Mobile Location-Based Augmented Reality Framework

Simon Burkard[1], Frank Fuchs-Kittowski[1,2(✉)], Sebastian Himberger[1],
Fabian Fischer[1], and Stefan Pfennigschmidt[2]

[1] HTW Berlin, Wilhelminenhofstr. 75a, 12459 Berlin, Germany
{S.Burkard,Frank.Fuchs-Kittowski,s.himberger,
fabian.fischer}@htw-berlin.de
[2] Fraunhofer FOKUS, CC ESPRI, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany
Stefan.Pfennigschmidt@fokus.fraunhofer.de

Abstract. GeoAR, or location-based augmented reality, can be used as an innovative representation of location-specific information in diverse applications. However, there are hardly any software development kits (SDKs) that can be effectively used by developers, as important functionality and customisation options are generally missing. This article presents the concept, implementation and example applications of a framework, or GeoAR SDK, that integrates the core functionality of location-based AR and enables developers to implement customised and highly adaptable mobile application with GeoAR.

Keywords: Augmented reality · Location-based AR · Geo-based AR
AR-SDK

1 Introduction

The mass distribution of powerful and easy-to-use mobile devices (smartphones, tablets, etc.) has led to the increased availability and use of location-based services. While location-specific information on mobile devices is often displayed on maps or in lists, an innovative user interface consists of information displayed as an augmented reality (AR) layer over the camera image of the mobile device.

The term "augmented reality" refers to the supplementation of the human visual perception of reality with digital, context-dependent information [1]. In mobile augmented reality (mAR), the image from the camera of mobile devices is used to extend the real, local environment of the user by displaying additional digital information in real time [2]. The tracking method employed to acquire the position and viewing direction of the user (pose) can be used to differentiate between two forms of mAR: in the geo-based approach (also known as location-based AR, or GeoAR), the pose is determined using the built-in GPS sensors and inertial measurement unit (IMU) of the smartphone. In the image-based approach (vision-based AR), the pose as well as objects in the vicinity of the user are identified using optical tracking methods [3].

J. Hřebíček et al. (Eds.): ISESS 2017, IFIP AICT 507, pp. 470–483, 2017.
https://doi.org/10.1007/978-3-319-89935-0_39

Mobile augmented reality has great economic potential. In spite of this, there has been little research and development in the area of location-based AR in recent years. The first available GeoAR software development kits (SDKs) such as Wikitude, Layar, and Metaio, have only a rudimentary range of functionality with few customisation options, or have disappeared completely from the market [4]. Instead, commercial companies as well as researchers have focussed on the development and improvement of vision-based AR (SLAM/3D tracking) in order to achieve the most exact positioning [5], but mobile vision-based AR is still not ready for the mass market. However, precise positioning is not necessary for many users and areas of application. The disadvantages of location-based AR compared to vision-based AR are therefore acceptable in many cases. The current example of Pokémon Go demonstrates that AR technology that is not based on complex image-based recognition methods can be very successful. However, there are currently no established GeoAR SDKs on the market for the implementation of custom applications without the need for expert knowledge in the areas of AR and computer vision.

In this article, a GeoAR SDK will be presented that supports the custom development of a wide range of GeoAR applications and simplifies and accelerates the development process. The framework is aimed primarily at experienced app developers who wish to create location-based AR applications with their own concrete ideas of functionality and design, but who do not wish to have to acquire expert knowledge in computer vision and AR in order to do so.

The article is structured as follows: starting with a brief introduction to mAR, the functionality of location-based AR is discussed in more detail (Sect. 2). A distinction is made between location-based and image-based AR, and both technologies are evaluated with regard to robustness and accuracy. Following this, common applications of mobile GeoAR are presented and on their basis, general functional requirements necessary for a wide range of GeoAR applications are derived (Sect. 3). Comparing these requirements with existing mobile GeoAR SDKs (Sect. 4), it is found that current SDKs lack certain functionality, and there is a market potential for a novel framework. The concept of such a novel GeoAR framework is detailed in Sect. 5. Finally, two mAR applications that have been implemented with the framework are presented in Sect. 6.

2 Mobile Location-Based Augmented Reality

In this section, a brief introduction to mobile augmented reality is given, followed by a discussion of the functionality of location-based AR.

2.1 Augmented Reality

The term “augmented reality” refers to the enrichment of the human perception of reality with additional, context-dependent, digital information in real-time [1]. The user of an AR application is presented with supplementary, virtual information within his/her field of view. This data has a fixed spatial relationship with objects in the real world. For example, computer-generated content can be superimposed on an image of the real world from the camera. This idea is already well established in many areas of everyday life, e.g., in TV broadcasts of soccer matches in which virtual off-side lines or virtual distance measurements are displayed over the real-life field (Fig. 1).

