Parcel's information visualization on mobile Device

Andriamasinoro Rahajaniaina[1] Jean-Pierre Jessel[2]
VORTEX Group
Institut de Recherche en Informatique de Toulouse
Université de Toulouse - Paul Sabatier, 118 Route de Narbonne
31062, Toulouse, France
{rahajani[1], jessel[2]}@irit.fr

ABSTRACT

This paper presents the use of Augmented Reality system for visualizing a distributed parcel's information in a mobile device using wireless network. Our system offers useful information (number of parcel, name of the owner, agronomic structure, juridical state, and adequate plant) related to each parcel geo-referenced according to the user's position and orientation. This information augmented the live images of the real environment surrounding the user. For receiving these live images, we used two kinds of camera: webcam for an Ultra Mobile Personal Computer and notebook, and SD camera for Pocket Pc hardware. We used inertial sensor MTi and Global Positioning System receiver to achieve user's position and orientation. Internet Communication Engine, an object-oriented middleware is used to ensure the connection between database servers and clients. The users can interact with the images of surrounding environment using classic interaction tools (stylus, buttons ...).

Keywords

Distributed Augmented Reality, Visualization of GIS data, Interaction with GIS data, Ubiquitous system.

1. INTRODUCTION

In Human Computer Interaction, quality of user interface is important. Augmented Reality (AR) is among of the technique used to perform an user interface for ubiquitous application. AR presents information in its context within a 3D environment. The goal is to create the impression that the virtual objects are part of the real environment. Geographical Information System (GIS) database takes an important place for an outdoor AR system.

Most of previews researches [Höl99][Käh06] [Rei07] used GIS database corresponding for building, streets in order to enhance the experiences of users (e.g. tourists, visitors). These systems overlaid digital information (such as building name, road name) on the real world in order to perceive remote or local geographical information.

Instead, the system presented here talks about visualization of parcel's characteristic depending on location and context. As GIS database stores numerous data, retrieving information from it is among of one critical point in a distributed AR system because it may generate latency during exchange. To overcome this problem, we show in detail our approach about retrieving information related to a Parcel from GIS database, and the adequate metaphor of visualization and interaction of them.

For receiving live images of real environment, we used two kinds of camera: webcam for an Ultra Mobile Personal Computer (UMPC) and notebook, and SD camera for Pocket Pc hardware.

In this paper, section 2 reviews the related work. After that, we describe in detail our *ARGisUbiq* system in Section 3. Finally, we conclude and provide possible perspectives for future investigations.

2. RELATED WORK

Outdoor AR systems have traditionally been reserved to use GIS databases related to buildings and streets in order to provide help to users. Several systems are presented hereafter.

First, Mars (Mobile Augmented Reality Systems) project [Höl99] 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. It allowed user to arrange the multimedia information according to chronological order. This system used a campus database to overlay labels on buildings seen through a tracked head-worn display. Users was able to request additional overlaid information, such as the names of a building departments, and to view related information, such as a department web page, on a hand-held display.

Archeoguide project [Gle06] was designed to increase real images of user's environment with virtual story information related to them.

Next, in [Lia05], the authors presented a prototype of an interactive visualization framework specifically designed for presenting geographical information in both indoor and outdoor environments. They used ESRI Shapefiles as input of their system. They represented 3D building geometry and others attributes. Participants can visualize 3D reconstructions of geographical information in real-time based on two visualization clients: a mobile VR interface and a tangible AR interface.

Then, ARscouting system [Rei07] introduces an outdoor AR system witch has run on UMPC using a camera and a GPS receiver to collect information about the environment. Mobile system has been used as a thin client. While exploring the environment, the scout takes several images for instance of a target building. These images are automatically annotated by current positioning data. The enriched data are then transmitted to a custom database (multimedia database) store. Whenever a new image is stored in the database, the reconstruction engine gets a notification and triggers the reconstruction process. The engine requires at least three different views in order to generate an initial 3D model. Each further image is added in an iterative way and updates the model accordingly within seconds. Once the reconstruction task is over, the server stores the virtual object and transmits it to the mobile client (scout) in order to increase user interface. The last one is based on the Studierstube platform.

