VIRTUAL VIEW SYNTHESIS QUALITY REFINEMENT

Tzu-Chin Lee, Chun-Liang Chien, Hsueh-Ming Hang

Dept. of Electrics Engineering, National Chiao Tung University, Taiwan, R. O. C.

ABSTRACT

A 3D virtual view synthesis system is to generate a virtual view at an arbitrary viewpoint from the texture and depth information of multiple reference views. There are several technical challenges in producing a high-quality synthesized view such as warping, blending, ghost artifact reduction and hole filling. Four tools (techniques) have been proposed in this paper to solve these problems. They are unreliable region extraction, fast backward warping, adaptive blending, and hole filling by using the constructed background models. Other techniques such as disocclusion detection, bicubic interpolation, and background model construction are also employed in the proposed algorithm. All above techniques have been designed and tested on the MPEG test sequences. Experimental results show that high-quality virtual views are generated with fewer artifacts.

Index Terms — Virtual view synthesis, backward warping, disocclusion, background, hole filling, FTV, 3DTV, DIBR

1. INTRODUCTION

Multimedia applications such as free viewpoint television (FTV), 3DTV and virtual reality provide viewers with 3D experience by presenting videos from different viewpoints to our eyes. However, transmitting the tremendous amount of 3D data is extremely inefficient even if compressed with multi-view video coding (MVC) [1]. Therefore, multi-view video plus depth (MVD) format has being explored to generate virtual views, and the most popular view synthesis system adopted by the ITU/MPEG standard uses depth image based rendering (DIBR) techniques [2].

View synthesis is the process of using the texture and depth information of the same scene captured from different camera positions to generate a virtual view at an arbitrary viewpoint where no camera was actually located. View synthesis based on two different views typically consists of three stages: 3D warping, view blending, and hole filling [3] as illustrated in Figure 1.

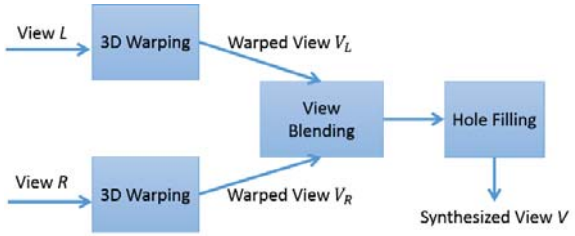


Figure 1. Flowchart of view synthesis using two input camera views L and R to synthesize a target virtual view V

At first, images at the target viewpoint is created based on the reference images using the warping techniques. Second, a virtual view is synthesized in between two reference views to reduce holes. The simplest way would be taking a weighted average of two images. Some methods also select the closest pixels base on the z-buffer method. A view blending method can be the combination of these strategies. The final step is to fill the remaining holes. Many hole filling methods have been developed such as linear interpolation using neighboring pixels or inpainting techniques [4, 5]. However, the remaining holes are not random regions of an image, it is often better to fill the holes using the background pixels rather than the foreground ones [6, 7].

The rest of paper is organized as follows. In Section 2, the 3D warping techniques are discussed. The proposed blending method is presented in Section 3. Section 4 describes our hole filling proposals. Experimental results are shown in Section 5. Finally, the conclusions are given in Section 6.

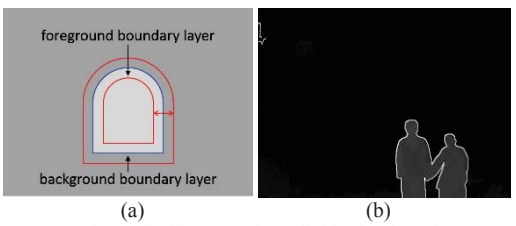
2. 3D WARPING TECHNIQUES

The proposed 3D warping techniques contain five stages: unreliable region extraction, forward depth warping (FDW), disocclusion hole detection, backward depth warping (BDW), and backward texture warping (BTW). First, to avoid boundary noise, the unreliable regions are extracted and excluded as described in Section 2.1. Then, we use FDW to generate the warped depth map at virtual view as described in Section 2.2. However, cracks due to the FDW process show up. Hence, the BDW is adopted to solve this problem and the details are described in Section 2.4. Then, the disocclusion holes are detected before performing the BDW process. The details are described in Section 2.3. This information will be used to speed up the BDW and improve the view blending method. Finally, we use BTW to synthesize the virtual color image as described in Section 2.5.

