Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/092L30

Mobile Augmented Reality Authoring System with 3D Modeling and Lighting Estimation

HanKyu Yoo and JongWeon Lee∗

Department of Digital Contents, Sejong University, Seoul 143-747, Korea

Received: 10 Jun. 2014, Revised: 10 Aug. 2014, Accepted: 12 Aug. 2014
Published online: 1 Apr. 2015

Abstract: Various augmented reality applications have been developed on smartphones and smart pads. Application developers generally provide contents for these applications. The authors propose a new mobile augmented reality authoring system that possesses unique characteristics compared to existing ones. The unique characteristics of the proposed system are a gesture-based interface, a lighting estimation procedure and a multi-freezing mode. The proposed system has been evaluated through user testing. The result of the user test shows that the proposed system is useful, easy to use, easy to learn, and satisfactory. Users can create augmented reality contents with shadows on site easily with the proposed system.

Keywords: In-situ authoring, modeling, augmented reality, gesture-based interaction, lighting estimation

1 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 with new experiences. The rapid spread of smart mobile devices such as smartphones and smart pads has made it possible to experience AR on smart mobile devices. Various AR applications have been developed on mobile devices 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 shifted from developer-created contents to user-created contents so the need for developing authoring systems for general users has emerged.

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

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

Langlotz et al. proposed in-situ authoring for mobile AR for small indoor and large outdoor working spaces [4]. A user begins authoring by generating 3D primitives, such as cubes and cylinders. The user scales or moves the generated 3D primitives or applied textures to them to create AR contents. Additionally, the system allows for annotating 2D contents and provides a freeze mode. This system is useful for indoor and outdoor AR environments. However, it can only create simple 3D models by extruding 2D footprint. This system has limitations with regard to creating realistic AR contents.

The proposed system shares some characteristics of the system developed by Langlotz et al. and uses 3D primitives and a freeze mode. However, the proposed system uses primitives as the basic blocks that can be combined to create more complex 3D models and modifies the freeze mode to improve efficiency. The proposed system also provides a lighting estimation procedure to create realistic AR contents.

∗ Corresponding author e-mail: jwlee@sejong.ac.kr

© 2015 NSP
Natural Sciences Publishing Cor.

2 The proposed system

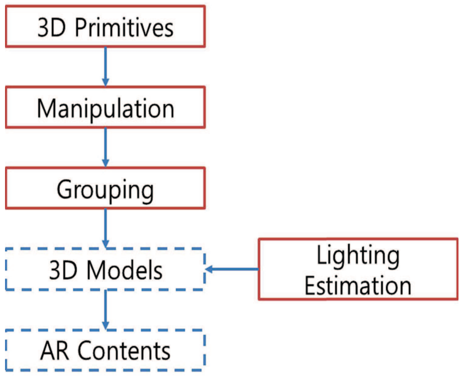
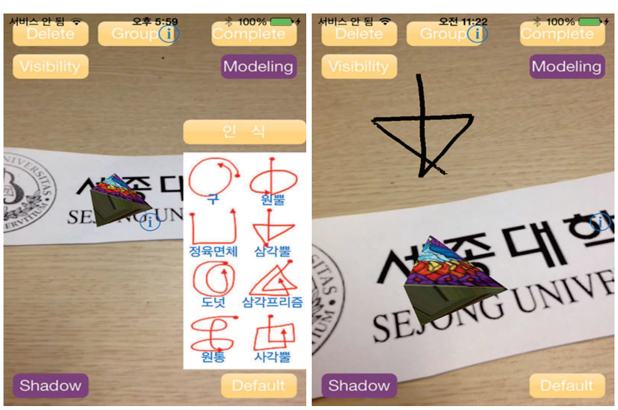


Fig. 1: The system flow of creating AR contents

The proposed system can be divided into four parts as shown in Figure 1. A user selects 3D primitives and manipulates them to the according 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 lighting positions or directions.

2.1 3D primitives

3D primitives are basic blocks that are used to create 3D models in the proposed system. 3D primitives consist of eight basic three-dimensional shapes (cone, cube, cylinder, sphere, triangular pyramid, square pyramid, triangular prism and torus). We chose the eight basic three-dimensional shapes because these shapes are familiar to people. People generally learn these shapes in their childhood [5].



(a) (b)

Fig. 2: Menus and gesture-based interface (a) Menu driven interface (b) Gesture-based interface

A user selects some of the 3D primitives and combines them to create a target 3D model. The user can choose one of 3D primitives using a menu driven interface or a gesture-based interface (Figure 2). We provide two types of interfaces so the user can choose one in which he or she prefers. A corresponding gesture is shown on each menu button so the user can remember a gesture and then use the gesture-based interface next time if he or she wants. The gesture-based interface can be designed not to occlude any information on the display while menus occlude some information as shown in Figure 2. This property of the gesture-based interface is important for mobile devices with smaller displays such as smartphones.

