View Synthesis for 3D Video Scene Composition

Amanda Wang, Chun-Liang Chien and Hsueh-Ming Hang

Electronics Engineering Department, National Chiao Tung University
1001 University Rd., Hsinchu 30010, Taiwan, R.O.C.
amanda.wang1289@gmail.com
jameschien@nctu.edu.tw
hmhang@mail.nctu.edu.tw

***Abstract*—Scene composition is a method widely used in movie and TV production. Merging two sets of 3D videos into one is a very challenging task. There are several main issues on video composition. Our focus is compositing two sets of videos with different camera motion parameters. The key techniques are the camera motion estimation and view synthesis technique used to produce the synthesized motion-compensated background video. We propose a refined backward warping technique for view synthesis and adopt the ICP algorithm to calculate the camera motion parameters.**

I. INTRODUCTION

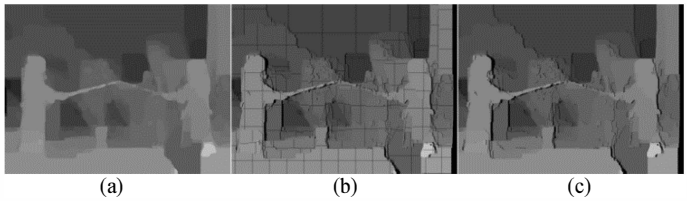
Scene composition is a method widely used in movie and TV production. The process is started by taking two different 3D video sequences acting as foreground and background, respectively. The first step is to adjust the orientation of background camera to match that of the foreground camera. The motion of the background camera should also matches the foreground camera. If both cameras are stationary, camera motion compensation will not be necessary. However, if any of the camera is not stationary, motion compensation is needed. Based on the new camera orientation and motion parameter, a new background sequence is synthesized.

The view synthesis process is used to create virtual views from the different views taken from several camera positions [1]. The view synthesis process consists of two steps: (1) warping of two nearest real views and (2) views merging. The backward warping method was considered to be better than the conventional forward warping method [2]. In the backward warping method, the position of a pixel is mapped backward from target view to the reference view. This method resolves the cracks or missing pixel value problem that may occur in forward warping.

Backward warping itself, however, is not perfect. There are some artifacts that may occur in the backward depth warping process. One of the tools that can be used to reduce the artifacts is the superpixel technology which was first introduced in [3]. Some previous studies use superpixel to refine the synthesized depth map [4] or to do hole-filling in depth warping result [5].

II. BACKWARD DEPTH WARPING

In the traditional depth warping algorithm, the forward depth warping (FDW) method is applied to the depth maps of both views to generate the virtual view depth map. This method, however, creates some cracks between pixels as shown in Fig. 1(b). In order to tackle this problem, the backward depth warping (BDW) method has been developed [6]. In BDW, we map the coordinates of pixels from the target view to the reference view; therefore, the cracks previously stated occur much less.



(a) (b) (c)

Fig. 1. Examples of depth warping. (a) The original depth. (b) The forward warping result. (c) The backward warping result.

The original forward depth warping follows the equation:

$$z_v \begin{bmatrix} u_v \\ v_v \\ 1 \end{bmatrix} = z_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} + \begin{pmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} t_x \\ 0 \\ 0 \end{bmatrix}. \quad (1)$$

From the above equation, we can derive a more general warping equation,

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

where

$$A = Q_v {Q_r}^{-1}, \qquad b = Q_v(c_r - c_v) \quad (3)$$

$$Q_r = K_r \cdot R_r, \qquad Q_v = K_v \cdot R_v \quad (4)$$

$$c_r = -{R_r}^T \cdot t_r, \qquad c_v = -{R_v}^T \cdot t_v \quad (5)$$

K_v and K_r denote the intrinsic camera parameters of virtual view and reference view, respectively. R_v and R_r denote the rotation matrix of virtual view and reference view, respectively. t_v and t_r denote the translation matrix of virtual view and reference view, respectively.

