

Reconfigurable implementation ARP based on depth threshold in 3D-HEVC^①

Zhu Yun (朱 筠)^{②***}, Zhou Jinna^{**}, Xie Xiaoyan^{**}, Jiang Lin^{***}, Wang Shuxin^{**}, Shen Xubang^{****}
 (^{*} School of Microelectronics, Xidian University, Xi'an 710071, P. R. China)
 (^{**} Xi'an University of Posts and Telecommunications, Xi'an 710121, P. R. China)
 (^{***} Xi'an University of Science and Technology, Xi'an 710054, P. R. China)
 (^{****} Xi'an Microelectronic Technology Research Institute, Xi'an 710065, P. R. China)

Abstract

Aiming at the high computational complexity and low efficiency of the advanced residual prediction (ARP) algorithm in 3 dimension high-efficiency video coding (3D-HEVC), the relationship between the depth value and ARP is analyzed. A fast ARP algorithm based on the depth value is proposed, which is implemented on the reconfigurable array processor developed by the project team. It uses a reconfigurable method to realize flexible switching between interview-ARP and temporal ARP. Experimental results show that while keeping the coding rate and the peak signal-to-noise ratio (PSNR) basically unchanged, the coding time of the six test sequences is reduced by 16.21% on average compared with HTM16.1. In contrast with non-reconfiguration, the average coding time is reduced by 52%, so the computational efficiency is improved.

Key words: 3 dimension high-efficiency video coding (3D-HEVC), advanced residual prediction (ARP), reconfigurable method

0 Introduction

In order to adapt to the development of 3D video, the 3 dimension high-efficiency video coding (3D-HEVC) standard adds a variety of new technologies, such as inter-view prediction technology^[1]. Since both texture video and depth video inter-view prediction algorithms in the 3D-HEVC test model are executed serially, reducing the computational complexity of inter-view prediction has always been a hot spot for scholars at home and abroad.

Scholars have proposed many optimization algorithms that can reduce the computational complexity of encoding between viewpoints. One of them is to reduce the complexity of model decision-making, thereby speeding up the selection of the main candidate prediction model^[2-4].

After analyzing the coding information of the inter-view and spatio-temporal correlation, Ref. [5] proposed an adaptive method to terminate the specific mode decision early, and reduce the calculation complexity of 3D-HEVC while maintaining almost the same rate-distortion performance. Ref. [6] proposed to skip

the prediction of the mode with a coding unit (CU) size of 64×64 based on the corresponding CU motion uniformity model. It could reduce the time of each step in the texture video coding process, but at the expense of coding performance. Ref. [7] based on the motion uniformity of texture video, made the early skip/merge mode decision and adaptive motion search range adjustment of disparity estimation (DE) and motion estimation (ME). Although this algorithm saves coding time, it also reduces the coding accuracy between viewpoints.

The above method utilized texture image coding information and improved coding efficiency to a certain extent by reducing reference frames and skipping related coding information. However, the texture map and the depth map in 3D-HEVC are closely related. By fully mining the characteristics of the depth map, the coding efficiency can be effectively improved. The depth information and motion correlation are used to quickly select the advanced residual prediction (ARP) algorithm to speed up the coding speed and reduce the algorithm complexity.

In terms of parallelism and improving the compu-

① Supported by the National Natural Science Foundation of China (No. 61834005, 61772417, 61634004, 61602377), the Shaanxi Provincial Co-ordination Innovation Project of Science and Technology (No. 2016KTZDGY02-04-02).

② To whom correspondence should be addressed. E-mail: zhuyun@xupt.edu.cn
 Received on Oct. 9, 2020

tational efficiency of algorithms, the hardware implementation of multimedia algorithms has been relatively mature. How to use the reconfigurable configuration method to ensure the function of the algorithm and improve the flexibility of the algorithm has become a research hotspot. Coarse grained reconfigurable arrays (CGRA) [8] with massively parallel computing functions and reasonable power consumption has become an effective solution for multimedia applications.

The reconfigurable computing system uses a very flexible high-speed computing structure to perform parallel processing, combining the flexibility of software algorithms with the high performance of hardware [9], while taking into account the flexibility of general-purpose processors and the high efficiency of application-specific integrated circuits [10]. It has gradually become a popular architecture [11-13]. Ref. [14] studied a reconfigurable intra prediction algorithm. Ref. [15] used 64 reconfigurable interpolators to meet different interpolator types. Ref. [16] designed a reconfigurable array that supports asymmetric execution mode calculations in two states of on and off for the calculation of sum of absolute differences (SAD) values, which can maximize the use of the hardware resources of the reconfigurable array processor.

