RSP-BT:An Advanced Parallel Method for Depth Map Motion Estimation

Xie Xiaoyan, Wang Anqi, Zhu Yun, Hu Chuanzhan, Du Zhuolin

*Xi'an University of Posts and Telecommunications, Xi'an, China

E-mail: 982725333@qq.com Tel: +86-29-19829497757

***Abstract*— Aiming at the high computational complexity and long encoding time caused by insufficient consideration of the homogeneous area in depth map motion estimation of 3D High Efficiency Video Coding(3D-HEVC), An optimization method to reduce search points based on block type (RSP-BT) is proposed. Firstly, according to the pixel distribution features of coding blocks, the coding blocks are classified into homogeneous or edge areas. And then the different search strategies are selected adaptively in the light of center-biased characteristic of the motion vectors for different types of area. Thereby, the overall calculation overhead is reduced. Experimental results show that, compared with the full-search algorithm, the proposed algorithm has an average Peak Signal to Noise Ratio (PSNR) loss of only 0.0519dB; Compared with the TZSearch (Test Zone Search) algorithm, the search points is reduced by 11.25% on average. In order to further improve the coding speed, a parallel implementation is given on the video array processor of the optimized algorithm, in the view of there is no data correlation among the matching operates for different coding blocks.The experimental results show that the average speedup of the designed parallel scheme is 2.8940.**

***Keywords*: depth map; search points; block type; parallel;video array processor**

I. Introduction

The Joint Collaborative Team on 3D Video Coding working group proposed an advanced 3D-HEVC standard [1] based on High Efficiency Video Coding (HEVC), which adopted Multi-View plus Depth (MVD),encoding format. MVD contains a sequence of texture maps and its corresponding sequence of depth maps. The depth map is used to represent the distance between different objects in the scene and the camera, and to draw more virtual viewpoints at the decoder. Both texture map sequence and depth map sequence are used to describe the same scene, but there is a huge difference between them. Texture map have rich details and a few homogeneous texture areas, but the homogeneous areas of depth map account for up to 85% [2] and the steep edges only exist in the edges of objects. Therefore, texture map coding needs to focus on the detailed information of each pixel, while depth map only needs to protect the edge information of each object in the image [3], which is used to ensure the quality of the synthetic viewpoint.

In order to make full use of the unique content characteristics of the depth map and reduce the coding complexity of the depth map inter prediction, experts and scholars have proposed some effective solutions. To reduce coding complexity, reference [2][4] divides the depth map into edge areas and homogeneous areas based on statistical methods to limit the coding unit (CU) division depth and prediction unit (PU) division modes of the depth map according to the classification results. In order to solve the problem of unnecessary computation caused by using the high complexity Full Search (FS) algorithm or TZSearch algorithm for block matching in the depth map motion estimation , the [5] propose to use a lightweight Diamond Search (DS), algorithm for block matching, but the problem that the edge area of the depth map is more difficult to predict is ignored. The [6] proposes to classify the depth map based on reference [5], using the TZSearch algorithm in the edge area and the small diamond search algorithm in the homogeneous area, which reduce the calculation complexity effectively while ensuring the coding quality. The above algorithms only use the characteristics of the depth map to reduce the encoding time of the depth map in the software algorithm level, these methods cannot meet the needs of real-time video encoding. Therefore, multimedia engineers try to transplant video algorithms to FPGA, ASIC, and GPU [7]. So in [8][9], a dedicated hardware architecture based on FS algorithm and DS algorithm respectively is designed.

In summary, algorithm optimization based on depth map characteristic and hardware acceleration are effective means to balance the computational complexity and coding efficiency of 3D-HEVC motion estimation, but the existing reference has not proposed a motion estimation algorithm which is easy to implement on hardware and makes full use of the characteristics of depth maps. In this paper, by adding a preprocessing module before motion estimation to classify the coding blocks, selecting different block matching search algorithms for edges and homogeneous areas to reduce the computational complexity of the homogeneous blocks while ensuring the coding quality. At the same time, the optimized algorithm is implemented in parallel based on the array processor, which accelerates the calculation speed of motion estimation and achieves a good trade off between algorithm efficiency, coding quality and hardware implementation.