The claimed MARA [Käh06] system implements hand-held, video-see through Augmented Reality for Nokia S60 mobile imaging devices equipped with additional sensors like a GPS receiver, accelerometers and a tilt compensated magnetometer. The system allows users to interact with their surrounding environment using the standard mobile device inputs. It allowed users to place hyperlink at their current location in order to give information about an object. The users could share or exchange all data with others connected users.

The following Section expands our *ARGisUbiq* platform.

3. ARGISUBIQ SYSTEM

ARGisUbiq system is an improved version of [And08] which runs on Windows Vista, Windows mobile XP and Windows CE dedicated for desktop PC or notebook, UMPC and Pocket Pc hardware devices. [And08] was a new architecture for a multiplatform AR which allowed the users to change in dynamic way their virtual workspace. The work plan is augmented by the virtual workspace. Each virtual workspace relied to several virtual objects. For adding virtual objects, we used a virtual menu inspired by the metaphor of forward and next buttons.

The main goal of *ARGisUbiq* system is to propose a new application AR GIS in agronomic domain that shows all information about a parcel according to the user's location. In our knowledge, this is the first AR system using parcel's information to enhance user's visualization interface.

Figure 1. The UMPC visualization tool

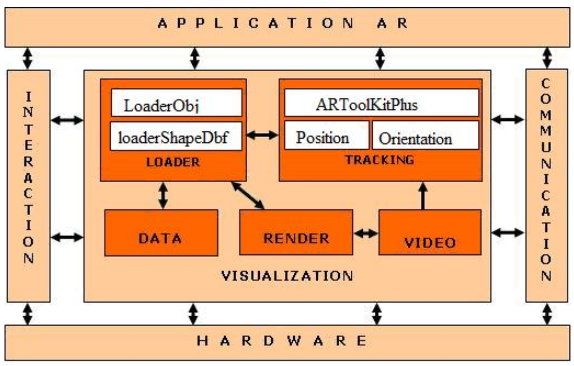

Figure 2. The software architecture

We added some module in the software architecture of [And08] in order to enhance its functionalities (see Figure 2):

- *Communication* module ensures the exchange between clients and servers, and between all modules.
- *LoaderShapeDbf* module is responsible of parcel's information loading from the Data module using the database's structure (see Figure 3).
- *Position* module and *orientation* module retrieve information from GPS receiver and MTi inertial sensor.

Our system is made of two modules: server module and client module (see Figure 5). About 90% of the task was run on the client part. Visualization module and communication module are the two main modules of the client. Data module may be available on the client and/or on the server module. The servers are used as database servers.

3.1 Information source

In the following section we present our database structure and explain the information's selection mode.

3.1.1 Database's structure

Like others outdoor AR systems, *ARGisUbiq* system uses GIS data as data source. As parcel's information related to agronomic and type of plant is unavailable on producer's map, we create our own database inspired from parcel database (using a vector format formed by shape files, index one and dbf one) (see Figure 3). In the Figure 3:

- *Parcel*'s table stores information about parcel in the public register of lands. It has five attributes: the numParcel indicates the parcel's number, numFeuille designates the number of page, numeSection is the number of section, codeCom signify common's code and nomCom is the name of common.
- *JuridicalState*'s table stores data related to the juridical state of parcel. *Num_Situat* and *libelle* are its attributes: the first one is number of the juridical state and the second one designates the label of the juridical state.
- *AgronomicalState* table is designed to stock data associate to the state agronomic. The attribute *Num_prte_a* is the number of the agronomical state. *Type_sol* indicates the structure of the parcel and *ph* is the ph of the land.
- *Adequateplant* table stores data related of all type of tilling. It has three attributes: *id_cult* indicates the identity of the tilling, *libelle* is the name of the tilling and *detail* relates to the detail of the tilling's feature.

