A HYBRID PIXEL-BLOCK BASED VIEW SYNTHESIS FOR MULTI-VIEWPOINT 3D VIDEO

Chenxia Li[1], Ronggang Wang[1],Wenmin Wang[1],Lingyu Duan[2], Wen Gao[2]

chenxiali@sz.pku.edu.cn, {rgwang, wangwm}@pkusz.edu.cn, {lingyu,wgao}@pku.edu.cn
[1]School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School
[2]National Engineering Laboratory for Video Technology, Peking University

ABSTRACT

View synthesis technique is important for multi-viewpoint 3D video applications. In our technical investigation, we find that some of the current view synthesis methods perform better in boundary areas while some others in non-boundary areas. For example, MPEG VSRS is the most popular view synthesis software, which integrates a series of advanced tools. Recently, Interval-Based Image Synthesis (IBIS) algorithm is proposed, which shows advantage over MPEG VSRS in non-boundary areas but still has some unpleasant artifacts in boundary areas. To take advantage of the merits of the above methods and overcome their shortcomings, we propose a Hybrid Pixel-block Based View Synthesis(HPBVS) method. Experimental results testify that our approach outperforms the state-of-the-art view synthesis methods from both the subjective and objective perspectives.

Index Terms —Multi-viewpoint, 3D video, View synthesis, Depth, Boundary artifact, Non-boundary artifact

1. INTRODUCTION

Multi-view video is a collection of video sequences of the same scene, synchronously captured by multiple cameras from different viewpoints. It benefits a large diversity of entertainment media and consumer market, including Three-Dimensional TV (3DTV), Free-viewpoint TV (FTV), human-machine interaction, tracking and 3D reconstruction, etc. However, in general, it is unrealistic to capture a scene from all viewpoints by real cameras, not only because of the high costs but also due to the massive quantity of data which need to be processed and transmitted. In order to overcome this challenge, several effective view synthesis schemes have been proposed[1, 2, 3].

As a promising view synthesis technology, Depth-Image Based Rendering(DIBR)[4] has received much attention in recent years. The depth signal is highly compressible and one can generate pictures at any intermediate viewpoints by knowing the depth of the scene[5, 6]. Many variants have been proposed based on DIBR. One effective solution is the bidirectional DIBR [7, 8, 9]: Warping from both reference viewpoints to get two preliminary synthesized virtual images; Get them merged; Fill the remaining holes in the last step. The representative methods are integrated into the view synthesis reference software MPEG VSRS [10, 11]. However, there still exist some unpleasant artifacts in views generated by MPEG VSRS, such as the unsmoothness in background areas (see Figure 7. (c)). Other artifacts are related to dis-occlusion errors induced by the depth estimation algorithms. Recently, an Interval-Based Image Synthesis (IBIS) algorithm is proposed[12], which shows advantage over MPEG VSRS in non-boundary areas but still has some unpleasant artifacts in boundary areas.

This work was partly supported by the grant of National Natural Science Foundation of China 61370115, and Shenzhen Basic Research Program of JC201104210117A, JC201105170732A, and JCYJ2012061450301623

Since some of the current view synthesis methods perform better in boundary areas while some others in non-boundary areas, we propose a hybrid approach which utilizes merits of existing techniques while overcoming their shortcomings, and the view quality of both boundary and non-boundary areas are improved. Firstly, we take advantage of merits of the pixel-interval based rendering idea from IBIS to deal with non-boundary areas; Then, to avoid foreground and background mixing phenomenon in object boundary areas, the bidirectional DIBR is utilized; At last, a ghost contour removal method is exploited to further improve the view quality.

The remainder of this paper is organized as follows: Section 2 presents the architecture of the proposed HPBVS approach. Section 3introduces the technical details of each part in HPBVS. In Section 4, the performance of our method is evaluated and compared with the state-of-the-art techniques. Conclusions are drawn in Section 5.

