Mobile Augmented Reality System for In-situ 3D Modeling and Authoring

Han Kyu Yoo
Dept. of Digital Contents
Sejong University
Seoul, Korea
tkehdkr@naver.com

Jong Weon Lee*
Dept. of Digital Contents
Sejong University
Seoul, Korea
jwlee@sejong.ac.kr

Abstract— **This paper proposes a mobile augmented reality system that can model 3D virtual objects and author augmented reality contents on site. The differences of the proposed system from the existing ones are an interaction approaches used to generate and manipulate primitives and additional features such as a shadow and a multi-freezing mode to create realistic augmented reality contents efficiently.**

***Keywords*— In-situ authoring, modeling, augmented reality, gesture recognition, interaction.**

I. INTRODUCTION

Augmented Reality (AR) is a technology that combines virtual and real worlds in real time to help users complete their work or to provide users new experience. The rapid spread of smartphones has made it possible to experience AR on smartphones. Various AR applications have been developed on smartphones using sensors such as a camera, a GPS and an inertial sensor. Contents of these AR applications are generally provided by application developers. Recently the paradigm has been shifted from developer-created contents to user-created contents. The needs for developing authoring systems for general users are emerged.

Research of authoring AR contents has been started about 20 years ago. The early AR contents authoring systems are desktop-based ones. AR contents require the relations between a real world and augmented virtual objects, so few researchers have developed in-situ authoring systems. Pikarski et al. [1] proposed an in-situ authoring system for outdoor environment with a head-mounted display and marker-attached gloves. Hand interactions were used to model 3D virtual objects and to manipulate them.

Several authoring systems have been developed on mobile devices such as tablets, smartphones and smart pads. Guven et al. [2] proposed ways to annotate text labels and audios on user's surroundings. They froze the current AR scene displayed on a tablet and edited it to create AR contents. Liu et al. [3] developed mobile augmented note-taking system. It allows users to put self-authored notes onto physical objects on site. These researches provide users efficient ways to author AR contents, but users can only create limited contents such as labels and notes using them.

Langlotz et al. [4] proposed in-situ authoring for mobile AR for small indoor and large outdoor working spaces. A user begins authoring by generating 3D primitives, such as cubes and cylinders. The user scales or moves the generated primitives or applied textures to them to create AR contents. Additionally the system allows annotating 2D contents and provides a freeze mode.

The proposed system shares some characteristics of the system developed by Langlotz et al. The proposed system also provides users modeling from primitives, primitive manipulation and freeze functions. The differences of the proposed system from the existing ones are an interaction approaches used to generate and manipulate primitives and additional features such as a shadow and a multi-freezing mode to create realistic AR contents efficiently.

II. METHOD

The proposed system can be divided into four parts as shown in Figure 1. A user selects primitives and manipulates them to the right positions and orientations. The resulting primitives are grouped together to create 3D virtual models. If it is desired, a shadow is added to the created virtual models after estimating a lighting position.

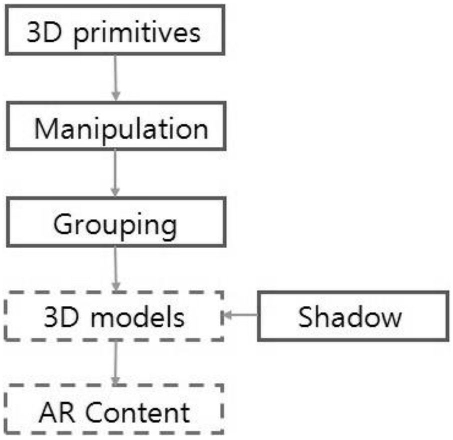


Figure 1. The system flow of creating AR contents

* Corresponding author

978-1-4799-3919-0/14/$31.00 ©2014 IEEE

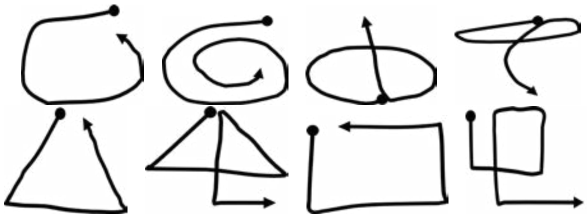


Figure 2. Sample gestures.

A. 3D Primitives

3D primitives are basic elements that are used to create 3D models in the proposed system. A user selects primitives that are components of target 3D models among the given 3D primitives. The target models are combined to create AR contents.