In backward depth warping, we need z_v to solve the equation. However, there is a chicken-and-egg paradox. We cannot obtained the depth value of virtual view before the warping process is completed; therefore, we need to try every possible value ranged in [0 255]. In practical situation, the

978-1-4673-7478-1/15/$31.00 ©2015 IEEE

pixel positions are floating numbers. Therefore, rounding all the values may cause matching errors. In order to fix this problem, the equation is changed to

$$z'\begin{bmatrix}u'\\v'\\1\end{bmatrix} = A^{-1}(z_v\begin{bmatrix}u_v\\v_v\\1\end{bmatrix} - b), \tag{6}$$

where *z'*, *u'*, and *v'* are floating numbers representing the derived depth value and positions of a pixel in the reference view. An error function is defined to represent the difference between these calculated positions and the actual positions in reference view.

$$E = \min_{\substack{u_r=\lfloor u'\rfloor \text{ or } \lceil u'\rceil \\ v_r=\lfloor v'\rfloor \text{ or } \lceil v'\rceil}} (u_r - u')^2 + (v_r - v')^2 + (D(u_r, v_r) - z')^2 \tag{7}$$

where $D(u_r, v_r)$ is the disparity at (u_r, v_r). Let $z' = z_v$, *u'* and *v'* can be derived based on the first minimum *E* is obtained while trying every possible z_v values in descending order. As shown in Fig. 1(c), cracks can be effectively removed with the BDW method.

III. Proposed Backward Warping Refinement Algorithm

The joint bilateral filter (JBF) [7] is a modified version of the bilateral filter [8], which uses the information in a low resolution depth image and its associated high resolution color image together to upsample the depth image. The JBF-based depth upsampling assume that the occurrences of edges between depth and color image are highly correlated [7]. The upsampled solution $\tilde{S}_p$ at the pixel *p* is obtained as:

$$\tilde{S}_p = \frac{1}{k_p}\sum_{q_\downarrow \in \Omega} S_{q_\downarrow} f(\|p_\downarrow - q_\downarrow\|) g(\|\tilde{I}_p - \tilde{I}_q\|) \tag{8}$$

where $p_\downarrow$ and $q_\downarrow$ are the pixels in a low resolution image. $S_{q_\downarrow}$ is the pixel value at $q_\downarrow$ in a low resolution depth image. $\tilde{I}_p$ and $\tilde{I}_q$ are pixel values at *p* and *q* in a high resolution color image. Ω is the neighborhood of $p_\downarrow$ and k_p is a normalizing term. *f*() and *g*() are the spatial and range filter kernel, respectively.

The JBF-based approaches were widely used to depth image upsampling [9, 10]. However, the existing methods of depth image refinement suffer from two major artifacts [11]: edge blurring and edge misalignment. The edge blurring artifacts occur when the color image has no edges, whereas the corresponding depth image has an edge. Then, the JBF behaves as a smoothing filter and lower the contrast of edges in the depth image. To reduce this type of artifacts, in addition to the spatial and color range filter kernel, we include the depth range kernel. Furthermore, instead of JBF, the superpixel method is adopted in this work.

Our proposed warping refinement algorithm uses superpixel technique to fix nonocclusion holes that occur during the depth warping process. First, we generate superpixels from the reference image and do the backward depth warping. When performing the depth warping, we label each pixel in target view with its corresponding superpixel number. We check the warping result for nonocclusion region and fill in the region with the average depth value of the corresponding superpixel. The flowchart of the proposed algorithm is shown in Fig. 2.

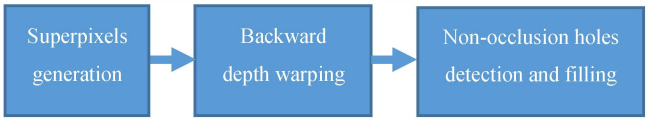


Fig. 2. Flowchart of proposed backward warping refinement

A. Superpixels Generation

In this method, we use SLIC superpixel described in [12]. In the original SLIC superpixel algorithm, the distance measure used is defined as:

$$\begin{cases} d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \\ d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \\ D_s = d_{lab} + \dfrac{m}{S} d_{xy} \end{cases} \tag{9}$$

Similar to the JBF, the superpixel method also include a range distance d_{lab} and a spatial distance d_{xy}. In the JBF method, both distance measures are adopted to decide the strength of smoothing. On the other hand, in the superpixel method, both distances are counted to decide whether two pixels belong to the same segment or not. Hence, edges will be better preserved if we consider the superpixel segmentation outputs. Thus, in our method, we use superpixel to refine depth warping result. Hence, we want to include the depth information in addition to the color information. We redefine the distance measurement D_s as:

$$\begin{cases} d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \\ d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \\ d_{depth} = \sqrt{(z_k - z_i)^2 + (z_k - z_i)^2} \\ D_s = w_1 d_{lab} + w_2 d_{depth} + \dfrac{m}{S} d_{xy} \end{cases} \tag{10}$$

As aforementioned, in addition to the spatial and color range filter kernel, the depth range kernel is also adopted to reduce the edge smoothing artifact. We further adjust the weighting of our superpixel by simplifying the distance measurement as

$$D_s = w_1 d_{lab} + w_2 d_{depth} + w_3 d_{xy} \tag{11}$$

where the weight w_1 and w_2 are adjusted to satisfy $w_1 < w_2$ and w_3 defines the compactness of the superpixels.

