Location-aware Visualization of VRML Models in GPS-based Mobile Guides

Stefano Burigat (1)
HCI Lab
Dept. of Math and Computer Science
University of Udine
via delle Scienze, 206
33100 Udine, Italy

Luca Chittaro (2)
HCI Lab
Dept. of Math and Computer Science
University of Udine
via delle Scienze, 206
33100 Udine, Italy

Abstract

In this paper, we present LAMP3D, a system for the location-aware presentation of VRML content on mobile devices. We explore the application of LAMP3D in tourist mobile guides: the system is used to provide tourists with a 3D visualization of the environment they are exploring, synchronized with the physical world through the use of GPS data; tourists can easily obtain information on the objects they see in the real world by directly selecting them in the VRML world. We discuss the design and development of the system and report about the feedback obtained from the informal user testing we carried out.

CR Categories: H.5.1 [**Information interfaces and presentation**]: Multimedia Information Systems – *Artificial, augmented, and virtual realities*. H.5.2 [**Information interfaces and presentation**]: User Interfaces – *Interaction styles, evaluation*. I.3.6 [**Computer graphics**]: Methodology and techniques – *Interaction techniques*.

Keywords: 3D virtual environments, VRML, mobile guides, mobile devices, interaction style, evaluation.

1. Introduction

In recent years, mobile computing devices such as PDAs or high-end mobile phones have become more and more widespread and powerful, with a corresponding increase in the complexity of the applications they can support. For example, it is expected that the number of users of 3D graphics applications for mobile devices (in particular, games) will dramatically increase in the near future [ARM 2002]. However, research on the use of 3D in mobile applications is still limited, also because the first generations of mobile devices had insufficient computational power to support real-time 3D graphics. The appearance of more powerful devices, sometimes equipped with special graphics accelerators, is now making it easier to experiment with mobile 3D graphics. Another reason for the limited availability of mobile 3D applications is the lack of generally accepted solutions suitable for their development. This is improving as well, with the definition of the first APIs for 3D graphics on mobile devices, such as OpenGL ES [Khronos 2004], Java mobile 3D Graphics (JSR-184) [Jcp 2004] and Mobile GL [Kaist 2004]. On top of these APIs, software rendering engines, such as Swerve [Superscape 2004], are being proposed. An interesting alternative to these solutions is given by VRML (and X3D), because they would allow a content developer to: i) re-use a large collection of existing Web-based 3D worlds in the mobile context, ii) develop content for different platforms (mobile and desktop) with the same tools and, iii) focus more on the application domain, avoiding the need to handle low-level details of 3D graphics.

While the possibility to view VRML content on mobile devices clearly extends the potential application areas of Web3D technologies, an assessment of which specific applications would benefit most is still needed. For example, if we consider that games will drive the market for mobile 3D applications in the near future, game developers will probably look for 3D graphics APIs that allow them to obtain more performance and more control on the underlying hardware.

Because of the previously mentioned advantages of VRML/X3D, an application area that could benefit from the presentation of Web3D content on mobile devices is tourism. When tourists are visiting a city they can be interested in different kinds of information (e.g., history, culture, art, entertainment, dining, sports, shopping, …). Therefore, they often take with them paper guides to be consulted when needed. However, this is not always a convenient and efficient way to obtain the needed information. In recent years, indeed, there has been a growing interest towards the development of *mobile tourist guides* [Baus et al., 2004]. These guides can be used on lightweight mobile devices, providing easy (partially automated) access to the various classes of information. Moreover, they can manage multimedia information, enriching the tourist experience, and they usually provide many other useful services for the tourist, such as tour planning, weather forecasts, and so on. By exploiting the physical location of users, mobile guides are often able to provide up-to-date contextual information while traveling and give the possibility to access the most useful services for a given location.

Two crucial services that are usually provided by most mobile guides are *navigation support* and *information delivery*. Navigation support allows users to obtain directions to navigate an environment and to locate themselves and points of interest in the surrounding area. Information delivery gives users the possibility to obtain information on the points of interest located in the visited area.

While the usually adopted solution to provide the two above mentioned services is to employ graphical representations such as 2D maps (which have the advantage of being familiar to users),

---

(1) e-mail: burigat@dimi.uniud.it

(2) e-mail: chittaro@dimi.uniud.it

the use of 3D representations of the environment may represent a more intuitive and effective way to provide information on the explored area. However, limitations of current mobile devices, such as the insufficient computational power and the limited screen space, make it difficult to obtain a smooth navigation of 3D representations, especially when complex environments (e.g., full cities) are taken into consideration. Nevertheless, with the increase in computing power of mobile devices, 3D graphics will probably be more widely used in this area and it is thus useful to explore how much this could benefit users and to what purposes.

