VIRTUAL VIEW SYNTHESIS USING RGB-D CAMERAS

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

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

ABSTRACT

A view synthesis problem is to generate a virtual view based on the given one or multiple views and their associated depth maps. We adopt the depth image based rendering (DIBR) approach in this paper for synthesizing the new views. No explicit 3D modeling is involved. Another component of this study is the popular commodity RGB-D (color plus depth) cameras. The color and depth images captured by a pair of RGB-D cameras (Microsoft Kinect for Windows v2) are our inputs to synthesize intermediate virtual views between these two cameras. Several methods include depth to color warping, disocclusion filling, and color to color warping are adopted and designed to achieve this target. One of our major contributions is a new disocclusion detection algorithm proposed to improve the disocclusion filling result. Furthermore, an improved camera calibration method is proposed to make use of the additional depth information. Good quality synthesized views are shown at the end.

Index Terms — View synthesis, camera calibration, backward warping, disocclusion filling, depth map, Kinect

1. INTRODUCTION

One key element of 3D and VR (Virtual Reality) systems is virtual view synthesis, constructing a new view (image) based on one or more given views. When the number of views is pretty large, the cost to transmit all views may be too expensive. Instead, the intermediate views can be synthesized from a few received views. After two videos plus their corresponding depth are captured, additional intermediate views can be rendered using depth image based rendering (DIBR) techniques [1].

Recently, there has been an increasing number of RGB-D cameras available at commodity prices, such as Microsoft Kinect and Intel Realsense. These cameras can capture both color and depth images in real time. Hence, they are very suitable for developing a real-time virtual view synthesis system. In this paper, we propose a view synthesis prototype system, in which we use images and depth map captured by Kinect for Windows v2 to synthesis intermediate virtual views between two RGB-D cameras. In our algorithm, at first, the original depth map is mapped to the original color image. We use the backward warping method to avoid generating cracks. Second, an improved disocclusion detection method is proposed to detect the disocclusion areas and then fill up them. Third, an improved camera calibration method is proposed to find the projection matrix between two RGB-D cameras. Fourth, the captured left and right color images are warped to an intermediate view based on their corresponding depth maps. Finally, a simple blending method is adopted to generate the virtual view image.

2. DEPTH TO COLOR WARPING

The proposed double RGB-D cameras system is composed of two Kinect v2 cameras. For view synthesis applications, which use color and depth image jointly, it is critical to know the calibration parameters of the sensors. The resolution of the original depth map captured by Kinect v2 (512×424) as shown in Figure 1(b) is much lower than that of the original color image (1920×1080) as shown in Figure 1(a). Although the color camera calibration problem has been thoroughly studied in the literature [2, 3], the joint calibration of depth and color images presents a few new challenges [4] including: (1) feature points such as the corners of checkerboard patterns are often indistinguishable from other surface points in the depth image; (2) the boundary points in the depth image are usually unreliable due to unknown depth reconstruction mechanisms; (3) most commodity depth cameras produce noisy depth images. In this work, we use the *MapDepthFrameToColorSpace* function in Kinect for Windows SDK to map the depth map to the color image as shown in Figure 1(c). However, if the off-the-shelf function is not available for some RGB-D cameras, we can use the camera calibration method described in Section 4 to obtain the calibration parameters of the RGB-D camera pairs. Note that the depth value of pixels on the left and right sides are unavailable due to the different angles of views between depth and color sensors. In the following processes, we truncate both sides of the color and depth images by the red lines indicated in Figure 1(a) and (d).

