Virtual View Synthesis Using Backward Depth Warping Algorithm

Du-Hsiu Li Hsueh-Ming Hang Yu-Lun Liu
Department of Electronics Engineering,
National Chiao-Tung University, Hsinchu, Taiwan
doom8199@commlab.tw, hmhang@mail.nctu.edu.tw, alex04072000@hotmail.com

***Abstract*—The virtual view synthesis reference software offered by the MPEG standard committee adopts the forward warping technique in projecting the depth map from the reference view to the target (virtual) view location. Often, this warping process results in many artifacts, holes and cracks, due to quantization errors and occlusion. In this study, we propose a backward warping process to replace the forward warping process, and the artifacts (particularly the ones produced by quantization) are significantly reduced. The subjective quality of the synthesized virtual view images is thus much improved.**

I. Introduction

In recent years, 3D video technology advances very fast and its applications are becoming more popular [1][2]. The MPEG (ISO/IEC Moving Picture Expert Group) committee kicked off a 3DAV (3D audio-visual) [3] standardization work item a few years ago. This on-going activity is going to define the virtual-view (free-viewpoint) video formats and their associated operations.

There are several elements in a complete virtual-view video system, such as camera calibration, depth estimation, multi-view video coding, virtual-view view synthesis [4], and 2D/3D multiview display. This study focuses only on the technique inside the view-synthesis component.

Roughly, the image synthesis techniques are classified into two categories: model-based rendering (MBR), and image-based rendering (IBR) [5,6]. The IBR technique is adopted by the MPEG/ITU committee (JVC-3V) for new view synthesis in the standard, and thus it becomes our focus in this study. In the depth image-based rendering (DIBR) scheme adopted by the JVC-3V group needs the depth information. The left-view and right-view images together with their depth maps are assumed available. The target (virtual) viewpoint image is synthesized in three steps. In the first step, the depth map at the target viewpoint is created based on the left-view depth map using the *forward warping* technique. Similarly, the right-view depth map is also projected to the target viewpoint using forward warping. In the second step, the left-view color image is mapped to the target viewpoint using the left-view warped depth map. Similarly, the right-view image is mapped to the target viewpoint. The final step is to combine these two mapped images into one image. The detailed procedure will be described in Section 2.

In this paper, we replace the *forward warping* technique in the first step described in the above by the *backward warping* technique. The reason for doing this is that the forward warped depth map often contains artifacts. The artifacts are becoming more serious when the left and right cameras are farther apart. However, there is no simple formula to calculate the backward warping. Because the depth map is typically quantized into 256 levels, it is thus possible to design a fast divide-and-conquer method to produce the warped depth map.

In the rest of this paper, the mathematical expressions of forward warping and its shortcomings are discussed in Section 2. In Section 3, we formulate our backward warping scheme. Section 4 shows some initial results. Some concluding remarks and future work are given in Section 5.

II. Virtual view synthesis systems

The dataflow of the DIBR virtual view synthesis procedure is drawn in Figure 1. The two input (reference) depth maps are projected, respectively, to a target (virtual) viewpoint by using the 3D warping technique. The 3D warping procedure is mainly divided into two steps. Firstly, Eq. (1) is used to project the reference view depth into the 3D world space. Secondly, Eq. (2) is used to project the 3D world space into the virtual view image plane. In Eq. (1) and Eq. (2), the matrix Q_r represents

978-1-4799-0294-1/13/$31.00 ©2013 IEEE

the *camera internal parameters*. The function of Q_r is to transform 3D camera coordinates into the image plane coordinates. The rotation matrix Q_v and the translation vector c_v consist of the *camera external parameters*. The function of the external parameters is to convert the world coordinates into the camera coordinate system.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = z_r {Q_r}^{-1} \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} + c_r \qquad (1)$$

In Eq. (1), (u, v) and (X, Y, Z) represent the image coordinates and the world coordinates, respectively, of an object point. The subscript r represents the reference view, z_r represents the reference view depth value at location (u_r, v_r).

$$z_v \begin{bmatrix} u_v \\ v_v \\ 1 \end{bmatrix} = Q_v \left(\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - c_v \right) \qquad (2)$$

In Eq. (2), the subscript v represents the virtual view.