In this paper, we present LAMP3D (Location-Aware Mobile Presentation of 3D content), a system for the location-aware presentation of VRML content on mobile devices. We explore the application of LAMP3D in tourist mobile guides: the system is used to provide tourists with a 3D visualization of the environment they are exploring, synchronized with the physical world through the use of GPS data; tourists can easily obtain information on the objects they see in the real world by directly selecting them in the VRML world (using a pointing device such as the PDA stylus or their fingers). While the approach we propose will be described in the context of a mobile tourist guide, it could be applied to other domains as well (e.g., urban planning, resource management, emergency response…), where an interactive location-aware 3D visualization of the environment could be useful.

The paper is organized as follows. Section 2 surveys related work. Section 3 describes in detail our approach, analyzing the different modules composing the system and their functionalities. Section 4 will provide some observations on the informal evaluation we carried out in the field to test strengths and weaknesses of the adopted solutions. Section 5 will provide conclusions and outline future research directions.

2. Related work

In recent years, there has been a continuous progress in mobile devices, wireless communication and localization technologies. This spawned several research projects concerning mobile tourist guides. Since the first pioneering prototypes, developed in the Cyberguide [Abowd et al. 1997] and GUIDE [Cheverst et al. 2000] projects, mobile guides have been progressively refined and improved and some commercial solutions have also been launched, although mainly focused on navigation (e.g., [TomTom 2004]).

What really sets apart a mobile guide from more traditional solutions (such as paper guides and desktop guides) is location-awareness. By knowing the position of the device, it is possible to easily provide users with location-aware services, thus increasing the usefulness of the guide. The current position of the user may be obtained by means of different techniques. For outdoor applications, satellite-based tracking approaches such as GPS are the most frequently used, possibly coupled with correction techniques such as DGPS (Differential GPS) or AGPS (Assisted GPS) to increase their accuracy. An alternative technique is represented by MPS (Mobile Phone Positioning) which exploits the cellular phone infrastructure (e.g., GSM or UMTS) to locate the user [Kalkbrenner and Koppe 2002]. For indoor applications, it is necessary to use other technologies such as radio frequency identification systems (RFID) [Ferscha 2002], or other wireless methods (such as Bluetooth, IrDA and ultrasounds) for locating and identifying objects [Vossiek et al. 2003].

Regardless of the specific technique used to determine the location, there are different ways to present this information on mobile devices. Most of the implemented location-aware mobile guides use 2D maps of the area where the user is located, pinpointing her position and usually providing visual information on the nearest points of interest and on the paths she has to follow to reach specific destinations (e.g., [Pospischil et al. 2001]). Maps are powerful tools for navigation because of the richness of information they can supply and the rate at which people can absorb this information. At the same time, maps require users to repeatedly switch from their egocentric perspective of the world to the exocentric perspective provided by the map and vice versa. This often requires significant mental effort and affects performance [Aretz and Wickens 1992].

Recently, some attempts have been made at exploring 3D graphics for mobile guides. Rakkolainen et al. [2001] have proposed a system that combines a 2D map of an area with a 3D representation of what users are currently seeing in the physical world, studying the effects of 3D graphics on navigation and wayfinding in a urban environment. They found that 3D models help users to recognize landmarks (i.e., distinctive features of an environment, such as churches and squares, that can be used as reference points during navigation) and find routes in cities more easily than traditional 2D maps. Unfortunately, the prototype was implemented on a laptop computer, not on a PDA. 3D city models for route guidance have been tested also by Kulju et al. [2002] who obtained similar results but highlighted the need for detailed modeling of buildings and additional route information such as street names. Unfortunately, their prototype uses only predefined animations and sequences of pictures, not interactive 3D worlds. Moreover, both projects focused only on navigation support and no information delivery service about points of interest was provided. The TellMarisGuide system [Laakso et al. 2003] supports tourists when they are visiting harbours by visualizing 3D maps of the environment along with more classical 2D maps. The 3D maps support navigation in a city and route finding to points of interest such as city attractions or restaurants. Due to limitations in the mobile clients used, only a limited number of buildings is realistically represented by the system, and, to the best of our knowledge, there is no feature allowing users to directly obtain more detailed information about the area they are visiting by interacting with the objects visualized in the 3D representation. Other systems combine video with 3D worlds: in particular, Brachtl et al. [2001] have developed a navigation system for indoor environments that guides users by showing animated walkthroughs in the form of movies, generated from 3D models of the explored area, while the INSTAR system [Narzt et al. 2003] mixes video images with 3D graphics to provide an augmented reality view of the real world for mobile navigation.

