A Hole Filling Technique in The Temporal Domain for Stereoscopic Video Generation

Jae-Il Jung and Yo-Sung Ho
Gwangju Institute of Science and Technology (GIST), Gwangju, Korea
E-mail: {jijung, hoyo}@gist.ac.kr Tel: +82-62-715-2258

***Abstract*— In this paper, we describe a stereoscopic video generation system from one-view color video and its corresponding depth information. This system allows stereoscopic data to be transmitted through a limited bandwidth, but it has an empty hole problem in the synthesized view. Therefore, we propose a hole filling algorithm referring to neighboring frames as well as spatial neighbors. We synthesize both color and depth videos by 3D warping, and linearly interpolate the holes in the depth video. With the interpolated depth values, we search the corresponding color textures in the temporal domain and fill holes in the color image with them. Experimental results show that the proposed algorithm fills the hole regions properly in the complex background.**

I. INTRODUCTION

As the 3D video service becomes popular, a variety of 3D display systems are selling well and various researches related to 3D image processing are being carried out [1][2]. Since the 3D video provides realistic and immersive feeling, these interests in 3D video have a bright prospect.

The 3D contents leading current 3D markets are produced on the basis of the stereoscopic image. The basic principle for stereoscopic perception is the binocular disparity of the human visual system [3]. Two slightly different images projected on the retinas of the eyes are fused in the brain. This process is simulated by having two cameras arranged with the same inter-ocular distance as the human eyes. The two cameras with coplanar image sensors will model the human visual system in respect to the difference in perspective between the two viewpoints. When each camera's image is presented only to the corresponding eye of the viewer, the eye-brain will fabricate the stereoscopic depth of the image.

However, the storage and transmission of stereoscopic video material involves a large amount of data. A stereoscopic video with a single left-right pair needs double raw data when comparing with a conventional 2D video. Therefore, a considerable research effort is focused on realizing compression to obtain savings in bandwidth and storage capacity.

Since a stereoscopic image pair essentially depicts the same scene from two different points of view, the independent coding of both images of a stereoscopic pair is redundant [4]. Multi-view video coding (MVC) can reduce amount of redundant data by using inter-view statistical dependencies [5].

The other approach for effective compression is encoding one-view color video and its corresponding depth map(one-view + one-depth) [6]. The depth video is a 2D array sequence whose pixel values represent distances of the color video. With the color and depth videos, the other color video of the stereoscopic pair is synthesized at a decoding part by depth image based rendering (DIBR).

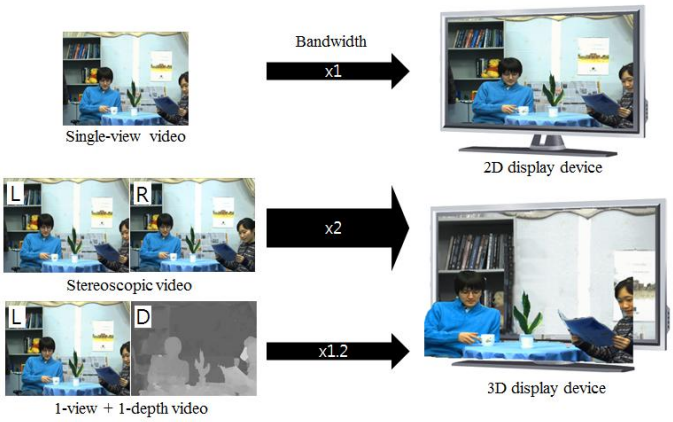


Fig. 1 Bandwidth Comparison of single-view, stereoscopic and one-view + one-depth videos

Because the depth map is relatively simple and a single-channel array, this format can reduce a total amount of data. Figure 1 shows the comparison of bandwidth of each data format. This method has an additional advantage that we can freely control depth ranges of videos according to types of display devices. It is a hot issue in 3D video [7].

Despite of the advantages, it has a critical problem. When a decoding part synthesizes a virtual view with color and depth videos, occlusion regions appear. The occlusion is newly exposed region due to view point change as shown in Fig. 2. Because the occlusion regions have no texture information in the input data, they look like holes on the synthesized video.

It significantly degrades the quality of the stereoscopic video. Various algorithms have been proposed to restore holes [8]. Most algorithms refer to neighbor textures of holes, and they are called inpainting algorithms. Since they make some textures out of nothing, their accuracy is not guaranteed, especially in complex regions.

In this paper, we explain the general procedure of stereoscopic generation and a temporal hole filling technique.

Because video data have a consecutive sequence of similar images, it is possible to find the texture information of hole regions from other neighboring frames.