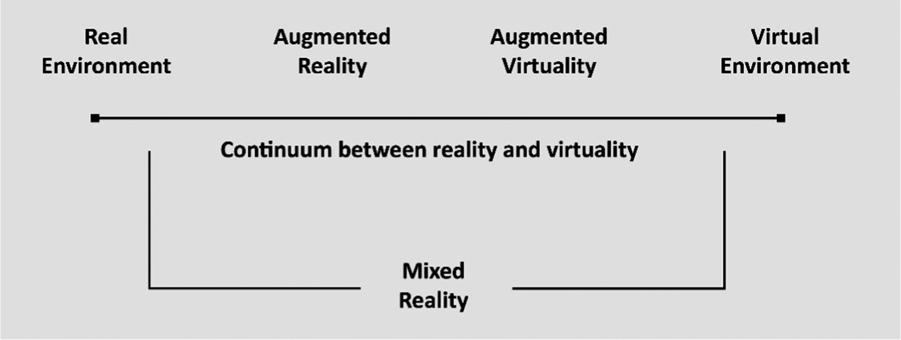


Fig. 1. Reality-virtuality continuum [6]

Augmented reality defines a reality-virtuality continuum. At either end stands complete reality or complete virtuality. In between the extremes lies the realm of mixed reality, which is characterised by different degrees of virtuality. In pure virtual environments, virtual reality (VR), the real surroundings are completely replaced by virtual ones and the user is totally immersed in the virtual world. On the other hand, the representation of additional information is at the forefront of AR; it is merely a supplement to the real world. Although the user must enter ("dive into") the virtual world and therefore interrupt contact with the real world, he/she continues to perceive the real surroundings. The real and virtual world, perceived simultaneously by the user, are unified.

2.2 Mobile Augmented Reality

In recent years, AR technology has become increasingly relevant in the context of mobile devices. In mobile augmented reality, mobile devices are used to merge the real and digital worlds in order to facilitate the perception of real and digital information on the local surroundings [2].

The real world is viewed through the camera of a mobile device (e.g., a smartphone) and is supplemented in real time with location-specific computer-generated content. The digital, geocoded information is put in spatial context and the view of the world is enriched by its presence. This enables a new way of perceiving places by presenting information from the past, present, or future, e.g., the representation of a building that is no longer/not yet visible. It also allows a better understanding and analysis of digital data on-site, allowing for better decision-making.

For a long time, mAR was a form of fundamental research with only a few expensive specialised applications for the few experts in the field. Today, modern mobile devices are a suitable hardware platform for mAR due to their high performance abilities. In particular, the most important sensors for implementing mAR are already integrated into smartphones. In addition to the camera for recording images, these include the IMU for determining the orientation (rotation) of the device as well as a GPS receiver for roughly determining the position. Moreover, these powerful mobile devices are widespread, user-friendly, and affordable so that mass usage of mAR applications is possible for everyone (workers, citizens, etc.).

In the past several years, mAR applications have been developed for various purposes [7, 8], e.g., in Tourism [9] (e.g., for displaying nearby hotels or tourist attractions), medicine [10], education [11], culture/museums [12], advertising/marketing [13, 14] (e.g., for visualising furniture in one's own home in 3D), and entertainment [15] (e.g., Pokémon Go). An example of some of these applications is shown in Fig. 2.

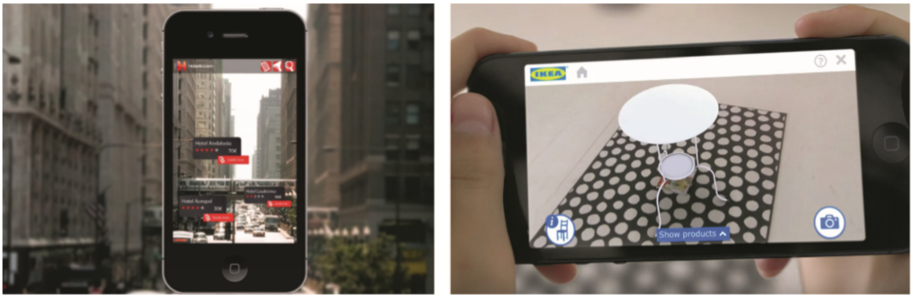


Fig. 2. Typical applications of mobile augmented reality: display of nearby points of interest (location-based AR) and visualisation of virtual furniture (image-based AR).

Due to the simple nature and widespread availability of mobile devices, the technical basis of mAR, and the numerous potential applications in many different fields, mAR has great economic potential [16]. Market research companies predict strong growth in this area in the coming years. For example, Juniper Research estimates the market will be worth over US$6 billion in 2021 [17].

2.3 Geo-Based (Location-Based) and Image-Based AR

In the implementation of mAR applications, it is fundamentally possible to differentiate between two different technologies based on the method employed by the device to determine its own position in 3D space. Establishing the position of the mobile device is an essential requirement for the implementation of AR, as information corresponding to objects in the field of view can only be positioned correctly on the screen given a knowledge of the camera position and projection. The position and pointing of the camera in 3D space is described by six degrees of freedom: three degrees of freedom for the orientation (rotation) and three for the position (translation).

With location-based AR (GeoAR), the rotation is determined solely by the IMU of the device, i.e., by a combination of readings from the gyroscope, accelerometer, and magnetometer. The GPS signal is used to roughly fix the location. This AR technology is based on established, robust, simple technology that is not very CPU-intensive, but it is primarily suitable for outdoor applications and is problematic inside buildings where no GPS signal is available. Additionally, the IMU sensors tend to "drift" during the rotation determination and often the viewing direction cannot be calculated exactly due to local disturbances in the global magnetic field [18].

