A Transmission and Interaction Oriented Free-Viewpoint Video System

You Yang, Mei Yu, Gangyi Jiang, and Zongju Peng

***Abstract*—** A transmission and interaction oriented free-viewpoint video system framework is presented in this paper. The proposed system framework is divided as distinctive but interconnected parts, from video encoder, adaptive-interactive disparity extractor, to virtual view rendering. Adaptive disparity algorithm and extractor is proposed and embedded to multiview video encoder, in order to fetch the desired coarse disparity map and enhance the encoder performance. The output bitstream for storage and transmission is organized as disparity layer and video layer. Virtual viewpoint images accessed in the procedure of client-system interaction will be interpolated via disparity based algorithm after the decoded disparity information is refined to single-pixel based disparity map. Experimental results show that, the bitstream volume of disparity map accounts for a little proportion when comparing with bitstream volume of video signals in transmission, while high quality of virtual viewpoint images can be provided to clients in real-time.

***Keywords*—**free-viewpoint video, disparity map, three dimensional television, view rendering.

I. Introduction

THERE dimensional television (3DTV) is considered as the most potential and valuable entertainment implementation for future consumers after the generation of digital television [1]. Multiview video system (MVS) is a promising approach to realize the 3DTV project, for its ability in providing interactive and immersive video services [2,3]. Free-viewpoint video (FVV), one of the important and advanced implementations in MVS, creates 3-D immersive experiences via various methods by showing a scene interactively from slightly different angles to the left and right eyes of viewer can be employed to utilize. On the other hand, the features from jitter-free human-computer interaction and perfection in visual experiences of FVV make this kind of video system more practical in development [4,5].

FVV system is an important and special application for MVS, for FVV can provide synthesized images for client viewpoint selection. European ATTEST project built a 2D/3D compatible system for broadcast networks [6]. In this system, traditional 2D video-plus-depth format is adopted for data representation. MERL set up a MVS from video acquisition, transmission to projector display [7]. Natural scene is captured by parallel camera array from different angles, stored in MVS server, and then transmitted via networks after simple processing. The terminal devices fetch and decode the video bitstream, display all the viewpoint scenes by projectors array. Microsoft Research Asia developed a MVS over IP multicast network for group of users to share the multiview video programs [8]. Switching points were inserted in the video bitstream for the convenient of user interaction. In a summery, these MVS framework prototypes show the simple implementations for future entertainment. However, all of these systems are unable to provide virtual viewpoint images, i.e., users can not obtain jitter-free interactive video services.

Virtual viewpoint images feature out the FVV system from MVS, and then a suitable data representation method for synthesis algorithm is required. Approaches in creating virtual view interactively can be generally divided as image-based rendering (IBR) and model-based rendering methods (MBR) [9]. MBR mainly solves the problems in computer graphics, and therefore the geometric information and model functions are critical for view synthesis. However, it is difficult to obtain the needed data directly from natural scenes for MBR based view synthesis. Hence, IBR is more suitable for FVV applications. IBR is a totally different approach for view synthesis, where object or scene is captured from various positions. From the recorded images, new views can be generated for camera positions not coinciding with the recording position. Since virtual views are synthesized directly from recorded scenes, dense array of cameras are critical in creating interactive FVV [10]. Ray-space interpolation techniques were proposed and submitted to ISO/MPEG/JVT as suitable methods to realize a full real-time FVV without any restrictions on scene [11]. However, difficulties arose such as calibrated dense camera array should be arranged within centimeter, powerful hardware system is needed for supporting, and huge amount of video data is required in obtaining the ray-space. Problems in ray-space technique make it still within the long progress to a whole successful FVV. Light field is another IBR approach targeting the real-time interactive FVV [12].

3D warping is a popular approach for image synthesis by using the corresponding depth map [13]. However, the rendering quality of virtual view may deteriorate due to disocclusion and discontinuities in depth map, especially when the viewer moves away from the original camera angle.

This work was supported in part 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). Manuscript Received May 13, 2007; Revised Novem. 18, 2007

You Yang, Mei Yu, Gangyi Jiang, and Zongju Peng are with the Faculty of Information Science and Engineering, Ningbo University, Ningbo, 315211, China. (corresponding author: Mei Yu, e-mail: yumei@nbu.edu.cn).

