

Article

An Architecture for Mobile Outdoors Augmented Reality for Cultural Heritage

Chris Panou ¹, Lemonia Ragia ^{2,*}, Despoina Dimelli ² and Katerina Mania ¹

¹ Department of Electrical and Computer Engineering, Technical University of Crete, Kounoupidiana, 73100 Chania, Greece; cpanou@isc.tuc.gr (C.P.); k.mania@ced.tuc.gr (K.M.)

² School of Architectural Engineering, Technical University of Crete, Kounoupidiana, 73100 Chania, Greece; dimelli@arch.tuc.gr

* Correspondence: lragia@isc.tuc.gr

Received: 10 October 2018; Accepted: 26 November 2018; Published: 30 November 2018



Abstract: In this paper, we present the software architecture of a complete mobile tourist guide for cultural heritage sites located in the old town of Chania, Crete, Greece. This includes gamified components that motivate the user to traverse the suggested interest points, as well as technically challenging outdoors augmented reality (AR) visualization features. The main focus of the AR feature is to superimpose 3D models of historical buildings in their past state onto the real world, while users walk around the Venetian part of Chania's city, exploring historical information in the form of text and images. We examined and tested registration and tracking mechanisms based on commercial AR frameworks in the challenging outdoor, sunny environment of a Mediterranean town, addressing relevant technical challenges. Upon visiting one of three significant monuments, a 3D model displaying the monument in its past state is visualized onto the mobile phone's screen at the exact location of the real-world monument, while the user is exploring the area. A location-based experience was designed and integrated into the application, enveloping the 3D model with real-world information at the same time. The users are urged to explore interest areas and unlock historical information, while earning points following a gamified experience. By combining AR technologies with location-aware and gamified elements, we aim to promote the technologically enhanced public appreciation of cultural heritage sites and showcase the cultural depth of the city of Chania.

Keywords: augmented reality; navigation; instant tracking; 3D reconstruction; cultural heritage; computer graphics

1. Introduction

Augmented reality (AR) is the act of superimposing digital artifacts onto real environments. Across the reality–virtuality continuum [1,2], AR is part of the broader mixed reality spectrum. It enables real-time mixing of computer generated content and real content and has already been employed for medical, military, manufacturing, and robotics training applications [3]. In contrast to virtual reality (VR), where the user is completely immersed in a synthetic environment, AR aims to supplement reality [4,5]. While early research limited the definition of AR in a way that required the use of specialized head-mounted-displays (HMDs), a taxonomy introduced by the authors of [6] defined that any system that combines virtual and real imagery, accurately registering (aligning) real and virtual objects with each other, and running interactively in three dimensions and in real-time, is considered an AR system. AR software development kits (SDKs) are available to develop AR applications [7]. Current advances based on novel, commercial AR displays such as Microsoft HoloLens and Magic

Leap, as well as the huge investment drawn towards their development, showcase AR's potential for integration into the every-day life of the consumer.

Based on mobile phones' current technical specification enabling them to render 3D content and combine it with camera input, we developed the proposed mobile AR (MAR) application for Android mobile phones, offering a real-time, on-site, 3D depiction and visualization of historical monuments of the old town of Chania, Crete, Greece. We present a location-based AR application for Android devices that provides a sightseeing experience that aims to challenge and motivate the visitors to further explore and uncover the city's underlying history. In this paper, we present the software architecture of the system, which seamlessly incorporates AR features for the challenging, sun-wrenched outdoor environment of a busy Mediterranean town. The system proposed includes gamified scenarios and advanced AR features that enhance user experience while the visitor walks around interest points. The main functionality is incorporated in a digital map including the AR camera view. The main focus of the AR feature is to superimpose 3D models of historical buildings in their past state onto the real world, while users hold their consumer-grade mobile phones while walking on-site. Historical information in the form of text and images is available at the same time.

We are faced with significant technical, registration, and tracking challenges that relate to the accurate registration of 3D content at the exact location of the real-world monument, without the use of black and white markers, which are normally placed on surfaces. Moreover, the visualization of 3D content outdoors, taking into account extreme sunlight, visitors obscuring views, inaccuracy of geo-location based on global positioning system (GPS), as well as unstable computer vision registration approaches, poses challenging technical issues. We review potential solutions to the registration problem for outdoor AR implementations, evaluate them, and put forward a stable solution for mobile outdoor AR visualization without the need for placing markers.

Taking into consideration the historical significance of the monuments, our approach offers the opportunity to interact with them in non-intrusive ways, thus eliminating the need to interfere with the remains and on-going archaeological research. Based on a mobile, personalized, location-aware experience taking place in various areas of Chania, Crete, we aim to showcase the city's cultural wealth. The mobile AR application is based on a database that stores historical information of a large variety of monuments. The database also stores a real-time record of personalized user visits in their selected areas of interest, which is only available to each specific user after signing in.

The old Venetian part of the city of Chania was selected as the main spatial location for its rich history, enriched by influences from other cultures. It is distinguished by the presence of historical monuments from the Minoan, Roman, Arabic, Venetian, and Ottoman periods until the Greek historical period. Adverse climate conditions, modern city planning, and rapid expansion have slowly compromised the state of these sites, thus endangering their historical value over time, as well as their original beauty. For this work, a specific route around the Venetian part of the city of Chania was selected as the main route that a tourist holding a mobile phone would traverse. In relation to the AR visualization, three monuments were selected, representing three distinct historical periods and architectural interests. These are the following: (a) the Glass Mosque (Figure 1a) from the 17th century [8]. This is the first mosque built in Crete and the only one remaining in the city. It included a minaret (Figure 1, right 1b, 1c), which was demolished. (b) The Saint Rocco temple is one of many Venetian churches in Chania (Figure 2) [9]. Only part of the church is in place, while the exterior facades are painted and the original texture has been lost. (c) The Byzantine wall, which is one of the most famous monuments in the city, built around the sixth and seventh century AD (Figure 3) [10].

The Glass Mosque (Figure 1) is located in the Venetian harbor of Chania and is the first mosque that was built in Crete and the only surviving one in the city, dating from the second half of the 17th century. Erected in honor of the first garrison commander of Chania, named Küçük Hasan, it is a jewel of Islamic art in the Renaissance. The mosque is a cubic building covered by a large hemispherical cupola supported by four ornate stone arches. Its western and northern parts are surrounded by a covered arcade of six small cupolas, which are open at the top, as is customary in

mosques. Around 1880, the arcade was covered with arched openings of neoclassic style. The small, but picturesque minaret was demolished in 1920 or in 1939. It was quite badly damaged by bombing during WWII. After suffering substantial damage during the war, it was finally restored and moved to Chania's archaeological museum. Later, it was used as either a warehouse, a folk art museum, the Information Office of the Hellenic Organization of Tourism, and recently as a home to seasonal events and exhibitions.



Figure 1. (a) The Ottoman Glass Mosque in its current state on the left (author's picture). Its original state in the middle (b) and on the right (c). Image (b) comes from G. Despotaki's archive and image (c) from M. Manousaka's archive.



Figure 2. Front side of the temple showing the northern (left) and the southern (right) part.

The Saint Rocco temple is a Venetian chapel on the northwest corner of Splantzia square, and consists of two different forms of vaulted roof aisles (Figure 2). Although the southernmost part is preserved in good condition, the northern and oldest one has had its exterior painted over, covering its stony façade, while a residential structure has been built on top.



Figure 3. The Byzantine wall and the part chosen to be 3D modeled in green (left). The demolished and built over towers of the wall (right).

The Byzantine wall was built over the old fortifications of the Chydonia settlement around the sixth and seventh century AD (Figure 3). Its outline is irregular with a longitudinal axle from the east to the west, where its two central gates were located. The wall consists of rectilinear parts, interrupted by small oblong or polygonal towers, many of which are now partly or completely demolished (Figure 4).

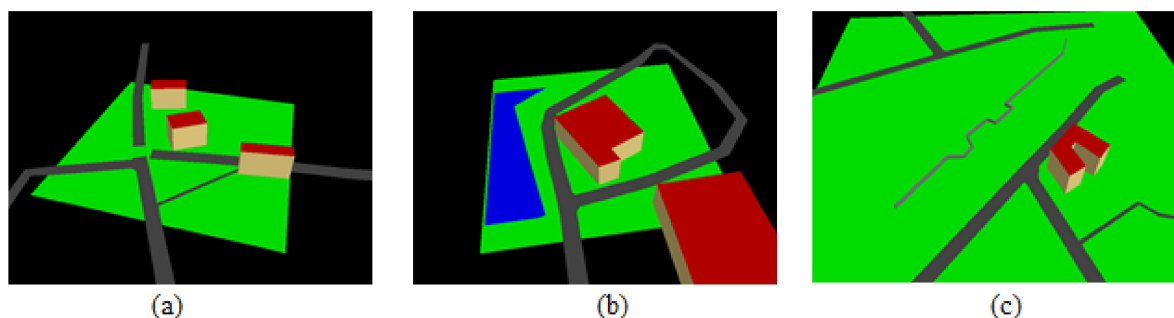


Figure 4. (a) Glass Mosque; (b) Saint Rocco temple; (c) Byzantine wall.

The scope of this work is to virtually restore partially or fully damaged buildings and structures on historic sites and enable visitors to see them integrated into their real environment through a mobile phone. Our aim is to design and implement a mobile AR application for Android devices that will help visitors interact with the city's monuments. We aim to deliver geo-located information to the users, through text, images, and AR, and help them document their visits. By integrating digital maps and a location-based experience, we aim to urge the users to further investigate interest areas in the city and uncover their underlining history. The AR mobile application developed aims to provide an easily extendable platform for future additions of digital content, and requires a moderate amount of development and technical expertise. The goal is to provide a complete and operational AR experience to the end-user by tackling significant AR technical challenges related to the accurate registration and positioning of the 3D content integrated into a mobile application.

Previous Work

In comparison with older see-through AR displays, which were head-mounted, based on cumbersome hardware and complicated software modules, a recent emergence in mobile technology has led to an integrated platform, ideal for the development of AR experiences, often referred to as mobile AR (MAR). MAR is a concept first conceived in the late 1990s, producing early AR systems for cultural and archaeological sites, mainly indoors. However, today's modern smartphones have brought AR to a wider audience. The presence of high processing power, cameras, and inertial and global positioning system (GPS) sensors provides the necessary components of an AR system in an ergonomic hand-held device.

In past years, AR has been utilized for a number of applications in cultural heritage. *Archeoguide* was first presented using AR for personalized tours in cultural heritage sites [11,12]. The system allowed users to experience a VR world featuring computer generated 3D reconstructions of ruined sites without isolation from the real world. Two years later, an extended version presented a personalized mobile guide for outdoor archaeological sites [13], employing the site of Ancient Olympia as a test case [14]. The system provided on-site help and AR reconstructions of ancient ruins. It made use of a compass, a DGPS (differential global positioning system) receiver integrating images streamed from a webcam, and the users' location and orientation. Visitors carried a backpack computer that performed the calculations, and wore a see-through head-mounted-display (HMD) to display the digital content. By today's standards, the system was cumbersome and heavy. Despite the ergonomic restrictions, it was very well-received by the visitors as it provided a unique site-seeing experience [14].

