Implementations and Evaluations of Location-Based Virtual City System for Mobile Phones

Hiroyuki Tarumi
Kagawa University
& SpaceTag, Inc.
Takamatsu, Kagawa 761-0301 Japan
tarumi@eng.kagawa-u.ac.jp

Kazuya Matsubara
Graduate School of Engineering
Kagawa University
Takamatsu, Kagawa 761-0396 Japan

Masahiko Yano
Fujitsu Shikoku Systems, Ltd.
Takamatsu, Kagawa 760-0027 Japan

***Abstract*— A system platform for location-based entertainment applications of mobile phones is described. With GPS and electronic compasses, some of today's mobile phones have rich functions for such entertainment as role playing games, treasure hunting, sightseeing with virtual information, etc. Our system model can manage virtual worlds that can be overlaid onto the real world with regard to location (latitude and longitude). Each virtual world is a set of virtual architectural objects and virtual creatures. Virtual creatures are active objects that can interact with users. In this paper, browser-based and BREW-based systems for virtual worlds are described. BREW is an efficient software platform for mobile phones. It enables autonomous location sensing and automatic redrawing, which will solve most of the problems found in evaluation sessions of the browser-based version.**

Keywords*- *mobile entertainment, location-based services, BREW, virtual worlds

I. INTRODUCTION

With internet accessing methods, cameras, higher resolution displays as well as GPS receivers, today's mobile phones can be regarded as general purpose multimedia terminals. Entertainment is, of course, one of the most expected application domains of such multimedia terminals. Popularity, low-cost, and light weight are important advantages of phones, compared with other mobile terminals like PDA or laptop PC.

Our goal is to develop and deploy a system with which people have experience of virtual worlds with their mobile phones. We have developed and evaluated overlaid virtual systems for mobile phones. Here, by "overlaid virtual" we mean a mixed model of real and virtual worlds, where they are related to each other with regard to location. Each virtual world has a same geographical structure with the real world (Fig. 1 [1]). A user can select and visit one (or even more) virtual world with his/her mobile phone terminal.

A virtual world consists of virtual architectural objects and virtual creatures. Virtual architectural objects are static objects like buildings. Virtual creatures are dynamic objects that can move or interact with other objects or users. In other words, a virtual creature is an active agent that can react to stimuli from other objects and users. They can also exchange messages with other agents. Sometimes we call virtual creatures just as agents.

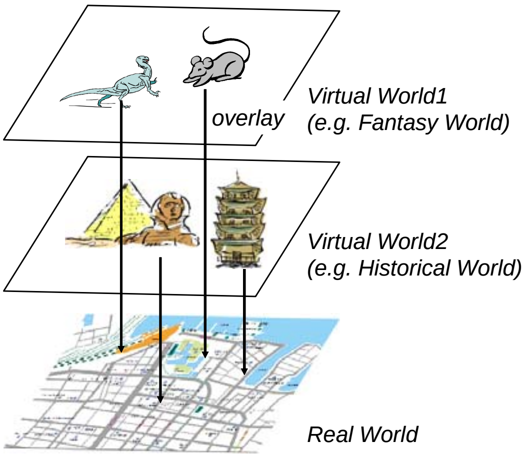

Figure 1. Overlaid Virtual Model

The virtual city system is an expansion of the SpaceTag system [1,2,3], which gives information objects to mobile terminals so that only limited users can see them. The access limitation is defined with regard to location and time. Each SpaceTag object has a specific geographical area where users can access the object. The virtual city system described in this paper is a descendant of the SpaceTag system, which allows 3D data as objects, while the original SpaceTag system only permitted text or image objects. On the virtual city system, a farther object appears to be smaller to users, but a closer object appears to be larger. If a user views the front face of an object from the north, she can see its back side from the south. A display of mobile terminal is like a window opened to the virtual world.

