New View Generation Method for Free-Viewpoint Video System

GANGYI JIANG*, LIANGZHONG FAN, MEI YU AND FENG SHAO
Faculty of Information Science and Engineering
Ningbo University
315211 Ningbo
CHINA
jianggangyi@126.com

Abstract: - View navigation is an attracted function of free viewpoint video (FVV) system. Disparity map which describes correspondence between adjacent views plays an important role in view generation. However, disparity estimation is complicated and time consuming for user side with constrained resource environment. In view of disparity compensation prediction algorithm is usually utilized at the encoder to lift up the coding efficiency, block based disparity map can be easily obtained without more extra efforts. Therefore, a framework of view generation oriented FVV system is presented in this paper, in which block based disparity map is generated and encoded with CABAC losslessly at the server side, while at the user side, the received block based disparity map is refined for view generation, and spatial correlation in block based disparity map is utilized to accelerate the generation of pixel-wise disparity map. Experimental results show that the proposed framework makes a tradeoff between the transmission cost and effort of view generation.

Key-Words: - Free viewpoint video, view generation, block based disparity map, disparity refinement, ray-space.

1 Introduction

It has been recognized that multi-view video signal processing is a key technology for a wide variety of future applications, including free viewpoint video (FVV) system, three-dimensional television, immersive teleconference and surveillance etc [1,2]. FVV can offer a 3D depth impression of the observed scenery and allow interactive selection of viewpoint and direction within a certain operating range, which is regarded as the most challenging audio-video application scenario[3,4].

A classical FVV system normally includes the components of multi-view capture, pre-processing, multi-view video coding (MVC) encoder, transmission, MVC decoder, view generation and display[5]. At the server side, the most difficult problem is how to improve the coding efficiency of multi-view video. While at user side, the most time-consuming parts are view generation and MVC decoding. Therefore, how to reduce the complexity of these two parts will alleviate the burden of a receiver. From view-point of the user, view generation should be fast so that real-time navigation can be achieved. Additionally, the quality of synthesized view should also be high enough to support the realistic impression.

In the JVT 23th meeting, multi-view video plus depth (MVD) as basic format for advanced future 3DTV[4] and FVV[6] systems which fulfills the requirement of rendering a wide range continuum of views at the decoder was proposed by Smolic and Mueller[7]. And then, Merkle, Smolic studied on the multi-view video plus depth representation and found out that coding artifacts on depth data strongly influences the reconstruction quality of rendered arbitrary views in a FVV scenario[8]. As a result, high bit rates are needed for depth data compression or doing post-processing on the decoded depth maps to instead.

Image-based rendering refers to a collection of techniques and representations that allow 3D scenes and objects to be visualized in a realistic way without full 3D model reconstruction[9,10]. For interactive photo-realistic applications, accurate pixel-wise disparity information is generally used to synthesize high quality novel views[11]. Light field, ray-space and lumigraph are similar ideas which can model radiances of a scene by a 4D plenoptic function[12-16]. A novel view from an arbitrary position and direction can be generated by appropriately combing image pixels from the existing views[4]. However, these benefits have to be paid by dense sampling of the real world with plenty of original view images. In real situation, only sparse multi-view images are compressed by MVC scheme and transmitted to the user side, and dense intermediate view images are generated by some interpolating means[5]. To interpolate dense intermediate view images, accurate disparity map, which determines the quality of synthesized image, will be calculated beforehand.

This paper is focused on arbitrary view rendering using block-based disparity map generated at the server side. Transmission cost of block based disparity map is discussed. Quality of reconstructed views is compared between two schemes, that is, block based disparity map scheme and pixel-wise disparity map scheme. This paper is organized as follows: Section 2 gives the framework of view generation oriented FVV system. Sections 3 to section 5 describe three important components including generation and compression of block based disparity map, disparity map refinement and arbitrary view generation in the proposed framework. Section 6 reports some experimental results on different test images and Section 7 gives the conclusion.

2 Framework Of View Generation Oriented Free Viewpoint Video System

In many applications, user side is a terminal with constrained resource environment, such as TV set, mobile phone or PDA etc. Its capability of memory and processing is not powerful enough to process complicated operations. Therefore, to realize free viewpoint navigation in these devices will confront with many technical problems.

Compared with the resource limitation at the user side, the complexity of server side can be left out of account. Thus, moving part of time-consuming task related with view generation to the server side is an alternative scheme. Fig.1 gives the proposed framework of view generation oriented FVV system, in which block based disparity map is generated and encoded with CABAC losslessly at the server side. At the user side, the received block based disparity map is refined for view generation, and spatial correlation in block based disparity map is utilized to accelerate the generation of pixel-wise disparity map. By using the refined pixel-wise disparity map, dense ray-space volume is obtained so that novel views can be rendered easily.