2. ARCHIETECTURE OF HYBRID PIXEL-BLOCK BASED VIEW SYNTHESIS

Figure 1 shows the flowchart of the proposed method. The Hybrid Pixel-block Based View Synthesis (HPBVS) method can be divided into four stages.

In the first stage, the left view and left depth are warped to the virtual view using a pixel-block based rendering method, which takes advantage of the merits of IBIS in non-boundary areas. And its shortages in boundary areas are overcome by leaving the dis-occlusion part as holes. The same process is performed on the right view and right depth. Thus, two preliminary synthesized images with holes are generated. The technique details are introduced in Section 3.1.

In the second stage, to remove ghost contours showed in Figure 4(a), a ghost contours removing process is conducted, whose details are given in Section 3.4.

In the third stage, the two preliminary synthesized images without ghost contours are merged into a hybrid one with a few holes.

In the last stage, the remaining holes are filled. The final synthesized view is obtained.

In the whole synthesis process, several masks are used to record the positions of the holes to aid the view synthesis process. More details are provided in Section 3.

3. TECHNICAL DETAILS OF HPBVPS

In this section, each component shown in Figure 1 is described in detail. It should be noted that the second stage-ghost contours removing is presented in Section 3.4. Since the ghost contours are noticed after the whole system is finished, this step is added as the second stage at last.

978-1-4799-1369-5/13/$31.00 ©2013 IEEE

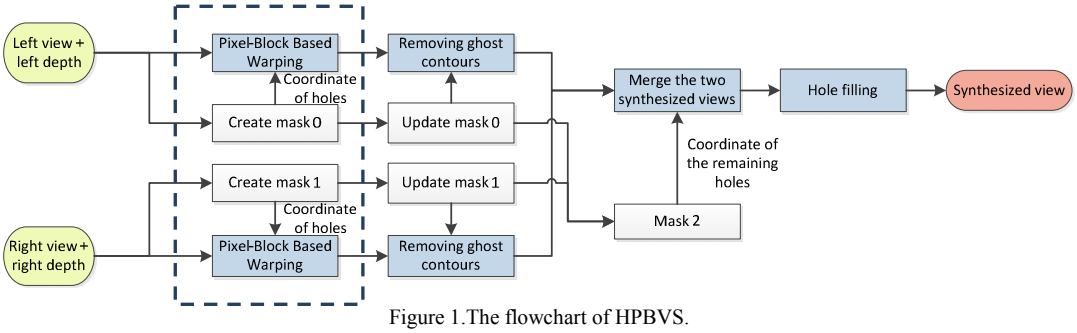


Figure 1.The flowchart of HPBVS.

3.1 Pixel-Block Based Warping From Reference Views

This pixel-block based method utilizes the idea of IBIS, which shows advantage in non-boundary areas. In this step, the left view and right view are warped to generate two preliminary synthesized images, as shown in Figure 2.For boundary areas where IBIS can't handle effectively, we just left them as holes, and fill them by bidirectional DIBR to avoid the foreground and background mixing problem in object boundary areas.

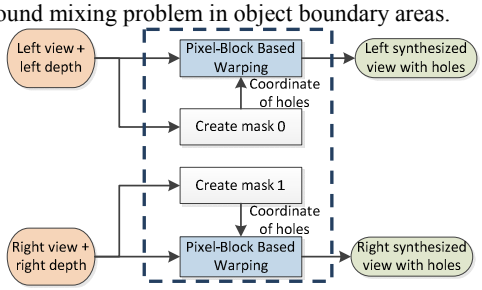


Figure2.Input & output of pixel-block based warping

3.1.1 Disparity Computing from Depth Map

We assume that the cameras are in parallel arrangement and the views are rectified such that no vertical disparities exist. Then a pixel in a reference view can be mapped to its target position by applying a horizontal disparity δ, which depends on the pixel depth z, which can be calculated as:

$$z = \frac{1}{\frac{v}{255}\cdot\left(\frac{1}{Z_{near}}-\frac{1}{Z_{far}}\right)+\frac{1}{Z_{far}}} \quad (1)$$

Where v is the grayscale between 0 and 255, stored in depth map. z_{far} and z_{near} are the minimal and maximal depth of the reference viewpoint. The horizontal disparity δ is then evaluated as:

$$\delta = \frac{f \cdot b}{z} \quad (2)$$

where f represents the focal length of the camera and b represents the horizontal distance between the reference camera and the virtual camera. As a result, a pixel with coordinates (u_r,v) in the reference view will be shifted to coordinate (u,v) in the virtual view with $u=u_r+\delta$. δ is usually a float-point value.