Realistic visualization of large and complex 3D models, such as those used in mobile guides, is a very important task for other application areas as well: scientific simulation, training, CAD, and so on. However, mobile devices do not include the specialized hardware typical of high-quality graphics boards, and it is thus not always possible to obtain a good quality level for the visualization. A possible approach to this problem is to carry out rendering on a powerful remote server (or a cluster of workstations) connected through a wireless network and display the results on the mobile device as a video sequence [Lamberti et al. 2003]. This solution has two advantages: the data to be visualized is processed by specialized hardware, thus bypassing the problem of the low computational power of mobile devices, and the source data is not transmitted to the client device, thus allowing for data independence. On the other side, due to the limited bandwidth of current wireless networks, this remote computation solution needs complex algorithms for the preparation of the data to be transmitted. Moreover, such a client-server architecture requires the availability of a suitable wireless

Object without info

Lat: 46.063316
Lon: 13.233590
Dir: -36.1

GPS-based navigation

Figure 1 – The user selects an object by tapping on it with a stylus.

Pillar with statue of Madonna and Child

Located at one edge of Piazza Matteotti, in front of S. Giacomo's church, this high pillar is surmounted by a statue of Madonna and Child, which was probably made in 1487, at the same time in which the Piazza was restored; the author of the statue is unknown.

Figure 2 – Information on the tapped object is visualized in a separate window.

network and is more difficult to set up for every possible environment.

3. The LAMP3D system

As reported in the previous section, most current mobile guides rely on the use of 2D maps to provide certain services to users. The effectiveness of this solution depends on the ease with which users can obtain the information they need. Automatically providing information to users without direct interaction with the guide could be useful when the information is complete and proper but, on the other side, this approach is not much flexible and can be sometimes too obtrusive. On the other hand, allowing users to request information when they need is a more flexible approach but is more complex for the user because it involves scrolling lists of available items or querying the system or trying to figure out where the relevant object of interest is located on a 2D map of the visited area.

With the LAMP3D system we explore the possibility to use 3D graphics on a PDA to provide users with content filtered according to their position and simplify the way information about objects of interest is obtained.

3.1 Overview

In the LAMP3D system, we combine a VRML representation of the currently visited area with the possibility for the user to request additional information by directly selecting the objects of interest in the 3D representation. This solution aims at making it easier for the user to obtain the desired information about an object. Indeed, the easiest way for the user to ask information about a building or some other object in a city is to point at it with a finger. Our system supports this behavior by allowing the user to touch objects of interest in the 3D representation of the city using a finger or a pointing device such as the PDA stylus (see, e.g., Figure 1).

Since a tourist is usually interested in the buildings or objects she is looking at, while information on other objects becomes more relevant only later, the 3D representation provided by LAMP3D is location-aware, being synchronized with the physical world through the use of GPS data. In this way, our system makes the information about the closest points of interest more easily accessible to the user: we thus propose a natural filtering criteria based on proximity. Our solution can be seen as a mix between current approaches to information presentation: information is automatically (visually) filtered and the 3D representation is always consistent with the actual user's view of the real world, but actual information on the visualized objects of interest must be requested by the user.

The available information about a selected object is provided in a separate window (see, e.g., Figure 2). Only textual information, organized in separate pages for better readability, was used for the purpose of testing our prototype, but adding richer media such as HTML pages with 2D pictures or videos is straightforward.

To maximize the flexibility in the use of the system, three navigation modes in the VRML world are available to the user:

1. *GPS-based navigation* is the standard navigation mode, based on the actual position and orientation of the user. The system is responsible for gathering the necessary information from a GPS device and for changing the viewpoint on the visualized 3D world so that it corresponds to the viewpoint of the user in the physical world.
2. In *manual navigation,* the user moves in the 3D world by tapping with the PDA stylus on specific buttons available in the user interface. This navigation mode may be useful for the user to examine the environment when off-line, thus acquiring information before actually visiting an area or after the visit has occurred.
3. In *replayed navigation*, the system uses position and orientation information previously recorded by a human guide or by users themselves to animate a virtual tour in the city. This navigation mode is supported by logging the data provided by the GPS unit and then feeding this data into the GPS-based navigation mode. This feature is valuable because, for example, it can be used both to prepare guided tours of an area and then propose them to tourists, and to record tourists' navigation behavior during a visit so that it can be subsequently analyzed with automatic tools, such as VU-Flow [Chittaro and Ieronutti 2004].