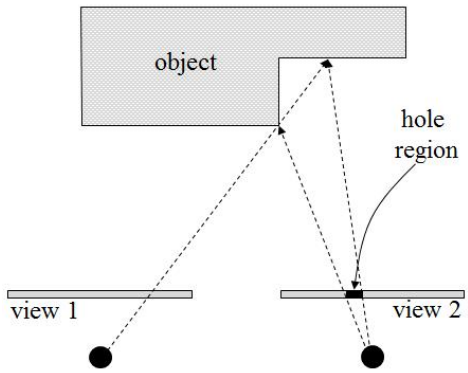


Fig. 2 occlusion appearance

It significantly degrades the quality of the stereoscopic video. Various algorithms have been proposed to restore holes [8]. Most algorithms refer to neighbor textures of holes, and they are called inpainting algorithms. Since they make some textures out of nothing, their accuracy is not guaranteed, especially in complex regions.

In this paper, we explain the general procedure of stereoscopic generation and a temporal hole filling technique. Because video data have a consecutive sequence of similar images, it is possible to find the texture information of hole regions from other neighboring frames.

II. STEREOSCOPIC VIDEO GENERATION

For the virtual view synthesis based on the depth video, we use the DIBR algorithm [9]. The depth video describes distances between the camera and objects in a scene, and 3D warping exploits the depth value to find the corresponding position between views. At first, we explain the 3D warping technique for viewpoint shifting. Then, we explain the temporal hole filling method.

Assuming that two cameras are calibrated, we can define the pixel correspondences between cameras with camera parameters. When a point $\tilde{M}$ in world coordinate is projected to a camera, a pixel $\tilde{m}$ in the image can be found using (1).

$$\tilde{m} = A[R \mid t]\tilde{M} \tag{1}$$

where A, R, and t denote the intrinsic matrix, rotation matrix, and translation vector, respectively. The representations of a single point in a scene $\tilde{M} = [X\ Y\ Z\ 1]^T$ and a projected point $\tilde{m} = [x\ y\ 1]^T$ are the homogeneous forms.

We can put back a pixel m_t in the transmitted image back the world coordinate using (2).

$$M_t = R_t^{-1} \cdot A_t^{-1} \cdot m_t \cdot d(m_t) - R_t^{-1} \cdot r_t \tag{2}$$

where the representations of A_t, R_t, and t_t describes camera parameters of the transmitted view. $d(m_t)$ is a depth value of the pixel m_t. After this backward projection, we project M_t into the virtual camera coordinates using (3).

$$m_t = A_v [R_v \mid t_v] M_t \tag{3}$$

As a result, we can find the relationship between two positions m_t and m_v. By applying this process to all pixels in the transmitted video, we can get the virtual view at another view position.

III. TEMPORAL HOLE FILLING

After 3D warping, we need to fill the holes in the synthesized image to improve the visual quality. The holes are classified into two types. The first holes are come from round-off errors during 3D warping, and they are easily filled with neighboring pixel values without considerable quality degradation. The second holes are induced by view position change and appear near object boundaries. In general, the size of the second hole is larger than that of the round-off hole, the second holes locate between foreground and background objects. These holes are main factors degrading viewing quality of stereoscopic videos, and it is not straightforward to fill the holes with proper values. We, therefore, focus on the second holes in this paper.

The transmitted video consists of a consecutive sequence of frames. Therefore the texture of hole regions can be found from other frames if the camera does not move during a certain period as shown in Fig 3. It is a common assumption of video inpainting technique.

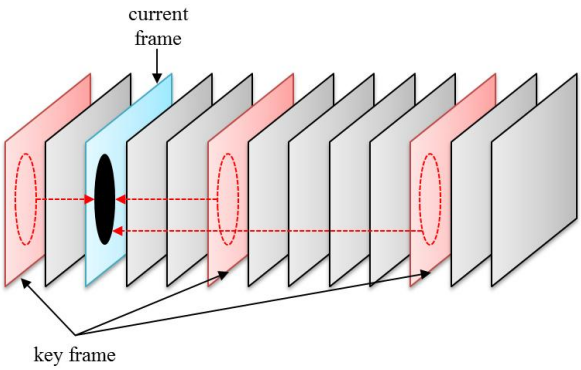


Fig. 3 Concept of temporal hole filling

It is an irrational way to search an available texture for the current hole in whole frames. Therefore, we set the key frames to which are referred.