The new framework is based on the following considerations:

1) To improve coding efficiency, correlations among multi-view videos are usually exploited. Disparity compensation prediction algorithm is used to calculate the correspondences between adjacent views. Thus the block based disparity map can be easily obtained at MVC encoding step without more extra computational efforts.

2) For view generation at the user side, pixel-wise disparity map is needed. Thus, if pixel-wise disparity map is generated and transmitted to the user side, disparity estimation can be avoided for the user side. However, in this case, the total bit rate will increase obviously because of the extra pixel-wise disparity map. Taking into account the tradeoff between the total bit rate and the resolution of disparity map, block based disparity map is adopted in our scheme. Since the size of block based disparity map is smaller than that of the original image, and the values of disparity map is restricted in the range of maximum disparity, the increased bit rate is small compared with original multi-view coding. While at the user side, even though the refinement of block based disparity map has to be done to obtain pixel-wise disparity map, the effort is not so hard since the disparity estimation can be restricted within a small range indicated by the coarse disparity provided by the block based disparity map.

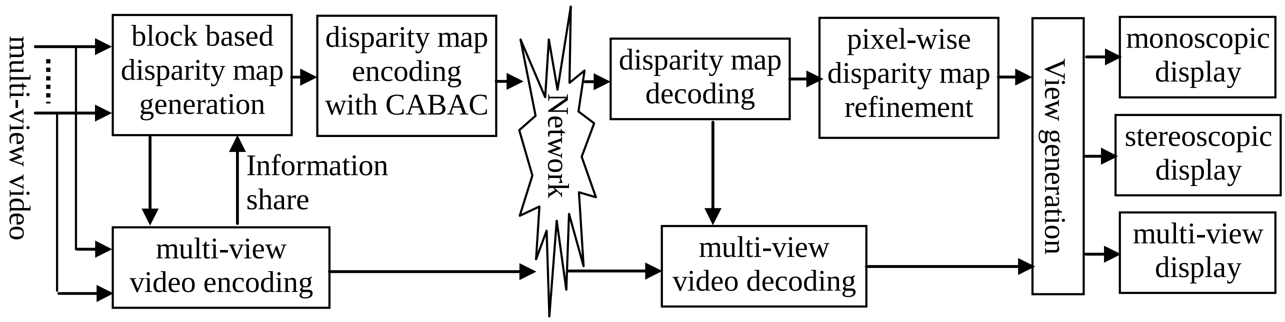
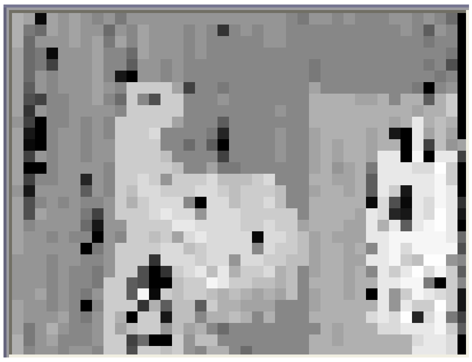


Fig.1: Framework of view generation oriented FVV system

(a) original view 1 of Akko&Rayo

(b) 4×4 block based disparity map

(c) 8×8 block based disparity map



(d) 16×16 block based disparity map

Fig.2: Disparity maps with different resolutions

3 Generation and Compression of Block Based Disparity Map

3.1 Discussion on resolution of block based disparity map

The resolution of disparity map not only influences the bit rate of the coded bit-stream but also the quality of rendered images. High resolution favors view generation but results low coding efficiency, while low resolution lowers the quality of rendered image if extra effort on disparity map refinement is left out. Fig.2 gives block based disparity maps with different resolutions, that is, 4×4, 8×8 and 16×16 block based disparity maps. In the figure, 4×4 block based disparity map is with the size of 160×120, while the size is 80×60 and 40×30 for 8×8 and 16×16 disparity maps, respectively. From the figure, it is seen that as the size of block increases, the accuracy of disparity map is reduced greatly.