A gesture-based interaction is used to load a 3D primitive to an AR scene. The user draws a line drawing similar to a target primitive on the display of a smart mobile device. The proposed system recognizes the user's drawings using the algorithm similar to the one described by M. Elmezain et al. in [6]. There are three main stages (gesture tracking, feature extraction and classification) in the proposed gesture recognition procedure. We use the orientation feature as the 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 equation (1).

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

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)[7,15].

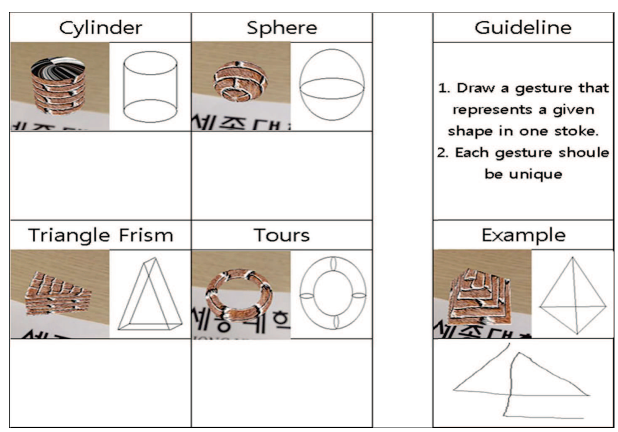


Fig. 3: The part of the user test form used for developing gestures

Currently the proposed system can recognize eight gestures. 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

© 2015 NSP
Natural Sciences Publishing Cor.

re-estimation algorithm. Each sequence is classified by the Viterbi algorithm [7,15].

Gestures for 3D primitives were developed from a user test. We asked ten possible users, who were college students, to draw unique gestures that represent given shapes in one stroke. Users were provided with one testing form that contained 3D models and their corresponding 2D drawings with the sample gesture of one primitive, triangular pyramid. The part of the testing form is shown in Figure 3.

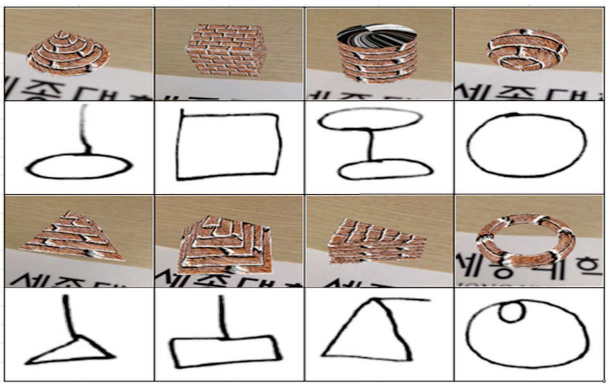


Fig. 4: Gestures created by one test user

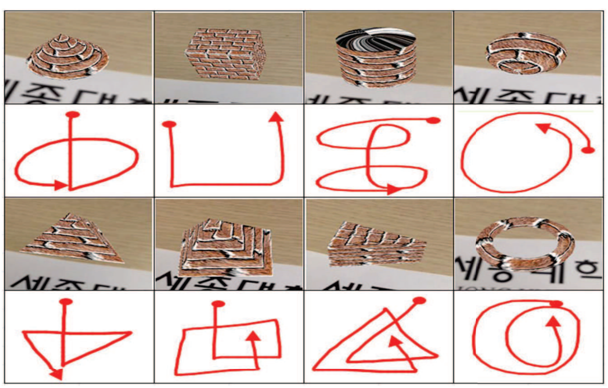


Fig. 5: Gestures used to load primitives in the proposed system

Test users generally represented 3D primitives using side views derived from either the top or bottom shape of a 3D primitive (Figure 4). Test users drew gestures that were similar to the side views of cubes, spheres, triangular prisms and torus. Test users drew gestures consisting of side views and/or the bottom shapes of a cone, triangular pyramid and square pyramid. The cylinder object was represented mostly using top and bottom shapes. Based on this result, we developed a gesture for each 3D primitive as shown in Figure 5. Gestures similar to side views of primitives were developed for cubes, spheres, triangular prisms and torus. Gestures containing a bottom shape of 3D primitives were developed for cones, triangular pyramids and square pyramids. The gesture of a cylinder was developed using its bottom and top shapes.

