VR@Home: An Immersive Contents Creation System for 3D User-Generated Contents*

Wonwoo Lee[1], Chinta Ram Babu[1], Jongweon Lee[2], and Woontack Woo[1]

[1] GIST U-VR Lab.
Gwangju 500-712, S. Korea
{wlee,crambabu,wwoo}@gist ac.kr
[2] Department of Digital Contents, Sejong University.
Seoul 143-747, S. Korea
{jwlee}@sejong ac.kr

Abstract. In this paper, we propose VR@Home, an immersive contents creation system for 3D User-Generated Contents(UGC). VR@Home system enables users to create realistic 3D contents for themselves and to interact with the contents. It also augments users in the virtual environment. The four components of VR@Home system, image-based modeling, real-time user segmentation, real-time soft shadow generation, and tangible user interface, provide solutions to the challenges emerging from the limitations of home environment. The proposed system is not only for 3D UGC creation, but also can be applied to interactive on-line education, personal broadcasting, and entertainment such as video games.

Keywords: User-Generated Contents, Image-based modeling, Background subtraction, Real-time soft shadow generation.

1 Introduction

Traditionally, people have just accepted the contents produced by contents production companies. There are not much convenient ways of participation to the contents production. With emerging Internet technologies, people do not remain as contents consumers anymore nowadays. Instead of being a passive contents consumer, each of them becomes an active producer of multimedia contents called User-Generated Content (UGC) or User-Created Content (UCC). In addition, UGC is shared through the Internet and it emerges as one of the important sources of the Internet contents.

The current UGCs, however, have limitations from the current computing environment. Firstly, Most of them are two-dimensional contents such as pictures, flash animations, and video clips. Two-dimensional contents do not provide enough immersion to users, since the contents contain non-realistic components and they are lack of 3D depth. Secondly, the current UGCs hardly provide interaction with users. The users can perform simple interactions with contents by mouse clicking and

* This research is funded by ETRI OCR.

K.-c. Hui et al. (Eds.): Edutainment 2007, LNCS 4469, pp. 81–91, 2007.
© Springer-Verlag Berlin Heidelberg 2007

keyboard typing. The reactions of the contents are pre-defined so that they can only provide the limited reactions to users' input.

To overcome these problems of the current contents, a concept called 'u-Contents' has been proposed as a new type of contents [1]. Considering an application for education or entertainment, the three characteristics of u-Contents, u-Realism, u-Intelligence, and u-Mobility, have their advantages as follows.

- **u-Realism:** Realistic contents can contribute to enhancement of a user's experience. Three-dimensional contents can provide additional information by enabling a user to see the contents from different viewpoint. Multimodal feedback provides more immersion and interactivity to users.
- **u-Intelligence:** Intelligence embedded in contents provides personalized reaction to users. This feature enables contents to give different feedback to users according to the user's context. Thus, it is possible to develop the user-adaptive applications.
- **u-Mobility:** Considering the ubiquitous computing environment as the future computing environment, the u-mobility of the contents among different devices let a user utilize the application anytime and anywhere. The contents with mobility can be free from the limitation of the hardware platform.

In this paper, we propose VR@Home system, an immersive contents creation system for 3D UGCs. Among the three characteristics of u-Contents, we focus on the realism of the created contents in this work. VR@Home system enables users to create realistic 3D contents for themselves and to interact with the contents. It also augments users in a virtual environment to enhance users' experience. VR@Home system consists of four components, 3D contents creation, real-time user segmentation, real-time soft shadow generation and tangible user interface. 3D contents creation enables users to create 3D contents easily through an image-based modeling method. A user can make a 3D model of an object by capturing its images from multiple viewpoints. Real-time user segmentation separates a user from a natural background scene to augment users in virtual space without a special hardware. For better composition of virtual contents, soft shadows are inserted to the scene in real-time. Tangible user interfaces let a user intuitively interact with the contents he/she created. Fig. 1 shows the flow diagram of VR@Home system.

Our image-based modeling method is fast with hardware acceleration and designed not to require a user's intervention during the modeling process. The real-time user segmentation detects highlights and shadows for robustness to lighting condition change in the background. We update the background model adaptively with the small changes in the background according to time. The real-time soft shadow generation method generates soft shadows from multiple hard shadows for real-time performance. The vision-based tangible user interface provides intuitive interaction. The proposed system can be applied to interactive on-line education, personal broadcasting, and entertainment such as video games.