3.2 Architecture

The architecture of the system is depicted in Figure 3. The *User Interface Module* allows the user to interact with the 3D world, access system options, look at status information and request additional information on the objects of interest. This module receives information (such as GPS status, user position and orientation, object information, …) from the other modules and presents them to users. The *GPS Module* is responsible to get data from the GPS unit and to provide the other modules with the actual position and orientation of the user. Internally, it parses the

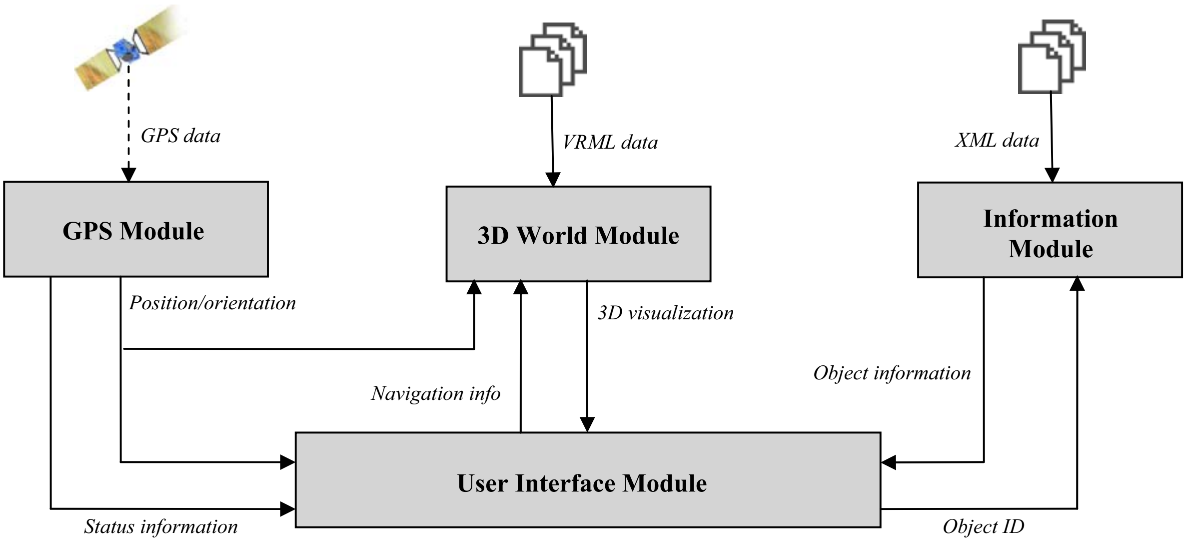

Figure 3 – The architecture of the system

flow of GPS data, acquiring the necessary information, and storing it in a proper data structure. It also manages the status of the GPS unit, providing this information to the user through the User Interface Module. The *3D World Module* is responsible for the management of the 3D representation. It exploits the Pocket Cortona browser [Parallelgraphics 2004] for the visualization of the VRML data and, among other functionalities, has to be able to update the viewpoint according to the navigation mode and to manage large-scale environments efficiently. Finally, the *Information Module* is devoted to manage the information on the objects of interest. When the user selects an object in the 3D world, this module retrieves the corresponding available data. All the information about the objects is stored locally on the device in XML files. However, it would be possible (in a future version of the system) to retrieve this data from a remote database and automatically format it according to the specific DTD (Document Type Definition) accepted by the system.

Piazza Matteotti (formerly Piazza San Giacomo)

Lat: 46.063103
Lon: 13.233252
Dir: 47.6

GPS-based navigation

Figure 5 – LAMP3D user interface

3.2.1 User Interface Module

Figure 4 depicts the interface of the first prototype system we have developed. Two main parts can be easily identified: an upper area where the actual 3D world is visualized and a lower area providing status information and tools for setting the system and moving the viewpoint if the user wants to manually navigate the 3D world. The lower area may be further subdivided into a *Menu area* containing all the commands for managing files, changing system options and activating the GPS, a *Control area* where the buttons for manually navigating the 3D representation are located, a *Status area* showing position and orientation of the user and providing feedback on what the system is currently doing and on the status of the GPS, an *Object information area* showing the currently selected object and availability of additional information (which can be retrieved by clicking the Info button).

In the Control area, four buttons allow the user to manually navigate the 3D world while the central button opens a separate window where the user can manually specify coordinates in the world (in the VRML coordinate system or as latitude/longitude) or can choose a predefined viewpoint among the available ones.

By using the options in the Menu area, the user can change some of the parameters of the system. In particular, it is possible to set the speed of navigation (in the case of manual navigation) and the viewpoint angle and set some GPS options such as communication port, operation mode (with real data or replayed data) or data logging (for recording data to be used in replayed navigation).

The status of the GPS, on the right side of the Status area, is provided by a letter indicating the navigation mode (M for manual navigation, R for replayed navigation and G for GPS-based navigation), a virtual led indicating if the GPS is active (green color), deactivated (gray color) or idle (yellow color), a color rectangle providing feedback on the received data (green color for valid data, red color for invalid data, gray color for no data availability) and an area showing the number of available satellites.

3.2.2 GPS Module

This module handles the data coming from the GPS unit, extracting the information needed to determine the actual position and orientation of the user and processing it to improve the accuracy.