A MAR application representing a historical tour guide including old photographs and information about a historical street has been reported in the work of [15]. An early overview of AR in cultural heritage highlighted the technical problems of AR development, mostly centered on registration and rendering issues, as well as having to rely on black and white markers for correct alignment [16]. Other AR approaches aimed to connect the excavation site with the artifacts in the museum in order to enable contextual awareness of the visitor [17]. The Augmented Representation of Cultural Objects ARCO system combined VR and AR, enabling museums to display their 3D, digitized

collections on the web [18]. The users could interact with sensor-enhanced physical objects while their digital reconstructions were simultaneously manipulated [19]. An AR framework for on-site visualization of archaeological data employed visual tags, overlaying 3D models on images provided in real-time by the phone camera, aiming to display accurately positioned information [20].

MARCH was a MAR application for digitally enhancing visits to prehistoric caves [21]. It was developed in Symbian C++, running on a Nokia N95. The system made use of the phone's camera to detect images of cave engravings and overlay an image of the original drawings on them, on-site. This was the first attempt of a real-time MAR application without the use of grey-scale markers. Instead, colored patches positioned at the corners of photographs were employed, viewing these and not the real-world environment. An integrated mixed reality system for recreating virtual human actors by integrating the real world and a 3D scene populated by them was presented by the authors of [22]. This platform was employed on the real-world site of ancient Pompeii [23].

With the emergence of smart mobile devices, sophisticated AR experiences were made possible, such as the one placed on the Bergen-Belsen memorial site. This was a former WWII concentration camp in northern Germany, which was burned down after its liberation [24]. The application integrated database interaction, reconstruction modeling, and content presentation in a hand-held device. The system was developed for an iPhone. Real-time tracking was performed based on the device's GPS and orientation sensors. Navigation was conducted via either maps or the camera. The system superimposed 3D reconstructed buildings on the actual site, visualizing their past and present state. Focusing on the promotion of cultural heritage in outdoor settings, VisAge aimed to transform users into authors of cultural stories in urban environments through physical space [25]. A story consisted of spatially distributed POIs (points of interest). Each POI was assigned its own digital content including images, text, or audio. A viewing tool was developed for mobile tablets in Unity3D using Vuforia's tracking library in order to overlay the digital content on the real space. The users could then follow these routes in the city and experience new stories.

Further work related to 3D reconstructions was shown in CityViewAR [26], a mobile outdoor AR application allowing the visualization of destroyed buildings after major earthquakes in Christchurch, New Zealand. Besides providing stories and pictures of the buildings, the main feature of the application was the ability to visualize 3D models of the buildings in AR. A practical solution presented by the authors of [27] guided groups of visitors in noisy indoor environments, employing analogue audio transmission and reliably trackable AR markers. Preparation of the environment with fiducials and supervision of the visits by experts was necessary in order to avoid accidents and interference with the working environment.

PRISMA combined the concept of tourist binoculars and AR technologies [28]. The real-world scene was enhanced by digital information, providing an interactive user-friendly system. AR interfaces for visiting cultural heritage sites employed multimedia sketches as a means of communication between experts [29]. An AR tourist guide was designed and implemented for mobile devices, which enabled tourists to present historical information [30]. Real-time tracking was performed using either computer vision techniques or sensors such as GPS and gyroscopes. Another approach puts forward a 3D reconstruction methodology applied to the restoration of historical monument based on tacheometry data [31]. AR technologies were developed to visualize restoration areas and their effect after restoration.

Putting forward a mobile, personalized, location-aware MAR experience taking place in the historical part of Chania, Crete, we aim to enhance user experience and interaction with cultural heritage sites and showcase the city's cultural wealth, as well as to address technical challenges related to registration techniques for AR outdoors. The mobile AR application presented features a database that holds records of historical monuments on-site. The database stores users' documentation of their visits and interactions in their areas of interest, available to each user that signs in. AR development of a city poses significant technical, as well as user interaction, challenges. Reliable position and pose tracking is paramount so that the 3D content representing the monuments in their past state

is accurately superimposed on real settings at the exact position required. This is one of AR's most significant research challenges. Our system features a geo-location and sensor approach, which, compared with optical tracking techniques, allows for free user movement throughout the site, independent of changes in the buildings' structure. Moreover, we combined this implementation with a hybrid registration technique (sensor-based and vision-based), in order to showcase the capabilities of future AR technologies.

2. AR Methodology

In this work, we present the design and implementation of a MAR (mobile augmented reality) application for Android devices that provides on-site 3D visualization of historical buildings located in the historical part of Chania, Greece. These are superimposed over their real-world equivalent, as part of a smart AR tourist guide. Further to 2D images and text often presented by mobile tourist guides, we aim to enrich the sightseeing experience by providing a means to visualize the past glory of these sites in the context of their real-world surroundings. Taking into consideration the historical significance of the monuments, our approach offers the opportunity to interact with them in non-intrusive ways, without physically interfering with the remains and on-going archaeological research. In order to accurately record the geo-location of the monument and provide real-time tracking, the standard sensors of a mobile phone are used. We do not place markers or patches on the monument and the implementation of the software does not require any physical contact with the monument.

2.1. The AR Registration Technical Challenge

AR registration is the degree to which 3D information is accurately placed and integrated as part of the real environment. The objects in the real and 3D scene should be correctly aligned with respect to each other, or the illusion that the two co-exist is compromised [32]. In contrast to VR, where such errors result in visual-kinesthetic conflicts, in AR, such conflicts are driven by the visual sense alone and are easier to detect. A user wearing a VR headset raising an arm and actually viewing a virtual one at the same time that is off by a few centimeters may not detect the spatial misplacement because of the conflict between the "sensed" position of the real arm and the "seen" position of the virtual one. In the corresponding AR application, the virtual arm should completely overlap the real one; therefore, such an error would be easily detectable.

Registration errors are defined as dynamic and static. Static errors are the ones that affect the AR scene even when both user and environment are in stasis. Sources of such errors can be the inaccurate calibration of mechanical parts, incorrect tracker-to-eye ratio, field-of-view parameters, optical lens distortions, and so on. Static errors depend mostly on mechanical parts and the correct initial calibration of the system, and can be accounted for to a very satisfying degree. Dynamic errors, on the other hand, are the ones that take effect if either the viewpoint (user) or the annotated object begins moving. For MAR, this error is by far the largest contributor to the registration problem and varies depending on the implementation.

In early AR systems, the single most important factor for dynamic errors was end-to-end system delays. A tracker reports user movements, and the system should then update the digital artifacts on the screen. This computation and its delivery should precede changes in the user's pose, which proved to be a very difficult task at the time. With today's hardware, system delays have been minimized and the main source of errors in registration is pose estimation, for example, position and orientation tracking. AR tracking is a complicated task with no single best solution.

In order to register virtual content in the real world, the pose (position and orientation) of the viewer with respect to some "anchor" in the real world must be determined. Depending on the application and technologies used, the real-world anchor may be a physical object, such as a magnetic tracker source or paper image marker, or may be a defined location in space, determined using GPS or dead-reckoning from inertial tracking.

In order for an AR system to overlay the world with digital information, it is required to track its position in 6 DoF (degrees of freedom), three for position and three for orientation. Inertial sensors, GPS positioning, and optical sensor scans provide the necessary data for such computations. In this paper, we will focus on AR pose estimation for consumer-grade smartphones. The main approaches are the following: vision-based tracking, which relies on the device's camera and ways to process the live feed; and sensor-based, which combines GPS positioning and the inertial sensors of the device.

Optical and sensor-based tracking were tested on-site. Two dimensional image recognition, which relies on natural feature detection algorithms, was tested using a variety of captured images of the annotated monuments. Those images are processed to produce 2D point clouds of the detected features to be identified by the mobile device in the camera's feed. When detected, pose and position estimations are available, relative to the surfaces they reproduce. While image recognition has presented great results thanks to fast hardware and improved algorithms, we could not rely on it. Outdoor environments present challenging conditions. Building façades provided a small amount of features to be recognized and, taking into account variations in lighting conditions, the sets of features provided to the AR system differed greatly when compared with the real scene. Thus, the targets were not reliably recognized. In the rare cases where tracking was possible, simple movements of the device resulted in miscalculations of the orientation. Taking also into account the cramped environment of a touristic site compromising the visibility of the targets, image recognition was not perceived as a realistic choice for AR registration and tracking in our case.

2.2. The AR Registration Approach

There is no single best solution to AR tracking and registration. Vision-based approaches are best suited for controlled and small environments, but their performance diminishes in wide and outdoor areas, where the sensor-based approaches provide the best results. Tracking in AR systems is an open problem. Although the future seems to lie within hybrid implementations, they are currently in their infancy and most often require additional hardware components.

The most important criterion when selecting an AR tracking solution is reliability. Graphics have little meaning when tracking is not possible. In this paper, we selected the geo-location approach and the instant tracking option of the Wikitude SDK. While a case can be made for the registration difficulties of the geo-location approach due to sensor filtering and low GPS accuracy, these implementations require fewer actions by the users and ensure that the AR experience will be delivered independent of external conditions. In contrast, the instant tracking option provides greater registration accuracy and eliminates the latency, but is susceptible to occlusions and requires additional actions by the users. The decision to include both implementations was taken in order to provide users who are unaccustomed with AR applications an intuitive way of visualizing the 3D models based on geo-location, while also being able to provide a more sophisticated AR experience with instant tracking. The Wikitude JavaScript API was selected because of its robust tracking, educational licensing option, documentation, large community, and efficient customer service.

The geo-location approach employs the GPS and the inertial sensors of the device so that when the specific location of the actual monument is registered, the 3D visualization of it would be displayed. Locations containing latitude and longitude information are received by the GPS, while the accelerometer and geo-magnetic sensors are used to estimate the pose of the device in the Earth's frame. A visual reconstruction is then matched to the user's position and viewing angle, displaying the overlaid models on the mobile phone's screen. This implementation offered the most reliable registration of 3D content, accurately superimposed on the real-world site, demanding less action from the users and, therefore, ensuring a robust and intuitive experience [30].

The preparation of the 3D models required the acquisition of historical information and their accurate depiction in scale with the real world. Because of the lack of accurate plots and outlines of the buildings, LiDAR (Light Detection and Ranging) and DSM (Digital Surface Models) data were exported from Open Street Maps and used to create the final models. Developing such models for a

mobile device means that the limited processing power and the requirements of the AR technologies need to be taken into account. Complex geometries can impair performance, so we followed a low-poly, high-resolution texture approach to avoid drops in frame rate.