The coding efficiency of the above research has been improved, but the data volume of the video algorithm is relatively large, and there are data correlation and data independence between different algorithms. If the application changes, the circuit structure needs to be redesigned, and the flexibility is low. The reconfigurable structure not only has great flexibility in program design, but also can be reconfigured to adapt to the evolving characteristics of video algorithms and other applications, which has more important research significance. This paper proposes a fast ARP algorithm based on depth value, and uses the reconfigurable array processor developed by the project team to implement ARP.

The rest of this paper is organized as follows. Section 1 analyzes the ARP algorithm in 3D-HEVC. Section 2 introduces the reconfigurable implementation of ARP algorithm based on depth threshold. Experimental results are shown in Section 3. Finally, Section 4 concludes this paper briefly.

1 Related work

1.1 ARP

The core of ARP in the 3D-HEVC inter-view prediction algorithm is to use the inter-view's residual information to reduce the redundancy of inter-views. The

principle of ARP is shown in Fig. 1, where V_0 represents a basic viewpoint, V_1 represents a non-basic viewpoint, D_c represents the current coding block, D_r represents the temporal reference block of the currently coded view, B_c represents the inter-view reference block, and B_r represents the temporal reference block of the base view. According to the reference block type of the current block, it is divided into temporal ARP and inter-view ARP.

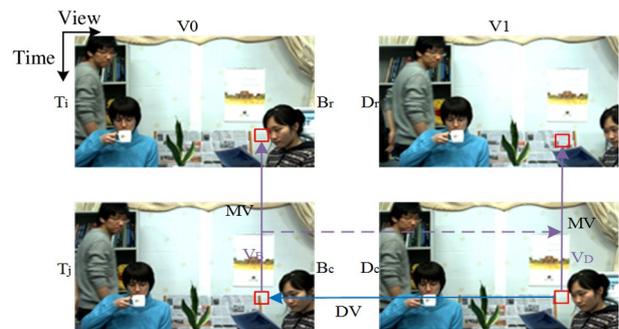


Fig. 1 Schematic diagram of ARP

There is redundant information in time and space between different views. When encoding ARP, motion estimation is used between the same views to improve the coding efficiency of motion compensation prediction (MCP). Disparity estimation is used between different views to improve the efficiency of disparity compensation prediction (DCP). Therefore, the entire prediction process has a higher computational complexity.

Since the time interval between two adjacent frames of the same view is very short, the efficiency of MCP in the time direction is higher than DCP. In a local area block, the correlation in inter-view of pixels between different views may be greater than the correlation in time. At this time, the efficiency of DCP in inter-views may be higher than MCP in time. This paper considers the correlation between depth information and motion, and proposes a fast selection ARP algorithm, which can reduce the complexity of the algorithm and save coding time.

The framework of ARP algorithm is shown in Fig. 2, where block 1 represents the current coding block at the current moment, block 2 represents the reference block between viewpoints at the current moment, block 3 represents the temporal reference block of the reference viewpoint, block 4 represents the temporal reference block of the current viewpoint.

If the reference block of the current block is a temporal reference block, temporal ARP is used at this time according to Eq. (1). If it is an inter-view reference block, inter-view ARP is used, according to Eq. (2). For the prediction accuracy, the residual in-

formation on the equations introduces a weighting factor w , which is 0, 0.5 and 1 respectively.

$$Block1' = Block4 + w \times (Block2 - Block3) \quad (1)$$

$$Block1' = Block2 + w \times (Block4 - Block3) \quad (2)$$

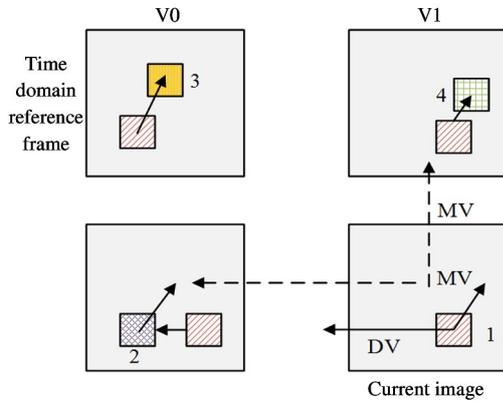
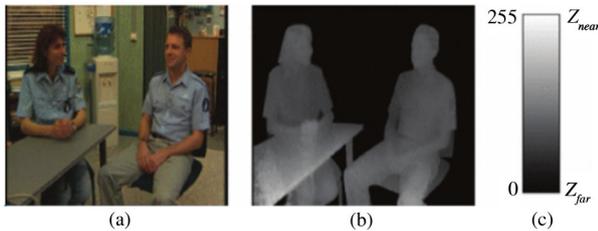


