Sensors and Materials, Vol. 29, No. 11 (2017) 1545–1554
MYU Tokyo

S & M 1448

Augmented-Reality-Based 3D Modeling System Using Tangible Interface

Chin-Hung Teng[1,2*] and Shueh-Shih Peng[1]

[1]Department of Information Communication, Yuan Ze University,
135 Yuan-Tung Road, Chung-Li, Taiwan, 32003, R.O.C.
[2]Innovation Center for Big Data and Digital Convergence, Yuan Ze University,
135 Yuan-Tung Road, Chung-Li, Taiwan, 32003, R.O.C.

(Received April 20, 2017; accepted August 9, 2017)

Keywords: augmented reality, 3D modeling, tangible interface, interactive modeling

Object 3D model creation is a difficult task and it typically requires professional modeling tools or devices. For nonexpert users who want to create simple 3D applications, these approaches are either too expensive or require extensive study and learning of the complex modeling tools. In this work, we develop a modeling system that allows an amateur to interactively create a simple 3D model for his/her applications. The system is developed on the basis of augmented reality (AR) technology with a tangible user interface. AR allows a user to operate virtual models in a real 3D environment; thus, the operation is more natural and intuitive than traditional 2D-graphical-interface-based modeling tools. A user can easily change his/her viewpoint by rotating an AR marker or moving the physical camera. An AR pen is also designed in the system, which serves as a tangible device that allows a user to select, translate, rotate, and modify the created model. With the AR pen, a user can scale the model and combine it with other models to form a large model with diverse shapes. In particular, we also include a free-form deformation (FFD) tool in the system so that the users can modify the shape of a model using the tangible interface and create an imagined object. To extend the application range of the proposed system, we port the system to mobile devices such as tablets or smartphones so that any user with suitable devices can create his/her 3D models using the proposed system.

1. Introduction

Creating an object 3D model is a very important step for many 3D-based applications such as virtual reality (VR), augmented reality (AR), games, and animation. Typically, the modeling task can be classified into two types: modeling for objects in the real world and modeling for imagined objects.

For objects in the real world, we can measure their 3D structure using an instrument. The instrument may be a 3D laser scanner, a depth sensor, or simply a camera or a number of cameras.(1) This type of system includes time-of-flight(2,3) and structured light(4) approaches. The advantage of using scanning devices lies on its convenience and high accuracy. However, these devices are sometimes expensive. Recently, because of the cheapness of ordinary RGB cameras, image-based modeling has received much attention. The approaches of image-based modeling can be further

*Corresponding author: e-mail: chteng@saturn.yzu.edu.tw
http://dx.doi.org/10.18494/SAM.2017.1703

ISSN 0914-4935 © MYU K.K.

classified as binocular stereo,(5) shape from contour,(6) photometric stereo,(7) and structure from motion.(8) Sometimes, some user interactions are required to complete the modeling task and this leads to the so-called interactive modeling. VideoTrace(9) is a representative method of image-based interactive modeling. This method is applied to an image sequence. The user needs to go through the image sequence and draws some polygons on the image to indicate the faces of an object. Later, the 3D positions of points on the polygons are determined, and the polygonal mesh representing the object is recovered. 3-Sweep(10) is an interactive approach that can create the 3D model of an object from a single image. This approach is developed on the basis of generalized cubes and generalized cylinders and is suitable for creating the 3D models of objects with such shapes. Reference 11 shows a sketch-based modeling method for 3D building. This system allows users to perform interaction on 2D photos and then define the 3D context according to sketched polygons. PhotoSketch(12) is another image-based interactive modeling system that can be used to create the model of a building from urban scenes.

In contrast to previous approaches that focus on creating models of existing objects, we can also create object models using 3D modeling software such as 3ds Max and Maya. The advantages of these tools are that they can be used to create the models of imagined objects; thus, they are widely used in the fields of game and animation. To provide the functionality of creating diverse and sophisticated 3D models, these systems are equipped with powerful modeling tools. However, this creates a high entry barrier for using these systems because utilizing these powerful modeling tools sometimes requires professional background knowledge of computer graphics. Moreover, these software tools use a traditional 2D graphical user interface and input devices such as a mouse for modeling. Thus, users need to switch different views such as top view, front view, and side view to correctly place or scale the 3D models. These features make these software tools suitable only for professional users.

