INTERMEDIATE VIEW SYNTHESIS BASED ON EDGE DETECTING

Yangang Cai[1,2], *Ronggang Wang*[1], *Tongbing Cui*[2], *Hao Lv*[2], *Siwei Ma*[2]

xiaoc@pku.edu.cn, rgwang@szpku.edu.cn,{tbcui,hlv,swma}@jdl.ac.cn
[1]School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School
[2]Institute of Digital Media, Peking University

ABSTRACT

Intermediate view synthesis is a key technique for free-viewpoint video applications. Various methods have been proposed in this filed. However, most of those methods are so complex that they can't be realized for realtime. In addition, a perfect depth image is hard to acquire for the mixtures of foreground pixels and background pixels at the object boundary. The errors in depth map would involve many holes in intermediate views when it is synthesized between the left-most view and right-most view based on depth map. How to handle the artifacts around object boundaries in intermediate view is another challenge. In this paper, we propose a simple but effective method to enhance the synthesized intermediate views based on edge detecting(SIVBE). Experiment results on typical test sequences show that the quality of intermediate view can be improved by 1.3 dB and the proposed algorithm achieves a reduction of 50% computational complexity on average.

***Index Terms*—** View synthesis, edge detecting, boundary artifact, depth map, 3D video

1. INTRODUCTION

With the rapidly dropping cost of digital cameras and great research progresses in the field of the multimedia, the next generation visual communication services might be deployed, such as 3D TV and free-viewpoint video [1]. Compared with 2D video, huge data quantities involved by multi-view video makes 3D video occupies more network bandwidth and storage. What's more the processing complexity of 3D video is much higher than that of 2D video.

Generally, the multi-view videos can be captured by several cameras in synchronization. And there are two usages about intermediate view synthesis. One is that we can get the free viewpoint from the multi-view video, through synthesis intermediate view. The other is multi-view video coding, as there is much redundancy among different views. Also multi-view Video Coding (MVC) standard has been proposed to improve the coding performance of multi-view by inter-view prediction. Recently, MPEG begins to be devoted to developing new framework of multi-view video coding. By this framework, the multi-view video is down-sampled into two/three views plus depth maps, and they are transmitted to the decoder after coded. In decoder, multi-view video is synthesized based on decoded two/three views and depth map by intermediate view synthesis method. So the bitrate of multi-view video and the number of views are decoupled. By this way, large amounts of redundancy are reduced. Suppose only the left-most view and the right-most view are available, how to get the intermediate views between them? Recently, various methods have been proposed to address this problem, such as 3D image warping, triangular mesh-based rendering, relief texture mapping and inverse mapping [2]. Also SMART [3], Boundary Filtering [4] and Image Dilation [5] were developed to enhance the intermediate view. Compared with the previous work, we propose a simple but effective method to enhance the synthesized intermediate views based on edge detecting(SIVBE) in this paper. The remainder of this paper is organized as follows. Section 2 gives the details of the proposed view synthesis method SIVBE. A λ-decision method in hole filling step would be introduced in Section 3. Section 4 provides the simulation results. Finally, Section 5 concludes the paper.

2. INTERMEDIATE VIEWPOINT VIEWS SYNTHESIS

To improve the quality of synthesized intermediate view, this paper proposes an edge-based rendering algorithm which takes advantages of edge information in the image. And the proposed method SIVBE can be used to alleviate artifacts in the rendering image. Here, two views were used to synthesize six intermediate viewpoint views. SIVBE uses the two disparity images which are computed from depth image values and camera parameters or computed by stereo matching algorithms. Compared with other views synthesis algorithm such as VSRS, SIVBE has lower computational complexity and better subjective quality of intermediate views. Since SIVBE is a forward mapping technique of the synthetic images, the possible occluded holes are padded through the following algorithm which would be introduced in Section 3.

2.1. Disparity Computing from Depth Map

As the depth data is provided in scaled and quantized form, so the true value z need to be obtained first. A typical scaling is to inverse depth scaling with the following equation [6]:

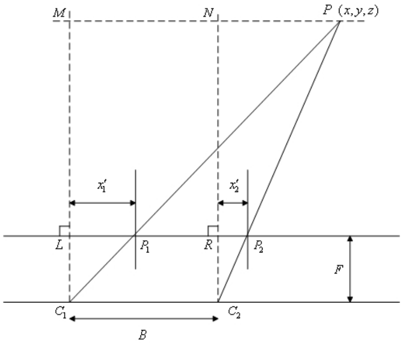


Fig. 1. Relationship of the two camera

$$z(u,v) = \frac{1}{d(u,v) \cdot ((1/z_{min}) - (1/z_{max})) + (1/z_{max})} \tag{1}$$