A key concept for any location-aware application is the cognitive map held in the form of mental images by the user [33]. Digital maps and an AR camera displaying the interest areas were integrated to assist in navigation through the geo-located content. The client-server architecture ensures that we provide personalized experiences by storing information about user visits and progress. Changes in the server can be conducted without interfering with the mobile application, allowing for an easily extendable platform where new monuments could be added as visiting areas. The monuments' information and assets are stored in a database and delivered to the mobile application on a location-request basis. The application was developed for Android in Java, and employs the Wikitude Javascript API to handle the AR views. It features a local database based on SQLite caching the downloaded content. The complete design and implementation is described in detail in the following sections.

The main screen of the proposed application is the map indicating the location of the user, the available POIs, as well as the main navigation buttons. The aim of the map is to help the user navigate the city. This navigation can also be accomplished via the camera view where the POIs are displayed in 3D space on the camera's surface. In its initial state, the user is shown the available monuments that can be augmented by 3D models as markers on the map. The path between them is shaded with polylines in order to visually integrate these points. Upon visiting the monuments, the user has access to the 3D reconstructions through the AR camera. The initial monuments act as an introduction to the application's main features. After users visit a specific monument, they are awarded a number of points. When all the available 3D models are visited, the game enters its main state. The interest areas of the system pop out as question marks on the map. The goal is to visit all interest areas and unlock them to earn more points. In order to unlock a POI, the user has to correctly classify it according to its respective historical period. Information concerning historical periods relevant to specific monuments can be acquired from the application. When a user correctly unlocks a monument, access to its historical information is available. The aim of this approach is to urge the user to closely observe these monuments, consult the information already unlocked, or even interact with the locals to get as much information as possible. The user is transformed as an active participant of the sightseeing experience instead of a passive spectator.

The historical and user-specific information is stored in a remote database connected to the mobile application via a REST (Representational State Transfer) web service. The application requests the additional interest areas based on a location request and updates the user's progress and monuments' statistics depending on the actions that took place. We can then display user-related rankings and leaderboards, creating a challenging and competitive experience that aims to further motivate the users to explore the city and subsequently promote its cultural heritage.

3. 3D Depiction of Historical Monuments

3.1. 3D Modelling and Texturing

In order to record the past state of the selected monuments, old photographs, historical information, and estimates from experts were utilized. The 3D models visualizing their past state will be presented in real size and superimposed over the real-world monument, and must be in proportion with their surroundings. Therefore, accurate measurements of their structure are necessary. Because of the lack of schematics and plots, we relied on data derived from online mapping repositories that provide outlines and height. In order to ensure historical accuracy and avoid the communication of false information, the final models and their reconstructed parts are in abstract form, depicting only the main structural elements of each monument. Instead of presenting information to its full extent, it can be represented by a decided level of abstraction, or 'level of detail', providing the

minimum information required. An abstract form is often used by architects to depict the general form of a structure, emphasizing its open-ended character function as an initiator of possibilities and potentialities. This can be related to how missing parts of a historical building could have appeared in the past, shedding light on archeological or historical uncertainty concerning its past form [34,35]. The focus of this work was not the accurate reconstruction of buildings, but mostly the AR component working as seamlessly and ‘in-place’ as possible in relation to the registration of 3D content with the real-world.

The outlines of the three monuments were acquired from Open Street Map (OSM). By selecting specific areas of the monuments on the map, we can then export a .osm file that contains the available information concerning that area, including building outlines and height, where available. This file is essentially an xml file comprising OSM raw data including roads, nodes, tags, and so on. The file is then imported into OSM2World, a Java application aiming to produce a 3D scene.

These representations are basic triangulated meshes of the outlines raised to reach the height value for each building. As is evident in relation to the Byzantine wall as it stands now, height data were not available to us. In relation to the two monuments shown in Figure 4, it is not clear whether the domes and roofs have been taken into consideration. These models formed an initial basis and any disproportions were to be corrected after on-site testing. The models were then exported to .obj format and imported to the Blender 3D modeling software.

On the basis of the basic structures of the buildings, the final 3D mesh to be included in the mobile AR application was created. The modeling process was focused on preserving a low vertex count as complex geometry compromises an interactive framerate in systems with low processing power such as mobile phones. During the initial stage, we only created the demolished part, to be overlaid on the real buildings (Figure 5).

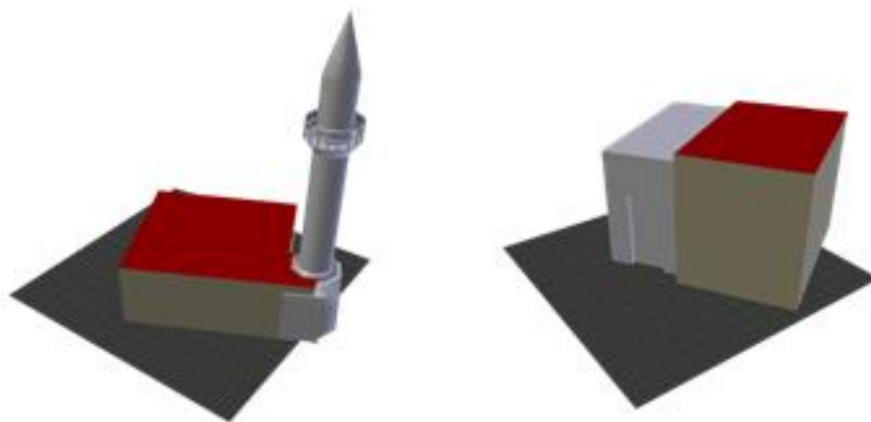


Figure 5. The 3D mesh of the 3D models in scale with the existing buildings.

However, on-site testing showed that the average accuracy of the GPS receiver of three meters and the constrained viewpoint in certain sites, such as the Saint Rocco temple, break the illusion of the 3D information coexisting as part of the real world because the 3D parts were not registered accurately with the real world. Instead, complete 3D depictions of the selected monuments were created so that they completely overlap the real ones. The most important aspect of this strategy is to keep the 3D models of the monuments in proportion in terms of size. The final scale and size were defined in Google Sketch-up, while the final model was positioned in the world coordinate system. The final models are shown in Figure 6.

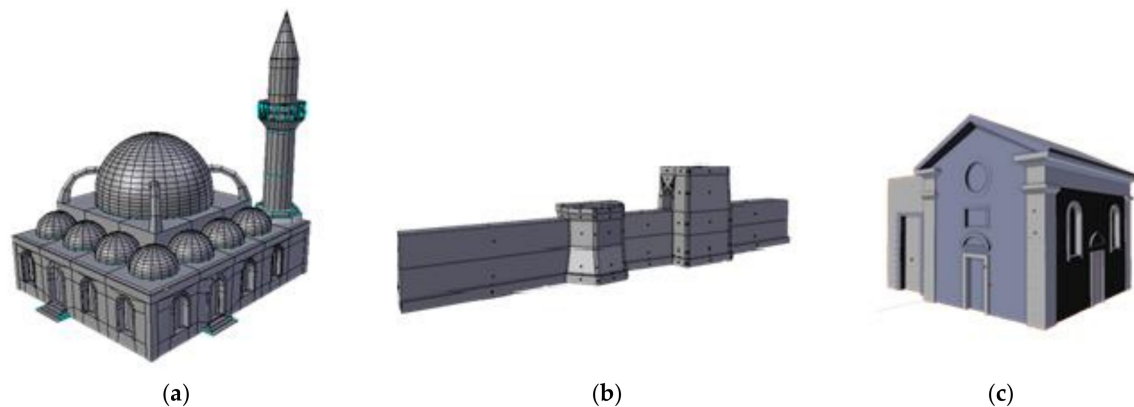


Figure 6. The final 3D meshes: (a) Glass Mosque; (b) Byzantine wall; (c) Saint Rocco temple [36].

These models constitute the state of the building in the respective historical periods. Reference images were used for modeling the existing parts of the buildings, while further details were added to keep them consistent with the buildings as they stand in the present time. We perceived this as a valid procedure, however, we accept that elements of historical structures visible at the present time should be validated by employing historical information. Such detailed work, especially in the domes and railings areas of the Glass Mosque (Figure 6a) and the column area in Saint Rocco (Figure 6c), unavoidably heightens the polygon count. However, the performance of displaying them on a mobile device was deemed satisfactory after testing. The Byzantine wall (Figure 6b), being the most abstract, resulted in 422 vertexes. The Glass Mosque resulted in 7614 vertexes and the Saint Rocco temple in 4919 vertexes.

Texture mapping is a method of adding photorealism and detail to a flat surface without adding extra geometry. A texture map is a bitmap image that is applied to the surface of a polygon, creating a high fidelity visual result. Images captured from the real monuments on site were used as references in order to locate the surface materials. The texture mapping procedure followed a multi-material approach. A diverse range of materials is assigned to different parts of the buildings. Each material is then assigned images as texture to the surfaces. Because of the lack of information, the actual texture of the reconstructed parts is unknown. The aim is to accurately represent the compositing material rather than the actual surface, taking into account the texture of the remains. Another approach would be to use supported hypotheses of similar constructions of the same era [37]. At this point of the procedure, testing of the rendering capabilities of the AR framework was conducted utilizing the derived models. The AR framework only supports a power of two png or jpeg single material texture map. This means that bumps, normal maps, and multi-textures cannot be included for better surface representation.

The materials that compose the entire texture are “baked” into one image that will serve as the final texture employing the 3D modeling software ‘Blender’. UV (ultra violet) mapping is the process of unwrapping the 3D shape of the model into a 2D map. This map contains the coordinates of each vertex of the model placed on an image. The materials that were assigned to the surfaces are then baked onto the image that forms the final texture. Taking into account that the monuments will be displayed on a mobile phone screen in real size, we needed high resolution textures. This unfortunately raises the final size of the texture files, however, the process results in a high quality visual result. The final texture resolution is 2048×2048 pixels (Figure 7). In order to light the scene, a simple hemi light was used, provided by 3D modelling software. The final models were exported without a light source. Although the AR framework supports lighting, having a static light in a real-world scene would impair the experience, and dynamically controlling the light source is not possible.

