A Framework for Vision-based Mobile AR Applications

Jan Robert Menzel
RWTH Aachen University
Aachen, Germany
menzel@cs.rwth-aachen.de

Michael Königs
RWTH Aachen University
Aachen, Germany
Michael.Koenigs@rwth-aachen.de

Leif Kobbelt
RWTH Aachen University
Aachen, Germany
kobbelt@cs.wrth-aachen.de

ABSTRACT

This paper analyzes the requirements for a general purpose mobile Augmented Reality framework that supports expert as well as non-expert authors to create customized mobile AR applications. A key component is the use of image-based localization performed on a central server. It further describes an implementation of such a framework as well as an example application created in this framework to demonstrate the practicability of the described design.

Author Keywords
Augmented Reality Framework, Image-based Localization, Authoring, Location-based Gaming

ACM Classification Keywords
H5.1. Information interfaces and presentation: Artificial, augmented, and virtual realities

General Terms
Performance, Design, Experimentation

INTRODUCTION

Mobile Augmented Reality (AR) applications can be found in a wide variety of use-cases, e.g. gaming [2], learning, edutainment, tourist guides [1] and pedestrian navigation [4]. Many of these scenarios can be based on the same application architecture and middle-ware as they share the same basic requirements. Our goal is to analyze these requirements and to present a unified framework that supports experts with programming skills as well as non-expert authors who cannot program to build such kind of applications. This distinguishes our framework from previously described domain specific frameworks (e.g. [10]).

FRAMEWORK REQUIREMENTS

In this section we discuss the typical requirements of a state-of-the-art AR framework that can be used to implement different kinds of applications. First we analyze the technical aspects such as the requirements to the hardware, sensors and localization technique. Next, we discuss how the framework can support the author with the application development to leverage the possibilities of the technology.

Hardware and Sensors

Since modern smartphones are becoming more powerful and are being equipped with a wide variety of sensors a modern AR framework should only rely on these commodity devices and the integrated sensors and not require specialized hardware like previously presented systems [2,5]. This leaves us in most cases with GPS, a digital compass, accelerometer, gyroscope and a camera as available sensors.

Localization Method

GPS provides a level of accuracy that is not accurate enough for convincing AR applications and in addition only works for outdoor scenarios [3]. Cell tower localization while having the advantage of working indoors provides an even less accurate localization than GPS. The compass, gyroscope and accelerometer can only provide information about the devices relative orientation and movements but not an absolute positioning.

To provide accurate localization for indoor and outdoor scenarios, the camera can be used as an additional sensor. Image-based localization for Augmented Reality is not a new approach, but in most cases it is done with markers which have to be placed in the real environment [8]. Such modifications of the environment are not always possible as the area that should be used is too large or not controlled by the application author. Also vandalism or aesthetic reasons can prohibit the use of markers.

Marker-less image-based localization requires a previously built image database of the environment, which is easy to create with a consumer camera for small areas. For large scale scenarios, projects like Google Streetview have proved the practicability of automatic acquisition of image databases of whole cities and even countries [11].

Image-based localization implemented on the smartphone does not scale very well due to memory and CPU limitations. Hence, when supporting large areas or a high level of detail through the same localization infrastructure the required database becomes too large to be practically handled on mobile devices. Implementing the localization on a central server helps to reduce the memory and computational requirements of the client and also reduces the energy consumption. Updating the database becomes easier as well.

Outdoor scenarios are more problematic than indoor locations for image-based localization as weather, daytime and seasons have a considerable influence on the appearance and the lighting conditions. On the other hand a GPS signal can be used to narrow down the search radius outdoors when sent as additional data to the server. Indoor localization is easier as the environment is more static but no rough GPS location can be provided to the server to assist the image-based localization. We suggest to use as much additional sensor data as possible to support the server but not to rely only on these sensors.

Relying on an image-based localization server distinguishes our framework from most marker-based or GPS-based frameworks (e.g. [8,10,12]) and applications.

Application Toolbox

To handle the aforementioned different AR application scenarios, the client has to be able to display virtual objects within a live video stream, provide the possibility of interacting with those objects by selecting them and support application dependent menus (see Figure 1). With an additional internal application state modeled as a state machine, these basic building blocks are sufficient to implement location based story telling, information systems, tourist guides and even point-and-click like adventure games which motivate the user to go to real locations to interact with virtual characters and even implement a virtual inventory.