2.1 Unreliable region extraction

Due to the imperfect depth map, some foreground boundary pixels are mistakenly denoted as the background pixels and then warped to the wrong position. Such boundary noise (also known as ghost artifact) need to be removed. To tackle this problem, we need to identify the reliable and unreliable depth regions in the original views. After that, the detected unreliable regions are excluded in the warping process as described in the next section. The procedure of unreliable regions extraction are described as follows.

1) Apply a Canny edge detector to the depth image to detect the significant depth discontinuities, as the blue contour shown in Figure 2(a). Then, mark a 7-pixel-wide area along the detected edges, as the region between red contour shown in Figure 2(a).
2) Split the areas marked in step 1 into the foreground and background boundary layers as the areas inside and outside the blue contour, respectively. The background boundary layer is considered unreliable, for example, the white part shown in Figure 2(b) is the unreliable region.



(a) (b)

Figure 2. (a) Schematic diagram of unreliable depth region extraction. (b) Unreliable depth regions in the reference view.

978-1-5090-3313-3/16/$31.00 ©2016 IEEE

2.2 Forward depth warping

To synthesis a virtual view, at first, two reference views are forward warped to the virtual view. The equation of 3D warping is given below [3].

$$\mathbf{p}_v = \frac{1}{z_v}\mathbf{K}_v(\mathbf{R}_v\mathbf{R}_r^{-1}(z_r\mathbf{K}_r^{-1}\mathbf{p}_r - \mathbf{t}_r) + \mathbf{t}_v) \quad (1)$$

where $\mathbf{p}$ is the position vector $(u, v, 1)$ in the image plane. The subscript r indicates an item associated with the reference view, and v indicates the virtual view. $\mathbf{R}_{3\times3}$ and $\mathbf{t}_{3\times1}$ represent the rotation matrix and the translation vector, respectively. $\mathbf{K}$ is the intrinsic matrix and z is the physical depth value. Equation (1) can be rewritten below.

$$z_v\begin{bmatrix}u_v\\v_v\\1\end{bmatrix} = z_r\boldsymbol{A}\begin{bmatrix}u_r\\v_r\\1\end{bmatrix} + \boldsymbol{b} \quad (2)$$

where

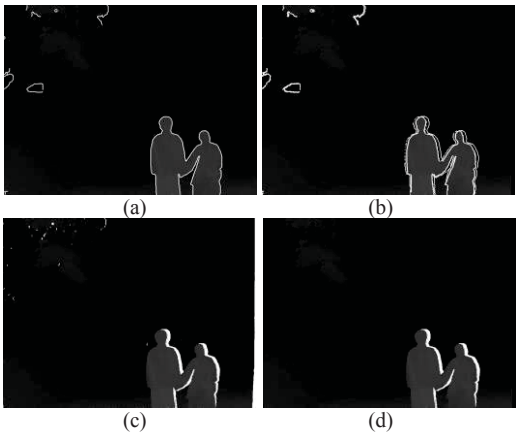
$$\boldsymbol{A} = \boldsymbol{Q}_v\boldsymbol{Q}_r{}^{-1}, \quad \boldsymbol{b} = \boldsymbol{Q}_v(\boldsymbol{c}_r - \boldsymbol{c}_v),$$
$$\boldsymbol{Q}_r = \boldsymbol{K}_r \cdot \boldsymbol{R}_r, \quad \boldsymbol{Q}_v = \boldsymbol{K}_v \cdot \boldsymbol{R}_v,$$
$$\boldsymbol{c}_r = -\boldsymbol{R}_r{}^{\boldsymbol{T}} \cdot \boldsymbol{t}_r, \quad \boldsymbol{c}_v = -\boldsymbol{R}_v{}^{\boldsymbol{T}} \cdot \boldsymbol{t}_v$$

In this equation, $\boldsymbol{A}$, $\boldsymbol{b}$, u_r, v_r and z_r are known. After forward warping, most of the pixels in the warped view are determined. However, the remaining pixels which never get mapped from the reference view are called crack or disocclusion. The cracks will be filled by backward warping as described in Section 2.4. On the other hand, the disocclusion hole will be detected as described in next section and will be excluded from backward warping and be filled by constructing the background as described in Section 4.

