An ARToolkit-based Application System Integrating Augmented Reality and 1-D Barcodes

Jong-Chih Chien#1, Li-Chang Liu#2, Jia-Yang Chen#3, Kun-Ming Yu#4, Chin-Hsien Lin#5

[1]*Dept. of Computer Science and Information Engineering, Kainan University*
Taoyuan County, Taiwan
[1]jcchien@mail.knu.edu.tw
[2,3,5]*Simpleact Incorporated*
Taipei, Taiwan
[2]lcliu2000@simpleact.com.tw, [3]yang@simpleact.com.tw
[4] *Dept. of Computer Science and Information Engineering, Chung Hua University*
Hsinchu, Taiwan
[4]yu@chu.edu.tw

***Abstract*— This paper proposes an ARToolkit-based system integrating augmented reality and 1-D barcodes by associating each 1-D barcode with a corresponding AR marker and 3-D graphics data. After scanning a barcode and retrieving the AR marker and 3-D graphics files associated to the numbers encoded in the barcode, the system will search for the symbol of the AR marker near the 1-D barcode for the purpose of displaying the graphics data on the AR marker. In this way, a 1-D barcode is not just a tool for inventory management, but by combining it with different AR markers, this system enables it for commercial advertisement of merchandises such as toy figurines which require small amount of graphics data.**

***Keywords*— ARToolkit, Augmented Reality, 1-D Barcode**

1. INTRODUCTION

In recent years, due to leaps in the progress in hardware improvements of computers and smart mobile phones, applications related to Augmented Reality (AR) are starting to show up in daily lives. Based on their usages, AR applications can be divided into the Marker AR category or the Marker-less AR category [1,2,3]. The main method of Marker AR applications in determining the display positions of the 3-D graphics is by applying image processing methods in detecting the marker's coordinates. Alternatively, the Marker-less AR applications use data provided by GPS and digital compass which use wireless signals to help resolve the display positions of the 3-D graphics.

Amongst all the softwares developed for AR applications, ARToolkit [4,5,6] is one of the most widely used due to its early entry into the market and provides GPL licensing for non-commercial uses. But the GPL-licensed source code provided by ARToolkit is limited in that it requires the marker symbol and its associated 3-D graphic file hard-coded into the program, so it is not flexible enough to provide a single program capable of displaying different 3-D graphics using a single marker while allowing the end users to update the graphics at any time. In order to improve the shortcomings mentioned above and allowing end users to view different 3D graphics within the same application while making it easier for manufacturer to use their existing products to provide additional entertainment values, this paper proposes an AR system that combined barcodes and ARToolkit which can dynamically retrieves and displays 3-D graphics. It enabled end users to scan and decode the barcodes on merchandise goods, then to download the AR marker symbols and the associated graphics information based on the decoded number. The downloaded information are then used to help located the AR marker symbol placed near the 1-D (or 2-D) barcode and use its coordinates for displaying the 3-D graphics. In this way, end users will be able to see different 3D graphics on different merchandises, and manufacturers who use tiny promotional stickers to promote their products can well use this system by converting their stickers, and use the pictures on their stickers as markers to enable AR views of 3-D models of their promotional products such as toy figurines which are low in complexity and requires small amount of graphics data so it would be possible to render them on embedded systems.

2. BARCODE-BASED AR SYSTEM

Our current implementation of the barcode-based AR system includes a server of graphics data and a PC-based AR application software, as shown in Figure 1. The graphics server is used mainly for storage and will respond to the AR application's requests of marker files and 3D graphics files that correspond to specific barcodes. The AR application software installs on a PC computer, using the computer's webcam to detect and resolve a one-dimensional barcode, then download the marker and its associated 3D graphics files from the graphics server, and finally display the 3D graphics on top of the actual marker.

