Abstract—In this paper, we propose a multi-view stereoscopic image synthesis algorithm for 3DTV system using depth information with an RGB texture from a depth camera. The proposed algorithm synthesizes multi-view images which a virtual convergence camera model could generate. Experimental results showed that the performance of the proposed algorithm is better than those of conventional methods.

Index Terms—Multi-view synthesis, Stereoscopic image, Depth Image based rendering (DIBR), Occlusion Processing

I. INTRODUCTION

The recent emergence of multimedia applications has motivated a considerable interest in the development of three-dimensional (3D) TV aiming at providing a real dynamic scene. In principle, ideal 3DTV system can be achieved by a hologram. It simultaneously services many viewers, and provide for each of them stereoscopic depth cue, lens accommodation cue, and motion parallax cue [1]. Nevertheless, the current level of technology does not allow real-time system to acquire holographic videos.

The field of image-based rendering has long focused on the problem of synthesizing images from geometric models. While image-based rendering technology has made significant strides towards achieving photorealism, the process of synthesizing accurate models still requires large computational complexity [2, 3].

Fig. 1(a) and (b) show a texture image and a depth image captured by the depth camera. Fig. 1(c) shows one of the multi-view images captured by a multiple-camera system. Fig. 1(d) is a depth image of Fig. 1(c) obtained by view interpolation technique using a layered representation [9]. Comparing Fig. 1(b) with (d), we can realize that more accurate depth information can be obtained from the depth camera.

In the system proposed by ATTEST, transmitted data is decoded at the 3DTV set-top box and decompressed color textures and depth maps are retrieved. And then, “virtual” views may be generated by using the depth image based rendering (DIBR) technique in real time. DIBR technique is based on shift-sensor algorithm [4, 5] where a parallel camera model is assumed so that a view angle can not be taken into account.

If a viewer moves his head, viewpoint may be changed. This phenomenon is called motion parallax cue and it is one of important factors for the viewer to experience the 3D effect [6, 7]. Motion parallax cue based on a virtual convergence camera model can be generated by using an RGB texture image and its corresponding depth information. In this paper, we propose a multi-view synthesis algorithm with this information from a depth camera.

![Fig. 1. Texture and depth images](a) Texture image obtained by a depth camera, (b) a depth map of (a), (c) An image obtained by multiple camera system and (d) depth image of (c).]
convergence camera model. First we synthesize multi-
view images by considering viewer’s movement at
different view angles; on plane (horizontal movement)
and on plane (vertical movement), as shown in Fig 2.
in what follow , with each synthesized view,
corresponding stereoscopic views are also generated
by using Eq (1).

\[
\begin{align*}
I(\theta) &= I_c T_x T_y(\theta) \\
&= \begin{cases} 
I(\theta)_L = I(\theta) - d \\
I(\theta)_R = I(\theta) + d
\end{cases} \\
I(\phi) &= I_c T_x T_y(\phi) \\
&= \begin{cases} 
I(\phi)_L = I(\phi) - d \\
I(\phi)_R = I(\phi) + d
\end{cases}
\end{align*}
\] (1)

Where \( I(\theta)_L, I(\theta)_R \) are the synthesized left and right
images each with a view point \( \theta \) and \( I(\phi)_L, I(\phi)_R \) are
the synthesized left and right images each with view point
\( \phi \), respectively. \( d \) is a disparity, \( I_c \) is an original texture
image, \( T_x \) and \( T_y \) are intrinsic and rotation
matrices.

Fig. 3 shows both of normal texture image and the
Corresponding depth map captured by a depth camera:
Photometric information and geometric information
are then allegedly contained in texture image \( I(x, y) \),
and a pixel-by-pixel depth map \( D(x, y) \), respectively.

A depth map can be simply converted to 3-D
coordinates by using Eq. (2).

\[
\begin{bmatrix}
P_x(x, y) \\
P_y(x, y) \\
P_z(x, y)
\end{bmatrix} = \frac{1}{D(x, y)} \begin{bmatrix}
S_x \xi \\
S_y \eta \\
S_z
\end{bmatrix}
\] (2)

where points are indexed by an ordered pair of integers
\((x, y)\). Scale parameters \( S_x, S_y \) and \( S_z \) would be a
function of camera’s internal parameters such as focal
length, pixel aspect ratio and pixel resolution.
traditional cel-animator. In which background is first painted, and then foreground moves over the background. By doing so, different background regions may be occluded and revealed. Foreground and background layers are determined by using a predefined threshold value \( T_{\text{layer}} \). If the intensity value of a depth map \( D(x,y) \) of some area is higher than the pre-set threshold value, the given area is determined as an object that is close to the camera, and it belongs to the foreground layer.