N-view-plus-M-depth method has been proposed as an extension to address the confinements of the video-plus-depth representation [14, 15]. Rendering techniques in system terminal are studied from this data representation. Holes due to depth discontinuities are able to be solved using pixels from neighbor view images. However, imperfections in depth maps may result in ghost-like shadows near the object boundaries, which can be removed by more sophisticated algorithms [17].

Data compression of depth map is then aroused as a new problem in multiview video coding (MVC) for FVV. Depth maps are captured originally via depth or infrared cameras simultaneously with ordinary camera array capturing scenes. Continuous depth data is very important in 3-D warping algorithms for high quality virtual image interpolation. Hence, these continuous data must be lossless compressed with recorded viewpoint signals in MVC encoder. Although the compression may be effected by a process of quantization and run-length encoding, it is reported that the data volume of loss encoded depth map account for 10%~20% of video bitstream [17]. This situation will result in unsatisfactory for some applications. For instance, with an image frame of 352x176 pixels having the depth data specified in 24 bits per pixel, approximately 250Kbytes of memory would be required for depth data storage. However, for real-time transmission of the image data, this volume of accompanying depth data needs typically to be reduced to 2Kbytes [18].

Disparity information is an alternation to depth map in real time FVV applications. Disparity map can be obtained via disparity estimation algorithm rather than depth camera, and it can be transformed easily to depth data for view synthesis. On the other hand, disparity map can be compressed with the similar method in H.264 encoder to compress the motion vector (MV). Therefore, N-view-plus-N-1-disparity MVC scheme is proposed here for our FVV implementations. However, disparity also shares the similar problem with depth in the procedure of image interpolation, i.e., perceptual and subjective quality of synthesized image will be affected by the quality of disparity map.

In this paper, a schematic framework for transmission and interaction oriented interactive FVV system is presented. Joint multiview video model (JMVM) is modified and employed for video signal compression. Coarse disparity map is fetched and lossless compressed with viewpoint signals in the encoder. The bitstream volume of disparity is proportional small and suitable for network transmission. Bitstreams for video and disparity are transmitted to user terminals via separated channels. Refined disparity map is obtained in the client decoder for virtual viewpoint image synthesis, and the interpolation procedure can be implemented in real-time.

II. Schematic Description for the Proposed FVV System

Fig.1 shows the schematic framework of the proposed system. As can be found in this framework, the raw multiview video signals are input to the JMVM MVC encoder to exploit the spatial and temporal redundancies in multiview video signals firstly. The algorithm for global disparity estimation in JMVM6.0 platform is replaced by the proposed coarse disparity estimation algorithm. MV obtained in JMVM6.0 and disparity vector from the proposed algorithm are both employed for consequent disparity estimation. Both vertical and horizontal disparity map are fetched and compressed separately for high quality image synthesis in client part. Bitstreams from N viewpoint video signals and N-1 disparity maps are stored in multiview video server. Accordingly, these bitstreams are transmitted to client terminals simultaneously via two different communication channels. The client device will receive and decode multiview video bitstreams first, and then show the recorded view scenes to client in real-time. Then, the coarse disparity bitstream is decoded and refined to single pixel-based disparity information in background processor. Virtual view scenes are synthesized via the refined disparity map and neighbor decoded scenes to ensure the high quality and jitter-free interaction between client and system.

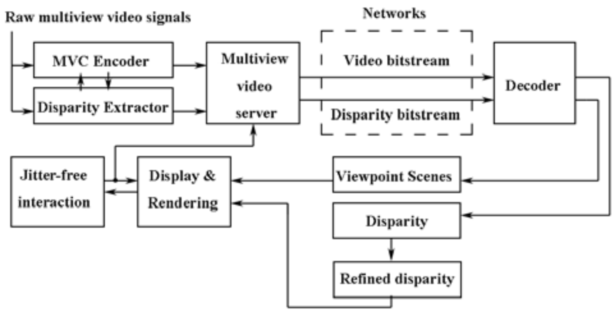


Fig.1. Schematic framework for the proposed FVV system.

