Dynamic 3D Human Actor Generation Method using a Time-of-Flight Depth Camera

Ji-Ho Cho, Sung-Yeol Kim, Yo-Sung Ho, *Senior Member*, IEEE, and Kwan H. Lee

Abstract — *In this paper, we present a novel method to generate a dynamic 3D human actor using a time-of-flight depth camera, which provides synchronized color and depth images in real time. In practice, depth data obtained from the depth camera have critical problems, such as optical noise, absence of depth information for certain regions, and unmatched boundaries. In this work, we significantly enhance the visual quality of raw depth images of the human actor by performing noise reduction, recovery of the lost hair region, and boundary refinement. The 3D surface of the human actor is then created by applying a 3D mesh triangulation technique on the enhanced depth images. Finally, we generate a dynamic 3D human actor by providing the surface with corresponding color images. Experimental results show that the proposed method produces a natural and immersive 3D human actor successfully while minimizing problems inherent in the current depth camera-based images.*[1]

Index Terms — 3D Human Actor Generation, Time-of-flight Depth Camera, Depth Image-based Rendering.

I. INTRODUCTION

Many researchers have studied immersive display environments and realistic virtual worlds since live avatars can represent human beings in them [1]. In general, we have been generated 3D human actors using computer graphic tools, and applied motion information estimated from a tracking system to avatars [2]. Although avatar-based approaches have achieved high-performance to realize a human actor, it is still short of replacing a human model due to lack of reality [3]. Augmentation of 2D human actor onto the virtual world has been also introduced. However, their applications are limited due to lack of 3D information.

We present a novel alternative in this paper, in which we can represent a 3D human actor by using a depth image-based rendering (DIBR) technique [4]. The previous works related to generating a human actor using DIBR were mostly based on a 'shape-from-X' method, such as shape-from-stereo [5], shape-from-silhouette [6], shape-from-shading [7] and shape-from-focus [8]. Although the DIBR technique is reasonable for reconstructing a 3D human model, it is restricted to special environments that are equipped with synchronized cameras. Furthermore, visual quality of the reconstructed human model is highly dependent on the number of cameras and their calibration results.

In order to increase the accuracy of depth information for a 3D human actor, projector-based active methods have been developed. Zhang *et al*. developed a real-time 3D shape measurement system to get 3D information by simultaneously using a color CCD camera and a black and white CCD camera with a digital fringe projector [9]. In addition, Waschbüsch *et al*. introduced a 3D video acquisition system that consists of a projector, two black and white cameras, and one color camera [10]. However, these projector-based systems make the human actor feel quite uncomfortable due to the projected light from the projector.

As sensor technologies for obtaining depth information advance, we now can capture more accurate per-pixel depth data of a real scene using an active time-of-flight (TOF) depth camera [11, 12]. These depth cameras can directly capture the shape and motion of a 3D human actor by integrating a high-speed pulsed infrared light source onto a conventional broadcast TV camera. Figure 1 shows the principle of a depth camera.

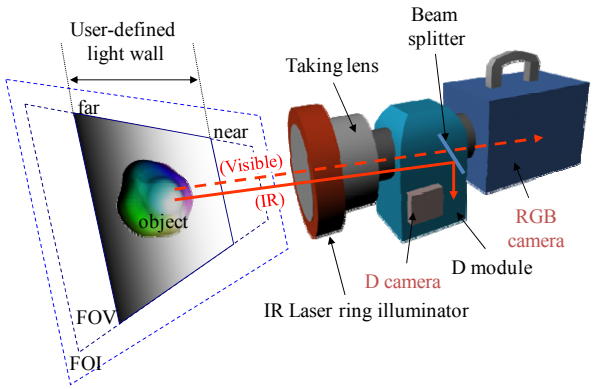


Fig.1. Principle of a time-of-flight depth camera

However, we cannot directly use the depth data of a 3D human actor from this camera due to some problems inherent in the system. In order to use the depth data properly, we need to resolve three main problems exist in the raw depth images.

1) *The captured depth images are very noisy*. The acquired depth images usually contain quantization errors and optical noise, mainly due to the reflectivity or color variation of objects. For example, for a flat black and white object, the white region shows a high depth value while the black region shows a relatively low depth value, although both regions are on the same plane.

[1] J.-H. Cho and K. H. Lee are with the Department of Mechatronics, Gwangju Institute of Science and Technology (GIST), Gwangju, KOREA (e-mail: jhcho@gist.ac.kr, khlee@gist.ac.kr)

S.-Y. Kim and Y.-S. Ho are with the Department of Information and Communications, GIST (e-mail: sykim75@gist.ac.kr, hoyo@gist.ac.kr)

Contributed Paper
Manuscript received October 13, 2008

2) *Depth data on shiny and dark surface regions can be lost*. The depth camera does not capture shiny and dark surfaces well, such as black leather and black hair. As the reflected lights from the dark and shiny surfaces are not only weak, but are scattered, the depth camera cannot detect the reflected infrared light. For a 3D human actor, it often causes the loss of hair region.