In order to synthesize multi-view images with convergence camera model, we introduce the inverse mapping method that rotates any object around the abscissae. By using equation (2), we can change any point in 2D into the one in 3D geometry system. Rotation and translation techniques are used to manifest the corresponding new points according to the given viewpoint by Eq. (3).

\[
[x', y', z'] = [x, y, z]T_R T_I \tag{3}
\]

where \( T_R \) and \( T_I \) are rotation and intrinsic matrices. It now allows us to obtain arbitrary viewpoint image through view angle compensation of foreground layer associated with the arbitrary viewpoint \( \theta, \phi \) by equation (3). In this case, the rotation matrix \( T_R \) will be replaced with \( T_{\theta, \phi}, T_{\theta, \phi} \) as shown in Eq. (4):

\[
T_{\theta, \phi} = \begin{bmatrix}
\cos \theta & 0 & -\sin \theta & 0 \\
0 & 1 & 0 & 0 \\
\sin \theta & 0 & \cos \theta & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
T_{\theta, \phi} = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
\cos \phi & 0 & -\sin \phi & 0 \\
\sin \phi & 0 & \cos \phi & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \tag{4}
\]

where \( \theta, \phi \) are horizontal and vertical angles at an arbitrary viewpoint, as mentioned earlier in this paper. The synthesized images at \( \theta = \pm 7^\circ \) are shown in Fig. 4.

![Fig. 4. Synthesized images for virtual convergence camera model at \( \theta = \pm 7^\circ \)](image)

### III. EXPERIMENTAL RESULTS

In this experiment, we try to synthesize arbitrary view point images. Two video sequences, “Interview” and “Orbi” are employed as test sequences. 100 frames are used in each sequence and each frame is composed of 4:2:0 YUV components. We have a RGB texture image and its associated per-pixel depth map image for each frame and the size of a frame is 720x576. The synthesized virtual left image is shown in Fig. 5. Fig. 5 (a) and (b) represent an original image and synthesized left image, respectively. You can see that objects that are closer to the camera moves over the background (see the gap between man’s left arm and a stool) in Fig. 5(b).

Fig. 6 shows synthesized left and right images (stereoscopic images) at intermediate view, as well as their difference image. As shown in Fig. 6(c), objects that are closer to the camera have larger disparity.

![Fig. 5. Original image and synthesized left viewpoint image.](image)

Synthesized sequences are also played in order to show the image quality, especially focused on an effect of multi-view adaptation in stereoscopic display. For the simulation, a 18-inch stereoscopic display screen with polarized lens is used. We set viewing area in a range of one to two meters from the screen and adjust the viewpoint angle between \( \theta, \phi = +8^\circ \) and \( \theta, \phi = -8^\circ \) from x- and y-axis. Finally, sixteen arbitrary and virtual stereoscopic images (4x4 viewpoints) are synthesized.

### V. CONCLUSION

In this paper, we proposed a multi-view stereoscopic image synthesis algorithm by using a depth camera. In order to represent both the stereoscopic depth cue and motion parallax cue, we produced the layered representation of texture image ordered by depth.

Experimental results show that the proposed algorithm performed a considerably enhanced performance within the predefined view area.
Fig. 6. Synthesized left and right images obtained at arbitrary viewpoint, $\theta = +2^\circ$ and their difference image; (a) Synthesize left image (b) Synthesize right image (c) Difference image.

REFERENCES


Hyun-Jun Choi has received his M.S. and Ph.D. degrees in 2005 and 2009 from Dept. of Electronic Materials Engineering of Kwangwoon University in Seoul, Korea. He was a research professor in Realistic Media Institute at Kwangwoon University. He is currently an assistant professor with the department of Information and Communication Engineering, Anyang University, Anyang-si, Korea. He is research interests are in optical image processing and 3D display.