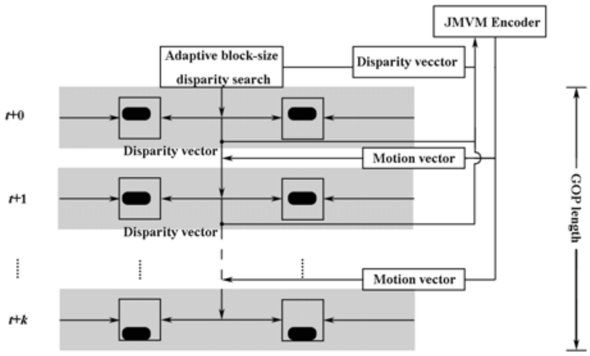


Fig.2. Interactive motion-disparity united disparity extractor scheme.

A. MVC Encoder and Disparity Extractor

Raw multiview video signals are captured via augmented camera array to the same scene. However, the huge amount of video signals becomes the great difficulty for storage and transmission in FVV system. Hence, an appropriate MVC scheme must be employed for signal compression. JMVM is the state-of-the-art encoder and adopted in the proposed system to exploit the spatial and temporal redundancies among and within viewpoints.

Disparity information is fetched interactively in the procedure of encoding. As depicted by Fig.2, disparity map is obtained through adaptive block-size disparity estimation algorithm at the very beginning of a group of picture (GOP). TABLE 1 gives the pseudo-code for our proposed recursive adaptie block-size dual-direction justification disparity estimation algorithm. Mean squared difference (MSD) metric is applied for block based comparison. The block size for comparison is adaptively selected by algorithm from 16x16 to 4x4, according to the texture characteristics of the frame. Lager block size will be used for planar areas, while smaller one is for boundary and texture regions. The obtained disparity information in t+0 frame pair will be applied to two different directions. First of all, disparity vector in t+0 can be applied to reduce the searching range for motion estimation between t+0 and t+1 frames for video signal compression. Originally, global disparity in JMVM6.0is fetched via separated algorithm at t+0 for different GOPs, and linear interpolated for consequent temporal frames to reduce the searching range of motion estimation. However, the global disparity is an overall parameter for a frame, but is not a precise one for each of the candidate blocks in estimation and compression. Therefore, the search efficiency for motion estimation will be further improved if precise disparity vector for each individual candidate block is obtained. The global disparity in JMVM6.0 is replaced by the fetched disparity from this proposed disparity extractor, in order to providing a relative precise searching range for each candidate block in motion estimation. The obtained disparity map at t+0 is also delivered to t+1 motion-disparity compound searching in the proposed extractor, which is the second direction. In Fig.3, the same object will give similar motion curve in different cameras. Hence, it will have same motion vector, and is calculated as x_2-x_0=x_3-x_1 and y_2-y_0=y_3-y_1. Supposing that the neighbor adjacent cameras are settled with parallel arrangement, the disparity in neighbor viewpoints of different time stamp will be equivalent. Therefore, the computation procedure of disparity vector will be suitable be expressed by x_1-x_0=x_3-x_2 and y_1-y_0=y_3- y_2. For ideal parallel camera arrangement, there will have horizontal parallax rather than vertical one. Hence, it will have y_1=y_0 and y_3=y_2. However, it is impossible to build a complete parallel camera system for scene capturing, which implies $y_1 \neq y_0$ and $y_3 \neq y_2$. Furthermore, we can not expect the disparity will satisfies the equation x_2-x_0 = x_3-x_1 exactly due to physical noises in cameras. Hence, the proposed disparity extractor for t+i ($0<i<$GOP_length) frame will use both the equations and small variation range. Furthermore, parallax estimation algorithm is implemented horizontally and vertically in the proposed system, preventing any losses in important information. Examples of obtained consequent horizontal disparity maps are given in Fig.4.

B. *Bitstream Structure*

The bitstreams stored in multiview video server is obtained in JMVM encoder and disparity extractor. As discussed above, the bitstreams are divided into separated two layers. As shown in Fig.5, the base layer is named MVC layer, contains the video signal compressed by modified JMVM6.0 encoder. The bitstream structure of this layer is formatted according to JMVM standard.