In contrast, image-based AR calculates the position of the camera based solely on an analysis of the camera image using image processing techniques. By recognising prominent points in the video feed from the camera (markers or natural feature tracking), both the rotation and translation of the camera relative to its environment can be determined [19]. Very complex and CPU-intensive algorithms are necessary to interpret and analyse the images. However, in ideal conditions, this method can facilitate very accurate and realistic AR overlays. A major challenge is to cope with external influences and produce acceptable results even with poor lighting conditions (e.g., in darkness), moving objects in the images, or featureless surroundings. Another disadvantage is that this technology does not scale in large environments, such as streets, and does not scale well to various distances to objects. This AR technology is more suited to applications relating to the direct environment of the device, e.g., displaying virtual information about objects in the immediate vicinity (see Fig. 2).

The robustness of image-based technology can be further improved by creating 3D models of the environment (model-based AR). The user can then pinpoint him/herself within the virtual model at runtime and further extend the model (SLAM) [20]. However, other sensors that are not yet built into commercially available smartphones (e.g., infrared sensors for depth measurement or a second camera for stereo vision) are typically required to do this.

Table 1 summarises the above-mentioned key characteristics that determine the reliability and utility of mobile augmented reality approaches (and provides quantitative information from [29]). These key characteristics are: (1) *localisation*, which determines the users' viewpoint in order to derives real world objects in the current scene and to display the relevant digital information in the correct position, (2) *speed* of determining the users' position, the relevant information and the visualisation of the information in the correct position, (3) *robustness*, like dependence on external infrastructure or battery consumption, and the ability to work with dynamically changing environments, (4) *scalability*, like scaling to larger areas, number of objects, and size of objects.

Table 1. Comparison of mobile augmented reality approaches

Metrics	Location-based	Image-based	Model-based
Localization accuracy	1.5–35 m	0.5–2 mm	0.5–20 mm
Localisation area	Large (GPS area)	3 m (markers)	10 m (objects)
Localisation speed	100–200 ms	20–140 ms	5–240 s
External infrastructure	GPS satellite	Optical markers	External sensors, model
Resistant to drifts	No	Yes	Yes
CPU/battery consumption	Low	High	Very high
Scale to large scenes	Yes (outdoor)	No (room)	No (building)

Overall, however, it seems that image-based AR is not ready for the mass market due to its high error rate and lack of robustness. Image-based AR can - under ideal conditions - allow very accurate and realistic AR visualisations, but it is very error-prone.

On the other hand, the disadvantages of location-based AR technologies (inaccurate positioning/orientation) are acceptable for many (outdoor) applications. These general advantages and disadvantages of the AR technologies are roughly sketched in Fig. 3.

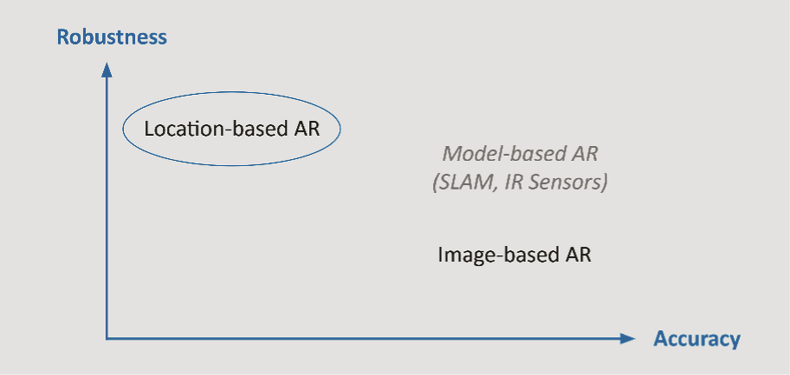


Fig. 3. Comparison of robustness and accuracy of mAR technologies

3 Applications of Geo-Based AR

Based on the typical usage of location-based AR applications, general functional requirements of an SDK for developing custom AR applications are derived in the following.

In the classical mAR applications, nearby points of interest (POIs) are displayed as markers in the camera overlay. However, location-specific information may not always be just simple markers. Complex 3D models referenced with a geographic coordinate or objects defined by several connected coordinates can be displayed as polylines or polygons. Considering even more complex uses for mAR, four different types of applications can be defined (cf., [21], see Fig. 4).

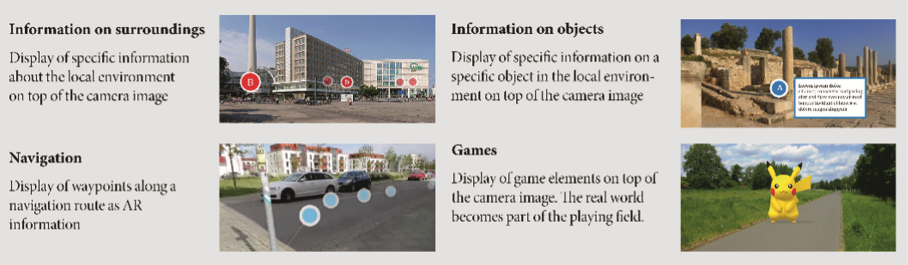

Information on surroundings
Display of specific information about the local environment on top of the camera image
Information on objects
Display of specific information on a specific object in the local environment on top of the camera image
Navigation
Display of waypoints along a navigation route as AR information
Games
Display of game elements on top of the camera image. The real world becomes part of the playing field.