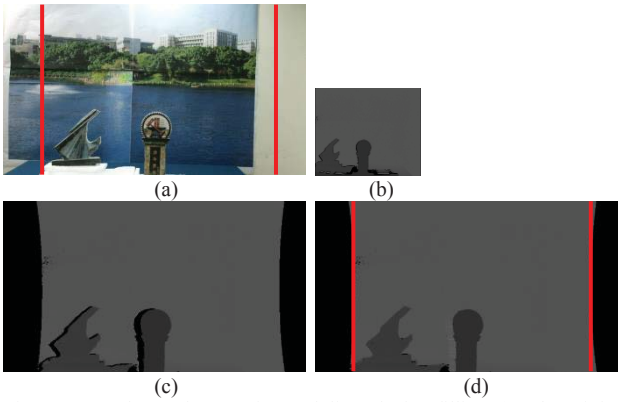


Figure 1. Depth to color warping and disocclusion filling. (a) The original color image captured by Kinect v2. (b) The original depth map captured by Kinect v2. (c) The warped depth map. (d) The disocclusion filled depth map.

3. DISOCCLUSION DETECTION AND FILLING

After the depth map is mapped to the color image, some disocclusion regions appear due to the distance between depth and color sensors. Note that disocclusion regions always locate along the transition zones between the foreground and the background regions, where the depth levels are different between these two regions [5]. To detect the disocclusion regions, we examine a row of pixels from the reference view. Let a pixel $p = (p_x, p_y)$ on the reference view be projected to the position $p' = (p'_x, p_y)$ in the synthesized view. After having processed pixel p, the next pixel to be processed is $q = (p_x + 1, p_y)$ and its projected position $q' = (q'_x, p_y)$. To reduce false alarms, we combine two disocclusion

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

detection methods [6, 7] to form a tighter disocclusion detection condition as follows.

$$q'_x > (p'_x + 1) \text{ and } \left(D(p_x, p_y) - D(q_x, p_y)\right) > T, \qquad (1)$$

where $D(i, j)$ is the depth value at position (i, j) and T is a predefined threshold value typically in the range of 3 to 5. However, some false alarms still occur due to the noisy depth map.

To resolve this problem, we propose a *disocclusion constant* to remove these artifacts. As illustrated by Figure 2, pixel D in the (image) view of depth sensor is mapped to pixel A in the view of the color camera; therefore, the region between B and C is disoccluded. We are looking for the relationship between d', the width of disocclusion region on the image, and r_1, the depth of foreground, and r_2, the depth of background. At first, based on ΔABC, d' can be derived as follows.

$$d' = df/r_2 \qquad (2)$$

where d is the width of disocclusion region in a scene and f is the focal length. Similarly, based on ΔABC, d can be derived as follows.

$$d = w'r_2/r_1 \qquad (3)$$

Finally, based on ΔABD, w' can be calculated by $w' = w(r_2 - r_1)/r_2$, where w is the distance between the central axis of color camera and the central axis of depth camera. By substituting w' back into Eq. (3), d can be expressed by Eq.(4).

$$d = \frac{w(r_2 - r_1)}{r_2} \cdot \frac{r_2}{r_1} = \frac{w(r_2 - r_1)}{r_1} \qquad (4)$$

Similarly, by substituting d back into Eq. (2), d' can be derived as

$$d' = \frac{wf(r_2 - r_1)}{r_1 r_2} \qquad (5)$$

Hence, we obtained the relationship between d', r_1, and r_2 as follows.

$$\frac{d'r_1r_2}{r_2 - r_1} = wf = DC \qquad (6)$$

where DC is constant because w and f are both fixed values for a captured image and depth map. We call this constant DC, the *disocclusion constant*. If the calculated DC value is out of a range, some of the disocclusion regions detected by Eq. (**1**) are incorrect. This constraint can be used to filter out most false alarm cases.

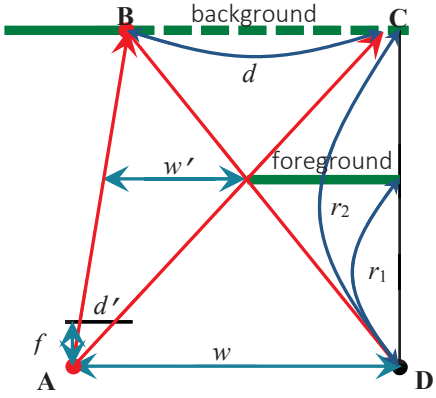


Figure 2. The illustration of the proposed disocclusion constant (*DC*). The disocclusion region between B and C appear in the image when the camera viewpoint is moved from D to A.

