemes under two video resolutions and average conditions, intuitively demonstrating that compared to other excellent methods, this algorithm performs well in reducing depth map encoding complicacy and ensuring video quality.

V. CONCLUSION

This paper provides a method to reduce the complexity of intra frame coding unit partitioning in VVC 3D video depth maps, which consists of two parts. The first part is a coding unit classification algorithm based on gradient matrix, which can classify CUs into three categories: simple CUs, fuzzy CUs, and complex CUs. It can skip the partitioning process of simple CUs in advance to reduce unnecessary encoding calculations and filter sample data for adaptive CNN models. The second part constructs two CNN models with consistent structures, which can predict the partition patterns of multiple sizes of CUs under VVC partitioning technology. These two adaptive CNN models complement each other and avoid unnecessary partition pattern judgments. The experimental results indicate that the proposed solution can diminish encoding time by 45.35%, while BDBR only increases by 0.23%, showing excellent performance compared to advanced algorithms. Therefore, our proposed method has contributed to 3D video encoding and accelerated its application speed in real life.

REFERENCES

[1] S. Saponara, K. Denolf, G. Lafruit, C. Blanch, and J. Bormans, "Performance and complexity co-evaluation of the advanced video coding standard for cost-effective multimedia communications," *EURASIP J. Adv. Signal Process.*, vol. 2004, no. 2, pp. 1–16, Mar. 2004.

[2] Y.-S. Ho and K.-J. Oh, "Overview of multi-view video coding," in *Proc. 4th Int. Workshop Syst., Signals Image Process., 6th EURASIP Conf. Speech Image Process., Multimedia Commun. Services*, 2007, pp. 5–12.

[3] Y. Chen, M. M. Hannuksela, T. Suzuki, and S. Hattori, "Overview of the MVC+D 3D video coding standard," *J. Vis. Commun. Image Represent.*, vol. 25, no. 4, pp. 679–688, May 2014.

[4] C. Park, "Efficient intra-mode decision algorithm skipping unnecessary depth-modelling modes in 3D-HEVC," *Electron. Lett.*, vol. 51, no. 10, pp. 756–758, May 2015.

[5] K. Zhang, J. An, H. Huang, J.-L. Lin, Y.-W. Huang, and S.-M. Lei, "Segmental prediction for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 11, pp. 2425–2436, Nov. 2017.

[6] Y. Chen, X. Zhao, L. Zhang, and J.-W. Kang, "Multiview and 3D video compression using neighboring block based disparity vectors," *IEEE Trans. Multimedia*, vol. 18, no. 4, pp. 576–589, Apr. 2016.

[7] W. Sun, O. C. Au, L. Xu, Y. Li, and W. Hu, "Seamless view synthesis through texture optimization," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 342–355, Jan. 2014.

[8] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021.

[9] Y.-L. Chan, C.-H. Fu, H. Chen, and S.-H. Tsang, "Overview of current development in depth map coding of 3D video and its future," *IET Signal Process.*, vol. 14, no. 1, pp. 1–14, 2020.

