Depth Map Refinement for View Synthesis Using Depth Sensors and Color Image Cameras

Yi-Wen Chiou
Department of Electronics Engineering,
National Chiao-Tung University,
Hsinchu, Taiwan
s7531234s@gmail.com

Jang-Jer Tsai
Sonix Technology,
Hsinchu, Taiwan
jangjer.tsai@gmail.com

Hsueh-Ming Hang
Department of Electronics Engineering,
National Chiao-Tung University,
Hsinchu, Taiwan
hmhang@mail.nctu.edu.tw

***Abstract* — Depth map acquisition is a critical component in a virtual view 3D system. Often the depth map is estimated from multiple images captured by a camera array. Yet the estimated depth map often contains a significant amount of defects such as in texture-less area. Thus, in this study we use low-cost active devices, Microsoft Kinects, to acquire two color images and their associated depth maps, which also contain errors and artifacts. To improve the synthesized image quality, we propose an image calibration and depth map refinement procedure. First, we calibrate the color images to make them parallel, collinear, and color consistent. Second, we align the image coordinates with that of the depth maps. Third, we reduce the depth map noises and defects by image processing tools and by using the color image information. At the end, we use MPEG VSRS tool to synthesize the virtual images. Experimental results show that the synthesized images using our refined depth maps are noticeably improved in visual quality.**

Index Terms- 3DTV, Kinect, View synthesis, Depth map.

I. INTRODUCTION

3D video becomes very popular in the recent years. Current mainstream view synthesis (VS) is depth image based rendering (DIBR) [1], which generates virtual viewpoint images by using 2D color images and their associated depth maps. MPEG view synthesis reference software (VSRS) [2] is a popular DIBR-based tool.

Depth maps can be acquired by passive stereo-matching methods or active depth sensors. The former methods are well-studied; but they do not work well on texture-less regions. Comparatively, the latter sensors provide more accurate depth maps in the texture-less regions. In 2010, Microsoft released Kinect, which is equipped with an active depth sensor in addition to a color image camera. Kinect uses the structured infrared laser light and infrared camera to generate the depth map of a scene [3]. Because of its accessibility, Kinect is used to provide the depth map in this study.

However, on the slick surface or radiant objects, Kinect may fail to give correct depth values. These regions thus contain 'holes' on the depth map. Additionally, Kinect cannot offer stable, consistent and precise depth information along the object edges and results in 'noises' on the depth map. Recursive joint bilateral filter (JBF) is proposed to remove the boundary noises and fill the holes [4]. However, some defects cannot be removed by only JBF filtering. Hence, we like to design a depth map enhancement scheme.

The depth sensor and color camera on Kinect are located at different positions. We have to align the captured color image and depth map before feeding them to VSRS. The 3D warping technique, which requires the intrinsic and extrinsic parameters of depth sensor and color camera, is a direct way to align the depth map with the color images. The Microsoft SDK (system develop kit) does not provide all the necessary information for 3D warping [5]. Therefore, we develop an alignment method without using the camera parameters.

Our view synthesis platform is MPEG VSRS, which uses two images and their associated depth maps. Thus, two Kinect devices are used in our experiments. Before feeding the Kinect-acquired depth maps and color images to VSRS, we propose a processing procedure to overcome the imperfectness of the images and depth maps provided by the Kinect and the synthesized image quality is thus improved. The entire procedure contains three steps: (1) calibration and rectification of two capture images, (2) alignment between image and depth map, and (3) refinement of depth maps. The rest of this paper is organized as follows. Section II describes our image and depth acquisition system. Section III shows experimental results. And Section IV concludes this paper.

II. VIEW SYNTHESIS SYSTEM

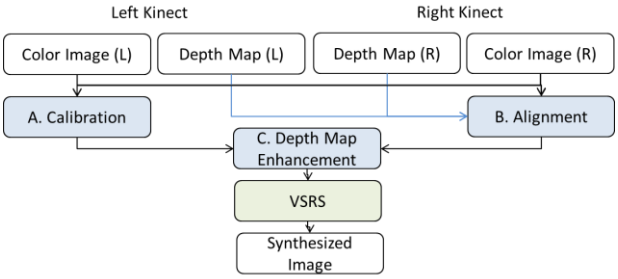


Figure 1: Proposed image and depth acquisition system for VS.