GPS data must satisfy the NMEA 0183 protocol [NMEA 2004]. Data formatted according to this protocol consists of a sequence of ASCII characters, subdivided in different sentences identified by their first six characters: the $ symbol, followed by the talker ID (which is always GP for GPS receivers) and the sentence ID. The identifier is followed by the sequence of actual data fields, and an optional checksum. The GPS data is parsed so that the relevant information can be extracted (only some of the NMEA sentences containing position data, in particular $GPRMC and $GPGSV, are examined while the rest, providing other information on the data, is discarded).

Since data received from the GPS unit might be affected by errors up to 10 meters, we process it to improve the accuracy. In particular, to estimate the starting position after the activation of the system, the first 10 valid data points are analyzed by allocating them to buckets according to their latitude and longitude (every bucket represents a cell in a grid that is obtained by taking the maximum and minimum values of longitude and latitude of the points and subdividing these intervals in equal parts). The mean value of the points in the most filled bucket is then considered as the possible starting position. The subsequent points are instead analyzed using the following heuristics. First, a threshold is applied to discard points that are too different from previous values. Then, if the speed of the user is close to 0 (that is, the user is not moving), an averaging technique is constantly applied to all data received during this condition (to improve the accuracy of the position), otherwise the point received can be directly used.

While the GPS unit provides also data about the current direction, this information is often unreliable. Thus, our GPS module employs an additional algorithm for the computation of the orientation of the user, exploiting the previously filtered position data.

The GPS module is also able to read GPS data from files rather than getting it from the GPS unit, to support the replayed navigation mode.

3.2.3 3D World Module

This module accesses and manipulates the VRML nodes representing the 3D environment shown to the user. The most important operation it performs is to correctly position the viewpoint of the 3D world, according to the data provided by the GPS module.

The values obtained from the GPS module cannot be directly used to update the viewpoint since they have to be further examined (and processed) to meet some constraints. In particular, the viewpoint must be contained inside the 3D world but, at the same time, it must not be inside the buildings composing the environment. Moreover, the viewpoint must be always at the same distance from the terrain (which can also be uneven). The viewpoint coordinates must also be converted from the latitude/longitude system to the VRML reference system. To support the above mentioned processing, the VRML file includes an instance of the following `GeoOrigin` **`PROTO`**:

```
PROTO GeoOrigin
[
     exposedField SFVec3f coordinates []
     exposedField SFVec3f multipliers []
     exposedField SFVec3f boundsCenter []
     exposedField SFVec3f boundsSize []
]
{}
```

The `boundsCenter` and `boundsSize` fields define the bounding box containing the world while the `coordinates` and `multipliers` fields are used to perform coordinates conversion (in particular, `coordinates` specifies latitude, longitude and altitude of the origin of the VRML world while `multipliers` specifies the multiplicative factors for the conversion).

Another issue that has been taken into consideration during the design of the system is the difficulty for current mobile devices to smoothly manage complex 3D worlds (such as whole cities) at once. Indeed, due to limitations in computational power and main memory, it becomes necessary to employ ad-hoc techniques which allow one to limit the quantity of data to be considered. In our system, we used a technique inspired by [Marvie and Bouatouch 2004]. The principle on which it is based is occlusion-culling. If only viewpoints at the street level are considered, virtual cities are indeed typical examples of densely occluded environments: only a small set of the primitives of the whole model will be visible at any given time instant. Walking in a city, only a small number of buildings will be indeed visible at any time. The model of an environment can thus be subdivided into cells, each one containing the set of objects (called PVS, *Potentially Visible Set*) which are visible inside its area. The generation of these cells can be carried out in a pre-processing phase using different approaches (e.g., region-based). During navigation in the environment, it is sufficient to know when a user moves from a cell to another and retrieve at that moment the corresponding cell data.

In our system, all cell management is carried out inside the VRML file, by a script. The script is activated when a user exits the cell she is currently in. At this point, the next cell is determined, all the objects in the new cell are identified and their visibility condition is set (while the objects in the old cell are deactivated). According to this information, the corresponding objects are loaded. The advantage of this solution is that it can be used with all existing VRML browsers (both for mobile devices and desktop computers).

The information on the cells is contained in an instance of the following `cellsSet` **`PROTO`**:

```
PROTO cellsSet
[
   exposedField MFVec3f cellSize []
   exposedField MFVec3f cellCenter []
   exposedField MFInt32 pvsIndex []
   exposedField MFInt32 pvsInfo []
   exposedField MFInt32 neighIndex []
   exposedField MFInt32 neighInfo []
   exposedField MFInt32 visibility []
]
{}
```