3) *The boundary of color images does not match well with its depth images*. When we calibrate the depth camera using an auto calibration tool, it does not guarantee an exact match between the boundaries of both images. The unmatched boundary, of course, causes unnatural results, since the 3D shape is generated from the measured depth information. Some important regions can be missed or unintended background data can appear as a texture.

Kim *et al*. developed a novel modeling algorithm to reduce optical noise in a depth image captured by a TOF depth camera using a series of methods including adaptive sampling, mesh triangulation, and Gaussian smoothing [13]. Their method provides a good platform to make useful depth images of a 3D human actor, however, they mostly concentrated on handling optical noise in depth images.

In this paper, we propose a new method of generating a dynamic 3D human actor using a current TOF depth camera. In this work, we propose some practical solutions that minimize the problems inherent in the depth camera system so that we can generate realistic and natural 3D human actors without serious degradation of visual quality.

The remainder of this paper is organized as follows. In Section 2, we provide a review of the proposed system to generate a 3D human actor using the depth camera and describe how the captured depth images are enhanced to generate a 3D human actor. Experimental results are shown in Section 3. Finally, we conclude in Section 4.

II. GENERATION OF A 3D HUMAN ACTOR

A. Overview of the System

Figure 2 illustrates an overall architecture of the proposed 3D actor generation system using a TOF depth camera. Basically, the depth camera produces both color and depth images simultaneously for each frame. First, the depth quality of a raw depth image is enhanced by three steps: noise reduction, recovery of the lost hair region, and boundary refinement.

To minimize optical noise in the depth image, we employ a new joint bilateral filtering approach that perverse edges well. We detect the lost hair region using a face detection technique and recover its depth information using a multi-seed region growing algorithm considering the depths of a face region. If there is no loss of depth in the hair region, this step will be skipped. Next, we match the boundaries between the color and depth images using a graph cut-based matting algorithm. After improving the quality of depth images, the depth and color images are coded by a video coder, such as a H.264/AVC

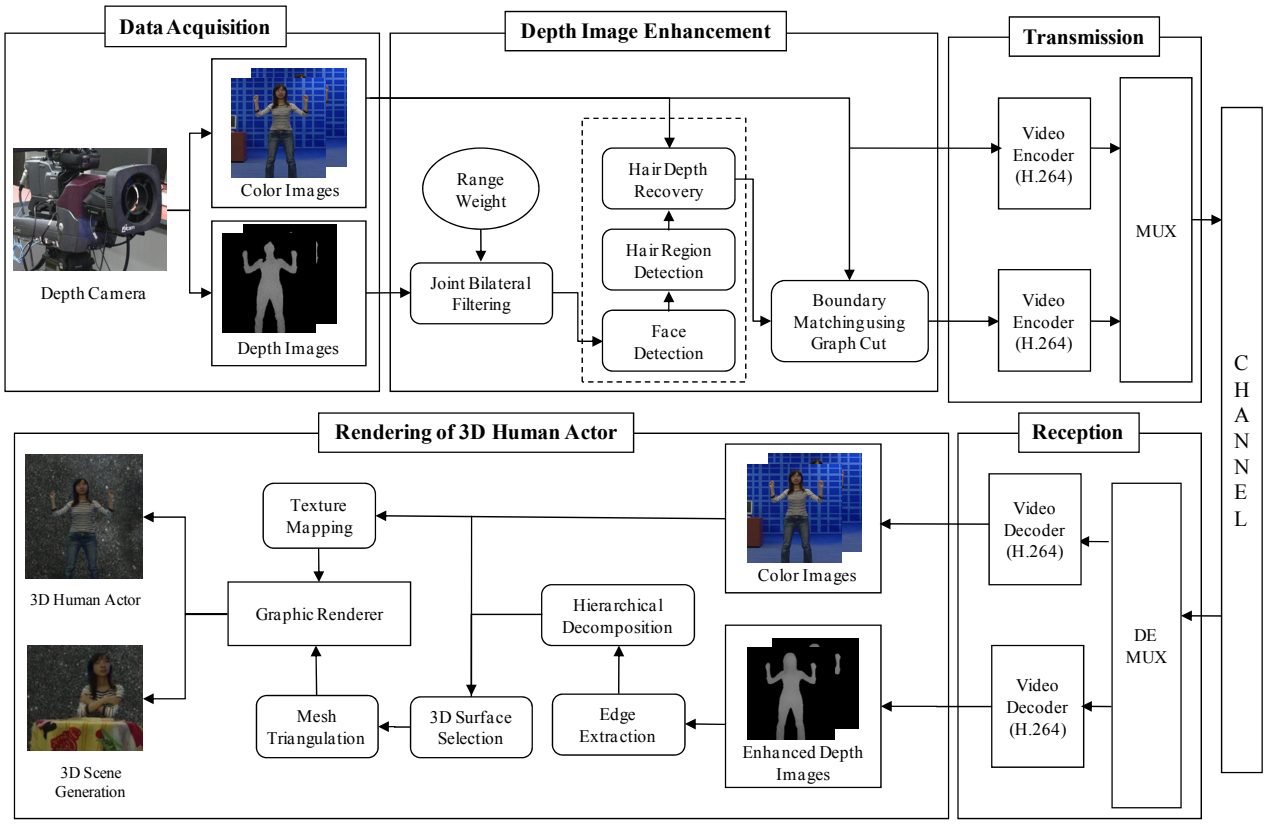
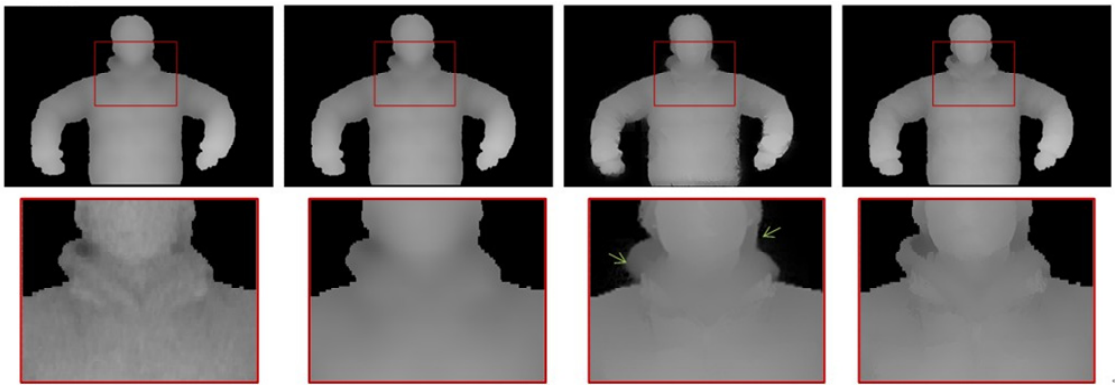