To synthesize the intermediate viewpoint views, the disparity map must be gained first. In the most basic binocular stereo geometry relationship, it consists of two identical cameras, and their coordinate axes are parallel each other. In Fig. 1, the correspondence pixel at position p_1 and p_2 comes from the projection of the same point P onto both left image and right image. Suppose that the coordinate origin and center of left lens are at the same position. With the triangle PMC_1 and triangle p_1LC_1, we get:

$$\frac{x}{z} = \frac{x_1'}{F} \tag{2}$$

Similarly, with the triangle PNC_2 and triangle p_2RC_2, we get:

$$\frac{x-B}{z} = \frac{x_2'}{F} \tag{3}$$

Finally, the disparity $(x_1' - x_2')$ is obtained using (1) and (3), as shown in the (4):

$$d = x_1' - x_2' = \frac{BF}{z} \tag{4}$$

2.2. View Interpolation

The conventional warping intermediate viewpoint views have many artifacts, which would damage the quality of rendering image. These artifacts are caused by the errors on depth maps. For instance, pixels around the object boundary are a mix of foreground pixels and background pixels which make the boundary blur(see Fig. 2).

As is well-known our eyes are more sensitive to object boundaries. In order to solve these artifacts, a z-weight subpixel interpolation with edge information was proposed in SIVBE based on [8], by distributing the contribution of each interpolated pixel with floating-point coordinates to the two nearest horizontal neighbors with integer coordinates. This method guarantees that all pixels in the rendering image are precise values and consequently alleviates the artifacts in the rendering image.

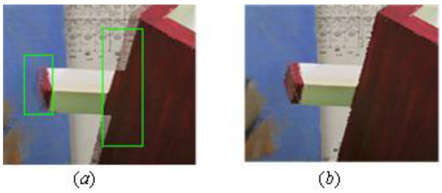


Fig. 2. (a) Blended pixels around object boundary (b) SIVBE

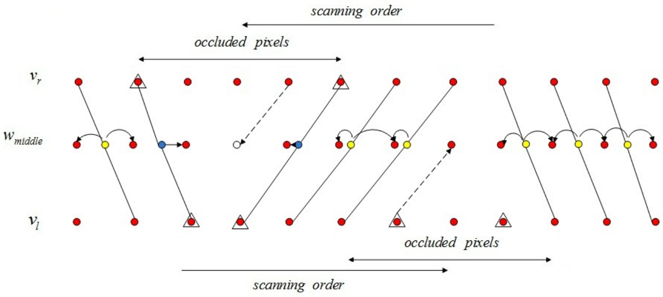


Fig. 3. Round the floating-point.

The red points mean the pixels at an integer location, the yellow mean the interpolated pixels with floating-point, the white mean the holes, and the triangle points mean the edge pixels.

A Canny edge detector [7] is used in SIVBE to find the edge information in the left-most view and the right-most view. The blue points in Fig. 3, which is interpolated by the edge pixels, are not propagated spread to the nearest two neighbor integer points. Here, the floating-point is to directly assign the pixel value to the nearest integer points. Then the float points set P are divided into two sets: Edge set E (there is a interpolated floating position $|x - x_0| \leqslant 0.5$) and Other set O. So the synthesis equation could be

$$w(x_0,y_0) = \begin{cases} w_e(\tau(x_0),y_0), & if(x_0,y_0) \in E \\ w_o(x_0,y_0), & if(x_0,y_0) \in O \end{cases} \tag{5}$$

$$\tau(x_0) = \underset{x \in P}{Argmin}\,|x - x_0| \tag{6}$$

Suppose the left-most view and the right-most view are rectified, the pixel and its correspondence have the same vertical coordinate. As shown in Fig. 3, to get their correspondence with the disparity map, the right-most view pixels are scanned line by line. The intermediate viewpoint view pixels are synthesized from the correspondences in the left-most and right-most views by (7)

$$w_{middle}(x-\alpha d, y) = (1-\alpha)v_l(x,y) + \alpha v_r(x-d,y) \quad (7)$$

where α is the distance ratio between the right-most view and left-most view and d is the disparity of pixel at the position (x, y). Obviously, α is a floating-point value, so the position $(x - \alpha d, y)$ of interpolated view pixel is a floating-point value. Using a round-off operation, the integer location (x_0, y_0) by (8) is obtained.

$$w_o(x_0,y_0) = Round(\frac{1}{C(x,x_0)}\sum_{|x-x_0|<1}\gamma(x,x_0)\cdot w(x,y_0)) \quad (8)$$

$$C(x,x_0) = \sum_{|x-x_0|<1}\gamma(x,x_0) \quad (9)$$

$$\gamma(x,x_0) = \frac{\alpha d_{x_l} + (1-\alpha)d_{x_r} + \varepsilon}{|x-x_0| + c_0} \quad (10)$$

Here c_0 and ε are constant to prevent overflowing, and they are set to be 0.1, 1.1 respectively. The procedure is illustrated in Fig. 3. Since there are occlusions, this procedure couldn't guarantee that every pixel has its correspondence in the other view. Thus, some holes may occur, and a method to fill those holes is proposed in the following section.