The disparity map is encoded with CABAC losslessly because error of disparity resulted from lossy coding degrades quality of rendered image significantly. Tables 1-3 give comparison of the above three disparity maps on bit rate where bit rates of corresponding two views resulted from Simulcast and HBP coding schemes are used as benchmark. In Table 2 and Table 3, the proportion η is defined by

$$\eta = \frac{bit\ rate\ of\ disparity\ map}{bit\ rate\ of\ disparity\ map\ and\ two\ views} \quad (1)$$

It is seen that bit rate of 4×4 block based disparity map brings about 6%~8% extra bit rate compared with traditional multi-view coding schemes when QP=22. The data are 1%~2% and 0.5% for 8×8 and 16×16 block based disparity maps, respectively. If QP is lifted up to 32, the proportion of bitstream of disparity map is increased in comparison with bitstream of multi-view video. The proportion is around 20% for 4×4 block based disparity map, which indicates that the transmission cost for 4×4 block based disparity map is too high. Even though 16×16 block based disparity map only brings no more than 2% extra bitstream, its usability is limited due to the lack of precision. 8×8 block based disparity map seems to make a tradeoff between transmission cost and effort of view generation.

3.2 Generation of block based disparity map

To obtain more accurate and smooth disparity map, a disparity estimation method considering neighborhood constraints is used at the server side. Let I_L and I_R represent the left and right images respectively, (x,y) denote the current block to be estimated. An energy function including data item E_{data} and smoothness item E_{smooth} is defined as

$$E(x, y, dv)=E_{data}(x, y, dv)+\lambda E_{smooth}(x, y, dv) \quad (2)$$

$$E_{data}(x,y,dv) = \sum_{i=-w}^{w}\sum_{j=-w}^{w} | I_L(x+j, y+i) - I_R(x+dv+j, y+i) | \quad (3)$$

$$E_{smooth} = \frac{1}{4}\sum_{i=0}^{3} | dv - dv_i | \quad (4)$$

The smoothness item utilizes neighboring disparity vectors to enhance smoothness of disparity map. The relationship between dv and dv_0~dv_3 is illustrated in Fig.3, where dv is disparity of current block. Smooth factor λ is a constant that controls the influence of the smoothness item on the energy function. Fig.4 shows the influence of smooth factor λ on 8×8 block based disparity map generation. It is seen that the higher the λ is, the smoother the disparity map is.

dv_0	dv_1	dv_2
dv_3	dv	

Fig.3: Smoothness constraints between disparity vectors

Table 1. Bit rates of block based disparity maps in comparison with bit rates of two views unit: kbps

Views			View_01	View_12	View_23	View_34
Bit rate of two views	Simulcast	QP=22	2963.62	2934.86	3021.26	3099.05
		QP=32	944.46	919.62	966.09	989.18
	HBP-MVC	QP=22	3466.75	3466.75	3827.23	3827.23
		QP=32	951.95	951.95	1009.69	1009.69
Bit rate of disparity map	4×4 block based disparity map		256.71	239.24	251.44	263.73
	8×8 block based disparity map		60.83	54.46	58.5	60.34
	16×16 block based disparity map		13.76	12.42	13.28	13.23

Table 2. Proportion of bit rate of block based disparity map in total bit rate of two views and disparity map (*QP*=22)

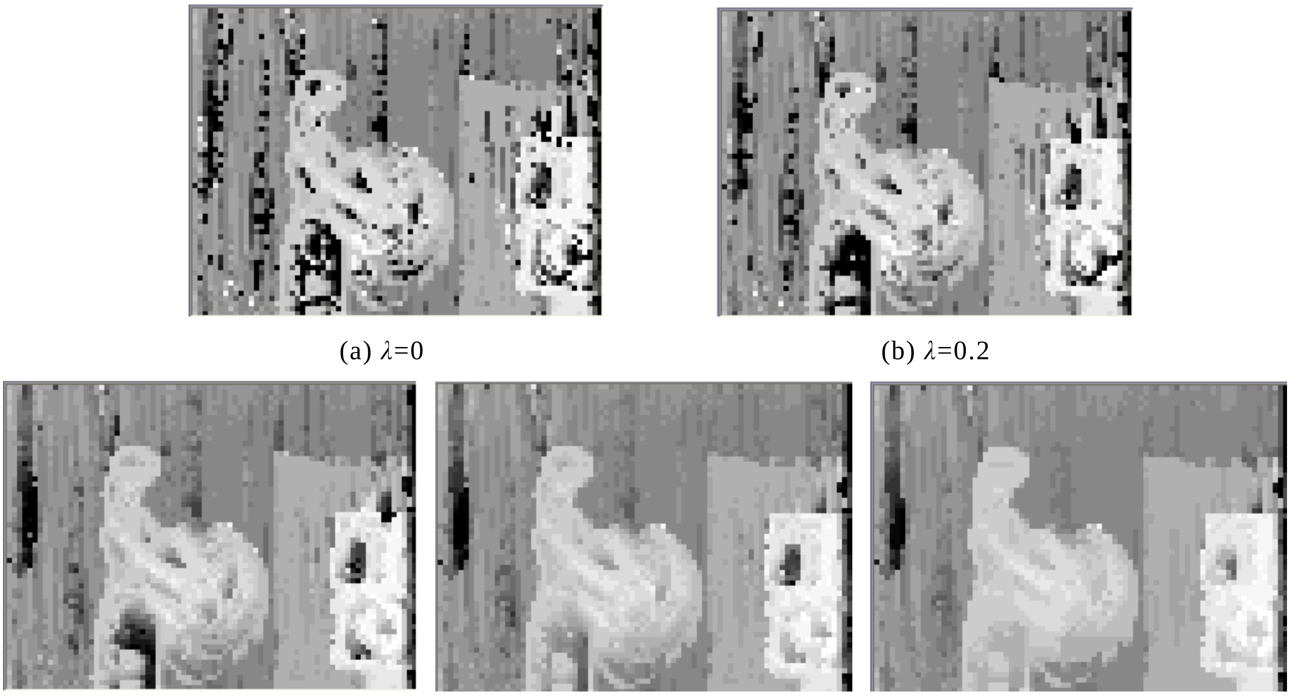
Disparity map	Coding schemes	Proportion η			
4×4 block based	Simulcast	7.97%	7.54%	7.68%	7.84%
	HBP-MVC	6.89%	6.46%	6.16%	6.45%
8×8 block based	Simulcast	2.01%	1.82%	1.90%	1.91%
	HBP-MVC	1.72%	1.55%	1.51%	1.55%
16×16 block based	Simulcast	0.46%	0.42%	0.44%	0.43%
	HBP-MVC	0.40%	0.36%	0.35%	0.34%