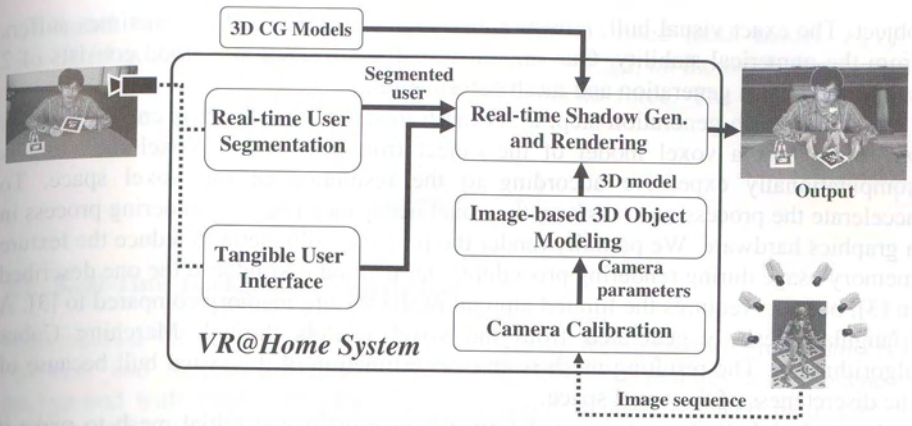

Fig. 1. Flow diagram of VR@Home

This paper is organized as follows. In chapter 2, we describe each of the four components of VR@Home system in detail. System implementation and experimental results are given in chapter 3. We provide conclusions and future works in chapter 4.

2 VR@Home System

In this chapter, we explain each component of VR@Home system. Since our system aims at running on home computers, there exist challenges emerging from the limitation of home environment compared to the controlled environment in a research laboratory. In each section, we first describe the challenges and explain our approach to solve them.

2.1 Realistic 3D Contents Creation

Creating 3D contents for virtual reality or augmented reality requires the knowledge about 3D CG modeling tools and it is time consuming work even for the people who are accustomed to use the tools such as Maya or 3DS Max. From this viewpoint, image-based modeling can be a alternative solution. It requires a sequence of images as the input data and less intervention of users for 3D object reconstruction. A 3D model created by image-based modeling techniques provides good realism compared to the one created by 3D CG modeling tools.

In VR@Home system, we provide an automatic way of smooth visual hull reconstruction method for creation of realistic 3D contents. Our image-based modeling method uses a sequence of images captured around the object. We get the instrinsic and extrinsic parameters of the camera through camera calibration. After camera calibration, the silhouette of the object is extracted. For silhouette extraction, we exploit a segmentation method based on color difference in normalized RGB space. The normalized RGB color model provides a better way of the measurement of color difference than RGB color model [2]. Then, we reconstruct a smooth visual hull of the object with silhouettes. It is possible to reconstruct the exact visual hull of an

object. The exact visual hull, however, has jagged surfaces and it sometimes suffers from the numerical stability. Our smooth visual reconstruction method consists of 2 step, initial mesh generation and mesh deformation.

In initial mesh generation step, a 3D mesh model of the object is created. Firstly, we reconstruct a voxel model of the object from silhouettes. Voxel modeling is computationally expensive according to the resolution of the voxel space. To accelerate the process we transform the voxel occupancy test to a rendering process in a graphics hardware. We partially render the projected silhouette to reduce the texture memory usage during rendering procedure. Our method is similar to the one described in [3], but ours requires the limited amount of the texture memory compared to [3]. A triangular mesh is generated from the voxel models through Marching Cubes algorithm [4]. The resulting mesh is an over-estimation of the visual hull because of the discreteness of the voxel space.

In mesh deformation step, we deform the over-estimated initial mesh to make it consistent with the input silhouettes. A force applied to a vertex of the mesh model as the scaled sum of silhouette force $F_s(V)$ and internal force $F_i(V)$.

$$F(V) = \alpha F_s(V) + \beta F_i(V) \cdot \tag{1}$$

where, α and β are scale factors.