Note that often the depth map is, in fact, a disparity map. The depth value detected by Kinect v2 is in millimeter (mm). There is a fixed mapping rule from a depth value to a (unique) disparity value and vice versa. Therefore, we use these two terms, depth map and disparity map, interchangeably.

In practice, it may not be convenient to access the actual values of w and f. To estimate an approximate value of DC, we collect data and calculate the distribution of DC based on measured points p' and q' in Eq. (1). As shown in Figure 3, we observed that most DC values are in the range of [2000, 3000]. Hence, a new constraint can be used to detect the invalid disocclusion regions using the following inequality,

$$\frac{d'r_1r_2}{r_2 - r_1} \leq 2000 \text{ or } \frac{d'r_1r_2}{r_2 - r_1} \geq 3000 \qquad (7)$$

In summary, a new disocclusion detection method is proposed based on the following two constraints:

1. Two horizontally connected pixels in the original depth map are separated in the warped depth map and the depth value of the right pixel should be closer to the camera than that of the left pixel as described in Eq. (1).
2. The ratio of Eq. (6) is set to be in the range of [2000, 3000] based on our experiments.

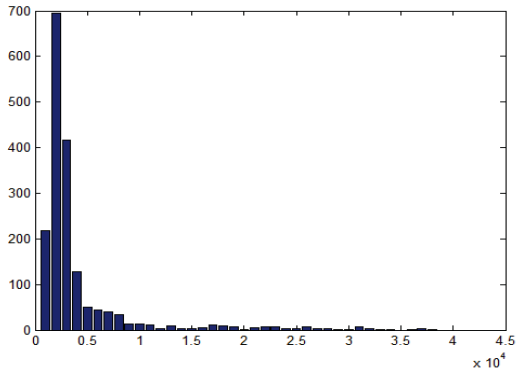


Figure 3. The histogram of disocclusion constant of the disocclusion regions, detected by the aforementioned disocclusion detection method.

For the holes originated from disocclusion, it is proposed to be filled always using the neighboring pixels that belong to the background. After the disocclusion regions are detected, we simply filled the disocclusion interval between p' and q' using the depth value of p' as shown in Figure 1(d).

4. CAMERA CALIBRATION

Since the proposed system consists of two RGB-D cameras, it is essential to identify the camera parameters through the camera calibration process. Our target is to derive the projection matrix from the left camera to the right camera and vice versa. As shown in Figure 4, the Word Coordinate System (WCS) is related to the Camera Coordinate System (CCS) by a rotation matrix **R** and a translation matrix **t**. Hence, a point Pw in the WCS can be represented as a point P_{cam} in the CCS as follows [8],

$$P_{cam} = \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix} = [\mathbf{R}|\mathbf{t}] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}. \qquad (8)$$

If we denote the camera's intrinsic parameters matrix as **A**, then P_{w} will be projected onto a pixel P on the camera plane and its coordinate can be derived as

$$P = \mathbf{A}P_{cam} = \mathbf{A}[\mathbf{R}|\mathbf{t}]P_w = \mathbf{M}_{4\times4}P_w \qquad (9)$$

Note that traditionally, the image coordinate system is two-dimensional on the image plane and the color camera calibration problem has been thoroughly studied in the literatures [2, 3]. However,

to include the depth information in P, the image coordinate system is three-dimensional in this work. Therefore, $\mathbf{M}$ is a 4×4 mapping matrix that projects a point in the WCS to a point on the image plane.

