

Collaborative View Synthesis for Interactive Multi-view Video Streaming

Fei Chen, Jiangchuan Liu, Yuan Zhao
School of Computing Science
Simon Fraser University
Burnaby, British Columbia
Email: {feic, jcliu, yza173}@sfu.ca

Edith Cheuk-Han Ngai
Department of Information Technology
Uppsala University
Uppsala, Sweden
Email: edith.ngai@it.uu.se

ABSTRACT

Interactive multi-view video enables users to enjoy the video from different viewpoints. Yet multi-view dramatically increases the video data volume and their computation, making realtime transmission and interactions a challenging task. It therefore calls for efficient view synthesis strategies that flexibly generate visual views. In this paper, we present a collaborative view synthesis strategy for online interactive multi-view video streaming based on Depth-Image Based Rendering (DIBR) view synthesis technology, which generates a visual view with the texture and depth information on both sides. Different from the traditional DIBR algorithm for single view synthesis, we explore the collaboration relationship between different viewpoints synthesis for a range of visual views generation, and propose Shift DIBR (S-DIBR). In S-DIBR, only the projected pixels, rather than all the pixels of the reference view, are utilized for next visual view generation. Therefore, the computation complexity of projection transform, which is the most computation intensive process in the traditional DIBR algorithm, is reduced to fulfill the requirement of online interactive streaming. Experiment results validate the efficiency of our collaborative view synthesis strategy, as well as the bandwidth scalability of the streaming system.

Categories and Subject Descriptors

H.5.1 [Multimedia Information Systems]: Video; C.2.1 [Network Architecture and Design]: Networking Communication

General Terms

Design, Performance, Experimentation

Keywords

multi-view streaming, depth-image based rendering, collaborative view synthesis

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

NOSSDAV'12, June 7–8, 2012, Toronto, Ontario, Canada.
Copyright 2012 ACM 978-1-4503-1430-5/12/06 ...\$10.00.

1. INTRODUCTION

With the rapid development of electronic and computing technology, multi-view video has recently attracted extensive interest due to greatly enhanced viewing experience. A variety of applications have emerged, such as, immersive teleconference, 3DTV and free viewpoint video. In those applications, the users can choose an arbitrary viewpoint to visualize sports or dynamic art actions. Furthermore, multi-camera systems bring users walk-through viewing experience through multiple local ray-space representation and 3-D model generation.

Unlike conventional single-view video systems, a multi-view video system allows the viewer to change viewpoint and to enjoy some special viewing experience such as View sweep and Frozen moment. It largely enhances the interactivity for users in entertainment orientated applications [1]. However, it also brings challenges to data delivery due to the huge data amounts to be transmitted. An interactive streaming system needs to make a good tradeoff between the transmission bandwidth and rendering quality [7]. Many researchers have been investigating this topic in the past few years [1, 5]. Several video formats have been developed for live or video on demand streaming, such as the multi-view video plus depth (MVD) format, in which the depth and texture information is provided to generate visual view for stereo or multi-view display [2]. The European project AT-TEST has developed auto-stereoscopic 3D display with the video streaming in MVD format for the left view and right view. Compared with the conventional 2DTV, less than 20% bandwidth is increased for auto-stereoscopic 3D display with the MVD format [6]. For multi-view streaming, the bandwidth can be further reduced through view synthesis with the MVD format, since the system can just transmit selected viewpoint videos rather than all. It also means view synthesis computation can be proceeded in the user's side, and the effectiveness to generate visual views greatly affects the rendering quality. Also, computation complexity and bandwidth adaptivity need to be considered for online interactive applications. As such an interactive multi-view streaming system has following demands.

1. *Rendering Quality.* A visual view is interpolated between transmitted views to reduce disparity of inter-view and smooth the view sweeping process;
2. *Efficiency.* The computation complexity should be reduced to guarantee the availability of interactive data;
3. *Bandwidth Scalability.* Given the different bandwidth

conditions, it should allow users to enjoy different view quality levels adaptive to their available bandwidth.