Figure 1 shows the flow chart of our entire system. Two Kinects are used to capture two color images of a scene and their associated depth maps. An image and depth processing

algorithm, which consists of calibration, alignment, and depth map enhancement, is developed to line up the color images, align the depth map with its corresponding color image and enhance the depth map quality. At the end, the MPEG VSRS is used to synthesize the virtual image.

Figure 1 shows three modules. (A) The *Calibration* module adjusts the color histograms of the two images to match their colors. Also, it rectifies two images to make them parallel and collinear. In other words, all the extrinsic parameters of the calibrated images (cameras) are the same except for the x-direction translation. (B) The Kinect SDK *NuiImageGetColorPixelCoordinateFromDepthPixel* function provides the relative coordinates of color images and depth maps but there are drifts between them. The *Alignment* module compensates for the vertical and horizontal drifts. (C) The *Depth Map Enhancement* eliminates the holes, reduces the noises in the depth map, and enhances the fidelity of depth map.

A. Calibration

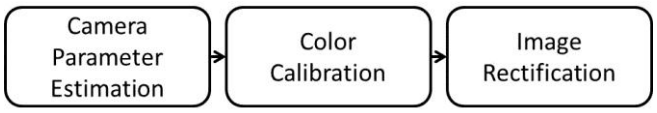


Figure 2: Calibration.

Figure 2 shows the procedure of *Calibration*. First, we *estimate camera parameters*. Camera parameters comprise the extrinsic parameters, which describe the camera location and optical axis in the 3D world coordinates, and the intrinsic parameters, which describe how a scene projects to the 2D image coordinates. Extrinsic parameters can be divided into two parts: translation matrix, which tells the location of the camera, and rotation matrix, which indicates the direction of camera optical axis.

Second, because the image camera located at different positions receive light from different angles and also may have different white balance and exposure settings, the image colors can be inconsistent; an example is shown in Figure 3. Therefore, we adjust the color histogram of one image to match the histogram of the other image [8]. The color differences in two images are thus significantly reduced (as shown in Figure 4). This color consistency is necessary for producing good synthesized images.

Third, we *rectify two color images*, so that the intrinsic parameters and the rotation matrix of the corresponding color cameras are the same. The rectification results are checked by the epipolar geometry. In Figure 5, the epipolar lines are not parallel on the original color images. In contrast, the epipolar lines in Figure 6 are parallel and collinear after rectification; that is, the rectified color cameras are collinear and parallel. Coping with the epipolar constraints reduces dramatically the computational complexity of the remaining processes in this VS system.

Figure 3: Original input color images

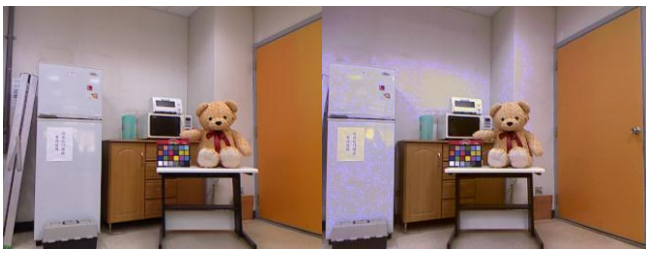


Figure 4: Color calibrated images

Figure 5: Epipolar lines before rectification

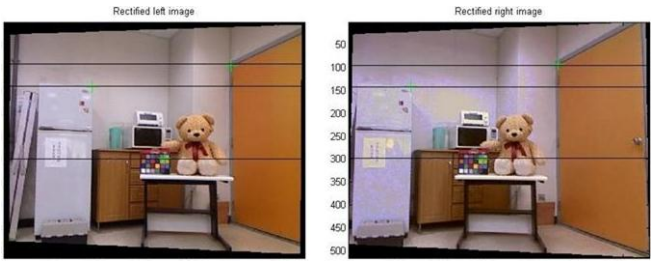


Figure 6: Rectified images with epipolar lines

B. Alignment

Figure 7 shows the original depth map and color image. By overlapping them, we find that these two pictures are unaligned. The optical axis of the depth camera and that of the color camera are somewhat off. This problem can be solved by 3D warping. However, the Microsoft Kinect SDK does not provide the infrared images, and thus the traditional camera calibration techniques cannot be used. To solve this problem, we propose an alignment procedure in Figure 8. It consists of three steps, SDK alignment, vertical drift compensation and horizontal drift compensation. To simplify the computation, only the translational compensation is considered in our scheme.