3.1.2 Pixel-Block Based Rendering

We know that each pixel is actually a square on the image plane that cannot be simply described as a point (u, v) in the coordinate system. Therefore, the u-th pixel block, whose coordinate is (u, v), is represented by a geometrical interval I(u) = [u, u + 1) on the image plane.

Generally, the rendering process can be divided into two steps. The first step is to identify all the reference pixels from the corresponding view (say, left view) that exhibits non null intersection with the target interval I(u), i.e. $I(u)\cap I(u_r+\delta)\neq\emptyset$, where δ is the disparity vector applied to reference position u_r. Figure 3 illustrates the strategy of pixel-block based warping. We refer to the set of candidates for target location u as C(u). Take the pixel-block (u, v) in Figure 3 as an example. By searching the corresponding row in the reference view, we find three candidates for (u, v). Thus C(u) consists of three members {P1,P2,P3}. Besides, a tolerance parameter t, initialized to 0, is utilized to enlarge the reference pixel-block to $I(u_r,t) = [u_r - t, u_r + 1 + t)$. For the target locations where no reference pixel is warped to, t is increased to 1. Thus some of the empty pixel-blocks can be covered. In Figure 3, the target pixel-block location (u', v) finally obtain an intersection with P4 after t being increased. For the pixel-blocks without intersection with any reference pixel-blocks (after t is increased to 1) are left as holes. These holes tend to cluster around the dis-occlusion areas (see Figure 4(c)).

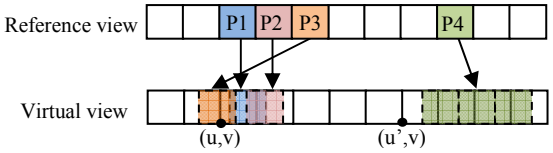


Figure3. Pixel-block based warping

Then, all the candidates in C(u) are sorted by depth, from the camera towards the background. Similar to the IBIS method, each candidate pixel-block participates in the blending process according to two weights of foreground $w_{j,f}$ and background $w_{j,b}$. The color and depth information of the target location u is then computed as:

$$\hat{Y}(u) = \frac{\sum_{j=0}^{|C(u)|-1}(w_{j,f}+\alpha_j \cdot w_{j,b})Y(j)}{\sum_{j=0}^{|C(u)|-1}(w_{j,f}+\alpha_j \cdot w_{j,b})} \quad (3)$$

where Y(j) is the intensity of the j-th reference pixel in C(u), corresponding to pixel-block $I_j(u_r,t)$. The weights $w_{j,f}$ and $w_{j,b}$ are calculated as the portion of the pixel-block of the j-th candidate in the foreground and background with respect to the target pixel-block I(u). The parameter α_j is used to control the mixing phenomenon between foreground and background, as function of the difference between the depth of the j-th and the first pixel. The calculation of these parameters is described in[12].

In particular, to avoid foreground and background mixing phenomenon, a candidate will be regarded as an invalid candidate and will not be involved in the calculating process in equation (3), if the depth difference between it and the first(j=0) candidate exceeds a threshold.

Different from the IBIS method, the above steps are performed on left and right viewpoint respectively. Two synthesized views with holes in dis-occlusion parts are generated. The holes in the two views will be filled using each other's information in the merging operation introduced in Section 3.2.