2.3 Disocclusion hole detection

A critical problem in the view synthesis system is how to deal with the hole regions after 3D warping. The disocclusion holes are regions which cannot be seen from reference view but appear in the virtual view. This happens due to the depth discontinuities between the foreground objects and the background. The steps of detecting disocclusion holes are as follows.

1) Find the depth edges in reference view and mark a 3-pixel-wide area along the detected edges as shown in Figure 3(a).
2) Warp the depth edges to the virtual view and mark it, as the white part shown in Figure 3(b).
3) If a hole has a significant amount of marks near its border, this hole is considered to be a disocclusion hole, as shown in Figure 3(d).



(a) (b) (c) (d)

Figure 3. (a) The depth edges in reference view. (b) The warped depth edges in virtual view. (c) Holes in virtual view. (d) Disocclusion hole.

2.4 Backward depth warping

Some pixels may have wrong depth values or never get mapped after FDW. To deal with this problem, the BDW method is adopted in this work; that is, we map pixels from the target virtual view to the reference view [8]. In our experiment, we found that BDW can fix the small cracks on the objects and it can also fix the background appear in the holes between two nearby foreground objects. To do the backward warping, we rearrange the forward warping equation (2) into

$$z'\begin{bmatrix}u'\\v'\\1\end{bmatrix} = \boldsymbol{A}^{-1}(z_v\begin{bmatrix}u_v\\v_v\\1\end{bmatrix} - \boldsymbol{b}) \quad (3)$$

where u', v' and z' are the floating numbers that represent the calculated position and depth value of the pixel in the reference view. The known values in this equation are $\boldsymbol{A}$, $\boldsymbol{b}$, u_v, and v_v. However, we do not have the depth value of virtual view depth z_v before the BDW process is completed. Hence, an error function is defined to determine the best matching value z_v as follows.

$$E = min[(u_r - u')^2 + (v_r - v')^2 + (D(u_r, v_r) - D')^2] \quad (4)$$

where (u_r, v_r) is substituted by the four nearest pixels of (u', v') as illustrated in Figure 4. $D(u_r, v_r)$ is the disparity on (u_r, v_r), and D' and D_v are the disparity corresponding to z' and z_v, respectively. Note that often the depth map is in fact a disparity map. There is a fixed mapping rule from a depth value z to a disparity value D and the other way round. Therefore, we use these two terms, depth map and disparity map, interchangeably.

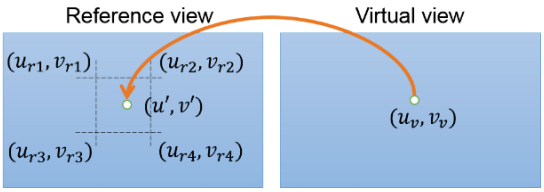


Figure 4. Backward depth warping process

To obtain the best matched virtual depth value, a simple way is to try every possible D_v in the range of [0, 255], and pick up the value with the minimum error E, described in equation (5). However, this method requires a massive computational complexity. Hence, to increase the speed, we propose a method consists of the following two techniques.

$$D_v^* = arg\min_{0\le D_v\le 255} E \quad (5)$$

First, we like to reduce the number of pixels needed for running the BDW process. Since most pixel values in the forward warped depth map are equal to the backward warped depth map, there are only two cases of the pixels that need to be processed by BDW:

1) The depth value is unavailable after FDW and it is not a disocclusion hole. We do not apply BDW to disocclusion holes because these pixels are not available from reference view.
2) The depth value is changed after applying the median filter, and thus is considered as unreliable depth pixel.

The second technique is to reduce the computation for every pixel which needs BDW. We observe that the depth value of a pixel is often strongly correlated to the depths of its neighboring pixels. Therefore, in testing the D_v value, we start from the maximum value of its neighbors in the 5×5 window centered at that pixel. Moreover, we test D_v in descending order, and accept the

value when the first local minimum E occur. Hence, we do not need to try 256 possible values.

From our experiment, the proposed method with above two techniques can greatly reduce the computation time, and it is about 500 times faster than the original BDW method.

2.5 Backward texture warping

After obtained a warped virtual depth map, we need to map the texture information to synthesize the virtual color image. We use BTW since the forward warping is not accurate enough as discussed before [7]. The backward warping shown in equation (3) can be used directly to calculate u' and v'. Note that (u', v') may not be an integer pixel position. To achieve better visual result, we use the bicubic interpolation method to interpolate the texture value at (u', v') using its neighbors rather than simply rounding (u', v') to the nearest pixel position.