Fig.2. Overall system architecture for generating a 3D human actor



(a) Original depth map (b) Bilateral filtering (c) Joint bilateral filtering (d) Our filtering

Fig.3. Joint bilateral filtering to reduce depth noise.

coder, and transmitted through a channel to the receiver side, respectively.

At the receiver side, the transmitted color and depth images are decoded, and the 3D surface of a human actor is generated by a mesh triangulation algorithm based on hierarchical decomposition of depth images [14]. Finally, we create a dynamic 3D human actor by covering its 3D surface with the color information provided by texture mapping.

B. *Noise Reduction of Depth Image*

In order to reduce noise of the raw depth image, we apply a joint bilateral filtering method [15]. The bilateral filter reduces the noise while preserving important sharp edges. Formally, for some position p of image I, the filtered value J_p is represented by Eq (1).

$$J_p = \frac{1}{k_p}\sum_{q\in\Omega} G_s\left(\|p-q\|\right)\cdot G_r\left(\|I_p - I_q\|\right)I_q \qquad (1)$$

where G_s and G_r are the space weight and the range weight, respectively. The term of Ω is the spatial support of the weight G_s, and K_p is a normalizing factor. The range weight G_r prevents from averaging across the edges.

In joint bilateral filtering [16], the range weight G_r is computed from a different image I' by Eq. (2).

$$J_p = \frac{1}{k_p}\sum_{q\in\Omega} G_s\left(\|p-q\|\right)\cdot G_r\left(\|I'_p - I'_q\|\right)I_q \qquad (2)$$

In the previous work, a flash image can be used as an image I'. The depth discontinuity in a depth image usually corresponds to the edges in its corresponding color image. Therefore, we apply a newly-designed joint bilateral filter onto the depth image with the range weight from the color data represented in Eq. 3.

$$J_p = \frac{1}{k_p}\sum_{q\in\Omega} G_s\left(\|p-q\|\right)\cdot G_{r1}\left(\|I'_p - I'_q\|\right)\cdot G_{r2}\left(\|I_p - I_q\|\right)I_q \qquad (3)$$

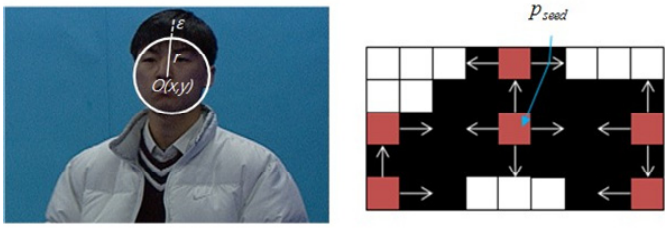
where G_{r1} takes color difference in RGB space, and G_{r2} uses depth difference. Figure 3 shows the comparison of three different filtering methods. As shown in Fig. 3, our method reduces the noise of a depth image efficiently while preserving some important sharp features without any unintended visual artifacts.

C. *The Recovery of Lost Hair Region*

Another main drawback of a raw depth image is the absence of depth value for shiny and dark surface regions. The depth image of a human actor with black hair captured by the depth camera often does not show the hair region, because the camera cannot capture shiny and dark surfaces. Since we set our objective to generate a 3D human actor, the loss of hair region becomes a critical problem.