Table 3. Proportion of bit rate of block based disparity map in total bit rate of two views and disparity map (*QP*=32)

Disparity map	Coding schemes	Proportion η			
4×4 block based	Simulcast	21.37%	20.64%	20.65%	21.05%
	HBP-MVC	21.24%	20.08%	19.94%	20.71%
8×8 block based	Simulcast	6.05%	5.59%	5.71%	5.75%
	HBP-MVC	6.01%	5.41%	5.48%	5.64%
16×16 block based	Simulcast	1.43%	1.33%	1.36%	1.32%
	HBP-MVC	1.42%	1.29%	1.30%	1.29%

Table 4. Influence of λ on compression efficiency of 8×8 block based disparity map and reconstruction of right view

Smooth factor λ	0	0.2	0.5	0.8	1
Reconstructed right view (dB)	31.27	31.26	31.22	31.19	31.17
Bit rate of disparity map (kbps)	60.83	55.68	46.26	39.99	34.90



(a) λ=0 (b) λ=0.2

(c) λ=0.5 (d) λ=0.8 (e) λ=1

Fig.4: Influence of smooth factor λ on 8×8 block based disparity map generation

Smooth factor λ not only influences disparity map itself, but also compression efficiency of disparity map and quality of reconstructed image. Table 4 shows such influences. It is seen that the bit rate of compressed 8×8 block based disparity map with respect to different λ varies greatly due to different smoothness of disparity map. When λ=1, the bit rate is almost half of the bit rate related with λ=0. But for quality of reconstructed right view, even though higher λ results lower quality, the difference is small. Considering the 8×8 block based disparity map should be refined at the user side, high compression efficiency is more important compared with quality of reconstructed view. Therefore, λ=1 is selected in the proposed framework.

4 Refinement of 8×8 Block Based Disparity Map

At the user side, pixel-wise disparity map is necessary for view generation. Based on the block based disparity map received from the server side, fast disparity refinement scheme can be designed. Similar to normal 2D image, disparity map also has strong spatial correlation. If disparity estimation is accurate, objects in a scene with same depth will have same disparity. To reduce complexity of disparity estimation, some blocks with accurate disparities are extracted directly from disparity map by using this smoothness assumption so that the re-estimation of these blocks can be avoided.

The disparity refinement algorithm is carried out according to the following steps:

Step1. In 8×8 block based disparity map, let dv be disparity of current block, and dv_i ($i \in [0, 7]$) be disparity of one of the eight neighbors of current block. If $f_{8\times8}(x, y)$ of current block (x, y) equals to 1 according to Eq(5), the current disparity dv is regarded as an accurate one, and it will be copied to each pixel within the corresponding block of pixel-wise disparity map. Otherwise, the current 8×8 block will be split into four smaller blocks with the size of 4×4, and each of them is marked as inaccurate sub-blocks.

$$f_{8\times8}(x, y) = \begin{cases} 1, & if\ \forall(|\,dv - dv_i\,|) < 2, i \in [0,7] \\ 0, & otherwise \end{cases} \tag{5}$$