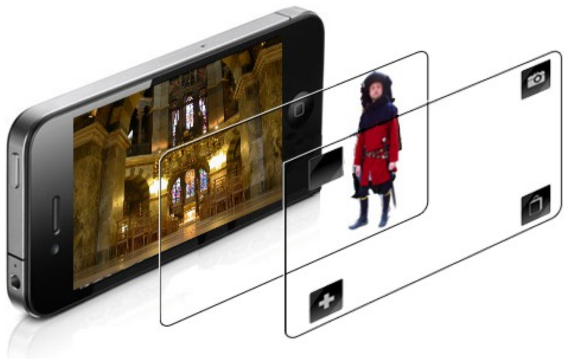


Figure 1. The AR and UI overlays of our mobile client

Application Authoring

The application framework should provide a higher level of abstraction such that even non-expert users can implement mobile AR applications and games. Experts with programming skills should be given further options to implement more specialized functionalities.

The basic logic should thus be implemented in a simple cross-platform scripting language to allow experts to implement more advanced AR applications without low level system programming. This also minimizes the OS specific parts of the client implementations which is useful in the heterogeneous smartphone market. Using the Python programming language for this purpose was suggested by Wang et al.[12], however, their approach still required programming skills from the authors. The authoring tools for non-expert users implemented as desktop applications can even include a simulator for the AR programs as the logic is implemented in platform independent scripts.

EXAMPLE FRAMEWORK OVERVIEW

We implemented a general mobile AR framework based on our requirement analysis including the following components (see Figure 2):

- A smartphone based client without additional hardware
- A remote image-based localization server
- Desktop authoring tools
- A general game logic implemented as a LUA script

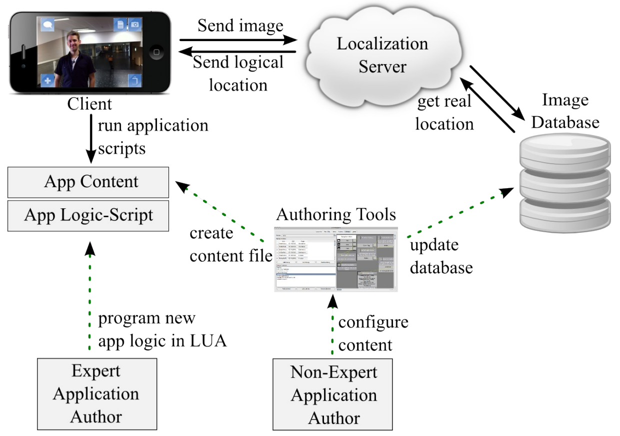


Figure 2. General framework overview: The dashed, green arrows are showing the application creation pipelines, the black solid ones the run-time data-flow.

In the following sections, we will discuss each component as we implemented them in our example framework and how they interact.

Mobile Client

The mobile client was implemented on iOS. The application logic for simple AR (gaming) scenarios using a state machine, dialog trees and point-and-click style interactions with virtual characters and objects was implemented as a LUA script. The actual application content (states of the state machine, objects, dialogs etc.) was authored with our tools (see below). More complex scenarios can be implemented by exchanging or extending this underlying logic-script without modifying the iOS application, while simpler modifications only require to change the content file which does not require any programming skills at all. The LUA scripting language was chosen as it is based on a cross-platform, open-source interpreter which does not need many resources and is thus well suited for mobile applications.

The responsibility of the client is to read sensor input (camera, compass, GPS and gyroscope), display the AR overlay rendered in OpenGL ES, run the scripting interpreter and communicate with the localization server. Since the applications are implemented in client-independent scripts, additional clients for other mobile platforms can be easily developed with a minimum of duplicated work.

Server-based Localization

The localization component in our framework was based on a localization server that manages a database of geotagged street level photos. A query image is efficiently matched against this database using an inverted index [6,9] and the original location of the highest-ranked spatially verified image is the basis for the returned location.

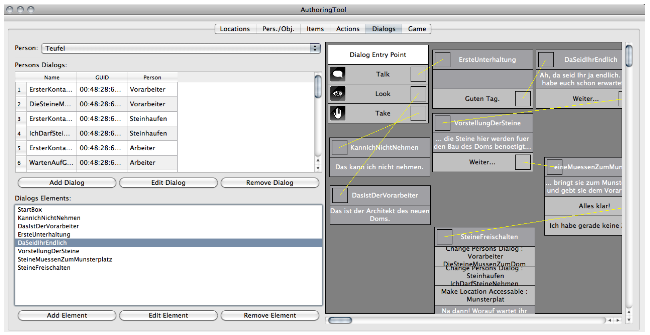
Depending on the application scenario, this location can be the real world global position or a logical application-dependent location. For example our test game required the player to find a bakery. As most bakeries belong to a small number of chains, we added the bakeries the user would most likely visit to the database, but also a number of company logos of these chains and connected all of these images to the same 'logical location'. This way we can support correct localization (from the game logic point of view) even at places the author did not expect the players to visit.