The `cellSize` field contains the dimension of the cells as vectors of Cartesian coordinates (for efficiency reasons, cells are represented as parallelepipeds aligned with the axes of the reference system). The index in the `cellSize` vector is used as identifier for the cell. The `cellCenter` field identifies the geometrical centers of the cells. These two fields are used to define a `ProximitySensor` that detects when the user exits the current cell. The `neighIndex` and `neighInfo` fields are used to define the adjacency relations among cells. In particular, the `neighIndex` field is used to identify the subset of cells in the `neighInfo` field adjacent to a given cell. Consider, for example, the following two instances:

```
neighIndex [0, 2, 5, 7]
neighInfo [1, 2, 0, 2, 3, 0, 1, 1]
```

and suppose we want to know the neighbors of cell 2. We then use the index of the cell (2) to access the `neighIndex` field, retrieving `neighIndex[2]=5` and `neighIndex[3]=7`. Thus, the neighbors of cell 2 can be identified by examining the `neighInfo` field from position 5 to position 6 (`neighIndex[3]-1`). In this example, they are represented by cells 0 and 1. Notice that this data structure allows one to define an arbitrary combination of cells of different dimensions and with a different number of neighbors. On the other side, in the case of many identical cells, this kind of data structure is less efficient. The `pvsIndex` and `pvsInfo` fields are used to identify the set of objects contained in the cells. Every object is identified with an integer number and the `pvsInfo` field contains such numbers. To retrieve the subset of objects belonging to a cell, the same procedure previously used to identify neighbors is used. The `visibility` field contains the visibility condition of the objects in the world (the index in the `visibility` field is used to identify the corresponding object). In the current implementation of the system, a 1 value corresponds to an active (and visible) object while a 0 value corresponds to an inactive (and invisible) object. Another possible implementation is to use a positive value to identify different Levels of Detail (LOD) for an object. While the previous fields are read only and are created during the pre-processing phase, the `visibility` field is manipulated by the cell management script during the visualization of the world. The actual objects composing a world are stored in a set of `Switch` nodes, one for each object. In particular, the `choice` field in each `Switch` node contains all the information (position, geometry, etc.) on the object while the `whichChoice` field is set to activate or deactivate the object according to the corresponding value contained in the `visibility` field of the `cellsSet` **`PROTO`** instance. Multiple representations of an object may be supported by storing different `LOD` nodes in the `choice` field and using the proper value for the `whichChoice` field. It is also possible to use the `Inline` node in the `choice` field so that object descriptions can be stored in different VRML files, even remotely, thus possibly providing a limited client-server architecture.

3.2.4 Information Module

This module manages the descriptive information on the objects of interest and provides this information to the User Interface Module. This information is stored in XML files that are formatted according to the following DTD:

```
<?xml version="1.0"?>
<! DOCTYPE GTNInfoFile [
<! ELEMENT GTNINFO (NAME, INFO) >
<! ELEMENT NAME (#PCDATA) >
<! ELEMENT INFO (ITEM*) >
<! ELEMENT ITEM (SHORT, LONG) >
<! ATTLIST ITEM Id ID #REQUIRED >
<! ELEMENT SHORT (#PCDATA) >
<! ELEMENT LONG (#PCDATA) >
]>
```

In particular, the INFO element contains a list of ITEM elements, each one associated with an object of interest (and identified by an ID). Each ITEM element includes a SHORT element, which is used to provide the user with some summary information about the related object through the Object information area in the User Interface, and a LONG element providing detailed information about the object. Using this DTD, it is thus possible to automatically generate XML files for the guide from a database.

3.3 Implementation

The LAMP3D system has been developed in eMbedded Visual C++ for the PocketPC 2002 platform. We used VRML as the description language for the 3D world and the Pocket Cortona browser to display the 3D world on a Compaq iPaq h3970, that features an Intel Xscale 400MHz processor and a 320x240 screen resolution. A Compact Flash GPS unit, supplying tracking data through the NMEA protocol, has been used to acquire position and orientation information.

In the current version of LAMP3D, all data (geometric data on the environment and information on the objects) is stored locally in the memory of the mobile device. Although this could be a limitation for a full-featured mobile guide, which might benefit from the possibility of downloading information wirelessly by exploiting a client-server architecture, it allows for a faster data retrieval and was sufficient for evaluating our approach in the field.

4. LAMP3D evaluation

To obtain information on the usefulness of LAMP3D and, more generally, better understand the potential and problems in the use of 3D graphics on mobile scenarios, we informally tested the system in the field in a square of the city of Udine (Figure 5). We must point out that the test was carried out on a limited group of users, thus providing only some preliminary indications (albeit useful, especially those concerning problems).

We activated data logging (allowing us to later use replayed navigation) and video recorded users to better study the behavior of the system. After starting the system, when a sufficient number of satellites was acquired (all icons in the GPS area had to be green), users freely moved in the square and used the PDA stylus

Figure 5 – LAMP3D field test: users are provided with faithful reconstructions of the objects they see in the real world.

to tap on their objects of interest in the synchronized 3D world and obtain information on them.