These gestures were tested with three participants. They tested each gesture twenty times. The recognition result was categorized into three groups: 1) gestures not detected even though they were in the database 2) gestures recognized correctly 3) gestures recognized incorrectly. The gesture recognition rate R was computed using equation (2) where N is the number of gestures and N_C is the number of correctly recognized gestures. The resulting average recognition rate was 98.3%.

$$R = \frac{N_C}{N} \times 100\%. \tag{2}$$

2.2 Manipulation and grouping

The Manipulation is composed of moving, rotating and scaling the virtual models and adding textures to them. We developed a dynamic constraint-based user interface to manipulate virtual objects. 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 mobile device.

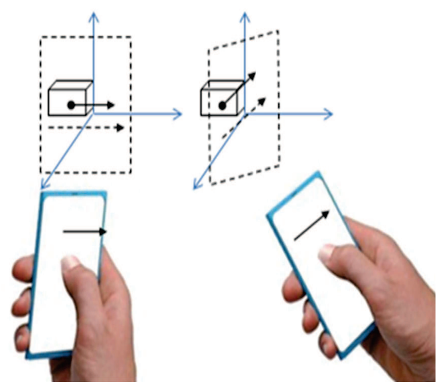
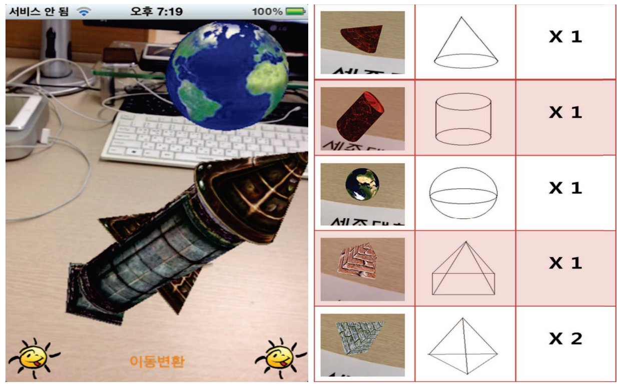


Fig. 6: Dynamically changing constraints

The constraint planes for translation motions are shown in Figure 6. The orientations of the left and the right mobile devices are different so the different corresponding constraint planes are defined. The dotted rectangular shape in Figure 6 indicates a constraint plane. Touch inputs on the display of a mobile device are directly mapped to the motions of a virtual object onto a 2D constraint plane. The virtual object can be scaled along any direction using the interface. A user can modify the size of a virtual object using the scaling procedure to create diverse models.

In the Grouping part, 3D primitives are grouped together to form a single model similar to our previous work [8,9]. A 3D model is created by combining primitives together in a similar way children assemble toy blocks to create an object. Examples are shown in Figure

© 2015 NSP
Natural Sciences Publishing Cor.



(a) (b)

Fig. 7: The models created by combining primitives. (a) Two models (b) Primitives used to build the two models

7 with 3D primitives used to create the models. The globe was created with a sphere. The rocket was created by combining a cone, a cylinder, a square pyramid, and two triangular pyramids. Textures were applied to the two models for adding realism to the models.

2.3 Light position/direction estimation

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

To render shadows correctly, we need to estimate the positions or directions of lighting sources. We developed two approaches to estimate them. The first approach utilizes the shadow and the virtual model of a real object. We apply the shadow estimating process developed in [9] to the proposed system. First, one real object in an AR scene is selected as a reference object (Figure 8 (a)). It is overlaid by the corresponding virtual model (Figure 8 (b)). It is better to select a simple object as a reference object because a user has to create a corresponding virtual model. Second, a user manipulates a virtual lighting source shown in Figure 8 (b) as a small sphere. When the virtual shadow created by the virtual lighting source is rendered closely to the real shadow of the reference object, we stop the estimating process as shown in Figure 8 (c). We consider the position of the virtual light as the estimated position of the real lighting source. If there are multiple lighting sources, we can repeat this procedure multiple times to estimate their positions or directions.

The second approach uses positions or directions of real lighting sources. A user moves around an AR environment and aligns the line connecting the center of the AR environment and his/her mobile device with the line connecting the center of the AR environment and a target light source. If the target lighting source is reachable by a user, the user can estimate the position of the target lighting source by putting the mobile device in front of the target lighting source. If the target lighting source is far from the user (e.g. lighting sources on a ceiling and sunlight), the user uses touch interaction to move the virtual light source along the estimated direction. We stop moving when the shadow created by the estimated lighting source is close enough to the shadow of the real object. (Figure 9)

(a) (b) (c)