Assume P_a and P_b are coordinates of a point P in WCS projected to camera a and camera b, respectively. According to Eq. (9), P_a and P_b can be expressed as follows

$$\begin{cases} P_a = \mathbf{M}_a P & (10) \\ P_b = \mathbf{M}_b P & (11) \end{cases}$$

where $\mathbf{M}_a$ and $\mathbf{M}_b$ are the mapping matrix of camera a and camera b, respectively. To obtain the relationship between P_a and P_b, P can be calculated by using the inverse matrix method as follows

$$P = \mathbf{M}_b^{-1} P_b \quad (12)$$

Then, we obtain the relationship between P_a and P_b by substituting Eq. (12) into Eq. (10) as follows,

$$P_a = \mathbf{M}_a \mathbf{M}_b^{-1} P_b \quad (13)$$

Equation (13) can be simplified as

$$P_a = \mathbf{H} P_b \quad (14)$$

where $\mathbf{H} = \mathbf{M}_a \mathbf{M}_b^{-1}$ is the projection matrix which project pixels in image coordinate of camera b to that of camera a.

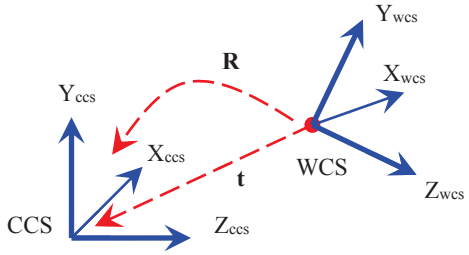


Figure 4. Extrinsic parameters of a camera.

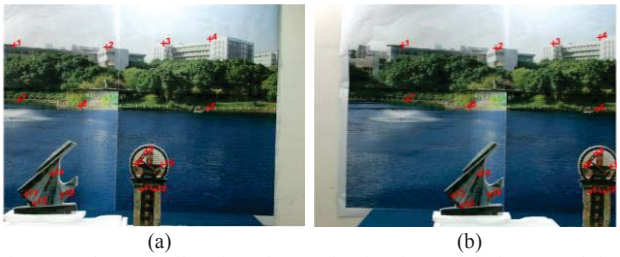
To compute the projection matrix $\mathbf{H}$, we may need N control points that specify the corresponding point pairs between camera a and camera b. Then, Eq. (14) can be derived as

$$[P_{a1} \quad \dots \quad P_{aN}] = \mathbf{H}[P_{b1} \quad \dots \quad P_{bN}] \Longrightarrow \mathbf{P_a} = \mathbf{H}\mathbf{P_b} \quad (15)$$

Finally, the least-squares (pseudo inverse) method can be used to calculate an optimal (in MSE) $\mathbf{H}'$ as follows.

$$\mathbf{H}' = \mathbf{P_a}\mathbf{P_b}^T\left(\mathbf{P_b}\mathbf{P_b}^T\right)^{-1} \quad (16)$$

The corners of checkerboard patterns are often used as the control points due to its high contrast and precise location. However, to quickly verify the proposed method, we manually pick up 16 control points on the corresponding positions of left and right images as shown in Figure 5. Then, the coordinates of each control point and the depth value of these points are combined to form P_{a1} to P_{a16} and P_{b1} to P_{b16}. Finally, the $\mathbf{H}'$ and $\mathbf{H}'^{-1}$ can be computed by using Eq. (16).



(a) (b)

Figure 5. The manually selected control points in (a) right image and (b) left image, respectively.

