Using a Mobile Device as an Interface Tool for HMD-based AR Applications

Kyong Joon Lee
Korea Institute of Science and Technology
39-1 Hawolgok-dong, Seongbuk-gu
Seoul, 136-791, Korea
+82-2-958-6825
kjoon@imrc.kist.re.kr

Sang Chul Ahn
Korea Institute of Science and Technology
39-1 Hawolgok-dong, Seongbuk-gu
Seoul, 136-791, Korea
+82-2-958-5777
asc@imrc.kist.re.kr

Hyoung-Gon Kim
Korea Institute of Science and Technology
39-1 Hawolgok-dong, Seongbuk-gu
Seoul, 136-791, Korea
+82-2-958-5771
hgk@imrc.kist.re.kr

ABSTRACT
In this paper, we suggest that we can use our mobile devices as an effective interface for HMD-based augmented reality (AR) applications. These days most of us can possibly send short message with our mobile phones and watch the movies on our PDAs. Personal hand-held devices are well customized that they deserve to be familiar interfaces for interaction with AR applications. We can also add more interaction metaphors as mobile devices are designed to be programmable and extensible for some extra modules. Furthermore, the display panel of mobile devices makes a good marker which is changeable to variable features on user’s demand.

Categories and Subject Descriptors
H.5.2 [Information Interfaces and Presentation]: User Interfaces – Input devices and strategies.

General Terms
Design, Reliability, Human Factors

Keywords
Mobile phone, PDA, mobile device, Augmented Reality, interface.

1. INTRODUCTION
Since Ivan Sutherland has introduced the first augmented reality (AR) application based on see-through HMD in the 1960s, there have been a number of prominent applications developed and showed to the public. Some of these works contributed to the advance in AR display and the others to tracking technology, but few have provided appropriate interaction tool to manipulate and modify virtual object rendered over physical object [2]. Therefore it is strongly necessary to design a proper method to interact with the AR application. And now that AR applications are basically on wearable computing environment, we have to keep in mind that the interface tool should be intuitive and simple, without any difficulties to carry out.

In terms of this, we have found that mobile devices, e.g., mobile phone or PDA, can be an effective interaction tool for AR applications. Among the several approaches that Ullman describes in his book to provide guidelines designing new user interfaces [3], we chose the method to use “found” object, which is originally designed for other purpose, but rearranged as an interface element. We developed the software program that is loaded to the device and runs on the device, making it work as an interface tool for AR applications.

In next section we briefly review related works about AR interaction technique and current approaches in mobile AR. Next we explain system architecture and the features of our mobile AR interface by means of interface design principle. In section 5 and 6 we describe in detail how we implement our work and show sample AR applications using our interface. Finally, in section 7 we present a summary and some conclusion remarks.

2. RELATED WORK
We introduce several methods that have been suggested in AR interaction techniques, AR interface and AR setting on mobile devices.

Kolsch at al. developed computer vision methods for hand gesture-based user interfaces [4]. These interfaces provide intuitive and simple way to interact with the AR system, but they have trouble with the limitation of tracking technology, e.g. speed and accuracy.

The HITLab NZ [13] introduced several tangible user interfaces including FingARTips [5] and AR MagicLenses [6]. FingARTips detects markers attached to index finger and thumb as a nipper and the user manipulates virtual object using gestures like picking and releasing of the fingers. Looser at al. created marker attached trackball to implement AR MagicLenses. The trackball is working like a conceptual magnifying glass as the user examines the virtual object. These interfaces were built to achieve specific purpose using principles of general interface design technique [3]. Therefore, they are very intuitive and well suited for the use of original purpose but they just perform limited functions and the user might take some time to get acquainted to new type of interface.
For mobile AR, Henrysson at al. introduced ARTennis, a simple collaborative AR game on a mobile phone [7], while Wagner at al. showed Invisible train game [8] on PDAs using a custom port of the AR Toolkit library to the mobile devices. Although they take the advantage of simple AR setting implemented on the mobile devices itself, they suffer from the restriction of rather small display panel of the mobile device, let alone all the good features of HMD-based AR setting, such as an immersive view, three-dimensional effect and free both hands. In fact, they are considered as an AR system itself, rather than an AR interface for interaction.