In this paper, we introduce a novel method that recovers the lost hair region. Previous techniques cannot be directly applied to our depth image, because they are based on neighbor information of the lost region [17, 18]. The proposed recovery algorithm consists of three steps: detection of the hair region, recovery of the boundary, and estimation of the hair shape. In this work, we use the recorded color image to detect the lost hair region.

For simplicity, we assume that the loss of depth data only occurs in the hair region. First, we find out the face region in a color image using the face detector [19]. Figure 4 illustrates the result of face detection. The initial seed point p_{seed} is found by using radius r and offset ε as shown in Fig. 4(a). When the coordinate of the center of face O is (x, y), the coordinate of the initial seed point, p_{seed}, is $(x, y+r+\varepsilon)$.



(a) Face detection (b) Multi-seed region growing

Fig.4. Detection of hair region

In order to search the hair region efficiently, we generate a multiple number of seed points near the initial seed point as shown in Fig. 4(b). Using a region growing algorithm based

on a multiple number of seed points, the black hair region can be detected. Next, the boundary of the lost hair region is recovered from the final silhouette. For estimating the depth value of the lost hair region, its boundary must be estimated. The boundary contour of the final silhouette is generated by a chain code method. The generated contour is represented by a color boundary list (CBL) sorted clockwise. We assign the depth value to all elements of the CBL considering its depth image; however, the depth value of the lost hair boundary region is zero, since the hair region is lost in the original depth image.

Two points, p_f and p_l, are chosen from the CBL, where p_f represents the first pixel with zero depth in the clockwise order, and p_l represents the last pixel with zero depth value. Therefore, all depth values of each pixel between p_f and p_l are zeros. In general, since the depth values of p_{f-1} and p_{l+1} are not the same, from p_f to p_l, the depth value of each point is calculated by linear interpolation using Eq. 4.

$$p_i.z = (1-\alpha)p_{f-1}.z + \alpha p_{l+1}.z, i = 0,\ldots,n \tag{4}$$

where n is the number of elements from p_l to p_f and $\alpha = i/n$.

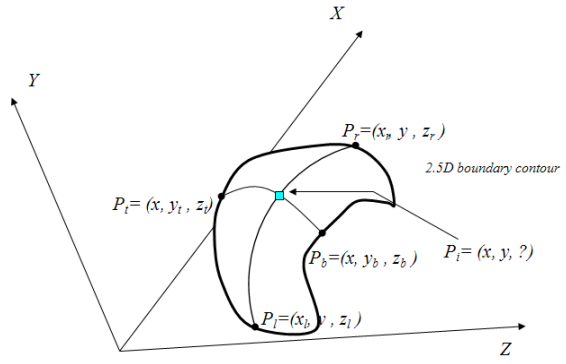


Fig. 5. Interpolation of hair region using quadratic Bézier curve.

Finally, we fill up the inside of the hair region. When we apply a chain code method to the silhouette of the acquired depth image, another contour list, depth boundary list (DBL), is generated. To get an outer boundary of the hair region, we apply the exclusive-or (XOR) on the two contour list CBL and DBL. In the previous step, we have assigned the depth value in the upper part of the hair boundary. The lower part of the hair boundary has its depth value, because it is extracted from the acquired depth image. Hence, the entire boundary now has depth information.

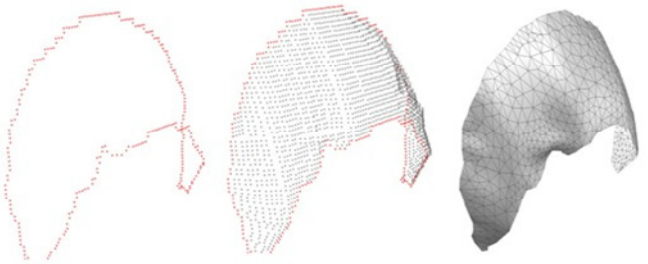
As the human head follows an elliptical shape, the linear interpolation technique returns an unnatural shape. To obtain a more natural shape, a quadratic Bézier curve that uses three control points is applied for interpolation [20]. Figure 5 shows how the inside region is interpolated using a quadratic Bézier curve. When certain point p_i is located in x, y position with depth value zero, the point p_i is computed by using Bézier curve with four points: p_l, p_r, p_t, p_b:

$$B_1(t_1) = (1-t_1)^2 p_b + 2t_1(1-t_1)p_{c1} + t_1^2 p_t \tag{5}$$

$$B_2(t_2) = (1-t_2)^2 p_l + 2t_2(1-t_2)p_{c2} + t_2^2 p_r \tag{6}$$

$$P_i = B(t_1,t_2) = \frac{B_1(t_1) + B_2(t_2)}{2} \tag{7}$$

where p_{c1} and p_{c2} is the control points of a quadratic Bézier curve. Figure 6 shows the result of recovered hair region.



(a) No interpolation (b) Interpolation result (c) Mesh generation

Fig. 6. Recovery of the lost hair region.

D. The Boundary Cleaning

When we render a 3D human actor using a color and a depth image as shown in Fig. 7(a) and Fig. 7(b), the corresponding color image is used as the texture for the actor. Since the depth camera does not catch exact boundary of a real object, it causes visual artifact as shown by Fig. 7(c).