Table 1. Pseudo-code for recursive adaptive block-size dual-direction justification disparity estimation algorithm.

line	operations
10	if (current_block_size < minimum_block_size)
20	then goto 90
30	else SetSearchRegion() & SetBlockSize()
40	DualDirectionDisparitySearch()
50	if (DisparityMatched())
60	then goto 80
70	else QuadriSplitSearchBlock() & goto 10
80	if (SearchOutAllRegion())
90	then Finish()
100	else goto 40

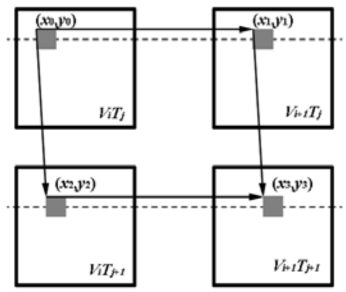
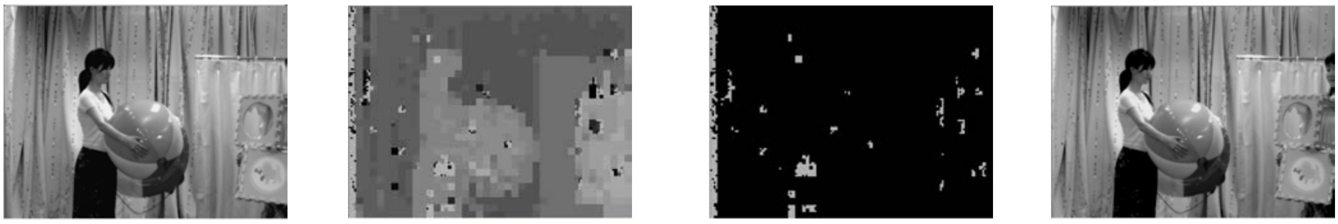
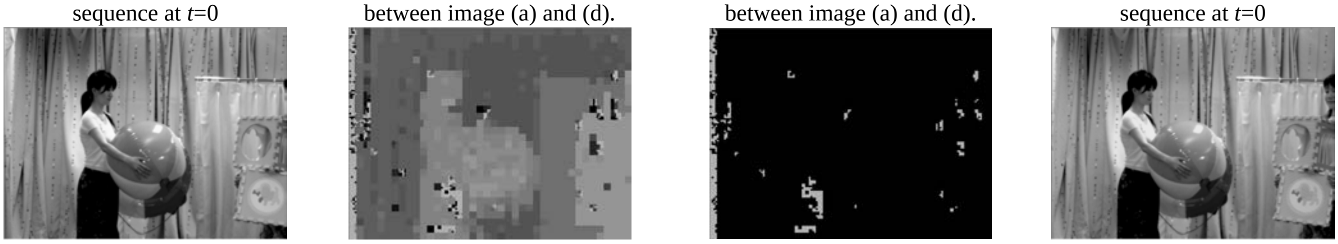


Fig.3. Motion-disparity compound disparity estimation method adopted in recursive adaptive block-size dual-direction justification disparity estimation algorithm.

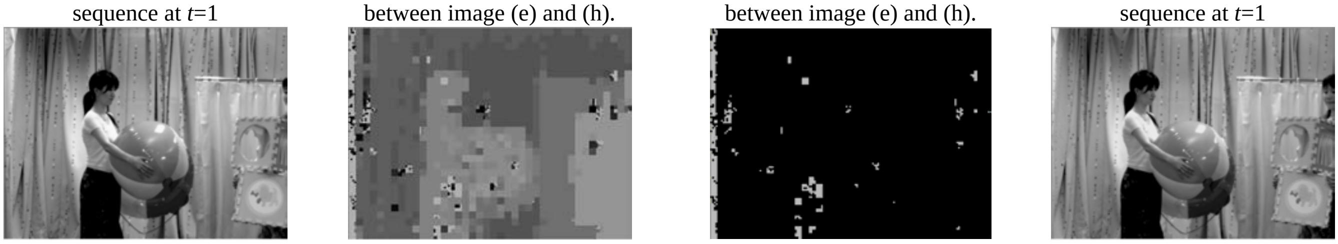
The enhancement layer is the disparity bitstream layer, which is the lossless compressed disparity map sequence by entropy algorithm. As shown in Fig.4, disparity map sequence is obtained through two neighbor viewpoint video sequences by disparity extractor. The block size in disparity map is adaptively selected according to texture or planar features in frame in the procedure of MSD based comparison. The data size of disparity map sequence is equivalent to one viewpoint video sequence before this sequence be input to entropy algorithm, while should be compressed without any loses due to its importance in virtual view synthesize at the part of client devices. Therefore, CABAC compression technology is employed to deal with the disparity map sequence. Furthermore, block merging and block filtering techniques are both applied before disparity signal compression. When two neighbor blocks in disparity map are with the same size and equivalent disparity value, they can be merged as a larger block to save the data size in disparity representation. Two 8x8 blocks can be merged as 16x8 or 8x16, while two 4x4 blocks will be integrated as 8x4 or 4x8. However, two 16x8 or 8x16 blocks will not be merged as 16x16, since according to the disparity estimation algorithm given Fig.3, larger block will be marked out rather than be split to smaller ones that containing exactly the same information with larger block. On the other hand, occlusive and exposed regions in one viewpoint are unable to find corresponding regions in the other viewpoint. For the convenient of data representation, block filtering technique is applied to fill out this block data. The occlusive block will not have disparity data inside, and the data will be obtained via disparity data of its neighbor upper and left blocks with $D = (D_u+D_l)/2$, where D_u and D_l is disparity value for upper and left block, respectively. After being processed with these two techniques, the disparity map sequence is lossless compressed with CABAC algorithm.