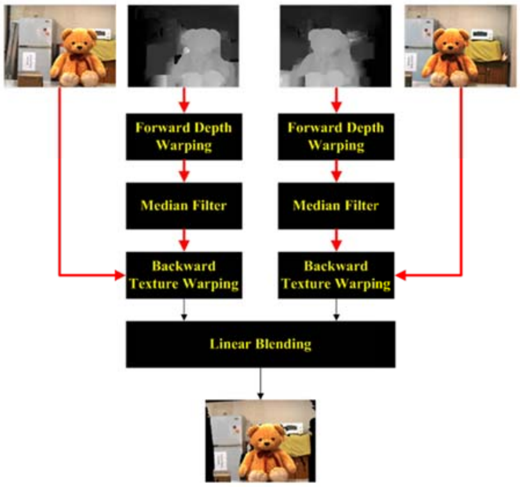


Figure 1. Virtual view synthesis.

If we merge Eq. (1) and Eq. (2), we derive the transformation between pixels on these two image planes. It becomes as follows.

$$z_v \begin{bmatrix} u_v \\ v_v \\ 1 \end{bmatrix} = z_r A \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} + b,$$

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

Assuming that these two cameras are identical and in parallel horizontally, and two camera images are rectified. In this case, only a horizontal shift (baseline) exists between the left and the right cameras. Hence, $z_v = z_r$. We adopt the ideal pinhole camera model assumption; Eq. (3) can be simplified to Eq. (4) at the same vertical coordinate:

$$u_v = u_r + d(z_r) \qquad (4)$$

wherein $d(z_r)$ is disparity. Ideally,

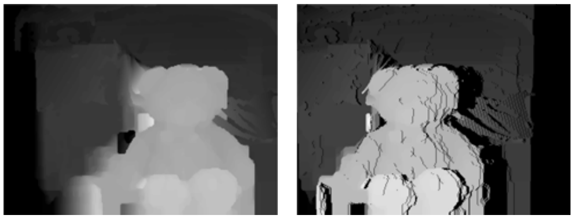
$$d(z_r) = \frac{f \cdot B}{z_r} \qquad (5)$$

where f is the camera focal length and B is the baseline of two cameras. In the actual cases, images are in the unit of pixels, which are integers. So, the disparity values are quantified (or rounded); that is, Eq.(4) is approximated by Eq.(6).

$$\widetilde{u_v} = u_r + round\{d(z_r)\} \qquad (6)$$

In the warping process, if there are two or more depth values mapped to the same virtual image location, we will select the minimum depth value to be the target view (virtual view) depth; that is, select the object point closest to the camera as the final virtual view image pixel value.

A typical forward warping on the depth map using Eq.(6) is illustrated by Fig. 2. It can be observed from Fig. 2 that after warping, the depth map contains *holes* and *cracks*. That is, the new depth value z_v at $\widetilde{u_v}$ is missing although all the possible u_r coordinates have been checked. This phenomenon is due to the depth value quantization and discontinuity (including occlusion). Because the depth value and the disparity value have a one-to-one correspondence relationship, if no distinction is needed, these two terms (disparity and depth) may be used interchangeably in the rest of text.



(a) Depth map of the left image

b) The forwardly warped depth map to a middle viewpoint

Figure 2. A depth (disparity) map warping example.

Fig.3 shows an example of mapping one line of disparity values mapped from the left camera (Cam_L) to the right camera (Cam_R). In this example, the image size is stretched from Cam_L to Cam_R. Because the depth values are quantized, they are shifted by one pixel leaving one-pixel gaps on the stretched (target) line.

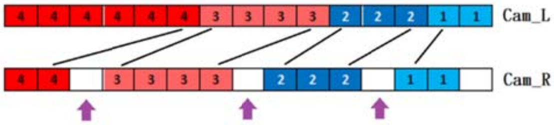


Figure 3. An example of disparity map warping errors.