A. *Warp depth map and fill holes*

What is important in our temporal hole filling algorithm is to find the corresponding texture of the hole region from other frames. It is not straightforward because objects moves and they can hide the hole region. Therefore, it is not guaranteed

that the co-located texture in the neighboring frame is the corresponding region of the current hole.

In order to find accurate the corresponding region, we consider the depth values of the hole region. During the 3D warping process, we warp not only the color video but also the depth video to estimate depth values of the holes. However, there are no depth values in the hole region, because they are also in the occlusion region. We estimate them with considering neighboring depth values. The texture of a depth image in general is simpler than that of a color image. Due to this property, linear interpolation of neighbor background's depth values provides the reliable result as shown in Fig. 4.

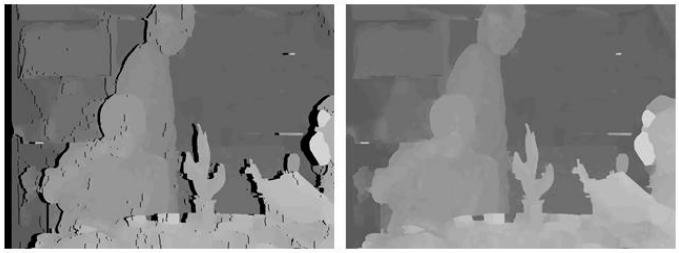


Fig. 4 Depth warping and hole filling

B. *Find corresponding textures*

After filling the holes in the depth video, we search the corresponding textures of the hole from other frames. We select a reference frame to get accurate textures, which has proper texture information. This process is pixel-wise, and is only applied to pixels in hole regions. For high accuracy, we design the block-based function as (4).

$$N_{ref\_frame}(i) = \arg\min_{t} \sum_{i \in n} w_t(cur\_d_i - ref\_d_{t,i}) \qquad (4)$$

where n means the neighbors of the current position i. cur_d is the depth value of the current frame, and ref_dt is the depth value of the t th frame. w_t is a weighting factor for a temporal distance between the current and t th frames, and it is designed as

$$w_t = exp(w | N_{cur\_frame} - t | +1) \qquad (5)$$

where N_{cur_frame} is the current frame number, and w is a controlling parameter. If the value of $\sum_{i \in n} w_t(cur\_d_i - ref\_d_{t,i})$ is greater than a certain value, (4) does not return the frame number.

By using (4) and (5), we select a reference frame for the hole, which has the similar depth values and is close to the current frame. After selection, we map the texture of the reference frame to the hole of the current frame as (6).

$$t_c(i) = \begin{cases} t_c(i) & if\ i \notin hole \\ t_{N_{ref\_frame(i)}}(i) & else\ if\ \exists Nref\_frame(i) \end{cases} \qquad (6)$$

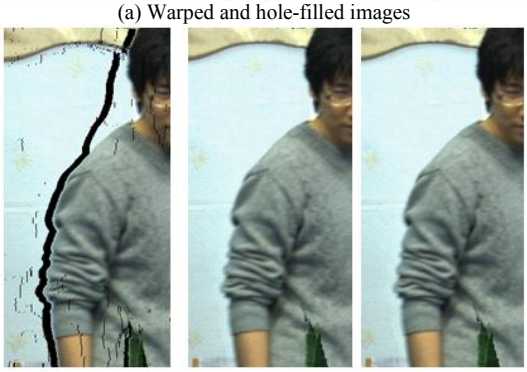
where $t_c(i)$ *and* $t_x(i)$ mean the textures of the position i in the current and the x th frame, respectively. If $N_{ref\_frame}(i)$ does not exist, our method cannot be applied. This case is mostly relevant the holes near static objects. In this case, we apply the conventional spatial inpainting technique to fill the holes.

IV. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed algorithm, we apply it to one-view and one-depth video. We synthesize the virtual view using 3D warping with the camera parameters given by sequence providers.

Figure 5 shows the results for the "Newspaper" sequence. The images in Fig. 5(a) are the warped and hole-filled images with our proposed algorithm. Fig. 5(b) shows the enlarged images of the hole region: without hole filling, with spatial hole filling [6], and with temporal hole filling. Our proposed algorithm shows the reliable results without boundary mismatches. Since "Newspaper" sequence has a simple background region, it is not easy to verify the differences between two algorithms.

