A NEW ARCHITECTURE FOR A MULTIPLATFORM AUGMENTED REALITY SYSTEM

Andriamasinoro Rahajaniaina and Jean-Pierre Jessel

Institut de Recherche en Informatique de Toulouse, Université Paul sabatier, 118 Route de Narbonne, Toulouse, France

Keywords: Augmented Reality, Wearable Multimedia Computing, Ubiquitous Augmented Reality, Human-Machine Interface.

Abstract: In this paper we describe a new architecture for augmented reality (AR) multiplatform hardware device, which works in dynamic workspace environment with 3D virtual models. The users can interact with the virtual model using mouse, keyboard and stylus as interaction tools. The work plan is formed by ARToolkitPlus' fudicial multimarker. For adding virtual objects, we propose a virtual menu inspired by the metaphor of forward and next buttons. The work plan is augmented by the virtual workspace. The user can choose his virtual workspace in dynamic way using this virtual menu and can choose a virtual object suited to the current workspace.

1 INTRODUCTION

Generally, Augmented Reality (AR) is an interface technology that uses head worn, personal computer display or handheld displays to allow users to see virtual images superimposed over the real world. In AR applications three-dimensional computer graphics appear to be fixed in space or attached to real objects.

AR techniques have been shown in several application areas as architecture Anthony, W., Steven, F., Blair, M.,William, M., Theodore, K., 1996, education Hannes, K., 2003, and game Zsolt, S., Erik, E., Michael, G., 1998. However, there is still research needed to see the AR content using heterogeneous platform device. Many related AR systems have run on a wide variety of hardware device using, for instance, personal computer Moligev, D., Kiyokawa, K., Billinghurst, M., Pair, J., 2002 personal digital assistant (PDA) Daniel, W., Schmalstieg D., 2005 or cell phone as display device.

We propose in this paper, a new approach of AR system using heterogeneous hardware device (such as display device) to see augmented content. There have been several examples of tabletop AR system for immersive virtual environment. These previous researches have worked in static workspace and have run on homogenous device. Our work differs from these because it runs on heterogeneous hardware and allows user to change his virtual workspace and associating a virtual object with a fiducial marker in dynamic way.

In this paper, we first review previous work and then describe our system overview. After that, we discuss the application of our Augmented Reality system. Finally, we conclude and provide possible perspective for the future research.

2 PREVIOUS WORK

Most of previous works have tackled homogenous platform hardware and have worked in a static workspace. Among them, we could introduce several works that are presented hereafter.

First, Irawati, S., Green, S., Billinghurst, M., Duenser, A., Heedong, K., 2006 has been an AR system on multimodal tabletop inspired by VOMAR Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K., Tachibana, K., 2000 which has run on personal computer hardware. This system suggests simultaneously two interaction ways. Actually, users are in a position to use a paddle and vocal command. So in order to particularly disambiguate the user speech and gesture inputs, a combination of time-based and domain semantics is implemented. Moreover, the user was able to associate a virtual object with a marker using virtual menu. Furthermore, the superimposition of two or more

Rahajaniaina A. and Jessel J. (2008).
A NEW ARCHITECTURE FOR A MULTIPLATFORM AUGMENTED REALITY SYSTEM.
In *Proceedings of the International Conference on Signal Processing and Multimedia Applications*, pages 336-341
DOI: 10.5220/0001940103360341

virtual objects by holding account of the objects' properties and theirs positions (left, right, up, down) is allowed by this system.

Secondly, we could quote the ARTHUR project Aish, F., Broll, W., Stoerring, M., Fatah, A., Mottram, C., 2004. Its goal has been for a planning urban on a table. To see the mixed environment, each user had worn a Head-Mounted Display (HMD) with a high resolution. Interaction tools are placeholder, and, in order to change the user interface gesture recognition has been used. The database of virtual object was duplicated in each node and synchronization was made for each state's change.

The next project is MARE Raphaël G., Jean-Dominique G., 2002. It has designed an AR system on tabletop that has allowed several users collaborating. This system has used a stereo optical see-through HMD as visualization tools. It is attached with magnetic or optical tracker, its workspace is divided in two areas: a public one is reserved for collaborative interaction, and, a private one is dedicated to put the real tools (mouse, keyboard, PDA …) and virtual tools used by the users. The virtual menu, which is used by users for choosing a virtual object, has been a part of these tools. The system can support several types of interaction like navigation, selection and manipulation. It is used to control access manager and personal viewer of user.