Another interesting aspect of the SpaceTag system is that it is bi-directional. Users can create new objects and put them into a virtual world. The virtual city system, of course, inherits the bi-directional feature. It allows entertainment application designers to design more exciting scenarios of application. For example, a game player would be able to create new creatures. Due to the limit of space, we do not describe detailed implementation of this aspect, in this paper.

Part of the research was supported by Kagawa Industry Support Foundation (Grant for Industry-Academia-Government Liaison for Research and Development) and the Okawa Foundation.

For entertainment applications of the virtual city system, we are supposing games like role-playing, treasure hunting, etc. For example, urban simulation can be implemented in a more realistic way. Role playing games would be implemented with virtual creatures moving around in a virtual city area. Activities of agents in the virtual world can be observed by human players in the real world. Multiple human players can join a game, where physical encounters and real (oral) communication between them may happen.

Supposed applications include sightseeing support systems. For example, we can give historical buildings that do not exist today as virtual architectural objects. Visitors can see them using their own phones. Virtual creatures may help visitors as city guides.

The overlaid virtual model is not only for entertainment. For example, we can suppose virtual presentation to the public, including advertisements, signboards, and digital arts. Also, mobile commerce functions can be embedded in these applications. The virtual city system is expected to be a common infrastructure for mobile public applications.

There are other research activities that give users a mixture of real and virtual things in outdoor environments [4-8], and those aimed at mobile entertainment applications [9,10]. However, none of them adopted popular mobile phone terminals on the current consumer market. By adopting them, we can avoid problems of terminal costs, terminal distribution, usability for wide range of people, and building a new communication infrastructure that should be available everywhere. Our goal is to let consumers join virtual worlds with only their own phones, without paying extra money for new devices.

II. BROWSER-BASED PROTOTYPE

We have developed a prototype system giving a small virtual world overlaid onto our university campus, for mobile phones with web browsers. We have evaluated it from the usability and performance aspects.

A. *System Configuration*

It is basically a client-server system. Clients are mobile phones on the Japanese market with a GPS (Qualcomm's gpsOne) function and internet accessibility. Terminals used for evaluation were A5303H II, A5401CA, and A5501T provided by KDDI with "au" brand[1], but other types of terminals can be used if they support GPS.

Most of the required functions are implemented at the server side. We do not provide any special software for terminals, since GPS functions cannot be controlled from user's software on the terminal, when we designed it.

The server's main function is to generate a static image of virtual city for each user. When a user accesses the server, location parameters are attached to the request message by the gpsOne server. The virtual city server can then detect the location of the user and compute distance and direction between the user and virtual objects.

Since these terminals do not have any orientation sensors, the system should detect the direction in which its user is facing. A user is asked by the server about the direction he would like to see. The user should select one of eight directions: N, NW, W, SW, S, SE, E, or NE. It is passed to the server to generate a still image of the virtual city that is supposed to be a view from the user.

Data of virtual objects are stored as LightWave 3D data files at the server side. A LightWave data is loaded to an image generation module written by Java using Java3D package, and converted to a 2D image (120 x 120 pixels). An image generation process is invoked by the servlet mechanism triggered by a user's request. Finally, the generated image is sent to the terminal as an HTML file, with distance and direction parameters as text. Note that the output is not updated automatically because the terminal cannot send a GPS request without a manual operation. Also, this version does not support dynamic behavior of virtual creatures.

B. *Evaluation of the Browser-based Prototype*

Evaluation sessions took place from January 11th to January 20th, 2004. The number of subjects was 49. Prior to the evaluation, we had attempted pre-test sessions and found that subjects did not always understand where they were in the virtual world, due to GPS errors (10 to 20 meters typically, more than 50 meters in the worst case). To solve this problem, we had introduced graphical appearance of real buildings into the user's view, by representing them as semitransparent objects.

For the evaluation sessions, we had prepared a small virtual world overlaid onto our campus. Fig. 2 shows a sample of user's terminal display. In this case, two virtual objects (a signboard and a house) and one real object for reference (a building) are shown. A direction into which the user is looking is indicated by text.