Fig. 8: Estimating 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

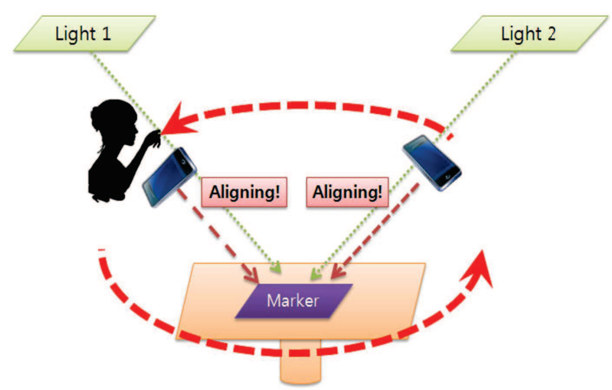


Fig. 9: Estimating lighting direction by aligning the mobile device with the direction of a light

The estimated lighting position or direction is used to create shadows of virtual models. We can also select one of predefined lighting types (fluorescent light, incandescent light bulb, sunlight) to improve the realism of the AR environment. The virtual shadows similar to real ones can be created using the proposed lighting position (or direction) estimation procedure and the selected lighting types as shown in Figure 10. We only applied simple hard shadows because of limited computation power on mobile devices. When the

© 2015 NSP
Natural Sciences Publishing Cor.

Fig. 10: Comparing virtual and real shadows in multi lighting condition

processing power of mobile devices is improved, more complex shadows can be rendered using the same lighting estimation procedure.

2.4 Multi-freezing mode

Tracking the camera attached to a smart mobile device (e.g. smartphone and smart pads) is required to use the presented authoring system to create AR contents. We initially developed the system using 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 [10, 11]. Since the tracking is required for authoring AR contents, we also added the freeze mode similar to existing authoring systems [4, 12, 13] to relieve possible fatigue caused by holding a mobile device for a long time. Users complained of difficulties caused by holding a smart device steady while creating AR contents in the previous user study.

Fig. 11: Multi-Freezing mode

The proposed freeze mode differs from existing ones. We call it a multi-freezing mode. We use multiple captured scenes to locate an augmented 3D object at the correct location in the AR environment. 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 in the 3D space. This multi-freeze mode was introduced in [13] and we applied it to the proposed system. One difference from the previous research is the way of showing the results. We only showed the augmented result in the main view in our previous research. In the proposed system, we show augmented results in the main view and the subviews, which are located at the bottom of the screen (Figure 11). Using the proposed approach, a user can view the results from several viewpoints at the same time, which reduces the number of view selections to locate a virtual object.

3 Scenario of using AR authoring application

The Figure 12 shows a scenario of creating AR content using the AR Authoring application, which is developed using the proposed authoring system. To start the application, a user has to prepare any image and a smart mobile device such as a smartphone or a smart pad (Figure 12 (a)). When the user captures the image using the camera on the smart mobile device, the image is stored in the database and used to track the smart device (Figure 12 (b)).

Menus to create AR contents are displayed as soon as the tracking starts (Figure 12 (c)). The object loading button on the middle of the right side of the display will activate the object-loading interface (Figure 12 (d)). The user can use gestures or menus to load one of 3D primitives. The detail information is described in the Section 2.1. Figure 12 (e) shows one cylinder and one cone, which are loaded by the user, with the default texture. The cylinder and the cone can be translated, rotated and scaled using the interface described in the Section 2.2 and the manipulation mode-changing button. The mode-changing button, which is located at the bottom right of the display, sets a manipulation type (Figure 12 (f)). These 3D primitives are grouped together using the grouping button located at the upper left side of the display to create a part of a castle wall (Figure 12 (f)).

The user can add textures to 3D primitives by using the texture interface that can be activated by the texture loading button located in the upper middle part of the display (Figure 12 (c, g)). The user first selects a 3D primitive on the display, after which he or she selects a texture by scrolling textures horizontally. The selected texture will be textured on the selected 3D primitive. Additional primitives can be loaded and grouped together to create the target model shown in Figure 12 (h).

© 2015 NSP
Natural Sciences Publishing Cor.