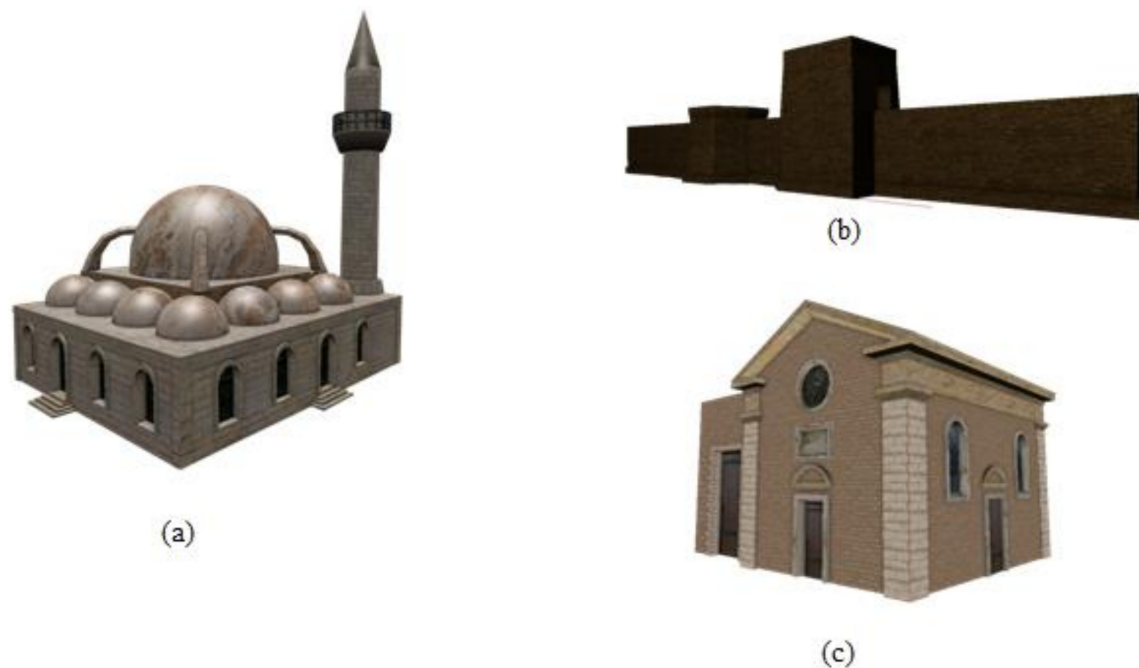


Figure 7. Final textured models, (a) Glass Mosque; (b) Byzantine wall; (c) Saint Rocco temple

3.2. Geo-Localization

Sensor approaches rely on the inertial sensors of the device for orientation tracking and on the GPS for positioning. For Android devices, these sensors can either be hardware components such as magnetometers, accelerometers, and gyroscopes or software components that rely on multiple physical components and sensor fusion to produce specific results, such as the linear-acceleration sensor, the orientation sensor, the gravity sensor, and others. These sensors can be categorized as motion sensors that measure acceleration forces and rotational forces along three axes of the device, such as the accelerometer, gyroscope, and positioning sensors that provide absolute measurements about the physical position of a device such as the magnetometer and the orientation sensors. The geo-location approach requires measurements relative to the Earth's frame of reference. This requires the combination of motion and positioning sensors to determine the orientation and their combination with GPS, which provides latitude and longitude measurements for position.

If the virtual objects are accurately positioned and oriented in the world coordinate system, the AR system is aware of the distance between them because that model is built into it. The digital content is associated with a geo-location that is then matched to the user's position and viewing angle. For these implementations, we used the HitlabNZ mobile AR framework and the Wikitude JavaScript API. These tracking techniques rely on cheap sensors present in every modern smartphone.

The most significant challenge is the registration errors introduced by the GPS accuracy and the latency derived from filtering sensor data. The AGPS (assisted global positioning system) present in mobile phones has an average accuracy of three meters. This error margin is proven to be impactful, especially in the case where we overlay the reconstructed parts of a monument onto the real-world. The illusion of digital and real coexistence completely breaks, even if the 3D model is registered slightly away from the real building. The orientation calculation is based on the accelerometer and the magnetic field sensors and a combination of these two with a gyroscope, if present. While the magnetic field sensor provides absolute measurements and needs no filtering, the accelerometer measures all forces that act on the device, which means that unwanted motions and mechanical noise need to be filtered out in order to isolate the force of gravity. This filtering process unavoidably introduces latency to the system following relative motions of the camera. For example, when a user moves the device to

look higher at an overlaid building, the 3D model will be dragged along with the motion until it is significant enough to pass through the filter.

In order for the 3D models of the monuments to be accurately displayed in combination with real-time viewing of the real world, an initial transformation and rotation needs to be applied. The 3D models were exported in .dae format and imported into Google Sketch-up. The final models are then processed through Google Sketch-up to geo-reference and position them onto the real world while looking at the screen of a mobile phone (Figure 8).

The area of the monuments provided by Google maps was projected onto a ground plane. We then positioned the monument on its counterpart on the map. Given that the proportions of the monuments are in line, the final model was scaled to fit on the outlines. The location of the monument was then added to the file and provided to the framework.

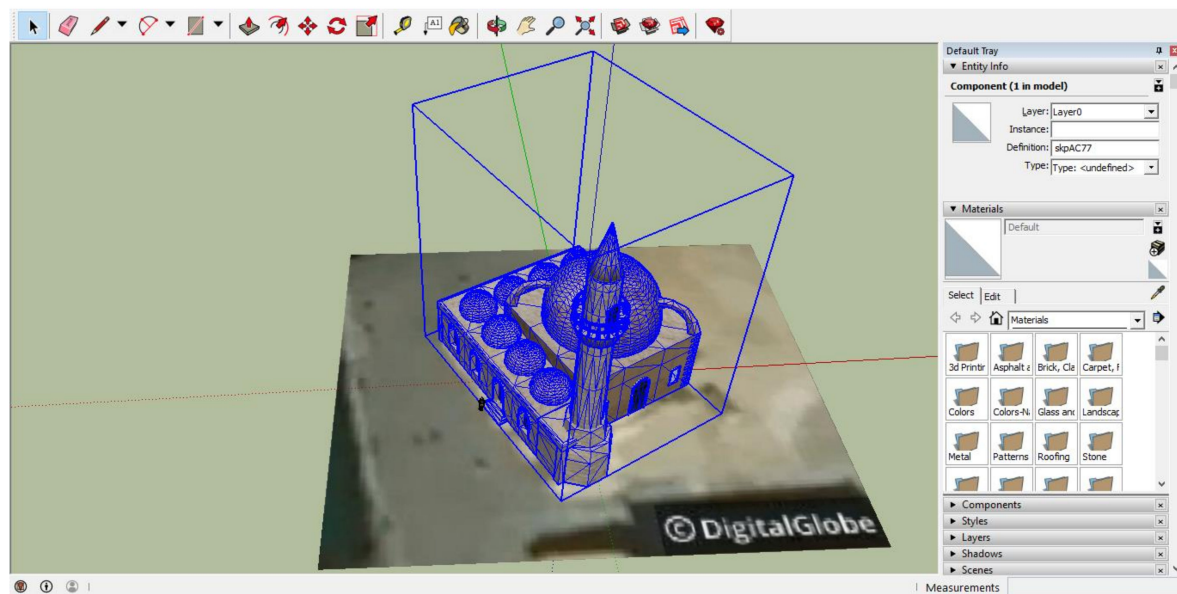


Figure 8. Geo-locating the model in Google Sketch-up.

In order to include the model in the AR framework, we exported it and used the provided 3D encoder to make a packaged version of the file together with the textures in the custom wt3 format. This file was then channeled to the MAR application.

4. System Architecture

4.1. Client-Server Implementation

One of the main challenges of MAR is the lack of established guidelines in the application and integration of AR technologies in outdoor heritage sites. The overall design of our system consists of two main parts, that is, the mobile client and the server. The server facilitates a database developed in MySQL, which holds the monument records (name, description, latitude, longitude, etc.) and user-specific information. While polygon geometry would be more accurate in defining and differentiating the areas of interest, circles centered on latitude and longitude points make calculations based on location far easier. Should the current model be expanded, polygon geometry would be a major consideration. The information is delivered to the mobile unit on a location request basis. The database is exposed to the users via a REST-full web service. A basic registration to the system is required. The functions provided to the users include storing data about visits, marked places, and the overall progress in an area, while all the monument information is delivered to the mobile device's views. The system architecture is shown in Figure 9.

4.2. Mobile Application

The mobile application's architecture has been designed to be extendable and to respond easily to changes in the underlying model of the server. It is based on three main layers (Figure 9). The *Views layer* is where the interactions with the users take place. Together with the background location service, they act as the main input points to the system. The events that take place are forwarded to the *Handling layer*, which consists of two modules following the singleton pattern. The data handler is responsible for interaction with the local content and communicating with the views, while the REST client is responsible for communicating with the web service. The *Model layer* consists of basic helper modules to parse the obtained JSON (JavaScript object notation) files and interact with the local database. The actions flow from the *Views layer* to the lower components. Responding to a user event or a location update, a call is made to the handling layer, which will access the model to return the requested data.

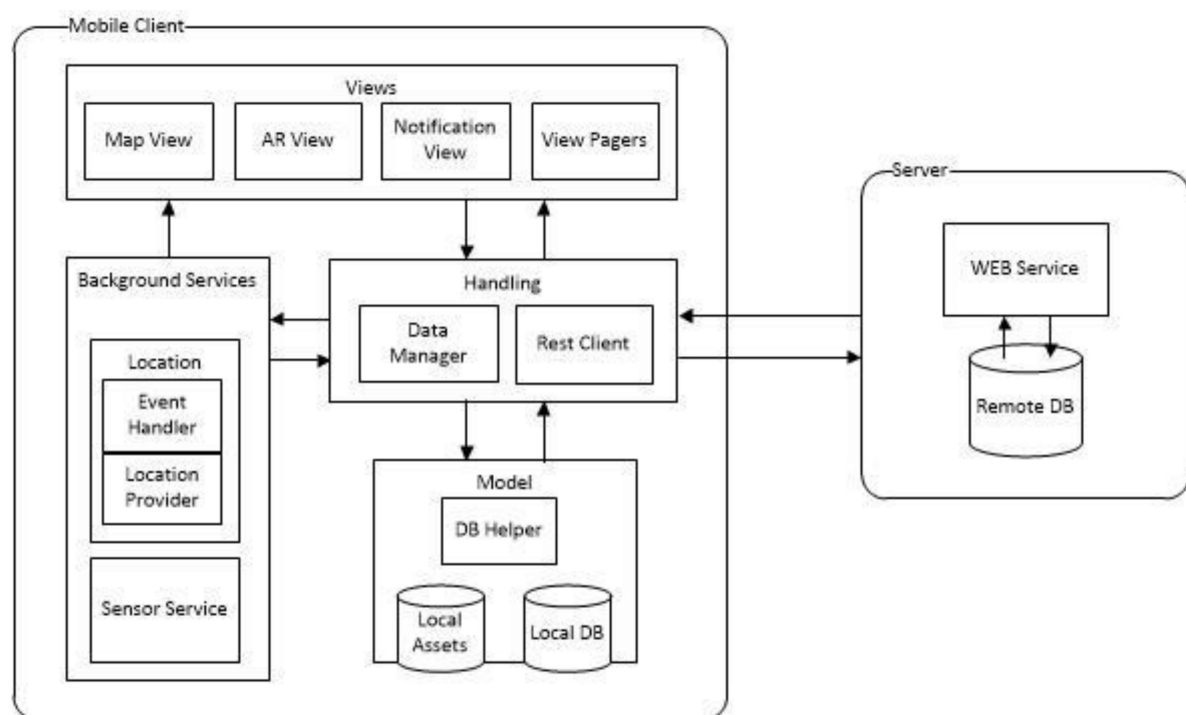


Figure 9. Architecture of the system. AR—Augmented Reality; DB—database.