In this paper, we present a collaborative view synthesis strategy for online interactive multi-view video streaming system. To reduce the transmission bandwidth, selected views are transmitted to users as reference views, which are utilized to generate the visual views to smooth view sweeping application. Specifically, a Shift DIBR (S-DIBR) algorithm is proposed to guarantee the rendering quality and efficiency. According to the available bandwidth, the users can also enjoy improved viewing experience with a scalable layer. The remainder of the paper is organized as follows. We first review the related work in Section II. In Section III, we outline the main structure of the interactive multi-view streaming system. The collaborative view synthesis strategy is presented in Section IV. Then we will describe the traditional DIBR algorithm and the new S-DIBR algorithm which we proposed to reduce computation complexity for a range of visual views synthesis in Section V. The experimental results are presented in Section VI. Finally, we conclude our work in Section VII.

2. RELATED WORK

In the DIBR research, effective view synthesis remains an open problem given different conditions. Xue et al. [11] proposed a visual view rendering method based on depth image, which makes a tradeoff between the image quality and computation complexity. Unfortunately the pre-process depth map by expanding object edge results in depth distortion, especially in complicated background environment. Daribo and Saito [3] improved the rendering quality through a bilateral filter by taking into account the strength of the depth discontinuity. Meanwhile, the computation complexity is increased as a tradeoff. Schmeing and Jiang [10] designed a faithful approach specially for the disocclusion problem, which provides realistic texture rather than mixture of nearby pixels. However, it is not appropriate for online interactive application, since the disocclusion area in transformed background is corrected manually. In [4] an inpainting-based layered depth view generation method was developed to reduce the amount of residual data to complete missing pixels from the main layer. These previous works considered the single view synthesis only. We on the other hand consider an efficient view synthesis method with low computation complexity to generate a group of visual views for the interactive application of multi-view streaming systems. To our knowledge, the collaborative view synthesis for the streaming system has been largely unexplored.

3. SYSTEM DESCRIPTION

In Fig.1, we outline the streaming structure of a multi-view interactive application system. The system is mainly based on the DIBR algorithm, which can generate visual view with the depth and texture information of reference views on both sides. A *depth map* records the depth value of each point on the frame, which represents the distance from the viewer to the object. A *texture map* records the pixel value of each point accordingly, which provides the color information of the object. The points in the reference views are projected to a 3D space, and then projected to the coordinates of the target visual view. From Fig.1, we can see the cameras with different colors, and each one stands

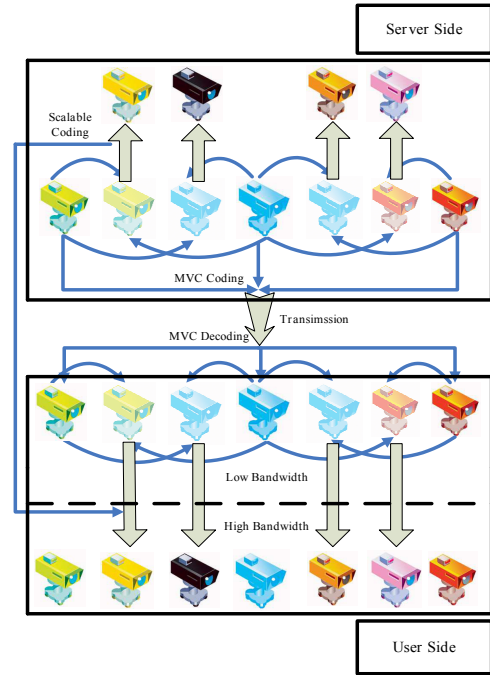


Figure 1: Multi view streaming structure

for a unique viewpoint. The opaque cameras are *reference views*, capturing depth and texture information, while the translucent cameras are *visual views* to be generated by the reference views. Actually, the streaming videos of these viewpoints are still captured in our system. They are used for scalable coding with synthesized views. A scalable layer is then transmitted to the users with high bandwidth.