3. HOLE FILLING BASED ON DEPTH EDGES

When we change the viewpoint, some areas are disappeared and some are newly exposed. And the errors in depth map would involve the wrong correspondence. So, hole area is caused by the pixels that does not have their correspondence. As shown in the Fig. 4, the point p in left view and point p'_1 in the right view are the projections of scene point P_1 respectively. Also the point p in left view and point p'_2 in the right view are the projections of scene point P_2 respectively. Since the scene point P_2 is a foreground pixel and is prior to the background pixel P_1. So the left view pixel p is only the projection of the point P_2. In this case, the right view pixel point p'_1 has not its correspondence. If there are many pixels without their correspondences in a large area, the holes would occur. Therefore occlusion is the key factor to cause holes.

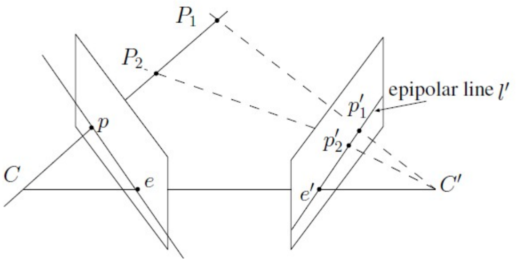


Fig. 4. Occlusion problem.[2]

In general, the depth values of pixel p_1 and p_2 are compared to determine occlusion. If the depth value of pixel p_1 is larger than pixel p_2, the pixel p_1 will be replaced by pixel p_2. However, some noise in depth may have the background pixel overwrite the foreground pixel. To remove these visible errors, a λ-decision method is proposed in SIVBE to classify the left-occlusion and the right-occlusion.

As the Layer extraction [6], depth edges are located by a Canny edge detector. Before the edge detection, the depth map we used is filtered by a median filter. Those edge information would be used in λ-decision.

$$\lambda_l = \sum_{|x-x_l|<\delta}\kappa \times med(d_x), \lambda_r = \sum_{|x-x_r|<\delta}\frac{1}{\kappa}\times med(d_x) \quad (11)$$

$$\kappa = \frac{e_r - x_r + c_0}{x_l - e_l + c_0} \quad (12)$$

Where x_l and x_r are the horizontal axis in origin view of the first neighbor of left view and right view, which have valid point correspondence in the other view. δ is the logarithm to base 2 of the image width. e_l and e_r are the horizontal axis in depth edges of the first neighbor of left hole region border and right hole region border respectively.

Between the relationship depth and disparity (4), the disparity value d_x and depth value z_x are inverse ratio. Therefore, the foreground region is with a higher λ value and we need the other view background region to fill the holes. Through the values of λ_l and λ_r, the left-occlusion and the right-occlusion would be determined. In other words, if $\lambda_l < \lambda_r$, the correspondence of v_r is invalid. The interpolated pixel is taken as

$$H_l(x_0,y_0) = v_l(x_0 - \theta_l(x_0) + \underset{x\in P}{Argmin}\,|x-x_0| + \alpha d_l, y_0) \quad (13)$$

Similarly, if $\lambda_l > \lambda_r$, the correspondence of v_l is invalid. The interpolated pixel is

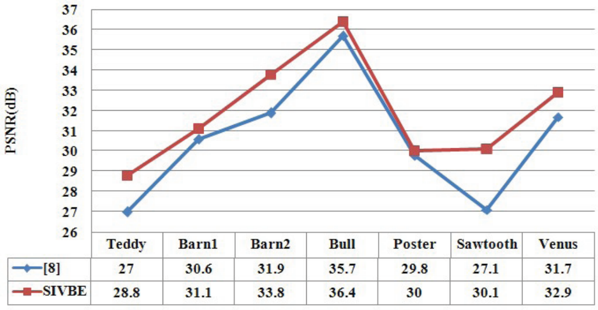
$$H_r(x_0,y_0) = v_r(-x_0+\theta_r(x_0)+\underset{x\in P}{Argmin}\,|x-x_0|-(1-\alpha)d_r, y_0) \quad (14)$$

In (13) and (14), $\theta_l(x_0)$ and $\theta_r(x_0)$ are the horizontal axis integer point in virtual view of the first neighbor of left hole region border and right hole region border respectively. Finally, the holes caused by occlusions are filled by this method.

4. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed algorithm, a simulation is done on data set [9]: "Teddy" (450 × 375), "Barn1" (432×381), "Barn2" (430×381), "Bull" (433×381), "Poster" (435×383), "Sawtooth" (434×380) and "Venus" (434×383). Compared with the recent method [8] in Fig. 5, the quality of intermediate views can be improved by 1.3 dB on average.