The detailed process flow of the AR application is shown in Figure 2, which shows the main steps as follows: (1) capture an image from the webcam and determine the existence of a one-dimensional barcode within the image; (2) decode the one-dimensional barcode; (3) send the decoded barcode number as an index to the graphics server and request the corresponding AR marker and 3D graphics files; (4) the AR application receive the delivered marker and 3D graphics file from the server; (5) search within the webcam image for the printed AR marker; and (6) calculate the transformation matrices according to coordinates of the AR marker and then display the 3D graphics on the printed AR marker using the transformation matrices.

In the stage of detection of one-dimensional barcode within a webcam-captured image, the fact that 1-D barcodes are straight vertical lines is used as feature for detection. In a given region, the gradient differences are calculated and summed for each pixel at 90 , 0, 45, and 135 degrees orientation and differences between the sums are greater than a certain threshold value, then this region is judged to be part of a one-dimensional bar code. The algorithm is as follows:

For a given nxn region extracted from the center of the screen, its values are first converted to binary where the value of black pixels are set to '1', and we denote the binary value for any pixel at (x,y) position as bin(x,y). Then for each pixel at $(x, y) \in n \times n$ position, the counters C0, C45, C90, and C135, are used for 0, 45, 90 and 135 degrees, respectively. These counts are calculated as follows:

if $\sum_{n=-1}^{1} \left| bin(x-n, y) - 1 \right|$ = 0, increment C0,

if $\sum_{n=-1}^{1} \left| bin(x-n, y+n) - 1 \right|$ = 0, increment C45,

if $\sum_{n=-1}^{1} \left| bin(x, y-n) - 1 \right|$ = 0, increment C90, and

if $\sum_{n=-1}^{1} \left| bin(x-n, y-n) - 1 \right|$ = 0, increment C135

and when these counts are obtained for the entire region, we calculate the possibility that this region is part of a barcode in this way:

if |C90-C0| > |C135-C45| AND
||C90-C0|-|C135-C45|| > threshold2, set
region_in_barcode = TRUE
else set
region_in_barcode = FALSE

In addition using the AR marker's coordinates for viewing the 3D graphics, such as by rotating the marker to achieve rotation of the 3D model, or moving the marker closer to the camera lens for zooming effect, the AR applications also has additional controls for the purposes of increasing viewing angles and inserting special effects that are marker-independent, that is, these controls allow rotations, zooming, and other effects without moving the marker. These additional controls are achieved by calculating the combined spatial transformation matrix and apply matrix multiplication to the coordinates of each triangle of the 3D model before drawing the triangle surface on the screen, and that each value within the transformation matrix is stored in memory. In this way, the 3D graphics can be quickly reset to its original position in order to facilitate a variety of interactive special effects. The order the multiplying the transformation matrices is defined as rotation first, followed by zooming, and finally the translation transformation matrix is applied. These transformation matrices are defined as follows [7]:

Translation:

$$[x' \; y' \; z' \; 1] = [x \; y \; z \; 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix}$$

Zooming:

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

Rotation around the X-axis

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

Rotation around the Y-axis:

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and}$$

Rotation around the Z-axis:

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. EXPERIMENTAL RESULTS

The AR application was tested on a laptop computer and its specifications are: Acer Aspire6530, CPU is AMD Atholon64x2 dual core@2.0GHz, available RAM is 3GB, and the operating system is Microsoft Windows Vista SP2. The AR application code was developed in c/c++ language, and the graphics server was developed on a .Net platform using ASP.

The Hiro pattern that came with the ARToolkit was chosen as the marker for the experiment. An experiment was first performed using various different sizes of the chosen marker to find out if size of the marker would cause a problem in detection. The sizes used were 4.3cm, 3cm, 2.1cm, 1.6cm, 1.2cm, 0.9cm, and 0.7cm on each side, and the AR application was able successfully detect every marker, as shown in Figure 3. However, the experiment showed that if the desired effect is rotate by 3D model by rotating the marker, then better performance can be achieved if the size of the marker is larger than $1cm^2$. In this experiment simulating manufacturer's stickers, a 0.9cm version of the Hiro pattern is chosen as the marker in order to facilitate placement of a similar-sized marker in an unused area near a 1-D barcode on the merchandise.