Fig. 2 ARP framework

1.2 Statistical analysis of ARP with different depth values

Fig. 3(a) represents a texture view of an image. Fig. 3(b) shows the corresponding depth view, where the white area represents the closer distance in the 3D



(a) Texture map; (b) Corresponding depth map;
(c) The relationship between color intensity and depth value

Fig. 3 Texture information and depth information

space, and the black area represents the longer distance. This paper divides the current coded macroblock into near-area, middle-area and far-area according to the depth value. As shown in Fig. 3(c), the near-area is marked as Z_{near} , and the far-area is marked as Z_{far} . The values of Z_{near} and Z_{far} are 255 and 0, respectively.

In order to obtain the correlation between the depth value and ARP, this paper divides the current coded macroblock into near-area, middle-area and far-area according to the depth value. Set the depth thresholds Z_0 and Z_1 as the judgment of the current macroblock according to Eq. (3). Suppose $Z_0 = Z_{near}$, $Z_1 = Z_{far}$, and then perform statistics on the selection results of the temporal ARP and inter-view ARP in different areas.

$$\begin{cases} Z_{CU} > Z_0 & CU \in near-area \\ Z_1 \leq Z_{CU} \leq Z_0 & CU \in middle-area \\ Z_{CU} < Z_1 & CU \in far-area \end{cases} \quad (3)$$

If the selection times of temporal ARP and inter-view APR are both 0 or if one of them is 0, Z_0 and Z_1 will decrease by 5 and increase by 5 respectively. Repeat the above steps until the number of executions of temporal ARP and inter-view ARP algorithms in different regions is counted, the final depth thresholds Z_0 and Z_1 can be obtained.

This paper counts the selection ratio of temporal ARP and inter-view ARP in different areas under different test sequences based on depth threshold. The texture maps and depth maps quantization parameter (QP) include (25, 34), (30, 39), (35, 42) and (40, 45), and the statistical results are shown in Table 1.

Table 1 Statistical results of the selection ratio of temporal ARP and inter-view ARP

Sequences	Z_0	Z_1	near-area		middle-area		far-area		
			inter-view ARP	temporal ARP	inter-view ARP	temporal ARP	inter-view ARP	temporal ARP	
1024 × 768	Balloons	180	90	10.30%	89.70%	6.80%	93.20%	5.20%	94.80%
	Kendo	150	70	19.40%	80.60%	10.20%	89.80%	3.90%	96.10%
	Newspaper	150	70	10.50%	89.50%	9.40%	90.60%	3.90%	96.10%
1920 × 1088	GT_Fly	180	90	5.10%	94.90%	12.70%	87.30%	11.70%	88.30%
	Poznan_Hall2	80	20	1.70%	98.30%	5.30%	94.70%	5.30%	94.70%
	Poznan_Street	150	70	4.40%	95.60%	19.80%	80.20%	11.40%	88.60%
	Undo_Dancer	130	70	5.10%	94.90%	10.60%	89.40%	10.60%	89.40%

1.3 Reconfigurable array processor

The array processor used in the experiment is shown in Fig. 4, including 4×4 processor elements (PEs), a hierarchical configuration network based on H-tree based reconfiguration mechanism (HRM) and a global controller. HRM provides a solution to realize

dynamic reconfiguration of different algorithms. The global controller determines the operation mode and selects appropriate functions of one or more PEs, and then unicasts and publishes the reconstruction configuration information on the HRM.