The *Handling layer* is the most important of the three because interactions, exchange of information, and synchronization pass through this layer. The REST client provides an interface for receiving and sending information to the remote database as requested by the other layers. It is responsible for sending data about user visits, saved places, updates in progress, and personal information. It also provides functions for receiving data from the server concerning POIs' information and images. It also allows for synchronizing and queuing the requests. While the REST client is responsible for interactions with the server, the data handler is responsible for managing the communication of the local content. Information received is parsed and stored in the local database. The handler responds to events from the views and background service and handles the business logic for the other components. It forms and serves the available information based on the state of the application.

The views are basic user interface components facilitating the possible interactions with the users. The *Map View* is a fragment containing a 2D map developed with the Google Maps API. It displays the user's location as obtained by the background service and the POIs as markers on the map. The *AR View* is based on the Wikitude JavaScript API and this is where the AR experiences take place. It is a *Web View* with a transparent background overlaid on top of a camera surface. It displays the 3D

models of the historical monuments while receiving location updates from the background service and orientation updates from the underlying sensor implementation. It also contains a *Navigation View*, where the POIs are displayed as labels on the real world. Interactivity is handled in JavaScript and is independent of the native code. The view pagers are framework specific user interface elements that display lists of the POIs, details for each POI, user leaderboards, and user profiles. Finally, the *Notification View* is used when the application is in the background and aims to provide control over the location service. It is a permanent notification on the system tray, where the user can change preferences of the location strategy and start, stop, or pause the service at will.

The aim of the standalone background service is to allow users to walk freely in the city while receiving notifications about nearby POIs. It is responsible for supplying the locations obtained by the GPS to the *Map View* and *AR View*. The *Location Provider* is the component responsible for obtaining the locations. It offers the option to swap between the Google Play Services API and the Android Location API, which are two different location strategies. In order to offer control over battery life and data usage, the users can customize its frequency settings from the preferences' menu. The views requesting location updates are registered as *Listeners* to the *Location Service* and receive locations containing latitude, longitude, altitude, accuracy information, and so on. The *Location Event Controller* is the component responsible for serving the location events to the registered views. The user's location is continuously compared with that of the available POIs. If the corresponding distance is in an acceptable range, the user is able to interact with the POI. The events sent include entering and leaving the active area of a POI. If the application is in the background, a notification is issued leading to the *AR View* and *Map View*.

The *Model layer* consists of standard storing units and handlers to enable parsing JSON files obtained from the server and interfaces to interact with the local database. It stores historical information about the current local, user-specific information and additional variables needed to ensure the optimal flow of the application. The local assets, including the 3D models and the html and JavaScript files required by the Wikitude API, are stored in this layer and provided to the handling layer as requested. The SQLite helper is the component responsible for updating the local storage and offers an interface to the data handler containing available interactions.

4.3. Server Implementation

The aim of the application server is to provide an online storage unit of historical and user-specific information. It employs a REST-full web service providing information to the mobile clients. The server holds a database of historical information as well as user information for authentication and accessibility control. The mobile units request the information and store it in their local database. The data can be queried to provide personalized information to the users and useful statistics about the city. The Web API exposes its resources via unique custom defined URLs held by the mobile client. Each key entity in the database schema is mapped to a relative path from the base URL of the server and for each entity identified with the URL, the client uses a different type of HTTP request methods (GET, PUT, POST, and DELETE) (Figure 10).

By accessing these URLs and defining the HTTP method, the clients can perform CRUD (create, read, update, and delete) operations on the underlying data. Information is transferred in JSON. Below, we outline the key entities of the database schema (Figure 11) and an explanation of their relationships:

a. Player Table

The system supports the registration of new users. Minimal requirements for this action are the email, username, and password, as well as certain demographic information. The *visits* and *places* tables are used to record the interactions of the user with the *scenes*. Each user has a collection of monuments visited or saved. The *player_plays_in_levels* table holds the score of each player for each level and is used to provide level specific leaderboards.

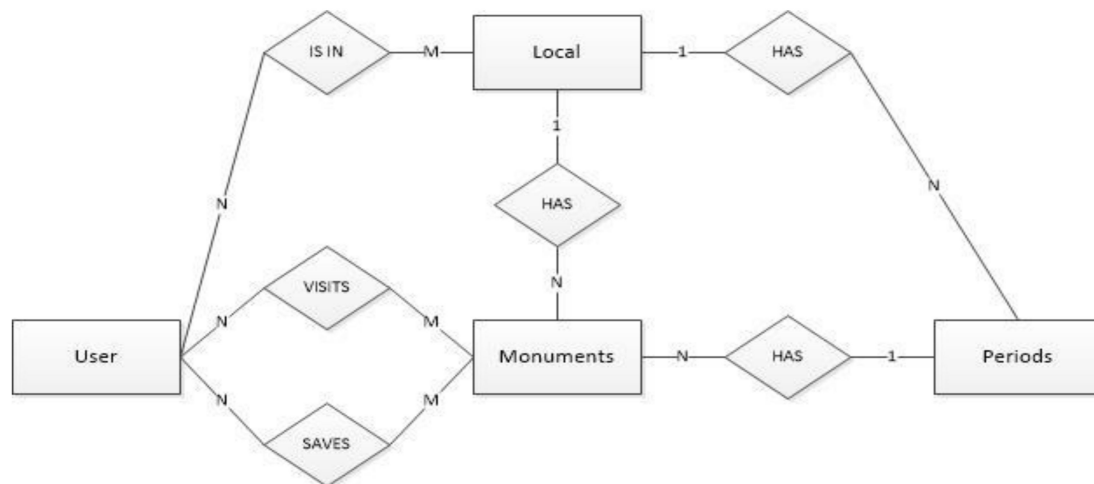


Figure 10. Entity Relationship diagram of the database.

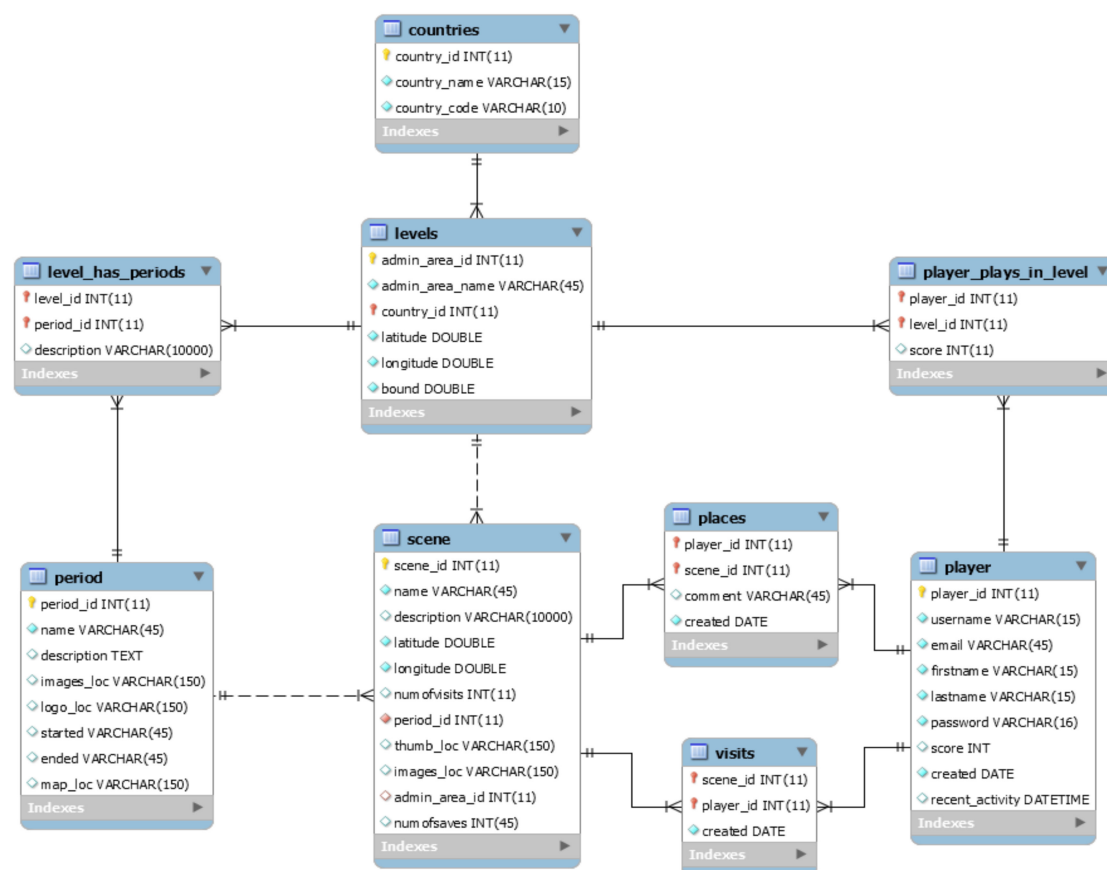


Figure 11. Database schema.

b. Scene Table

The *scene* table holds the records concerning the monuments. Each monument is uniquely identified with an auto-incremented ID. For each monument, the system records its name, description, latitude, and longitude values, as well as relative paths to its images. Similar to the players table, the *visits* and *places* entities keep track of additional statistics about each monument. Each monument can be a member of only one level (explained below) and relates to only one historical period.

c. Period Table

The *period* table holds the information concerning the historical periods used to classify the monuments. Each period has a name, description, paths to its images, and ended and started dates. The period table is used to classify the monuments and provide additional historical information about the *levels*.

d. Levels Table

The *levels* table is used to identify the playable areas that the system supports. Each level has a location described in latitude and longitude values, as well as a radius (*'bound'*) that sets its boundaries. Each monument in the database corresponds to a playable area and each area can relate to a number of periods defined by the *level_has_periods* table. In the current state, the database only holds information about the city of Chania, Greece, but it can be easily expanded to additional areas.

4.4. Augmented Reality Activity

The AR activity was implemented based on the Wikitude JavaScript API. The Wikitude SDK is based on web technologies (HTML, JavaScript, CSS). In order to integrate the web view with Android, the SDK provides a specific view component called *ARchitectView*, which we add to the activity's layout. Typical HTML pages are then loaded, located in the assets folder utilizing the API to create objects in AR. The first step is integrating the API with the activity's lifecycle. During the *onCreate()* call, the *ARchitectView* object is initialized and an interface is created that communicates with the pages. Information is transferred in JSON, parsing relevant arguments sent or received by the AR experiences.