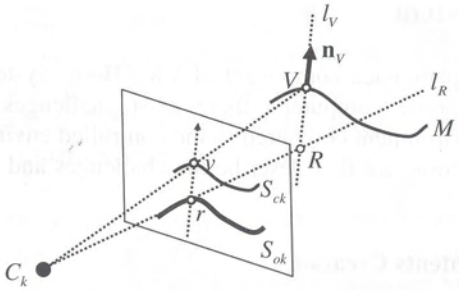

Fig. 2. Calculation of silhouette force

Silhouette force enforces silhouette consistency with the original input silhouette to the resulting deformed mesh model. For a vertex V in 3D, we project it to camera C_k and obtain an image point v. If v is not on the current silhouette S_{ck}, which is the projection of the current mesh model onto the camera C_k, the silhouette force is set to zero. Otherwise, we calculate the intersection r of the line passing through v with the contour of original silhouette S_{ok}. To make both silhouettes identical, we move V to the 3D point R which is the intersection of the line l_V and l_R. The silhouette force associated with the camera C_k is calculated as equation 2. $F_{sk}(V)$ is calculated for all the cameras which sees V without occlusion. The total silhouette force $F_s(V)$ on V is the average of $F_{sk}(V)$.

$$F_{sk}(V) = dist(R,V) \cdot \mathbf{n}_V \tag{2}$$

where, $dist(R,V)$ is the distance between R and V and $\mathbf{n}_V$ is the normal vector of V.

Internal force $F_i(V)$ is for smoothness of the deformed mesh model. $F_i(V)$ is calculated as vector difference with adjacent vertices $V_{adj(k)}$ on the mesh. It prevents a vertex from too much deformation by the silhouette force. The Internal force is defined as the equation 3.

$$F_i(V) = \frac{1}{n}\sum_{k=1}^{n}\left(V - V_{adj(k)}\right) \tag{3}$$

2.2 Real-Time Background Subtraction

Conventional VR Studio system uses chroma-keying technique to segment objects in real-time and to augment them in a virtual scene. It requires a homogeneous background with a specific color, such as green or blue and special hardwares. In addition, the lighting condition of the studio should be controlled precisely for best segmentation result. Considering the home environment where VR@Home system is supposed to be used, the space is not large enough for the conventional segmentation system and lighting condition is hardly controllable. Thus, the conventional segmentation approach is not proper from a practical viewpoint.

Several methods are proposed for object segmentation from a natural background [5-7]. However, they suffer from camera noise and lighting condition changes in the background caused by a moving object, especially a user in our context. To achieve robust real-time user segmentation from a natural background, we provide a hybrid method based on background subtraction that preserves the accurate boundary of moving object. Each color channel of HSV color model processed independently in order to improve the accuracy of segmentation results. Fig. 3 shows the overall structure of the proposed background subtraction. The proposed method consists of five sequential stages.

At first, we determine the background model from certain number of static background frames, before the objects move in. In the background modeling step, the background pixels are modeled as Gaussian distribution $N(\mu, \sigma^2)$ with mean μ and variance σ^2, are updated recursively during the background training. For the i-th frame, the mean and variance is determined as follows.

$$\begin{array}{lllll} \mu_0 = I_0; & i=0 & \sigma^2{}_1 = (I_1-\mu_0)^2 & & i=1 \\ \mu_i = (1-\alpha)\mu_{i-1}+\alpha I_i & i\geq 1 & \sigma^2{}_i = (1-\alpha)\sigma^2{}_{i-1}+\alpha(I_i-\mu_i)^2 & & i\geq 2 \end{array} \tag{4}$$

where, α is the learning rate of the linear model.
The mean of the pixels at (x,y) in the N continuous frames can be formulated as

$$B(x,y) = \left[\mu_i(x,y)\middle| i = N\right] \tag{5}$$

Similarly, the N frame standard deviation can be obtained by

$$\sigma_N(x,y) = \left[\sqrt{\sigma_i^2(x,y)}\middle| i = N\right] \tag{6}$$

where, $B = <B^H, B^S, B^V>$ and $\sigma_N = <\sigma^H{}_N, \sigma^S{}_N, \sigma^V{}_N>$.