Then, the EMMIE Andreas B., Tobias H., Steven F., Blair M., Clifford B., 1999 framework introduces a hybrid user interface for AR systems that enables information management using a wide range of hardware devices. EMMIE's environment manager component addresses the needs of Ubicomp by providing techniques such as mixed reality interaction and privacy management to organize virtual information on several displays shared by multiple users.

Mooser, J., Lu, W., Suya, Y., Neumann, U., 2007 Proposed an AR user interface framework specifically designed to expose disparate data sources through a single application server. It uses a multi-tier architecture to separate back-end data retrieval from front-end graphical presentation and UI event handling. For target recognition and pose estimation, the system uses TriCode fiducials. The recognition and pose estimation can run a mobile client in real time, with the identity of a newly detected TriCode then passed to the server as a single numerical value. Data can exist in many forms, such as relational databases, sensor readings, or locations of other users. Once the user interface structures have been built, the application server serializes them into a simple XML stream, which is sent back to the mobile client. Having received the XML stream, the client has been built its own local version of the user interface structure. The user has been shown the mixed content through a Sony Vaio UX Micro PC display.

The acclaim MARS (Mobile Augmented Reality Systems) project Höllerer, T., Feiner, S., Terauchi, T., Rashid, G., and Hallaway, D., 1999 presented in 1999 by Columbia University was one of the first truly mobile augmented reality setups which allowed the user to freely walk around while having all necessary equipment mounted onto his back.

The AR Phone project Mark, A., David, J., Daniel, C., Adam, H., 2003 was an AR system where the tracking task was dedicated to an AR server. Cell phone has been used as thin client and an access point was relayed the exchange between the client and the AR server. The tracking task is dedicated to ARToolKit Billinghurst, M., Kato, H., Weghorst, S., Furness, T. A., 1999 library running on the AR server.

More recent works have used lightweight wearable devices such as PDA and Smartphone as display device.

In Daniel, W., Thomas, P., Florian, L., Dieter, S., 2005, Daniel Wagner presented an AR system collaborative running on a PDA. The users have been able to interact with the virtual objects using a stylus. The tracking task is dedicated to Opentracker Reitmayr, G., Schmalstieg, D., 2001 library and is run on the PDA itself. The platform is based on studierstube Schmalstieg, D., Fuhrmann, A., Hesina, G., Szalav´ari, Z., Encarna, c., L.M., Gervautz, M., Purgathofer, W., 2002 framework optimized for a mobile light device. The system used ACE (the Adaptive Communication Environment) for network communication abstraction.

AR tennis Henrysson, A., Billinghurst, M., Ollila, M., 2006 proposed a system AR on a table running on a Smartphone. This last one used as tangible interaction and visual tools. It has used the ARToolkit's version revised by Henrysson as tracking tools and OpenGL ES (Open Graphics Library for Embedded System) as render engine. It has allowed two users to collaborate. The system also provided audio feedback when the ball touches the racket.

The sections that follow expand our system overview.

3 SYSTEM OVERVIEW

3.1 Hardware Setup

The main goal of our application is to propose a new architecture for a multiplatform hardware device augmented reality. In this work, we have used three kinds' hardware platforms:

- A notebook Sony Vaio, dual core 1,66 GHz, which has 2Go of RAM and 160 Go hard disk, with a USB Logitech chat webcam 640*480 colours resolution, and the operating system Windows Vista;
- A Samsung Q1 ultra mobile 800Mhz, which has 128Mhz of RAM and 60 Go hard disk, with the operating system Windows XP mobile, and a USB Logitech chat webcam 640*480 colours resolution;
- And a Pocket PC Dell Axim X51v 624Mhz Intel xscale processor, which has 480x640 32 bits display and 512 Mo RAM, with the Windows mobile 5.0 operating system and a IEEE 802.11b wireless network. We have extended it a SD camera 240*320 colours resolution, attached via a SD jacket.

These hardware devices have their specificities and constraints (memory space, disk space …).

3.2 Software Architecture

As indicated in section 3.1, each hardware device has his specificity. Therefore, we propose a design and implementation software hold account these specificities. The system is formed by two modules: visualization and interaction modules (see Figure 1). Both of them are presented below:

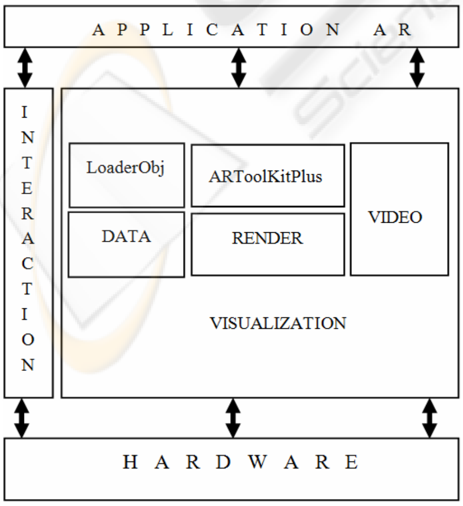


Figure 1: Software architecture.

