AUGMENTED REALITY ENVIRONMENT USING A WEB BROWSER

Content Presentation with a Two-Layer Display

Kikuo Asai
National Institute of Multimedia Education, 2-12 Wakaba, Mihama-ku, Chiba, Japan

Hideaki Kobayashi
The Graduate University for Advanced Studies, Shonan Village, Hayama, Kanagawa, Japan

Keywords: Augmented reality, Web browser, two-layer display, content presentation, and character recognition.

Abstract: We developed a prototype system to make an augmented reality (AR) environment using a Web browser. Although AR technology has the potential to be used in various applications, authoring/editing AR contents is a problem on the wide spread. Graphics expertise and knowledge of computer programming are necessary for creating contents for AR applications. We used a Web browser as a presentation tool so that users who had experience in creating Web contents could reuse the multimedia data as AR contents and modify the AR contents by themselves. A two-layer display was used in the system in order to superimpose virtual objects onto a real scene for creating an AR environment. Another problem is about identifying objects in the real scene. An open-source library, ARToolkit, is often used as an image-processing tool for detecting the position and orientation of markers as well as identifying them. However, the markers must be registered in the system in advance, and the registration becomes tedious work when many markers are used such as Japanese kanji characters. The optical character reader (OCR) middleware was implemented as a character-recognition function for Japanese character markers. This paper describes the system design and software architecture for constructing an AR environment using a Web browser display and the demonstration of the prototype system.

1 INTRODUCTION

Augmented reality (AR) technology enables us to enhance recognition of the real world by superimposing virtual objects onto a real scene. AR has the potential of creating a new environment that seamlessly connects a virtual space to the real world. The AR environment has great advantages such as spatial awareness (Biocca et al., 2003(over the real world and tangible interaction (Poupyrev et al., 2002(over a virtual space.

Various applications have been developed based on AR's potential. However, authoring/editing AR contents currently requires computer graphics expertise and knowledge of computer programming, which is one of the reasons that AR has not generated the killer application. Some toolkits for authoring/editing AR contents have been developed such as AMIRE (Grimm et al., 2002(and DART (MacIntyre et al., 2004(that allow the user to edit the contents without being familiarized with computer graphics and programming.

Unfortunately, these toolkits are not well suited for creating simple AR contents because using the toolkits involves several complicated tasks in the production process and additional tools for which users must learn the basic skills. Our idea involves using a Web browser as a presentation tool. The browser supports various data formats that enable us to reuse multimedia data. A Web browser may be obtained for free, and the users who create AR contents probably have some experience with a markup language such as HTML, VRML, or XML.

In addition to the data formats it supports, another advantage of using a Web browser is that plug-in software can be installed for additional functions for extended data formats. However, using a Web browser causes a problem when

Asai K. and Kobayashi H. (2006).
AUGMENTED REALITY ENVIRONMENT USING A WEB BROWSER - Content Presentation with a Two-Layer Display.
In *Proceedings of WEBIST 2006 - Second International Conference on Web Information Systems and Technologies - Internet Technology / Web Interface and Applications*, pages 481-486
DOI: 10.5220/0001250604810486

superimposing virtual objects onto a real scene, so we used a two-layer display for presenting virtual objects over images in the real scene.

In marker-based tracking, the markers that are used for identifying an object and detecting its position and orientation in the scene have to be registered in advance. Registering many markers would be tedious work, especially when using language characters as markers. We used an optical character reader (OCR) middleware for recognizing Japanese character markers.

Our goals are to develop a simple method for superimposing virtual objects over the real scenes and to enable users to create AR contents composed of multimedia data of various formats. This paper describes the design and architecture of a prototype that uses a two-layer display for making an AR environment with a Web browser.