A gesture-based interaction is used to load a primitive to an AR scene. The user draws a line drawing similar to a target primitive on the display of a smartphone. The proposed system recognizes user's line drawings using the algorithm similar to one described by M. Elmezain et al. in [5]. There are three main stages (gesture tracking, feature extraction and classification) in the proposed gesture recognition procedure.

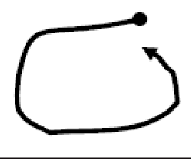
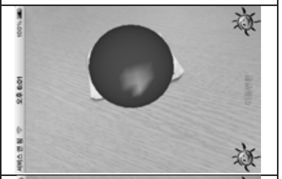
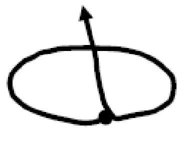
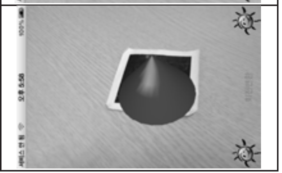
The most gesture-based interaction systems have difficulties to determine when a gesture starts and when it ends. In the proposed system, we assume the entire gesture is meaningful one so we do not need to estimate the starting and the ending points of an input gesture. The input gesture is recorded as the sequence of 2D points (x_t, y_t).

We use the orientation feature as a main feature in the proposed system. The orientation is determined between two consecutive points from the sequence of 2D points, which represents the input gesture, by (1).

$$\theta_t = \arctan\left(\frac{y_{t+1}-y_t}{x_{t+1}-x_t}\right)\ t = 1, 2, \dots, T-1 \qquad (1)$$

where T indicates the length of the input sequence of 2D points. The orientation is quantized into eight directions by dividing it by 45 degrees. The quantized sequence is used as an input to HMM (Hidden Markov Model).

TABLE I. EXAMPLES OF GESTURES AND CORRESPONDING PRIMITIVES

Name	Gesture	Primitive
Sphere		
Cone		

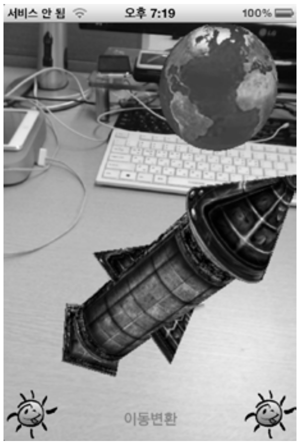


Figure 3. The models created by combining primitives.

Currently the proposed system can recognize eight gestures as shown in Figure 2. The quantized sequence for each gesture is modeled by Left-Right (LR) model with varying number of states based on its complexity. The minimum number of states is seven for the proposed system. The quantized sequences are trained using the Baum-Welch re-estimation algorithm. Each sequence is classified by the Viterbi algorithm [6].

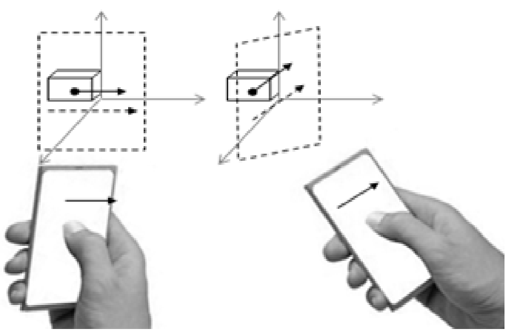


Figure 4. Dynamically changing constraints.

Eight gestures are assigned for eight primitives (sphere, tours, cube, cylinder, cone, triangular pyramid, triangle and quadrangular pyramid) in the proposed system. A user can create various 3D models using the eight primitives. We will add more primitives to create more complex 3D models. Example gestures and corresponding primitives are shown in Table 1.

We tested the gesture recognition system with three participants. Each participant tested each gesture for twenty times. The recognition result was categorized intoN_{ND} , not detected even though the gesture was in the DB,N_{IC} incorrect classification, and N_C , correct classification. The gesture recognition rate R was computed using the equation (2) where N is the number of gestures and N_C is the number of the correctly recognized gestures. Table 2 shows the evaluation result of the recognition system.

$$R = \frac{N_C}{N} \times 100\% \qquad (2)$$

TABLE II. EVALUATION OF GESTURE RECOGNIZATION RESULT

Name	N	N_C	N_{ND}	N_{IC}	R
Sphere	60	54	0	6	90.0
Tours	60	50	1	9	83.3
Cone	60	54	1	5	90.0
Cylinder	60	56	1	3	93.3
Triangle	60	55	0	5	91.6
Triangular pyramid	60	52	2	6	86.6
Cube	60	54	1	5	90.0
Quadrangular pyramid	60	50	2	8	83.3
Total	480	425	8	47	88.5

The best and the worst gesture recognition rate was 93.3% and 83.3%. The average gesture recognition rate is 88.5%. The primitives with lower recognition rates were confused with primitives with similar shapes.