According to questionnaire to subjects, they were very much interested in the virtual city application with mobile phones (average=4.35 in five grades rating: 5=best and 1=worst). Many subjects answered that they would like to visit

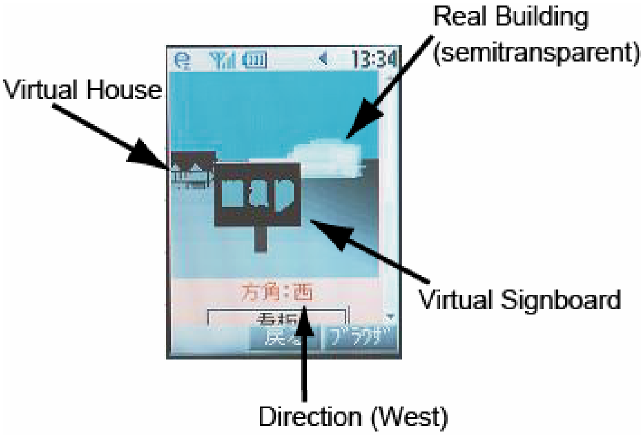

Figure 2. An Example of Mobile Phone's Display

---

[1] http://www.au.kddi.com/

the virtual city again if the virtual city continued to expand (average=4.20.) We found no statistically significant differences between male and female or between students' majors, for these results.

However, the following problems were found by their evaluation.

(a) They had to select a direction and create a mental model of city by their own efforts.

(b) The graphic image was not updated automatically.

(c) Size of generated images was small. A virtual city scene was represented by an image of 120 x 120 pixels.

(d) System response was not satisfactory. Especially, subjects complained about the time to obtain GPS data (about 15 seconds).

(e) GPS error.

If we used the newest types of mobile phone (with electronic compass, BREW architecture, QVGA wide display, and of course real-time GPS tracking), problems like (a), (b), (c), and (d) would be solved or alleviated. Problem (e) cannot be easily solved in a short time, but some new phone terminals already have Bluetooth transmitter/receiver or printed code (e.g. 2D barcodes) readers, which will be able to be used as location sensors with physical tags embedded in real cities.

With these results, we can expect a better system with the next generation mobile phone terminals. In section III, we will describe a BREW-based solution.

C. Q-based Agents

On the evaluated prototype, virtual creatures are not really supported in the sense that they could not move or interact with other virtual objects or users. After the evaluation, we have developed mechanisms for interaction between a user and a virtual creature and for active behavior of virtual creature.

We have adopted Q [11] language to control virtual creatures. Q is a scenario description language for multi-agent systems. It is based on the Scheme language and supports scenario description, including event-driven actions. On our system, virtual creatures are agents in terms of Q system. Fig. 3 is a simple example of interaction between a user and a virtual creature. In this case, the dog is saying "Hello," as a response to a user's utterance. Of course this kind of interaction function is useful for game scenarios.

The Q engine is installed at the server side, and invoked by HTTP requests from the terminal. The invocation mechanism is implemented with the servlet mechanism. Actions, like movements or interaction with the user, are also supported. For example, a virtual creature (= a Q agent) may be triggered by a particular phrase (e.g. "go away") from the user. By adopting Q language, the system gives an easy way to define agents and, of course, to describe game scenarios.

Figure 3. Interaction with A Q-based Agent

III. BREW-BASED IMPLEMENTATION

BREW[2] [12] is a software platform for CDMA-based mobile terminals designed by Qualcomm, Inc. In Japan, some of "au" brand mobile phones released from KDDI support the technology. With BREW, we can develop terminal-side software with C or C++ language. Also, rich API is supported to control mobile phone hardware, including GPS receivers and electronic compasses. Especially, some terminals including W21S[3] (shown in Fig. 4) have GPS receivers, electronic compasses, and also a QVGA (240 x 320 pixels) wide displays. For these terminals, we can develop terminal-side software supporting real-time tracking of location and orientation. We expect that it will solve most of the problems found in the evaluation mentioned in section II.

