# 2012 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future

## Ho Chi Minh City, Vietnam
## 27 February – 01 March 2012

IEEE

**IEEE Catalog Number:** CFP1256A-PRT
**ISBN:** 978-1-4673-0308-8

# 2012 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future



| | |
|---|---|
| IEEE Catalog Number: | CFP1256A-ART (Xplore) |
| ISBN 13: | 978-1-4673-0309-5 (Xplore) |
| IEEE Catalog Number: | CFP1256A-PRT (Print) |
| ISBN 13: | 978-1-4673-0308-8 (Print) |

**Printed copies of this publication are available from:**

Curran Associates, Inc
57 Morehouse Lane
Red Hook, NY 12571 USA
Phone: (845) 758-0400
Fax: (845) 758-2633
E-mail: curran@proceedings.com

Produced by IEEE eXpress Conference Publishing

For information on producing a conference proceedings and receiving an estimate, contact conferencepublishing@ieee.org
http://www.ieee.org/conferencepublishing

TABLE OF CONTENTS

# Virtual View Synthesis using Temporal Hole filling with Bilateral Coefficients

Jae-Il Jung and Yo-Sung Ho

Gwangju Institute of Science and Technology (GIST)
123 Cheomdan-gwagiro, Buk-gu, Gwangju 500-712, Republic of Korea
{jijung, hoyo}@gist.ac.kr

***Abstract*—A depth image-based virtual view synthesis always induces a hole problem, and the holes degrade viewing quality of the synthesized image. In this paper, we propose a temporal hole filling algorithm referring to neighboring frames in the temporal domain. In order to find corresponding textures for a current hole, we synthesize both color and depth videos by the 3D warping technique, and linearly interpolate the holes in the depth video. With the interpolated depth values, we search the corresponding color textures from the neighboring key frames and fill them. The experimental results show that the proposed algorithm can find appropriate textures in the temporal domain and fill the hole regions even in complex scenes.**

***Keywords-hole filling; virtual view synthesis; temporal domain; sterescopic video***

## I. Introduction

Recently, the interest in 3D video is attracting much attention and a 3D video service becomes popular. In addition, a variety of mass products for 3D display are selling. To keep pace with the rapid changes in 3D markets, various researches related to 3D image processing are being carried out [1-4].

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 based on the stereoscopic image, which is captured by two cameras at different position. The basic principle for stereoscopic perception is a binocular disparity of the human visual system. 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 [5].

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 [6]. A conventional 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 [7]. Multi-view video coding (MVC) can reduce amount of redundant data by using inter-view statistical dependencies [8].

The other approach for effective compression is encoding one-view color video and its corresponding depth map(one-view + one-depth) [9]. 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). Figure 1 shows the geometric structure of virtual and original views in stereoscopic video system.

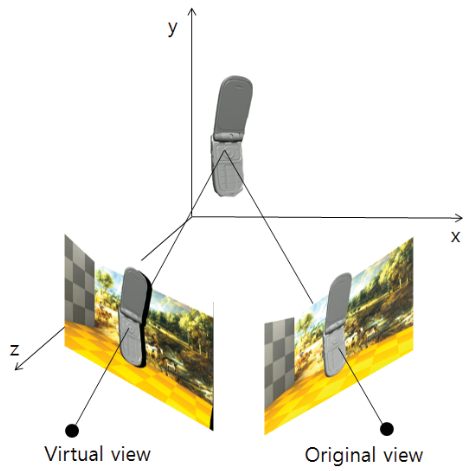



Figure 1. Geometric structure of the virtual and original views

The depth map is relatively simple, and it is a single-channel array. Therefore, this method can reduce a total amount of data. Figure 1 shows the comparison of bandwidth of each data format. This method has an additional advantage. We can freely control depth ranges of videos according to types of display devices. It is a hot issue in 3D video [10].

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.



978-1-4673-0309-5/12/$31.00 ©2012 IEEE

Because the occlusion regions do not have texture information, they look like holes on the synthesized video.

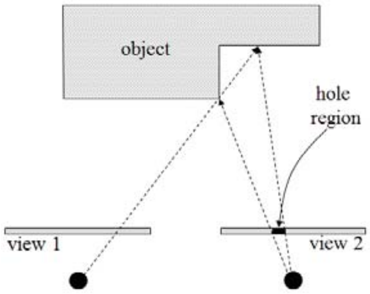



Figure 2. occlusion appearance

They degrade the quality of the stereoscopic video. Various algorithms have been proposed to restore holes [11-13]. Most algorithms refer to neighbor textures of holes, and they are called inpainting algorithms. Since they make some textures out of nothing, the accuracy is not guaranteed, especially in complex regions.