The AR application can read a special 3D file with unusual specification; this 3D file is built by extracting from a standard 3DS file the number of triangles, vertex coordinates, material data, and the normal vectors then stored. In addition, the side length, width, and height of the 3D model's bounding box are also stored in this new file format in order to facilitate future access of this 3D model by other embedded systems and the drawing it using OpenGL-ES instructions. And prior to storing the values of vertex coordinates the center of the 3D model is first shifted to the origin in order to facilitate zooming and set the light source positions for drawing.

Figure 4 is the screen for when the AR application started to detect whether the captured image contained a one-dimensional barcode. Figure 5 shows that after a successful detection and decoding of a one-dimensional barcode, the marker associated with the barcode and the 3D data file were downloaded, then application searched for the location marker and identified it. Figure 6 shows that rotation of the 3D model was achieved by rotating the marker. While the end user is looking the 3D model, they can press the number keys to select the marker-independent interactive features, such as zooming as show in Figure 7, rotation about the X-axis, as shown in Figure 8, rotation about the Y-axis as shown in Figure 9, rotation about the Z-axis as shown in Figure 10 and Figure 11 shows for special effect of applying explosion on the model which decouples objects within the model. These controls can help users to view the 3D models in detail in from every angle and also create entertainment values.

4. CONCLUSIONS

The barcode-based AR system proposed in this paper was successful in using a one-dimensional barcode as an index to locate and download the marker and 3D graphics for display, and changed the way ARToolkit statically reads in 3D models into dynamically access 3D graphics files so that by simply updating the AR marker, the manufacturers can set the expiration on a certain marketing or advertising project, and not be dependent on ARToolkit's original method of using fixed marker and the 3D model. Future researches will explore the effect of different markers [8], such as using the one-dimensional barcode not just as an index but as an AR marker itself. Although this approach will introduce some problems in the display of the 3D graphics, but it can reduce the burden on the end users to print markers near the barcode so that even general merchandise can be used to experience AR-based advertising, and makes it easier for the manufactures to add values to their products.

ACKNOWLEDGMENT

Part of the result presented in this paper is from the results of the year 98 annual industry-academia joint research and development (JAID) plan "Two-dimensional Barcode for Interactive Advertising Technology Development Plan" (Plan No.: S09800226-005) supported by the Industrial Development Bureau of the Ministry of Economic Affairs of the Republic of China (Taiwan) .

REFERENCES

[1] “Enhance Everyday Living using Virtual Information and Interactive Elements - Application of Augmented Reality” http://www.find.org.tw/find/home.aspx?page=news&id=5638 。
[2] “Augmented Reality has Become the New Wave and Walmart is Actively Participating” http://www.find.org.tw/find/home.aspx?page=news&id=5645 。
[3] “Augmented Reality-enabled Mobile Phone Browser-Layar” http://www.find.org.tw/find/home.aspx?page=news&id=5647 。
[4] ARToolkit Download page http://www.hitl.washington.edu/artoolkit/download/。
[5] Meiguins, B.S., Carmo, R., Almeida, L., Goncalves, A.S., Pinheiro, S.C.V., Garcia, M., Godinho, P., “Multidimensional Information Visualization using Augmented Reality,” Proceedings of the 2006 ACM International Conference on Virtual Reality Continuum and its Applications, pp. 391-394, 2006.
[6] Hu, Q., Liu, T., Yao, Y., “An Easy System of Spatial Points Collection Based on ARToolkit,” Proceedings of 2009 WRI World Congress on Computer Science and Information Engineering, Vol. 1, pp. 582-586, 2009.
[7] Pulli, K., Aarnio, T., Miettinen, V., Roimela, K. and Vaarala, J., Mobile 3D Graphics with OpenGL ES and M3g, Morgan Kaufhann, 2008.
[8] Ababsa F., Mallem, M., “A Robust Circular Fiducial Detection Technique and Real-time 3D Camera Tracking,” Journal of Multimedia, Vol. 3, No. 4, pp. 34-41, 2008.