4.1 VRML world creation

The task of modeling 3D representations of complex environments (such as city areas) is not easy and represents a research area on its own. For example, Schilling and Zipf [2003] propose a technique to generate 3D worlds in an automated way from 2D geo-data. In our case, to create a detailed VRML model of the square, we tried to re-use a pre-existing model, which we previously developed for the Web [Udine3D 2004]. The opportunity to use the same representation for different platforms is clearly an advantage of the use of the VRML language and, given the availability of many Web3D reconstructions of real places, could lower the cost of producing ad-hoc content for mobile devices.

However, we had to modify the 3D world since it was too detailed and did not allow us to obtain an acceptable frame rate on the considered PDA (we could only get 1-2fps). In particular, we simplified some of the most complex objects, removing unneeded geometries (for example, reducing the level of detail of the objects users could not see closely anyway, such as the roofs of the buildings) and deleting some elements of the representation. We also lowered the resolution of textures and, wherever possible, substituted unnecessary ones (such as the texture of some paving stones) with the use of the VRML `Material` node.

4.2 Test results

In general, users found the system easy to use because of the minimal effort needed to interact with it in GPS-based navigation mode. Moreover, users had no difficulty matching objects in the physical world with the 3D representation.

While the comments of the users about the combination of 3D graphics and the direct interaction with objects to get information were positive, some negative issues emerged during the use of the system. An important problem was the occasional low accuracy of the positioning due to poor precision of the GPS data (despite the filtering algorithms employed by the system). Therefore, in some occasions, especially with cloudy weather, the visualized 3D representation did not adequately correspond to the actual viewpoint of the user in the physical world. However, the algorithm discussed in the GPS module section managed to reduce imprecision to a tolerable level. Another related problem concerned the viewpoint: when the user is moving, her current orientation can be automatically obtained from the GPS data but when the user stops estimating orientation can be more difficult, if not impossible. To improve orientation accuracy, the best solution would be to employ an electronic compass, even if this would increase weight and size of the mobile device (unfortunately, current compact GPS units for PocketPC do not include an electronic compass).

With respect to the 3D representation, the frame rate achieved was heavily influenced by the time interval between subsequent GPS position data. Indeed, manual navigation was smoother than GPS-based navigation but the frame rate of both was much lower than desktop computers, and usually not higher than 4-5fps. Nevertheless, users found the graphic quality sufficient but they pointed out some problems which occurred in certain situations. For example, when the user is near a building, the low resolution of the textures becomes evident. Moreover, it is at the moment impossible to modify the vertical orientation of the viewpoint while navigating and it is thus difficult to easily examine a tall object when the user is too close to it.

Finally, users complained for the lack of visual indications about which objects have additional information associated: the Information Area in the User Interface gives feedback to the user (possibly a "Object without info" message) only after selecting an object.

5. Conclusion and future work

As pointed out by previous research in this area, using 3D in a mobile guide seems to be a promising direction. From the experience with the prototype described in this paper, having the possibility to actually see in a mobile device what one is looking at in the physical world and to easily request information on the objects of interest by pointing at them are features which users find useful and natural to use. Unfortunately, there are still substantial problems. Computational limitations of current mobile devices do not allow for a sophisticated use of 3D graphics and a tradeoff must be reached between performance and quality of the representation. Moreover, the precision of the positional data must be improved to provide a sufficient level of navigation assistance to users by means of 3D representations. However, it is highly likely that, in a more complex environment than the one we used (for example, inner-city streets), sometimes there would be insufficient satellite visibility to allow for an accurate GPS tracking. In those situations, alternative approaches must be used to inform the user. Our system, for example, always shows the number of available satellites and a qualitative estimate of the data accuracy, and, when no useful data is available, the user can navigate the 3D representation by using manual navigation. We feel that, at the moment, the best solution for a mobile tourist guide would be to integrate traditional procedures for navigation assistance (e.g., 2D maps) with very specific uses of 3D graphics (such as allowing users to examine objects and obtaining information on them as we did in this paper). Nevertheless, the situation is gradually improving and more powerful devices equipped with 3D accelerators will become available.

Our research is now proceeding in two directions. On one side, we aim at identifying areas (besides tourist guides) where the use of location-aware, mobile 3D graphics could be valuable. On the other side, we are developing more efficient algorithms and employing different procedures to improve the LAMP3D system. For example, to reduce GPS errors, we will try to exploit the EGNOS (European Geostationary Navigation Overlay Service) service [ESA 2004] which complements the GPS signals with a network of ground stations and geostationary satellites, improving the accuracy and giving integrity to the US positioning system.

6. Acknowledgements

Fulvio Fornasin played an important role in the implementation and testing of a preliminary version of the system described in this paper.

The work reported in this paper has been partially supported by the Italian MIUR under the project "Web-based management and representation of spatial and geographical data", subproject "User Interfaces for the Visualization of Geographical Data on Mobile Devices".