The system includes both the server side and the user side. According to the bandwidth level, the users can be divided into a *high bandwidth group* and a *low bandwidth group*. After the server receives the videos captured by multi-camera, the reference view videos are chosen from these original videos. Then the streaming process provides the service with two layers, the *base layer* and the *scalable layer*. The base layer is prepared for all users and transmitted with multi-view coding (MVC) for further compression. When the users receive the base layer, view synthesis is proceeded with the reference views from MVC decoding. The scalable layer is then prepared for the high bandwidth group users to enjoy higher viewing quality. It is generated with visual views and original views in the server side, making use of scalable video coding (SVC). Our system utilizes both MVC and SVC for the efficiency and scalability. The two coding strategies can proceed in parallel, so the computation time does not delay the server or users for online streaming. From above description, we can see the view synthesis is utilized in both server side and user side. Therefore it is necessary to develop an efficient view synthesis strategy for the interactive streaming system.

4. VIEW SYNTHESIS COLLABORATION

Different from traditional view synthesis methods, which mainly focus on the accuracy of single visual view generation, we consider how to efficiently generate a group of visual views with different viewpoints for interactive appli-

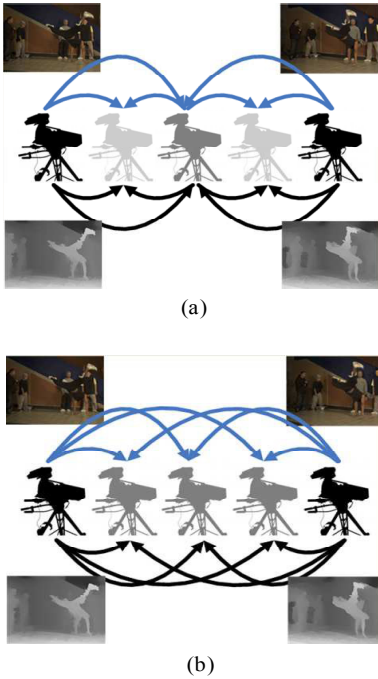


Figure 2: View Synthesis Model: (a) Middle Synthesis (b) Shift Synthesis

cations. Before analyzing the view synthesis algorithm, we consider two kinds of view synthesis collaboration strategies from Fig.2. The two reference views (opaque) provide texture and depth information on both sides for the generation of three visual views (translucent). The upper and lower lines show the texture and depth matching, respectively. In the middle synthesis pattern Fig.2 (a), the middle view among the visual views is generated in priority, and the remaining visual views are synthesized with the middle view and reference views. In the shift synthesis pattern Fig.2 (b), a range of visual views with different viewpoints are generated directly from the two reference views.

In our system, a shift synthesis is utilized to prepare for the texture maps of visual views for interactive applications. Comparing with the middle synthesis, there are three advantages in the aspects of efficiency, accuracy and buffer saving. (1) Given that a GPU supports the parallel computing, the shift synthesis pattern outperforms the middle synthesis pattern without waiting for the synthesized results. (2) In the middle synthesis pattern, the depth distortion during view synthesis is brought into the generation of a new visual view. The rendering quality deterioration accumulates as the number of visual view generation increases; (3) In the middle synthesis, the synthesized depth map of the visual view has to be reserved in both server side and user side for a new visual view generation. By contrast, in the shift synthesis pattern, the buffer storage space is reduced without reservation for the depth map of the visual view.

5. VIEW SYNTHESIS ALGORITHM

In this section, we first give an overview on the DIBR algorithm [6], which generates the visual view with the main reference view and the assistant reference view as shown in Fig.3. Then we will illustrate the S-DIBR to effectively

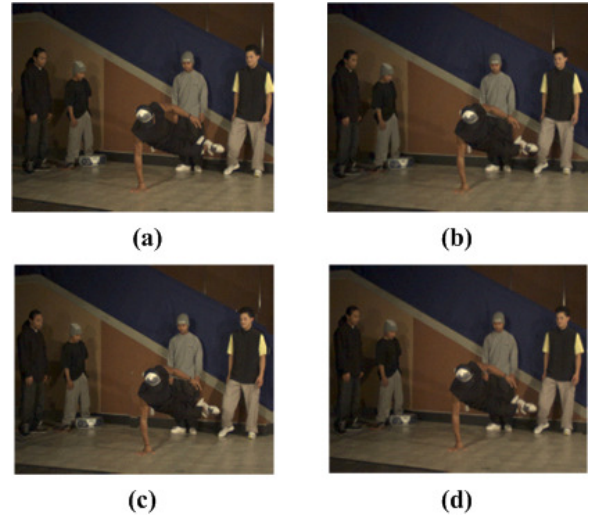


Figure 3: (a) Main reference view (b) Assistant reference view (c) Synthesized view (d) Original view

generate a group of visual views between the reference views with low computational cost.