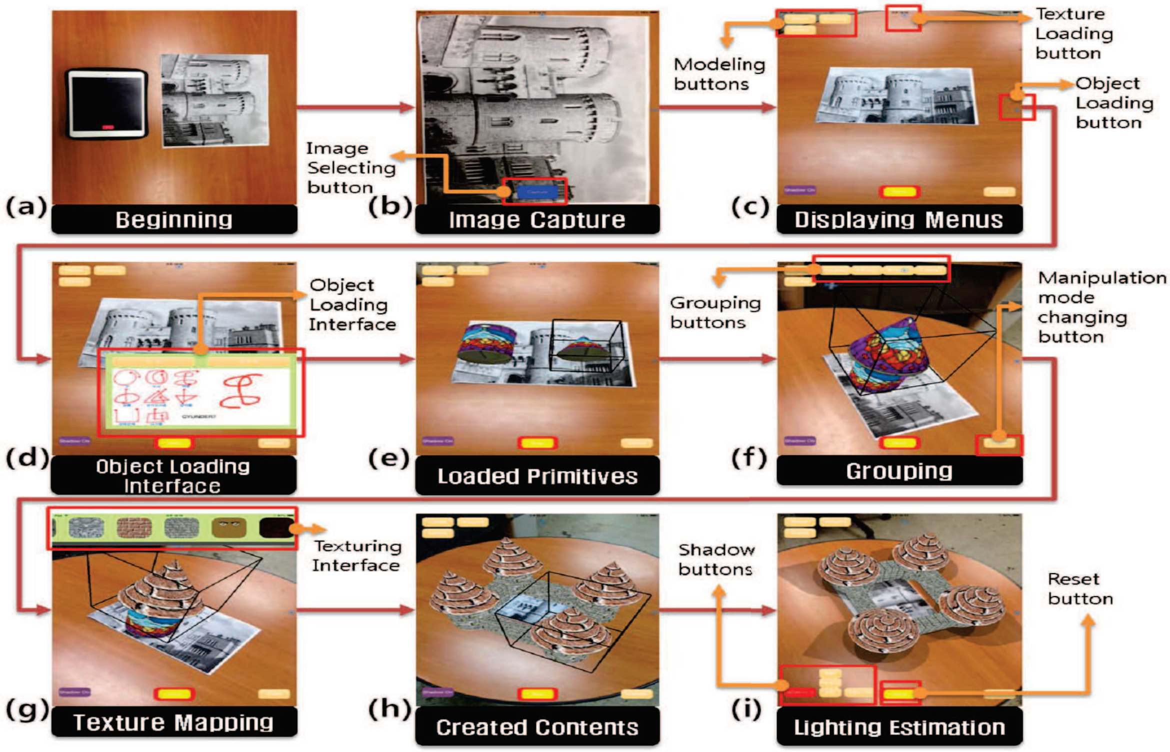


Fig. 12: Example scenario of creating AR contents using the AR authoring application

Shadows can be added by estimating lighting positions or directions. The shadow buttons located at the bottom left side of the display will activate the lighting estimation mode. The user can estimate lighting positions or directions using one of two approaches described in the Section 2.3. The shadows of the created contents are rendered based on estimated lightings (Figure 12 (i)). The user can click the Reset button to create new content with a new image.

4 Evaluation

We designed and performed a user experiment to evaluate the presented mobile AR authoring system. We examined the following four aspects in the experiment: 1) usefulness 2) ease of use 3) ease of learning and 4) user satisfaction.

4.1 Participants

Ten participants with normal or corrected vision took part in the experiment. They volunteered to be in the experiment and given small gifts in return for their participation. All participants are in their twenties and familiar with smartphones. Seven participants use smartphones more than three hours every day. One participant uses a smartphone for two hours every day. Other two participants use it for one hour every day. All of the participants had heard about AR and nine among them had used some kinds of modeling tools before. We selected young participants for the experiment because they were more willing to learn new technologies.

4.2 Experiment

The proposed authoring system was presented to participants on a smart pad (iPad mini: Dual core 1000MHz ARM Cortex-A9 processor, 5 megapixels camera, 1024 x 768 resolution, 312g weight). We tested the system with smart pad because the display area of a smartphone was too small to fully test the proposed system even though the system could be demonstrated on a smartphone.

Table 1: Pre-Experiment Questionnaire

1	Have you heard about augmented reality before?
2	Have you used any 3D modeling tools(e.g. 3DMax) before?
3	How many hours do you use a smartphone every day? (1) None (2) 0.5 hour (3) 1 hour (4) 2 hours (5) More than 3 hours

© 2015 NSP
Natural Sciences Publishing Cor.