III. Depth map backward warping

The forward depth warping often results in many artifacts, which can be difficult to remove. Therefore, we develop a backward depth warping algorithm to significantly reduce the small cracks in the warped depth map. The proposed procedure is shown in Fig. 4. Conceptually, we break the warped (target) depth (disparity) values into many layers. Each target depth layer is calculated using the HIP inverse mapping procedure (to be described). Then, we combine (merge) all layers to form the final (target) depth map.

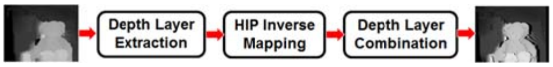


Figure 4. Backward depth warping algorithm.

According to the depth quantization formula specified by Microsoft:

$$z_q = round\left\{(N-1)\cdot\frac{z^{-1}-z_{far}^{-1}}{z_{near}^{-1}-z_{far}^{-1}}\right\} \tag{7}$$

In the above formula, z_{near} and z_{far} are respectively the minimum and the maximum depth values of the entire map, z and z_q are, respectively, the original depth value and the quantized depth value, and *N* represents the number of quantization levels (represented by $\lceil\log_2 N\rceil$ bits). Typically, *N* is 256 (=8 bits).

Use the depth quantization formula of Eq. (7) and plug it into Eq. (3) and do some simplifications, we obtain the following equation.

$$\begin{bmatrix}u_v\\v_v\\1\end{bmatrix}\sim H(z_q)\begin{bmatrix}u_r\\v_r\\1\end{bmatrix} \tag{8}$$

wherein $H(z_q)$ is

$$H(z_q) = A + [0\quad 0\quad 1]\otimes\left(\frac{z_q}{N-1}\cdot\left(z_{near}^{-1}-z_{far}^{-1}\right)+z_{far}^{-1}\right)\cdot b \tag{9}$$

In the above formula, A and b are the matrix and the vector in Eq. (3) under the assumption that the two pictures are rectified. And $\otimes$ denotes the Kronecker product. If $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{B}$ is a $p \times q$ marix, then the Kronecker product $\mathbf{A}\otimes\mathbf{B}$ is the $mp \times nq$ block matrix given below,

$$\mathbf{A}\otimes\mathbf{B} = \begin{bmatrix}a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B}\\ \vdots & \ddots & \vdots\\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B}\end{bmatrix}\ . \tag{10}$$

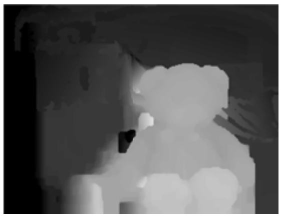
Eq. (9) defines a homography. In other words, a plane in the world coordinates is projected onto two camera images, and there exists a homography relationship between these two corresponding image patches. This transformation is called the *homography induced by a plane* (HIP) [7]. We decompose a quantized depth map into *N* planes or layers (usually, *N*=256), each plane corresponds to a homography in Eq. (9). We calculate the homography of each layer with its quantized depth z_q, separately. That is, we calculate the inverse of $H(z_q)$, $H^{-1}(z_q)$, for each layer. Then, for each target pixel coordinate (u_v, v_v), we compute its reference depth map coordinate (u_r, v_r) by using

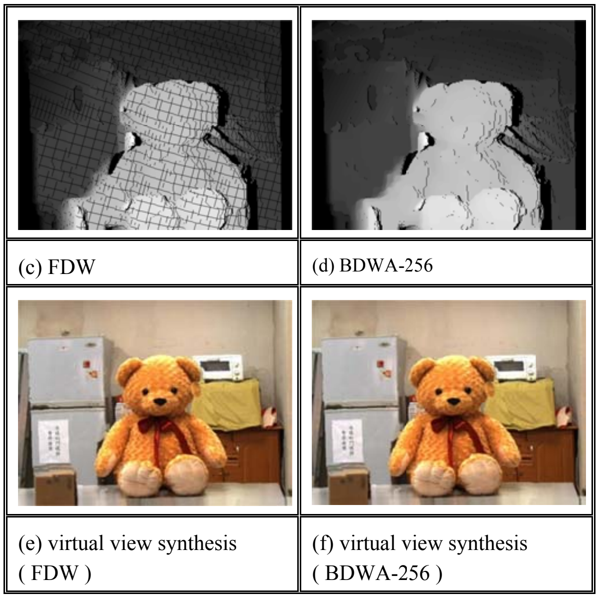
$$\begin{bmatrix}u_r\\v_r\\1\end{bmatrix}\sim H^{-1}(z_q)\begin{bmatrix}u_v\\v_v\\1\end{bmatrix}. \tag{11}$$