Fig. 4. Types of applications of location-based AR

Area Information. Specific information about the user's environment is displayed as additional AR information in the camera image. The information is displayed when relevant objects are within the field of view of the camera, e.g., tourist attractions, petrol stations, or rivers. Georeferenced data can be presented in very different formats on the image, for example, individual POIs can be displayed as icons (e.g., hotels or tourist attractions), polylines (e.g., rivers), polygons (e.g., flood risk maps), or georeferenced 3D objects (e.g., wind turbines). As an example, the GewässerBB app shows rivers as blue polylines for which hydrological information is available [22].

Object Information. Specific information on a particular object in the immediate environment can visualised as additional AR information, for example, details of building facades, exhibits in an open-air museum, an excavation site, or a product. The content is displayed by superimposing the virtual information on the image of the object when the user points the device at a particular object. For example, certain details, on which further information is available, are highlighted in the AR view of the historical House of Olbrich [23].

Navigation. AR navigation can be seen as another type of application. While the user proceeds towards a given destination, georeferenced waypoints (or arrows) as well as simple navigational information are shown in the camera image along a route. This provides an alternative form of displaying the information to the classic map-based navigation. An example of such navigation, which also provides the user with ecological information in order to promote sustainable living, is given in [24].

Games. maR games present a game as an overlay on the camera view of the real world. Location-specific game elements are represented as AR objects. The playing field is thereby an extension of the real world. Known examples of mAR games are Pokémon Go or Ghost Hunter. In Ghost Hunter, the player hunts ghosts which come through the walls of the room. A ghost-hunter gun, with which the player can target and shoot the ghosts, is displayed in the camera image [25].

A large number of different functionalities are required from a developer's perspective in order to individually create such applications. Based on an analysis of the typical (aforementioned) geoAR application types (object information, environmental context information, navigation, and gaming) as well our own existing geoAR applications in the context of

- **flood management** (e.g. flood risk maps, current water levels and flood warnings, historical flood markers, and an educational flood trail) developed in several projects like MAGUN [22] and
- **urban disaster and safety management** (e.g. incident reporting, navigation in disaster situation) developed in several projects like City.Risks,

the following general functional requirements were derived for an SDK envisioned for the development of the widest possible range of applications:

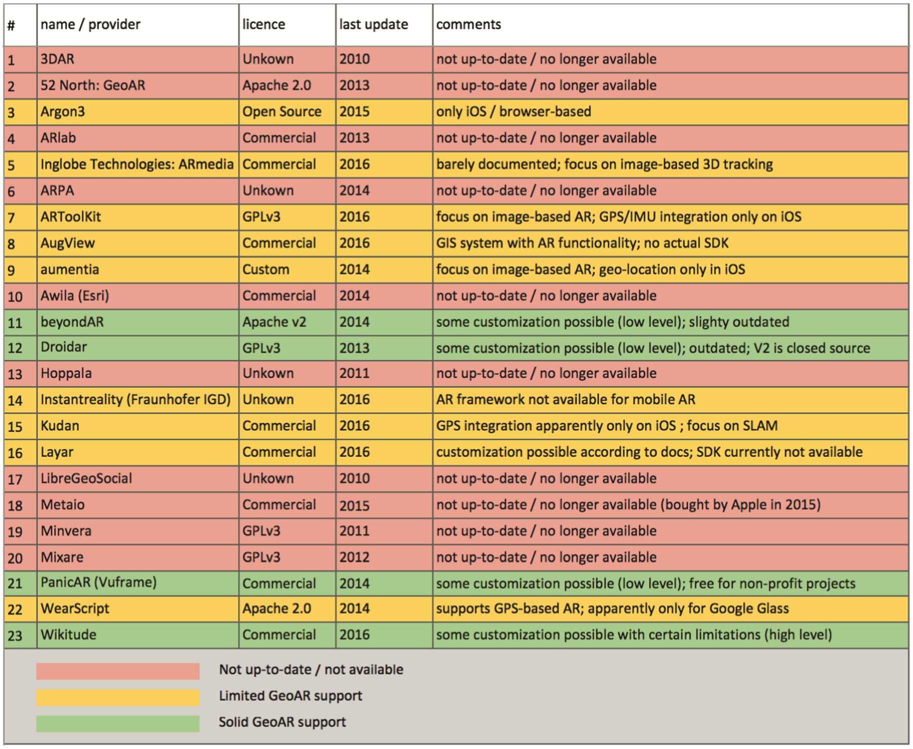
- **Presentation of spatial objects with one geographic reference (POI, 3D model) or several geographic references (polyline, polygon).** Location-specific objects with a single coordinate (3D model, POI) or several coordinates (polyline, polygon) must be displayed in the correct position on the screen.
- **Dynamic creation of adaptable content.** It must be possible to dynamically generate and flexibly scale the AR content. This means that it must be possible to

create and delete objects at runtime. The objects themselves must be as customisable as possible so that the design and appearance of the AR content (size, colour, icon style, etc.) is determined by the user context.