Our goal is to project the depth map from the depth sensor position C_{depth} to the color image camera position C_{color} as illustrated by Figure 9. Figure 9 shows the relationship in disparity between the depth sensor and color image camera. Herein, *f* represents focal length, *Z* is depth, *b'* is baseline, I_{depth}, $I_{virtual}$, and I_{color} are the image planes of three cameras, andC_{depth},$C_{virtual}$, and C_{color} are the centers of lens of these cameras.

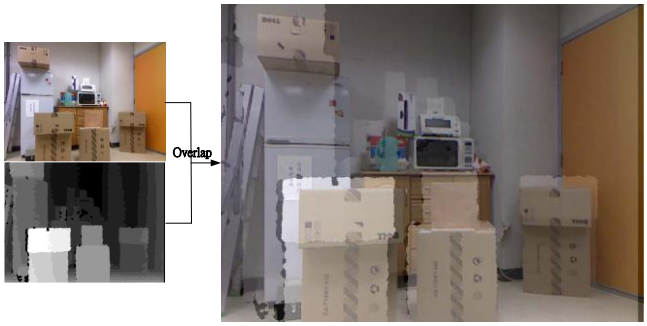


Figure 7: Drifts between color image and depth map supplied by Kinect

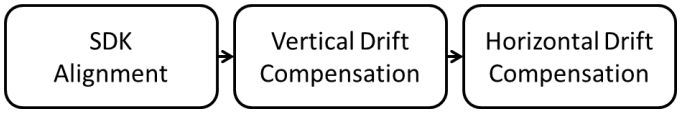


Figure 8: Alignment

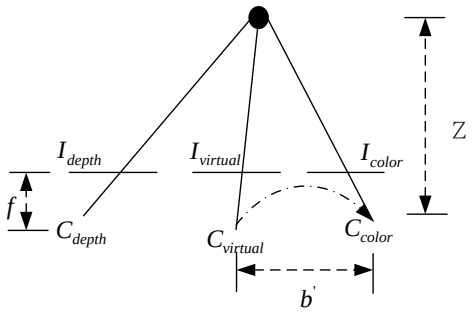


Figure 9: Disparity relationship among 3 cameras

First, the *Kinect SDK function "NuiImageGetColorPixelCoordinateFromDepthPixel"* provides the initial (relative) coordinates of depth map and color image [9]. However, there exist obvious horizontal and vertical drifts between the color image and depth map, as shown in Figure 10. Conceptually, this function projects the depth map from the depth sensor position C_{depth} to a virtual camera position $C_{virtual}$.

Second, we do *vertical drift compensation*. Because the depth map does not offer disparity information in the vertical direction, it is difficult to compensate for the virtual drift by using the depth information. We observe several images in various scenes; the vertical drift seems to be a constant. That is, the Kinect SDK function produces a constant virtual drift on the depth map. Sharp object edges seem to be able to help in drift estimation. We use two methods to estimate the vertical drift values.