Table 1 shows the execution time in SIVBE and VSRS. Samples of the synthesized images are shown in Fig. 6.



	Teddy	Barn1	Barn2	Bull	Poster	Sawtooth	Venus
[8]	27	30.6	31.9	35.7	29.8	27.1	31.7
SIVBE	28.8	31.1	33.8	36.4	30	30.1	32.9

Fig. 5. View interpolation results about PSNR

Table 1. View Interpolation Execution Time(ms)

Index	Teddy	Barn2	Bull	Poster	Venus
SIVBE	94.0	93.0	94.0	78.0	94.0
VSRS	188.0	172.0	188.0	187.0	172.0
Ratio	50.00%	54.07%	50.00%	41.71%	54.65%

5. CONCLUSIONS

In this paper, a simple and effective intermediate view synthesis method based on edge detecting is proposed. And the proposed method SIVBE can be used to alleviate artifacts in the rendering image. Compared with the current technique in the General mode of MPEG view synthesis reference software (VSRS), SIVBE has a low computational complexity. From the experimental results, the proposed SIVBE technique saves as much as 50% computation complexity for view synthesis with even better visual quality than previous works. In the future the SIVBE technique will be further studied for performing in a real-time communication services.

6. ACKNOWLEDGE

This work was partly supported by the grant of Key Projects in the National Science & Technology Pillar Program 2011BA-H08B03, and Shenzhen Basic Research Program of JC201104-210117A, JC201105170732A, JCYJ20120614503O1623.

7. REFERENCES

[1] A.Kubota, A.Smolic, M.Tanimoto, T.Chen, and C.Zhang, "Multi-view imaging and 3DTV," IEEE Signal Process. Mag. vol. 24, no. 6, pp. 10-21. Nov. 2007.

[2] Morvan, Y (Yanninck), "Acquisition, compression and rendering of depth and texture for multi-view video," PhD thesis, Technische Universiteit Eindhoven, 2007.

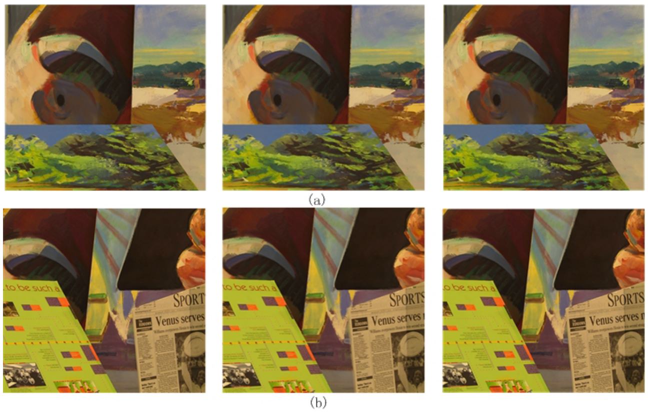


Fig. 6. Samples of synthesized images: (a) Bull, (b) Venus. The most left and right images are input stereo pair v_l, v_r, and intermediate images are synthesized views where $\alpha = 0.5$

[3] Yin Zhao, Ce Zhu, Zhenzhong Chen, Dong Tian, Lu Yu, "Boundary Artifact Reduction in View Synthesis of 3D Video: From Perspective of Texture-Depth Alignment," Broadcasting, IEEE Transactions on, vol.57, no.2, pp.510-522, June 2011.

[4] Cheon Lee, Yo-Sung Ho, "Boundary Filtering on Synthesized Views of 3D Video," Future Generation Communication and Networking Symposia, 2008, FGCNS '08, Second International Conference on, vol. 3, pp. 15-18, 13-15 Dec. 2008.

[5] Kwan-Jung Oh, Sehoon Yea, Yo-Sung Ho, "Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-D video," Picture Coding Symposium, PCS 2009, pp.1-4, 6-8 May 2009.

[6] K. Mller, A. Smolic, K. Dix, P. Merkle, P. Kauff, and T. Wiegand, "View Synthesis for Advanced 3D Video Systems," EURASIP Journal on Image and Video Processing, Special Issue on 3D Image and Video Processing, vol. 2008, Article ID 438148, 11 pages, 2008.

[7] J. Canny, "A computational approach to edge detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 6, pp. 679-698, 1986.

[8] Xiaoyu Xiu, Pang, D., Jie Liang, "Rectification-Based View Interpolation and Extrapolation for Multiview Video Coding," Circuits and Systems for Video Technology, vol. 21, no. 6, pp. 693-707, June 2011.

[9] Scharstein, Daniel and Szeliski, Richard, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," International Journal of Computer Vision, Vol. 47, pp. 7-42, 2002.