

ARMob - Augmented Reality for Urban Mobility in RMob

Fabrizio Borgia
Université Toulouse III - Paul
Sabatier
118 route de Narbonne
31062 Toulouse CEDEX 9,
France
fabrizio.borgia@uniroma1.it

Maria De Marsico
Sapienza Università di Roma
Dipartimento di Informatica
Via Salaria 113
00198 Rome, Italy
demarsico@di.uniroma1.it

Emanuele Panizzi
Sapienza Università di Roma
Dipartimento di Informatica
Via Salaria 113
00198 Rome, Italy
panizzi@di.uniroma1.it

Lorenzo Pietrangeli
Sapienza Università di Roma
Via Salaria 113
00198 Rome, Italy
lorenzo.pietrangeli@gmail.com

ABSTRACT

This paper describes the design and development of a location-based Augmented Reality (AR) application for mobile devices. The application provides real time data about transportation in a urban area. It can be set along the line of the continuous creation of richer and more complex interaction modalities between users and data. A relevant element in this strategy is the visual enrichment of the real scene perceived through the mobile camera, by superimposing to it a set of user-relevant information. In the presented work, this information is related to nearby bus stops and to the arrival of next buses. More details, such as routes, distances etc. can be displayed on demand in order to gain awareness of the surrounding infomobility data. The presented application is included in a multiservice framework named RMob, developed for the city of Rome.

Categories and Subject Descriptors

H.3.5 [On-line Information Services]: Web-based services; H.5.2 [User Interfaces]: Graphical user interfaces (GUI)—*Augmented Reality*; H.5.2 [User Interfaces]: User-centered design; I.3.6 [Methodology and Techniques]: Interaction techniques

General Terms

Design, Human Factors

Keywords

Augmented Reality, Mobile Interaction, Android, iOS, Urban Mobility

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AVI '12, May 21-25, 2012, Capri Island, Italy

Copyright 2012 ACM 978-1-4503-1287-5/12/05 ...\$10.00.

1. INTRODUCTION

Moving within the urban jungle is a problem that modern man must face every day. Since 2003 ([2]) the definition of the term “Intelligent Transportation System” (ITS) and the development of supporting frameworks triggered new application challenges. Therefore, the citizen has one more ally in his struggle against time, and ITS assumes a critical role in improving one of people’s everyday life aspects: moving and traveling. The presented application has been designed to meet users’ needs whenever they want to plan a route via public transportation (buses) through the city.

Let’s sketch out a simple use case. Imagine a typical user - name is Kevin - who needs to travel from point A to point B in its own town. He knows where to go, but, in order to plan his way around, he requires a number of information, such as stop distances and directions, arrival times, etc.

The actions that Kevin needs to perform are the following:

- Look for buses on their way to his current position
- Identify buses passing through the desired destination
- Decide which bus to take to save time
- Reach up the proper bus stop

Kevin should be able to gather the data he needs in a short time, possibly using his smartphone or any other kind of mobile device. In this case, he should be supported by a very simple interaction modality since he will most likely use the service while moving.

This paper describes ARMob, a module that implements Augmented Reality (AR) in RMob. RMob is a mobile app for real-time information in urban transportation. ARMob has been conceived to help the users whenever they find themselves in a situation somehow similar to the one depicted in the above use case. It focuses on providing fast, clear and reliable information about the public transportation network, relieving the user from the burden of complex interaction and data input (such as keyboard usage). Thanks to an appropriate augmented reality usage, ARMob is able to allow user interaction just by camera rotation and single

taps.

ARMob is currently available for Android-based devices, and under development for iOS.

2. RELATED WORK

The increasing precision of new positioning systems (e.g., GPS for outdoor use, and WiFi for the indoor), their decreasing cost and a continuously expanding market are leading to the creation of an everyday wider and more flexible network of interconnections between data and users as well as of “intelligent” context-aware systems. Context-aware computing opens new possibilities through the use of mobile systems and, in particular, of location-based systems.