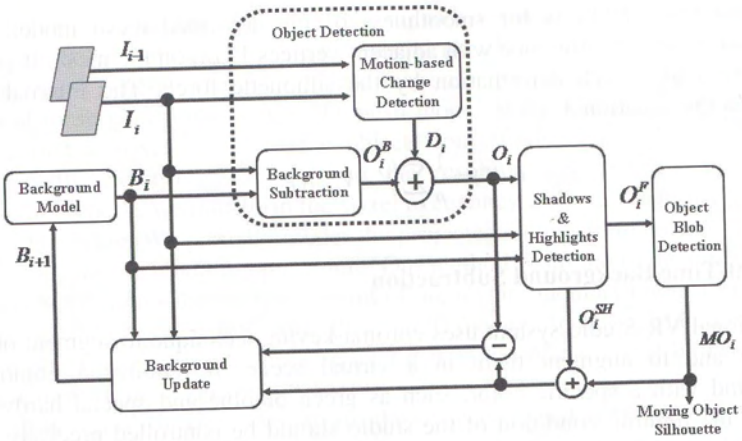

Fig. 3. Real-time background subtraction algorithm

In the background subtraction step, first, the incoming frame is compared with the background model. Background subtraction is performed for every pixel $I_i(x,y)$ by calculating Mahalanobis distance δ through the following equation

$$\delta\ (x,y) = |I_i(x,y) - B(x,y)| \tag{7}$$

Then, the binary object map $O^B{}_i$ is generated. Similarly, we perform motion-based change detection where the difference D_i between the current and previous frame, is compared with the mean of the difference image by using the normalized standard deviation of the frame deference. This step preserves the boundary of the moving object and also detects the ghosts, the regions that have small motion. Finally we perform the logical OR operation between the results from both the stages. The binary mask which representing the moving object regions can be generated as $O_i = D_i \vee O_i^B$.

We perform the shadow and highlight detection step where the shadow and highlight pixels are detected from the foreground mask by using the normalized luminance, and background subtraction of Hue and Saturation component. The pixels belonging to shadows or highlights have similar chromaticity with the background but lower brightness for shadows and higher brightness for highlights. With these properties we classify the foreground pixels of O_i

As post-processing, we perform blob detection procedure for detecting the blobs in an object binary mask O_i which is acquired from the previous stage. Firstly, we apply 3x3 median filtering to reduce the salt and pepper noise, and then, marker extraction step is performed to locate the blobs. Then, the geodesic dilation is applied on the extracted maker. The proposed post-processing is able to identify the object blob and tried to eliminate the noise blobs. Finally, we update the background model with incoming images. The background update procedure is different for pixel positions which are detected as belongs to foreground, as ghosts and part of the background. The entire procedure will be repeated for every incoming frame.

2.3 Real-Time Soft Shadow Generation

For realistic composition of 3D contents with background, we create soft shadow under specified lighting condition. Rendering soft shadows sometimes requires high computation. Since VR@Home system is designed to work in real-time on a home PC, we apply a simplified approach based on mesh transformation to it for a reasonable result.

Our mesh scaling method renders multiple simple hard shadows by using shadow volume method to create penumbra and umbra. When we render a hard shadow on the scene, the 3D model which makes the shadow is scaled. The rendered multiple hard shadows create a stratified shadow. Fig. 4 shows the concept mesh scaling method.

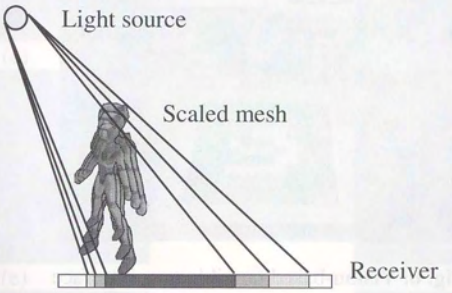

Fig. 4. Mesh scaling approach for soft shadow generation

After the stratified shadow generation, we obtain shadow map of the stratified shadow. Then, we blur the obtained shadow map by using low-pass filter to create a final soft shadow of the object. Fig. 5 shows the shadow map before and after blurring the stratified shadow. Finally, the soft shadow map is applied to the 3D scene.