(a) Akko viewpoint 26 sequence at t=0 (b) Horizontal disparity map of t=0 between image (a) and (d). (c) Vertical disparity map of t=0 between image (a) and (d). (d) Akko viewpoint 27 sequence at t=0



(e) Akko viewpoint 26 sequence at t=1 (f) Horizontal disparity map of t=1 between image (e) and (h). (g) Vertical disparity map of t=1 between image (e) and (h). (h) Akko viewpoint 27 sequence at t=1



(i) Akko viewpoint 26 sequence at t=2 (j) Horizontal disparity map of t=2 between image (g) and (i). (k) Vertical disparity map of t=2 between image (i) and (l). (l) Akko viewpoint 27 sequence at t=2

Fig.4. Examples of resultant horizontal disparity map obtained by the proposed disparity estimation algorithm.

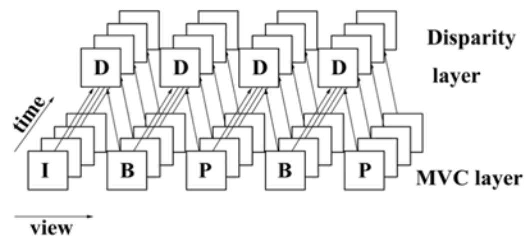


Fig.5. Bitstream structure of compressed videos and disparity maps.

C. Virtual View Rendering

Bitstream of disparity map sequences obtained in disparity extractor on the part of encoder are transmitted to decoder part accompany with video signal bitstreams. Two-layer bitstream structure shows the transmission mode for the proposed FVV system. Disparity maps will be restored to their original form since they were compressed by entropy algorithm without any loses. However, these disparity maps are unable to be used directly for virtual viewpoint image interpolation due to their image resolutions. According to the disparity estimation algorithm, the variation range of block size in disparity map is from 16x16 to 4x4. In other words, the obtained disparity map is in coarse resolution which is in need of refinement.

The procedure of disparity refinement is organized by two stages. In the first stage, the coarse disparity map decoded from the received bitstream is down-sampled and every 4x4 block is represented by a single pixel. This down-sampled image is then interpolated according to the following equation.

$$D_c=(w_A\times D_A+w_B\times D_B)/(w_A+w_B) \tag{1a}$$
$$w_A=1/((f_A-f_C)^2+1) \tag{1b}$$
$$w_B=1/((f_B-f_C)^2+1) \tag{1c}$$

where the parameters D_A and D_B are disparity values in the position A and B, which is depicted in Fig.10. w_A and w_B are weight factors for D_A and D_B, and these factors are obtained via the pixel values f_A, f_C and f_B in corresponding positions in the decoded frame. On the other hand, as shown in Fig.6, the interpolation can be implemented horizontally and vertically.

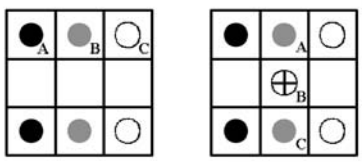


Fig.6. Disparity map interpolation with horizontal and vertical

directions

After being processed in the first stage, the disparity map is equivalent to a 2x2 block based disparity map. Then, in the second stage of processing, this 2x2 block based disparity map will be further refined to single pixel-based disparity map. The processing method is exactly the same in stage one. Fig.7 is showing the refined disparity map of Fig.4(b).