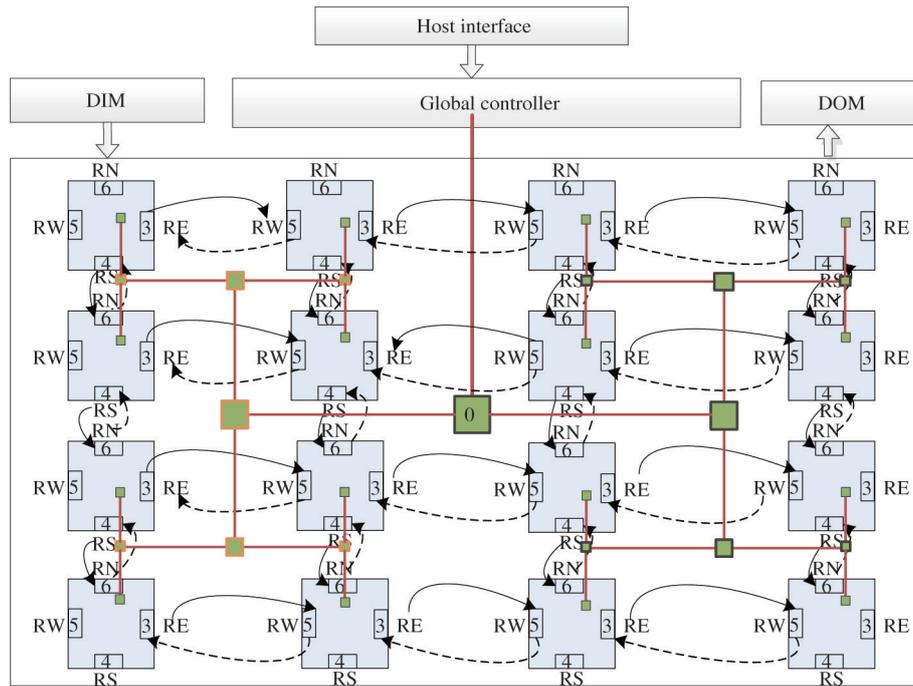


Fig. 4 Diagram of reconfigurable array processor with HRM

2 Implementation of reconfigurable ARP based on depth threshold

2.1 Optimized ARP based on depth threshold

According to the different types of input test sequences, the ARP algorithm is optimized based on the depth threshold. The process is shown in Fig. 5.

First, determine the area to which the current coded macroblock belongs according to the depth thresh-

old. If the current macroblock is a near area, the type A sequence selects temporal ARP, skipping the inter-view ARP, type B or C sequence selects temporal ARP and inter-view ARP. If the current macroblock is a far area, the type B sequence executes temporal ARP, skipping the inter-view ARP; the type A or type C sequence chooses to execute temporal ARP and inter-view ARP. In other cases, choose to execute temporal ARP and inter-view ARP.

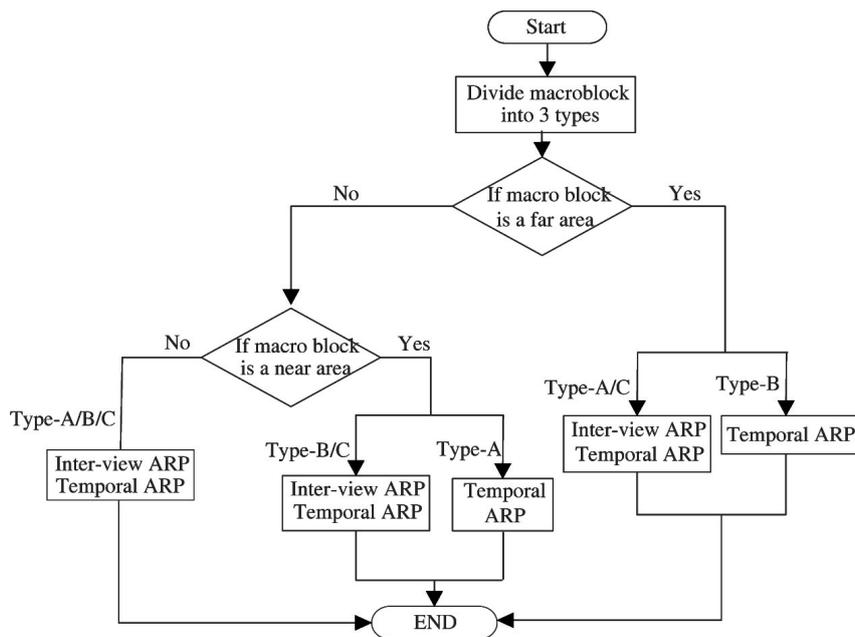


Fig. 5 Optimized ARP flowchart

2.2 Reconfigurable ARP implementation

On the array processor that the project team studied early, this paper proposes a reconfigurable APR implementation method, based on the depth threshold to select switch between temporal APR and inter-view APR. This method can effectively reduce unnecessary hardware resource consumption. Fig. 6 shows a schematic diagram of reconfigurable ARP by using a 4×4 array structure and transmitting related commands through the global controller. The specific implementation process is as follows.