B. *Manipulation and Grouping*

The Manipulation is composed of moving, rotating and scaling the generated primitives and adding textures to them. We developed dynamic constraint-based user interface to manipulate primitives. It changes a constraint plane, which defines a translation area, a rotation axis and a scaling direction, dynamically according to the orientation of a user's smartphone. The constraint planes for translation motions are shown in Figure 4. The orientations of the left and the right smartphones are different so the different corresponding constraint planes are defined. The dotted rectangular in Figure 4 indicates a constraint plane.

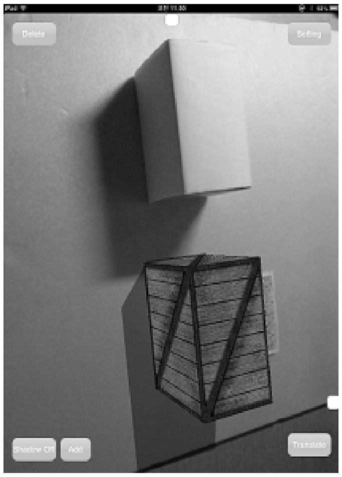


Figure 5. Rendering a virtual shadow similar to a real one.

Multi-touch inputs on the display of a smartphone are directly mapped to the motions of a primitive on a 2D constraint plane. A primitive can be scaled along any direction using the interface. A user can modify the size of a primitive using the scaling procedure to create diverse models. In the Grouping part, primitives are grouped together to form a single model similar to our previous works [7, 8]. A 3D model is created by combining primitives together as a child creates an object by putting blocks together. Example models are shown in Figure 3.

C. *Shadows*

3D models used to create AR contents can be created using the proposed modeling steps or uploaded from a DB. After adding 3D models to an AR scene, shadows can be added to them. Shadows improve the reality of AR contents. A content user can recognize the relative positions between virtual objects or virtual and real objects if shadows are correctly rendered as shown in Figure 5.

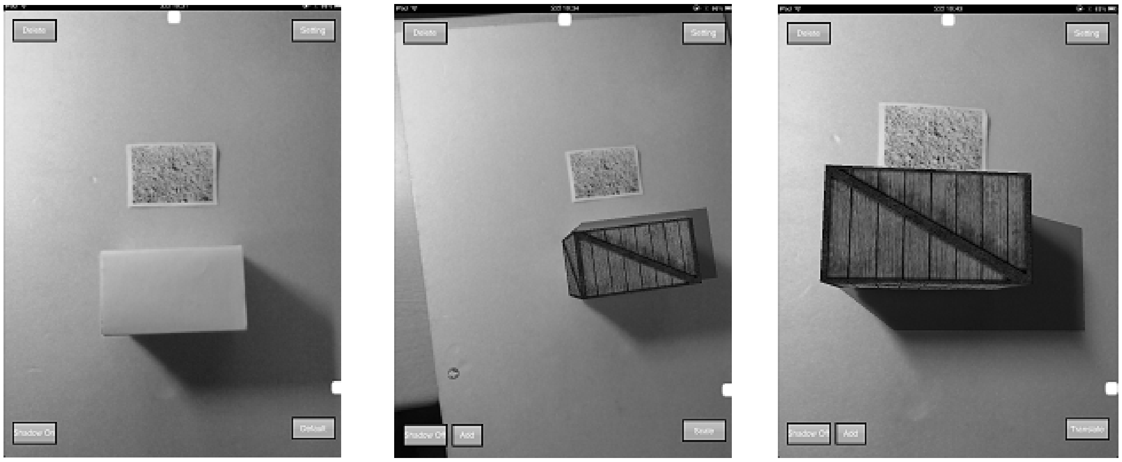
To render shadows correctly we need to estimate the positions of lighting sources. We add a simple part, Shadow, to estimate a lighting source. We apply the shadow estimating process developed in [8] to the proposed system. First, one real object in an AR scene is selected as a reference object (Figure 6 (a)). It is overlaid by the corresponding virtual object (Figure 6 (b)). It is better to select a simple object as a reference object because a user has to create a corresponding virtual object. Second, a user manipulates a virtual lighting source shown in Figure 6 (b) as a small sphere. When the virtual shadow is rendered closely to the real shadow of the reference object, we stop the estimating process as shown in Figure 6 (c). The estimated lighting position or direction is used to create shadows of virtual models. If the reference object was closely located to virtual objects in AR scene, the shadows of the virtual objects will closely resemble ones of the real objects. If we could estimate multiple lighting sources, this restriction would be less important.