5.1 DIBR Algorithm Overview

In DIBR, Each point with a pixel in the texture map relates to a grayscale recorded by depth map. The actual depth value can be calculated as:

$$Z = \frac{1}{\frac{D}{255}(\frac{1}{MinZ} - \frac{1}{MaxZ}) + \frac{1}{MaxZ}} \quad (1)$$

where D is the grayscale of the sample point and Z is the depth value, $MinZ$ and $MaxZ$ are the depth values of the nearest and farthest points in the real world, respectively. The following steps then proceed with depth map and texture map. In the projection transform process, the points in the reference views are projected to a 3-dimensional space and then projected to the target visual view. After that, a median filter is employed as edge filter to enclose the small holes and smooth disconnections of the depth map. The texture map of the visual view is blended in the projected view and rendered according to the depth value. The remaining disocclusions generated by depth warping are inpainted by the texture map from other reference views.

5.2 Proposed S-DIBR Algorithm

From [9] we know the view projection can be described by the equation:

$$\lambda_2 p_2 = K_2 R_2 [K_1 R_1]^{-1} \lambda_1 p_1 + K_2 (t_2 - t_1) \quad (2)$$

where K_n , and R_n , for $n \in [1, 2]$, are the intrinsic camera parameter and the rotation metric, respectively. The value λ_1 and λ_2 denote the depth value of the point in the target visual view and the reference view. The depth values are supposed to be the same with short horizontal shift, and the K_n and R_n are identical given that the cameras have the same setup parameters for the same scene. Equation (2) can be rewrote as follows:

$$\lambda p_2 = \lambda p_1 + K(t_2 - t_1)$$

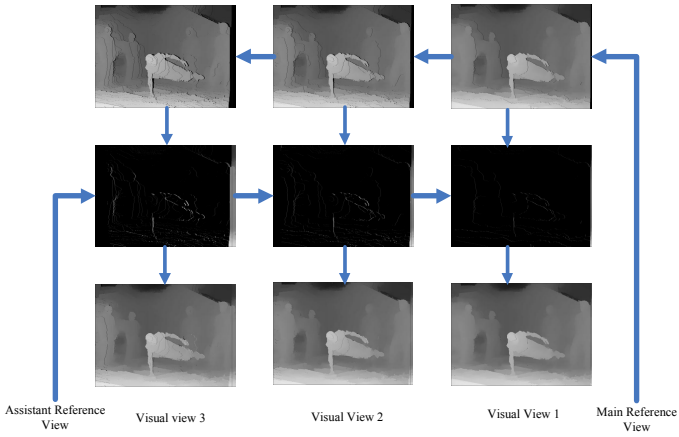


Figure 4: S-DIBR View Synthesis

which can be worked out as

$$\lambda \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} + \begin{pmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \Delta x \\ 0 \\ 0 \end{pmatrix}$$

and it simplifies to

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \begin{pmatrix} \frac{f}{\lambda} \Delta x \\ 0 \end{pmatrix} \quad (3)$$

where f is the camera's focal length and (x_0, y_0) is the offset coordinate of intrinsic parameters. Suppose that the distance between the two reference views is d , and the total number of synthesized views to be generated is N . According to Equation (3), the shift value S_0 from the main reference view to the visual view is:

$$S_0 = \frac{fdn}{\lambda N}, \text{ where } n = 1, 2, \dots, \frac{N}{2} \quad (4)$$

Accordingly, the shift value S_1 from the visual view to the assistant reference view is:

$$S_1 = \frac{fd}{\lambda N} \left(1 - \frac{n}{N}\right), \text{ where } n = 1, 2, \dots, \frac{N}{2} \quad (5)$$

