Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 94 (2016) 97 – 104

Procedia Computer Science

The 13th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2016)

Advanced Algorithms for Location-Based Smart Mobile Augmented Reality Applications

Rüdiger Pryss[a,*], Philip Geiger[a], Marc Schickler[a], Johannes Schobel[a], Manfred Reichert[a]

[a] *Ulm University, Institute of Databases and Information Systems, James-Franck-Ring, Ulm, 89081, Germany*

Abstract

During the last years, the computational capabilities of smart mobile devices have been continuously improved by hardware vendors, raising new opportunities for mobile application engineers. Mobile augmented reality is one scenario demonstrating that smart mobile applications are becoming increasingly mature. In the AREA (Augmented Reality Engine Application) project, we developed a kernel that enables such location-based mobile augmented reality applications. On top of the kernel, mobile application developers can easily realize their individual applications. The kernel, in turn, focuses on robustness and high performance. In addition, it provides a flexible architecture that fosters the development of individual location-based mobile augmented reality applications. In the first stage of the project, the LocationView concept was developed as the core for realizing the kernel algorithms. This LocationView concept has proven its usefulness in the context of various applications, running on iOS, Android, or Windows Phone. Due to the further evolution of computational capabilities on one hand and emerging demands of location-based mobile applications on the other, we developed a new kernel concept. In particular, the new kernel allows for handling points of interests (POI) clusters or enables the use of tracks. These changes required new concepts presented in this paper. To demonstrate the applicability of our kernel, we apply it in the context of various mobile applications. As a result, mobile augmented reality applications could be run on present mobile operating systems and be effectively realized by engineers utilizing our approach. We regard such applications as a good example for using mobile computational capabilities efficiently in order to support mobile users in everyday life more properly.

Peer-review under responsibility of the Conference Program Chairs

Keywords: Mobile Augmented Reality; Location-based Algorithms; Mobile Application Engineering; Augmented Reality

1. Introduction

The proliferation of smart mobile devices on one hand and their computational capabilities on the other have enabled new kinds of mobile applications. So-called millenials, people born after 1980, pose demanding requirements with respect to the use of mobile technology in everyday life. Amongst others, they want to be assisted by mobile technology in their leisure time. For example, when walking around in Rome with its numerous ancient spots, the

* Corresponding author. Tel.: +49-731-5024136 ; fax: +49-731-5024134.
E-mail address: ruediger.pryss@uni-ulm.de

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.08.017

smart mobile device shall provide related information about these spots in an intuitive way. In such a scenario, location-based mobile augmented reality is useful, e.g., if a user is located in front of the St. Peter's Basilica, holding his smart mobile device with its camera switched on towards the Basilica, the camera view shall provide additional information (e.g., worship time).

Our developed AREA kernel supports these scenarios. More precisely, AREA detects predefined *points of interest* (POIs) within the camera view of a smart mobile device, positions them correctly, and provides additional information on the detected POIs. This additional information, in turn, is interactively provided to the user. For this purpose, the user touches on the detected POIs and then obtains further information. Three technical issues were crucial when developing AREA. First, POIs must be correctly displayed even if the device is hold obliquely. Depending on the attitude of the device, the POIs then may have to be rotated with a certain angle and moved relatively to the rotation. Second, the approach to display POIs correctly must be provided efficiently to the user. To be more precise, even if multiple POIs are detected, the kernel shall display them without any delay. Third, the POI concept shall be integrated with common mobile operating systems (i.e., iOS, Android, and Windows Phone). To tackle these challenges, the *LocationView* concept was developed. Additionally, an architecture was designed, which shall enable the quick development of location-based mobile augmented applications on top of the kernel [1,2,3].

The AREA project was started four years ago. Already one year after the first kernel version had been finished, it was integrated with various mobile applications. During these projects, three issues emerged that are not properly covered by the AREA kernel so far. First, the changed characteristics of the used mobile operating systems need to be addressed. Second, the continuously growing number of POIs must be handled more efficiently. Third, new features were demanded. Table 1 summarizes the evolution of AREA from its first to its second version.

Table 1. AREA Versions

	AREA Version 1 (AREA)	**AREA Version 2 (AREAv2)**
Android App	√	√
iOS App	√	√
Windows Phone App	√	*under development*
POI Algorithm (all mobile OS)	*LocationView Algorithm* [1,2,3]	*RenderingPipeline Algorithm*
Clustering Algorithm (all mobile OS)	*X*	*AREACluster Algorithm*
POI Coordinate System (all mobile OS)	*GPS*	*GPS, ECEF, ENU, Virtual3D*
Position Change Sensors (all mobile OS)	*SensorFusion(Compass, Accelerometer)*	*SensorFusion(Compass, Gyroscope, Accelerometer)*
Architecture (all mobile OS)	*Version* 1	*Version* 2
Overall Sensor Management Android	*Own approach*	*Own approach*
Overall Sensor Management iOS	*Own approach*	*Built – in functions OS*
Overall Sensor Management Windows Phone	*Own approach*	*under development*

ENU=East-North-Up Coordinate System, ECEF=Earth-Centered Earth-Fixed Coordinate System, GPS=Global Positioning System

The changed characteristics of mobile operating systems as well as performance issues with many POIs are addressed by the development of a new kernel and architecture called AREA Version 2 (AREAv2) (cf. Table 1, *AREAv2*). Moreover, AREAv2 provides three new features. The first one deals with so-called POI clusters. If a huge number of POIs causes many overlaps on the camera view, it is difficult for users to precisely interact with single POIs inside such cluster. In order to precisely select (i.e., do not touch the wrong POI as they are positioned to close to each other) a single POIs inside a cluster, a new feature was developed. The second feature we developed connects POIs through lines in order to visualize tracks. For example, such a track may be used as the cycle path a user wants to perform in a certain area. The third feature highlights areas (e.g., football fields). From a technical perspective, the added features are demanding if they shall be supported in the same manner on different mobile operating systems.

This work presents fundamental concepts developed in the context of AREA Version 2 (AREAv2). Section 2 presents the architecture of AREAv2. In Section 3, the coordinate system used by AREAv2 is introduced, while Sections 4 and 5 present the algorithms for POI and cluster handling. Section 6 illustrates the use of AREAv2 in practical scenarios. Section 7 discusses related work and Section 8 concludes the paper.

2. Architecture

The architecure used for AREAv2 is shown in Fig. 1. It enhances the one used in the first version of AREA [1,2,3] and comprises **nine** major components (cf. Fig. 1).