3. ADAPTIVE BLENDING

A virtual view synthesized in between two reference views can significantly reduce the holes by blending the two warped reference views. A drawback of the simple blending rules is that inconsistent pixel values from both views are counted to form the warped image [9]. This method often leads to *double edges artifacts* and *blurring*.

To solve this problem, we propose an improved blending method which uses the weighted average only when two warped views have similar depth and color values. Otherwise, we choose the more reliable warped reference pixel as the pixel in the synthesized view. The proposed adaptive blending algorithm is described by the following pseudo codes.

Algorithm 1 The adaptive blending method

```
if |D_L(u,v) − D_R(u,v)| < th_1 and |I_L(u,v) − I_R(u,v)|^2 < th_2
    I_v(u,v) = (1 − α)I_L(u,v) + αI_R(u,v)
    D_v(u,v) = (1 − α)D_L(u,v) + αD_R(u,v)
else
    if D_L(u,v) is more reliable
        I_v(u,v) = I_L(u,v); D_v(u,v) = D_L(u,v)
    else if D_R(u,v) is more reliable
        I_v(u,v) = I_R(u,v); D_v(u,v) = D_R(u,v)
    else
        if D_L(u,v) > D_R(u,v)
            I_v(u,v) = I_L(u,v); D_v(u,v) = D_L(u,v)
        else
            I_v(u,v) = I_R(u,v); D_v(u,v) = D_R(u,v)
        end
    end
end
```

where the subscript v indicates an item associated with the virtual view and the subscript L and R indicate an virtual view item projected from the left and the right cameras, respectively. I is the color value and D is the disparity value of a target pixel. The thresholds, th_1 and th_2 , are used to check whether the two warped depth values and color values, respectively, are close to each other. The weighting factor α is calculated based on the baseline distance as follows:

$$\alpha = \frac{|t_V - t_L|}{|t_V - t_L| + |t_V - t_R|} \tag{6}$$

where t is a translation vector for each view.

To decide which warped reference pixel is reliable, the holes in an image are our criterion. If the pixel is a hole, it is considered the most unreliable. Otherwise, the pixel near disocclusion hole is considered less reliable than those far from disocclusion hole. If the reliability of two pixels are the same, we choose the one which is near the camera because the foreground object blocks the background object. In our experiments, the improved blending method can effectively reduce the double edges artifacts and blurring as the experimental results shown in Section 5.

4. BACKGROUND RECONSTRUCTION AND HOLE FILLING

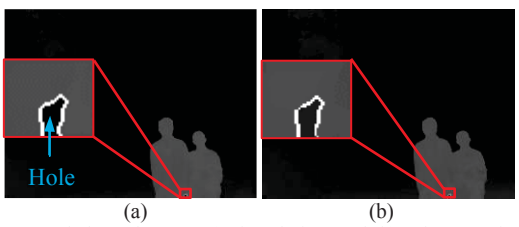
After the view blending process, some disocclusion holes may still remain. Hence, it may be helpful if we can get information from the other frames to find the texture in the disocclusion region.

The proposed method is to exploit the texture and depth information from whole sequence to generate the background images (or background models) [10]. This technique is often called background modeling. The conventional background modeling techniques use only the color image information but we now also use the depth information. Then, we synthesize the virtual background by using two reference background models. The warping process is the same as described in Section 2 and the blending process described in Section 3. Figure 5 shows our constructed background synthesized by using two reference background models. Finally, we fill the disocclusion holes in virtual view by using the constructed background after careful justification. The hole filling method is given below.

1) Warp the hole boundary, as shown in Figure 6(a), back to the reference view, as shown in Figure 6(b), and check its surrounding depth values.
2) Estimate a reasonable depth value of the hole based on the minimum and a threshold derived from the surrounding depths shown in Figure 6(b).
3) Fill the hole pixel in virtual view by the constructed background if the background depth value sits in the acceptable range.
4) Use the traditional hole filling method to fill the rest of remaining holes.

The above steps are developed to ensure the correct use of the constructed background models. Due to the proper use of the depth information in the filling process, we can precisely decide which holes should be filled by the constructed background even when the hole is surrounded by the foreground objects. The experimental results are shown in Section 5.