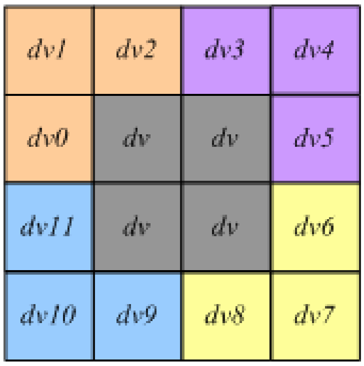
Fig.5(a) gives disparity map after 8×8 block based consistency test. The white areas indicate the inaccurate blocks, while the gray regions are copied directly from the 8×8 block based disparity map. As we can see, the accurate blocks are either background of a scene or at middle part of foreground objects.

Step2. In disparity map obtained by Step1, for those inaccurate 4×4 sub-blocks, the disparity consistency of 4×4 sub-blocks is done as follows. For the top left 4×4 sub-block as shown in Fig.5(b), if its disparity is consistent with disparities of its three neighbors, that is, *dv1*, *dv2* and *dv3*, the *dv* of the top left 4×4 sub-block is thought to be accurate so that it will be copied to the corresponding pixels within the pixel-wise disparity map. The procedure of consistent checking of the other three 4×4 sub-blocks in Fig.5(b) is similar with the top left sub-block. Then the disparities of inaccurate 4×4 sub-blocks will be searched again and compared with its four neighbors, similar as Eq.(5) except that the eight neighbors are replaced with four neighbors.

Fig.5(c) gives the disparity map after step2, in which the white pixels are those inaccurate blocks even after the 4×4 sub-block verification.

Step3. In the pixel-wise disparity map, the inaccurate 4×4 disparity is applied to accelerate the refinement of inaccurate regions. Let *dv* denote current disparity of an inaccurate pixel and it is used as initial position of searching process. A small range for searching the optimal disparity is defined as [*dv*-4, *dv*+4]. Therefore, for pixel that has initial block disparity, the computation complexity of disparity estimation is decreased greatly. Fig.5(d) shows the final refined pixel-wise disparity map.

(a) verified 8×8 disparity map



(b) neighborhood for 4×4 disparity consistency test

(c) verified 4×4 disparity map

(d) final pixel-wise disparity map

Fig.5: Disparity consistency test

5 View Generation

Intermediate view interpolation is vital for IBR(Image Based Rendering) based FVV system, since it determines the quality of synthesized virtual view. When the correspondences between view n and view n+1 are obtained, the dense intermediate disparity map at α position can be generated through the following steps.

(1) For the pixel at position x in the view n with disparity d, the position of the corresponding pixel in the intermediate view can be obtained by simply scaling the disparity according to α which indicates the location of the intermediate view between view n and view n+1. Disparity αd is assigned to the pixel at position $x+\alpha d$ in intermediate view. In this case, the values of the pixels between view n and view n+1 are interpolated in term of α. This is illustrated in Fig.6 by the pair of arrows which are opposite with each other.

(2) The intermediate disparity map obtained by the above step will have some holes. For the holes in corresponding intermediate view, the pixels within the holes are regarded as occluded pixels. These pixels can only be seen in one view, and a simple strategy is introduced to estimate their disparities. As we known, the disparities of background objects are normally smaller than that of the foreground objects, and in general,

background objects may be occluded by foreground objects. Therefore, for the occluded pixels, the nearest matched pixels at the left and right direction are compared. Let d_L and d_R denote the corresponding disparities with respect to the left and right directions, then the disparity d of current occluded pixel is determined by

$$d = \begin{cases} d_R\text{, and left occluded,} & \text{if } d_L \geq d_R \\ d_L\text{, and right occluded,} & \text{otherwise} \end{cases} \tag{6}$$

In this case, the pixels are interpolated according to occlusion type. If current pixel is labeled as left occluded pixel, that is, $d_L \geq d_R$, the pixel at the view n indicated by the disparity d_R is directly copied to the interpolating pixel. The similar operation can be done to the right occluded pixel. If the disparity map of intermediate view is obtained, the interpolation of intermediate view is easy to be done.

Ray-space representation describes the rays in a scene as a 4D function $f(x,y,\theta,\varphi)$, where (θ,φ) denotes the direction of the ray, (x,y) denotes the intersection of the ray and the reference plane, and $f(x,y,\theta,\varphi)$ represents the intensity of the specific ray. An important feature in ray-space is that an image with respect to a certain viewpoint is given as a sub-space of ray-space. Fig.7 shows the relationship between real space and ray-space. Intermediate views with respect to virtual cameras, denoted as the gray ones in the figure, are required to be rendered. The relationship between the virtual view and the ray-space can be explained as

$$X = x + z\tan\theta \tag{7}$$

Therefore, when dense sampling of a scene is acquired, it is comparatively easy to synthesize a novel view.