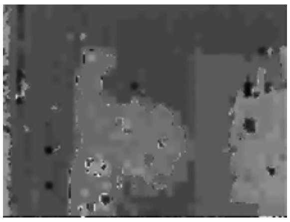


Fig.7. Refined disparity map in the client devices.

In the procedure of client-system interaction, virtual viewpoint image will be interpolated when terminal client accessing the viewpoint that is not captured originally by camera. Virtual viewpoint image can be rendered via disparity map and its neighbor decoded viewpoint images which are recorded by camera originally. According to the schematic system framework shown in Fig.1, the adopted view rendering technique in client part will depend on the obtained available information. Disparity based rendering algorithm will be applied when camera parameters are not available. On the other hand, disparity data can be transformed to depth data if camera parameters can be obtained. In this situation, 3D Warping algorithms can be applied for virtual viewpoint image synthesis. Therefore, the proposed system is compatible for both disparity and depth based image synthesis techniques for the convenient of client applications.

The first step for disparity based virtual viewpoint image synthesis is determining the location of virtual viewpoint between two recorded adjacent neighbor viewpoints. Let α be a scalar and denoting the proportional location between adjacent left and right viewpoint, and clearly, α satisfies $0\leq\alpha\leq1$.The pixel value for position x and y in virtual viewpoint image can be determine via following equation.

$$I_\alpha(x,y) = I_L(x+\lfloor ad \rfloor, y) = I_R(x+\lfloor (1-a)d \rfloor, y) \quad (2)$$

where d is the disparity value between left and right viewpoint images, I_L, I_R and I_α is pixel value for corresponding image on position (x,y). It can be implied from the Eq.(2) that the image interpolation procedure is easy in calculation, and furthermore, the client can obtain the synthesized image in real-time. Therefore, disparity based FVV system is a simple framework for real-time client-system jitter-free interaction.

III. Experiments and Results

In this section of experiment, we first compare the bitstream size of disparity map sequences with multiview video signals. This experiment will figure out the proportional transmission accumulation besides the video bitstream. Next, virtual view images will be interpolated in the procedure of client-system interaction.

A. Bitstream Experiments

AKKO and Breakdancer multiview video sequences are selected in our interactive FVV system experiments. AKKO test sequences are captured by 3x5 parallel camera arrays, while Breakdancer is also recorded by parallel arrangement [19]. Both of them are suitable for disparity extractor. JMVM6.0 MVC encoder from MERL is modified and used in the proposed FVV system for multiview video signal compression. The parts of global disparity estimation and interpolation in JMVM encoder are replaced by the proposed disparity extractor, and all disparity data needed for each of blocks in JMVM are obtained by this extractor. On the other hand, MV for each block obtained in JMVM encoder is delivered to disparity extractor for disparity-motion vector united disparity estimation algorithm. Disparity maps are obtained from both horizontal and vertical directions, although horizontal variation between two neighbor views is very small. In other words, there are two disparity maps fetched between two neighbor viewpoints simultaneously.

The compression results are shown in Table 2 and Table 3. As depicted by the results, the bitstream size of disparity map sequence accounts for a small proportion of the bitstream for two views. For example, the sum size of bitstream for views 26 and 27 is 18109648 bits, while their disparity map is compressed to 781626 bits, which accounts for 4.3161% of these two viewpoints. In other words, the accumulation on video signal bitstream transmission will be slightly increased due to the bitstream from disparity maps. However, the accumulation on bitstream is inevitable for 3D display in client device, and this 2-views-1-disparity data representation format is the minimum requirement for basic display needs. Moreover, the thorough accumulation is also small if all video bitstream and disparity map are needed for transmission. As shown by Table 2, the total bitstream size of disparity sequence accounts for 7.109% of the overall bitstream for video signals.

B. Interaction and Interpolation

Virtual viewpoint images will be accessed by client in most of the time in the procedure of human-computer interaction, since the number of cameras for scene capturing is limited. Moreover, the perceptual quality of these synthesized images will greatly affect the usability of FVV system.

In this part of experiment, virtual viewpoint images can be interpolated in real-time, since refined disparity maps are obtained in background processor, which will not affect the efficiency foreground displayer. Fig.8 provides 9 consequent interpolated virtual views between AKKO view 29 and 30 at t=200, with scalar α=0.1~0.9. Fig. 9 also shows 9 consequent synthesized virtual viewpoint images between Breakdancer view 0 and 1 at t=0 with α=0.1~0.9. As can be found in these images, the perceptual quality of images is appreciated for clients.