- **User interaction with AR objects.** The developer should have control over certain user events. These events include, for example, instances when an object is clicked or when objects appear in or disappear from the camera focus.
- **Access to camera controls and picture.** Access to the camera is desirable in order to control the camera or gain access to the camera image, for example, to process or store the current camera image.

4 SDKs for Location-Based AR

There are a number of AR SDKs available that are designed to facilitate and accelerate the implementation of AR applications [26]. In a market analysis, existing SDKs were systematically studied with regard to their functionality and possible applications in order to determine the extent to which the functional requirements identified above are fulfilled (Fig. 5).



#	name / provider	licence	last update	comments
1	3DAR	Unkown	2010	not up-to-date / no longer available
2	52 North: GeoAR	Apache 2.0	2013	not up-to-date / no longer available
3	Argon3	Open Source	2015	only iOS / browser-based
4	ARlab	Commercial	2013	not up-to-date / no longer available
5	Inglobe Technologies: ARmedia	Commercial	2016	barely documented; focus on image-based 3D tracking
6	ARPA	Unkown	2014	not up-to-date / no longer available
7	ARToolKit	GPLv3	2016	focus on image-based AR; GPS/IMU integration only on iOS
8	AugView	Commercial	2016	GIS system with AR functionality; no actual SDK
9	aumentia	Custom	2014	focus on image-based AR; geo-location only in iOS
10	Awila (Esri)	Commercial	2014	not up-to-date / no longer available
11	beyondAR	Apache v2	2014	some customization possible (low level); slighty outdated
12	Droidar	GPLv3	2013	some customization possible (low level); outdated; V2 is closed source
13	Hoppala	Unkown	2011	not up-to-date / no longer available
14	Instantreality (Fraunhofer IGD)	Unkown	2016	AR framework not available for mobile AR
15	Kudan	Commercial	2016	GPS integration apparently only on iOS ; focus on SLAM
16	Layar	Commercial	2016	customization possible according to docs; SDK currently not available
17	LibreGeoSocial	Unkown	2010	not up-to-date / no longer available
18	Metaio	Commercial	2015	not up-to-date / no longer available (bought by Apple in 2015)
19	Minvera	GPLv3	2011	not up-to-date / no longer available
20	Mixare	GPLv3	2012	not up-to-date / no longer available
21	PanicAR (Vuframe)	Commercial	2014	some customization possible (low level); free for non-profit projects
22	WearScript	Apache 2.0	2014	supports GPS-based AR; apparently only for Google Glass
23	Wikitude	Commercial	2016	some customization possible with certain limitations (high level)

🟥 Not up-to-date / not available
🟨 Limited GeoAR support
🟩 Solid GeoAR support

Fig. 5. Overview of available SDKs with GeoAR support

In a comprehensive literature analysis, relevant AR SDKs were identified and selected using search engines, link collections (e.g., [27, 28]), and established scientific literature databases (e.g., ACM Digital Library, IEEE Xplore Digital Library, Science-Direct, SpringerLink). The SDKs found in this search were subsequently examined and evaluated with respect to the functionality listed in the previous section as well as non-functional requirements (such as supported programming languages and platforms, available licences and licence cost, documentation, and current status).

Approximately 50 mAR SDKs were identified in this way. Of these, about 20 explicitly support location-based AR technology, facilitating mAR using GPS-based positioning and georeferenced content (GeoAR SDK). However, it is clear that most of the GeoAR SDKs are highly outdated or even officially discontinued or bought out (e.g. Metaio, Vuforia). This confirms the trend that many development studios and research companies are currently focusing purely on image-based tracking methods for AR solutions.

The few available, up-to-date GeoAR SDKs (Wikitude, DriodAR, PanicAR, and beyondAR) only support some of the aforementioned functionality. The depiction of objects with a single geographic coordinate (POIs, 3D models) is supported as a classic use case by all SDKs. However, the visualisation of geographic objects that are defined by a collection of coordinates (e.g., a river as a polyline) cannot be directly implemented with the available SDKs.

Customisation of the appearance of AR objects is also not generally possible to the desired extent. Thus, the developer is often bound to the design specifications of the SDK. Furthermore, access to the two-dimensional screen coordinates of the rendered objects is usually restricted, meaning that objects cannot be dynamically expanded or superimposed with their own additional content. Similarly, access to the underlying camera image in high resolution is generally restricted, making it impossible to further process or save the image.

Overall, there are very few usable SDKs available for location-based AR applications. Existing SDKs are either out-of-date or only offer limited functionality with few options for customisation such that they do not support the development of a broad range of custom mAR applications.

5 Concept of a Mobile Location-Based AR Framework

The weaknesses of the existing GeoAR SDKs reveal the need for a new framework that offers more extensive functionality and possibilities for customisation. The concept of such a framework is presented in the following.

The framework is intended to facilitate and speed up the development of as many of the aforementioned applications as possible while still allowing the developers as much freedom as possible to implement their individual ideas with regard to appearance and functionality. The framework is therefore designed as a low-level framework with the intention of internally performing the mathematical and core technological functions of location-based AR that would otherwise require expertise in the field of computer vision. It is suitable for app developers who have experience in writing mobile applications but who do not wish to have to acquire expertise in the fundamental mechanics of AR technology. Figure 6 shows the basic concept of the framework.