3. SYSTEM ARCHITECTURE

Figure 1 shows system overview of typical AR application using our mobile device as an AR interface. Stereo camera is attached on the frontal part of HMD, sending the view of the user’s original sight to AR system. Then AR system detects the marker on mobile device, rendering virtual object attached on the marker and sends it back to HMD. It is important to notice that user’s input using keypad transmitted through wireless communication to AR system and affects rendering of virtual object.

4. INTERFACE FEATURES

As Billinghurst presented the key interface elements for designing AR interfaces [2] (Figure 2), the basic goal of interface design is mapping user input onto computer output using an appropriate interaction metaphor. We tried to design our interface based on this goal.

4.1 Familiarity

In recent years, as the personal hand-held devices are widely deployed and almost every people in social activity have at least one mobile device, we are well accustomed to the use of these devices. A number of people don’t even look at the keypad while they write a short message with their own mobile phone. That is, these devices are already easy to use and customized. Therefore if these devices could work as AR interaction tools, it would be possible to significantly shorten the time for the user to get used to the interface. This contributes to physical elements of key interface elements as familiarity gives more efficiency to the user.

4.2 Extendible interaction metaphor

Almost every hand-held mobile device has at least one data connection terminal on its peripheral. We can attach some sensor module to add the interaction metaphor beyond controlling button or keypad. For instance, if accelerometer sensor is attached to the device, we can expect more intuitive interaction using our gesture or motion. Commercial accelerometer sensor module is already available in the market, with which the user can enjoy the mobile game with gesture, see Figure 3.

Even better, as today’s mobile devices getting more built-in functionality. For instance, Samsung’s up-to-date mobile phone has several built-in sensors like accelerometer, gyro, GPS, digital compass and thermometer; they could provide information about gesture, motion, location, direction and temperature. This information can be applied to make more intuitive interaction metaphor in near future.

4.3 Variable marker

Typical AR application requires accurate tracking of physical object because virtual object needs to be rendered over physical object. The display panel of mobile device makes a great visual marker as it illuminates itself (unlike printed marker) and it is detectable in a rather harsh environment, say, in a dark room. As we assumed in subsection 4.2, if an accurate posture sensor added, it would be far easier to track the physical object.

In addition, single display panel of mobile device is able to play the role of a number of markers as we can simply change the digital marker on the display panel. We don’t have to hold various analog paper markers with both hands (or select it from messy markers on the table) but we just manipulate a button and display panel already express another marker at once.
5. IMPLEMENTATION

We have implemented our mobile device AR interface on two kinds of representative hand-held device: mobile phone and PDA. The software program running on those devices draws the appropriate marker on their display panel and interacts with the AR system through wireless communication channel. We use the ARToolkit [1] in the AR system for tracking a visual marker and rendering an augmented object on the marker.

The hardware setup of our system consists of HMD (Deocom Victor SX-1) and a stereo camera (two small USB digital camera modules, Logitech Quickcam) attached on the frontal part of HMD, see Figure 4. The HMD has diagonal FOV of 80° and 1280x1024 of default resolution. Stereo camera input is streamed into the AR system (main computation unit, see Figure 1) and output back into HMD in order to perform see-through effect for each eye so that the user feels three-dimensional effect of virtual object. For the main computation unit, we use an ordinary desktop computer with Pentium 4 processor and dual display output. We also finished the test on a laptop (Sony VGN-S38LP) with a single camera input and a single display output. (So far, no laptop computer supports dual display output terminal to be connected to each LCD input terminal of HMD.) But we generally worked with the desktop in order to get the user more immersive and realistic AR experience by stereo vision.

5.1 Mobile phone

In general, mobile phone vendors or telecom service providers support software development kit (SDK) for their product, Qualcomm’s Brew [9], for example. We used the SDK called SK-VM [10], clean-room implementation of J2ME, developed by SK-Telecom Corporation in Korea. The SDK typically has the functions handling display panel, sound, vibration, button inputs, wireless communication and external hardware module connected through data communication terminal. The software running on the mobile phone is designed to show appropriate marker on user’s demand and send the button interaction to the AR system through wireless internet provided by telecom service providers (or Bluetooth, if possible). Additional external module attached to the phone (see Figure 3) gives information to the software using serial communication protocol. We upload developed software on a specific WAP (Wireless Application Protocol) server and have it downloaded to a mobile phone in the same way we download mobile game contents. That is to say, every general mobile phone can possibly play the role of AR interface without any special care. Users certainly feel more comfortable with their own phone, which is extremely familiar device for them.