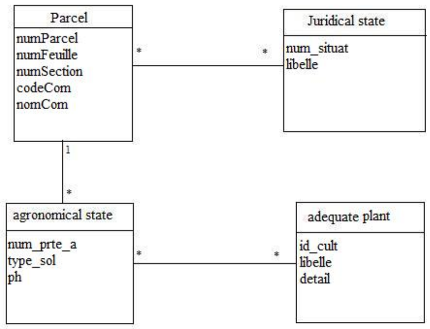

Figure 3. The database's structure

We can note that this database may be available on the clients and/or on several replica servers. The clients establish a connection to the servers using the communication module based on Internet Communication Engine (Ice) [hen03] when local database is unavailable. This later case occurred while the device have not enough space disk (UMPC or notebook) or space memory (Pocket Pc) for storing the database or it is deleted.

3.1.2 Information's selection mode

Information retrieval procedure occurs when the camera's orientation turns to the ground. GPS receiver (TomTom wireless GPS receiver) and MTi inertial sensor are used for tracking the camera's position and orientation. The GPS receiver exchanges information with the client using a BlueTooth connection whereas the connection between client and MTi sensor is established by a serial communication. Orientation values provide by MTi relate to the orientation of its coordinate system S(x, y, z) according to the fixed global coordinate system G(X, Y, Z) (see Figure 4).

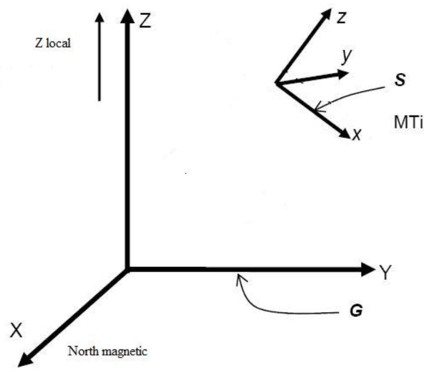

Figure 4. The coordinate system of the MTi

Each polygon in the shapefile is delimited by a bounding box. In order to know the existence of parcel according to user's location, we check if user's position is in the bounding box of its record. When his location is included in the limit, we save the number of row and bounding box of all corresponding record. Whether the record number is more than one, we take account only the record (polygon) having its barycenter nearby the camera's position. After that, we use the row number related to the selected parcel for searching others information (agronomical state, adequate plant, juridical state …) in the associated tables. While the user's location is always in the bounding box, we work (research a new nearest barycenter of polygons) with these existing number rows and bounding box values of each record. We made it in order to reduce the amount of requests to the database which produce latency indeed for lightweight hardware devices.

3.2 Distributed architecture

As we describe above, *ARGisUbiq* system is composed of two modules: client module and server module which use geo-referenced GIS database. The Clients connect to the servers using WLAN network. Our framework uses Internet Communication Engine (ICE) and IceE (Ice Embedded: a lightweight version of Ice for mobile devices) to ensure the connection between the clients and the servers.

With the aim of having flexible data distribution, we duplicate on several servers our database and IceGrid services is used to establish load balancing between the client and all replica servers. It provides a convenient way to distribute an application to a set of computers, without the need for a shared file system or complicated scripts. Each server may have one registry which control one or several node's activities. Registry implements locator service and the locator object is available on the registry client endpoints (IP address or hostname and port number). A Node monitors the load of their computers (servers) and reports this information to the registry. This one uses this information to decide which endpoints of the object adapters to return to a client. In general, the server selects one or a set of endpoints which have the least-load statistics.

In presence of several Ice servers, one which have master registry is the master server and others one are slaves. Slave or master server property is specified in their configuration files. A first locate request activates the application server automatically (starting the Ice server process). Activation usually occurs as a side effect of indirect binding, and is completely transparent to the client. Node is responsible of this activation task when it receives registry's order. Node sends responses to a registry according to its configuration file (the replica's number to include in the registry's response is specified in this file).