(a) Warped and hole-filled images

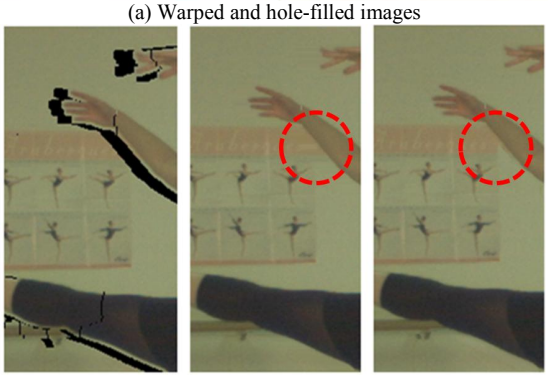


(b) Comparison of hole filling

Fig. 5 "Newspaper" sequence

Therefore, we chose another sequence whose textures of backgrounds are complex. Figure 6 is the result of the "Ballet" sequence. As you can see in the central image of Fig. 6(b), the conventional spatial inpainting algorithm cannot effectively restore holes near the background whose textures are complex (red dotted box).

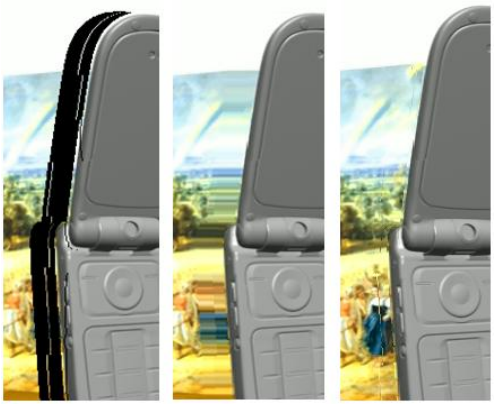
(a) Warped and hole-filled images



(b) Comparison of hole filling

Fig. 6 "Ballet" sequence

(a) Warped and hole-filled images



(b) Comparison of hole filling

Fig 7 "Mobile" sequence

Figure 7 is the result of the "mobile" sequence, and this sequence has very complex background textures. This sequence emphasizes the difference between conventional algorithm and our algorithm. The conventional inpainting algorithm is not able to restore the hole region, because spatial neighbors do not have background texture. In case of our method, the hole regions are estimated from temporal neighbors, and it can reconstruct textures in the background region.

V. CONCLUSIONS

To transmit stereoscopic video data through a limited bandwidth, stereoscopic video generation has been proposed. Although it improves coding efficiency, it has the hole filling problem. In this paper, we proposed the hole filling algorithm using texture information of neighboring frames. We synthesize both color and depth videos by 3D warping, and interpolate holes in the warped depth video. With the interpolated depth values, we find texture information of holes from neighboring frames. The experimental results show the proposed algorithm provides better results than the conventional spatial inpainting technique, especially for sequences having complex backgrounds.

ACKNOWLEDGMENT

This research is supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Developement Program 2011.

REFERENCES

[1] Y. Kang, E. Lee, and Y. Ho, "Multi-Depth Camera System for 3D Video Generation," Proceedings of International Workshop on Advanced Image Technology, pp. 44(1-6), Jan. 2010.

[2] S. Yoon and Y. Ho, "Multiple Color and Depth Video Coding Using a Hierarchical Representation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 11, pp. 1450-1460, Nov. 2007.

[3] M. Johanson, "Stereoscopic Video Transmission over The Internet," Proceedings of the IEEE Workshop on Internet Applications, pp. 12-19. July 2001.

[4] Y. Kim, J. Lee, C. Park, and K. Sohn, "MPEG-4 Compatible Stereoscopic Sequence Codec for Stereo Broadcasting," IEEE Transactions on Consumer Electronics, vol. 51, no. 4, pp. 1227-1236, Nov. 2005.

[5] ISO/IEC JTC1/SC29/WG11, "Vision on 3D Video", N10357, Feb. 2009.

[6] I. Shin and Y. Ho, "GPU Parallel Programming for Real-time Stereoscopic Video Generation," International Conference on Electronics, Information, and Communication, pp. 315-318, July 2010.

[7] E. Lee, H. Heo, and K. Park, "The Comparative Measurements of Eyestrain Caused by 2D and 3D displays," IEEE Transactions on Consumer Electronics, vol. 56, no. 3, pp. 1677-1683, Aug. 2010.

[8] Z. Tauber, Z. Li, and M. Drew, "Review and Preview: Disocclusion by Inpainting for Image-Based Rendering," IEEE Transactions on Systems, Man, and Cybernetics-PART C: Applications and Reviews, vol. 37, no. 4, pp. 527-540. July 2007.

[9] A. Smolic and D. McCutchen, "3DAV Exploration of Video-Based Rendering Technology in MPEG," IEEE Transactions on Circuits and Systems for Video Technology, vol. 14 no.3, pp.348-356, Mar. 2004.