We only applied simple hard shadows because of limited computation power on smartphones. When the processing power of smartphones is improved, more complex shadows can be rendered with the same estimating procedure.

D. *Multi-Freezing Mode*

Tracking the camera attached to a smartphone is required to use the presented authoring system to create AR contents. We initially developed the system using the marker-based tracking. We extend the system to use natural patterns for tracking. It is easy to use natural patterns for tracking since there are few well-known algorithms and SDKs [9, 10]. Since the tracking is required for authoring AR contents, we also add the freeze mode similar to existing authoring systems to relieve possible pains caused by holding a smartphone for long time. Users complained pains caused by holding a smartphone while using an AR contents in a user study.

The proposed freeze mode differs from the existing freeze modes. It uses multiple captured scenes and we call it a multi-freezing mode. We use multiple captured scenes to locate an augmented 3D object at the correct location of the AR world. If only one captured scene is used, it is difficult to confirm whether the augmented object is located at the desired position, especially along the direction orthogonal to the captured scene. More than two captured scenes are required to locate an augmented object at the desired position.



(a) (b) (c)

Figure 6. Estimating a lighting position or direction. (a) Selecting a real object. (b) Overlaying the selected real object using the corresponding virtual object. (c) Estimating the lighting position by aligning the real shadows and the virtual shadows.

III. CONCLUSION

We proposed a new mobile AR system that could create simple 3D models and author AR contents on site. The proposed system has some characteristics similar to existing authoring systems. The proposed system also has unique characteristics. The gesture-based triggering, multi-touch based interactions and the multi-freezing mode were developed to provide users more intuitive and simple interactions. Shadows were added to AR scene by estimating the light source of a real world to improve the realism of AR contents.

The proposed system still needs additional improvement. The number of primitives has to be increased to create more complex objects. We could add a vision-based modeling function to model real objects using a smartphone's camera. The system is able to estimate a single lighting source currently. We need to estimate multiple lighting sources since there are more than one lighting sources in a real environment. The estimated lighting position is currently static, so the proposed system cannot generate shadows correctly on dynamic lighting conditions. We also need to evaluate the proposed system and to expand the proposed system for outdoor environment.

ACKNOWLEDGMENT

This research (Grants No. C0133918) was supported by Business for Cooperative R&D between Industry, Academy and Research Institute funded Korea Small and Medium Business Administration in 2013. This research is also supported by Ministry of Culture, Sports and Tourism and KOCCA in the CT R&D Program 2011.

REFERENCES

[1] W. Piekarski and B. Thomax, "Tinmith-metro: new outdoor techniques for creating city models with an augmented reality wearable computer," Proc. of IEEE the fifth international symposium on wearable computers, 2001, pp. 31-38.

[2] S. Guven, S. Feiner, and O. Oda, "Mobile augmented reality interaction techniques for authoring situated media on-site," Proc. IEEE and ACM International Symposim on Mixed and Augmented Reality, 2006, pp. 235-236.

[3] C. Liu, J. Diehl, S. Huot, and J. Borchers, "Mobile augmented note-taking to support operating physical devices," Mobile Augmented Reality: Design Issue and Opportunities – 1st Workshop on Mobile Augmented Reality, MobileHCI, 2011.

[4] T. Langlotz, S. Mooslechner, S. Zollmann, C. Degendorfer, G. Reitmayr, and D. Schmalstieg, "Sketching up the world: in situ authoring for mobile augmented reality," J. Personal and Ubiquitous Computing, vol. 16:6, Aug. 2012, pp. 623-630.

[5] M. Elmezain, A. Al-Hamadi, and B. Michaelis, "Hand trajectory-based gesture spotting and recognition using HMM," Proc. International Conference on Image Processin, 2009, pp. 3577-3580

[6] X. D. Huang, Y. Ariki, and M. A. Jack, "Hidden Markov Models for speech recognition," Edinburgh Univ. Presse, 1990

[7] J. Y. Park and J. W. Lee, "Tangible augmented reality modeling," Proc. Entertainment Computing, LNCS No. 3166, Sep. 2004, pp. 254-259.

[8] D. Trien and J. W. Lee, "3DARModeler: a 3D modeling system in augmented reality," J. Computer Systems Science and Engineering, vol. 4:2, 2008, pp.

[9] Qualcomm Vuforia, https://www.vuforia.com

[10] Metaio Augmented Reality SDK, http://www.metaio.com/sdk