From Equation (3-5), we know that the shift value S is proportional to baseline d and inversely proportional to the depth value λ . After the projecting process, some areas are disoccluded without depth or texture information. It also means the same number of pixels are occluded or projected out of boundary. Therefore the projection from other reference views is needed for inpainting disocclusion area. According to [8], the projection transform is the heaviest computation in the view synthesis process. In the traditional DIBR algorithm, each pixel from the reference view is projected to the target visual view for disocclusion area inpainting. It is necessary because the shift value of the points in the disocclusion area can not be computed without the depth information. Each pixel from the reference view has the potential to be projected in this area. However, in the view synthesis process for a group of different viewpoints, the projection transform is restricted to the hole filling area of the last transform after the first projection transform is completed. Given that $View(0)$ and $View(N+1)$ are the reference views, we take them as $R(0)$ and $R(N+1)$ for short. There are N views from $View(1)$ to $View(N)$ to be synthesized, and we denote them as $V(1), V(2), \dots, V(N)$ for

short. In our collaborative strategy, $R(0)$ is the main reference view and $R(N+1)$ is the assistant reference view for the view synthesis of $V(1)$ to $V(N/2)$. And they exchange the roles for the view synthesis from $V(N/2+1)$ to $V(N)$. The shift value S from $V(0)$ to $R(N/2)$ is $\frac{df}{2\lambda}$. And then S decreases every $\frac{df}{N\lambda}$ until it reaches $V(1)$. Fig.4 shows the collaborative view synthesis process between the main reference view and the assistant reference view in S-DIBR. According to Equation (3), the disocclusion problem in the texture map happens when disconnections occur in the depth map. As to a pair of neighboring pixels $Pixel_a$ and $Pixel_b$ with $\lambda_b - \lambda_a \neq 0$, we have the following equation

$$S_b - S_a = \frac{fd(\lambda_a - \lambda_b)}{\lambda_a \lambda_b} \quad (6)$$

Given that the cameras shift horizontally, λ_a and λ_b keep constant after the view shift. Therefore the disocclusion area is proportional to the baseline d . When users sweep from $R(0)$ to $R(N+1)$, there are N visual views to be generated between $R(0)$ and $R(N+1)$. We have the computation latency T as follows:

$$T = N(T_p + T_m + T_t + T_h) = NT_p + \delta \quad (7)$$

where T_p , T_m , T_t , and T_h are related to the time cost in pixel projection, median filtering, texture matching and hole filling process, respectively. In DIBR, all the pixels on both reference frames are required for pixel projection. Since T_p is proportional to the frame size P , the time cost in pixel projection is $T_p(D) = 2NKP$, where K is the time coefficient for pixel projection of a single frame.

In S-DIBR, only the projected pixels rather than all the pixels of the reference views are utilized for the next visual view generation. The disocclusion area reduces as the shift value between the main reference view and visual view decreases. Suppose P_i^M and P_i^A are the sets of projected pixels for $V(i)$ in warping process from the main reference view and the assistant reference view respectively, then we have the following relation:

$$\begin{cases} P_i \subset P_{i+1}, & P = P^A \\ P_{i+1} \subset P_i, & P = P^M \end{cases} \text{ where } i \in 1, 2, \dots, \frac{N}{2} \quad (8)$$

Let α_i be the possibility of the pixels projected from P_{i+1} to P_i , and from Equation (8) $P_i^M = \prod_{i=1}^{N/2} \alpha_i^M$, $P_i^A = \prod_{i=1}^{N/2-i+1} \alpha_i^A$. So we have $T_p(S)$ for S-DIBR as follows:

$$\begin{aligned} T_p(S) &= \sum_{j=1}^N T_{j,p} \\ &= 2K[(P_1^M + P_1^A) + (P_2^M + P_2^A) \dots (P_{N/2}^M + P_{N/2}^A)] \\ &= 2KP[(\alpha_1^M + \prod_{i=1}^{N/2} \alpha_i^A) \dots + (\prod_{i=1}^{N/2} \alpha_i^M + \alpha_1^A)] \\ &= 2KP \left(\sum_{j=1}^{N/2} \prod_{i=1}^j \alpha_i^M + \sum_{j=1}^{N/2} \prod_{i=1}^j \alpha_i^A \right) \end{aligned} \quad (9)$$