Table 2: Subjective Preference Questionnaire

Lighting Position/Direction Estimation Procedure	
1	It is useful for estimating lighting positions (or directions).
2	It helps me to estimate lighting position (or directions) effectively.
3	It is easy to estimate lighting positions (or directions).
4	I can estimate lighting positions (or directions) without written instructions.
5	I easily remember how to estimate lighting positions (or directions).
6	It is easy to learn to estimate lighting positions (or directions).
7	I am satisfied with the way to estimate lighting positions (or directions).
8	The way to estimate lighting positions (or directions) is fun to use.
Modeling Procedure	
1	The modeling procedure is useful.
2	It helps me to create 3D objects effectively.
3	The modeling procedure is easy to use.
4	I can use the modeling procedure without written instructions.
5	I learned to use the modeling approach quickly.
6	The modeling system is easy to learn.
7	I am satisfied with the modeling approach.
8	The modeling system is fun to use.
The Proposed System	
1	It is useful.
2	It helps me become more effective.
3	It makes the things I want to accomplish easier to get done.
4	It is easy to use.
5	It is simple to use.
6	I can use it without written instructions.
7	I learned to use it quickly.
8	I easily remember how to use it.
9	It is easy to learn.
10	I am satisfied with it.
11	I would recommend it to a friend.
12	It is fun to use.
13	I am satisfied with the contents I created using the system.

At the beginning of the experiment, we asked participants to fill out a pre-experiment questionnaire to assess their knowledge of the areas related to the testing system. The pre-experiment questionnaire used in the experiment is shown in Table 1. Next, we gave a brief explanation about augmented reality, the objective and procedure of the experiment and the functions of the presented system. Following the explanation, participants tried the presented system until they decided to create the given augmented reality content. We showed participants an image, which is one view of the target augmented reality environment, and asked them to create an augmented reality environment similar to the given one.

Fig. 13: Sample contents created by test users

Participants had to create two models (a rocket and globe), to manipulate them to the desired location and orientation, and to estimate lighting positions. The sample results created by participants are shown in Figure 13. After participants created a target augmented reality environment, we asked them to fill out the questionnaire shown in Table 2 to measure preferences about the proposed system and to collect opinions about the system. We selected some questions in the USE Questionnaire developed by Arnold M. Lund [14]. We did not use all questions in the USE Questionnaire so as to reduce the time required to fill out the questionnaire.

4.3 Evaluation results and discussion

The task completion times are summarized in Figure 14. The average task completion time was 9 minutes and 37 seconds and the standard deviation was 2 minutes and 12 seconds. The task completion time was the sum of the modeling time and lighting position estimation time. The average modeling time was 8 minutes and 53 seconds and the average lighting position estimation time was 44 seconds. These data were measured after participants practiced with the system for less than 10 minutes. Only one participant practiced for 25 minutes. That participant wanted to use the system longer because she was enjoying it.

We measured the subjective preference for the proposed system that has two procedures, the modeling and the lighting position (or direction) estimation procedures, using the questionnaire with the 7-point Likert scale. The subjective preferences of the proposed system and two procedures are summarized in Figure 15.

Average subjective preference scores were higher than 5 points for all four aspects; usefulness, ease of use, ease of learning, and satisfaction. The standard deviations of subjective preference scores ranged from 0.74 to 1.68. The highest standard deviation came from the ease-of-use aspect of the proposed system. One participant gave 2 points for all questions regarding the ease-of-use aspect.

© 2015 NSP
Natural Sciences Publishing Cor.

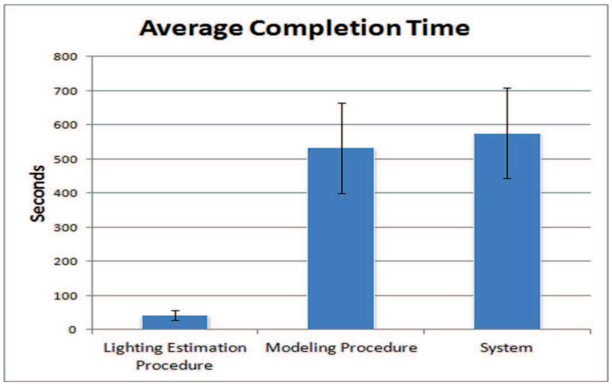


Fig. 14: Average task completion times for the system and lighting estimation and modeling procedures

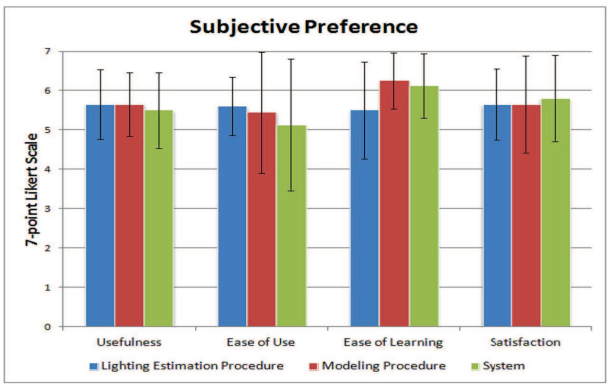


Fig. 15: Average subjective preference scores for the system and lighting estimation and modeling procedures

Participants considered that the modeling procedure and the proposed system were easy to learn. They gave the highest scores for the ease-of-learning metric for the modeling procedure and the proposed system. Since participants were already familiar with construction toys like blocks, they were able to learn the modeling procedure and the proposed system easily. The ease-of-learning metric of the lighting estimation procedure achieved the lowest score even though participants considered the lighting procedure easier to use than the modeling procedure and the proposed system. We think that the unfamiliar concept of estimating lighting position caused the lower ease-of-learning score for the lighting estimation procedure.