Symbian OS[4] may be another candidate for our software platform. However, it cannot be used with GPS on terminals used in Japan. It is currently not as popular as BREW in Japan, neither.

Fig. 4 shows the configuration of our first prototype for BREW-based terminals[5]. It uses Mascot Capsule's Micro3D graphics package[6], which can import LightWave 3D data through a conversion tool. On this version of prototype, all data of virtual objects are stored within the mobile phone terminal by pre-loading. Hence it runs basically stand-alone, except for communication with the gpsOne location server.

GPS and e-compass data are obtained periodically. E-compass data is taken every one second, and GPS data is obtained every 15 seconds. (These interval values are tunable.) After getting those data, the rendering engine draws the virtual view. Rendering is suppressed until the user takes a big turn. Currently, a turn of less than ten degrees does not cause redrawing.

We have evaluated this prototype in the same area as the evaluation session mentioned in section II, and found:

1. It takes about 0.8 seconds for drawing one page.

---

[2] http://brew.qualcomm.com/brew/

[3] http://www.au.kddi.com/seihin/kinobetsu/seihin/w21s/ (in Japanese)

[4] http://www.symbian.com/

[5] The photograph was taken in a building in order to get a clearly contrasted image. Of course, outdoor use is usual.

[6] http://www2.mascotcapsule.com/

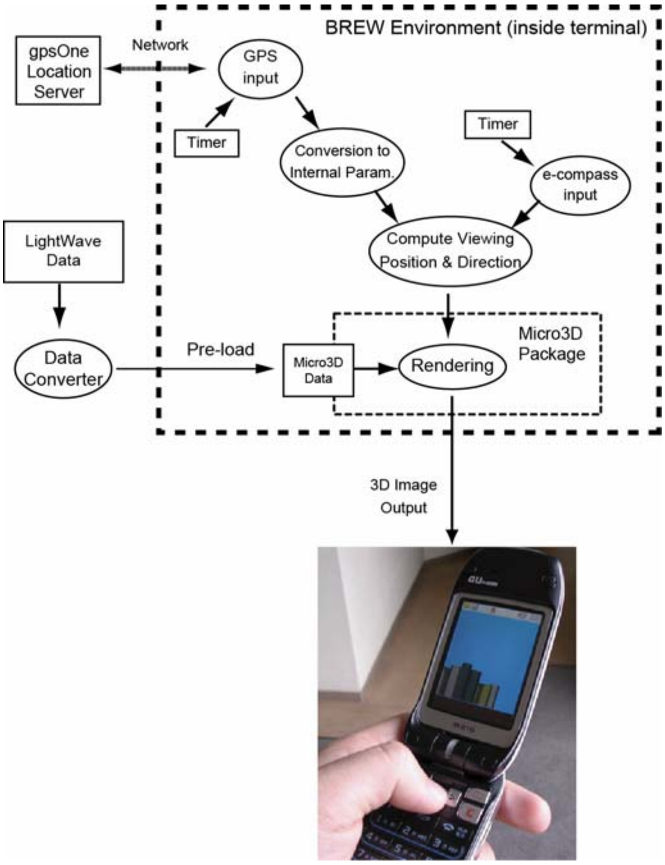

Figure 4. Configuration of the Brew-based System Prototype

2. When the user takes a turn, the graphics follows smoothly without frustration.

3. GPS errors still sometimes make users lose their positions in the virtual world. We are now tuning its graphical parameters and density of virtual objects, and have found it is getting better.

We are now designing a next client-server version, where we use the SpaceTag server[7] for the management of virtual objects. A virtual object data has attributes of its location and 3D representation in the SpaceTag server. 3D graphics data are also cached in the terminal in order to redraw it smoothly. Due to the restriction of terminal's memory and mobility of users, the cache should be updated when a user moves to a particular distance. However, even if a user does not move, the cache should be updated in some cases, because the virtual world is active and attributes of virtual objects (e.g. location) are subject to change. In these cases, triggers to update the cache are sent from the server.