Figure 10: Depth map drift aligned using the Kinect SDK returned values

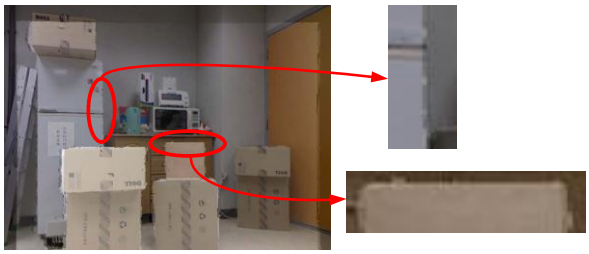


Figure 11: Aligned depth map and color image

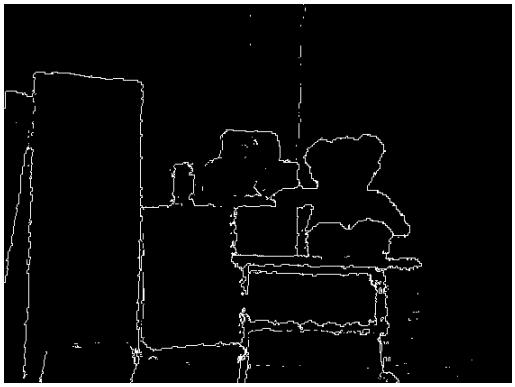


Figure 12: Depth edge map

Figure 13: Color edge map

Edge-matching estimation

We compute the depth edge map (Figure 12) from the depth maps by using the method suggested in [10]. Similarly, the color edge map (Figure 13) is computed from the color image. We then estimate the vertical drifts from the depth edge map and the color edge map. Statistics show that the vertical drift is a constant, and the value is about 3

pixels. When we move the entire depth map 3-pixel upward, the vertical drift mostly disappears.

Shadow-matching estimation

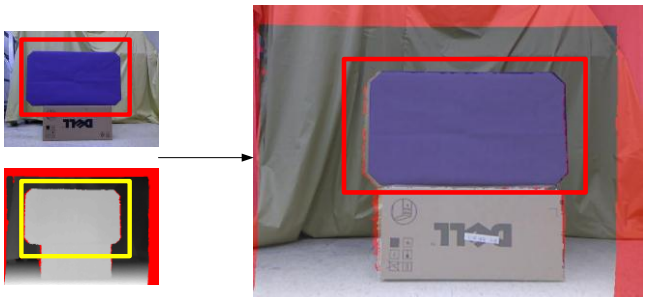


Figure 14: Shadow-based estimation method

When we look at an object in the color image, the same object in the depth map may be viewed as its shadow. Using Eq. (1), we compute the *Matching_level* between an object and its shadow. As exemplified in Figure 14, we compute the *Matching_level* of the blue board. N_{total_pixel} denotes the total number of pixels on the blue board. We manually decide the depth (value) range of the blue board. For all pixels on the blue board, we count the number of pixels, $N_{correct\_pixel}$, of which the depth value is within our decided depth range.

$$Matching\_level = \frac{N_{correct\_pixel}}{N_{total\_pixel}} \quad (1)$$

By moving the entire image upward by various amounts (in the unit of pixel), we obtain their corresponding *Matching_levels*. Table 1 shows that the *Matching_level* is the highest, when we move the depth map 3-pixel upward.

Table 1: Shadow-based estimation results

Pixel upward	Matching-level
1	0.9866
2	0.9894
3	0.9904
4	0.9887
5	0.9854

Third, we do horizontal drift compensation. In Table 2, we measure the depth, disparity and focal length to compute the baseline between the virtual camera and the color image camera. Referring to Figure 9 and Eq. (2), we project the depth map from the virtual camera position $C_{virtual}$ to a color image camera position C_{color}.

$$D = \frac{f \cdot b}{Z} \quad (2)$$

In Eq. (2), *D* represents the disparity.

Finally, the aligned depth map and color image are shown in Figure 11. Note that the depth map and color image match well without horizontal and vertical drifts.

C. Depth Map Enhancement

Figure 15 shows the proposed depth map enhancement procedure. First, we reduce random noises by a joint bilateral filter (JBF) and fill up holes by applying another JBF recursively [4]. The JBF uses the pixel distance and the R/G/B intensities in its computation. Then, the color image helps in reducing defects along object edges. At last, the median filtering is used to remove the pepper-and-salt noises.