2.1 ITS - Intelligent Transportation System

The “ITS” definition includes any system implemented with the aims described in ([2]), e.g., the provision of mobility information to the user, the advanced control of the vehicle of its navigation, and the management of emergencies and accident situations. It is interesting to consider that several studies about the behavior of citizens identify the cause for rejection of the use of public transportation in the lack of information about related services, both before and during travel ([1]). In the cited work, the authors define the phases a passenger goes through during planning or taking a ride by public transport services: they are characterized as “pre-trip to destination”, “at-stop”, “on-board”, and “pre-trip to origin”. Thanks to the present technological support, information might be provided in a variety of modalities. Of course, the best, cheapest and more flexible of these is to exploit mobile phones.

2.2 Augmented Reality

In this work, AR is meant to be the real time juxtaposition of informative levels (e.g. virtual and multimedia elements, geo-located data, and so on) on a screen that displays the real scene. In addition, the Context Aware Technologies allow to include in AR scenarios specific information about the surrounding context. The “augmenting” elements of our framework, can be visualized on the display of a high performance mobile device equipped with GPS, a magnetometer (digital compass) and an accelerometer, a high speed internet connection (UMTS, HSDPA, WiFi), and with the possibility to visualize the stream coming from a built in camera. They key to exploit AR is the software. AR programs allow the developer to blend animation or contextual digital information in the display of the real world. Such programs are also known as AR browsers. ARMob sets itself in this category. After inspecting the state of the art in this field, we found out that there are many AR mobile browsers at the moment. Each one provides interesting features and cues for new user interaction modalities. Some browsers, like Layar [3] allow to develop applications within their AR engines, compensating the loss of some degrees of freedom with the simplifications of developer’s work. Other browsers, like Mixare [5], provide code to include or use in a custom-made AR application. Anyway, none of them focuses on ITS. For several reasons, we opted for developing ARMob using an AR browsing engine made by us. First of all, we preferred not to rely on code written by outer stakeholders, in order to have full control of our application and to prevent undesired behaviours. Furthermore, as already mentioned, ARMob is included the RMob multiservice framework, so we felt the

need for a highly-customized application. ARMob needs to be able to exploit the existing web-services we developed for ITS management on the server-side and to integrate itself with the other modules of the client-side application. Another reason to develop ARMob on our own is the opportunity to gradually increase its power, in order to improve its performance in the ITS context. Finally, the code we developed for ARMob can be easily adapted to other contexts where POI (Points of Interest) other than bus stops are considered.

3. ARMOB AND RMOB

ARMob needs RMob architecture in order to operate properly. RMob architecture (Figure 1) is designed like a multi-service framework to obtain a more flexible environment for a collaborative development. This architecture is divided into two layers, a central server and a number of different mobile clients, according to the client-server paradigm. The server collects ITS data from the available sources, and is designed to cache such data to improve the global performance. The most important data sources queried in real-time are currently two:

- **Atac** (the urban mobility platform developed by Rome’s transportation company) is queried to obtain data about bikesharing and carsharing parking availability, bus stops and estimated arrival times for buses.
- **Google Maps** is used to obtain maps, routes and estimated arrival times for buses.

The AR application developed in the present work exploits one of the services provided by RMob server, named “Stop-sAround”. Its features can support the “at-stop” phase, according to the definition of ([1]). This is a core class of services, since they are designed to support the users throughout their whole trip, from the starting to the ending moment. Moreover, thanks to them changes in the original plan can be handled, when delays or other accidents hinder its prosecution. After receiving a proper HTTP request from a client, the service returns an XML response containing detailed data about bus stops and incoming buses around user’s location. This work has been developed on Android-based

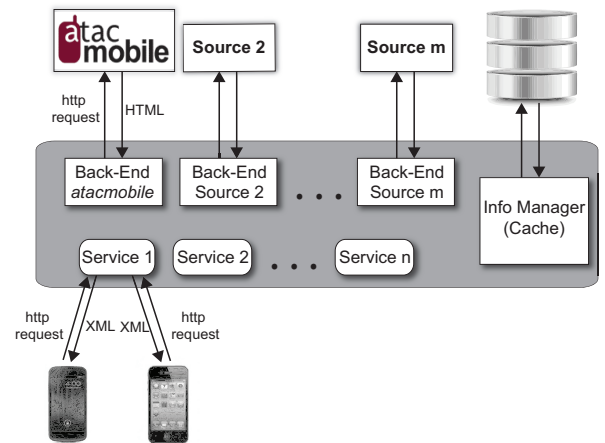


Figure 1: RMob Architecture

devices, hence it is written in Java programming language. The Integrated Development Environment (IDE) used to develop ARMob is Eclipse with an extra plug-in provided by Google named Android Development Tools (ADT). Furthermore, the localization in indoor environments has been enhanced by adding the localizations source provided by Skyhook ([8]). This source is accessible through the Skyhook Core Engine, a library that is incorporated in the ARMob application.