IV. CONCLUSION

We have described browser-based and BREW-based implementations of the virtual city system, based on the overlaid virtual model. By adopting this model, the system provides entertainment media designers with a flexible design framework for end-user's virtual experiences.

The most important design goal of the virtual city system is to avoid extra costs of consumers for new devices. It is a more realistic approach than other outdoor virtual systems.

By evaluating the browser-based prototype, we have found that subjects were well interested in the virtual city concepts. This result suggests that the virtual city platform is expected to be used for entertainment applications. Many of the problems found in the evaluation of the browser-based prototype are solved by the BREW-based design. We have designed standalone and client-server versions of BREW-based prototype, and evaluated the standalone version.

We still have future work. (1) GPS errors are inevitable. Until more precise location systems are available to consumers, we need alternative means to compensate errors and to help users. (2) We need a mechanism for integration of the BREW-based system and Q language in order to implement direct interaction between users and agents. It will be designed by providing another communication channel between the terminal and the Q-engine in the server, using BREW's network protocol support.

REFERENCES

[1] H. Tarumi, K. Morishita, M. Nakao, and Y. Kambayashi, "SpaceTag: an overlaid virtual system and its application", Proc. of Int. Conf. on Multimedia Computing and Systems (ICMCS'99), Vol. 1, pp. 207-212, 1999.

[2] H. Tarumi, K. Morishita, Y. Ito, and Y. Kambayashi, "Communication through virtual active objects overlaid onto the real world", Proc. of 3rd Int. Conf. on Collaborative Virtual Environments (CVE 2000), ACM, pp. 155-164, 2000.

[3] H. Tarumi, K. Morishita, and Y. Kambayashi, "Public applications of SpaceTag and their impacts", Digital Cities: Technologies, Experiences and Future Perspectives, T. Ishida and K. Isbister (Eds.), Lecture Notes in Computer Science, Vol. 1765, pp.350-363, 2000.

[4] H. Schnadelbach, et al., "The Augurscope: a mixed reality interface for outdoors", Proc. of CHI 2002, ACM, pp.9-16. 2002.

[5] S. Feiner, B. MacIntyre, T. Hollerer, and A. Webster, "A touring machine: prototyping 3D mobile augmented reality systems for exploring the urban environment", Proc. of 1st Int. Symp. on Wearable Computers (ISWC '97), IEEE, pp.74-83, 1997.

[6] T. Pyssysalo, et al., "CyPhone -- bringing augmented reality to next generation mobile phones", Proc. of Designing Augmented Reality Environments 2000, ACM, pp.11-21, 2000.

[7] K. Satoh, M. Anabuki, H. Yamamoto, and H. Tamura, "A hybrid registration method for outdoor augmented reality", Proc. of IEEE and ACM Int. Symp. on Augmented Reality, pp.67-76, 2001.

[8] T. G. Kanter, "Attaching context-aware services to moving locations", IEEE Internet Computing, Vol. 7, No. 2, pp.43-51, 2003.

[9] A. D. Cheok., et al., "Game-City: a ubiquitous large area multi-interface mixed reality game space for wearable computers", Proc. of 6th Int. Symp. on Wearable Computers (ISWC.02), IEEE, pp.156-157, 2002.

[10] M. Flintham, et al., "Where on-line meets on-the-streets: experiences with mobile mixed reality games", Proc. of CHI 2003, ACM, pp. 569-576, 2003.

[11] T. Ishida, "Q: a scenario description language for interactive agents", IEEE Computer, Vol.35, No. 11, pp. 42-47, 2002.

[12] N. Leavitt, "Will wireless gaming be a winner?", IEEE Computer, Vol.. 36, No. 1, pp.24-27, 2003.

---

[7] For the scope of this paper, you may just regard a SpaceTag server as a web server that can distribute data of virtual city objects.