(a) Color image (b) Depth image (c) Artifacts in boundary

Fig. 7. Artifacts in object boundary regions.

In this paper, we compensate the boundary of a human actor using color information. First, we make a trimap by the depth image as shown in Fig. 8(a). The trimap generation consists of three steps: thresholding, erosion, and dilation. The depth image represents depths of the foreground object only as shown in Fig. 8(b). We convert the depth image into a binary mask image M by simple thresholding. In general, all unknown areas belong to the region that is the border of a foreground object and a background. Hence, a definite foreground area T_F can be erosion of M, a definite background T_B can be inverse of dilation of M, and the unknown area T_U is inverse of $T_F \cup T_B$.

Now, we have the color image and corresponding trimap image. Accurate boundary information of the foreground object can be estimated via a graph-cut segmentation algorithm. The algorithm that we used is briefly described as below:

1) Initialize two sets of Gaussian Mixture Model (GMM) from the definite foreground and background region.

2) Construct a graph G = (V, E) with a set of nodes V and a set of weighted edges E. Each node in the graph corresponds to a pixel in the image. Add two virtual nodes, one for the foreground, and another for the background. The weight of edges connecting nodes and virtual nodes are computed from GMM. The weight of edges linked neighborhood nodes are computed by the difference of two color vectors of the corresponding pixel.

3) Minimize the energy via min-cut algorithm [21].

(a) Trimap (b) Refined boundary (c) Reduced artifacts

Fig. 8. Boundary refinement with color and depth images.

All pixels in unknown region are classified into a foreground or a background. Figure 8(c) shows the boundary cleaning result using graph-cut segmentation

E. Rendering of a Dynamic 3D Human Actor

For rendering the 3D human actor, we employ a DIBR technique using meshes. In this paper, we utilize the hierarchical decomposition of a depth image to render the human actor [14]. In hierarchical decomposition, we divide the region of a depth image uniformly using grid cells as shown in Fig. 9, and remove the grid cells including no depth information. We classify remained grid cells into two categories according to the existence of edge information: edge-grid cells and no-edge-grid cells. In order to maintain the visual quality of a 3D human model, we set the size of a grid cell as 4×4 or 8×8.

When we divide an edge-grid cell into four sub-grid cells and the number of the sub-grid cells including edge information is more than one, we use a full modeling mode. Otherwise, we select one of quad-tree modes according to the sub-grid cell including edge information: up-left, up-right, down-left, and down-right quad-tree modes.

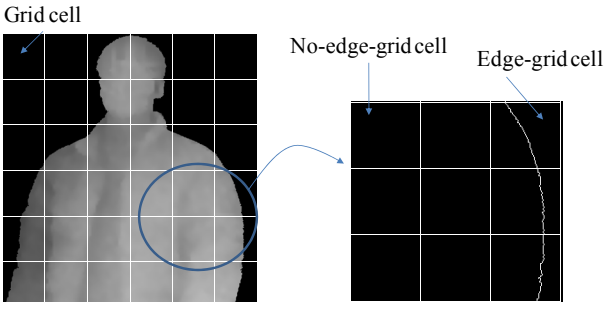


Fig. 9. Hierarchical decomposition of depth images

When the full modeling model is employed, we generate the 3D surface of an edge-grid cell with 44 triangles and 25 pixels extracted from a depth image. However, since we do not need to render the region of no depth information, we ignore the triangle of which one of vertices is on the region. As a result, we render the 3D surface of the edge-grid cell with less than 44 triangles. Likewise, we represent the 3D surface of the edge-grid cell with less than 20 triangles and 14 pixels in a quad-tree mode. For each no-edge-grid cell, we generate its 3D surface with 2 triangles and 4 pixels.

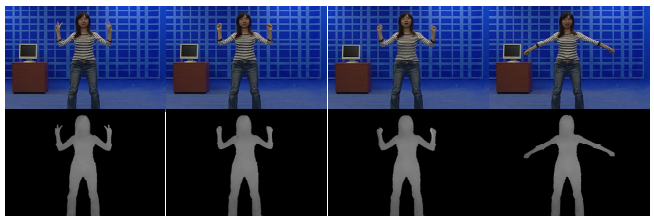
TABLE 1
GENERATION OF 3D MESH TRIANGULATION

Mode	# of pixels	# of triangles	Shape of surface
No-edge-grid Cell	4	2	
Up-left Quad-tree	14	20	
Down-left Quad-tree	14	20	
Up-right Quad-tree	14	20	
Down-right Quad-tree	14	20	
Full modeling	25	44	

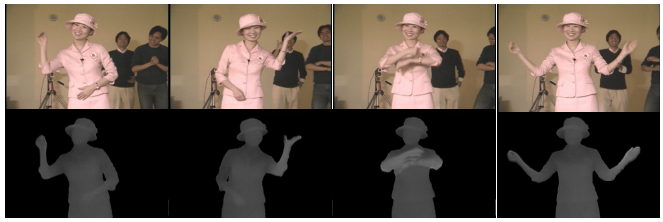
III. EXPERIMENTAL RESULTS

We have tested the performance of our proposed 3D human actor generation method using four different actors. In this experiment, we employed a depth camera system [11]. The first actor is captured by a close-up shot using a depth camera and the others are captured by a long shot. The image resolution of test color and depth images is 720×486.