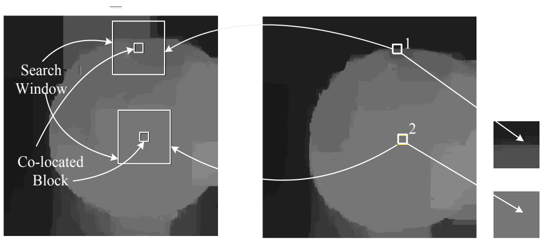
II. MOTIVATION

The purpose of motion estimation is to find the best matching block in the reference frame for the current coding block in a certain search window according to the search algorithm. The sum of absolute difference (SAD) criterion is usually used to judge the matching degree of the coding block. The lower the SAD value, the higher the matching degree of the two coding blocks. The calculation formula of SAD is shown in (1).

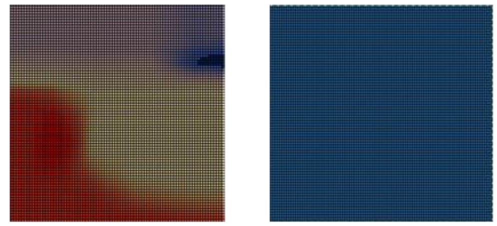
$$SAD(x,y)=\sum_{m=1}^{M}\sum_{n=1}^{N}\left|f_i(m,n)-f_{i-1}(m+x,n+y)\right| \tag{1}$$

Wherein, for the coding block of size M × N, *fi (m, n)* represents the pixel value at the coordinate *(m, n)* in the i-th frame, *fi-1 (m + x, n + y)* represents the pixel value at the coordinates *(m + x, n + y)* in the i-1 frame. *(x, y)* represents the motion vector, that is, the relative displacement between the current block and the reference block.

Fig.1 summarizes the SAD values of the coding blocks in the Balloons test sequence of the depth map. Fig.1(a)(b) is a partial area of the 45th frame and the 46th frame, respectively. The part labeled 1 and 2 shown in Fig.1(b) are two 16 × 16 coding blocks of the 46th frame. It can be seen that the coding block labeled 1 is located in the edge area, and the coding block labeled 2 is located in the homogeneous area. Their co-located blocks and search window areas in the 45th frame are shown in Fig.1 (a). Fig.1 (c) (d) shows the heat map of SAD values obtained by applying the FS algorithm to the coding block 1 and coding block 2 of the 46th frame in the corresponding search window respectively. Among them, the blue area represents a lower SAD value, and the red area represents a higher SAD value, which represents the best value and the worst value, respectively, and the yellow area and other colors are the transition of the best value and the worst value. It can be seen that the SAD value changes in the edge area of the depth map are complex, and the lower SAD value is far from the center of the search area, so the motion vector does not show obvious center offset characteristics; the SAD value of the flat area of the depth map changes smoothly, And the best SAD value is highly concentrated in the vicinity of the center of the search area, and its motion vector has a center-baised characteristic.



(a) frame 45 (b) frame 46



(c) the heat map of edge area (d) the heat map of homogeneous area

Fig. 1 the SAD distribution of edge area and homogeneous area

Therefore, the motion vectors in the edge area are too scattered, and a sophisticated search algorithm must be selected to find the best SAD value in the search window. For homogeneous areas, the center-baised distribution of its motion vectors can be used, so a lightweight fast search algorithm can be used to reduce the number of search and improve the convergence speed.