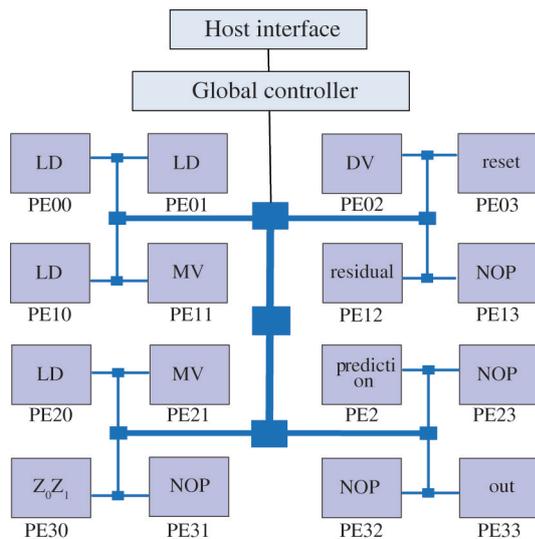


Fig. 6 Reconfigurable scheme of ARP algorithm

Step 1 Data preparation.

External data and commands are stored on the host. Instructions or configuration information are loaded into on-chip memory.

PE00 loads the original block data from the external data input memory (DIM), and then distributes the data onto PE02 and PE21 through the shared storage in PE. PE01, PE10, and PE20 load Dr, Bc, and Br, respectively. Bc sends the data onto PE11 when it obtains the best prediction block.

Step 2 Judgment.

The depth thresholds Z_0 and Z_1 is stored in PE30. The area where the current macroblock is located is judged by Eq. (3). If the current macroblock is a far-area, the flag 8888 is stored in address 160 of PE30. If the current macroblock is a near-area or middle-area, the flag 8888 is stored in address 160 of PE30, and the flag 9999 is stored in address 161. After the judgment is over, two flags can be obtained through the HRM feedback network, where 8888 represent the temporal ARP, and 9999 represents the inter-view ARP.

Step 3 Issue instructions.

If the flag obtained by HRM is only 8888, the temporal ARP algorithm instruction is issued. The PEs that can receive instructions are PE01, PE02, PE03, PE10, PE11, PE12, PE20, PE21, PE22 and PE33. After all configuration information is issued, use the CALL instruction to start these PEs works.

If the HRM detects the flags 8888 and 9999 at the same time, the temporal ARP algorithm is issued first, and then the completion flag is written into the shared memory after completing the execution. When HRM detects this completion flag through the feedback network, the inter-view ARP algorithm is issued. The specific PEs issued are PE00, PE01, PE02, PE03, PE10, PE11, PE12, PE20, PE21, PE22, PE30, and PE33. After all configuration information is issued, use the CALL instruction to execute these PEs.

Step 4 Execution.

PE11 calculates the prediction block of the Br base view temporal reference block. PE02 calculates the prediction block of Bc. PE21 calculates the prediction block of Dr. The reference block with the smallest SAD value calculated by the three-step search method is the optimal prediction block found. The Br prediction block and the Bc prediction block are used as residuals to obtain temporal ARP residual data, which is stored in PE12. The data obtained by summing the prediction block of the Dr time-domain reference block in PE21 and the residual data value in PE12 is stored in PE33. Finally, the data is output to the external data output memory (DOM), and a block of temporal ARP prediction can be obtained.

PE31 calculates the prediction block of the reference block between Bc views. PE11 calculates the prediction block of the temporal reference block. PE21 calculates the prediction block of the Br base view temporal reference block. The data in PE11 and PE21 are subtracted one by one, and then the residual block information is obtained and stored in PE22. The data in PE31 and PE22 are summed one by one, the final information is stored in PE33, and finally output to the DOM, then an inter-view ARP prediction block can be obtained.

3 Experimental results and performance analysis

3.1 Experimental test conditions

The test sequences in this paper includes 1920×1088 sequences (*Undo _ Dancer*, *Poznan _ Hall2*, *Poznan _ Street*) and 1024×768 sequences (*Kendo*, *Balloons*, *Newspaper*), using the 3D-HEVC Test Model (HTM) version 16.1, compiled by Visual Stu-

dio 2017. The texture map and depth map quantization parameters QP are specified as (25, 34), (30, 39), (35, 42) and (40, 45). Three viewpoints are used for coding, and the coding sequence is from the middle viewpoint to the left viewpoint and then to the right viewpoint. The middle viewpoint is the basic viewpoint. The coding test parameters are shown in Table 2.