Where $T_{j,p}$ is the time cost in pixel projection process for $V(j)$. Since $T_p \gg T_m + T_t + T_h$ and $\delta^M \approx \delta^A$, we have the computation latency reduction ratio φ as follows:

$$\varphi = \frac{T(D) - T(S)}{T} = 1 - \frac{\sum_{j=1}^{N/2} (\prod_{i=1}^j \alpha_i^M + \prod_{i=1}^j \alpha_i^A)}{N} \quad (10)$$

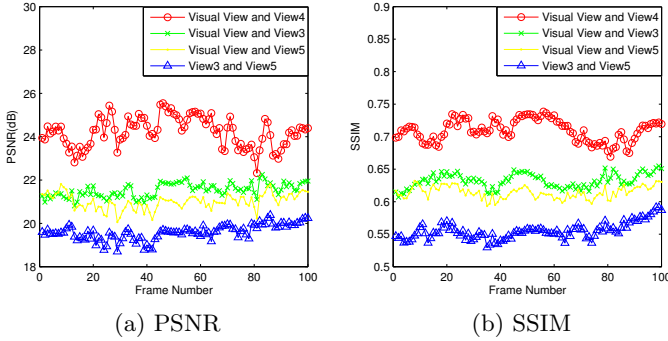


Figure 5: Different view comparison

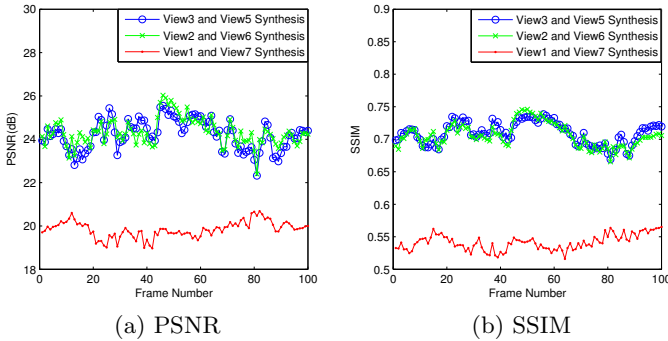


Figure 6: Synthesized with different reference view

6. EVALUATION

In the following experiments, our multi-view video data comes from the "Breakdancing" of Microsoft visual interactive group including 8 cameras setup in a horizontal line to capture the same scene from different viewpoints. The resolution for each frame is 1024×768 , and there are 100 frames to be rendered at $15fps$. The experiments are proceeded in the computer with *Pentium(R)D* CPU $3.4 GHz$ and $2.00 GB$ of RAM.

6.1 View Synthesis Analysis

Our system is developed to provide interactive applications for users to enjoy multi-view videos. When the users sweep the views, the disparity between the views would corrupt the rendering quality. From Fig.5. we can see the great disparity between the original views ($O(i)$ for short) $O(3)$ and $O(5)$. In order to sweep the view smoothly, the visual view ($V(i)$ for short) is generated by $O(3)$ and $O(5)$. From the PSNR and Structural Similarity (SSIM), the $V(4)$ is rendered close to $O(4)$. When the user sweeps the views from $O(3)$ to $O(5)$, it will first arrive to $V(4)$ as the transitional view and then reaches the destination view. Therefore it shows that the interpolation of the visual view is helpful to smooth the view sweeping. As the distance between the two reference views increases, the view synthesis quality decreases. We perform the view synthesis for $V(4)$ with different pairs of reference views, $R(3)$ and $R(5)$, $R(2)$ and $R(6)$, $R(1)$ and $R(7)$ respectively. From Fig.6, we can see that the quality of view synthesis with reference views $R(2)$ and $R(6)$