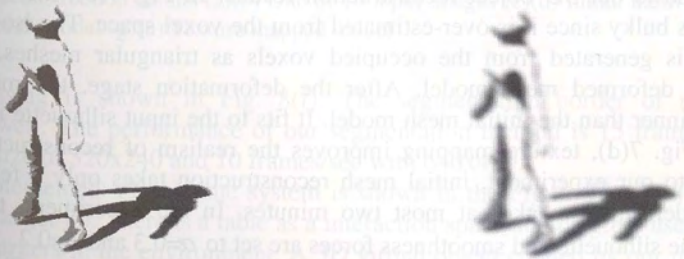

Fig. 5. Soft shadow map generation from stratified shadow (a) Shadow map obtained from initial stratified shadow (b) Soft shadow map after blurring

2.4 Tangible User Interface

In most virtual reality applications, different types of tracking devices have been used for interaction with virtual contents. Commercial products are available on the market

[8,9]. Most tracking devices, however, are too expensive to use personal purpose and they are not designed for home environment. Wired devices make users feel fatigue when they wear the devices for a long time. In contrast, a vision-based user interface does not need additional hardware, except a camera. Since there are no wires attached to a user, the user can have more freedom than the hardware-based interfaces.

In VR@Home, we adopt a vision-based tangible user interface. Tangible user interface lets users feel the existence of virtual contents. Thus, they can interact with virtual contents intuitively. We use multiple markers are used for robust 3D interaction. Fig. 6 shows tangible objects used in VR@Home system.

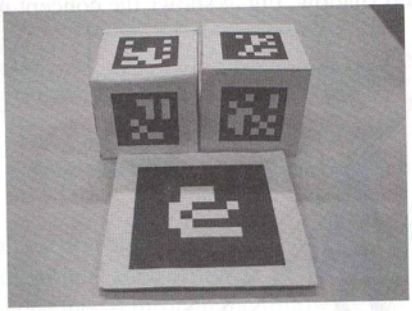

Fig. 6. Vision-based tangible user interface

3 Implementation and Results

We implemented the proposed system on a workstation with Xeon 2.5GHz CPU and NVIDIA QuadroFX3400 video adapter. The size of images used for 3D object modeling is 640x480. For background subtraction, an IEEE1394 camera captures 640x480 images in 30 frames/second.

Fig. 7 shows our 3D object reconstruction results. In Fig. 7(b), the initial mesh model looks bulky since it is over-estimated from the voxel space. The iso-surface of the model is generated from the occupied voxels as triangular meshes. Fig. 7(c) depicts the deformed mesh model. After the deformation stage, the mesh model becomes thinner than the initial mesh model. It fits to the input silhouette images. As shown in Fig. 7(d), texture mapping improves the realism of reconstructed model. According to our experiment, initial mesh reconstruction takes only a few seconds and mesh deformation takes at most two minutes. In the experiment, the scaling factors of the silhouette and smoothness forces are set to α=0.3 and β=0.2.

Fig. 8 depicts user segmentation result with the proposed algorithm. Fig. 8(a) and (b) show the background and the current frame images, respectively. Initial segmentation result is shown in Fig. 8(c). There are errors in the initial segmentation since the lighting condition of the scene is changed when the user entered to the scene. The light reflected from the user's body changes the colors of the background pixels. By detecting shadows and highlight in the current frame as shown in Fig. 8(d) and (e), we reduce the segmentation error. The final segmentation result after