2 RELATED WORK

ARToolkit (Kato et al., 2000(is a set of C/C++ open-source libraries, which has widely been used for coding programs for AR applications. The primary advantage of ARToolkit is to work with a single camera operating under visible lighting conditions. Programming skills are required for coding the programs but are not required for using the runtime samples for attaching VRML models to markers. ARToolkit is distributed with sample programs that show users how to attach VRML models to markers.

AMIRE is a set of graphical authoring/editing tools, which has a component-based framework focused on specific AR domains. Although AMIRE does not involve coding programs directly, a visual language interface has to be treated for integrating components in an object-oriented structure. DART is also an authoring/editing tool built on Macromedia Director. Director is an object-oriented environment for creating 2-D and 3-D animations. A basic knowledge of using Director is required for using DART for developing AR applications.

These authoring/editing tools are not necessary for creating simple applications, and the runtime samples of ARToolkit would be enough for superimposing virtual objects over a real scene. However, the ARToolkit samples are not compatible with various multimedia data formats and registrations of many character markers. In our system, a Web browser enables us to reuse multimedia data of various formats, and OCR may solve the registration problem.

Multi-layer displays that were originally developed for CAD simultaneously display drafts and 3-D structures on the same monitor. While NTT Cyber Space Laboratories used a multi-layer display for developing a stereoscopic display, depth-fused 3-D (DFD) (Suyama et al., 2001(, and the display is currently on the market. We used a two-layer display for constructing an AR environment.

3 SYSTEM

Figure 1 shows an example of a superimposed presentation on a two-layer display. The English words are superimposed over the Japanese characters. The user holds the two character markers, and the two translated words are presented in the different windows. The recognized hiragana markers are enclosed by the pink rectangles in the camera-image window, and the two English words are set at the corresponding locations in the presentation window. When the user moves the markers, the English words track the markers with the position and orientation.

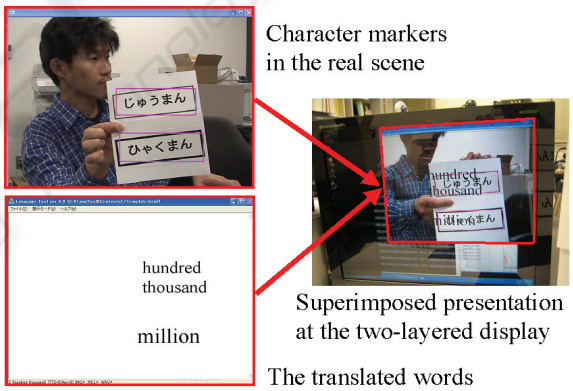


Figure 1: An example of superimposed presentation on two-layer display.

3.1 Design

We here describe the design and functions of our prototype system for making an AR environment. The system detects square frames in a video image and checks whether the characters in the square frames correspond to the words that were registered in advance. Based on the character markers, the functions display texts, images, and 3-D models and play sound, or generate a synthesized voice.

A design that incorporates a Web browser, a two-layer display, and an OCR makes our system unique.

1) Web browser—A Web browser was used in the prototype as a presentation tool. Using a Web browser has some distinct advantages. Web browsers can support various multimedia data formats and plug-in tools. Users have had problems using AR technology for authoring/editing contents because programmers and graphic experts usually develop AR applications. Using a Web browser improves the flexibility of authoring/editing and reusing the existing contents, which makes creating simple AR applications possible for novice users. Moreover, the Web browser supports plug-in tools and extended data formats.

2) Two-layer display—Some people might claim that using a Web browser is not an AR-based system because the virtual objects may not be spatially registered on the real scene. However, the system was designed as follows in order to keep the AR features. The first is detecting the position and orientation of the identified markers and reflecting them in the presentation on the browser window. The other is implementing a two-layer display in the system that enables us to present virtual objects that are superimposed on the real scene. Although spatial registration is not actually achieved, the virtual objects track the markers at the approximate positions of the markers in the real scene.