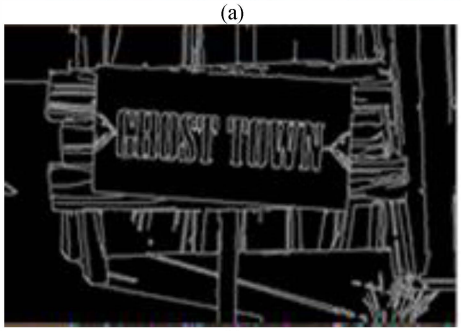
Adjusting the weighting variables may, however, cause some problems. In some cases, parts of image with similar or same depth value which belong to different objects may be segmented into a single superpixel as shown in Fig. 3. Due to the purpose of superpixels in this paper, this situation is not

acceptable. One superpixel should not belong to more than one object.

Fig. 3. Example of problem caused by weighting adjustment on superpixels

In order to tackle this problem, the edge information is also used. First, an edge detection algorithm is applied to the image to obtain the edge information as shown in Fig. 4(b). Upon getting the edge information, it is then overlapped with the superpixel contours as shown in Fig. 4(c). The resulted contour is the union of edge contour and the original superpixel contour. It becomes the new superpixel segmentation result.

(a)



(b)

(c)

Fig. 4. Combining clustered superpixel in (a) with (b) edge information to get the (c) final superpixel segmentation

B. *Backward Depth Warping*

We use the backward depth warping method as described in Section II. However, in addition to depth value mapping, we also map the superpixel number of each pixel from reference view to target view. The superpixel number is what we used in the process of filling up the depth value of nonocclusion holes.

C. *Nonocclusion Holes Detection and Filling*

There are two types of holes that may occur during the depth warping process: (1) occlusion holes and (2) nonocclusion holes. We want to reduce the number of nonocclusion holes since the pixels within the holes actually belong to certain objects. We define nonocclusion holes as the holes which are surrounded by pixels inside a superpixel.

1	1	1	1
1	0	1	1
1	0	0	1
1	1	1	1

(a)

1	1	1	2
1	1	0	2
1	1	0	2
1	1	1	1

(b)

1	1	0	2
1	1	0	2
1	0	0	2
1	1	0	2

(c)

Fig. 5 Examples of types of holes where (a) and (b) shows non-occlusion holes while (c) shows occlusion hole

As an example shown in Fig. 5(a), a hole is surrounded by pixels inside a single superpixel. However, in some cases the hole may be surrounded by more than one superpixes. In the case shown in Fig. 5(b), we define the hole to be a part of a superpixel if at least 70% of the surrounding pixels belong to the same superpixel. The hole in Fig. 5(c) is surrounded by pixels not from the same superpixel; therefore, it is classified as occlusion hole and no hole-filling process will be done.

The nonocclusion holes are filled with the average depth value of the corresponding superpixel. Another way of doing the hole-filling using superpixel is by applying the warping algorithm for each superpixel separately and perform inpainting as in [13] to fill in the hole at each superpixel.

IV. CAMERA MOTION COMPENSATION

Background and foreground may have different camera motion parameters; hence, we need to align them. We want to adjust the movement of the background camera to follow that of the foreground. In order to do so, we need to first extract the camera motion parameters of both cameras.

A. *Camera Motion Estimation Using Iterative Closest Point Algorithm*

Our proposed method uses the Iterative Closest Point (ICP) Algorithm [14] to estimate the camera motion parameter. ICP is a well-known technique in computer graphics. The basic idea of ICP is to get 2 sets of point clouds as the input, one acts as target and the other one as the source cloud. The algorithm applies a transformation on the source cloud in order to match the target cloud. The resulting transformation matrix can be considered the camera motion parameter.

B. *ICP Point Cloud Generation*

In order to generate 2 sets of point clouds, we take 2 consecutive frames from the video sequence and select N points from each frame. After getting the N points, we project the 2D points onto real world coordinate (X, Y, Z). Our point selection starts at the center of an image. We iteratively take the points with an interval I from the center moving outward. The point selection method is shown in Fig. 6.