5.2 PDA

PDA usually has its own OS and its API provides far more functionalities than mobile phone’s SDK. It also has larger display panel than the mobile phone, and built-in wireless LAN card and Bluetooth module as well, which reduces the cost of wireless communication (Wireless internet for the mobile phone costs a lot!). We used HP iPAQ 5550, where Microsoft PocketPC 2003 installed as an OS, to run and test the software application. The software is created on the desktop using the IDE, Microsoft Embedded C++ 4.0 in order to work same functions as stated in subsection 4.1. Developed software is downloaded through USB sync cable. Additional module is also acceptable for the PDA.

6. SAMPLE APPLICATION

In this section, we introduce sample applications that show how a mobile device works as an AR interface. Two applications are presented; AR 3D Tetris is AR modification for 3D Tetris [11], of which the goal is to complete a layer with various types of falling blocks without gap which will cause the layer to disappear. Mobile AR Book is an electronic book with virtual contents on the marker which is changing as the user press the next page button, which is analogous to turning a page in a physical book.

6.1 AR 3D Tetris

First, we implement AR 3D Tetris that runs on the display panel of mobile devices. We apply the 3D Tetris engine [11] for desktop computer to be rendered as real 3D scene using ARToolkit. In ordinary 3D Tetris on the desktop computer screen, we have to press specific keys to rotate and change the view of blocks, stacks and frames. But in our AR 3D Tetris, which is rendered over the marker of mobile devices, we can freely rotate the view as we just rotate the mobile device. To move a falling block, in normal 3D Tetris, we have to consider if the current view is rotated so that direction key is chosen correctly, which is very confusing and annoying job. In contrast, when we play the AR 3D Tetris, all direction key is quite intuitive; we just choose the direction and press that the arrow key pointing that direction. Figure 5 shows the PDA setup used for experiment and the AR 3D Tetris running appeared on the visual marker of the PDA.
We are going to develop the accelerometer sensor module to detect the tilt information of mobile device and have this information to move the falling block to the direction the device is tilted. As we tilt the device, the falling block moves along the slope pretending it is pulled by gravity. This would give more intuitive interaction metaphor to our AR interface.

6.2 Mobile AR Book
Using changeable marker on display panel of mobile devices, the Mobile AR book is easily implemented. For the previous AR Magic book [12] demonstrated by HITLab NZ [13], we have to print several markers on papers for each page of the book, but with our AR interface, we just need some image files of those markers. Interaction metaphor for turning a page would be pressing the arrow button of mobile device. Figure 6 demonstrates how the application runs on the mobile phone. It is remarkable that the virtual object is larger than the display panel of the mobile phone providing effect that the display panel is expanded.

Owing to the capability of interaction, we can change the content of the book or add some commentary on the book. Assumed that we read the educational AR book, like Physics, we might want to free-fall a virtual ball or change the direction of virtual magnetic field. And for the exercise problems at the end of the chapter, we could type the answer using the keypad of mobile phone.

7. CONCLUSION
In this paper we have described the use of mobile device as an interaction tool for HMD-based AR environment. We presented system overview and explained several features of our AR interface by means of the principle of AR interface design; it is already familiar and customized interface to the user who possesses it and supplementary interaction metaphors can be added as well as it has a potential to make a good marker for detection. Sample applications are provided to support our opinion in terms of usability and efficiency of the interface.

For the future work, we are going to find more interaction metaphors that can be added to the interface which is going to make the mobile device richer and more general interaction tool. We also have a plan to explore more advanced method using the
posture information combined with the visual information in order to detect accurate position and posture of the physical object.

8. ACKNOWLEDGMENTS
Software support related to 3D Tetris has been provided by Greg Zipkin and Zoe Wood in California Polytechnic State University, San Luis Obispo.

9. REFERENCES