3) Character recognition using OCR—Most AR applications use markers for identifying objects and detecting their position and orientation in real scenes. However, users have to register each marker in the system in advance, and this causes difficulties if more than 100 markers are involved. OCR is thus useful for recognizing letters and characters and for detecting words as character markers. Although a specific version of the OCR software needs to be installed in order to allow for appropriate character recognition (because OCR depends on the language being used), using OCR greatly expands the number of words that can be used in AR systems.

3.2 Configuration

Figure 2 shows the configuration of the prototype system. The arrow indicates direction of the flow of the processing data. A camera captures video images including the character markers held by the user, and the video is fed into a PC.

First, the recognition function of the PC detects the markers and the characters by matching them to the marker data or by finding words with the OCR. We created specific character markers for a command in order to control the presentation. For example, the user can change the format of the presentation from text to image or the size of the text font, by presenting the control command marker.

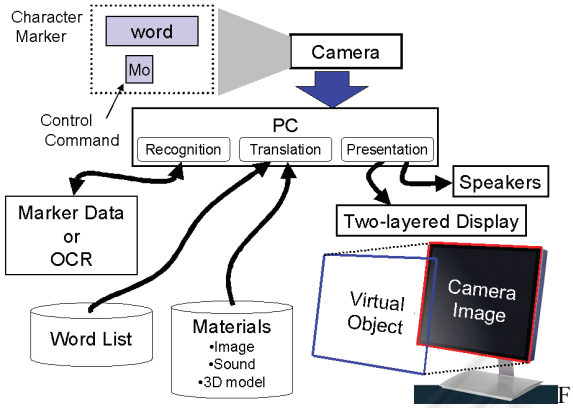


Figure 2: Configuration of prototype system.

Table 1: Beginning of word list.

ひらがな	カタカナ	漢字	英語	タイ語
あい	アイ	愛	love	ความรัก
あいがんする	アイガンスル	哀願する	entreaty	วอน
あいこくしん	アイコクシン	愛国心	patriotism	ใจรักชาติ
あいことば	アイコトバ	合い言葉	password	รหัสผ่าน
あいさつ	アイサツ	挨拶	greeting	การทักทาย
あいしょう	アイショウ	愛称	nickname	ชื่อเล่น
あいじょう	アイジョウ	愛情	affection	ความรัก
あいしょう	アイショウ	相性	affinity	ชะตากัน
あいず	アイズ	合図	signal	สัญญาณ
あいすくりーむ	アイスクリーム	*	ice cream	ไอสครีม
あいすこーひー	アイスコーヒー	*	iced coffee	กาแฟเย็น

Hiragana Katakana Kanji — Japanese; English; Thai

The words that were recognized with the OCR are then counted as possible matches to the word shown by the user, and are looked up in the word list. If the word matches a word in the word list, it is translated into the designated format such as the text of the other language, an image, a 3-D model, or sound, which has been prepared for the presentation in advance. Table 1 lists a sample from the word list.

The user views two windows that are superimposed at the two-layer display. One window contains the camera image and the other the contents. The camera-image window shows the input video-image including the character markers. The content window presents information such as the translation texts, images, and 3-D models that are related to the recognized words.

The prototype system does not have a video transmission function. When the system is used in telecommunication, video images have to be transmitted to remote sites separately through some network such as the Internet or satellite communications. A two-layer display is capable of superimposing the camera image and the contents, and presents them simultaneously. The virtual objects are spatially registered to the camera images of the real scene based on the positions and

orientations of the character markers. Instead of texts, images, or 3-D graphics, sounds are played using the sound files in the word list. When the user selects the voice mode, a voice synthesizer reads out loud for the recognized word.

3.3 Software Architecture

Figure 3 shows the software architecture of the prototype system. The software is mainly composed of image processing and presentation parts. The two parts have a server-client relationship, and they exchange data using socket communications. We used UDP/IP to support multiple clients that transmit and receive control signals with a server on the network. The multiple clients can share the information based on the recognized word at the server. This software architecture allows users to apply the system to support for telecommunication. For example, a presenter makes a speech using keywords while the video images are processed at the server. The processed information is then multicast to remote sites where the audiences see the video with the keywords translated to their mother tongues.