Often for location-based or Augmented Reality applications the precise orientation is also not needed but just the logical location with respect to the actual application logic (e.g. 'in front of a bakery and looking at the store' instead of a real world position). In this case the server can handle the logical location look-up for the client.

In our experiments the transmission time of the images over a mobile internet connection was the actual performance bottleneck (~4s over 3G), not the image matching on the server (~0.5s). To reduce the transmission time, we send a scaled down (640x480) JPEG compressed image. The feature extraction is handled by the server because usually the images are smaller than the compressed SIFT descriptors (61kb vs. 401kb on average, tested for 607 query images from the user study). From the features, visual words are derived to speed up the matching [9]. The data needed for transmission could be reduced by calculating the visual word assignments on the client and just transmitting those (~1.6kb). However, the memory requirements are too high for current smartphones (~500MB for the visual word quantization index alone in our test setup).

Authoring Tools

In our cross-platform authoring tools the non-expert user can add new virtual characters or objects and link them to real world locations. For each character a set of dialog trees can get created and edited by placing text boxes in a visual editor and connecting answer options to responses of the character (see Figure 3). The end nodes of a dialog tree can trigger events of the game, for example moving characters to other locations, editing the virtual inventory of the player or unlocking additional dialog trees.



Person: Teufel

Persons Dialogs:

	Name	GUID	Person
1	ErsterKonta...	00:48:28:6...	Vorarbeiter
2	DieSteineM...	00:48:28:6...	Vorarbeiter
3	ErsterKonta...	00:48:28:6...	Steinhaufen
4	IchDarfStei...	00:48:28:6...	Steinhaufen
5	ErsterKonta...	00:48:28:6...	Arbeiter
6	WartenAufG...	00:48:28:6...	Arbeiter

Dialogs Elements:

StartBox
KannIchNichtNehmen
DasIstDerVorarbeiter
ErsteUnterhaltung
DaSeidIhrEndlich
VorstellungDerSteine
SteineMuessenZumMunsterplatz
SteineFreischalten

ErsteUnterhaltung
Guten Tag.

DaSeidIhrEndlich
Ah, da seid Ihr ja endlich. habe euch schon erwarte
Weiter...

VorstellungDerSteine
... die Steine hier werden fuer den Bau des Doms benoetigt...
Weiter...

KannIchNichtNehmen
Das kann ich nicht nehmen.

DasIstDerVorarbeiter
Das ist der Architekt des neuen Doms.

eineMuessenZumMun
... bringt sie zum Munst und gebt sie dem Vorar
Alles klar!
Ich habe gerade keine

SteineFreischalten
Change Persons Dialog : Vorarbeiter DieSteineMussenZumDom
Change Persons Dialog : Steinhaufen IchDarfSteineNehmen
Make Location Accessable : Munsterplat
Na dann! Worauf wartet ihr

Figure 3. The desktop authoring tool used for managing the dialog trees.

Besides conversation, the author can also model other actions the player can perform on virtual characters or objects. For example the author can define a 'pick up', 'look at' or 'use object' action that can get connected to different events.

We also included basic authoring functionality into the mobile iOS client to extend the dialog trees with additional options on the move. The mobile authoring is only limited by the smaller screen size of the smartphone as the application content can be altered on the fly and could also be uploaded from the client to a server and thus be shared with other users.

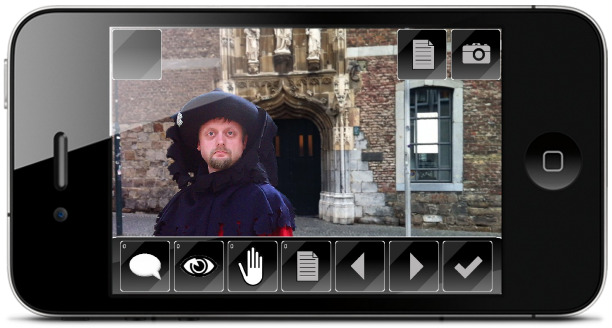


Figure 4. UI and AR view of our game: The user can choose an action (e.g. 'talk to', 'view at') and then select a virtual character or object in the AR see-thru view.

EXAMPLE APPLICATION

To test our framework, we developed a sample application that is intended as a pervasive game type tourist guide in the city center of our home town. Designed as a mixture of a classical guide and a game it has some elements of a classical 'point-and-click' adventure while being mostly linear with respect to the locations the player has to visit.