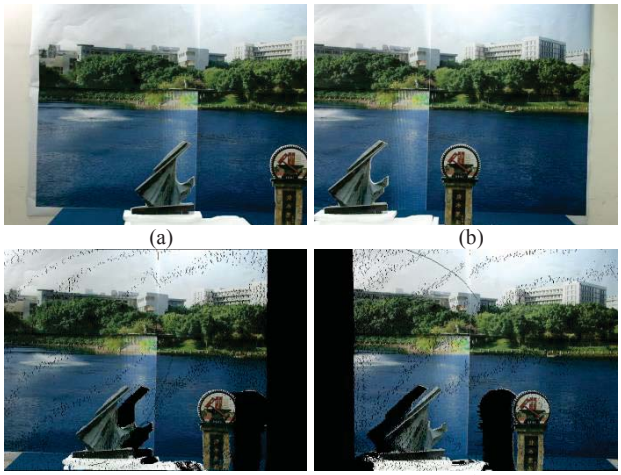
5. FORWARD WARPING AND BLENDING

We now consider a virtual view synthesized only between two reference views. Hence, we can project both left and right reference image onto an intermediate view by multiplying the translation elements (t_x, t_y, and t_z shown in Figure 6) in the matrix $\mathbf{H}'$ and $\mathbf{H}'^{-1}$ by a fraction number in the range of [0, 1]. For example, to project both the left and right images onto the middle point of two camera views, we can multiply the translation elements in matrices $\mathbf{H}'$ and $\mathbf{H}'^{-1}$ by 0.5 and the projection results are shown in Figure 7(c) and (d), respectively.

$$\begin{bmatrix} X_a \\ Y_a \\ Z_a \\ 1 \end{bmatrix} = \begin{bmatrix} & \mathbf{R} & & t_x \\ & & & t_y \\ & & & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_b \\ Y_b \\ Z_b \\ 1 \end{bmatrix}$$

Figure 6. The translation elements in projection matrix $\mathbf{H}'$ and $\mathbf{H}'^{-1}$.

To blend pixels from different reference views being warped to the same position, one option is to average the two synthesized pixels derived based on two reference views. But there are cases that one view has holes (noise, occlusion, etc.) or the associated depth values are not consistent. The more sophisticated algorithm is needed to produce better visual quality. Due to the page limit, the details are not discussed here. In this work, we simply filled the holes in one synthesized image with the pixel in the other.



(a) (b)

(c) (d)

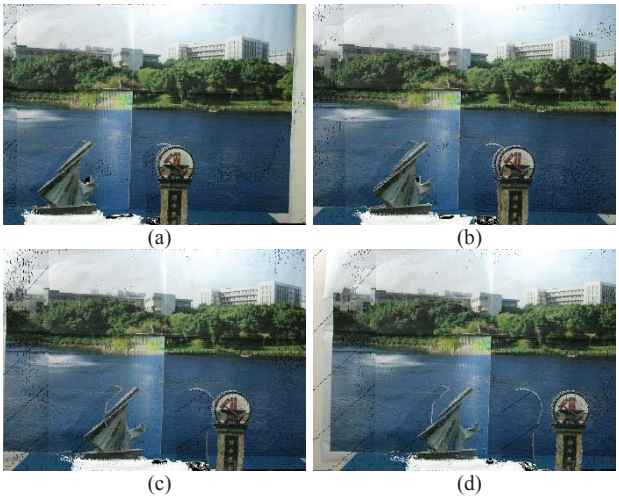
Figure 7. The forward warping results. (a) and (b) The original images captured by left and right cameras, respectively. (c) The projected image which is warped from the left image. (d) The projected image which is warped from the right image. Holes are due to occlusion and noise.

6. RESULTS AND DISCUSSIONS

To synthesize a series of intermediate virtual views, we multiply the translation elements in the matrices $\mathbf{H}'$ and $\mathbf{H}'^{-1}$ by 0.2, 0.4, 0.6, and 0.8. Then, the projected left and right images are blended into these 4 views as shown in Figure 8(a)–(d), respectively.

The final synthesized results are generally satisfactory. However, the produced image quality can be further improved by the following methods. 1) The depth map contains rather strong noise and missing pixels, which lead to wrong decisions in disocclusion detection and amendment. 2) The disocclusion filling result of the warped depth map can be improved by a more sophisticated method such as inpainting [9]. 3) For camera calibration, the manually selected control point coordinates may be inaccurate and the conventional checkerboard calibration with automatic corner detection may produce more accurate transformation matrices. 4) The color transfer method [10] can be adopted in the blending

process to reduce the color difference between two reference images. 5) The Laplacian pyramid method [11] can be adopted in the blending process to smooth the boundary between the original regions and filled regions. Furthermore, Tian *et al*. [12] have mentioned that the pin-hole errors can be commonly observed along the depth boundaries. Such errors are mainly caused by the fact that there are not sufficient sampling points in the color image and depth image along the depth discontinuities in the reference view, so the pin-hole errors may still exist even with perfect depth maps. However, splatting [13] is a well-known technique to reduce such problems. Some of the above-mentioned methods are in progress and will be reported shortly.