If the depth value in the neighborhood of (u_r, v_r) is z_q, it is assigned to be the depth value at location (u_v, v_v). After the depth assignments of all layers are done separately, then, we merge these *N* warped depth layers into one depth map. When the same location has multiple depth values, the smallest depth is adopted as discussed earlier. The algorithm is called Backward Depth Warping Algorithm with N planes (BDWA-N).

IV. Experimental results

We implement the entire synthesis system on a PC platform. The results are shown in Fig.5. Fig.5(a) and (b) are the original left image and its depth map. The depth map is calculated using the MPEG reference software, DERS (Depth Estimation Reference Software) [8]. The depth map in Fig.5(c) is produced by using the forward warping technique to map the left depth map to the right camera position. Fig.5(d) is the depth map produced by using the BDWN-256 technique described in the last section. It is clear that the backward warping produces a better quality depth map, which has fewer holes and cracks. If we examine Fig.3, every target pixel on Cam_R can find a nearest pixel on Cam-L when it is backwardly mapped to Cam_L. This is not the case in the forward warping process.

	
(a) left reference image	(b) left reference depth map



(c) FDW

(d) BDWA-256

(e) virtual view synthesis (FDW)

(f) virtual view synthesis (BDWA-256)

Figure 5. The impacts of forward/backward warping algorithm on synthesized images.

When these two depth maps together with the original left image are used to synthesize the right image, the BDWN-256 depth map leads to a better subjective quality image as shown in Fig.5 (e) and (f). The detailed portions of images are enlarged and displayed in Fig.6.

V. Conclusions

In this paper, we propose a backward warping algorithm for mapping the depth map from the reference view to the target view. Generally, this backward warping method reduces the artifacts during the warping processes. Consequently, the synthesized images can achieve better subjective visual quality with fewer post-processing steps. However, its computational complexity is much higher. In the case of 256 planes, we need to calculate the inverse mapping of Eq.(8) on every plane. (In fact, the zero-value plane is not in use and thus, there are only 255 planes.)

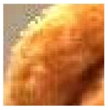
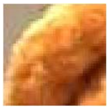
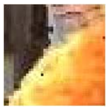
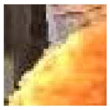
	Ground Truth	FDW	BDWA-256
ROI 1			
ROI 2			
ROI 3			

Figure 6. Enlarged portions of virtual view synthesized images using the forward/backward depth wrappings.

We can think of several ways to speed up the computations. The number of different depth values in a typical depth map is much smaller than 255. Also, even when the original depth map contains a number of different depth values, similar values may be merged and represented by one single value without noticeable distortion. The fast backward warping algorithms are now under development.

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.

References

[1] M. Tanimoto, M.P. Tehrani, T. Fujii, and T. Yendo, “Free-Viewpoint TV,” IEEE Signal Processing Magazine, vol. 28, Issue 1, pp. 67-76, January 2011.

[2] A. Smolic, “3D video and Free-viewpoint Video - From capture to display,” Pattern Recognition, vol. 44 , pp.1958–1968, 2011.

[3] K. Muller, “Developments in Depth-Enhanced 3D Video Coding,” Proc. International Workshop on Advanced Image Technology, Dec. 2013.

[4] ISO/IEC JTC1/SC29/WG11, MPEG document M15672, “View synthesis software and assessment of its performance,“ July 2008.

[5] P. Kauff, et al., “Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability,” *Signal Process: Image Communication,* pp. 217-234, Feb. 2007.

[6] R. Szeliski, “Image-Based Rendering,” Computer Vision Algorithms and Applications, chap. 13, August 2010.

[7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge University Press, 2004.

[8] M. Tanimoto, T. Fujii, and M. Panahpour, *Depth Estimation Reference Software DERS 5.0*, ISO/IEC MPEG M16923, Oct. 2009.