Table 1: Rendering Quality and Efficiency

Visual View	W-DIBR	S-DIBR
V(4)	26.5dB/0.73	25.9dB/0.71
V(4,5)	25.6dB/0.72	25.8dB/0.7
V(3,4,5)	22dB/0.63	24.7dB/0.68
V(2,3,4,5,6)	20.2dB/0.62	20dB/0.6

are generally the same to that with $R(3)$ and $R(5)$. The view synthesis quality degrades sharply when the shift distance is increased to 6 views (5 visual views included). There are two reasons for this. First, from Equation (4), the shift value is proportional to the depth value. The depth distortion accumulates gradually, and it leads to mismatch during the hole filling process. Second, in S-DIBR, it is supposed that the parallel cameras are arranged in horizontal line and the depth value is constant during the view shift process. Actually, the cameras usually make a circle with an appropriate radian to capture the same scene. (e.g. in "Break Dancer" the radian is about 30 degree between the camera on the left end and the one on the right end.) And the depth value changes as the view angle changes. If the radian change can not be ignored, the depth value will not keep constant. These two reasons prevent S-DIBR from the application of long distance shift, which we expect to address in the future.

6.2 Performance Comparisons

We next compare the collaborative view synthesis algorithm S-DIBR with conventional warping DIBR (W-DIBR for short) on rendering quality and computing efficiency. The $V(4)$ is taken as the median visual view. The reference views are the neighboring views (e.g. $R(2)$ and $R(6)$) are the reference views for visual view $V(3, 4, 5)$, and in S-DIBR $R(2)$ and $R(6)$ are the main reference views for $V(3)$ and $V(5)$, respectively.) We generate the visual view with W-DIBR and S-DIBR and compute the average PSNR and SSIM, respectively. As the shift value increases, more visual views are generated for view interpolation to reduce the disparity during view changes. From Tab.1, we can see S-DIBR have the similar PSNR and SSIM with W-DIBR when just one single visual view is synthesized. The rendering quality of both W-DIBR and S-DIBR is maintained when the number of visual views are increased to 2. The rendering quality of W-DIBR decreases sharply when 3 visual views need to be generated. Meanwhile, S-DIBR still keeps the performance since the main reference views provide the major texture. The S-DIBR rendering quality begins to deteriorate for 5 visual views generation, due to the accumulation of depth distortion and radian expansion.

In order to compare the efficiency of W-DIBR and S-DIBR, we measure the average synthesis time over 100 frames. From Fig.7, we compare the computation latency of view synthesis, which includes the pixel projection, rendering, and hole filling.¹ For one visual view synthesis, the W-DIBR and S-DIBR have the similar time cost. In 2 visual view generation, about 15% of synthesis time is saved in S-DIBR. The gap between W-DIBR and S-DIBR keeps expanding and finally reaches 30% in 6 visual view generation. Therefore, as the visual view number increases, the efficiency of collaborative view synthesis becomes more obvious.

¹Note that median filtering is not included, but it is compatible to our S-DIBR algorithm.