The AR activity is started from the map activity with an *intent*. That intent contains a key-value pair specifying the AR experience (HTMLpage) selected to be loaded. Three AR experiences are designed, that is, the *ARNavigation*, *3DModelAtGeoLocation*, and *InstantTracking*. Each one is a separate HTML page loaded at the point of creation of the *ARActivity*. After initialization, the activity binds to the *LocationService* to receive location updates. The acquired locations are sent to the *ArchitectView* to draw the AR objects on the screen. A single native AR activity runs each AR page and is responsible for initializing each one in a different manner depending on the intent passed on by the *MapsActivity*. Creating each AR experience follows the standard web development process. The user interface (UI) includes a 3D scene containing the AR objects and standard 2D elements designed with JQueryryin order to provide additional control over the content.

The AR navigation page displays camera screen scenes at their geo-locations. After the page is loaded, a JSON array is provided including scene information derived from the native code. The array is then parsed and for each scene, a marker object is created to be displayed on the screen. The marker object provides its own logic in order to animate its changes between selected and deselected states using *AR.PropertyAnimations*.

5. Interface and User Experience

The main goal is to create an application that provides visitors of a historical city a MAR sightseeing experience including easily accessible historical information. On-site 3D depictions of monuments are provided in a MAR context, as well as information such as text and images. A complete location-aware experience promoting the exploration of a city's cultural wealth is put forward, addressing key technological challenges related to registration and tracking of AR technologies. In this section, we present the final screens of the application and explain in detail the flow of the experience.

Upon the activation of the application, the user is welcomed by a splash screen and is requested to create an account or login with an existing one. After the login process, the application checks the user's location and requests the POIs and the historical information relevant to the area from the server while it transfers to the main screen. The map activity is the main screen of the application and facilitates the core of the functionality, as shown in Figure 12. POIs are displayed on the map in

their corresponding geo-locations. By clicking on a marker, the user views the info window of the POI containing a thumbnail, and the distance between POIs. By interacting with the bar at the bottom of the screen, the user can navigate to the remaining pages of the application. These include the profile, leaderboards, library, and preferences, while the middle round button is used to swap between map and camera navigation.

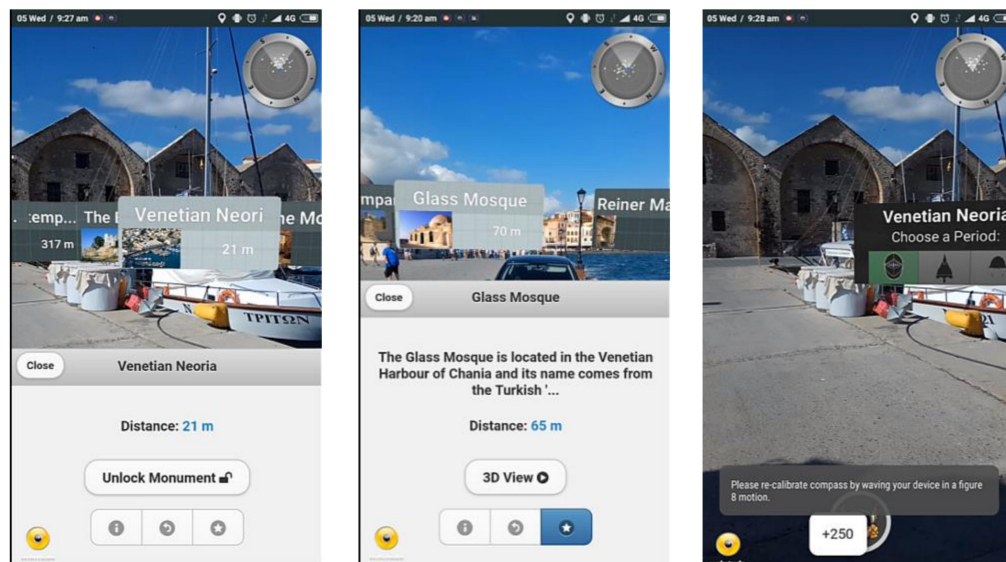


Figure 12. Navigation and classification in the camera view.

In the camera view, the content is displayed as 2D labels that contain basic information concerning the POIs. The POIs are displayed as dots on the radar to assist navigation. Users can save the POIs for later reference, access the reconstruction if available, or focus the marker and return to the map. The main goal of the experience is to reveal the available information with easy to control navigation conditions in order to facilitate efficient interaction. The UI displaying map activity is shown in detail in Figure 12.

The POI information is stored in the monuments table (Figure 13). It contains information concerning latitude, longitude, name, brief description, description, image locations, periods, and locality. The monuments are delivered on a local basis and are classified in periods to help manage the content. The locality is used to cluster the monuments in playable areas for the users. Their structure complies with ISO 3166-1 and combines countries and admin areas to ensure uniqueness. The server receives a location request by the client and then queries the database to find the POI information. The users can see the monument's specific information by selecting the items on the list, as shown in Figure 13. In this activity, the user can acquire historical information, including text and images; mark and save monuments; or get directions to specific locations.

In order to facilitate the client–server communication, a REST-full web service using is developed [38]. If such information exists, it is sent back to the server. The API exposes its resources via unique custom-defined URLs held by the mobile client. Each entity is defined as a resource and has its own URL. The service maps each resource with an implementation for varied HTTP request methods (GET, PUT, POST, and DELETE). By accessing these URLs and defining the HTTP method, the clients can then perform all CRUD (create, read, update, and delete) operations on the data. The NM relationships in the diagram are defined as sub-resources exposed by their parents links. Visits and places, for example, can be accessed from the users and monuments resources. For example, a client can make a get request to personalized visits or a get request accessing the overall amount of visits of a monument. Each response from the main resources contains links to sub-resources providing the client with the necessary information for communicating and enabling the server to evolve independently. Information is transferred in JSON.

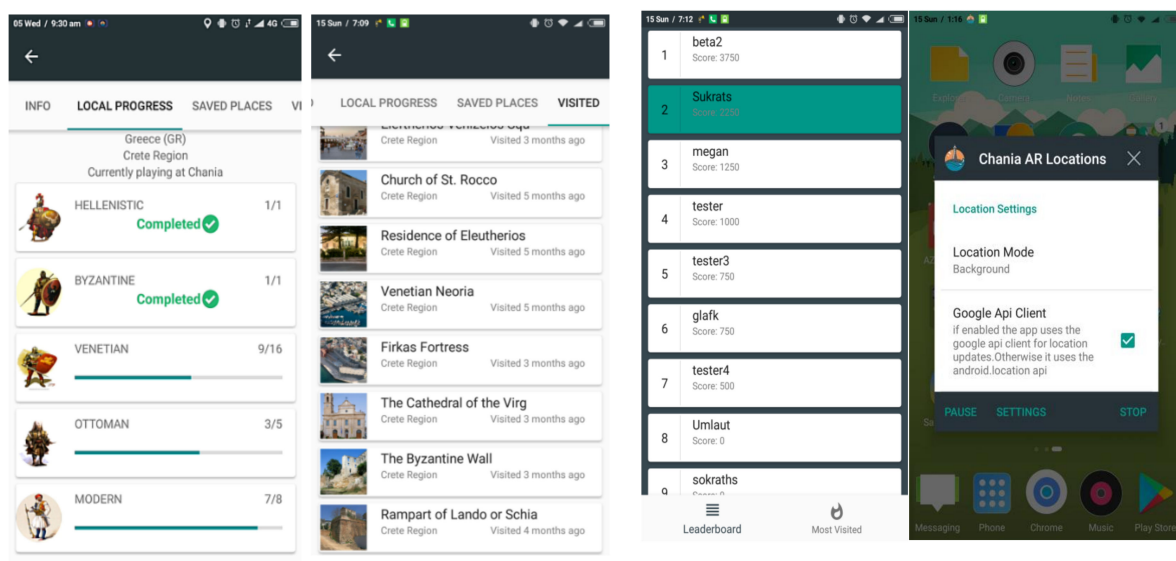


Figure 13. Local standings (left). Background service controls (right).

During the initial state of the application, the user is shown the available 3D models representing the historical monuments on the map. After visiting the monuments and viewing them in AR, the rest of the POIs' locations are unlocked and displayed on the map as question marks, shown in Figure 14. The goal is to visit them and classify them to the historical periods based on their architectural characteristics and clues obtained in the library page, as well as from the already visited monuments. The exploration of the POIs can be conducted freely by employing the background service of the application. The users can enable or disable this functionality from the settings. The aim of this approach is to urge the visitor to closely observe the monuments, revisit and consult the information available, and even interact with each other and locals to make the decision. The more areas they visit and unlock, the more points they earn for themselves. The overall progress in the city can be viewed in the leaderboards page, accessible from the map view.

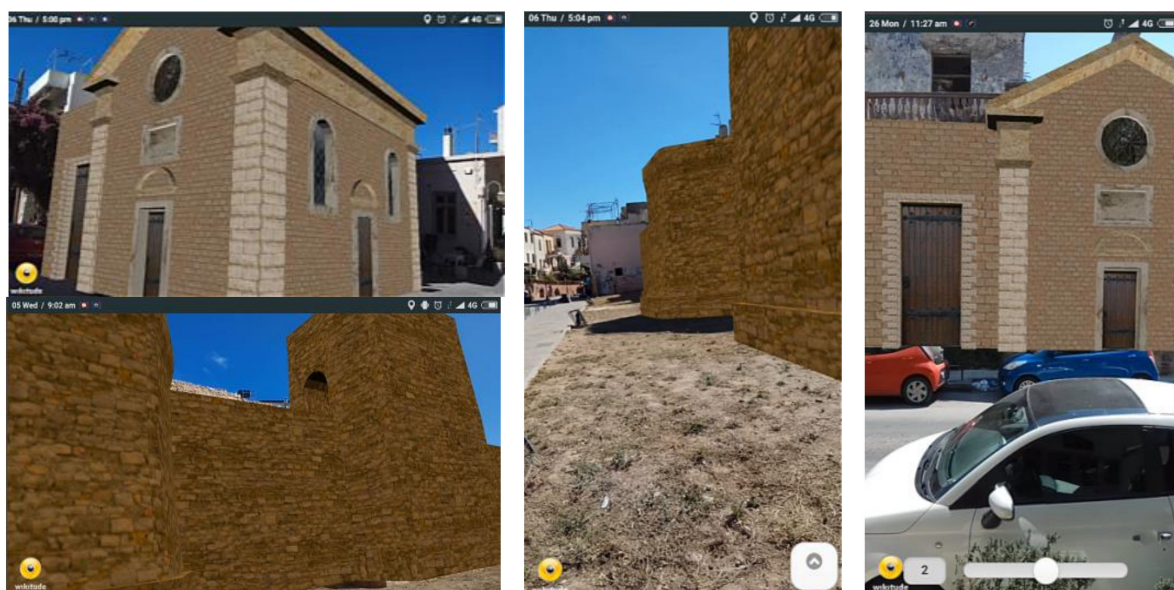


Figure 14. 3D depictions of the demolished towers of the Byzantine wall and the facial restoration of the Rocco temple.