3.2 Creating Hybrid Synthesized View

In the previous step, two synthesized images with a lot of holes are generated, most of which cluster in the dis-occlusion areas. Now we merge these two images by weighted averaging and

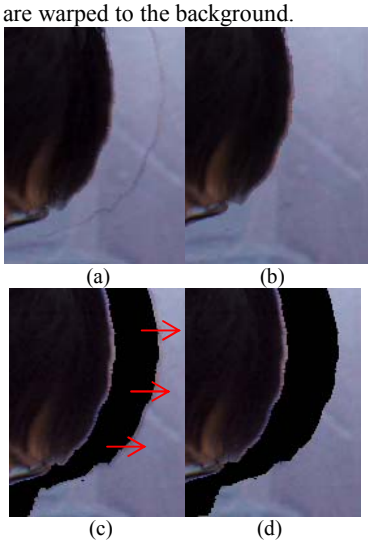
take the baseline spacing into consideration to allocate larger weighting factor to the synthesis resulted from a reference view closer to the virtual view. Let l_L be the baseline distance between the left reference view and the virtual view, and l_R be the baseline distance between the right reference view and the virtual view. Then the weighting factor for synthesized view from left can be defined as $w_L=l_R/(l_L+l_R)$, while the weighting factor for synthesized view from right can be defined as $w_R=l_R/(l_L+l_R)$. One synthesized image with some small holes is then generated.

3.3 Hole Filling

For the synthesized image from the merging process, its remaining holes need to be filled now. Similar to VSRS, the holes are filled by the neighboring pixels that belong to the background. It would be fine to simply propagate the background pixel values into the hole in case of narrow baseline spacing. Hole-filling operation gives us the final synthesized virtual view.

3.4 Ghost Contours Removing

Experiments show that there are still some severe artifacts for some sequences (see Figure 4(a)) after the steps from Section 3.1 to section 3.3. Inaccuracy in depth maps causes textures to be warped to wrong places in the virtual image. This is most visible along edges of objects, where the discontinuities in depth maps are high. This results in ghost contours—foreground textures on edges, which are warped to the background.



(a) (b)

(c) (d)

Figure 4. Removal of ghost contours. (a)ghost contour (b)ghost contour removed (c)dilating direction (d)set ghost contours holes

We notice that ghost contours in the projected views occur on background side borders of dis-occlusions. An effective solution is to dilate the hole area towards background side for several pixels, and then fill them in the merging process [8]. The removal of ghost contours with dilation is illustrated in Figure 4. Then, we need an additional step to determine which side of the dis-occlusions is background. The main idea is to compare the depth value of both sides of the dis-occlusion area. The side with smaller depth value is considered as background. Average depth of five nearest neighboring pixels is calculated for sake of robustness. The average depth is denoted by *d_left* and *d_right* for left and right side respectively. Then the background side is determined according to the following rules:

$$d\_left - d\_right \begin{cases} > threshold: & \text{left side is background, d=1} \\ < -threshold: & \text{right side is background, d=2} \\ else & \text{: not ghost contour area, d=0} \end{cases}$$

where d represents the dilating direction: 1 for leftward, 2 for rightward, and 0 for doing nothing as no ghost contours exist in this case.

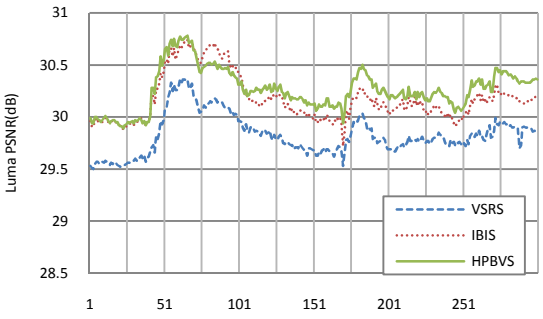
4. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed method, experiments are carried out on test sequences shown in Table 1. The last column gives the camera IDs of two reference viewpoints and the corresponding virtual viewpoint. The performance of our proposed method is quantified based on Peak Signal Noise Ratio (PSNR) and is compared to MPEGVSRS(version 3.5) [11] and the IBIS method [12].Table 2 shows the average PSNR of the first 300 synthesized frames of the sequences in Table 1. Average PSNRs of these three methods (see Figure 6) show that our proposed HPBVS method outperforms VSRS and IBIS and is robust on various sequences.