Table 2 Experimental test parameters

Sequence number	Test sequence	Resolution	Frame rate	Viewpoint order
S01	Poznan _ Hall2	1920 × 1088	25	6-7-5
S02	Poznan _ Street	1920 × 1088	25	4-5-3
S03	Undo _ Dancer	1920 × 1088	25	5-1-9
S04	Kendo	1024 × 768	30	3-1-5
S05	Balloons	1024 × 768	30	3-1-5
S06	Newspaper	1024 × 768	30	4-2-6

3.2 Performance analysis

As shown in Table 3, the $\Delta Bitrate$, $\Delta PSNR$ and Enc_{time} compared with HTM16.1 under different QPs

Table 3 Coding performance comparison between the proposed algorithm and the original HTM16.1

Sequence	QP	$\Delta Bitrate$	$\Delta PSNR/dB$	Enc_{time}
Poznan _ Hall2	25/34	0.3833	-0.0332	8.90%
	30/39	0.5100	0.0236	9.30%
	35/42	0.7334	-0.1219	10.30%
	40/45	0.0800	0.0099	11.30%
Poznan _ Street	25/34	2.3570	-0.0327	18.50%
	30/39	2.1234	-0.0630	20.80%
	35/42	0.3600	-0.0455	22.30%
	40/45	-0.1367	0.0304	23.50%
Undo _ Dancer	25/34	10.8433	-0.0422	19.70%
	30/39	5.6933	0.0214	21.70%
	35/42	1.1733	0.0984	23.70%
	40/45	0.9534	-0.0076	25.00%
Kendo	25/34	-48.1693	-0.0064	18.70%
	30/39	-30.5920	0.0212	18.80%
	35/42	-18.4840	-0.0396	17.40%
	40/45	-10.6160	0.0416	17.30%
Balloons	25/34	0.7160	-0.0472	15.10%
	30/39	0.2680	-0.0129	13.50%
	35/42	-0.0120	0.0124	10.10%
	40/45	-0.0720	-0.0385	15.30%
Newspaper	25/34	0.4000	0.0081	12.30%
	30/39	0.4467	-0.0002	12.90%
	35/42	0.4840	-0.0568	14.20%
	40/45	-0.0213	-0.0325	14.10%
Average	-	-3.3574	-0.0131	16.45%

are respectively given. It can be seen from the experimental results that the optimized ARP proposed in this paper based on the depth threshold keeps the *Bitrate* and the peak signal-to-noise ratio (PSNR) basically unchanged, but Enc_{time} is reduced by 16.45% on average.

Comparing the optimized ARP algorithm proposed in this paper with HTM16.1, Fig. 7 shows PSNR and structural similarity (SSIM) comparison curves of the synthesized image. In Fig. 7(a), most of the PSNR curves of this method are higher than the PSNR curve of HTM16.1. A few PSNR curves are lower than the PSNR curves of HTM16.1, and the maximum difference is less than 2 dB. In Fig. 7(b), compared with HTM16.1, only the average SSIM value of S03 is reduced by 0.00916, the other test sequences' SSIM value have been improved. Experiments indicate that the method will not affect the synthesized video quality.

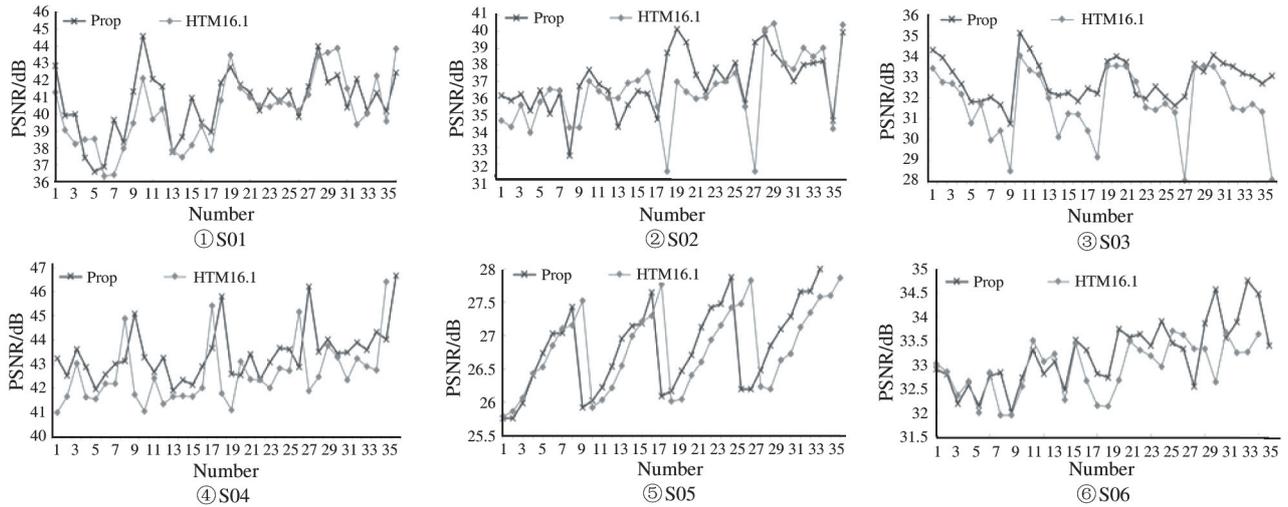
3.3 Reconfigurable experimental results analysis