With the emergence of more and more 3D applications, users have more demand for creating their own 3D models. Sometimes, the required models are very simple and do not require very powerful modeling tools to create them. Hence, for those users without professional skill, they need a simple and cheap approach to create sometimes maybe very simple models. In view of this, in this work, we develop a modeling system that is user-friendly for nonexpert users. This system has three major characteristics.

First, we develop the system using AR technology. As stated by Liang and Green, "… the traditional mouse-based interaction style is a bottleneck to 3D geometric design, as it forces the user to decompose a 3D task into a series of 1D or 2D tasks".(13) Using AR technology can naturally avoid some of the problems as we can see the models through the physical camera and change the viewpoint by directly moving the camera. This makes the model visualization more intuitive and convenient compared with the 2D-graphical-user-interface-based modeling software.

Second, we introduce a tangible interface for model manipulation. A tangible interface has been proven to be a good method for human–machine interaction in AR application.(14) By tangible interface, we can directly "touch" the models and control them. We call this tangible interface the AR Pen and it allows us to select, translate, and rotate the models by directly stirring the models using the AR Pen. In particular, we also include a free-form deformation (FFD) technique in our system so that we can modify the shape of a model through the AR Pen. This tangible interface further makes the operation of our system more natural than traditional modeling software.

Third, we port our system to a mobile device and exploit the touch screen of the mobile device to increase the operation flexibility. Thus, the user can simply touch the screen to select the modeling function and then use the AR Pen to complete the modeling task. Thus, the modeling task is not limited to a fixed place with a specific desktop computer. Our system allows users to perform modeling anytime and anywhere as long as they have our system installed on their mobile devices and two AR markers that used to define the workplace and the AR pen.

Actually, AR-based modeling is not a new idea and some systems have been developed. Piekarski proposed an interactive modeling system for outdoor environments using AR technology.(15) Piekarski proposed the concept of working planes and his system can allow a user to create the model of a distant object via the working planes. However, Piekarski's system required a number of sensors for head-mounted display tracking and the user needs to wear the computer when walking in outdoor environments. Bastian *et al.* proposed an interactive modeling system for AR applications.(16) In this system, the user needs to first select the object and then the system can segment the object from the image. Later, the silhouettes of the object from different viewpoints are extracted and a shape from a contour technique is applied to create the 3D model of the object.

ARpm(17) is a polygon editing system that combines 3ds Max and AR technology. In this system, users can see models via the head-mounted display and the AR marker. With a wireless pointer device, the user can directly edit the models using the device. 3DARModeler(18) is another system based on 3ds Max and AR. It can be seen as a simplified 3ds Max where users can employ a number of AR markers for placing and creating the models. However, both previous systems are implemented in the PC environments, which limits their workplace.

The position of our system is quite similar to those of the previous two systems. However, the goal of our system is not to replace the powerful 3ds Max but provide a modeling tool that allows nonprofessional users to create simple models for their simple applications. Thus, we do not include powerful modeling tools in the system but instead provide several intuitive and natural ways in the system so that any user without computer graphics knowledge can use the system easily. Moreover, we port the system to a mobile device to extend the workplace so that a user can use the system anytime and anywhere.

The paper is organized as follows. In Sect. 2, we first introduce our system architecture and then describe how we design the system. Then, we discuss how to employ our system through a modeling example in Sect. 3. Finally, we conclude in Sect. 4.

2. System Design

2.1 System architecture

Our system architecture is depicted in Fig. 1. We employ Unity as our developing platform as it provides a powerful 3D physics engine and can easily deploy the app into Android and iOS systems. The AR engine we used is the Vuforia SDK that is initially developed by Qualcomm. In Vuforia, the image used for placing virtual models and camera tracking is called the image target. We used two image targets to create our system as shown in Fig. 2. One image target is applied to create our virtual modeling environment and we called it the workplace. The second image target is used to create a virtual pen, which we called the AR Pen. The AR Pen is used to manipulate the models.