The VISUALIZATION module is formed by five components: video capture, ARToolkitPlus, LoaderObj, Data and render.

ARToolkitPlus Daniel, W., 2001 framework is used for tracking a fudicial marker's position in a video image with respect to the camera position. It works by identifying markers in a video stream and calculate the orientation and translation of the camera in a reference coordinate system centered at a marker image. It is derived from ARToolKit library and written in c++ language. We choose this library because it is an open source toolkit for optical tracking and it runs on multiplatform such as WIN32, WINCE, and Linux. It performs the following steps: 1) Turn captured images into binary images; 2) Search the binary image for black square regions; 3) For each detected square the pattern is captured and matched against templates; 4) Use the known square size and pattern orientation to calculate the orientation and translation of the camera; 5) Draw the virtual objects.

VIDEO component is responsible for video capture from a camera. This module is ensured by Microsoft Windows DirectShow, or by native software of a camera for the platform which not supported Microsoft Windows DirectShow.

RENDER is composed by OpenGL and OpenGL ES responsible of object render. The render engine used depends on the platform which runs the application. For OpenGL ES implementation, we use Hybrid Rasteroïd 3.1 Hybrid Graphics at which includes the 1.1 OpenGL ES version. This last one uses EGL to access to the native configuration of lightweight device. OpenGL ES is essentially a state machine for a graphics pipeline, and EGL is an outside layer used to keep track of this group of graphic's state and maintain framebuffers and another rendering surfaces. EGL ensures also the controls and provides access to the device display and the possible rendering configurations of the device.

DATA is a component formed by two kinds of text files: one for stored the virtual workspace's information and another's one for virtual objects databases. Each virtual workspace is connected to one virtual object databases. However, virtual objects databases can be used by several virtual workspaces. Our Virtual objects are in obj file format generated by the modeler 3D Blender. It is the free open source 3D content creation suite, available for all major operating systems under the GNU General Public License. The work plan is constituted by six specific fiducial multimarkers of

ARToolkitPlus (see Figure 2) where two of them are dedicated for virtual menu (see Figure 3).

LoaderObj is our own loader library for obj file format which can run on multiplatform hardware. It is developed with c++ language and used OpenGL and OpenGL ES as renders engines. The loader detects on fly the current platform and uses a render engine corresponding of this.

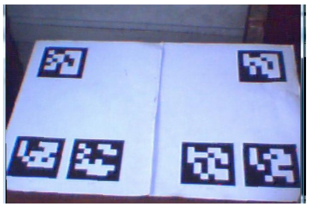


Figure 2: Work plan.

The INTERACTION module manages the interaction between the system and the users. For a personal computer device, we have used mouse, space mouse and keyboard input as interaction methods and stylus and buttons input are used for the lightweight mobiles devices as PDA. These interactions methods are chosen because they exist in several hardware devices. To move a virtual object in the virtual workspace, we have inspired the metaphor of dragging a virtual object on the surface and put it on one fiducial marker chosen by user. In this works, a fiducial marker is associated with dynamic or static ways for one virtual object depending of user's choose and the current workspace.

3.2.1 Virtual Menu

As described section 3.2, two of the six fiducials multimarkers of ARToolkitPus are dedicated for our virtual menu (see Figure 3).

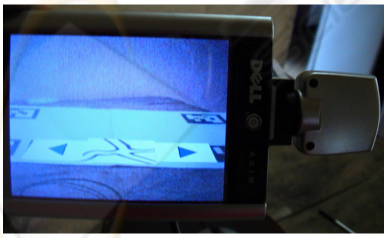


Figure 3: Virtual menu.

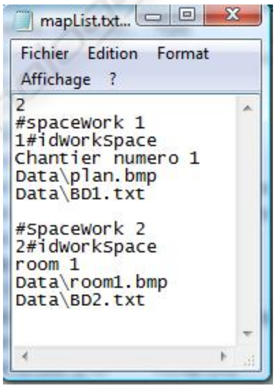
We use the visibility, the position of the four corners of fiducial marker and stylus or mouse positions in screen coordinate system to detect what virtual button is cliqued. A virtual object indicates the kind of button is superimposed over the fiducials markers when they are visible (see Figure 3). Virtual furniture appears between the virtual buttons when one of those is visible.