If the depth map motion estimation algorithm is to be implemented on hardware, the selected search algorithm template must not only fit the depth map SAD value distribution but also have a more natural operation parallelism. The TZSearch algorithm is recommended in the 3D-HEVC standard because it can achieve coding quality similar to the FS algorithm and can also reduce coding complexity. However, the TZSearch algorithm uses multiple search templates, such as diamond templates and raster templates, which makes the search path uncertain and the data read irregularly, which is not conducive to the parallel implementation of the algorithm. Considering that the FS algorithm has a simple calculation structure and is easy to parallel processing, this paper uses the FS algorithm for the edge area of the depth map. In addition, considering that the DS algorithm can make full use of the center-baised distribution characteristics of the motion vectors, the center point of the search window is preferentially calculated, and it has better prediction results for video sequences with small motions [10], which is very suitable for a large homogeneous area of the depth map in this paper, so the DS algorithm is selected to deal with the homogeneous area of the depth map.

III. PRE-PROCESSING MODULE

A. *Threshold judgment method of algorithm switching*

If using different search algorithms for different types of area coding blocks, a pre-processing module needs to be added to determine whether the coding block is homogeneous or edge. There are two types of commoy used edge detection methods for depth images. One is to use edge detection operators to extract the edges of the image. This method takes a lot of convolution operations during the calculation process, which increases the amount of calculation and the edge detection operator is sensitive to the image noise. The other type is to obtain the empirical threshold for distinguishing edges and homogeneous areas through statistical analysis. The reference adopting this method are: Reference [11] uses the gray level co-occurrence matrix to count the non-zero numbers in the matrix values formed by different CUs to classify the coding blocks, but there is still a problem of large amount of calculation. Reference [6] only uses the maximum difference of the four corners of the coding block to classify the depth map. If the maximum difference is greater than the threshold, it is determined as an edge block, otherwise it is determined as a homogeneous block, which greatly reduces the calculation overhead, but the case where the edge is inside the coding block will cause a misjudgment. Therefore, this paper introduces the difference degree *Pmax* to classify the depth map based on the reference [6] for this problem.

Take the depth image of the first frame of the test sequence Balloons as an example, as shown in Fig.2. Numbers 1-4 are homogeneous area coding blocks, and their pixel values have no change. Numbers 5-8 are edge area coding blocks, and their pixel values have obvious boundary changes. In this paper, the pixel values of the four corners of an encoding block are used as reference pixel values. The four reference pixel values are subtracted from all pixel values inside the encoding block to obtain the absolute difference sum, and the largest absolute difference sum is used as the *Pmax*.

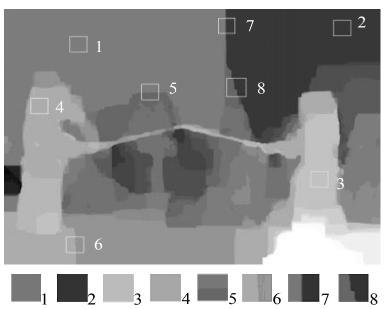


Fig. 2 the Sequence of Kendo

In order to mine the relationship between the distribution characteristics of the pixel values of the coding block and *Pmax*, this paper uses the mode selection results of 3D-HEVC intra prediction to classify the difference degree of the coding block of the depth map. In 3D-HEVC intra prediction, in addition to the conventional 35 intra prediction modes, a depth modeling model (DMM) is newly added specifically for the edge area. Therefore, if a coding block selects the normal intra prediction mode as the best prediction mode, that is, the coding block has a greater probability of being a homogeneous area, and when DMM is selected as the best prediction mode, the coding block is more likely to be an edge area.

To this end, this paper carried out prediction mode data statistics on the 3D-HEVC test platform. Under the Common Test Condition (CTC) configuration, using Kendo and Poznan_Street test sequences, the *Pmax* of 8 × 8, 16 × 16, 32 × 32 CUs of different sizes and the intra prediction selected by the CU are calculated, and statistics the probability density of *Pmax* in each interval. It is found that for the 8 × 8 block when the value is greater than 800, the probability of selecting DMM is very large, when the value is less than 800, the probability of selecting traditional intra prediction is very large, so the value 800 is used as the threshold for the 8 × 8 block. Using the same method can get the threshold of other block as shown in Table I.

Table I Threshold of different block size

BlockSize	8×8	16×16	32×32
Threshold	800	2700	6300

Table II shows the algorithm steps of the improved depth map motion estimation search algorithm.