4. ARMOB USER INTERFACE

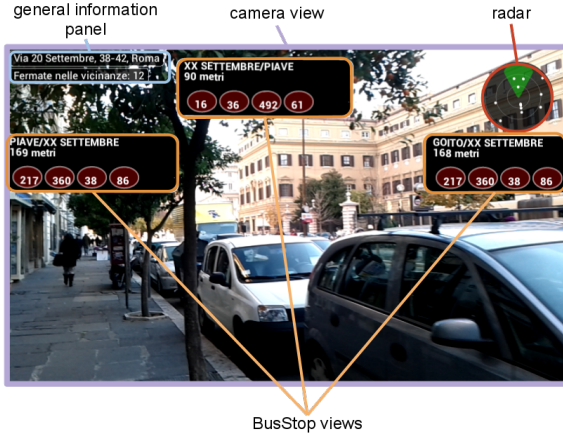


Figure 2: ARMob in street-level view modality

ARMob user interface (Figure 2) is composed by several layers, whose specific data are displayed according to the following layout:

- The **radar**: located on the upper-right part of the screen, shows an overview of nearby bus stops in a 300 mt. (customizable) range from the user; the bus stops are represented as small dots on the radar.
- The **general information panel**: located on the upper-left part of the screen, shows reverse geocoding data, i.e. addresses determined through the geographic coordinates of the device location; ARMob status notifications are also included in this layer.
- The **BusStop views** shows the following information about nearby stops: stop name, distance, and bus lines arriving there. The user can obtain details on demand by tapping on any of the displayed bus stops.
- The **camera view**: a layer showing the stream provided by the built-in camera.

The layers in ARMob interface have a strict arrangement (Figure 3). The radar (and the general information panel) is always on the topmost layer, so that the user can always have the overview of the stops in foreground. The next in the rendering order, is the bus stop view set, followed by the raw camera view.

Both the arrangement of interface layers and the layout of interface elements have been accurately chosen in order to



Figure 3: Layers arrangement in ARMob Interface

assure both usability of the application, whose interaction just requires camera rotation and single taps, and the overall readability of the provided information. Some informal usability tests helped us in this process.

In particular, five users were involved in a field experiment, where they were asked to use the system in the most natural way. Some issue arose, that will be discussed in detail in the next session. However, the main lesson learned is that occlusion may play a crucial role in the overall interface effectivity. In fact, it is not possible to anticipate when and where a layer will overlap with other interesting information, so that corrective actions must be provided to the user.

5. AUGMENTED REALITY ISSUES IN ARMOB

During the modeling phase of ARMob, a set of issues arose about user interaction the most notable of which are describe below.

5.1 Sensor noise reduction

The first issues detected was the sensor (accelerometer and magnetometer) noise, that hampered proper rendering and interaction with the bus stops on the screen. Such noise caused a heavy flickering and unpredictable rotations on the screen, making it actually impossible for the user to read and tap (or having any interaction whatsoever) with the bus stops. This problem was solved by implementing a low-pass ([4]) filter to smooth our set of data from the sensors thus stabilizing the movement of the bus stops on the screen.

5.2 Projection

Another relevant problem was related to the projection of a bus stop on the screen. In fact, it was difficult to determine the right position on the screen as a function of coordinates and altitude of the real bus stop. This problem heavily affected user interaction and thus deserved an important effort. We solved it by using the perspective projection technique ([7]) describe below. The perspective projection technique is based on three steps: A **coordinates' transformation**: Given a single bus stop with its geographic coordinates, we map its coordinates in a virtual space ob-

taining the following vector:

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} (lat_{USR}, lon_{BS}) -^* (lat_{USR}, lon_{USR}) \\ alt_{BS} - alt_{USR} \\ (lat_{BS}, lon_{USR}) -^* (lat_{USR}, lon_{USR}) \end{bmatrix}$$

where

$$lat_{USR}, lon_{USR}, alt_{USR}$$

are respectively latitude, longitude and altitude of the user's location and

$$lat_{BS}, lon_{BS}, alt_{BS}$$

are respectively latitude, longitude and altitude of the bus stop, and the

$$-^*$$

operator represents the earth distance between coordinates defined using the WGS84 ellipsoid ([6]) model. We take into account the sign (negative, positive) of the distance in order to correctly map the bus stops.