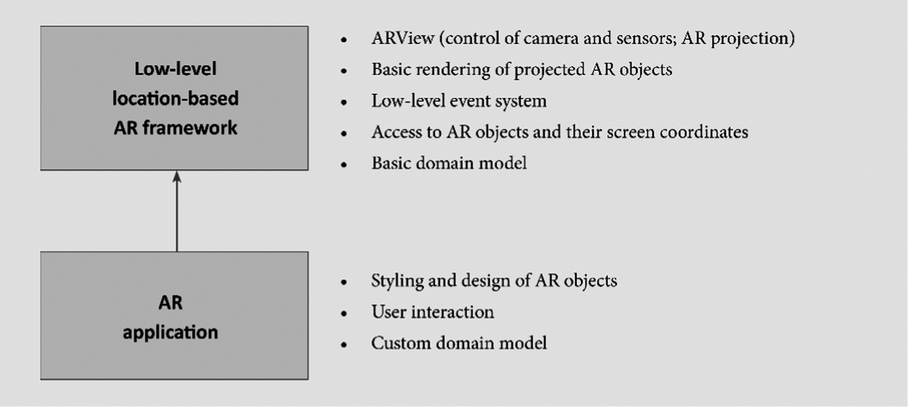

Low-level location-based AR framework
- ARView (control of camera and sensors; AR projection)
- Basic rendering of projected AR objects
- Low-level event system
- Access to AR objects and their screen coordinates
- Basic domain model
AR application
- Styling and design of AR objects
- User interaction
- Custom domain model

Fig. 6. Concept of a low-level location-based AR framework

A central part of the framework is an AR view (GeoARView) with its own lifecycle that displays the camera image as well as the overlayed, rendered AR objects. This core component consists of several layers which take care of internal functionalities like initialising and displaying the camera video stream (CameraLayer) or displaying AR objects (ARLayer) as well as user interface elements (e.g. radar and debug views). AR specific calculations and processes are encapsulated in separate classes, i.e. the conversion of three-dimensional geographic coordinates into corresponding two-dimensional screen coordinates on the basis of the current position and rotation of the device using the intrinsic camera projection matrix (e.g. ProjectionTranslation) or the rendering of an AR object (e.g. ARObjectRenderer). These classes are parts of the ARLayer.

Furthermore, an underlying model is created for the GeoAR objects to be displayed on the screen. This can either be represented by a single geographic reference point (POI) or by a list of connected reference points (polyline, polygon). Individual geographic points are initialised with their geographic coordinates, i.e., longitude, latitude, and height above sea level (optional). The appearance and design of these points on the screen (icon style, colour, size, transparency, etc.) can be changed at any time. Their appearance can also be adjusted according to context, for example, changing the transparency of a point depending on its distance from the user. Access to the distance between the user and the respective objects, as well as the 2D screen coordinates of the rendered objects, is always possible. The user of the framework has full control to the styling by implementing their own custom renderer.

A simple event model can be used to react to user interaction. Listeners exist to react when, for example, the device is rotated and new objects become visible or objects disappear out of the field of view. Additionally, all objects can be found within a definable region of the screen. This allows a reaction to click events from the user, for example. The listeners can also access the current feed from the camera in full resolution. Figure 7 shows the class diagram of the framework.

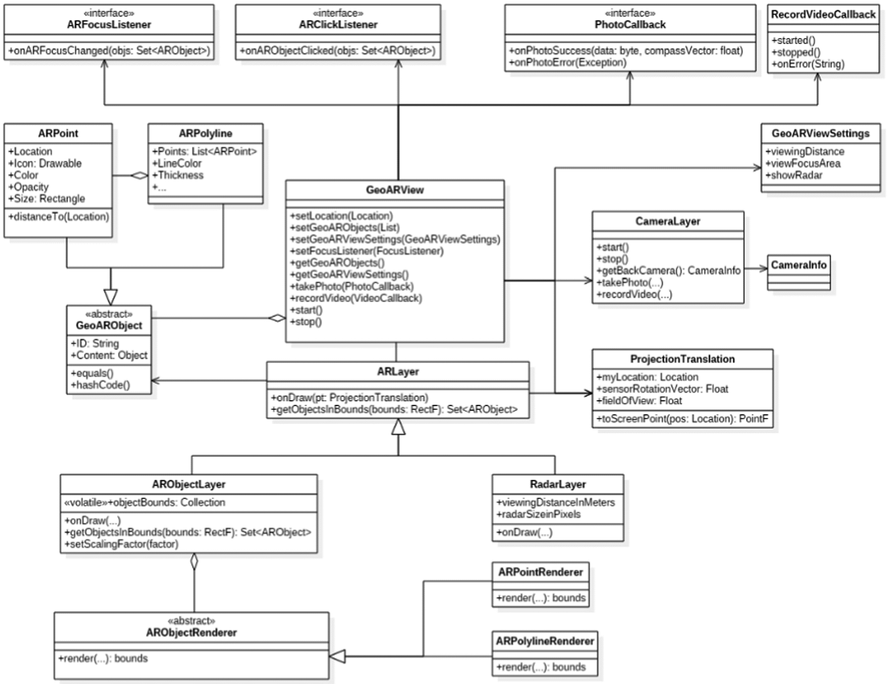


Fig. 7. Class diagram of the framework for location-based AR for Android

6 Implementation and Usage

A prototype of the framework was implemented for the Android platform and was used in two applications of the EU project, City.Risks. The overall idea of this project is to develop IT solutions to prevent and mitigate security risks in urban areas. With the aid of smartphones in smart cities, citizens actively contribute to combatting crime and increasing the sense of security.