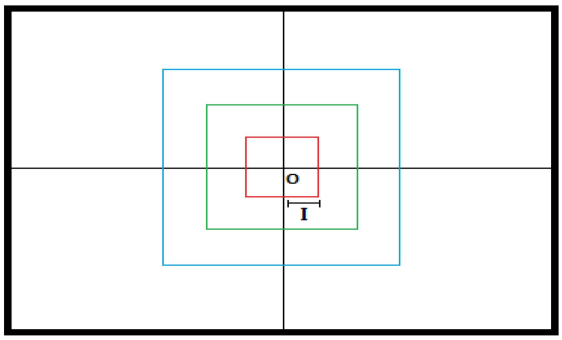


Fig. 6 Point selection for ICP Input

C. *ICP Optimization*

Camera motion estimation needs to be done throughout the whole video sequence. In order to reduce the processing time, some acceleration technique is used. Let T be the transformation matrix and T_i be transformation from frame i to frame $(i + 1)$. Assuming the camera transition between frames is consistent, we can assume that

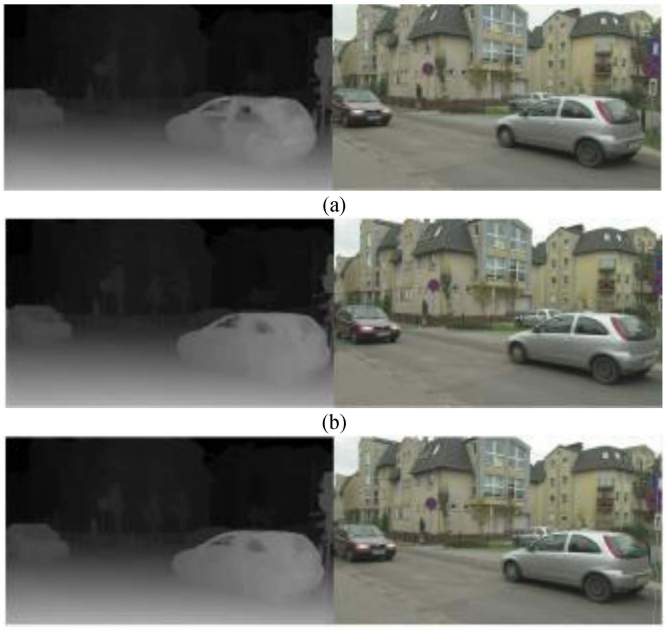
$$T_{i+1} = T_i + \Delta \tag{12}$$

In the original ICP algorithm, T is always initialized as 0. However, in our case, we use T_i as the initial transformation matrix for frame $(i + 1)$ to $(i + 2)$. This approach reduces the number of iterations needed in the ICP process.

V. Experimental Results

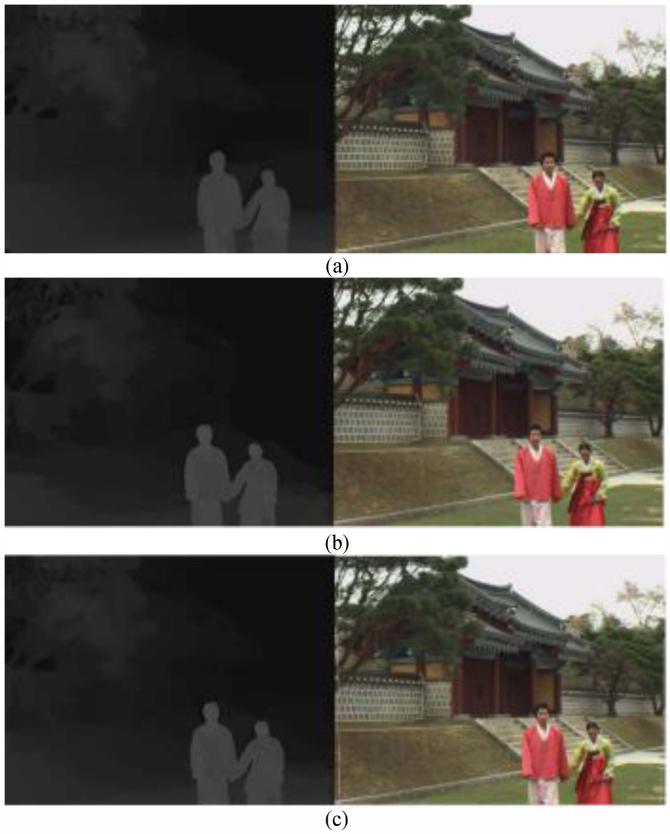
We test two MPEG sample sequences, Poznan and Lovebird, with the proposed algorithm. As shown in Fig. 7(a) and (b), the depth maps are not perfect because they are estimated based on the disparity between the left and right views. If we synthesize a virtual view in the middle of left and right views with the original depth map, many artifacts appear in the synthesized images. On the other hand, after the depth map was refined using the proposed method, the artifacts which caused by nonocclusion holes are greatly reduced and the synthesized result is much better as shown in Fig. 7(c). Another example was shown in Fig. 8.