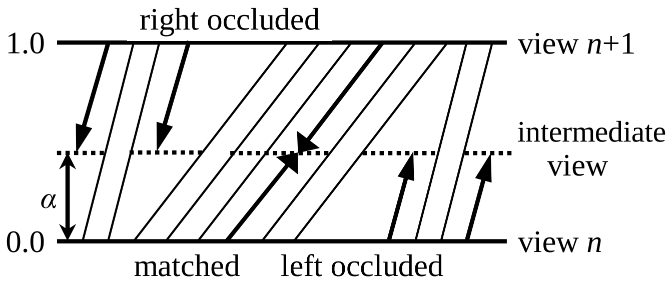
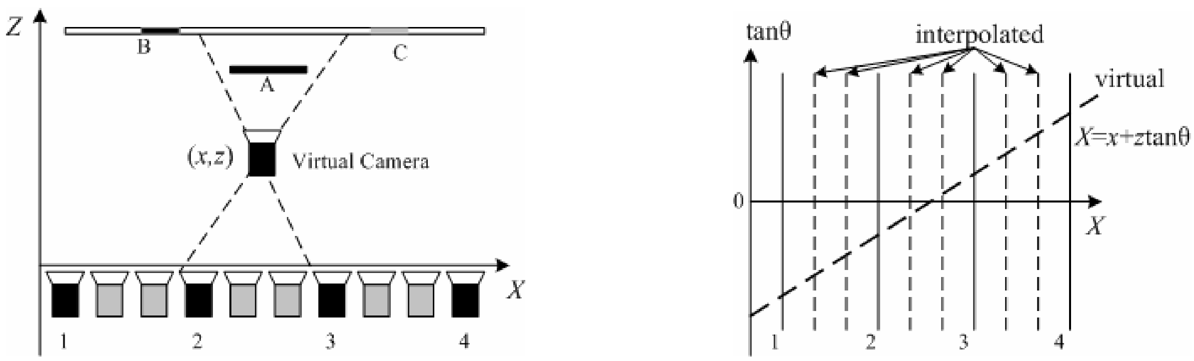


Fig.6: Generation of disparity map of intermediate view



(a) real space (b) ray-space

Fig.7: Relationship between real space and ray-space

5 Experimental Results and Analysis

5.1 Performance analysis of pixel-wise disparity map and 8×8 block based disparity map

We have compared the performance of pixel-wise disparity map and 8×8 block based disparity map in transmission cost and quality of reconstructed views. The pixel-wise disparity map is also generated by the disparity estimation method considering neighborhood constraints mentioned in section III except that the searched disparity of 7×7 current block is allocated to the center pixel of the block instead of all pixels of the 7×7 block. The pixel-wise disparity map is then encoded with JMVM 4.0[17]. AKKO&RAYO is used as the test sequence. Table 5 gives bit rates with respect to pixel-wise disparity map encoded by JMVM with different QP and 8×8 block based disparity map encoded by CABAC. Table 6 shows the corresponding PSNRs of reconstructed views. It is seen that the scheme of 8×8 block based disparity map reduced about 81.00%, 59.37% and 11.00% bit rate compared with the scheme of pixel-wise disparity map when *QP* is set to be 22, 27 and 32 respectively. When *QP* is 37, the bit rate of pixel-wise disparity map becomes lower than that of 8×8 block based disparity map. However, the quality of reconstructed views is not satisfied compared with the scheme of 8×8 block based disparity map. It should be noted that as the QP grows, the bit rate of encoded pixel-wise disparity map is reduced, but the quality of reconstructed views which reflects the accuracy of disparity map is also reduced due to the lossy encoding. By contrast, the bit rate of 8×8 block based disparity map is invariable by taking the advantage of lossless encoding. The refinement of 8×8 block based disparity map can further lift up the quality of reconstructed views about 1dB.