We also asked participants about the contents they created using the proposed system. Most participants were satisfied with their outcomes. Three participants were not satisfied with their outcomes and subsequently gave the lowest average preference score among the group. The resulting outcomes could affect participants preference about the proposed system.

Participants also expressed positive and negative points of view about the proposed system. Most participants commented that the system was easy to use. Some participants said that the system was fun, especially when adding shadows to the scene. Some negative comments were made about the functionalities of the proposed system. They suggested adding a ”Copy” function to allow them to copy the virtual models and a ”Cancel” function that would let them go back to the previous state while modeling a virtual object. They also complained about the scaling and rotating operations.

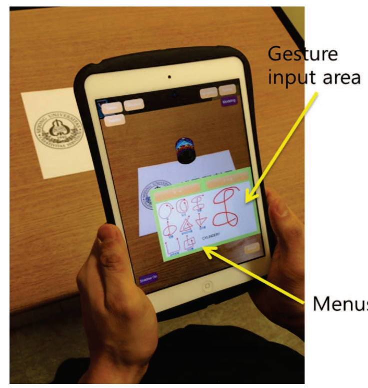


Fig. 16: Providing menus and a gesture input area concurrently to compare the frequency in use of a menu driven interface and a gesture-based interface

We also compared the preferences concerning two interfaces, a menu driven interface and a gesture-based interface, used for loading primitives to a scene. We gave menus and a gesture input area concurrently to participants when they wanted to load a primitive as shown in Figure 16 and measured the number of uses of menus and gesture inputs to create the given augmented reality world. The average number of uses for menus and gestures was 3.3. The number of participants who used only one interface, menus or gestures, is 2. Neither interface was preferred by participants. This was a surprise. We assumed participants would use menus more than gestures because they were more familiar with menus. We expect that users will use the gesture-based interface more than menus when they become familiar with the gesture-based interface.

5 Conclusion

Various augmented reality applications have been developed on smart mobile devices such as smartphones and smart pads. Application developers generally provide contents for these applications. Few applications have included authoring functionalities with which users can create their own contents with limited capabilities. We proposed a new mobile AR authoring system that enables users to create more realistic augmented reality contents.

© 2015 NSP
Natural Sciences Publishing Cor.

The proposed system has unique characteristics compared to existing ones. The unique features can be summarized as:

1. The primitive-based modeling system is easy to learn and use. This type of modeling system has been introduced in previous research studies. We added the gesture-based interface to the modeling system. Different gestures corresponding to different 3D primitives were derived from the results of the user test. We found that users generally drew gestures that use side views, which consist of the top and bottom shapes of the 3D primitives. Gestures for the proposed system were created using these results so that users could easily remember them.

2. We developed a new lighting estimate procedure. Users only need to move around the augmented reality space to align a light direction and their mobile device to estimate the lighting position. If the light is located far from the users, users first estimate the lighting direction and move the estimated lighting along the estimated direction to estimate the lighting position. Users estimated lighting positions in real environments easily. They estimated 2 lightings on a ceiling in less than a minute.

3. We applied the multi-freezing mode similar to that in previous research to the proposed system. In previously conducted research, users could only view an augmented scene on the main screen. We modified the previous research so users could view augmented scenes on the main screen and subviews on the bottom of the screen. Using this functionality, users did not move views as much as was required by previous research to locate the virtual object to a target position.

The proposed system has been evaluated to test the performance and the subjective preference of the system. Important findings are as follows:

1. Test users considered the proposed system and the modeling and lighting estimation procedures useful, easy to use, easy to learn and satisfactory. The average preference scores in the 7-point Likert scale for all areas were higher than 5 points.

2. The system allowed users to create augmented reality contents in a short time after practicing for less than 10 minutes. This also showed that the system was easy to learn and easy to use for creating augmented reality contents.

3. The gesture-based interface was preferred just as much as the menu-driven interface. This was an encouraging result because it could be difficult to use menus for smartphones with small display areas. Menus can occlude too much area for devices with small display areas. This can result in preventing users from viewing important information on the display area. The gesture-based interface can be used for devices with small display areas. The gesture input area can be transparent so users can view important information while they are using the gesture-based interface.