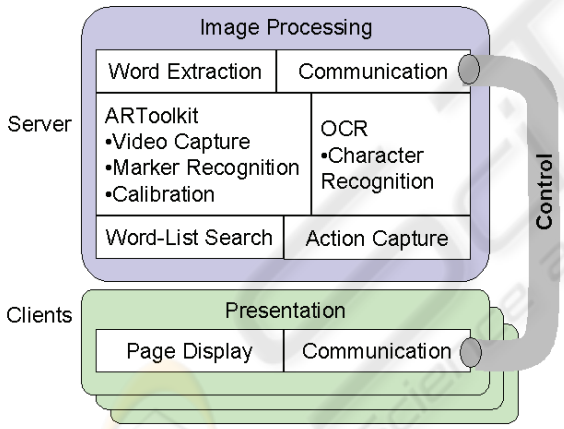


Figure 3: Software architecture.

Character Recognition

The way an image is processed depends on the number of characters on the character marker. First, the OCR tries to recognize some characters in order to detect words in the character marker. If nothing or only one character is found, the process is passed to ARToolkit. If the character matches another one in the pre-registered marker data, that character is then used as a control command. Otherwise, the message "no word" appears on the display. If the OCR detects some words, they are checked as possible matches against the words in the word list.

We used ARToolkit for identifying a control command because the number of control commands is currently around 30, and registering the markers is not tedious work. The ARToolkit was also used for detecting square frames—determining their position and orientation.

We used the OCR middleware tool "Yonde!! KoKo" (A.I. Soft) for the Japanese character recognition. OCR is well suited for extracting the characters of specific languages even though it does not recognize the script or special symbols in minor languages. The OCR tool recognizes the JIS first-level kanji and hiragana, katakana, the English alphabet, and Roman numerals. These characters do not have to be registered in advance.

Word Detection

An original word list was created for conducting the language translation because using a word list is efficient on word detection. We did not use a commercial translator. The translation is done by referring to the compact word list from a specific field of interest to the user. In the prototype system, Hiragana, Katakana, and Kanji were designated as the source characters, and everyday English or Thai words were the destination words.

The word detection has not been perfect, and incorrect results were sometimes obtained. The main reason for this was that the characters derived from the OCR were not always correct, depending on the lighting condition and the movements of the character marker. We observed that a statistically significant number of characters were recognized incorrectly. For example, the small characters of Japanese Hiragana were often mistaken for the normal-size Hiragana characters.

The word detection was thus conducted using the following correction process. The characters that were recognized by the OCR were first looked up in the word list. If the group of characters did not match any words in the word list, one character at a time in the group was substituted with a candidate character, and the corrected group was then checked against the word list. This process was repeated until a match was found.

Reducing the number of times the page is refreshed

The contents were presented in a layout that was based on the position of the markers held by the user. An HTML file was created to place the text, images, or 3-D models at the marker positions in the real scene. A page is renewed by recreating and reloading the new file. However, refreshing the page frequently destabilizes the presentation, and should thus be kept to a minimum.

The change in position of the virtual objects was controlled smoothly without renewing the page by

using a JavaScript function—a new file was created and reloaded in the Web browser when the layout of the virtual objects changed with the rotation and zoom of the markers.

This method of control also helps reduce traffic of signals multicasting between the server and the clients. We set three levels of signal transmission. Traffic was not generated when the detected words were identical to the previous ones with the same layout. Only the layout information was transferred when the detected words were identical to the previous ones but had a different layout. When the detected words and their layout changed, their detected words and the layout information were multicast.