Table II Threshold of different block size

Algorithm:depth map motion estimation based on block type
1：Compute Pmax
2：**if Pmax>TH**
3：Motion Estimation using FS
4：**else**
5：Motion Estimation using DS
6: **output:**Motion Vector

B. *Improved algorithm performance*

In order to compare the effect of the motion estimation algorithm for depth maps based on block classification, this paper uses 1-50 frames of five public video sequences with different resolutions and different motion conditions, which are simulated and verified on the MATLAB platform. The number of SAD calculations in motion estimation represents the number of algorithm searches, and the number of searches can reflect the computational complexity of the algorithm. Therefore, in this paper, the calculation complexity of the algorithm is measured by the number of SAD calculations of the algorithm, and the performance of the algorithm is measured by the PSNR of the image. Table III shows the percentage change of the search algorithm proposed in this paper compared to the FS algorithm, DS algorithm and TZ algorithm. The calculation formula is shown in formula (2).

$$SavingPt = \frac{Pt_{prop} - Pt_{compare}}{Pt_{compare}} \times 100\% \tag{2}$$

Wherein, *Ptprop* represents the search points of the search algorithm proposed in this paper, *Ptcompare* represents the search points of the FS algorithm, DS algorithm or TZ algorithm.

It can be seen from Table III that compared with the FS algorithm, the search algorithm proposed in this paper can greatly reduce the number of search points, and the search points of all test sequences are reduced by an average of 77.19%. Compared to the TZ algorithm, the search points of all test sequences are average 11.25% reduction. Compared with the FS algorithm, the Shark sequence has a smaller decrease in the number of search points, compared with the TZSearch algorithm, it is the only test sequence with increased search points. This is because many moving objects in the foreground and background areas of the Shark sequence and the motion range is large resulting edge increase of the Shark sequence. Compared with the FS algorithm and the TZSearch algorithm, the number of search points in the Poznan_Hall2 sequence is reduced by a larger amount. This is because the flat area of the sequence accounts for a higher proportion, and the motion of the object is smaller.

Table III Reduced search points after optimization

Percent Change in the Number of Search	Change in search points compared to FS	Change in search points compared to TZ
Balloons	-77.90%	-13.88%
Kendo	-77.73%	-13.00%
Poznan_Hall2	-79.54%	-19.86%
Shark	-72.46%	7.25%
Undo_Dancer	-78.34%	-16.77%

PSNR is a indicators to measure image quality. Table IV is the increased PSNR of the image compared with the FS algorithm in this paper, which is recorded as ΔPSNR. As can be seen from Table IV, compared with the FS algorithm, the average PSNR loss of the algorithm in this paper is 0.0519dB. When the search points is greatly reduced, the image quality is similar to the FS algorithm. In consideration of the search points and image quality, the algorithm proposed in this paper can reduce the number of SAD calculations by 11.25% compared to the TZSearch algorithm when the PSNR is similar to the FS algorithm, so the proposed method is slightly better than the TZSearch algorithm.

Table IV Optimized algorithm PSNR value loss

Compared algorithm	Balloons	Kendo	Poznan_Hall2	Shark	Undo_Dancer
Full search	-0.0419dB	-0.0316dB	-0.0453dB	-0.0947dB	-0.0462dB

IV. Parallelization of Depth Map Motion Estimation

A. video array processor structure

The video array processor[12] used in this paper is a reconfigurable video array processor independently developed by the project team, which supports various video codec standards such as H.264 / AVC, MVC, H.265 /HEVC. The processor logically divides the array into processing element Group (PEG), and each PEG is composed of a 4 × 4 process element (PE) array. Each PE can not only access its own memory and registers, but also access the data of other PEs in the same PEG through the communication method of adjacent interconnection and shared storage. Since most of the data processing in the video algorithm is performed by N × N rectangular blocks, this dedicated architecture can more effectively deal with the parallel design of the video algorithm than other structures.