Figure 5. The constructed background synthesized by two reference background models.



(a) (b)

Figure 6. Hole boundary on (a) virtual view and (b) reference view, respectively.

5. EXPERIMENTAL RESULTS

We tested our proposed algorithm on the MPEG test sequence "lovebird". Two reference views are captured from camera 4 and camera 8. The final synthesized view of the 100th video frame by the view synthesis reference software (VSRS version 3.5) and our proposed approach are compared in Figure 7. Figure 8 to Figure 11 are the enlarged details in Figure 7. The visual quality of the final synthesis view by the proposed method is better than that by VSRS.

Figure 8(a) shows an example of boundary noise in the synthesized view, where a ghost copy of the foreground contour appear in the background. Figure 8(b) shows that our method, taking the reliability information into account, can significantly reduce the boundary noise. The drawback of the simple blending rules such as double edges artifacts and blurring are shown in Figure 9(a) and Figure 10(a). Figure 9(b) and Figure 10(b) show that the synthesized view with the adaptive blending method can solve this problem. Here, th_1 is set to 20 and th_2 is set to 3000. Figure 11 shows that our hole filling method using the constructed background performs well even when the hole is surrounded by foreground objects.

(a) (b)

Figure 7. The synthesized views by (a) VSRS and (b) the proposed approach, respectively.

(a) (b)

Figure 8. The details of one part of the synthesized views by (a) VSRS and (b) the proposed method, respectively.

(a) (b)

Figure 9. The details of a portion of the synthesized views by (a) VSRS and (b) the proposed method, respectively.

(a) (b)

Figure 10. The details of a portion of the synthesized views by (a) VSRS and (b) the proposed method, respectively.

(a) (b)

Figure 11. The details of a portion of the synthesized views by (a) VSRS and (b) the proposed method, respectively.

6. CONCLUSIONS

In this paper, we propose four tools to improve the quality of synthesized virtual views. First, the unreliable region extraction can reduce the ghost artifact due to the misalignment between texture images and depth images. The second tool is a fast algorithm of BDW, which can fix small cracks and some artifacts caused by FDW. We also reduce the computational complexity of the time consuming BDW. The third one is an adaptive blending algorithm, which can handle double edges artifacts and blurring in the synthesized view, and produce better subjective quality results. Finally, we propose a novel hole filling method to decide whether a hole should be filled by the constructed background model or not. Furthermore, some techniques such as disocclusion hole detection, bicubic interpolation, and background model construction are also employed in the proposed approach. Based on the experimental results in this study, it is found that our proposed approach performs very well on improving the subjective quality of the synthesized virtual views.

7. ACKNOWLEDGEMENT

This work was supported in part by the MOST, Taiwan under Grant MOST 103-2221-E-009 -065 and by the Aim for the Top University Project of National Chiao Tung University, Taiwan.

8. REFERENCES

[1] Y.-S. Ho and K.-J. Oh, "Overview of multi-view video coding," in *Systems, Signals and Image Processing, 2007 and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services. 14th International Workshop on*, 2007, pp. 5-12.

[2] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," in *Electronic Imaging 2004*, 2004, pp. 93-104.

[3] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila, "View synthesis techniques for 3D video," *Applications of digital image processing XXXII, proceedings of the SPIE,* vol. 7443, pp. 74430T-74430T, 2009.

[4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 417-424.

[5] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing,* vol. 13, pp. 1200-1212, 2004.

[6] K.-J. Oh, S. Yea, and Y.-S. Ho, "Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-d video," in *Picture Coding Symposium, 2009. PCS 2009*, 2009, pp. 1-4.

[7] S. Zinger, L. Do, and P. de With, "Free-viewpoint depth image based rendering," *Journal of visual communication and image representation,* vol. 21, pp. 533-541, 2010.

[8] D.-H. Li, H.-M. Hang, and Y.-L. Liu, "Virtual view synthesis using backward depth warping algorithm," in *PCS*, 2013, pp. 205-208.

[9] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "View generation with 3D warping using depth information for FTV," *Signal Processing: Image Communication,* vol. 24, pp. 65-72, 2009.

[10] Y.-L. Liu and H.-M. Hang, "Background modeling using depth information," in *Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA)*, 2014, pp. 1-4.