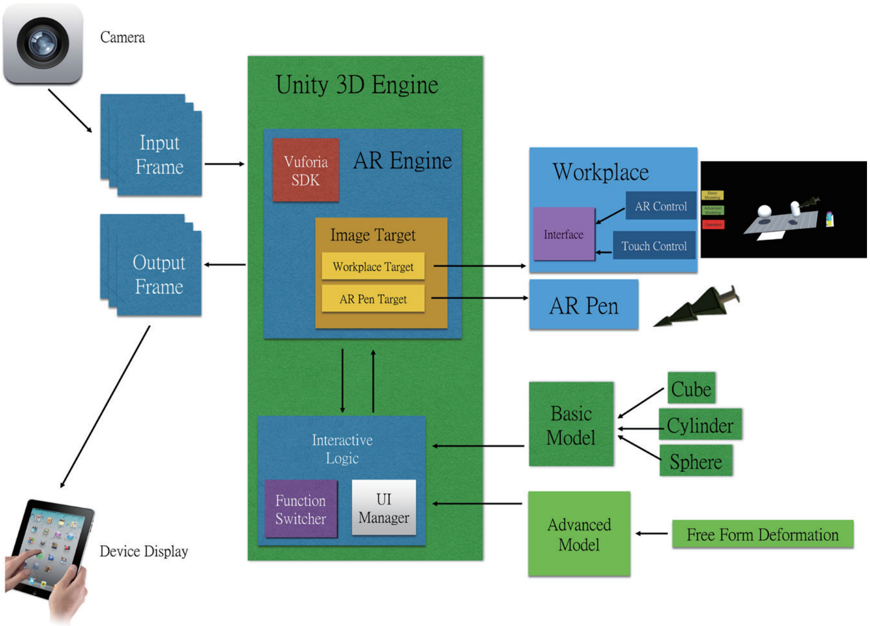


Fig. 1. (Color online) System architecture of the proposed modeling tool.

(a) (b)

Fig. 2. (Color online) Two image targets that are used to define our (a) workplace and (b) AR Pen.

Inside the Unity engine, an interactive logic unit is devised to manage the user interface and switch different modeling functions. Our system has a basic tool that can be used to generate basic primitives such as a cube, a cylinder, and a sphere, and an advanced tool, the FFD, which can be employed to modify the shape of a model. Currently, the system is just a prototype with limited functions but we will include more tools and more primitives in the future to enrich its modeling capability.

The system is operated as follows. When the app is activated, the device will continuously capture images from the built-in camera and detect the two image targets. If the image target representing the workplace is detected, a virtual environment is placed on the image target and displayed on the screen of the mobile device (see Fig. 3). When the image target representing the AR Pen is detected, a virtual pen appears on the screen. The user can move the AR Pen to select different tools and use the pen to manipulate the model such as translation, rotation, scaling, and model deformation. We present our methods for manipulating the models in the next section.

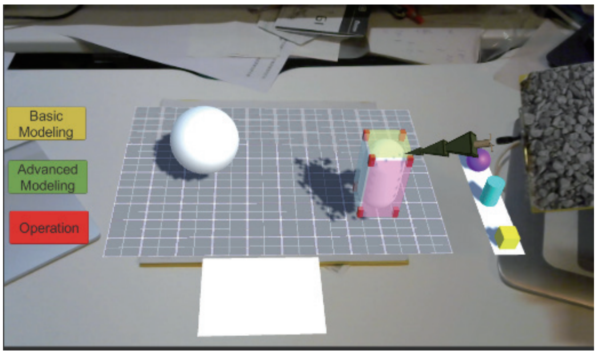


Fig. 3. (Color online) A screenshot of our interactive modeling system.