B. Implementation of parallelization of depth map motion estimation based on array processor

Through the analysis of the motion estimation algorithm, it is found that each coding block performs block matching according to its original pixel value, reference pixel value and corresponding search algorithm to obtain the best matching block of the coding block. The data required for block matching in the next coding block does not depend on the calculation result of the previous coding block, so different coding blocks have no data correlation when calculating current block matching. Therefore, parallel operations can be used for the calculation of multiple encoding blocks to reduce encoding time.

C. Data multiplexing of reference pixels

In this paper, the natural parallel structure of the video array processor is used, and the 8 × 8 size coding block is used as the object to parallelize the motion estimation of the depth map with a search window of 16 × 16. If block matching is performed on different coding blocks in paralle. the cycle number of the motion estimation algorithm can be greatly reduced, but the circuit area and power consumption are increased. Therefore, a reasonable choice of parallel calculation of the number of coding blocks can achieve the effect of making full use of resources and reducing coding time. This paper selects the parallel-computed coding block from the perspective of maximizing the multiplexed reference pixel of the coding block read first. According to the motion estimation algorithm, there are a large number of reference pixels overlapping between the search windows of the four adjacent coding blocks in the upper, lower, left, and right. As shown in Fig.3, the reference pixels in the 9th to 16th columns of the search window of the coding block 1 are the 1st to 8th columns of the search window of the coding block 2. The reference pixels in the 9th to 16th rows of the search window of the coding block 1 are the 1st to 8th rows of the search window of the coding block 3. The 8×8 search area enclosed by the 9th row and 9th column, 9th row and 16th column, 16th row and 9th column, 16th row and 16th column of the coding block 1 is the the reference pixels enclosed by the 1th row and the 1th column, the 1th row and the 8th column, the 8th row and the 1th column, the 8th row and the 8th column of the search window of the coding block 4. The total number of pixels of the reference pixels of the coding blocks 2, 3, and 4 and the search window of the coding block 1 coincide is 196. If the search windows are loaded from off-chip storage for four coding blocks respectively, a large number of reference pixels will be read repeatedly, increasing the total number of calculation cycles. However, if the reference pixels stored locally in coding block 1 are multiplexed into the reference pixels of coding blocks 2, 3, and 4, the data multiplexing of the reference pixels in coding block 1 can reach 75%. Therefore, the number of times to access the off-chip memory is effectively reduced, and the time of the motion estimation algorithm is shortened. In summary, this paper chooses to calculate four 8×8 coding blocks in parallel.

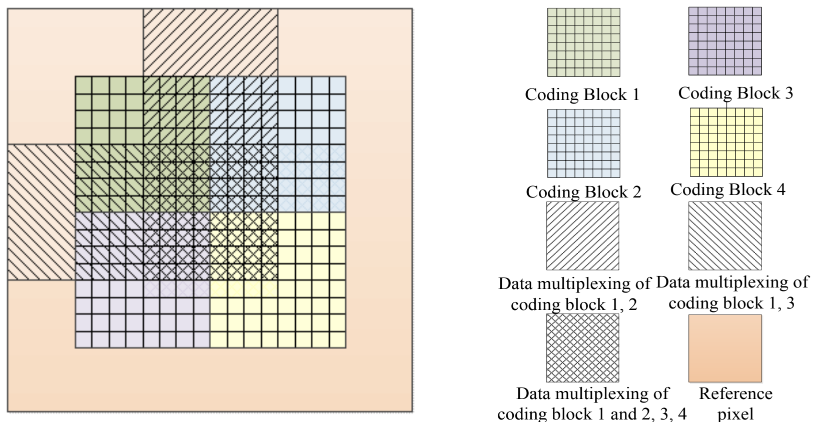


Fig. 3 Data multiplexing of search window of coding block 1

D. Depth map motion estimation algorithm parallel mapping

In this paper, a PEG is used to parallelize the four 8 × 8 coding blocks to perform the threshold calculation and block matching of the optimization algorithm and combine with the reference pixel data update method. The map of the algorithm is shown in Fig.4. Data Input Memory (DIM) and Data Output Memory (DOM) are off-chip storages, which cache the original video sequence and reference video sequence respectively;PE00 and PE33 are original pixel and reference pixel loading modules respectively; PE01, PE02, PE03 and PE13 are *Pmax* calculation modules; PE11, PE12, PE20 and PE21 are modules that use the FS algorithm to calculate the SAD value, and PE22, PE23, PE30 and PE31 are modules that use the DS algorithm to calculates the SAD value ; PE32 is the data output module. The specific steps are as follows:

Step 1: Load the original data. PE00 accesses DIM, reads a 16×16 coding block, and distributes the original pixel values of coding blocks 1, 2, 3, and 4 to PE01, PE02, PE03, and PE13;

Step 2: Load the reference data. PE33 accesses the DOM, and the reference pixel values of the coding blocks 1, 2, 3, and 4 are sent to PE11, PE12, PE20, and PE21 according to the above-mentioned method of updating the reference data.

Step 3: Four PEs calculate *Pmax* for four coding blocks in parallel. If *Pmax*> $TH_{8\times8}$, jump to the Step 4; If *Pmax* <$TH_{8\times8}$, jump to Step5.

Step 4: PE11, PE12, PE20 and PE21 perform block matching for the current coding block according to the FS algorithm. The motion vector corresponding to the best SAD value of the reference block is stored in PE32.

Step 5: PE22, PE23, PE30 and PE31 perform block matching for the current coding block according to the DS algorithm. The motion vector corresponding to the best SAD value is stored in PE32.

Step 6: PE32 outputs motion vectors.

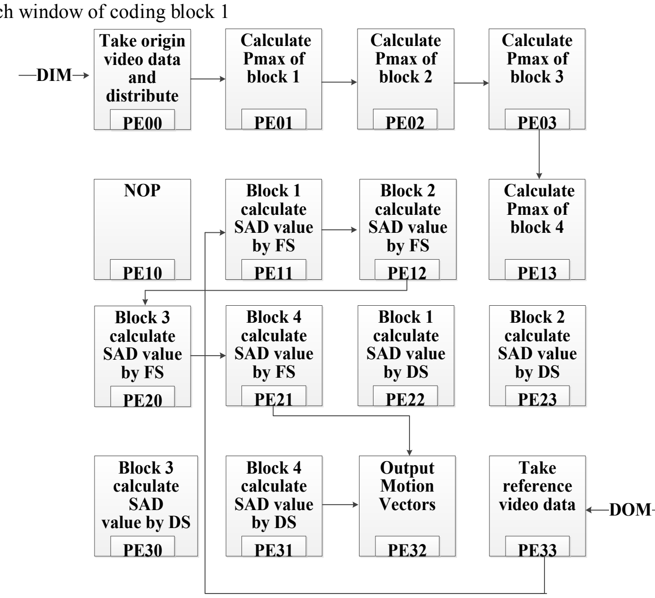


Fig. 4 Map of optimized motion estimation algorithm

V. Experimental Results and Analysis

This paper verifies the feasibility of the parallel implementation of the above depth map motion estimation algorithm based on the video array processor. Firstly convert the test sequence into a binary sequence that can be recognized by the array, then store the original frame and the reference frame in the off-chip storage DIM and DOM. Secondly initialize the instructions of the parallel scheme to instruction storage of corresponding PEs. Finally Carry out simulation verification. Measure the performance of parallelization on the video array processor through the acceleration ratio and hardware resource analysis. This article uses the serial execution time of a single PE as the serial processing time. The average serial execution time of the eight public video test sequences is 1.308 seconds, and the parallel execution time is 0.452 seconds. The average speedup can reach 2.894, which reduces the execution time by 65%. Adopt Xilinx's ISE14.7 development environment for synthesis, select BEEcube's

BEE4 series development board for FPGA test. The comprehensive results are shown in Table V. It can be seen that only 11722 LUTs and 33641 Registers of hardware resources are needed, and the maximum operating frequency can reach 122MHz. Literature [8] is an improved algorithm and hardware architecture based on the DS algorithm. It can simultaneously calculate all the search points in a PU, but at the cost of consuming a large amount of hardware resources, the hardware resources used are 8.75 times of this article. Literature [9] is a hardware architecture designed on the basis of the FS algorithm. In the case of a frequency comparable to that of this paper, the hardware resources are 1.2 times that of this paper.