IV. CONCLUSION

A transmission and interaction oriented free-viewpoint video system framework is presented and analyzed in this work. The auto-adaptive dual-justify disparity extractor is proposed to fetch the disparity information between adjacent neighbor viewpoints. Disparity data is mainly applied for virtual viewpoint image synthesis in the procedure of client-system interaction and displaying. Furthermore, the disparity vector is also applied in JMVM encoder to improve the coding efficiency, especially to reduce computer consumptions in motion estimation and compensation. On the other hand, disparity maps are lossless compressed with entropy algorithm, in order to guarantee the perceptual virtual viewpoint image quality for clients.

Table 1. Bitstream experiments for AKKO multiview video sequence

Viewpoint	JMVM bits. data size	Sum of two viewpoint data size	Bits. size of disp. map	Proportion
26	11257768	18109648	781626	0.043161
27	6851880	14848320	746396	0.050268
28	7996440	15758568	756522	0.048007
29	7762128	17154800	790873	0.046102
30	9392672			
Total	43260888	----	3075417	0.071090

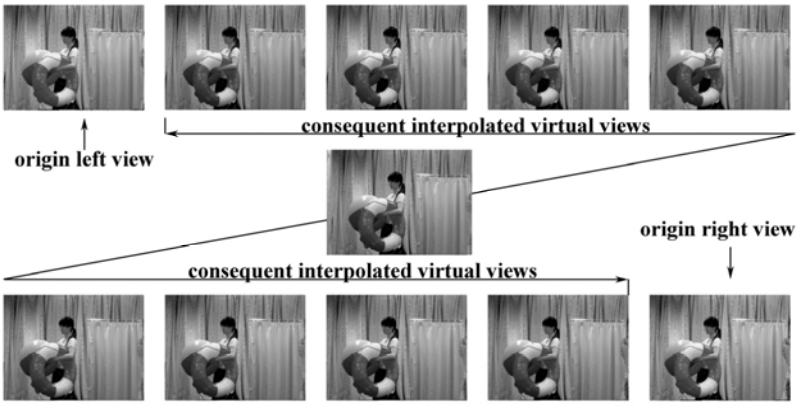


Fig.8. 9 consequent interpolated virtual views with scalar α=0.1~0.9 between the AKKO views 29 and 30 at t=200.

Table 3. Bitstream experiments for Breakdancer multiview video sequence

Viewpoint	JMVM bits. data size	Sum of two viewpoint data size	Bits. size of disp. map	Proportion
0	11233128	19274872	798392	0.041421
1	8041744	17999240	930392	0.051691
2	9957496	18020832	255480	0.014177
3	8063336	17162120	804625	0.046888
4	9098784	17438384	873863	0.050111
5	8339600	18088856	792456	0.043809
6	9749256	18817744	763913	0.040595
7	9068488			
Total	64483344	----	5219121	0.070958