The test image for the 1st and the 2nd actor is composed of single frame. On the other hand, the test image for the 3rd and the 4th actor is composed of 100 frames, respectively. Especially, since the 4th actor wears a hat and there is no depth absence in its depth images, it is not needed to recover depth information of the hair region for the 4th actor. The others needed depth recovery of hair region. For 3D human actor rendering, we set the size of grid cell as 4×4. Figure 10 shows the test image for the 3rd and the 4th actors.



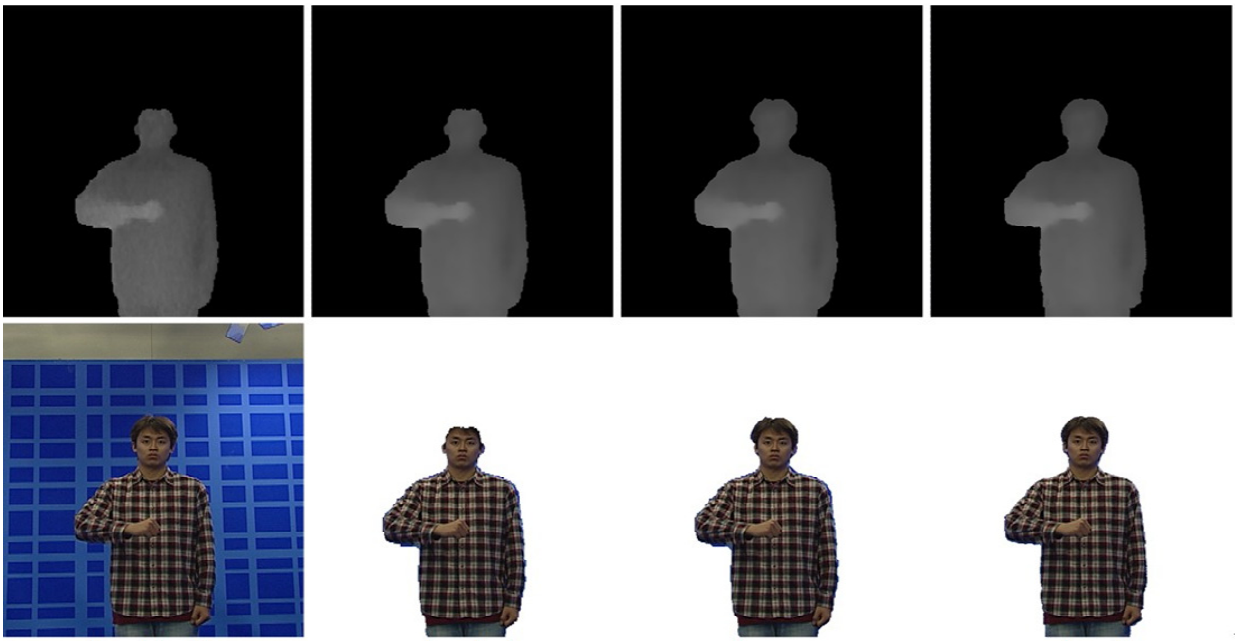
a) Test image of the 3rd actor



b) Test image of the 4th actor

Fig. 10. Test images captured by the depth camera

a) Depth image enhancement for the 1st actor



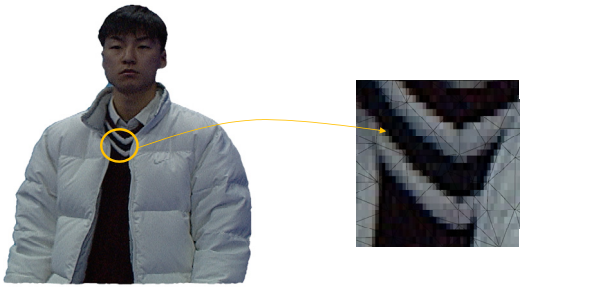
b) Depth image enhancement for the 2nd actor

Fig. 11. Depth image enhancement results

Figure 11(a) and Figure 11(b) show the result of depth image enhancement for the 1st and the 2nd actor images, respectively. The first column of the figure indicates the raw images of both color and depth captured by the depth camera, and the second column shows the result of a human actor after the reduction of optical noise. The third column shows the results of a human actor after optical noise reduction and hair recovery. The last column is the final 3D human actor that has completed the boundary refinement. As shown in Fig. 11, we could successfully resolve three main problems existed in the raw depth image.

Figure 12 shows the result of a 3D human actor after a mesh triangulation. As shown in Fig. 12(a), since we reconstructed the 3D surface of the human actor differently according to the existence of edges in the depth image, we could render the human actor fast and maintained the significant surface dramatically. Furthermore, as shown in Fig. 12(b), the 3D human actor can be viewed successfully

when the viewer changes the angle of the viewpoint significantly.