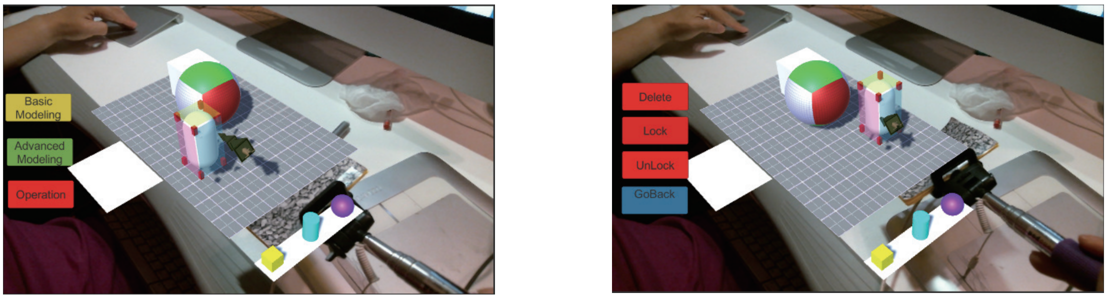
2.2 Model manipulation

A screenshot of the proposed modeling system is shown in Fig. 3. On the left part are three buttons that are used to activate the functions of basic modeling, advanced modeling, and model operation. The user can touch the screen to invoke these functions. On the right part are three primitives (sphere, cylinder, and cube) that allow a user to easily generate the corresponding shapes. A user can directly select the primitive using the AR Pen and move it to the workplace (a grid plane) as shown in Fig. 3. A bounding box enclosing the selected model is shown to indicate that the model is being manipulated. The user can then change the shape of the model with the provided tools.

For a modeling system, inspecting a model from different viewpoints is a critical step. Since our system is developed based on AR technology, we can directly move the physical camera to inspect the model. Zooming in/out is achieved by moving the camera forward and backward. Moreover, we can also rotate the whole virtual environment to find a suitable orientation to view and operate the system by rotating the image target.

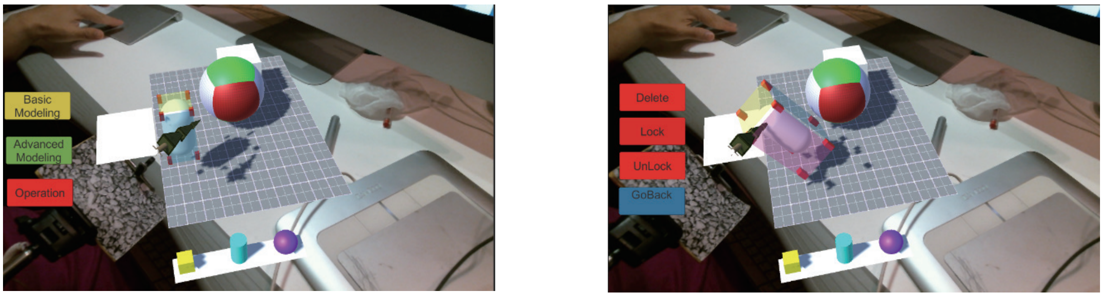
Our system provides three basic ways to operate the created model. The first one is translation. We can translate a model by using the AR Pen to stick the model and then move the AR Pen to the desired position as illustrated in Fig. 4. The second operation is rotation. Rotating a model is achieved by using the AR Pen to stir the model as shown in Fig. 5. The final operation is scaling. Model scaling is completed by using the AR Pen to touch one of the faces of the model's bounding box and move the AR Pen. The model is then scaled along the normal of the selected face as illustrated in Fig. 6.

Translation is straightforward. The detailed operations of rotation and scaling are depicted in Fig. 7. For rotation, when the AR Pen touches the bounding box of the model, the touching point is recorded as the initial point. Then, we can continuously move the AR Pen and the final position of the tip of the pen is denoted as the end point. The initial point, end point, and model center form a plane and the normal of the plane is the rotation axis. The rotation angle is the angle inscribed by the vectors formed by the three points as shown in Fig. 7(a). For scaling, the initial point and end point are defined similarly to rotation. The scaling ratio is defined by dividing the distance between the end point and the object center by the distance between the initial point and the object center, as illustrated in Fig. 7(b).