[10] H. Kim and Y.-H. Kim, "Advanced multi-layer depth video coding method in VVC," in *Proc. IEEE Int. Conf. Consum. Electron.-Asia (ICCE-Asia)*, Oct. 2022, pp. 1–3.
[11] C. Liu, K. Jia, and P. Liu, "Fast depth intra coding based on depth edge classification network in 3D-HEVC," *IEEE Trans. Broadcast.*, vol. 68, no. 1, pp. 97–109, Mar. 2022.
[12] Y. Li, G. Yang, A. Qu, and Y. Zhu, "Tunable early CU size decision for depth map intra coding in 3D-HEVC using unsupervised learning," *Digit. Signal Process.*, vol. 123, Apr. 2022, Art. no. 103448.
[13] H. Hamout and A. Elyousfi, "A computation complexity reduction of the size decision algorithm in 3D-HEVC depth map intracoding," *Adv. Multimedia*, vol. 2022, pp. 1–12, Jun. 2022.
[14] N. Omran, A. Maraoui, I. Werda, and B. Hamdi, "Fast partition algorithm in depth map intra coding unit based on multi-deep convolution neural network," *J. Real-Time Image Process.*, vol. 21, no. 1, pp. 1–10, Feb. 2024.
[15] Y. Li, N. Zhu, G. Yang, Y. Zhu, and X. Ding, "Self-learning residual model for fast intra CU size decision in 3D-HEVC," *Signal Process., Image Commun.*, vol. 80, Feb. 2020, Art. no. 115660.
[16] J. Chen, B. Wang, H. Zeng, C. Cai, and K.-K. Ma, "Sum-of-gradient based fast intra coding in 3D-HEVC for depth map sequence (SOG-FDIC)," *J. Vis. Commun. Image Represent.*, vol. 48, pp. 329–339, Oct. 2017.
[17] G. Ding, X. Lin, J. Wang, and D. Ding, "Accelerating QTMT-based CU partition and intra mode decision for versatile video coding," *J. Vis. Commun. Image Represent.*, vol. 94, Jun. 2023, Art. no. 103832.
[18] S. Zhao, X. Shang, G. Wang, and H. Zhao, "A fast algorithm for intra-frame versatile video coding based on edge features," *Sensors*, vol. 23, no. 13, p. 6244, Jul. 2023.
[19] K. Wang, H. Liang, S. Zhang, and F. Yang, "Fast CU partition method based on extra trees for VVC intra coding," in *Proc. IEEE Int. Conf. Vis. Commun. Image Process. (VCIP)*, Dec. 2022, pp. 1–5.
[20] W. Kuang, X. Li, X. Zhao, and S. Liu, "Unified fast partitioning algorithm for intra and inter predictions in versatile video coding," in *Proc. Picture Coding Symp. (PCS)*, Dec. 2022, pp. 271–275.
[21] M. Amna, W. Imen, and S. Fatma Ezahra, "Fast multi-type tree partitioning for versatile video coding using machine learning," *Signal, Image Video Process.*, vol. 17, no. 1, pp. 67–74, Feb. 2023.
[22] D. Liu, Y. Li, J. Lin, H. Li, and F. Wu, "Deep learning-based video coding: A review and a case study," *ACM Comput. Surv.*, vol. 53, no. 1, pp. 1–35, 2020.
[23] T. Senoh, Y. Ichihashi, H. Sasaki, and K. Yamamoto, "Simple multi-view coding with depth map," *3D Res.*, vol. 4, no. 2, pp. 1–12, Jun. 2013.
[24] P. Ndjiki-Nya, B. Makai, A. Smolic, H. Schwarz, and T. Wiegand, "Video coding using texture analysis and synthesis," in *Proc. Picture Coding Symp.*, Saint-Malo, France, 2003.
[25] Q. Zhang, N. Zhang, T. Wei, K. Huang, X. Qian, and Y. Gan, "Fast depth map mode decision based on depth–texture correlation and edge classification for 3D-HEVC," *J. Vis. Commun. Image Represent.*, vol. 45, pp. 170–180, May 2017.
[26] A. Tissier, W. Hamidouche, S. B. D. Mdalsi, J. Vanne, F. Galpin, and D. Menard, "Machine learning based efficient QT-MTT partitioning scheme for VVC intra encoders," *IEEE Trans. Circuits Syst. Video Technol.*, 2023.
[27] G. Tang, M. Jing, X. Zeng, and Y. Fan, "Adaptive CU split decision with pooling-variable CNN for VVC intra encoding," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2019, pp. 1–4.
[28] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-curves*, document ITU SG16 VCEG-M33, 2001.
[29] G. Sanchez, L. Agostini, and C. Marcon, "A reduced computational effort mode-level scheme for 3D-HEVC depth maps intra-frame prediction," *J. Vis. Commun. Image Represent.*, vol. 54, pp. 193–203, Jul. 2018.
[30] H.-B. Zhang, C.-H. Fu, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, "Probability-based depth intra-mode skipping strategy and novel VSO metric for DMM decision in 3D-HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 2, pp. 513–527, Feb. 2018.
[31] H. Hamout and A. Elyousfi, "Fast depth map intra coding for 3D video compression-based tensor feature extraction and data analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 1933–1945, Jul. 2020.

LINA SI received the master's degree in computer technology from the Huazhong University of Science and Technology, Wuhan, China, in 2010. She is currently a Professor with Zhengzhou University of Light Industry, China. Her current research interests include video processing, Bayesian estimation, and intelligent computation.

AOHUI YAN received the B.S. degree in computer science and technology from Zhengzhou University of Light Industry, Zhengzhou, China, in 2022, where he is currently pursuing the master's degree in software engineering with the College of Computer Science and Technology. His current research interests include image processing, deep learning, 3D video coding, and extensions of the versatile video coding.

QIUWEN ZHANG (Member, IEEE) received the Ph.D. degree in communication and information systems from Shanghai University, Shanghai, China, in 2012. Since 2012, he has been a Faculty Member with the College of Computer and Communication Engineering, Zhengzhou University of Light Industry, where he is currently an Associate Professor. He has published over 30 technical articles in the fields of pattern recognition and image processing. His major research interests include 3D signal processing, machine learning, pattern recognition, video codec optimization, and multimedia communication.