3D rendering result Wireframe of the rendered 3D actor

a) 3D human modeling for the 1st actor

- 15 degree 0 degree + 15 degree

b) Changing viewpoint

Fig. 12. 3D human actor generation using mesh triangulation

Figure 13 shows the rendering results for the test image of the 3rd actor. Since the depth information on its hair region is recovered and the boundary of the actor is refined, the 3D human actor generated by the proposed method shows much better visual quality than the one generated from raw depth images.

The 1st frame The 15th frame The 50th frame

(a) 3D human actor generated by raw depth images

The 1st frame The 15th frame The 50th frame

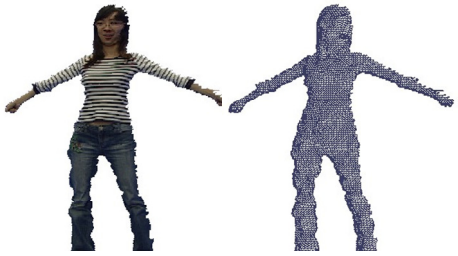
(b) 3D human actor generated by enhanced depth images

Fig. 13. 3D human actor generation for the 3rd actor

In order to compare the visual quality of 3D human actors generated by raw depth images and enhanced depth images, we show the wire frame of the 50th frame of the 3rd human actor. As shown in Fig. 14, since we reduce the optical noise in raw depth images, the 3D surface of the 3D human actor generated by the proposed method show much smoother and natural compared to the one generated from raw depth images.

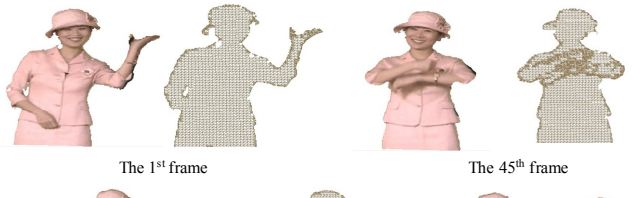
Figure 15 shows the result and wire frame of the dynamic 3D human actor generation for the 4th actor.

(a) Wire frame of 3D human actor generated by raw depth image



(b) Wire frame of 3D human actor generated by enhanced depth image

Fig. 14. Visual quality comparison



The 1st frame The 45th frame

The 75th frame The 99th frame

Fig. 15. 3D human actor generation for the 4th actor

In addition, Table 2 shows the result of average computation time for each frame to generate the 3D human actor with the test images of the 3rd and the 4th actor. In this experiment, we generate and render the 3D human actor in about 0.7 ~ 0.9 sec/frame. We expect to reduce the computation time when we optimize our algorithm of depth image enhancement and employ fast rendering technique that uses graphics processing unit (GPU).

TABLE 2
THE COMPUTATION TIME FOR GENERATING A 3D HUMAN ACTOR

Processing	The 3rd actor	The 4th actor
Noise reduction	0.156 sec/frame	0.109 sec/frame
Hair recovery	0.185 sec/frame	0.125 sec/frame
Boundary refinment	0.360 sec/frame	0.250 sec/frame
3D mesh generation	0.212 sec/frame	0.257 sec/frame
Total	0.913 sec/frame	0.741 sec/frame

IV. CONCLUSION

In this paper, we proposed a novel scheme for generating a dynamic 3D human actor using a depth camera. We resolved major problems that exist in the raw depth image which includes optical noise, loss of depth data for hair region, and

boundary mismatch. Experimental results show that our method produces a natural and realistic 3D human actor model successfully. We expect the proposed 3-D human actor generation system to be used in the next-generation 3D multimedia applications.

ACKNOWLEDGMENT

This research was supported by the Korea Science and Engineering Foundation (KOSEF) through the National Research Lab. Program funded by the Ministry of Education, Science and Technology (No. R0A-2007-000-20049-0) and the Ministry of Knowledge Economy, Korea, under the Information Technology Research Center support program supervised by the Institute of Information Technology Advancement. (No. IITA-2008-C1090-0801-0017).

REFERENCES

[1] P. Debevec, A. Wenger, C. Tchou, A. Gardner, J. Waese, and T. Hawkins, "A Lighting Reproduction Approach to Live-action Compositing," *Proc. of SIGGRAPH*, pp.547-556, 2002.

[2] Motion Capture Systems from Vicon Peak, http://www.vicon.com, 2005.

[3] S.Y. Lee, I.J. Kim. S.C. Ahn, and H.G. Kim, "Active Segmentation for Immersive Live Avatar," IET Electronics Letters, vol. 40, no. 20, pp. 1257-1258, 2004.

[4] A. Ignatenko and A. Konushin, "A Framework for Depth Image-Based Modeling and Rendering," *Proc. of Graphicon*, pp. 169-172, 2003.

[5] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality Video View Interpolation Using a Layered Representation," *Proc. of SIGGRAPH*, pp. 600-608, 2004.