Table 5. Comparison of pixel-wise disparity map and 8×8 block based disparity map in bit rate

Pixel-wise disparity map coded by JMVM (kbps)			8×8 block based disparity map coded by CABAC (kbps)	reduced bit rate (%)
QP	22	175.85	33.40	-81.00
	27	82.21		-59.37
	32	37.53		-11.00
	37	16.67		+100.33

Table 6. Comparison of pixel-wise disparity map and 8×8 block based disparity map in PSNR of reconstructed views unit: dB

Views			view1	view2	view3	view4
Pixel-wise disparity map (JMVM)	QP	22	30.37	31.36	30.59	29.37
		27	29.18	30.18	29.55	28.34
		32	27.94	28.85	28.42	27.07
		37	26.94	28.11	27.18	26.01
8×8 block based disparity map (CABAC)			31.17	32.37	31.93	30.08
Refined 8×8 block based disparity map (CABAC)			32.27	33.56	32.85	30.96

5.2 Performance of the proposed method in view generation

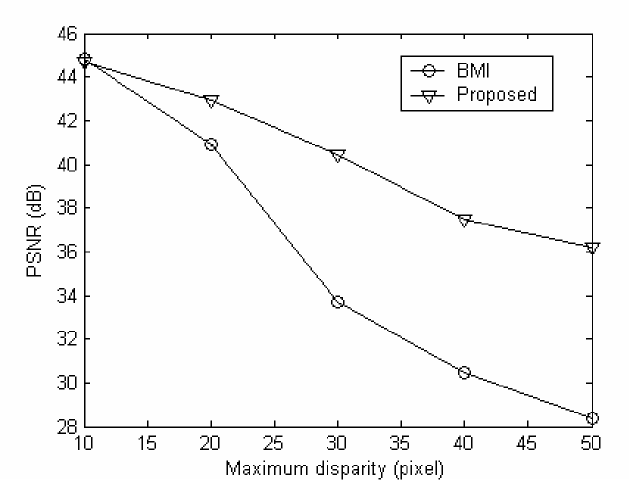
Experiments are also performed to evaluate the efficiency of the proposed method in view generation. The test image Xmas with 101 parallel views is used to evaluate intermediate view interpolation without camera parameter. The 3DTV test data Akko&Kayo is used to render arbitrary view image. Akko&Kayo is captured by 3×5 camera array, here, 5 images in the middle row are selected, the maximum disparity between two anchor images is 20 pixels.

Fig.8 shows *PSNR* and running time of interpolating intermediate view with different camera intervals. In the figures, the proposed method outperforms BMI(Block-based Matching Interpolation) method[18], because it can obtain more accurate disparities. Moreover, the proposed method runs much faster than BMI because it has initial disparity for searching.

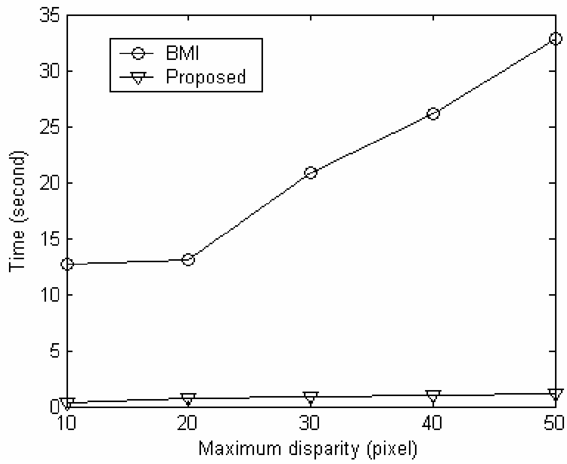
For rendering virtual viewpoint image, as illustrated in Fig.9, dense intermediate viewpoint images are firstly interpolated by the proposed method. Here, 19 intermediate viewpoint images are interpolated between two anchor view images according to the sampling theory. Then, these dense multi-view images are projected into ray-space. When given a virtual viewpoint position, the corresponding data can be extracted from the interpolated dense ray-space. Fig.9(a) shows a virtual view at the baseline, while Fig.9(b) and Fig.9(c) show the virtual views at forward and backward positions from the baseline respectively. As we move forward, details of a scene can be seen, meanwhile, the field of view becomes smaller. On the contrary, when we move backward, the field of view becomes larger. The emergence of black regions at top and bottom in Fig.9(c) is resulted from the fact that dense intermediate views are interpolated in horizontal direction but not vertical direction. From the rendered virtual view images, it is seen that the synthesized images have photo-reality without ghosting artifacts.

6 Conclusions

In this paper, a framework of view generation oriented free viewpoint video system is presented, which focus on the requirement and capability of user side. The calculation of disparity map, which is required at the user side, is moved to the server side, based on the fact that the user side is a terminal with constrained resource environment and block based disparity map can be easily obtained at the server side due to the usage of disparity compensation prediction at the MVC encoder. This change will alleviate the burden of view synthesis and it is more suitable for real time applications.