In addition, the user test suggested that we had to add and modify a few functions of the proposed system. We need to add a ”Back” button and ”Copy” button. Test users wanted to move to the previous state using the back button when they made some mistake. Currently, users have to rebuild from scratch. Test users also wanted to copy an existing object to create the duplicate of the existing one, which was not allowed in the system.

Test users also complained about the scaling and rotating functionalities. They mentioned it was difficult to scale and to rotate the virtual object with the system. We need to modify the system to provide users with more intuitive scaling and rotating functionalities.

The proposed system still needs additional improvement. The modeling procedure could be improved by adding a subtraction function. Users create virtual objects by adding primitives in the proposed system. Users can create more complex virtual objects by subtracting some part of a primitive using another primitive. We could also add a vision-based modeling function to model real objects using a smart-devices camera.

Acknowledgement

This research was supported by Ministry of Culture, Sports and Tourism and KOCCA in the CT R&D Program 2011. Also, this research (Grants No. C0133918) was supported by the Business for Cooperative R&D between Industry, Academy, and Research Institute funded by the Korea Small and Medium Business Administration in 2013. The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] W. Piekarski, B. Thomax. ”Tinmith-metro: new outdoor techniques for creating city models with an augmented reality wearable computer.”, Fifth International Symposium on. IEEE, 31-38 (2001).

[2] S. Guven, S. Feiner, O. Oda. ”Mobile augmented reality interaction techniques for authoring situated media on-site.”, the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality. IEEE Computer Society, 235-236 (2006).

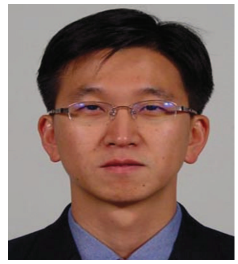
[3] C. Liu, J. Diehl, S. Huot, J. Borchers. ”Mobile augmented note-taking to support operating physical devices.”, Mobile Augmented Reality: Design Issue and Opportunities 1st Workshop on Mobile Augmented Reality, MobileHCI, (2011).

[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.”, Personal and Ubiquitous Computing, **16**, 623-630 (2012).

[5] Common 3D shapes. http://www.mathsisfun.com/geometry/common-3d-shapes.html.

© 2015 NSP
Natural Sciences Publishing Cor.

[6] M. Elmezain, A. Al-Hamadi, and B. Michaelis. ”Hand trajectory-based gesture spotting and recognition using HMM.”, International Conference on Image Processin, 3577-3580 (2009).
[7] X. D. Huang, Y. Ariki, and M. A. Jack. ”Hidden Markov Models for speech recognition.”, Edinburgh Univ. Presse, (1990).
[8] J. Y. Park and J. W. Lee. ”Tangible augmented reality modeling.”, Entertainment Computing, LNCS No. 3166, 254-259 (2004).
[9] D. Trien and J. W. Lee., ”3DARModeler: a 3D modeling system in augmented reality.”, Computer Systems Science & Engineering, **4**, (2008).
[10] Qualcomm Vuforia SDK. https://www.vuforia.com
[11] Metaio SDK. http://www.metaio.com/sdk
[12] G. A. Lee, U. Yang, Y. Kim, D. Jo, K. H. Kim, J. H. Kim and J. S. Choi. ”Freeze-Set-Go interaction method for handheld mobile augmented reality environments.”, ACM Symposium on Virtual Reality Software and Technology, 143-146 (2009).
[13] J. G. Lee and J. W. Lee. ”Interface for In-situ Authoring of Augmented Reality Contents on a Mobile Device.”, Information-An International Interdisciplinary Journal, **16**, 8037-8052 (2013).
[14] A. M. Lund. ”Measuring Usability with the USE Questionnaire.”, STC Usability SIG Newsletter, **8**, (2001).
[15] Eddy, Sean R. ”Profile hidden Markov models.” Bioinformatics 14.9, **14** 755-763 (1998).



JongWeon Lee was born in 1966. He received the B.S. degree in Electrical Engineering from Ohio University in 1989, the M.S. degree in Electrical and Computer Engineering from University of Wisconsin at Madison in 1991, and Ph.D. degree in Computer Science from University of Southern California in 2002. He is presently Professor of Department of Digital Contents at Sejong University. His research interests include augmented reality, human-computher interaction and serious game.

HanKyu Yoo was born in 1985. He received the B.S. degree in Internet Engineering from Sejong University, Seoul in 2012, and the M.S degree in Digital Contents from Sejong University, Seoul in 2014.