Table V Comparison of hardware resources results

Structure	[8]	[9]	The paper
Chip Series	VIRTEX7	VIRTEX6	VIRTEX6
LUTs	294K	40.8K	33.6K
Registers	14K	13.2K	11.7K
Frequency	100MHZ	125MHZ	122MHZ

VI. CONCLUSION

This paper proposes an optimization method combining algorithm and parallel design for motion estimation of depth map. First, according to the characteristics of the depth map, the motion estimation search algorithm is optimized. Experimental results show that compared with the FS algorithm, the average PSNR loss of the five test sequences is only is 0.0519dB; compared with the TZSearch algorithm, the number of motion searches is reduced by 11.25% on average. Then, based on the video array processor, the proposed improved method was designed in parallel, and the number of parallel calculation coding blocks was selected from the perspective of maximizing the reuse of reference data, which further reduced the consumption of hardware resources.

REFERENCES

[1] Tech G, Chen Y, Müller K, et al. Overview of the Multiview and 3D Extensions of High Efficiency Video Coding[J]. IEEE Transactions on Circuits & Systems for Video Technology, 2016, 26(1):35-49.

[2] Si Xiaohua, Wang Guozhong,Li Guoping et al. Fast joint texture depth algorithm based on edge-modeling for 3D-HEVC coding[J]. Computer Engineering and Applications, 2016, 52(18): 183-187.

[3] Zhang Qiuwen , Zhang Na, Wei Tao, et al. Fast depth map mode decision based on depth–texture correlation and edge classification for 3D-HEVC[J].Journal of Visual Communication and Image Representation, 2017, 45:170-180.

[4] Lei Jianjun, Duan Jinhui, Wu Feng, et al. Fast mode decision based on grayscale similarity and inter-view correlation for depth map coding in 3D-HEVC[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2016, 28(3): 706-718.

[5] Saldanha M, Sanchez G, Zatt B, et al. Complexity reduction for the 3D-HEVC depth maps coding[C]//2015 IEEE International Symposium on Circuits and Systems (ISCAS). Lisbon:IEEE,2015:621-624.

[6] Saldanha M, Sanchez G , Zatt B , et al. Energy-aware scheme for the 3D-HEVC depth maps prediction[J]. Journal of Real-Time Image Processing, 2017, 13(1):55-69.

[7] Kazuyuki Miyazawa, Hiroharu Sakate, Shun-ichi Sekiguchi, et al. Real-time hardware implementation of HEVC video encoder for 1080p HD video[C]// 2013 Picture Coding Symposium (PCS). 2013.

[8] Dinh C, Nguyen T, Pham C, et al. A novel parallel hardware a rchitecture for inter motion estimation in HEVC[J]. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 2017, 9(1-3): 83-88.

[9] Aksehir Y, Erdayandi K, Ozcan T Z, et al. A low energy adaptive motion estimation hardware for H. 264 multiview video coding[J]. Journal of Real-Time Image Processing, 2018, 15(1): 3-12.

[10] Wang Wenchang. Research of Real-time Motion Estimation Algorithm Based on Video Stream in Mobile Augmented Reality System[D]. Liaoning: School of Software, Northeastern University,2017:31-32.

[11] Chen Jing, Liao Jie, Zuo Jiabao, et al. Fast Depth Intra-Coding for 3D-HEVC based on Gray-Level Co-occurrence Matrix[J]. Journal of Imaging Science and Technology, 2019, 63(3): 30406-1-30406-8.

[12] Zhu Yun,Jiang Lin,Shi Pengfei,et al. Parallelization of intra prediction algorithm based on array processor[J]. High Technology Letters, 2019,25 (01): 74-80