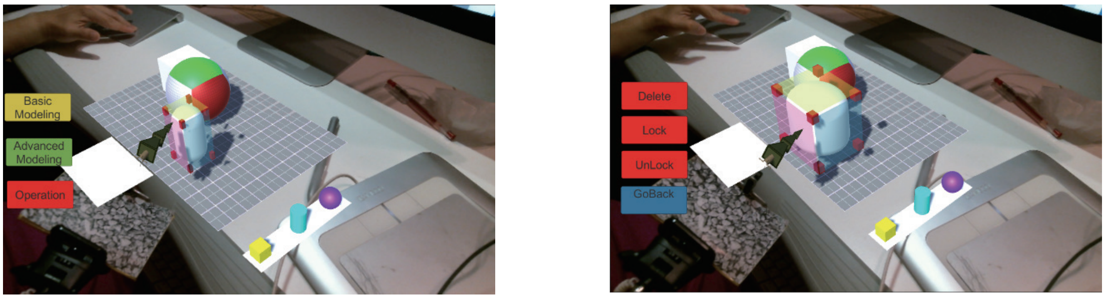
(a) (b)

Fig. 4. (Color online) Model translation is achieved by sticking the model with the AR Pen and then moving the AR Pen.



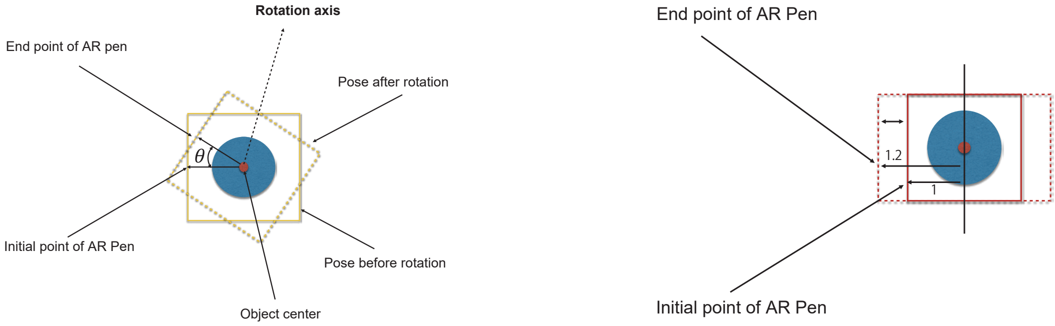
(a) (b)

Fig. 5. (Color online) Model rotation is achieved by using the AR Pen to stir the model.



(a) (b)

Fig. 6. (Color online) Model scaling is achieved by pushing/pulling one face of the bounding box of the model.



(a) (b)

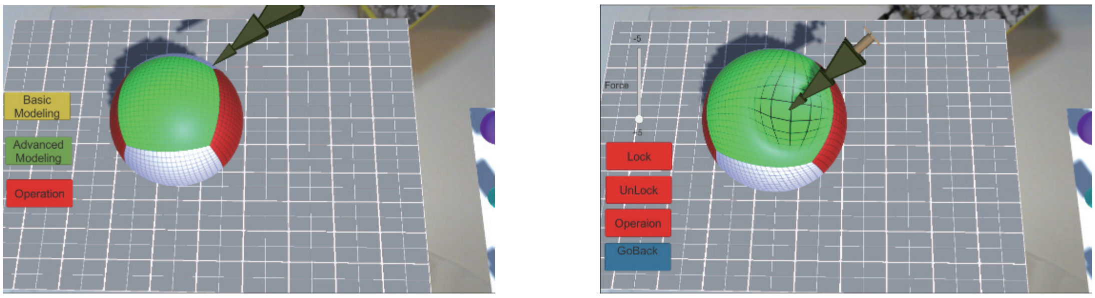
Fig. 7. (Color online) An illustration of (a) object rotation and (b) scaling.