An optimized ARP algorithm based on depth threshold is proposed. It uses a data-level parallel method of reconfigurable implementation of temporal ARP and inter-view ARP. Table 4 shows the calculation time of the algorithm during the reconfiguration of every test sequence. Compared with non-reconfiguration, the experimental results show that the average encoding time after reconstruction is reduced by about 52%. This can effectively improve the computational efficiency of the algorithm.

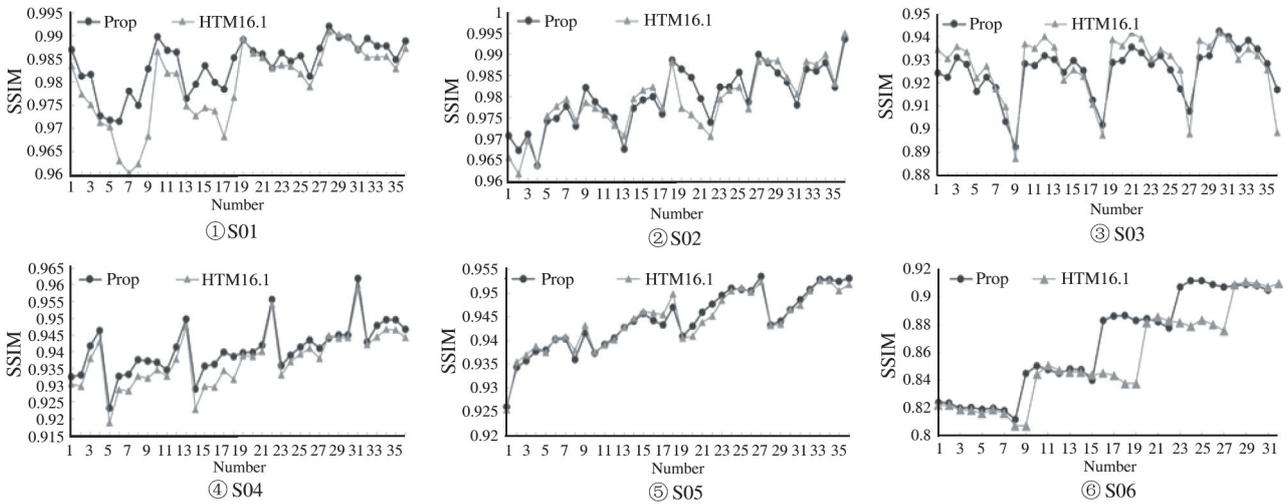
Table 5 shows the test results using Xilinx's Virtex-6 XC6VLX550T FPGA chip. The experimental results show that the reconfigurable method saves 42.2% of Slice Registers and 46.8% of LUTs, saving unnecessary hardware overhead. When switching between temporal ARP and inter-view ARP, it is not necessary to redesign the circuit structure. This reconfigurable method can be flexibly configured, which provides much more flexibility than specific integrated circuit.

4 Conclusion

Through statistical analysis of the relationship between depth value and ARP algorithm, this paper proposes an optimized ARP algorithm based on depth threshold. Experimental results show that compared with HTM16.1, the coding time of the test sequence is reduced by 16.21% on average when the coding rate and PSNR remained basically unchanged. The array processor developed by the project team is reconfigurable to implement the flexible switching of temporal ARP