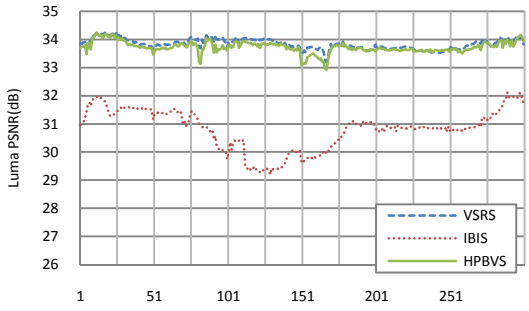
Figure 7 compares details of the synthesized view. Figure 7(b) shows that HPBVS achieves good results in boundary areas. In Figure 7(c), there are some visible cracks in views generated by VSRS and IBIS, while the view generated by HPBVS is very smooth and most close to the real view. The results verify that HPBVS performs well in both boundary and non-boundary areas.

Table 1.Testsequences

name	Resolution	Frames	Camera NO.
Newspaper	1024*768	300	2,4→3
Kendo	1024*768	300	3,5→4
Balloons	1024*768	300	3,5→4
Café	1920*1080	300	2,4→3



(a)



(b)

Figure 5.PSNR comparison. (a)newspaper (b)café

Table 2. Average PSNR

sequences \methods	VSRS	IBIS	HPBVS
Newspaper	29.82	30.17	30.26
Kendo	37.66	38.06	38.11
Balloons	36.62	37.02	36.84
Cafe	33.83	30.80	33.72
Average	34.48	34.01	34.73

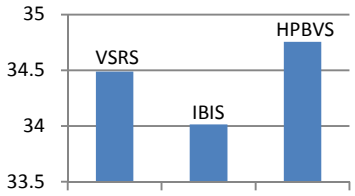


Figure 6.Average PSNRs over test sequences.

5. CONCLUSIONS

The state-of-the-art view synthesis methods create unpleasant visual effects either in the boundary areas or non-boundary areas. In this paper, a hybrid pixel-block based view synthesis method which utilizes merits of existing techniques and overcome the shortcomings is proposed. Experimental results testify the effectiveness of our proposed HPBVS method.