2.3 FFD

We include a FFD in our system to create diverse models. An example of FFD is shown in Fig. 8. FFD allows us to locally change the shape of a model. We put a local force on a neighborhood of the application point as illustrated in Fig. 9(a). The direction of the applied force is from the point of application [red dot in Fig. 9(a)] toward the surrounding vertices [blue dots in Fig. 9(a)] and the force is inversely proportion to the square of the distance (d) between the point and the surrounding vertices as follows:

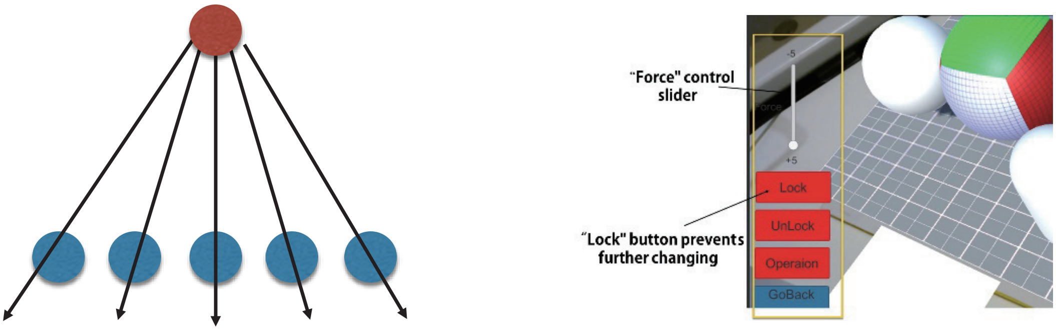
$$F_v = \frac{F}{\varepsilon + kd^2}, \tag{1}$$

where we include an ε in the denominator to prevent zero distance. The parameter k is used to control the size of the influenced region. A large k indicates that a large neighborhood of vertices is influenced by the applied force and a small k means that the FFD is more local. F in Eq. (1) is a force controlled by the user and can be adjusted with a control panel, as illustrated in Fig. 9(b). Figure 9(b) also shows a function of our system. The "Lock" button can be used to lock a model to prevent it from changing by the unintentional operation of the AR Pen.



(a) (b)

Fig. 8. (Color online) An example of FFD.



(a) (b)

Fig. 9. (Color online) (a) Force applied on a local set of vertices. (b) Control panel of FFD.

3. Modeling Example

Although our system provides only several simple modeling tools, it can create a variety of models with different shapes. In this section, we demonstrate the modeling capability of the proposed system via a modeling example.

The model we want to create is a cup on a small table. This model is composed of two spheres, a cylinder, and a cube with suitable scaling and deformation. We first scale the cube to form the desktop and then use the cylinder as the table leg. One sphere is scaled to form the bottom plane of the table and one sphere is deformed to form the cup. Finally, all objects are put together to complete the model. The detailed modeling procedures are shown in Fig. 10.

From this example, we can see that by combining different shapes with suitable scaling and FFD, we can achieve diverse model creation. Nevertheless, some models with a specific shape such as a revolution surface are still not easily generated by the proposed system. Hence, we will include more modeling tools such as extrusion and revolution surfaces to enhance the modeling capability of the proposed system in the future.