[6] A. Hengel, A. Dick, T. Thormahlen, B. Ward and P.H.S. Torr, "VideoTrace: Rapid interactive scene modelling from video," *Proc. of SIGGRAPH*, article 86, 2007

[7] R. Zhang, P.S. Tsai, J.E. Cryer, and M. Shah, "Shape from Shading: A Survey," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 690-706, 1999.

[8] S.K. Nayar and Y. Nakagawa, "Shape from Focus," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no. 8, pp. 824-831, 1994.

[9] S. Zhang and P. S. Huang, "High-resolution, Real-time Three-Dimensional Shape Measurement," *Optical Engineering*, vol. 45, no. 12, paper no. 123601, pp. 1-8, 2006.

[10] M. Waschbüsch, S. Würmlin, D. Cotting, and M. Gross, "Point-sampled 3D Video of Real-world Scenes," *Signal Processing:Image Communication*, vol. 22, no. 2, pp. 203-216, 2007.

[11] G.J. Iddan and G. Yahav, "3D Imaging in the Studio and Elsewhere…," *Proc. of SPUE Videometrics and Optical Methods for 3D Shape Measurements*, pp. 48-55, 2001.

[12] M. Kawakita, T. Kurita, H. Kikuchi, and S. Inoue, "HDTV Axi-vision Camera," *Proc. of International Broadcasting Conference*, pp. 397-404, 2002.

[13] S.M. Kim, J. Cha, J. Ryu, and K.H. Lee, "Depth Video Enhancement for Haptic Interaction Using a Smooth Surface Reconstruction," *IEICE Trans. on Information and Systems*, vol. E89-D, no. 1, pp.37-44, 2006.

[14] S.Y. Kim, S.B. Lee, and Y.S. Ho, "Three-Dimensional Natural Video System Based on Layered Representation of Depth Maps," *IEEE Trans. on Consumer Electronics*, vol. 52, no. 3, pp. 1035-1042, 2006.

[15] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images, " *Proc. of International Conference on Computer Vision*, pp. 839-846, 1998.

[16] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe and K. Toyama, "Digital Photography with Flash and No-flash Image Pairs," *Proc. of SIGGRAPH*, Vol. 23, No. 3, pp. 664-672, 2004.

[17] L. A. Torres-Ménez, G. Dudek, and P. D. Marco, "Inter-Image Statistics for Scene Reconstruction," *Proc. of Canadian Conference on Computer and Robot Vision*, pp. 432-439, 2004.

[18] J. Davis, S. Marschner, M. Garr, and M. Levoy, "Filling Holes in Complex Surfaces using Volumetric Diffusion," *Proc. of International Symposium on 3D Data Processing, Visualization, Transmission*, pp. 428-438, 2002.

[19] OpenCV library, http://www.intel.com/technology/computing/opencv/.

[20] D. Marsh, *Applied Geometry for Computer Graphics and CAD*, Springer, 1999.

[21] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts", Proc. Of International Conference on Computer Vision, 1999, pp. 377-384.

Ji-Ho Cho received the MS degree in Information and Communications in 2005 from Gwangju Institute of Science and Technology in Gwangju, Korea, and worked during the following eight months as a academic guest at Swiss Federal Institute of Technology ETHZ in Zürich. He recently started working on his PhD degree at the Intelligent Design and Graphics Laboratory at GIST. His main interests lie in 3D Video and computational photography.

Sung-Yeol Kim received his B.S. degree in Information and Telecommunication engineering from Kangwon National University, Korea, in 2001, and M.S. and Ph.D degree in Information and Communication Engineering at the Gwangju Institute of Science and Technology (GIST), Korea, in 2003 and 2008, respectively. His research interests include digital signal processing, video coding, 3D mesh representation, 3D mesh compression, 3D television, and realistic broadcasting.

Yo-Sung Ho received both B.S. and M.S. degrees in electronic engineering from Seoul National University, Korea, in 1981 and 1983, respectively, and Ph.D. degree in Electrical and Computer Engineering from the University of California, Santa Barbara, in 1990. He joined the Electronics and Telecommunications Research Institute (ETRI), Korea, in 1983. From 1990 to 1993, he was with Philips Laboratories, Briarcliff Manor, New York, where he was involved in development of the advanced digital high-definition television (AD-HDTV) system. In 1993, he rejoined the technical staff of ETRI and was involved in development of the Korea direct broadcast satellite (DBS) digital television and high-definition television systems. Since 1995, he has been with the Gwangju Institute of Science and Technology (GIST), where he is currently a professor in the Information and Communications Department. His research interests include digital image and video coding, image analysis and image restoration, advanced coding techniques, digital video and audio broadcasting, 3D television, and realistic broadcasting.

Kwan H. Lee received his M.S. and Ph.D degree at North Carolina State University in 1985 and 1988 respectively. He worked as an assistant professor at Northern Illinois University from 1988 to 1994. He is a professor in the Mechatronics Department at Gwangju Institute of Science and Technology (GIST) since 1995. His research interests are in CAD and computer graphics, that include geometric modeling, photorealistic rendering, reverse engineering and rapid prototyping. Currently he is the director of the Immersive Contents Research Center at GIST and focuses on his research in immersive modeling and realistic material modeling.