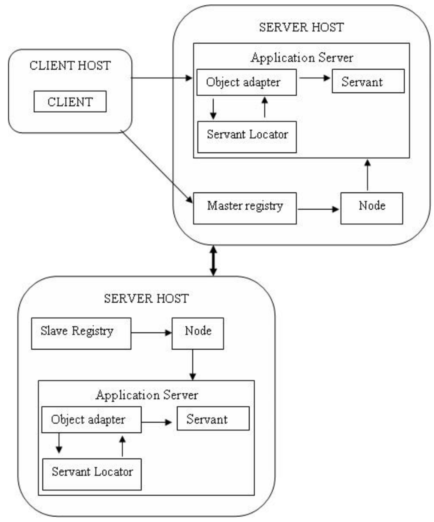

Figure 5. The distributed architecture

Selected application server uses object adapter in order to obtain information about an object (parcel) requested by a proxy client. Then, object adapter attempts requests to a servant which is a direct responsible of one or more objects.

Notice that multithreading is supported by Ice server. In fact, this property allows more clients to establish connections in the same time.

The master replica knows all of its slaves, but the slaves are not in contact with the others. If the master replica fails, the slaves can perform several vital functions that should keep most applications running without interruption. Eventually, however, a new master replica must be started to restore full registry functionalities. For a slave replica to become the master, the slave must be restarted.

Client module uses Ice/IceE in order to retrieve parcel's information from the replica server when a local database is unavailable. We deactivate proxy cache (that contains all information about previous server) and set a timeout to cache locator in order to make load balancing. IceGrid's load balancing capability assists the client in obtaining an initial set of endpoints for the purpose of establishing a connection. Before attempting locate request to a server location, client checks its cache locator. We randomize selection of object adapter's endpoints used by proxy client to establish connection in order to collect all object's information. One Client's request is formed by endpoint and object id.

As these properties are unavailable on IceE, in fact, to simulate the functionality of load balancing, we set a timeout for an established connection and close it in order to establish another one to other endpoints when the timeout is expired.

3.3 Visualization metaphors

With the aim of seeing augmented view, users must hold account information's selection principle. Once this one is respected, users saw real video augmented by parcel map, juridical situation, agronomical state and adequate kind of tilling related to the user's location. To achieve it, we propose two visualizations metaphors: *textview* mode and *hybridview* mode. In the *textview* mode, the scene is augmented by virtual text and aural information related to a parcel and the position of user. In the *hybridview* metaphor, user interface is enhanced by virtual text, parcel map and audio information. The blue point on the map is the user's position.

Figure 6. The hybridview mode

Figure 7. The hybridview mode on the UMPC

We combine *landscape* and *portrait* mode with *textview* and *hybridview* when users use lightweight hardware as visualization tools. The transition between the two metaphors depends on the way which the user holds his PDA.

Figure 8: The landscape hybridview mode

Figure 9: The portrait hybridview mode

Figure 10. The landscape Textview mode

The *hybridview* is the default visualization metaphor for UMPC and PC notebook, and the *textview* metaphor is for PDA.

3.4 Interaction metaphor

We propose a possibility for user to choose visualization metaphor using *textview* and *hybridview* menus. We decide to use classical interaction tools like menu, stylus, and button because these are available on each device that we use as visualization tools. When the user selects one of both menus using his stylus, the user interface changes according to the menu item selected. After that, the menu item changes to another one.

As described above, when using a PDA, the transition between the two metaphors depends on how the user holds his PDA and the value of pitch angle (Θ) from the inertial sensor MTi (Θ value between -5.0° and 0.0° for portrait mode and landscape mode for others values).

If the user needs additional information related to the kind of tilling, he selects "more info" and listens the aural information.