Table 2. Baseline estimation

Distance(mm)	Disparity(pixel)	Baseline(mm)
900	7	12.16
1050	6	12.16
1200	5	11.54
1350	4	10.38
1500	4	11.54
Average Baseline Length : 11.56(mm)		

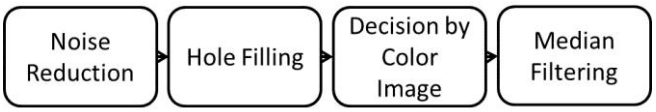


Figure 15: Depth map enhancement procedure

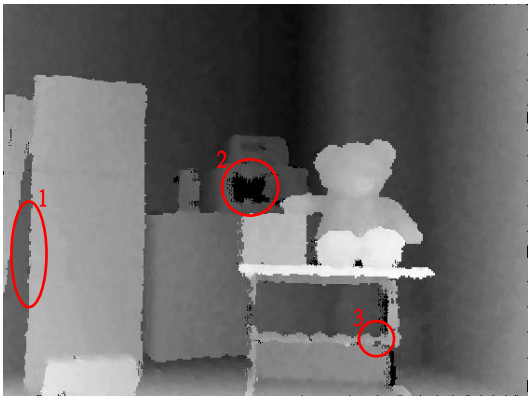


Figure 16: Depth map with defects

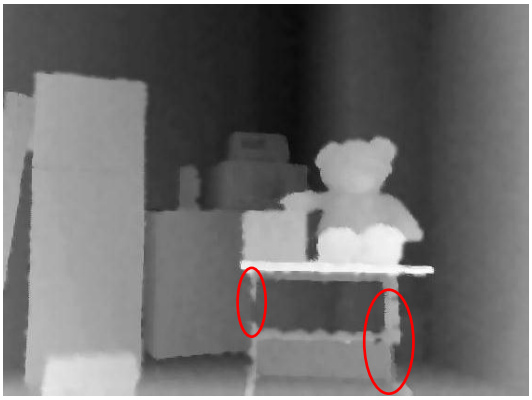


Figure 17: Depth map after recursive JBF (NR and HF)

Figure 16 shows a typical depth map with defects. After noise reduction (NR) and hole filling (HF), we obtain Figure 17. Clearly, all holes have been filled up and the depth edge is improved. However, some defects (the red ellipses in Figure 17) on the depth map are not removed although JBF

improves the depth map quality. These defects are further reduced by *decision by color image.*

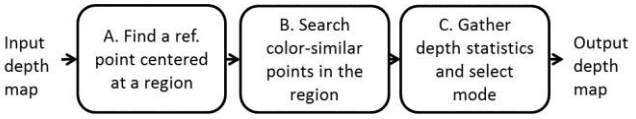


Figure 18: Procedure of decision by color image

Figure 18 shows the *decision by color image* procedure, which includes three steps.

A. We first identify "boundary" depth map pixels by a one-dimensional vertical search. If a depth value is significantly different from its vertical neighbors and it is identified as a reference (ref.) point. Typically, a ref. point locates at object boundary. We then specify a rectangular *search region* centered around a ref. point on the depth map. We try to correct the depth value of this pref. point as shown in Fig. 19(a).
B. The *search region* on the color image is the collocated region corresponding to the depth map search region. Use the color image value of the ref. point as the reference; pick up the image pixels in the search region with similar colors as shown in Fig. 19(b). The similar color pixel locations form a *decision region*.
C. Back to the depth map and collect the depth values in the decision region. The depth value of the ref. point is changed to the majority in the decision region.

Similarly, we identify the boundary depth pixels by horizontal searches. The following steps are identical to the above. This procedure is sensitive to the number of color-similar pixels in the collocated region. Therefore, the region size has a strong impact on the output image quality. We use 19×19 as the region size in this study based on experiments.

At the end, there still exists pepper-and-salt noise on the object edges of the depth map, which is due to small number of color-similar pixels. We use the median filter (5×5) to remove these noises.