Game Description

The game story is set during the construction of the Aachen cathedral in the middle-ages. The player has to help organizing the funding for the construction of the building by talking to virtual characters (see Figure 4) and visit important historical places of the city. During the game he/she will also learn about the legend of the construction.

The design goal of the game was to guide a tourist around the city from one interesting place to the other and to tell a background story of the city though the virtual characters.

User Study

To evaluate the appropriateness of the user-interface, the localization interface and accuracy as well as the success of telling a story this way, we conducted a user study. In total 10 persons of various occupations participated in the test. In addition to the smartphone the users were also equipped with a Garmin eTrax Vista GPS that logged their movement during the study. This device was chosen because it has a more accurate GPS positioning than current phones. Our goal was to test if image-based localization can compete with GPS even assuming optimal conditions in favor of GPS as we expect the quality of smartphone GPS sensors to improve up to the quality of current dedicated GPS devices in the future. Note that in our user study we limited ourselves to just use the photo as the localization input and not to send additional information as proposed earlier. Our motivation for this was to to test the robustness of the image-based localization approach in case no GPS signal is available. This would always be the case in indoor scenarios.

While the localization worked technically quite well and was even outdoors more accurate than the GPS track recorded in parallel, we learned that users not familiar with image-based localization tend to photograph other landmarks than we had expected. As this localization technique needs a dense and overlapping coverage by photosgraphs, our initial database contained mostly whole building facades. Some users however photographed primarily street signs and similar small features for the localization. Adding images of those objects fixed all localization problems.

The overall feedback by the users was positive. While 8 participants had experiences with smartphones and also 8 participants with point-and-click adventure games, only two test players had previous experiences with AR applications. However, no one had problems interacting with the UI or the virtual characters. The presentation of a city guide in the form of an interactive AR game was well received and all participants would use this kind of tourist guide again.

CONCLUSION AND OUTLOOK

We analyzed the general requirements of a flexible AR framework and proposed a set of techniques and tools to meet these. A sample application based on this framework demonstrated that our goal was met from a technical point of view. A user study showed that the resulting game, user interaction methods and localization techniques work well both from a technical, as well as a usability point of view.

The presented framework proved to be flexible enough to be the basis for different applications. Using our tools, it was possible to create a basic prototype of a campus guide / interactive information system within one day including the creation of an image database for localization.

Vision based localization proved to be robust and accurate enough for general location-based and AR applications. We used the localization server not only to detect actual positions, but also classes of locations based on shared features in the form of very similar looking bakeries of the same company. This idea can be extended to add object detection to the server and provide localization as well as object detection within the same framework.

Placing the localization component on a remote server allows for easy updates of the database as well as the matching technique: In later tests with our framework the localization server was replaced by an implementation of Sattler et al.[7] using the same client-server interface, so no changes were needed on the iOS client. This technique gave us a median localization accuracy of less than 1.3 meters.

REFERENCES

1. Bay, H., Fasel, B., Gool L.: Interactive Museum Guide: Fast and Robust Recognition of Museum *Objects. In Proc. Int. Workshop on Mobile Vision, 2006*
2. Herbst, I. Braun A. McCall R. Broll W: Time-Warp: Interactive Time Travel with a Mobile Mixed Reality Game. *Mobile HCI, 2008*
3. Jacob, J., Coelho, A.: Issues in the development of location-based games *International Journal of Computer Games Technology, 2011*
4. Li, M., Mahnkopf, L, Kobbelt, L.: The Design of a Segway AR-Tactile Navigation System. *Pervasive, 2012*
5. *Pac-Manhattan - http://www.pacmanhattan.com/*
6. Philbin J., Chum O., Isard M., Sicic J., Zisserman A.: Object Retrieval with Large Vocabularies and Fast Spatial Matching. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2007*
7. Sattler T., Leibe, B. Kobbelt, L.: Fast Image-Based Localization using Direct 2D-to-3D Matching. *ICCV,2011*
8. Schmalstieg D., Wagner D.: Mobile Phones as a Platform for Augmented Reality. *Proceedings of the IEEE VR 2008 Workshop on Software Engineering and Architectures for Realtime Interactive Systems, 2008*
9. Sivic, J., Zissermann, A.: Video Google: A Text Retrieval Approach to Object Matching in Videos. *ICCV, 2003*
10. Tutzschke J, Zukunft O.: FRAP: A Framework for Pervasive Games. *EICS, 2009*
11. Vincent, L.: Taking Online Maps Down To Street Level. *IEEE Computer 40(12): 118-120, 2007*
12. Wang, Y., Langlotz, T., Billinghurst, M., Bell, T.: An Authoring Tool for Mobile Phone AR Environments. *In Proc. New Zealand Computer Science Research Student Conference, 2009*