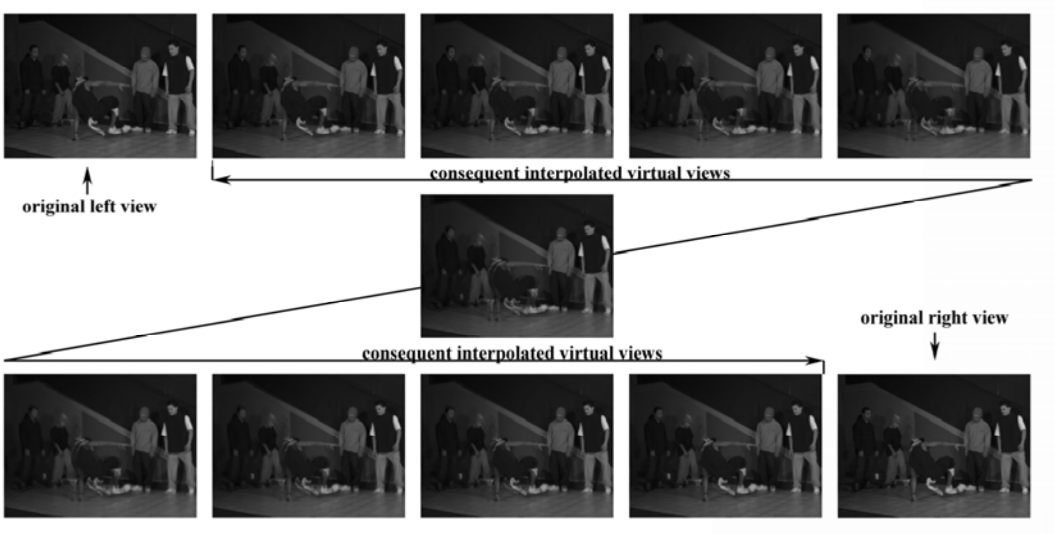


Fig.9. 9 consequent interpolated virtual views with scalar α=0.1~0.9 between the Breakdancer views 0 and 1 at t=0.

Experimental results show that, if the network bandwidth is able to carry video bitstream, it will have no problem for network to carry the two-layer structure bitstream to client. The volume of disparity bitstream accounts for a small proportion to the total size of two-layer structure bitstream. Furthermore, the system can provide high quality virtual viewpoint images for displaying in the procedure of client-system interaction in real-time. In a conclusion, the proposed system can provide high quality FVV services with low computer resources.

REFERENCES

[1] C. Fehn, Depth-Image-Based Rendering, Compression and Transmission for a New Approach on 3D-TV, Proc. of SPIE, vol. 5291, 2004, pp. 93-104.

[2] Introduction to Multiview Video Coding, ISO/IEC JTC 1/SC 29/WG11, N7328, July 2005.

[3] Requirements on Multi-view Video Coding v.8, ISO/IEC JTC1/SC29/WG11, N9163, Lausanne, Switzerland, July 2007.

[4] M. Tanimoto, Free Viewpoint Television – FTV, Picture Coding Symposium 2004, Dec. 2004.

[5] M. Tanimoto, T. Fujii, FTV – Free Viewpoint Television, ISO/IEC JTC1/SC29/WG11, M8595, Jul. 2002.

[6] A. Redert, M. O. de Beeck, C. Fehn, et al, ATTEST: Advanced Three-dimensional Television System Technologies. Proc. of 3DPVT, 2002, pp. 313-319.

[7] W. Matusik, H. Pfister, 3D TV: A Scalable System for Real-Time Acquisition, Transmission, and Autostereoscopic Display of Dynamic Scenes, ACM Transaction on Graphics, vol. 23, no. 3, 2004, pp: 811-821.

[8] J.-G. Lou, H. Cai, and J. Li, InteractiveMultiview Video Delivery Based on IP Multicast, Advances in Multimedia, Volume 2007, Article ID 97535, doi:10.1155/2007/97535

[9] A. Kubota, A. Smolic, et al., Multiview Imaging and 3DTV, IEEE Signal Processing Magazine, vol.24, no.6, 2007, pp.22-33.

[10] M. Tanimoto, Overview of Free Viewpoint Television, Signal Processing: Image Communication, vol.21, no.6, 2006, pp. 454-461

[11] P. Nabangchang, T. Fujii, et al., Experimental System of Free Viewpoint Television, Proc. of SPIE, vol.5006, 554-563, 2003.

[12] M. Levoy, P. Hanrahan, Light Field Rendering, Proc. of SIGGRAPH '96, 1996.

[13] L. McMillan, An Image-Based Approach to Three-Dimensional Computer Graphics, PhD. dissertation, University of North Carolina, 1997.

[14] A. Smolic, K. Mueller, P. Merkle, et al, 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, 23rd Meeting: San Jose, Apr. 2007.

[15] P. Merkle, A. Smolic. K. Mueller, T. Wiegand, MVC multi-view plus depth, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT-X064, 24rd Meeting: Geneva, Jul. 2007.

[16] E. Kurutepe, M. R. Civanlar, A. M. Tekalp, Client-Driven Selective Streaming of Multiview Video for Interactive 3DTV, IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no 11, 2007, pp.1558-1565.

[17] U. Fecker, A. Guenegues, I. Scholz and A. Kaup, Depth map compression for unstructured lumigraph rendering, Proceedings of SPIE, 2006, vol. 6077; part 1, pages 60770B

[18] http://www.freepatentsonline.com/EP0846308. html

[19] Description of Core Experiments in MVC, ISO/IEC JTC1/SC29/WG11, W8019, Montreux, Switzerland, April 2006.