The **projection** of the bus stops on the screen: By following the simple mathematical rule underlying perspective, the virtual coordinates are projected directly toward the eye, (represented by the vector e), and they are drawn where they meet a view plane in front of the eye. The following variables are defined to describe this transformation:

$$a = (a_x, a_y, a_z)$$

$$c = (c_x, c_y, c_z)$$

$$width = camera\ width$$

$$\alpha = camera\ view\ angle$$

$$distance = \frac{\left(\frac{width}{2}\right)}{\tan\left(\frac{\alpha}{2}\right)}$$

$$\theta = (\theta_x, \theta_y, \theta_z)$$

Where vector a is the virtual coordinates vector calculated in the previous step. Vector c represents the coordinates of the camera in the virtual space. Vector θ represents the rotations angles of the camera.

Vector a needs to be rotated according to the camera coordinates and rotations angles, in the following way:

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \left(\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} - \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} \right)$$

$$\begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{bmatrix} \begin{bmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

Vector d represents (for each bus stop) the definitive virtual coordinates adapted according to the camera rotation and position.

In the end the bus stops need to be **rendered on the screen** ([7]), taking into account their distances from the camera in the following way:

$$x = distance \times \left(\frac{d_x}{d_z}\right) \quad y = distance \times \left(\frac{d_y}{d_z}\right)$$

5.3 Occlusion

When dealing with AR, it's almost impossible to avoid dealing with occlusion issues. In our case, it affected the graphical rendering of different bus stops. Two techniques were designed and adopted at the same time. The first one allows the user to choose between two possible view modalities

- The *street-level view* (e.g. if s/he is walking towards a physical bus stop)
- The *upper-levels view* (e.g. if s/he is standing in their office, house etc. at the fifth floor)

The second technique takes advantage of the geopositioning module to continuously obtain the location of the user while s/he moves towards a bus stop in order to update the perspective projection in real-time removing existing occlusions. This way, the closest bus stop will be better displayed than the ones in the distance and will always be shown in foreground. Combining the two solutions, we were able to give the user a faithful and usable representation of reality also while moving or inside a building.

6. CONCLUSIONS

ARMob focuses on bus stops, exploiting AR to inform users about nearby bus stops and arriving buses with effortless interaction and real-time update. ARMob is integrated in the RMob app that will be shortly published on the Android Market. Future work will include the recognition of approaching buses, with all issues related to tracking a moving object. Information about that particular bus can be superimposed on it.

7. ACKNOWLEDGMENTS

This work was partially funded by Sapienza University of Rome under project "iCARA: infomobilità usando un sistema context-aware con funzioni di Realtà Aumentata". We also thank ATAC and the RMob work-group.

8. REFERENCES

- [1] B. Caulfield and M. O'Mahony. An examination of the public transport information requirements of users. *IEEE Transaction On Intelligent Transportation Systems*, 8(1), 2007.
- [2] Definition of "intelligent transport system" provided by the european commission. http://ec.europa.eu/transport/its/index_en.htm.
- [3] Layar. www.layar.com.
- [4] Android developer guide. <http://developer.android.com>.
- [5] Mixare. www.mixare.org.
- [6] National Geospatial-intelligence Agency. World Geodetic System 1984 (WGS 84). <http://earth-info.nga.mil/GandG/wgs84/index.html>.
- [7] P. Shirley, M. Gleicher, S. R. Marchner, E. Reinhard, K. Sung, W. B. Thompson, and P. Willemsen. *Fundamentals of Computer Graphics 2nd Edition*. A.K. Peters, Wellesley, MA, USA, 2005.
- [8] Skyhook. www.skyhookwireless.com.