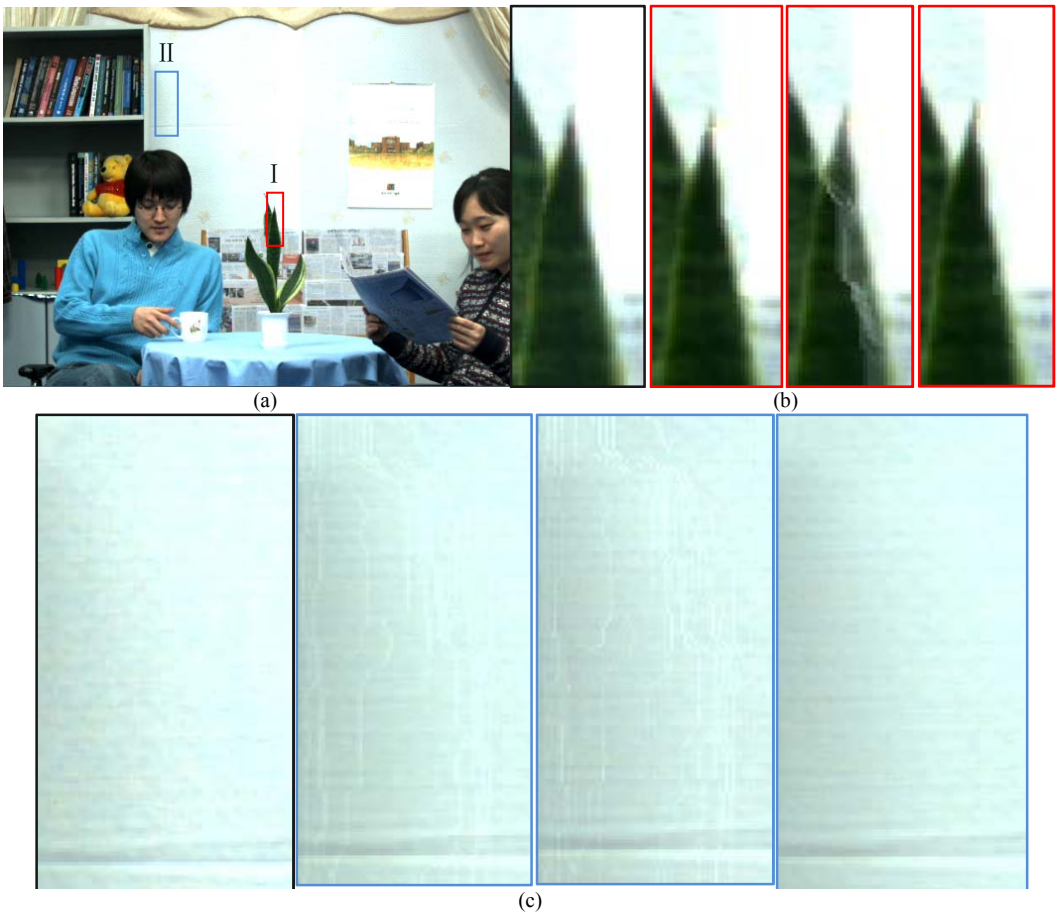


Figure 7.Subjective comparisons. (a)original view(newspaper, camera 3, 0-th frame) (b) block Ⅰ (from left to right: ground truth, VSRS, IBIS, HPBVS) (c) block Ⅱ (from left to right: ground truth, VSRS, IBIS, HPBVS)

6. REFERENCES

[1] Zitnick, C. Lawrence, et al. "High-quality video view interpolation using a layered representation." *ACM Transactions on Graphics (TOG)*. Vol. 23. No. 3. ACM, 2004.

[2] Y. Morvan, "Acquisition, compression and rendering of depth and texture for multi-view video," Ph.D. thesis, Eindhoven University of Technology, 2009.

[3] Lang, Manuel, et al. "Nonlinear disparity mapping for stereoscopic 3D." *ACM Transactions on Graphics (TOG)* 29.4 (2010): 75.

[4] Fehn, Christoph. "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV." *Electronic Imaging 2004*. International Society for Optics and Photonics, 2004.

[5] Muller, K., et al. "Coding and intermediate view synthesis of multiview video plus depth." *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009.

[6] Kim, Woo-Shik, et al. "3-D video coding using depth transition data." *Picture Coding Symposium (PCS), 2010*. IEEE, 2010.

[7] Mori, Yuji, et al. "View generation with 3D warping using depth information for FTV." *Signal Processing: Image Communication* 24.1 (2009): 65-72.

[8] Zinger, Sveta, and Luat Do. "Free-viewpoint depth image based rendering."*Journal of Visual Communicationand Image Representation* 21.5 (2010): 533-541.

[9] Feng, Ya-mei, et al. "Asymmetric bidirectional view synthesis for free viewpoint and three-dimensional video." *Consumer Electronics, IEEE Transactions on* 55.4 (2009): 2349-2355.

[10] Tian, Dong, et al. "View synthesis techniques for 3D video." *SPIE Optical Engineering+ Applications*. International Society for Optics and Photonics, 2009.

[11] "Viewsynthesisreference software(VSRS)3.5," Tech.Rep.,ISO/IEC JTC1/SC29/WG11, Mar.2010.

[12] Paradiso, Vincenzo, Maurizio Lucenteforte, and Marco Grangetto. "A novel interpolation method for 3D view synthesis." *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2012*. IEEE, 2012.