3.4 Implementation

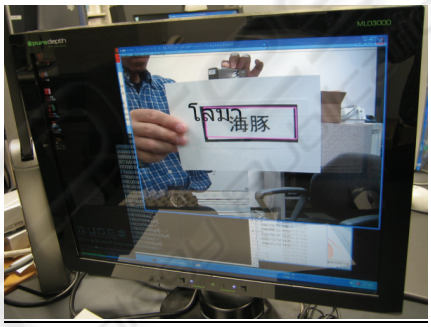
Figure 4 demonstrates snapshots of each presentation with (a) text, (b) an image, and (c) a 3-D model. The system can control the presentation format of text, images, and 3-D models by inserting a control command marker. When the control marker, “Im” or “Mo,” reaches the character marker, the corresponding image or the 3-D model is presented at the marker’s location. Only text appears when there is no control marker. We used a VRML plug-in tool, Cortona (Cortona(, for viewing the 3-D models in the Web browser.

The HTML format is advantageous for creating AR contents thus making authoring/editing the contents and reusing the multimedia data easier. Using a Web browser as a presentation tool also enables us to install plug-in software, even though the system does not offer much control flexibility of the plug-in software. Figure 5 demonstrates a snapshot of presenting protein data from the Protein Data Bank, in which a plug-in tool, Chime (Chime(, was implemented. The snapshot represents the 3-D structure of a virus related to avian influenza. The plug-in software increases the number of presentation formats beyond that which is normally supported by default in the Web browser. Although sounds cannot be demonstrated in this paper, when the user selects the sound files as the destination, they are played based on the sort of the markers.

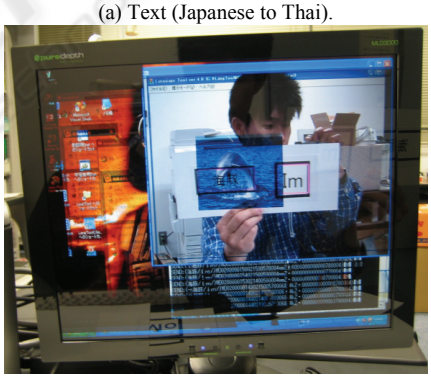
The prototype system was implemented on two desktop PCs, each with a 2.4-GHz Pentium IV processor for the image processing, and a 1-GHz Pentium III processor for the presentation. A DV camera (DCR-HC1000, SONY) was connected to the image-processing PC through an IEEE 1394 cable. The PureDepth MLD^{TM} 3000 that was used as a two-layer display had specifications of a 17”-diagonal size monitor and 1280x1024x2 resolution.

Although the frame rate was roughly 20 frames per second, a delay of around one second was observed between the time that the character marker appeared and the time the text displayed. When the markers were quickly shifted in the camera image, unstable recognition was observed with the OCR.

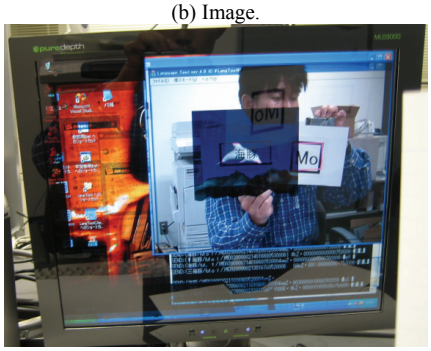
We also checked accuracy of the word identification, measuring a recognition rate of the character markers under a static condition. 8 markers of the Japanese Hiragana and Kanji characters were prepared as samples for the evaluation test. The simple camera (Qcam Pro 4000, Logicool) with 640 x 480 pixels was used for capturing images.



(a) Text (Japanese to Thai).



(b) Image.



(c) 3-D model.

Figure 4: Snapshots of content presentations with (a) text, (b) an image, and (c) a 3-D model.