We also compare the results of backward depth warping before and after applying the proposed refinement method. Fig. 9 is the comparison between backward depth warping with and without refinement. We observe that the quality of the depth map can be improved by using the superpixel technique to fill in the detected nonocclusion holes.



(a) (b) (c)

Fig. 7. Results of applying the proposed method on PoznanStreet sequence: (a) left reference view, (b) right reference view, (c) synthesized view



(a) (b) (c)

Fig. 8 Results of applying proposed method on Lovebird sequence: (a) left reference view, (b) right reference view, (c) synthesized view

(a) (b) (c)

Fig. 9. Comparison between (a) reference depth map, (b) backward depth warping result, (c) backward depth warping with superpixel refinement result

Finaly, we test the scene composition with our motion compensation method on the PoznanHall sequence and Lovebird sequence. The couple in Lovebird sequence acts as the foreground and PoznanHall sequence acts as the background. The background sequence which has camera motion to the left while the foreground camera is stationary. Therefore, we can test our motion compensation method. We synthesize each frame on background sequence to match the stationary camera of the foreground. Without motion compensation, the relative movement bewteen the foreground and background is mismatched, and thus the composition result is unreal as shown in Fig. 10(a). On the other hand, if the motion of background is compensated with the proposed ICP method, the relative movement between the foreground and background is matched and the composition result is more real as shown in Fig. 10(b).

(a) (b)

Fig. 10 Scene composition results. (a) Example of scene composition without motion compensation; notice that the background camera moves to the left (b) Scene composition result after applying camera motion compensation; notice that the background camera is stationary.

VI. CONCLUSIONS

The backward depth warping method can significantly reduce artifacts produced in the forward depth warping method. However, some artifacts may still occur in the process. The proposed algorithm uses a newly proposed popular superpixel as a tool to improve the depth warping result. Often, the quality of synthesized view depends on the quality of the synthesized depth map. By minimizing the artifacts in depth warping result, we can produce a better quality image.

ICP algorithm was employed to estimate camera motion parameter of video sequences. In the process of applying ICP on consecutive frames, we can reduce the number of iterations by setting the previous transformation as the starting transformation of the current frame.

ACKNOWLEDGEMENT

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

REFERENCES

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

[2] R. C. Gonzalez, *Digital image processing*: Pearson Education India, 2009.

[3] R. Xiaofeng, and J. Malik, “Learning a classification model for segmentation,” in Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, 2003, pp. 10-17 vol.1.

[4] G. Chaurasia, S. Duchene, O. Sorkine-Hornung, and G. Drettakis, “Depth synthesis and local warps for plausible image-based navigation,” *ACM Trans. Graph.*, vol. 32, no. 3, pp. 1-12, 2013.

[5] T. Tezuka, K. Takahashi, and T. Fujii, “Superpixel-based 3D warping using view plus depth data from multiple viewpoints,” 2014, pp. 90111V-90111V-8.

[6] D.-H. Li, H.-M. Hang, and Y.-L. Liu, “Virtual view synthesis using backward depth warping algorithm,” in Picture Coding Symposium (PCS), San Jose, USA, 2013, pp. 205-208.

[7] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, “Joint bilateral upsampling,” *ACM Trans. Graph.*, vol. 26, no. 3, pp. 96, 2007.

[8] C. Tomasi, and R. Manduchi, "Bilateral filtering for gray and color images." *Computer Vision, 1998. Sixth International Conference on*, pp. 839-846.

[9] D. Chan, H. Buisman, C. Theobalt, and S. Thrun, “A Noise-Aware Filter for Real-Time Depth Upsampling,” in Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications - M2SFA2 2008, Marseille, France, 2008.

[10] C. Jinwook, M. Dongbo, H. Bumsub, and S. Kwanghoon, "Spatial and temporal up-conversion technique for depth video." *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pp. 3525-3528.

[11] Y. Soh, J.-Y. Sim, C.-S. Kim, and S.-U. Lee, "Superpixel-based depth image super-resolution." pp. 82900D-82900D-10.

[12] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, *SLIC Superpixels*, No. EPFL-REPORT-149300, 2010.

[13] 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.

[14] P. J. Besl, and N. D. McKay, “A method for registration of 3-D shapes,” 1992, pp. 586-606.