7. References

ABOWD, G.D., ATKESON, C.G., HONG, J., LONG, S., KOOPER, R., AND PINKERTON, M. 1997. Cyberguide: A Mobile Context-Aware Tour Guide. *Baltzer/ACM Wireless Networks 3*, 5, 421-433.

ARETZ, A. J., AND WICKENS, C. D. 1992. The Mental Rotation of Map Displays. *Human Performance 5*, 4, 303-328.

ARM Ltd. 2002. ARM 3D Graphics Solutions. http://www.arm.com/miscPDFs/1643.pdf

Baus, J., Cheverst, K., and Kray, C. 2004. A Survey of Map-based Mobile Guides. In Meng, L., Reichenbacher, T. and Zipf, A. (Eds), *Map-based mobile services – Theories, Methods, and Implementations*, Springer-Verlag.

Brachtl, M., Šlajs, J., and Slavík, P. 2001. PDA Based Navigation System for a 3D Environment. *Computers & Graphics 25*, 4, 627-634.

Cheverst, K., Davies, N., Mitchell, K., and Friday, A. 2000. Experiences of Developing and Deploying a Context-Aware Tourist Guide: The GUIDE Project. In *Proceedings of MobiCom 2000*, ACM Press, New York, 20-31.

Chittaro, L., and Ieronutti, L. 2004. A Visual Tool for Tracing Users' Behavior in Virtual Environments. In *Proceedings of AVI 2004*, ACM Press, New York, 40-47.

EGNOS. 2004. http://www.esa.int/esaNA/egnos.html

Ferscha, A. 2002. Contextware: Bridging Physical and Virtual Worlds. Reliable Software Technologies – AE2002, *Lecture Notes in Computer Science*, Springer-Verlag, London.

JCP. 2004. JSR 184: Mobile 3D Graphics API for J2ME. http://www.jcp.org/en/jsr/detail?id=184

KAIST. 2004. MobileGL – The Standard for Embedded 3D Graphics. http://ssl.kaist.ac.kr/ramp/

Kalkbrenner, G., and Koppe, E. 2002. Mobile Management of Local Infrastructure. *IEEE SoftCOM.*

KHRONOS. 2004. OpenGL ES – The Standard for Embedded Accelerated 3D Graphics. http://www.khronos.org/opengles/

Kulju, M., and Kaasinen, E. 2002. Route Guidance Using a 3D City Model on a Mobile Device. Workshop on Mobile Tourism Support, Mobile HCI 2002 Symposium, Pisa, Italy.

Laakso, K., Gjesdal, O., and Sulebak, J.R. 2003. Tourist information and navigation support by using 3D maps displayed on mobile devices. Workshop on Mobile Guides, Mobile HCI 2003 Symposium, Udine, Italy.

Lamberti, F., Zunino, C., Sanna, A., Fiume, A., and Maniezzo, M. 2003. An Accelerated Remote Graphics Architecture for PDAs. In *Proceedings of Web3D 2003*, ACM Press, New York, 55-61.

Marvie, J.E., and Bouatouch, K. 2004. A Vrml97-X3D Extension for Massive Scenery Management in Virtual Worlds. In *Proceedings of Web3D 2004*, ACM Press, New York, 145-153.

Narzt, W., Pomberger, G., Ferscha, A., Kolb, D., Muller, R, Wieghardt, J., Hortner, H., and Lindinger, C. 2003. Pervasive Information Acquisition for Mobile AR-Navigation Systems. In *Proceedings of WMCSA*, Wiley-IEEE Press, 13-20.

National Maritime Electronics Association. 2004. NMEA 0183 Interface Standard v3.01. http://www.nmea.org

ParallelGraphics. 2004. http://www.parallelgraphics.com/pro-ducts-/cortonace/

Pospischil, G., Kunczier, H., and Kuchar, A. 2001. Lol@: A UMTS Location Based Service. In *Proceedings of the International Symposium on 3G Infrastructure and Services*, Athens, Greece.

Rakkolainen, I., and Vainio, T. 2001. A 3D City Info for Mobile Users. *Computer & Graphics 25*, 4, 619-625.

Schilling, A., and Zipf, A. 2003. Generation of VRML City Models for Focus Based Tour Animations – Integration, Modeling and Presentation of Heterogeneous Geo-Data Sources. In *Proceedings of Web3D 2003*, ACM Press, New York, 39-47.

Superscape. 2004. Swerve Client. http://www.superscape.com/pr-oducts/swerve_client

TomTom. 2004. http://www.tomtom.com

Udine3D. 2004. http://Udine3d.uniud.it

Vossiek, M., Wiebking, L., Gulden, P., Wieghardt, J., Hoffman, C., and Heide P. 2003. Wireless Local Positioning: Concept, Solutions, Applications. *IEEE Microwave Magazine 4*, 4, 77-86.