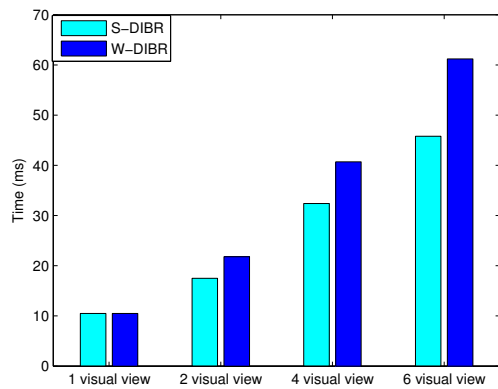


Figure 7: Computation Latency

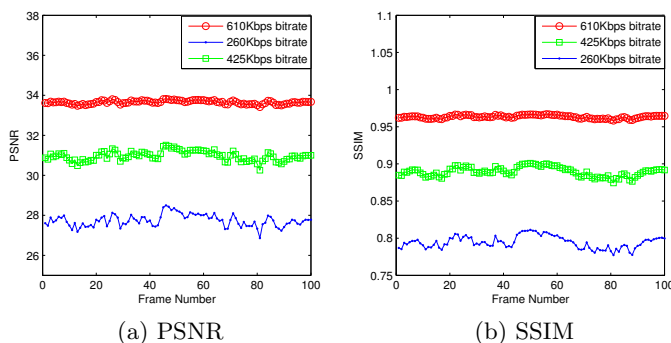


Figure 8: Different bit rates comparison

6.3 Scalable Rendering

The resolution of the frames in the view video is $256 * 192$ compressed from $1024 * 768$ and the frame rate of streaming is $15fps$. The original texture map is $144K$ and the corresponding depth map is $30K$ before compression. We now measure the PSNR and SSIM value for each frame of the video with the bitrate of $260Kbps$, $425Kbps$ and $610Kbps$. In Fig.8, the different bitrates respectively stand for 3 reference views streaming, 5 reference views streaming, and 5 reference views plus scalable layer streaming. We can see that the rendering quality in bitrate of $260Kbps$ is low, with the trading off of bandwidth saving. The rendering quality is improved greatly with bitrate of $425Kbps$, because more reference views are provided and the view shift distance is reduced by about 50%. As to the bitrate of $610Kbps$, the view synthesis strategy and scalable layer cooperate to yield high rendering quality. The rendering quality with high bandwidth is also stable, because the impact of the depth distortion in view synthesis is alleviated.

7. CONCLUSION

In this paper, we have developed a collaborative view synthesis strategy for multi-view streaming system. Making use of DIBR, it has effectively reduced the amount of data transmission in traditional multi-view streaming systems, and has realized the scalable rendering according to different bandwidth levels. The proposed S-DIBR algorithm

has reduced the computing complexity in pixel projection transform, which is the heaviest computation task in the traditional DIBR algorithm. The results from experiments have shown that our view synthesis for interpolation is helpful to smooth the view sweeping application. The evaluation between S-DIBR and W-DIBR has demonstrated significant performance improvement with respect to view synthesis quality and computation complexity. Furthermore the streaming quality in our streaming system has been analyzed with different bandwidth levels. However, the depth distortion and radian expansion problem still prevents S-DIBR from effectively serving long distance cameras, which will be addressed in our future work.

8. REFERENCES

- [1] A. Boukerche, R. Jarrar, and R. W. Pazzi. A novel interactive streaming protocol for image-based 3d virtual environment navigation. In *Proc. IEEE ICC'09*, pages 1 – 6, Dresden, June 2009.
- [2] G. Bravo, D. Luat, S. Zinger, and P. With. Real-time free-viewpoint DIBR on GPUs for 3DTV systems. In *Proc. IEEE International Conference on Consumer Electronics ICCE'11*, pages 1–4, Berlin, Germany, Sept. 2011.
- [3] I. Daribo and H. Saito. Bilateral depth-discontinuity filter for novel view synthesis. In *Proc. IEEE International Workshop on Multimedia Signal Processing MMSP'10*, pages 145 – 149, Oct. 2010.
- [4] I. Daribo and H. Saito. A novel inpainting-based layered depth video for 3DTV. *IEEE Trans. Broadcast.*, 20:533 – 541, June 2011.
- [5] Y. Ding and J. Liu. Efficient stereo segment scheduling in peer-to-peer 3d/multi-view video streaming. In *Proc. IEEE P2P'11*, Kyoto, Japan, Aug. 2011.
- [6] C. Fehn. Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3d-tv. In *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems XI*, pages 93–104, 2004.
- [7] E. Kurutepe and T. Sikora. Feasibility of multi-view video streaming over P2P networks. In *Proc. IEEE 3DTV Conference*, pages 157–160, Istanbul, May 2008.
- [8] J. Lai, C. Chen, and S. Chien. Architecture design and analysis of image-based rendering engine. In *Proc. IEEE International Conference on Multimedia and Expo ICME'11*, pages 1–6, Barcelona, Spain, July 2011.
- [9] D. Luat, S. Zinger, and P. Dewith. Conversion of free-viewpoint 3d multi-view video for stereoscopic displays. In *Proc. IEEE International Conference on Multimedia and Expo ICME'10*, pages 730–1734, Singapore, July 2010.
- [10] M. Schmeing and X. Jiang. Depth image based rendering: A faithful approach for the disocclusion problem. In *Proc. IEEE 3DTV-CON'10*, pages 1–4, June 2010.
- [11] J. Xue, M. Xi, D. Li, and M. Zhang. A new virtual view rendering method based on depth image. In *Proc. IEEE Asia-Pacific Conference on Wearable Computing Systems APWCS'10*, pages 147 – 150, Kaohsiung, Taiwan, May 2010.