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

Figure 8. Synthesized images, (a)–(d). They are projected by multiplying the translation elements in the projection matrix by 0.2, 0.4, 0.6, and 0.8, respectively.

7. CONCLUSIONS

In this work, the virtual view images are synthesized based on two RGB-D images captured by two Kinect v2 cameras. A series of processes are adopted and designed to accomplish the view synthesis goal. These steps include depth to color warping, disocclusion filling, camera calibration, forward warping, and image blending. Particularly, our contributions are on the disocclusion region detection/compensation and RGB-D camera calibration. The falsely detected disocclusion regions are often incorrectly filled, and thus produce significant visual artifacts. Hence, an improved disocclusion detection method is proposed to reduce these false detections. In our experiment, most falsely detected disocclusion regions can be effectively removed by the proposed method. Moreover, we modify the traditional two-dimensional RGB camera calibration method so that it can be used to calibrate the RGB-D camera. Our method not only performs the color to color camera calibration but also does the depth to color calibration. We demonstrate that the simple RGB-D cameras can be used to build a virtual view system and good quality intermediate can be synthesized.

8. ACKNOWLEDGEMENT

This work was supported in part by the MOST, Taiwan un-der Grant MOST 104-2221-E-009 -069 -MY3 and by the Aim for the Top University Project of National Chiao Tung University, Taiwan.

9. REFERENCES

[1] C. Fehn, “Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV,” 2004, pp. 93-104.

[2] R. Y. Tsai, “A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses,” *IEEE Journal of Robotics and Automation,* vol. 3, no. 4, pp. 323-344, 1987.

[3] Z. Zhengyou, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 22, no. 11, pp. 1330-1334, 2000.

[4] Z. Cha, and Z. Zhengyou, “Calibration between depth and color sensors for commodity depth cameras,” in *IEEE International Conference on Multimedia and Expo (ICME)*, 2011, pp. 1-6.

[5] Y. Chao, T. Tillo, Z. Yao, X. Jimin, B. Huihui, and L. Chunyu, “Depth Map Driven Hole Filling Algorithm Exploiting Temporal Correlation Information,” *IEEE Transactions on Broadcasting,* vol. 60, no. 2, pp. 394-404, 2014.

[6] "MPEG-3DV View Synthesis Reference Software," ftp://ftp.merl.com/pub/avetro/3dv-cfp/software/VSRS_software.zip.

[7] V. Jantet, C. Guillemot, and L. Morin, “Joint projection filling method for occlusion handling in Depth-Image-Based Rendering,” *3D Research,* vol. 2, no. 4, pp. 1-13, 2011/11/12, 2011.

[8] R. Hartley, and A. Zisserman, *Multiple view geometry in computer vision*: Cambridge university press, 2003.

[9] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, “Simultaneous structure and texture image inpainting,” *IEEE Transactions on Image Processing,* vol. 12, no. 8, pp. 882-889, 2003.

[10] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, “Color transfer between images,” *IEEE Computer Graphics and Applications,* vol. 21, no. 5, pp. 34-41, 2001.

[11] P. Burt, and E. Adelson, “The Laplacian Pyramid as a Compact Image Code,” *IEEE Transactions on Communications,* vol. 31, no. 4, pp. 532-540, 1983.

[12] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila, “View synthesis techniques for 3D video,” 2009, pp. 74430T-74430T-11.

[13] J. Shade, S. Gortler, L.-w. He, and R. Szeliski, “Layered depth images,” in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998, pp. 231-242.