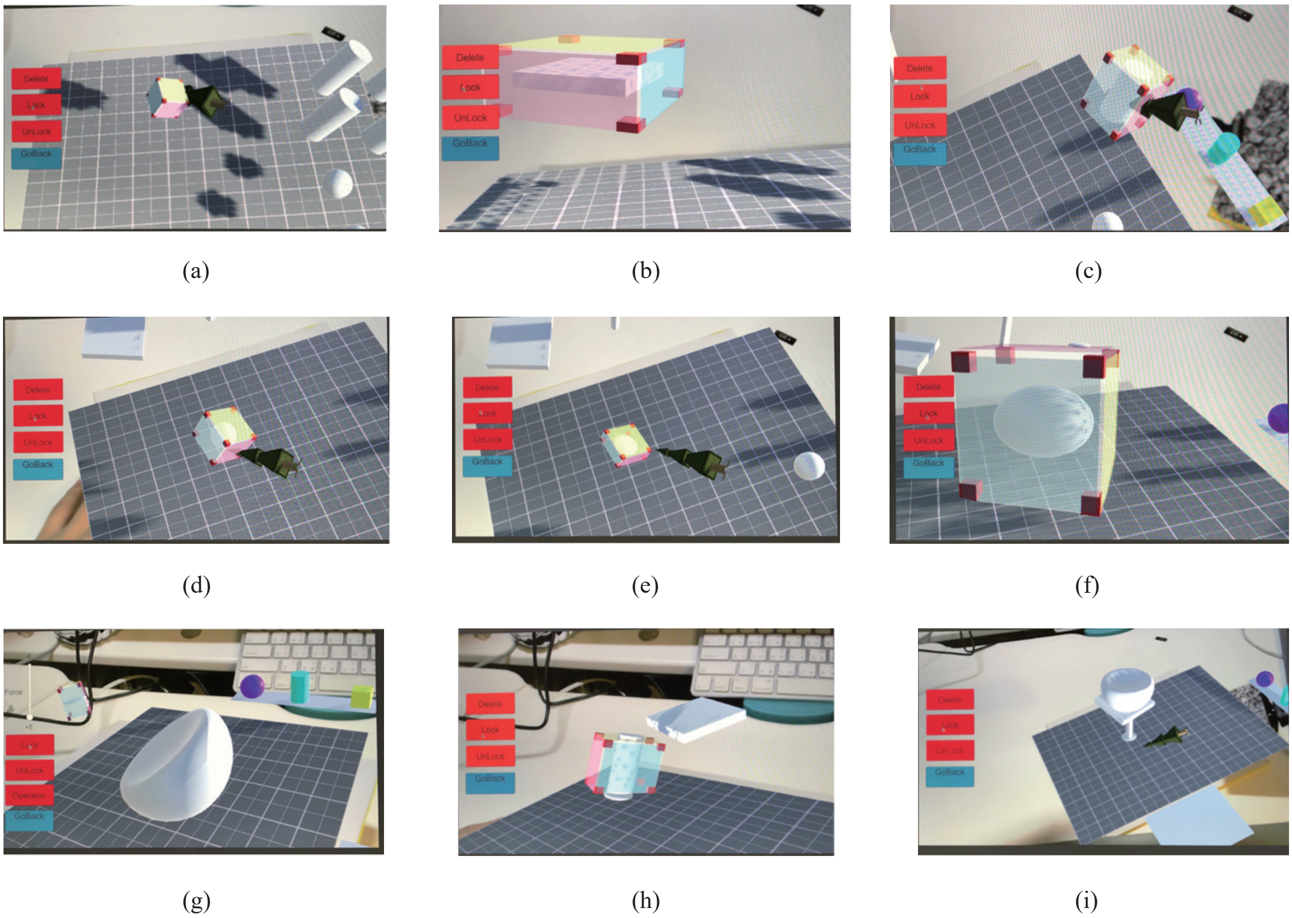


Fig. 10. (Color online) A modeling example: a cup on a small table. (a) Create a cube. (b) Scale the cube so that it can be used as a desktop. (c) Create a cylinder as the table leg. (d) Create a sphere. (e) Scale the sphere so that it can be used as a bottom plane of the table leg. (f) Create a sphere. (g) Use the FFD tool to reshape the sphere so that it can be deformed as a cup. (h) Put all objects together to form the final object. (i) A cup on a small table.

4. Conclusions

3D Modeling is a very important step for many 3D-based applications. Current approaches for 3D modeling include modeling using powerful 3D software and reconstruction of the object model using specially designed instruments. However, both of them have their drawbacks for nonprofessional users. For nonprofessional users who want to create simple models for their simple applications, they will not use expensive scanning instruments or complex modeling software for their purpose. Hence, an inexpensive and more intuitive modeling system is necessary.