The 3D models of the historical monuments are the main feature of the MAR activity, as shown in Figures 14 and 15, situating 3D information in the context of the respective real-world surroundings. The 3D models are accessed from the navigation views. When the user is in close proximity to the monuments, the location event handler informs the views to update the content and enable the AR experiences. In this screen, a reconstructed 3D model of the monument is overlaid on the camera and the GPS and inertial sensors are exploited to display the monument on its real-world location. The users can freely move around the real-world site to view the monuments from all available angles. They can click on the model to get information or access the slider, available at the bottom right, to change the representation. The user can shift between visualizing the full-scale 3D model or specific reconstructed parts. It is possible (but not in this case) to download multiple reconstructions relevant to one building, potentially with each one representing more than one historical period, signifying the monument's evolution over time.

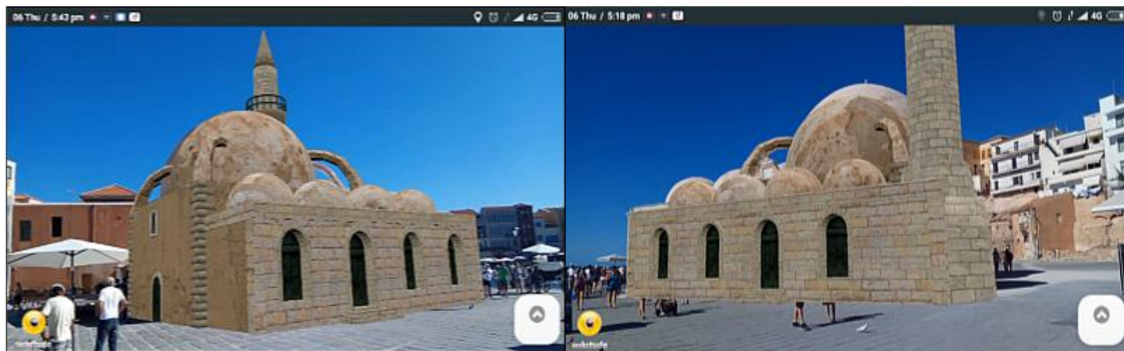


Figure 15. 3D depiction of the Glass Mosque featuring the now demolished minaret, as seen by the mobile's camera [36].

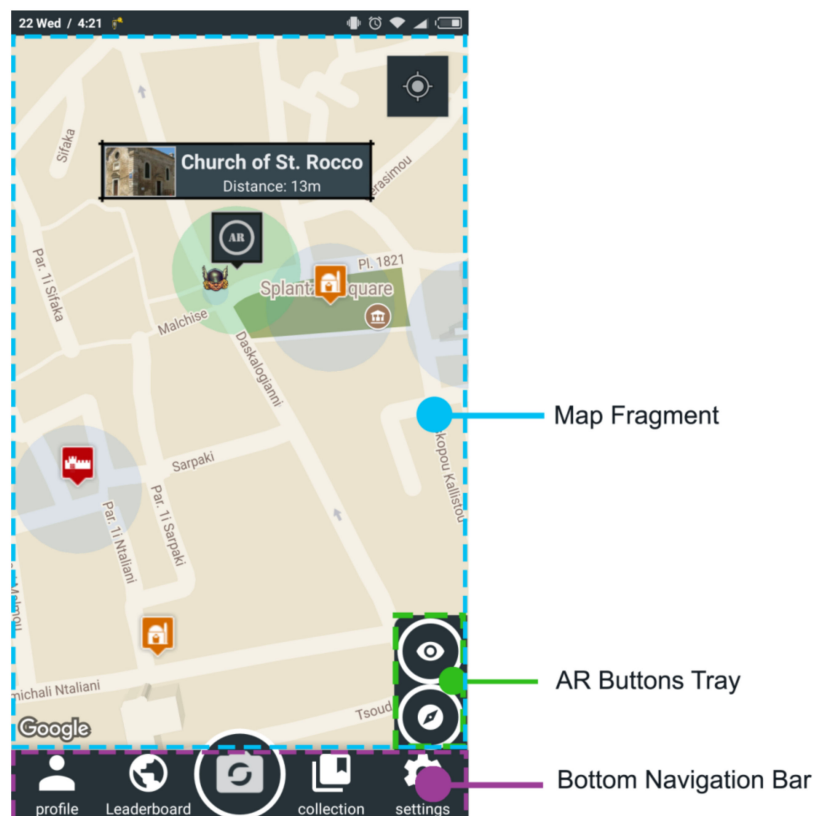


Figure 16. Map activity user interface.

The library page is where a collection of historical information is displayed. It consists of a view pager containing historical periods in chronological order. Each page has a historical briefing consisting of an image showing the active area for that period and a list including the monuments that have been correctly classified. The locked monuments are contained in a separate list at the end of the page.

The profile and leaderboard page follow the same architecture. In the profile page, the user can swap through pages containing personalized information, as well as observe local progress and lists of saved and visited places. By clicking on the progress plates, the users can be transported to the library for the selected period. By selecting a monument, users are transferred to specified detail pages unique to each monument. The leaderboard page can be used to get information about the city. The user's position is highlighted on a list showing the current standings for the city. User progress can be compared with that of other visitors (Figure 16).

6. Evaluation

6.1. Technical Evaluation

Field tests were conducted by employing two devices, that is, a Samsung Galaxy S3 Neo (Android 4.4, RAM 1.4 GB, 1.2 GHz quad-core, Wi-Fi, GPS, geo-magnetic sensor, accelerometer, gyroscope) and a Xiaomi Redmi note 4xAndroid 6.0, RAM 4GB, 2GHz octa-core, GPS, Wi-Fi, geo-magnetic sensor, accelerometer, gyroscope). A large difference between the two devices concerning the accuracy of their GPS receivers was observed. The S3 Neo presented an average accuracy of eight meters, while the Redmi Note presented the expected accuracy of three meters. When placed at wider areas, both devices performed better, 3m for S3 Neo and 1.4m for Redmi Note. Although geo-location accuracy represents the confidence level of the receiver and is not precise, actual measurements remained consistent with the acquired values. This indicates that depending on the deployed device, the accuracy of the resulting AR registration may vary independent of the underlying implementation. In relation to orientation tracking, both devices performed as expected. As both devices are equipped with the same sensors, they performed similarly during the instant tracking testing.

A well-known issue concerning the instant tracking method is the inconsistency between our calculation of the device's orientation and the respective orientation calculated by the API. This measurement has been observed to vary by up to five degrees when the geo-magnetic sensor reports low accuracy values. This error is minimized by recalibrating the sensor while moving the device in an 'eight' figure motion. The user is informed when the sensor accuracy is low.

Another well-known AR issue occurs when the users leave the AR application's screen during tracking; for instance, when the phone rings. The users are required to return to the AR camera view and reinitiate tracking.

6.2. User Evaluation

In order to evaluate the resulting application, continuous informal usability tests were conducted throughout the development of the system, adopting the 'think aloud' method, while users experienced early versions of the app, on-site and in the lab. Quantitative evaluation metrics were not employed because of the complexity of the MAR experience in an outdoor, warm, busy environment. Detailed user comments were recorded.

The more persistent user comments discussed the accuracy of registration of the AR camera view during navigation. Most users found that the use of the camera instead of the map limited their movements and perception of their surroundings and refrained from using it besides when locating specific sites and classifying the monuments. During the classification process, the AR camera was useful as it helped locate the specified monument.

The AR camera in conjunction with the constant usage of the GPS for extended time periods resulted in rapid battery consumption. Addressing such issues, the AR camera view was defined as a standalone activity instead of a replacement to the map in the main activity.

User comments drove the development of the gallery as well as the 'save' option placed on the monument details page. The implemented functionality allowed users to save monuments from either the list in the collections screen or from the info window when pressing on the marker at the map screen. The info window functionality was mostly missed and the users found it useful to be able to save a monument to their visited places when browsing through its details. Therefore, this was transferred to the details page, while the list functionality remained as is. In relation to the gallery option, the initial design contained a link, which when pressed, the users transferred to a full screen view of the images. The users suggested that a gallery region should be included that shows the images in the details page and when an image is pressed, the application will then proceed to the full screen view of the images. The implemented region follows a carousel fashion where the users can swipe right and left in order to browse through the images. When an image is pressed, it is brought forward in full screen.

In relation to users' views regarding the AR experiences, users' reactions were quite positive. Most users had never been acquainted with a similar capability through their smart phones and were excited to view specific 3D reconstructions superimposed on real-world surroundings, visualized on their mobile phone's screen. Although the registration difficulty of accurately aligning 3D content was commented on by most users, the geo-location approach proved easy-to-use and intuitive. The instant tracking method proved challenging for an unaccustomed audience. However, after initial explanation and minimal guidance, users proceeded to experiment with 3D models placed on the annotated area, overlaid over the real-world building they represented, in its past state.

7. Conclusions

In this paper, we presented the design of a MAR application for consumer-grade mobile phones with the ultimate goal of increasing the synergy between visitors and cultural heritage sites. In addition to exploring standard web and application content, the proposed platform offers a novel approach for visualizing historical information on-site. By employing 3D models of the historical monuments through AR, user experience is enhanced, bridging the gap between digital content and the real-world environment. An expandable platform was put forward that can easily envelop additional historical sites requiring little preparation. The proposed system will enable future experts to display their digitized collections using varied forms of data presentation.

Although outdoor mobile augmented reality presents several localization and registration challenges, it provides novel experiences to a wide audience. The availability and technological advances of modern smartphones allow for the development of reliable AR experiences that enhance the understanding of historical datasets.

AR is a field that has just reached the wider public. Mobile AR is based on the limited processing power and tracking characteristics of today's technologies, making robust AR hard to achieve. However, AR is rapidly evolving with the addition of sophisticated tracking and registration algorithms, as well as specialized hardware.

Tracking and registration in AR are far from solved. Plugins that allow the development of AR applications based on game engines such as Unity3D and the Unreal engine, as well as new platforms for developers and novel displays, would allow a new generation of sophisticated interactions and high fidelity graphics, and would greatly improve user experience.

User-friendly authoring tools in order to create and publish content would contribute to AR's wide adoption. Experts could create new AR scenes by uploading media assets and placing them on a map, while users would be able to browse through the geo-located information and even create and modify their own content. By reducing the size of the assets, AR experiences could then be available over the Internet.

In its current state, the application presented in this paper does not support social interactions between the users, apart from the posted overall rankings. Extending the platform to include comments,

likes, shares, and so on by adding a communication layer would increase users' interest in heritage sites and enhance motivation to engage with cultural applications.

Gamification around heritage sites and applications is gaining increasing value as it engages visitors and allows for new means of interacting with cultural heritage information. AR and VR are tools that the cultural heritage community could exploit so that interaction with historical datasets can be complemented by new forms of visualization and interaction. Combined with scavenging and treasure hunts, location-aware storytelling could enhance exciting immersive experiences related to cultural heritage awareness and increase visitor involvement and engagement.