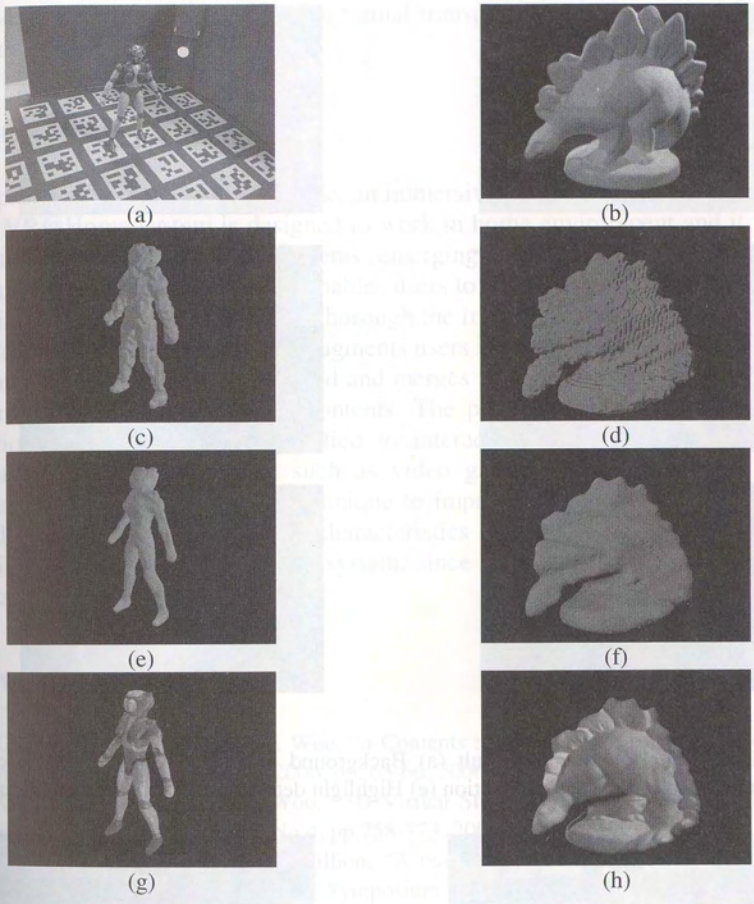

Fig. 7. 3D reconstruction results (a)(b) One of 16 input images (c)(d) Initial mesh model (e)(f) Deformed mesh model (g)(h) Texture mapped model

post-processing is shown in Fig. 8(f). The segmentation border of the user is preserved well. The performance of our segmentation method is 15 frames/sec with the image size of 320x240 and 10 frames/sec with 640x480.

The implemented VR@Home system is shown in the Fig. 9. Fig. 9(a) shows our experimental set up. There is a table as a interaction space and tangible user interfaces made of markers in the environment. A 3D virtual object created by our image-based modeling method is augmented on a marker in Fig. 9(b). Through the tangible user interface, the user can freely interact with the virtual object. In Fig. 9(c), the background subtraction method we propose is applied. The user and the table are segmented from a natural background in real-time. We change the user's background scene to a 3D virtual environment. In Fig. 9(d), soft shadows are rendered in the scene to improve the realism of the augmented virtual object. The user interactively controls a virtual light source to change the shadow drawn by the virtual object. The shadow

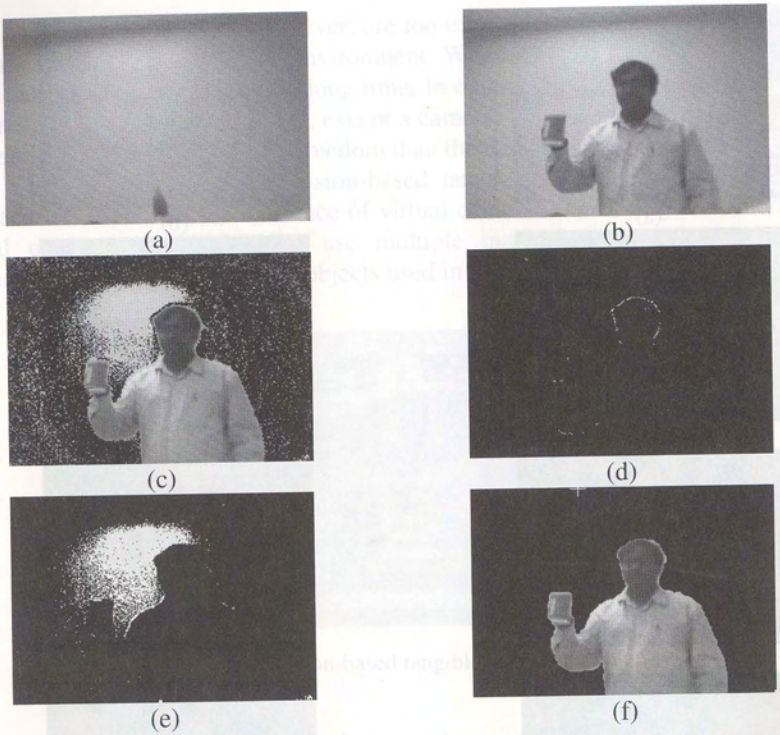

Fig. 8. Background subtraction result (a) Background image (b) Current frame (c) Initial segmentation result (d) Shadow detection (e) Highlight detection (f) Final segmentation result