Two applications for different use cases with integrated AR were developed using the location-based AR framework described in this article (see Fig. 8). One application uses mAR to show information about ongoing crime incidents in the city, explore crime-related data in this area, and to actively report issues. Another application allows the user to navigate out of a dangerous area to a safe destination using AR methods.

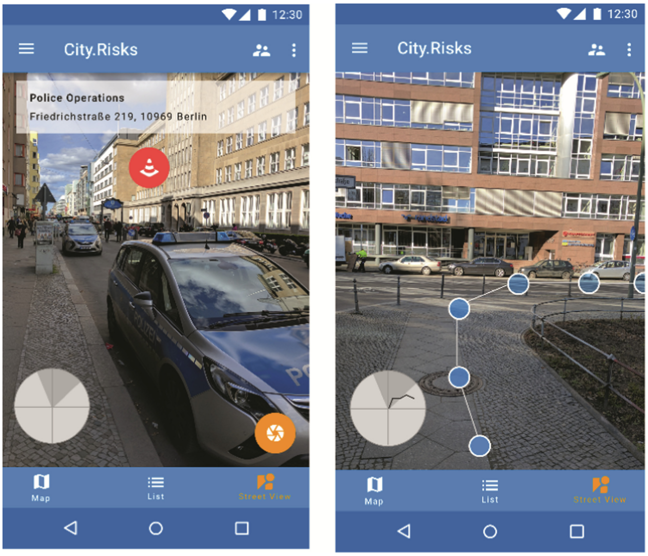


Fig. 8. City.Risks mAR applications: nearby incidents (left), Safety-aware navigation (right)

7 Summary

With the new generation of ubiquitous, powerful mobile devices (smartphones, etc.), a technical infrastructure is available that allows the development of a wide variety of mobile AR applications – especially in the field of location-based AR – as well as widespread use of these mAR applications by anyone.

In this article, the concept of a low-level development framework that facilitates the creation of a broad range of custom location-based AR applications has been presented. Geo-based AR technology has disappeared from sight in the last years as many companies have focused more on the optimisation of image-based AR technologies. Despite inaccuracies in the position determination, location-based AR is useful for many applications. However, developing one's own GeoAR application is often not possible without special expertise. The few existing SDKs can only be used under certain conditions as important functionality is missing or there are very few customisation options.

In contrast, the framework presented here allows the implementation of custom ideas with regard to appearance and functionality, thus enabling the creation of a wide range of different types of AR applications. It has been designed as a low-level framework for experienced app developers with no expertise in the field of computer vision. For this reason, it performs the mathematical and core technological functions of location-based AR.

Two applications from the EU project City.Risks were prototyped on Android using the framework. Later in the project, the viability of the framework will be more closely investigated. In particular, the degree of inaccuracy in the determination of the position and orientation of the device that can be tolerated will be explored. The extent to which this inaccuracy can be compensated by the use of additional methods (e.g., use of GPS bearing or feature tracking to improve the estimate of the rotation) will also be studied.