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

## II. VIRTUAL VIEW GENERATION

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

### A. Depth image based rendering

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

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

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

We can put back a pixel $\boldsymbol{m}_t$ in the transmitted image back the world coordinate using (2).

$$\boldsymbol{M}_t = \boldsymbol{R}_t^{-1} \cdot \boldsymbol{A}_t^{-1} \cdot \boldsymbol{m}_t \cdot \boldsymbol{d}(\boldsymbol{m}_t) - \boldsymbol{R}_t^{-1} \cdot \boldsymbol{r}_t \tag{2}$$

where the representations of $\boldsymbol{A}_t$, $\boldsymbol{R}_t$, and $\boldsymbol{t}_t$ describes camera parameters of the transmitted view. $\boldsymbol{d}(\boldsymbol{m}_t)$ is a depth value of the pixel $\boldsymbol{m}_t$. After this backward projection, we project $\boldsymbol{M}_t$ into the virtual camera coordinates using (3).

$$\boldsymbol{m}_t = \boldsymbol{A}_v[\,\boldsymbol{R}_v \mid \boldsymbol{t}_v\,]\,\boldsymbol{M}_t \tag{3}$$

As a result, we can find the relationship between two positions $\boldsymbol{m}_t$ and $\boldsymbol{m}_v$. By applying this process to all pixels in the transmitted video, we can get the virtual view video having holes.

### B. Hole filling

The warped image always contains the hole due to viewpoint change, and the hole degrades viewing quality of 3D video. In order to effectively fill holes, various inpainting algorithms have been proposed [11-13].

Telea proposed an inpainting algorithm based on propagating an image smoothness estimator along the image gradient, and he estimate the image smoothness as a weighted average over a known image neighborhood of the pixel. The fast marching method is used to propagate image information [11].

Shin *et al.* proposed a fast and effective hole filling technique with consideration of depth values [12]. They reflect the shape and width of objects, and fill the holes with proper textures. Their algorithm is designed in parallel and implemented on GPU.

Although various algorithms have been proposed, they cannot effectively cover the holes in virtual view because the virtual view has large holes, and the hole frequently appear between foreground and background. In this paper, we introduce an effective hole filling algorithm using temporal information.

## III. TEMPORAL HOLE FILLING

As we mentioned, we need to fill the holes in the synthesized image to improve the visual quality. The holes are classified into two types. The first type is come from round-off errors during 3D warping, and it is easily filled with neighboring pixel values without considerable quality degradation. The second type is induced by shifting objects. In general, the size of these holes is large and located between foreground and background objects. These holes are main factors degrading the visual quality, and it is not straightforward to fill the holes with proper values. We focus on the holes of the second type, in this paper.

Figure 3. Set of reference keyframes

The transmitted video consists of a consecutive sequence of image frames. Therefore the texture of hole regions can exist in other frames if the object near the holes is moving. Since it is an irrational way that we search an available texture in whole frames, we set the several key frames to which are referred as Fig. 3.

### *A. Depth Image Interpoation*

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 we cannot guarantee 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. Thus, we warp not only the color video but also the depth video. 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, simple copy of neighbor background's depth values provides the reliable result as shown in Fig. 4.

Figure 4. Depth warping and hole filling

### *B. Find Corresponding Textures*

After generating the virtual depth video, we search the corresponding textures of the holes from the reference key frames. At first, we select the appropriate reference key frame, which contain the texture information of the current hole. This process is pixel-wise, and is only applied to pixels in hole regions. We introduce the concept of bilateral coefficient in our algorithm, and it is extended with consideration of depth, color, and frame distance as (4).

$$N_{ref\_frame}(i) = \arg\min_{i} \text{cost}_{depth}(i) + \text{cost}_{framedistance}(i) + \text{cost}_{color}(i) \quad (4)$$

where cost$_{depth}(i)$, cost$_{framedistance}(i)$, and cost$_{color}(i)$ are costs of depth difference, frame distance, and color difference for current keyframe $i$. cost$_{depth}(i)$ measures the depth difference between the reconstructed depth $d$ of the current position and the depth $d_{keyframe}$ of the reference keyframe $i$ at co-located position. It is defined as (5)

$$\text{cost}_{depth}(i) = w\exp(d_{keyframe}(i) - d)^2 \quad (5)$$

where $w$ is the weighting factor to control the influence of the cost. The cost$_{framedistance}(i)$ penalizes the distance between current frame and the keyframe candidate, and defined as (6).

$$\text{cost}_{framedistnace}(i) = |\, frame_{keyframe}(i) - frame \,| \quad (6)$$

The cost$_{color}(i)$ penalizes the color difference of neighboring pixels, and it consider sensitivity of human eyes. The cost is defined as (7).