4 APPLICATION

We have tested our system for a virtual urban planning and a virtual interior design. User can change workspace suited to his need. In fact, it can be applied to various domains.

In the first application example, the idea is to augment our work plan by an image of urban planning which formed our virtual workspace and user can choose and put related virtual furniture. In the second application we use an image of virtual floor to augment the work plan.

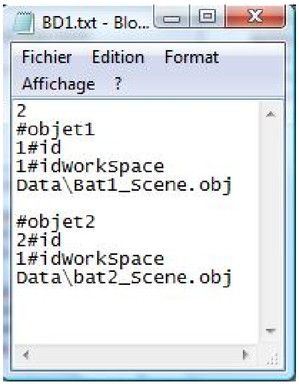
As indicated in section 3.2, the text file maplist stores the information about exist virtual workspace. It is illustrated in Figure 4. The first line corresponds to the workspace's number. Workspace information is composed by five fields: a Workspace's key, a label, a virtual workspace's file name and a virtual objects database's file name.



```
2
#spaceWork 1
1#idWorkSpace
Chantier numero 1
Data\plan.bmp
Data\BD1.txt

#SpaceWork 2
2#idWorkSpace
room 1
Data\room1.bmp
Data\BD2.txt
```

Figure 4: Maplist file.



```
2
#objet1
1#id
1#idWorkSpace
Data\Bat1_Scene.obj

#objet2
2#id
1#idWorkSpace
Data\bat2_Scene.obj
```

Figure 5: Sample of database file.

As for the database file (see Figure 5), it stores the information concerning the virtual objects that the user can manipulate. The first line is the number of the virtual objects in the current databases; the virtual objects databases have three fields: an object's key, a key of workspace (foreign key) and a file name of the virtual object.

Our virtual workspace is divided in two parts: the virtual menu area and the virtual workspace itself. The virtual menu is inspired by the metaphor of forward and next buttons (see Figure 6). The user can scroll these virtual furniture when he clicks to one of virtual buttons.

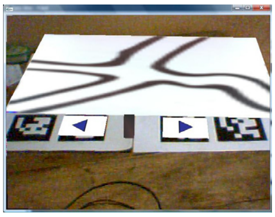


Figure 6: Virtual workspace.

Depending on the hardware configuration, the user has different ways to change his virtual workspace. He can use pop up menu or enter a key when he works with desktop computer, and, he can use the start button and the stylus for lightweight device. It could be underlined that these two approaches could be used with the Samsung Q1 ultra mobile. This workspace modification consists of two steps. First, the user must activate the procedure of change so that the virtual furniture in the virtual menu is substituted by the virtual workspace. After that, he can select new virtual workspace in virtual menu. Once he made his choose, the virtual furniture related for this new workspace appears in the virtual menu. This change could be seen through the hardware display device (see Figure 7).

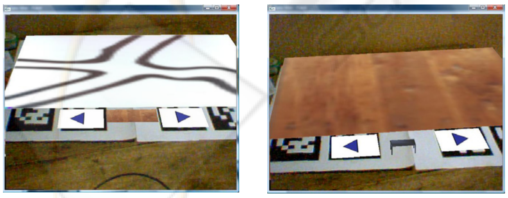


Figure 7: Change of workspace.

For adding virtual furniture in the virtual workspace, the user have to, first, pick and choose it in the virtual menu using mouse or stylus input, and then, drag it on the surface to place it on a fiducial marker (single marker) in the virtual workspace.

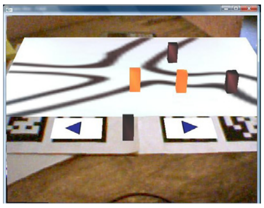


Figure 8: Virtual urban planning arranged.

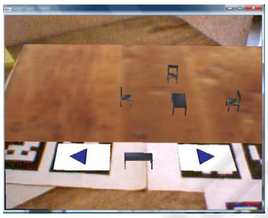


Figure 9: Virtual room arranged.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we describe a new architecture for an augmented reality multiplatform on table system oriented components. Our work is different from previous AR on table system in several important ways. Unlike other AR table top system, our AR system can run in multiplatform hardware device and work in dynamic virtual workspace. The change of workspace is done in dynamic way according to user needs. Thus, it could be applied to various domains. Our system also maintains virtual objects relationships with virtual workspace.

In this system, user can use mouse, space mouse, keyboard, button and stylus input as interactions methods. We used metaphor of forward and next button for the virtual menu in order to firstly, minimize the number of fudicial marker used for the virtual menu and secondly to allow user scrolling a virtual object in current databases.