In this paper, we presented an interactive modeling system based on AR technology. The purpose of this system is not to replace professional modeling software, but provide a friendly way for users without solid modeling background knowledge to create simple models for their simple applications. We employed AR technology to eliminate the inconvenience of the traditional 2D graphical user interface. We utilized a tangible interface (AR Pen) to replace the operation of traditional 2D input devices such as a mouse. We devised a tangible interface for model manipulation, including translation, rotation, scaling, and FFD. A modeling example was presented and the results demonstrated the feasibility of the proposed system. As we developed the system using the Unity platform, the system can be easily ported to mobile devices. Thus, a user with a mobile device and two AR markers can perform the modeling task anytime and anywhere.

Acknowledgments

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant Nos. MOST 104-2221-E-155-032 and MOST 104-3115-E-155-002.

References

1 J. Shi, Z. Sun, and S. Bai: The Visual Computer (Published online: 10 December 2016).
2 A. Swadzba, B. Liu, J. Penne, O. Jesorsky, and R. Kompe: Proc. Int. Conf. on Computer Vision Systems (Applied Computer Science Group, Bielefeld, 2007).
3 Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt: Proc. Int. Conf. on Computer Vision and Pattern Recognition (IEEE, San Francisco, 2010) p. 1173.
4 C. Rocchini, P. Cignoni, C. Montani, P. Pingi, and R. Scopigno: Graphics Forum **20** (2001) 299.
5 A. Geiger, J. Ziegler, and C. Stiller: Proc. IEEE Intelligent Vehicles Symp. (IEEE, Baden-Baden, 2011) p. 963.
6 J. Y. Zheng: IEEE Trans. Pattern Anal. Mach. Intell. **16** (1994) 163.
7 T. Higo, Y. Matsushita, N. Joshi, and K. Ikeuchi: Proc. 12th IEEE Int. Conf. Computer Vision (IEEE, Kyoto, 2009) p. 1234.
8 M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch: Int. J. Comput. Vision **59** (2004) 207.
9 A. Hengel, A. Dick, T. Thormahlen, B. Ward, and P. H. S. Torr: ACM Trans. Graphics **26** (2007) Article No. 86.
10 T. Chen, Z. Zhu, A. Shamir, S.-M. Hu, and D. Cohen-Or: ACM Trans. Graphics **32** (2013) Article No. 195.
11 M. Schwärzler, L.-M. Kellner, S. Maierhofer, and M. Wimmer: Proc. 21st ACM SIGGRAPH Symp. on Interactive 3D Graphics and Games (ACM, San Francisco, 2017).
12 G. Wolberg and S. Zokai: The Visual Computer (Published online: 7 April 2017).
13 J. Liang and M. Green: Proc. 3rd Int. Conf. CAD and Computer Graphics (Beijing, 1993) p. 217.
14 N. Hagbl, R. Grasset, O. Bergig, M. Billinghurst, and J. El-Sana: ACM Comput. Entertain. **12** (2014).
15 W. Piekarski: Ph.D. Thesis, University of South Australia (2004).

16 J. Bastian, B. Ward, R. Hill, A. van den Hengel, and A. Dick: Proc. IEEE Int. Symp. Mixed and Augmented Reality (IEEE, Seoul, 2010) p. 199.

17 P. Fiala and N. Adamo-Villani: Proc. Int. Symp. Mixed and Augmented Reality (IEEE, Vienna, 2005).

18 T. V. Do and J.-W. Lee: Int. J. Electr. Comput. Syst. Eng. **4** (2010) 145.

About the Authors

Chin-Hung Teng received his Ph.D. degree in Electrical Engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2005. He is currently an assistant professor at the Department of Information Communication, Yuan Ze University. His research interests include color image processing, computer vision, computer graphics, image-based modeling, and augmented reality.

Shueh-Shih Peng received his M.S. degree in Information Communication from Yuan Ze University, Taiwan, in 2017. His research interests include augmented reality and AR-based applications design.