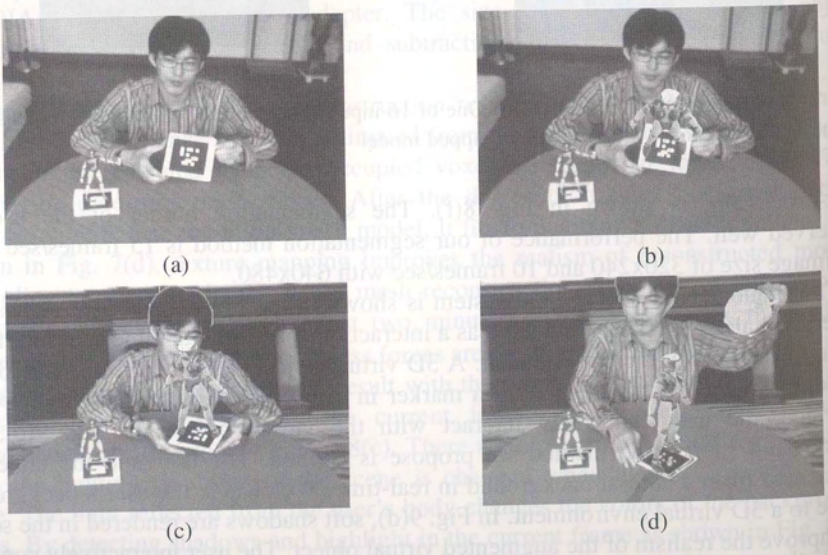

Fig. 9. Implementation of VR@Home (a) Experimental setup of VR@Home system (b) Augmentation of user-generated content (c) Background replacement by background subtraction (d) Real-time shadow rendering for better realism

of the virtual object is projected on a virtual transparent plane on the marker where the virtual object is augmented.

4 Conclusions

In this paper, we proposed VR@Home, an immersive contents creation system for 3D UGCs. VR@Home system is designed to work in home environment and it provides four functions to solve the problems emerging from the limitations of home environment. The proposed system enables users to create realistic 3D contents easily and to interact with them intuitively thorough the image-based modeling and tangible user interfaces, respectively. It also augments users in a virtual space by real-time user segmentation from natural background and merges virtual contents with soft shadows to improve realism of the virtual contents. The proposed system is for 3D UCGs generation, but it can also be applied to interactive on-line education, personal broadcasting, and entertainment such as video games. As future works, we will perform further research in each technique to improve its performance. In addition, we will focus on adding the other characteristics of u-Contents, u-Intelligence and u-Mobility, to our contents creation system, since we focused on the realism of the contents in this work.

References

1. SJ. Oh, W. Lee, Y. Park, and W. Woo, "u-Contents : A New Kinds of realistic contents in ubiquitous smart space," ISUVR2006, pp. 63-66, 2006.
2. N. Kim, C. Park, G. Kim, W. Woo, "3D Virtual Studio for Natural Inter-Acting," IEEE Trans. on SMC: Part A, Vol.36, No.4, pp.758-773, 2006.
3. J. M. Hasenfratz, M. Lapierre, F. Sillion, "A real-time System for Full Body Interaction with Virtual Worlds," Eurographics Symposium on Virtual Environ-ments, pp. 147-156, 2004.
4. William E. Lorensen and Harvey E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," Computer Graphics (Proc. of SIGGRAPH '87), Vol. 21, No. 4, pp. 163-169, 1987.
5. T. Horpnsm, D. Hanvaod and L. S. Davis, "A statistical approach for real time robust background subtraction and shadow detection," In Proc. of IEEE Frame Rate Workshop, 1999.
6. A. Elgammal, D. Harwood, and L. S. Davis, "Non-parametric model for background subtraction," Proceedings of the 6th European Conference on Computer Vision, vol. III, pp. 751 –767, 2000.
7. D. Hong and W. Woo, "A background subtraction for a vision-based user interface," In Proceedings of ICICS-PCM, pp. 263-267, 2003.
8. Intersense Inc. (http://www.isense.com/)
9. Polhemus (http://www.polhemus.com/)