Author Contributions: C.P. is responsible for data analysis, carrying out the on-site development, and implementation of the software. L.R. verified the geo-location methods. K.M. supervised the technical development of this work and provided the theoretical formalism. D.D. planned the historical routes and checked the historical validity of the imagery results. All authors discussed the results and contributed to the final manuscript.

Funding: This research received no external funding.

Acknowledgments: In this part, we would like to thank Wikitude for providing us with an educational license for their Augmented Reality SDK.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Milgram, P.; Kishino, F. A taxonomy of mixed reality visual displays. *IEICE Trans. Inf. Syst.* **1994**, E77-D, 1321–1329.
2. Milgram, P.; Takemura, H.; Utsumi, A.; Kishino, F. Augmented reality: A class of displays on thereality-virtuality continuum. In Proceedings of the Telemanipulator and Telepresence Technologies, Boston, MA, USA, 31 October–1 November 1994; Volume 2351, pp. 282–292.
3. Mekni, M.; Lemieux, A. Augmented reality: Applications, challenges and future trends. In Proceedings of the 13th International Conference on Applied Computational Science of Recent Advances in Computer Engineering Series, Kuala Lumpur, Malaysia, 23–25 April 2014; Volume 20, pp. 205–214.
4. Kersten, T.P.; Tschirschwitz, F.; Deggim, S. Development of a virtual museum including a 4d presentation of building history in virtual reality. In Proceedings of the International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences, 2017 3D Virtual Reconstruction and Visualization of Complex Architectures, Nafplio, Greece, 1–3 March 2017; Volume XLII-2/W3, pp. 361–367.
5. Kersten, T.P.; Tschirschwitz, F.; Lindstaedt, M.; Deggim, S. The historic wooden model of Solomon's Temple: 3D recording, modelling and immersive virtual reality visualisation. *J. Cult. Herit. Manag. Sustain. Dev.* **2018**. [CrossRef]
6. Azuma, R.T. A survey of augmented reality. *Presence* **1997**, 6, 355–385. [CrossRef]
7. Amin, D.; Govilkar, S. Comparative Study of Augmented reality SDKs. *Int. J. Comput. Sci. Appl.* **2015**, 5, 11–26.
8. Mosque YialiTzami. Available online: <http://www.explorecrete.com/chania/EN-chania-07-mosque.html> (accessed on 10 July 2018).
9. Church of Saint Rocco. Available online: http://www.chaniahistory.gr/index.php?option=com_k2&view=item&id=208:church-of-st-rocco&Itemid=1474&lang=en (accessed on 10 July 2018).
10. Byzantine Wall of Chania. Available online: <https://www.kastra.eu/castle.php?kastro=kydonia> (accessed on 10 July 2018).
11. Hildebrand, A.; Daehne, P.; Christou, I.T.; Demiris, A.; Diorinos, M.; Ioannidis, N.; Almeida, L.; Weidenhausen, J. Archeoguide: An Augmented Reality based System for Personalized Tours in Cultural Heritage Sites. In Proceedings of the International Symposium of Augmented Reality, Munich, Germany, 5–6 October 2000.
12. Stricker, D.; Karigiannis, J.; Christou, I.T.; Gleue, T.; Ioannidis, N. Augmented reality for visitors of cultural heritage sites. In Proceedings of the International Conference on Artistic, Cultural and scientific Aspects of Experimental Media spaces, Bonn, Germany, 21–22 September 2001.

13. Vlahakis, V.; Ioannidis, N.; Karigiannis, J.; Tsotros, M.; Gounaris, M.; Stricker, D.; Gleue, T.; Daehne, P.; Almada, L. Archeoguide: An augmented reality guide for archaeological sites. *IEEE Comput. Gr. Appl.* **2002**, *22*, 52–59. [[CrossRef](#)]
14. Vlahakis, V.; Karigiannis, J.; Tsotros, M.; Gounaris, M.; Almeida, L.; Stricker, D.; Gleue, T.; Christou, I.T.; Carlucci, R.; Ioannidis, N. Archeoguide: First results of an augmented reality, mobile computing system in cultural heritage sites. In Proceedings of the 2001 VAST '01 Conference on Virtual Reality, Archeology, and Cultural Heritage, Athens, Greece, 28–30 November 2001; ACM: New York, NY, USA, 2001; pp. 131–140.
15. Haugstvedt, A.C.; Krogstie, J. Mobile Augmented Reality for Cultural Heritage: A Technology Acceptance Study. In Proceedings of the IEEE International Symposium on Mixed and Augmented Reality, Science and Technology, Atlanta, Georgia, 5–8 November 2012; pp. 247–255.
16. Noh, Z.; Sunar, M.S.; Zhigeng, P. A Review on Augmented Reality for Virtual Heritage System. In Proceedings of the Edutainment '09 4th International Conference on E-Learning and Games: Learning by Playing, LNCS 5670, Banff, AB, Canada, 9–11 August 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 50–61.
17. Zollner, M.; Keil, J.; Wüst, H.; Pletinckx, D. An augmented reality presentation system for remote cultural heritage site. In Proceedings of the 10th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST 2009, Julians, Malta, 22–25 September 2009; Debattista, K., Perlingieri, C., Pitzalis, D., Spina, S., Eds.; University of Malta: Msida, Malta, 2009.
18. Sylaiou, S.; Mania, K.; Karoulis, A.; White, M. Exploring the Relationship between Presence and Enjoyment in a Virtual Museum. *Hum. Comput. Stud.* **2010**, *68*, 243–253. [[CrossRef](#)]
19. Wojciechowski, R.; Walczak, K.; White, M.; Cellary, W. Building virtual and augmented reality museum exhibitions. In Proceedings of the 9th International Conference on 3D Web Technology, Monterey, CA, USA, 5–8 April 2004; pp. 135–144.
20. Niedermair, S.; Ferschin, P. An Augmented Reality Framework for On-Site Visualization of Archaeological Data. In Proceedings of the 16th International Conference on Cultural Heritage and New Technologies, Vienna, Austria, 12–15 November 2011; pp. 636–647.
21. Choudary, O.; Charvillat, V.; Grigoros, R.; Gurdjos, P. MARCH: Mobile augmented reality for cultural heritage. In Proceedings of the 17th MM '09 ACM International Conference on Multimedia, Beijing, China, 19–24 October 2009.
22. Papagiannakis, G.; Magnenat-Thalmann, N. Virtual Worlds and Augmented Reality in Cultural Heritage Applications. In *Recording Modeling and Visualization of Cultural Heritage, Proceedings of the International Workshop, Centro Stefano Franscini, Monte Verita, Ascona, Switzerland, 22–27 May 2005*; Baltsavias, M., Gruen, A., van Gool, L., Pateraki, M., Eds.; Taylor and Francis Group: Boca Raton, FL, USA, 2006; pp. 419–430.
23. Papagiannakis, G.; Magnenat-Thalmann, N. Mobile Augmented Heritage: Enabling Human Life in Ancient Pompeii. *Int. J. Arch. Comput.* **2007**, *5*, 395–415. [[CrossRef](#)]
24. Pacheco, D.; Wierenga, S.; Omedas, P.; Wilbricht, S.; Knoch, H.; Verschure, P.F.M.J. Spatializing experience: A framework for the geolocalization, visualization and exploration of historical data using VR/AR technologies. In Proceedings of the Virtual Reality International Conference, Laval, France, 9–11 April 2014.
25. Julier, S.J.; Schieck, A.F.; Blume, P.; Moutinho, A.; Koutsolampros, P.; Javornik, A.; Rovira, A.; Kostopoulou, E. VisAge: Augmented reality for heritage. In Proceedings of the 5th PerDis'16 ACM International Symposium on Pervasive Displays, Oulu, Finland, 20–22 June 2016.
26. Lee, J.Y.; Lee, S.H.; Park, H.M.; Lee, S.K.; Choi, J.S.; Kwon, J.S. Design and implementation of a wearable AR annotation system using gaze interaction. In Proceedings of the Consumer Electronics (ICCE), Digest of Technical Papers, Las Vegas, NV, USA, 9–13 January 2010; pp. 185–186.
27. Hable, R.; Rößler, T.; Schuller, C. evoGuide: Implementation of a tour guide support solution with multimedia and augmented-reality content. In Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia, Ulm, Germany, 4–6 December 2012.
28. Fritz, F.; Susperregui, A.; Linaza, M. Enhancing cultural tourism experiences with augmented reality technologies. In Proceedings of the 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST), Pisa, Italy, 8–11 November 2005.
29. Martinez, M.; Munoz, G. Designing augmented interfaces for guided tours using multimedia sketches. In *MIXER*; Citeseer: University Park, PA, USA, 2004.

30. Střelák, D.; Škola, F.; Liarokapis, F. Examining User Experiences in a Mobile Augmented Reality Tourist Guide. In Proceedings of the 9th PETRA '16 ACM International Conference on Pervasive Technologies Related to Assistive Environments, Corfu Island, Greece, 29 June–1 July 2016.
31. Ragia, L.; Sarri, F.; Mania, K. Precise photorealistic visualization for restoration of historic buildings based on tacheometry data. *J. Geogr. Syst.* **2018**, *20*, 115–137. [[CrossRef](#)]
32. Azuma, R.T.; Baillot, Y.; Behringer, R.; Feiner, S.; Julier, S.; MacIntyre, B. Recent advances in augmented reality. *IEEE Comput. Gr. Appl.* **2001**, *21*, 34–47. [[CrossRef](#)]
33. Liarokapis, F.; Brujic-Okretic, V.; Papakonstantinou, S. Exploring Urban Environments using Virtual and Augmented Reality. *J. Virtual Real. Broadcast.* **2006**, *3*, 1–13.
34. Guidi, G.; Russo, M. Diachronic 3D reconstruction for lost Cultural Heritage. International Archives of the Photogrammetry. *Remote Sens. Spat. Inf. Sci.* **2011**, *38*, 371.
35. Rodríguez-Gonzálvez, P.; Muñoz-Nieto, A.L.; del Pozo, S.; Sanchez-Aparicio, L.J.; Gonzalez-Aguilera, D.; Micoli, L.; Haynes, I. 4D Reconstruction and visualization of Cultural Heritage: Analyzing our legacy through time. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *42*, 609. [[CrossRef](#)]
36. Panou, C.; Ragia, L.; Dimelli, D.; Mania, K. Outdoors Mobile Augmented Reality Application Visualizing 3D reconstructed Historical Monuments. In Proceedings of the 4th International Conference on Geographical Information Systems Theory, Applications and Management, GISTAM, Madeira, Portugal, 17–19 March 2018; pp. 59–67.
37. Guidi, G.; GonizziBarsanti, S.; Micoli, L.L.; Malik, U.S. Accurate Reconstruction of the Roman Circus in Milan by Georeferencing Heterogeneous Data Sources with GIS. *Geosciences* **2017**, *7*, 91. [[CrossRef](#)]
38. Fielding, R.T. Architectural Styles and the Design of Network-Based Software Architectures. Ph.D. Thesis, University of California, Berkeley, CA, USA, 2000.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).