4. Experimentation and Results

This first prototype was tested with three users: the first client has used Q1 Samsung with 800 Mhz Celeron M ULV processor, 256 Mo RAM and the two other clients have used a Pocket PC dell axim x51v with 624Mhz Intel xscale processor, 64Mo RAM. We have used a database of common formed by 180,000 parcels and each parcel is formed by 10 up to 20 vertices. We have tested two different scenarios: first, we have used a local database: as we have loaded the database in the memory at the first time, the Q1 client has run after 5s of the database loading and 22s for the two PDA clients. After this step, the Q1 client was able to achieve 25-30 fps (frame per second) and 17-20fps for the PDA clients during the exchange with the data in memory. In the second test, the database is duplicated on three replica servers. One master registry and two slave registries. Each slave registry has had its own node which monitors two applications servers. The locate request from the client to the registry has spent 0.3s for Q1 client and 0.7s for PDAs clients. After that, the Q1 client was able to 23-28fps and 15–20fps for the PDAs clients during the exchange with the replica server.

After these tests, we asked the users about ergonomic of user interface and about the visualization hardware device: 80% of the users are satisfied about user interface but 50% only for hardware device.

5. Discussion

As we saw, the difference between the results using the local database and replicate database was small. It's not surprising because we have added to the client a functionality to reduce the number of exchange with the replica server (see section 3.1.2). It's also due to the performance of our replica servers. 50% of users only are satisfied for hardware device ergonomic because most of users prefer using wireless inertial sensor instead of using MTi. It is easy to use.

From these results we can deduce designs guidelines for choose of hardware device in future AR application.

6. Conclusion and future work

In this paper, we addressed the problem of enhancing user's contextual perception of the real word using GIS data on several hardware and software platform. To tackle this, we have proposed the *ARGISUbiq* multiplatform architecture which exploits Mobile Augmented Reality principles to improve user'sinteraction with GIS data. As we use specifically built GIS data, we described our database's structure and how to select appropriate information related to user's position. Our distributed application is based on Internet Communication Engine, an object-oriented middleware, used to ensure the connection between database servers and clients. To avoid eventual problem with database server, we duplicate our database on several servers and we use Icegrid services to provide load balancing between all servers. Some clients are able to access concurrently to a selected server.

We are entirely satisfied with our first results. In the future work, instead using MTi sensor we plan to use low cost or embedded inertial sensor and image based techniques to compute the user's orientation

7. REFERENCES

[And08] Andriamasinoro Rahajaniaina and Jean-Pierre Jessel, A new architecture for a multiplatform Augmented Reality System, Proc. In International Conference on Signal processing and multimedia Applications, Porto Portugal, 2008.

[Gle06] Gleue and P. Daehne, Design and implementation of a mobile device for outdoor augmented reality in the archeoguide project, In Virtual Reality, Archaeology, and Cultural Heritage International Symposium, Glyfada, Nr Athens, Greece, 2001.

[hen03] Henning et al., Distributed Programming with Ice, ZeroC: http://www.zeroc.com/Ice-Manual.pdf, 2003.

[Höl99] Höllerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallaway, Exploring MARS: Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System, Computers and Graphics, 23(6), Elsevier Publishers, 1999.

[Käh06] Kähäri and D. J. Murphy, MARA-Sensor based Augmented Reality System for Mobile Imaging Device, 5th IEE and ACM International Symposium on Mixed and Augmented Reality, Santa Barbara, 2006.

[Lia05] Liarokapis, I. Greatbatch, D. Mountain, A. Gunesh, V. Brujic-Okretic and J. Raper, Mobile Augmented Reality techniques for GeoVisualisation, Proc. 9th International Conference on Information Visualisation, IEEE Computer Society, London, 2005.

[Rei07] Reitinger, C. Zach and D. Schmalstieg, Augmented RealityScouting for interactive 3D reconstruction, in Procedings of IEEE Virtual Reality Conference, Charlotte NC, USA, 2007.