$$\begin{aligned} \text{cost}_{color}(i) = {} & 0.257\,|\, r_{keyframe}(i) - r \,| \\ & + 0.504\,|\, g_{keyframe}(i) - g \,| \\ & + 0.098\,|\, b_{keyframe}(i) - b \,| \end{aligned} \quad (7)$$

By applying the winner-takes-all algorithm to (4), we select an optimized reference frame for the current hole, which has the similar depth and color 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 (8).

$$t_c(i) = \begin{cases} t_c(i) & \text{if } i \notin hole \\ t_{N_{ref\_frame(i)}}(i) & \text{else if } \exists Nref\_frame(i) \end{cases} \quad (8)$$

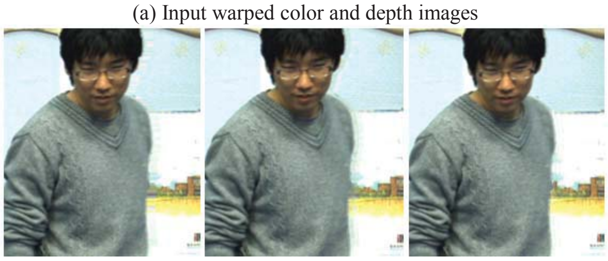
where $t_c(i)$ *and* $t_x(i)$ mean the textures of the position $i$ in the current and the $x$ th frame, respectively. If $N_{ref\_frame}(i)$ does not exist, our method cannot be applied. This case is mostly relevant the holes near static objects. In this case, we apply the conventional spatial inpainting technique [8] to fill the holes.

## IV. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed algorithm, we applied our algorithm to the stereoscopic video generation system. We synthesized the virtual view using the 3D warping technique with each camera parameter. We compared the proposed algorithm with the conventional hole filling algorithms: Method A[2], and Method B[8].

Figure 5(a) shows the input warped color and depth images of *newspaper* sequence. We applied the three algorithm to fill the hole region caused by warping, and the results are demonstrated in Fig. 5(b). While Method A and Method B blur the boundary of the objects, the proposed algorithm shows the reliable results without boundary mismatches. However the difference between the algorithms is not noticeable, since *newspaper* sequence has a simple background region.

(a) Input warped color and depth images



(b) Hole filling results: Method A, B and the proposed algorithm
Figure 5. "Newspaper" sequence

Therefore we chose another sequence whose textures of backgrounds are complex. Figure 6 is the result of the *cafe* sequence. As you can see in the figure, the conventional spatial inpainting algorithm cannot effectively restore the text in the background, but the proposed algorithm effectively restores this region.

(a) Input warped color and depth images

(b) Hole filling results: Method A, B and the proposed algorithm

Figure 6. *cafe* sequence

(a) Input warped color and depth images

(b) Hole filling results: Method A, B and the proposed algorithm

Figure 7. *mobile* sequence

Figure 7 is the result of *mobile* sequence, and this sequence has very complex background textures. This sequence emphasizes the difference between the conventional algorithms and our algorithm. Both conventional algorithms severely degrade viewing quality, because it is impossible to estimate textures in the hole from spatial neighbors. In case of our method, the hole regions are estimated from temporal neighbors, and it can reconstruct textures in the background region.

## V. CONCLUSION

For virtual view synthesis, the hole filling is very important part, and it runs a tight ship of entire systems. In this paper, we proposed a 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 by using extended bilateral coefficients. It considers not only color and frame distance, but also depth difference. The experimental results show the proposed algorithm provides better results than the conventional spatial inpainting technique, especially for sequences having complex backgrounds.

## 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] E. K. Lee and Y. S. Ho, “Generation of Multi-View Video Using a Fusion Camera System for 3D Displays,” IEEE Transactions on Consumer Electronics, vol. 56, pp. 2797-2805, Nov. 2010. [1]

[3] 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.

[4] P. Merkle, K. Muller, and T. Wiegand, "3D Video: Acquisition, Coding, and Display," IEEE Transactions on Consumer Electronics, vol. 56, pp. 946-950, May 2010.

[5] E. Lee, H. Heo, 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.

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

[7] 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.

[8] 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.

[9] Z. Liang and W. J. Tam, "Stereoscopic image generation based on depth images for 3D TV," *Broadcasting, IEEE Transactions on,* vol. 51, pp. 191-199, 2005.

[10] ISO/IEC JTC1/SC29/WG11, “Vision on 3D Video”, N10357, Feb. 2009.

[11] 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.

[12] A. Telea, “An image inpainting technique based on the fast marching method,” JOURNAL OF GRAPHICS TOOLS., vol. 9, pp. 23-34, 2004.

[13] 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.