References

1. Azuma, R.T.A.: A survey of augmented reality. Presence - Teleoperators Virtual Environ. **6**(4), 355–385 (1997)
2. Höllerer, T., Feiner, S., Terauchi, T., Rashid, G., Hallaway, D.: Exploring MARS: developing in-door and outdoor user interfaces to a mobile augmented reality system. Comput. Graph. **23**(6), 779–785 (1999)
3. Fuchs-Kittowski, F.: Mobile Erweiterte Realität. WISU **41**(2), 216–224 (2012)
4. Rautenbach, V., Coetzee, S., Jooste, D.: Results of an evaluation of augmented reality mobile development frameworks for addresses in augmented reality. Spatial Inf. Res. **24**(3), 221–223 (2016)
5. Amin, D., Govilkar, S.: Comparative study of augmented reality SDK's. Int. J. Comput. Sci. Appl. **5**(1), 11–26 (2015)
6. Milgram, P., Takemura, H., Utsum, A., Kishino, F.: Augmented reality - a class of displays on the reality-virtuality continuum. In: SPIE Conference on Telemanipulator and Telepresence Technologies, vol. 2351, pp. 282–292 (1994)
7. Adhani, N.I., Awang Rambli, D.R.: A survey of mobile augmented reality applications. In: International Conference on Future Trends in Computing and Communication Technologies, pp. 89–95 (2012)
8. Mehler-Bicher, A., Reiß, M., Steiger, L.: Augmented Reality - Theorie und Praxis. De Gruyter Oldenbourg, München (2011)
9. Linaza, M.T., Marimon, D.: Evaluation of mobile augmented reality applications for tourism. In: Fuchs, M., et al. (eds.) Information and Communication Technologies in Tourism, pp. 260–271. Springer, Vienna (2012). https://doi.org/10.1007/978-3-7091-1142-0_23
10. Maier-Hein, L., et al.: Towards mobile augmented reality for on-patient visualization of medical images. In: Handels, H., Ehrhardt, J., Deserno, T., Meinzer, H.P., Tolxdorff, T. (eds.) Bildverarbeitung für die Medizin - Algorithmen - Systeme - Anwendungen, pp. 389–393. Springer, Berlin (2011). https://doi.org/10.1007/978-3-642-19335-4_80
11. Bischoff, A.: Dienste für Smartphones an Universitäten - ein plattformunabhängiges Augmented Reality Campus-Informationssystem für iPhone und Android-Smartphones. In: Roth, J., Werner, M. (Hrsg.) Ortsbezogene Anwendungen und Dienste. 8. GI/KuVS-Fachgespräch, pp. 127–135. Logos, Berlin (2011)
12. Haugstvedt, A.C., Krogstie, J.: Mobile augmented reality for cultural heritage. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2012), pp. 247–255 (2012)
13. Stampler, L.: Augmented reality makes shopping more personal - new mobile application from IBM Research helps both consumers and retailers. IBM Research (2012). http://www.research.ibm.com/articles/augmented-reality.shtml
14. Scott, G.D.: Enabling smart retail settings via mobile augmented reality shopping apps. Technol. Forecast. Soc. Change **124**, 243–256 (2016). https://doi.org/10.1016/j.techfore.2016.09.032
15. Joseph, B., Armstrong, D.G.: Potential perils of peri-Pokémon perambulation: the dark reality of augmented reality? Oxf. Med. Case Rep. **10**, 265–266 (2016). https://doi.org/10.1093/omcr/omw080

16. Inoue, K., Sato, R.: Mobile augmented reality business models. In: Mobile Augmented Reality Summit, pp. 1–2 (2010). www.perey.com/MobileARSummit/Tonchidot-MobileARBusiness-Models.pdf
17. Barker, S.: Augmented Reality - Developer & Vendor Strategies 2016–2021. Juniper Research (2016)
18. Blum, J.R., Greencorn, D.G., Cooperstock, J.R.: Smartphone sensor reliability for augmented reality applications. In: Zheng, K., Li, M., Jiang, H. (eds.) MobiQuitous 2012. LNICSSITE, vol. 120, pp. 127–138. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40238-8_11
19. Marchand, E., Uchiyama, H., Spindler, F.: Pose estimation for augmented reality: a hands-on survey. IEEE Trans. Vis. Comput. Graph. **22**(12), 2633–2651 (2016). https://doi.org/10.1109/TVCG.2015.2513408
20. Lahdenoja, O., Suominen, R., Säntti, T., Lehtonen, T.: Recent advances in monocular model-based tracking: a systematic literature review. Technical report, no. 8 (August 2015), University of Turku (2015)
21. Jeon, J., Kim, S., Lee, S.: Considerations of generic framework for AR on the web. In: AR on the Web, vol. 6 (2010) http://www.w3.org/2010/06/w3car/generic_framework.pdf
22. Fuchs-Kittowski, F., Simroth, S., Himberger, S., Fischer, F.: A content platform for smartphone-based mobile augmented reality. In: International Conference on Informatics for Environmental Protection (EnviroInfo 2012), Shaker, Aachen, pp. 403–412 (2012)
23. Keil, J., Zollner, M., Becker, M., Wientapper, F., Engelke, T., Wuest, H.: The House of Olbrich — an augmented reality tour through architectural history. In: IEEE International Symposium on Mixed and Augmented Reality - Arts, Media, and Humanities (ISMAR 2011), Basel, pp. 15–18 (2011)
24. Yu, K.M., Chiu, J.C., Lee, M.G., Chi, S.S.: A mobile application for an ecological campus navigation system using augmented reality. In: 8th International Conference on Ubi-Media Computing (UMEDIA 2015), pp. 17–22 (2015). https://doi.org/10.1109/umedia.2015.7297421
25. Armstrong, S., Morrand, K.: Ghost hunter – an augmented reality ghost busting game. In: Lackey, S., Shumaker, R. (eds.) VAMR 2016. LNCS, vol. 9740, pp. 671–678. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39907-2_64
26. Amin, D., Govilkar, S.: Comapartive study of augmented reality SDK's. Int. J. Comput. Sci. Appl. (IJCSA) **5**(1), 11–26 (2015)
27. SocialCompare: Augmented Reality SDK Comparison. http://socialcompare.com/en/comparison/augmented-reality-sdks
28. Wikipedia: List of Augmented Reality Software. https://en.wikipedia.org/w/index.php?title=List_of_augmented_reality_software&oldid=743628426
29. Bae, H., Walker, M., White, J., Pan, Y., Sun, Y., Golpavar-Fard, M.: Fast and scalable structure-from-motion based localization for high-precision mobile augmented reality systems. mUX J. Mob. User Exp. **5**(1), 4 (2016)