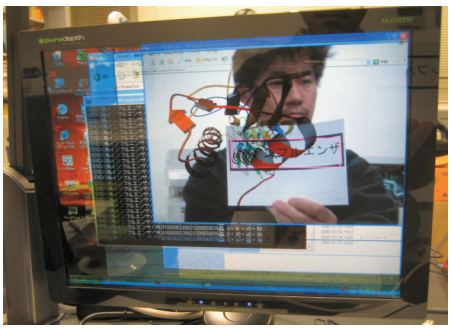


Figure 5: Presentation using a plug-in tool for the Protein Database.

Figure 6 shows the average rate of the correct recognition for the sample characters. The sort of marks corresponds to difference of the slant for the character markers. “30” means that the character marker was set at direction of 30 deg. from the line of sight of the camera. The recognition rate decreases with distance of the character markers from the camera and slant of the markers from the line of sight. When markers are presented vertically onto the camera, the stable recognition is obtained at a distance from 20 to 40 cm.

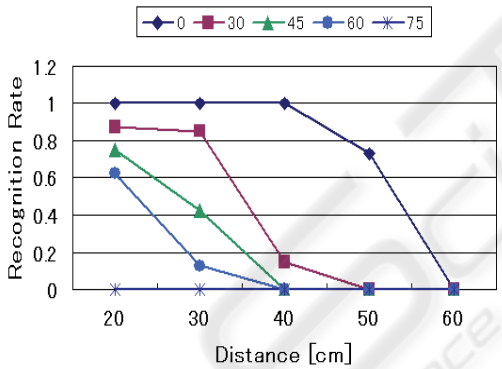


Figure 6: Average rate of the correct recognition.

4 SUMMARY

We developed a prototype system that presents contents with a two-layer display in order to make an AR environment using a Web browser. Using a Web browser as a presentation tool may improve the authoring/editing of AR contents and the reusability of multimedia data. The system can be used for supporting telecommunications by presenting additional information related to text during a videoconference.

We used a word list to identify the words shown by the user. For some words, however, the system performance decreased due to the time required to search the words. An acceleration function is required for improving the efficiency of retrieving information in the system. Our future plans include improving the accuracy of the character recognition using the OCR.

ACKNOWLEDGMENTS

This research was partially supported by a grant-in-aid for scientific research (17300283) from the JSPS. “Yonde!! KoKo” is a trademark of the A. I. Soft Co., Ltd. Yoshihiko Masui provided the Thai words in the word list.

REFERENCES

Biocca, F., Rolland, J., Plantegenest, G., Reddy, C., Harms, C., et. al. Approaches to the design and measurement of social and information awareness in augmented reality systems. *Proc. Human-Computer Interaction International: Theory and Practice,* pp.844-848, 2003.

Chime, a PDB plug-in tool for a Web browser, http://www.mdli.com/index.jsp

Cortona, a VRML plug-in tool for a Web browser, http://www.parallelgraphics.com/products/cortona/

Grimm, P., Haller, M., Paelke, V., Reinhold, S., Reinmann, C., and Zauner, J. AMIRE – authoring mixed reality. *Proc. IEEE International Augmented Reality Toolkit Workshop*, 2002.

Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K., and Tachibana, K. Virtual object manipulation on a table-top AR environment. *Proc International Symposium on Augmented Reality,* pp.111-119, 2000.

MacIntyre, B., Gandy, M., Dow, S., Bolter, J. D. DART: a toolkit for rapid design exploration of augmented reality experiences. *Proc. International Conference on User Interface Software and Technology*, 2004.

Poupyrev, I., Tan, D. S., Billinghurst, M., Kato, H., Regenbrecht, H., and Tetsutani, N. Developing a generic augmented reality interface. *Computer,* vol.35, pp.44-50, 2002.

Suyama, S., Takada, H., Uehara, K., Sakai, S., and Ohtsuka, S. A new method for protruding apparent 3D images in the DFD (Depth-Fused 3D) display. *SID 01 Digest,* vol.54.1, pp.1300-1303, 2001.