In the future, we are going to make collaborative setup in order to establish one or more local or remote participants within the meeting setup via networking technologies like services discoveries, synchronization of data. We plan also to review our object loader in order to have greatest performance of our system.

REFERENCES

Aish, F., Broll, W., Stoerring, M., Fatah, A., Mottram, C., 2004. Arthur - an augmented reality collaborative design system. In *1st European Conference on Visual Media Production (CVMP 2004),* p. 277-287.

Andreas B., Tobias H., Steven F., Blair M., Clifford B., 1999. Enveloping Users and Computers in a Collaborative 3D Augmented Reality. In *Proc. of International Workshop on Augmented Reality (IWAR'99)*, San Francisco, CA, USA, p. 35-44.

Anthony, W., Steven, F., Blair, M.,William, M., Theodore, K., 1996. Augmented reality in architectural construction, inspection and renovation. In *Proceedings of ASCEThird Congress on Computing in Civil Engineering*, Anaheim, CA, p. 913-919.

Billinghurst, M., Kato, H., Weghorst, S., Furness, T. A., 1999. A Mixed Reality 3D Conferencing Application. *TechnicalReport R-99-1 Seattle: Human Interface Technology Laboratory, University of Washington.*

Daniel, W., 2001. ARToolKitPlus, from http://studierstube.icg.tu-graz.ac.at/handheld_ar/ artoolkitplus.php.

Daniel, W., Thomas, P., Florian, L., Dieter, S., 2005. Towards Massively Multi-User Augmented Reality on Handheld Devices. In *Proceedings of the Third International Conference on Pervasive Computing (Pervasive 2005)*, Munich, Germany.

Daniel, W., Schmalstieg D., 2005. A handheld augmented reality museum guide. In *IADIS Mobile Learning 2005*.

Hannes, K., 2003. Collaborative Augmented Reality in Education. *Keynote Speech at Imagina Conference*.

Henrysson, A., Billinghurst, M., Ollila, M., 2006. AR Tennis. In *Proc. 33d International Conference and Exhibition on Computer Graphics and Interactive Technologies (SIGGRAPH 2006)*.

Höllerer, T., Feiner, S., Terauchi, T., Rashid, G., and Hallaway, D., 1999. Exploring MARS: Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System. *Computers and Graphics, 23(6), Elsevier Publishers*, p. 779-785.

Hybrid Graphics at, from http://www.hybrid.fi.

Irawati, S., Green, S., Billinghurst, M., Duenser, A., Heedong, K., 2006. "Move the couch where? ": developing an augmented reality multimodal interface. In *IEEE and ACM International Symposium on Mixed and Augmented Reality,* p. 183-186.

Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K., Tachibana, K., 2000.Virtual Object Manipulation on a Table-Top AR Environment. In *Proceedings of the International Symposium on Augmented Reality (ISAR 2000)*, p. 111-119.

Mark, A., David, J., Daniel, C., Adam, H., 2003. AR. Phone: Accessible Augmented Reality in the Intelligent. Environment. In *Proc. of OZCHI2003*.

Moligev, D., Kiyokawa, K., Billinghurst, M., Pair, J., 2002. AR Pad: an interface for face-to-face AR collaboration. In *proceedings of Conference on Human Factors in Computing Systems CHI '02*, Minneapolis, Minnesota, USA Pages: 654 – 655.

Mooser, J., Lu, W., Suya, Y., Neumann, U., 2007. An Augmented Reality Interface for Mobile Information Retrieval. In *Proceedings of IEEE international Conference on Multimedia and Expo*, p. 2226-2229.

Raphaël G., Jean-Dominique G., 2002. MARE: multiuser augmented reality environment on table setup. In *Proceedings of the International Conference on Computer Graphics and Interactive Techniques,* ACM SIGGRAPH 2002, p. 213 – 213.

Reitmayr, G., Schmalstieg, D., 2001: An open software architecture for virtual reality interaction. In*: VRST '01: Proceedings of the ACM symposium on Virtual realitysoftware and technology*, Banff, Alberta, Canada, ACM Press, p. 47–54.

Schmalstieg, D., Fuhrmann, A., Hesina, G., Szalav´ari, Z., Encarna, c., L.M., Gervautz, M., Purgathofer, W., 2002: The studierstube augmented reality project. Presence:Teleoperators and Virtual Environments 11, p. 33–54.

Zsolt, S., Erik, E., Michael, G., 1998. Collaborative Gaming in Augmented Reality. In *Proceedings of VRST'98*, Taipei, Taiwan, p. 195-204.