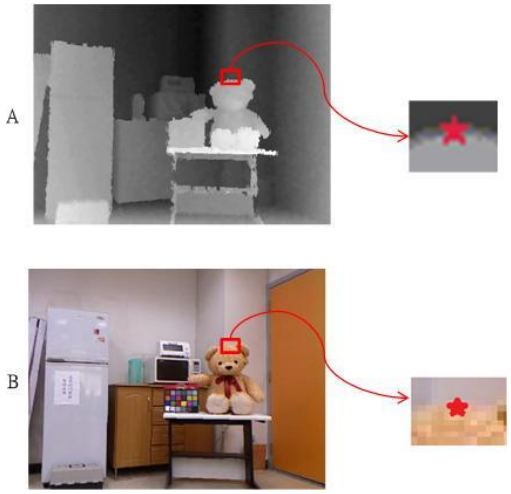


Figure 19: Illustration of decision by color

III. EXPRIMENTAL RESULTS

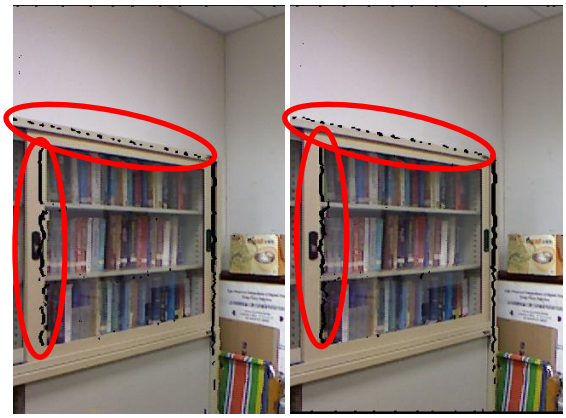
The depth values that Kinect can detect range from 80cm to 400cm. Kinect does not show depth information beyond this range. The objects outside this range are likely 'holes' on the depth map. Therefore, we set up our test scenes within this range.

Limited by space, we show two experiments in this paper. First, what is the impact of depth map alignment? Second, how does our proposed depth refinement affect the quality of synthesized image? We use the MPEG VSRS to synthesize the middle view, and the resolutions of depth map and color image both are both 640x480.

First, we use the view generation [6] to present the impacts of the *alignment*, and we project the color image to the 9.6 cm right side of color camera. Figure 20(a) shows the virtual view by using the original depth map. Figure 20 (b) shows the synthesized image by using aligned depth map. It is clear that the occlusion regions in Fig. 20(a) are incorrectly located (not on the object boundary). The occlusion regions in Figure 20(b) are aligned with the object boundaries. Therefore, the *alignment* procedure has significant impact to the synthesized image quality.

Second, we show the synthesized images by using the original and the refined depth maps. The first comparison is Figure 21 and Figure 22. Figure 23 shows the enlarged portion of Figure 21 and Figure 22. The table legs are broken in Figure 23(a). In contrast, Figure 23(b) presents a better result near table legs. Figure 24 shows another enlarged portion of Figure 21 and Figure 22. Figure 24(a) has fragments on the left and right side of desk top. In comparison, Figure 24(b) shows a better result.

Figure 25 and Figure 26 are the second comparison. Figure 27 shows enlarged portion of Figure 25 and Figure 26. The right hand and the feet of the bear are noisy in Figure 27(a). In contrast, the picture quality improves in Figure 27(b). In Figure 28, the bear ears are closer to the captured image in Figure 28(b) than in Figure 28(a).