(a) PSNR comparison of images after synthesis



(b) SSIM comparison of images after synthesis

Fig. 7 Comparison of image parameters after synthesis

Table 4 Comparison of coding time before and after reconstruction

Sequence	No-reconfigurable/ns	Reconfigurable/ns	Δ Time/%
Poznan _ Hall2	1 179 762	606 412	48.6
Poznan _ Street	1 244 032	633 859	49
Undo _ Dancer	1 254 156	627 078	50
Kendo	1 265 404	645 356	49
Balloons	1 758 490	633 095	64
Newspaper	1 299 110	675 537	48
Average	1 333 492	696 602	52

Table 5 Performance comparison of hardware parameters

---	Frequency/MHz	Slice Registers/k	Slice LUTs/k	RAM/FIFO
Only temporal ARP	123.409	13.7	39	41.5
Only inter-view ARP	123.409	14	40	41
Reconfigurable both of them	123.320	16	42	31.5

and inter-view ARP. Compared with non-reconfiguration, the average coding time is reduced by 52%, which improves the computational efficiency of the algorithm and reduces the consumption of hardware resources.

Reference

- [1] Chan Y L, Fu C H, Chen H, et al. Overview of current development in depth map coding of 3D video and its future[J]. *IET Signal Processing*, 2019, 14(4):1-14
- [2] Lei J, Sun J, Pan Z, et al. Fast mode decision using inter-view and inter-component correlations for multiview depth video coding[J]. *IEEE Transactions on Industrial Informatics*, 2015, 11(4):978-986
- [3] Shen L Q, Zhang Z Y, Zhang X P, et al. Fast TU size decision algorithm for HEVC encoders using Bayesian theorem detection[J]. *Signal Process: Image Communication*, 2015, 32(10):121-128
- [4] Zhang Q W, Zhao J C, Huang X P, et al. A fast and efficient coding unit size decision algorithm based on temporal and spatial correlation[J]. *Optik*, 2015, 126(21):2793-2798
- [5] Zhang Q W, Chang H W, Huang X P, et al. Adaptive early termination mode decision for 3D-HEVC using inter-view and spatio-temporal correlations[J]. *AEU-International Journal Electronics and Communications*, 2016, 70(5):727-737
- [6] Zhang Q W, Wei S C, Su R J. Low-complexity texture video coding based on motion homogeneity for 3D-HEVC[J]. *Scientific Programming*, 2019, 2019: 1-13
- [7] Liu S, Wei S, Zhang Q. Low complexity texture video coding for multi-view video system reliability[J]. *International Journal of Performability Engineering*, 2018, 14(8):842-1847
- [8] Tu F B, Yin S Y, Ouyang P, et al. Reconfigurable architecture for neural approximation in multimedia computing[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019, 29(3):892-906
- [9] Vliet L W, Corporaal H. Coarse grained reconfigurable architectures in the past 25 years: overview and classification[C]//2016 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation (SAMOS), Agios Konstantinos, Greece, 2016: 235-244
- [10] Akbari O, Kamal M, Afzali-Kusha A, et al. PX-CGRA: polymorphic approximate coarse-grained reconfigurable architecture[C]//2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 2018: 413-418
- [11] Ahn M, Yoo D, Ryu S, et al. The acceleration of various multimedia applications on reconfigurable processor[C]//2013 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, USA, 2013: 238-239
- [12] Zhu Y, Jiang L, Wang S, et al. Design of reconfigurable array processor for multimedia application[J]. *Multimedia Tools and Applications*, 2018, 77(3): 3639-3657
- [13] Raghavachari R, Sridharan M. Review of recent trends in coarse grain reconfigurable architectures for signal processing applications[J]. *Advances in Systems Science and Applications*, 2018, 18(1): 41-58
- [14] Azgin H, Mert A C, Kalali E, et al. Reconfigurable intra prediction hardware for future video coding[J]. *IEEE Transactions on Consumer Electronics*, 2017, 63(4):419-425
- [15] Pastuszak G, Trochimiuk M. Architecture design of the high-throughput compensator and interpolator for the H.265/HEVC encoder[J]. *Journal of Real-Time Image Processing*, 2016, 11(4): 663-673
- [16] Jiang L, Wu X, Cui J X, et al. Dynamic reconfigurable implementation of SAD algorithm in HEVC motion estimation[J]. *Journal of Beijing University of Posts and Telecommunications*, 2018, 41(4):37-43 (In Chinese)

Zhu Yun, born in 1981. She received her B. S. and M. S. degrees in Microelectronics and Solid State Electronics from Lanzhou University in 2003 and 2006 respectively. She joined the faculty of Xi'an University of Posts & Telecommunications. Since 2014, she has been working on the Ph. D. degree at Xidian University. Her research interests are in the field of array processors.