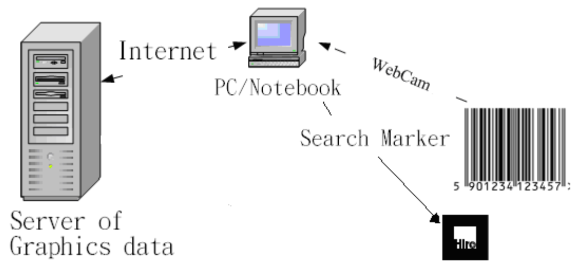

Figure 1. Structure of Barcode-based AR System

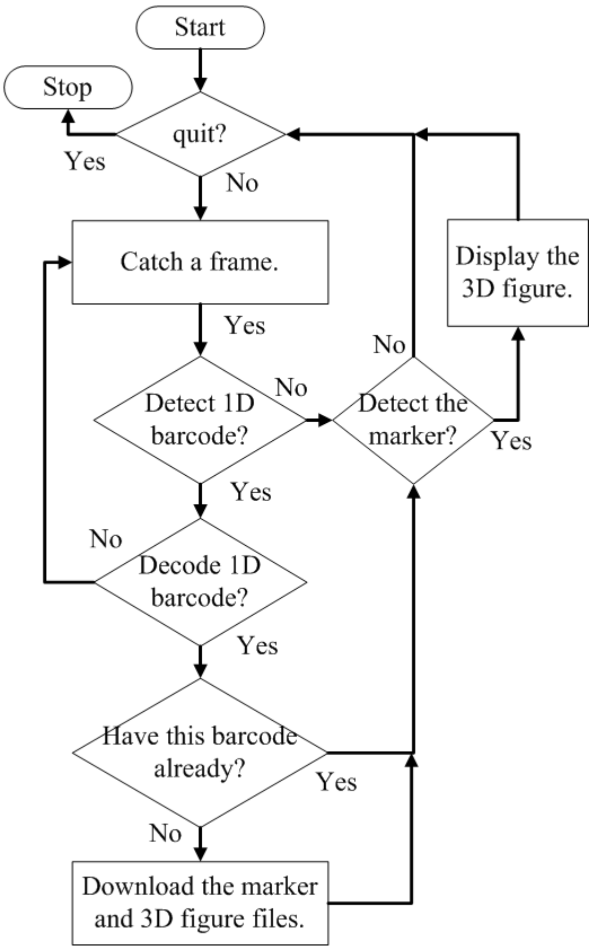

Figure 2. Process Flowchart of the AR Application Software

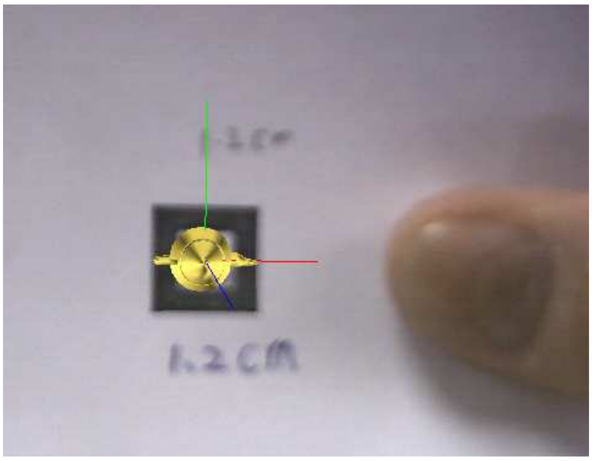

Figure 3. Detection on various sizes of marker

Figure 4. Operating the AR application software (Screen 1)

Figure 5. Operating the AR application software (Screen 2)

Figure 6. Operating the AR application software (Screen 3)

Figure 7. Screen showing the Zoom function of the AR application software

Figure 8. Screen showing the Rotation around the X-axis function of the AR application software

Figure 9. Screen showing the Rotation around the Y-axis function of the AR application software

Figure 10. Screen showing the Rotation around the Z-axis function of the AR application software

Figure 11. Screen showing the Explosion function of the AR application software