(a) *PSNR* of interpolated intermediate view



(b) run time of interpolating intermediate view

(c) original intermediate image

(d) BMI (*PSNR*=28.36dB,*TIME*=32.8s)

(e) Proposed method (*PSNR*=36.17dB,*TIME*=1.1s)

Fig.8: Result of intermediate view generation of Xmas (maximum disparity=50)

(a) virtual view at baseline

(b) virtual view forward 0.5m

(c) virtual view backward 0.5m

Fig.9: Results of arbitrary view rendering

Acknowledgment This work was supported by the Natural Science Foundation of China (grant 60472100, 60672073), the Program for New Century Excellent Talents in University (NCET-06-0537), Natural Science Foundation of Ningbo (grant 2007A610037).
Gangyi Jiang is correspondence author.

References:

[1] Ishfaq Ahmad, Multiview Video: Get Ready for Next-Generation Television, *IEEE Distributed Systems Online*, vol.8, no.3, 2007, art. no. 0703-o3006.

[2] A. Smolic, D. McCutchen, 3DAV Exploration of Video-Based Rendering Technology in MPEG, *IEEE Transactions on Circuits and Systems for Video Technology*, vol.14, no.3, 2004, pp.348-356.

[3] H. Kimata, M. Kitahara, K. Kamikura, et al., Low-Delay Multiview Video Coding for Free-Viewpoint Video Communication, *Systems and Computers in Japan*, vol.38, no.5, 2007, pp.15-29.

[4] A. Smolic, K. Mueller, P. Merkle, et al., 3D Video and Free Viewpoint Video - Technologies, Applications and MPEG Standards, *IEEE Int. Conf on Multimedia and Expo*, 2006, pp.2161-2164.
[5] M. Tanimoto, T. Fujii, H. Kimata, S. Sakazawa, Proposal on Requirements for FTV, *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-W127, 23rd Meeting*: San Jose, California, Apr. 2007.
[6] A. Smolic, and P. Kauff, Interactive 3D Video Representation and Coding Technologies, *Proc. of the IEEE, Special Issue on Advances in Video Coding and Delivery*, vol.93, no.1, Jan. 2005.
[7] Aljoscha Smolic, Karsten Mueller, Philipp Merkle, Nicole Atzpadin, Christoph Fehn, Markus Mueller, Oliver Schreer, Ralf Tanger, Peter Kauff, Thomas Wiegand, Multi-view video plus depth (MVD) format for advanced 3D video systems, *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT-W100*, San Jose, USA, 21-27 April, 2007.
[8] Philipp Merkle, Aljoscha Smolic, Karsten Mueller, Thomas Wiegand, MVC: Experiments on Coding of Multi-view Video plus Depth, *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6*, *JVT-X064*, Geneva, CH, 29 June - 5 July, 2007.
[9] S.C. Chan, H.-Y. Shum, K.-T. Ng, Image-Based Rendering and Synthesis, *IEEE Signal Processing Magazine,* vol.24, no.6, 2007, pp.22-33.
[10] C. Zhang T. Chen, A Survey on Image-Based Rendering-Representation, Sampling and Compression, *EURASIP Signal Processing: Image Commun.*, vol.19, no.1, 2004, pp.1-28.
[11] E. Kurutepe, M. R. Civanlar, A. M. Tekalp, Interactive Transport of Multi-view Videos for 3DTV Applications, *Journal of Zhejiang University Science*, vol.7, no.5, 2006, pp.830-836.
[12] M. Levoy, P. Hanrahan, Light Field Rendering, Proc. of Siggraph, 1996, pp. 31-42.
[13] X. Tong, R. M. Gray, Interactive Rendering from Compressed Light Fields, *IEEE Trans. Circuits Syst. Video Techn*., vol.13, no.11, 2003, pp.1080-1091.
[14] N. Fukushima, T. Yendo and T. Fujii, Real-Time Arbitrary View Interpolation and Rendering System Using Ray-Space, *Proceedings of SPIE Three-Dimensional TV, Video and Display*, 2005, pp.250-261.
[15] L. Zhong, M. Yu, Y. Zhou, G. Jiang, Quality and Performance Evaluation of Ray-Space Interpolation for Free Viewpoint Video Systems, *Lecture Notes in Computer Science, LNCS3992*, 2006, pp.367-370.
[16] S. J. Gortler, R. Grzeszczuk, R. Szeliski, M. F. Cohen.: The Lumigraph, *Proc. of SIGGRAPH-96*, 1996, pp.31-42.
[17] JMVM 4.0 Software, *ISO/IEC JTC1/SC29/WG11, W208*: San Jose, California, USA, 21–27 April, 2007
[18] T. Yendo, T. Fujii, M. Tanimoto, Ray-Space Acquisition and Reconstruction within Cylindrical Objective Space, *Proceedings of SPIE Stereoscopic Displays and Virtual Reality System*, vol.6055, 2006, pp.299-306.