(a) using the orginal depth map, (b) using the aligned depth map

Figure 20: Impact of aligned depth maps on synthesized images

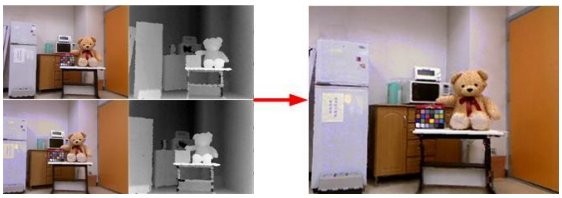


Figure 21: View synthesis by using the original depth map

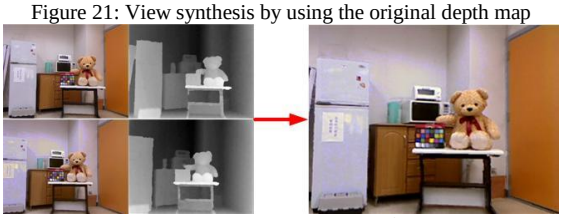


Figure 22: View synthesis by using the refined depth map

(a) using original depth map (b) using refined depth map

Figure 23: Enlarged synthesized images

(a) using the original depth map (b) using the refined depth map

Figure 24: Enlarged synthesized images

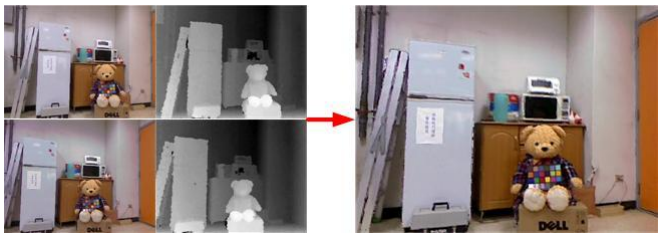


Figure 25: View synthesis by using the original depth map

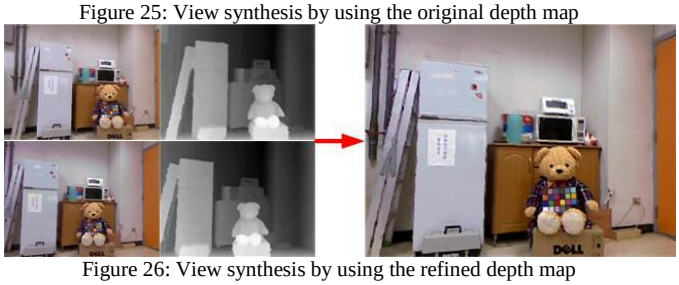


Figure 26: View synthesis by using the refined depth map

Acknowledgment: This work was supported in part by the NSC, Taiwan under Grants 98-2221-E-009-087-MY3 and by the Aim for the Top University Project of National Chiao Tung University, Taiwan.

(a) using the original depth map (b) using the refined depth map

Figure 27: Enlarged synthesized images

(a) using the original depth map (b) using the refined depth map

Figure 28: Enlarged synthesized images

IV. CONCLUSION

In this paper, we propose an image calibration and depth map refinement procedure to enhance the depth map quality for view synthesis. Two main contributions are *alignment* and *depth map enhancement*. In the alignment procedure, we align the depth map and the color image by analyzing the image drifts. In the *depth map enhancement* procedure, we combine JBF, HF, decision by color image, and median filter to remove the depth map defects. In this study, we set up a view synthesis system using Kinect to acquire color images and depth maps, we then improve the depth maps, and achieve better synthesized images.

V. REFERENCES

[1] C. Fehn, "Depth-Image-Based Rendering (DIBR), compression and transmission for a new approach on 3D-TV," *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems XI*, pages 93 - 105, San Jose, CA, USA, Jan. 2004.

[2] ISO/IEC JTC1/SC29/WG11, N11631, "Report on experimental framework for 3D video coding", Oct. 2010.

[3] Microsoft, " XBOX360+Kinect, " *http://www.xbox.com/zh-TW/Kinect/*, 2012.

[4] M. Camplani and L. Salgado, "Efficient spatio temporal hole filling strategy for Kinect depth maps," *IS&T/SPIE Int. Conf. on 3D Image Processing (3DIP) and Applications*, CA, USA, pp. 82900E 1-10, Jan. 2012.

[5] J. Smisek, M. Jancosek, and T. Pajdla, "3d with Kinect," in *ICCV Workshop*, Barcelona, Spain, pp. 1154-1160, 2011.

[6] S. Lee and Y. Ho, "Real-time Stereo View Generation using Kinect Depth Camera," in Proc. *APIPA Annual Summit and Conference (APSIPA ASC)*, Xi'an, China, pp. RS3.12(1-4), Oct. 2011.

[7] A. Fusiello, E. Trucco, and A. Verri , "A compact algorithm for rectification of stereo pairs," *Machine Vision and Applications*.,vol. **12**, pp. 16–22, 2000.

[8] F. Porikli, "Inter-camera color calibration using cross-correlation model function," in *IEEE Int. Conf. on Image Processing*, vol. 2, pp. II-133-6 ,2003.

[9] Microsoft, " Kinect SDK, " *http://www.microsoft.com/ en-us/kinectforwindows/*, 2012.

[10] H. Yu, G. Yang, Y. Zhou, and Z. Jin, "Improving Depth Estimation through Fusion of Stereo and TOF," *International Conference Multimedia Technology (ICMT)*, Beijing, China, pp. 26-28, 2011.