https://doi.org/10.1007/s12652-017-0654-6

ORIGINAL RESEARCH

Engineering an advanced geo-location augmented reality framework for smart mobile devices

Ugo Erra[1] · Nicola Capece[1]

Abstract

We present a client–server framework for the development of mobile applications that use augmented reality to visualize geo-located data. We provide a scalable and flexible architecture for development and management of the client, server, and data that are used by the applications. The idea is based on the concept of a layer that contains information related to the real environment that is displayed by the device. Information that enhances reality is identified as points of interest by their geographic coordinates. Layers can also be organized in a hierarchical model as a tree-like structure. We present four case studies that demonstrate the flexibility and generalizability of our framework: management of electrical power line failures, support for hydrogeological monitoring, management of sensor networks, and geo-localization of tourist attractions.

Keywords Augmented reality · Geo-location augmented eeality · Mobile applications

1 Introduction

Augmented reality (AR) is an altered reality in which perceived normal reality is overlaid with artificial/virtual sensory information. In more formal terms, AR is the overlay of virtual elements generated by computer onto the visual perception of physical reality, through a camera (Azuma 1997). AR technology allows the user to see the real world augmented by virtual objects (Hincapie et al. 2011). In other words, AR provides an environment in which virtual and real objects coexist. Commercially, AR is mainly found in mobile applications and, in recent years, has attracted wider interest from business. It is a natural way to explore 3D objects and data, because it brings virtual objects into the real world (Geroimenko 2012). The fields of application are potentially endless, and include: the display of information (Carotenuto et al. 2014); marketing and advertising (Erra and Colonnese 2015); education and training (Wang et al. 2017); and games (Das et al. 2017).

Geo-location AR is the combination of AR, a global positioning system (GPS), and location-based services. The fundamental idea is to retrieve the location of a device by using GPS and then overlay onscreen the information about points of interest (POI) by using AR technology. Therefore, geo-location AR focuses on determining the position of a user and then overlaying the geographical information on the screens of dedicated devices in an interactive way, rather than just working with positioning services. Although geo-location AR has already been widely implemented in technologies, geo-location AR has made the leap into the mainstream media only recently. Some of the best AR applications available in the market are primarily location-based; one that is particularly famous is Pokemon Go (Wingfield and Isaac 2016). However, even though current usage is still largely centred on the gaming and entertainment industry, geo-location AR has the potential to revolutionize business processes and enterprise workflows (Global Industry Analysts 2015).

Several frameworks support the development of AR applications, and the best known are ARToolKit (Furht 2011), Wikitude (Madden 2011), and Metaio ("Metaio DOC doc description 2015). These SDKs use computer vision techniques to calculate the real camera position and orientation relative to square shapes or a flat textured surface, allowing the programmer to overlay virtual objects. The precise tracking provided by ARToolKit and Wikitude

✉ Ugo Erra
ugo.erra@unibas.it

Nicola Capece
nicola.capece@unibas.it

[1] Dipartimento di Matematica, Informatica ed Economia, Università della Basilicata, Viale dell'Ateneo Lucano 10, Potenza, Italy

enable the rapid development of interesting AR applications. However, these AR tools are designed primarily by computer scientists for computer scientists. They require the use of programming languages such as C or C++ and are a poor match for general work practices and for people who may also be interested in using it as an approach to software prototyping for geo-location AR applications.

In this paper, we focus on applications that display geo-localized data as virtual objects. These applications can run on devices equipped with a webcam and sensors such as a gyroscope, a compass, and a GPS. We have developed a software architecture that is built on top of Metaio (recently acquired by Apple) that can manage different layers. These layers contain information related to the real environment that is displayed by the device. Information that enhances reality is identified by a geo-marker. This is a real object that is recognized by the system (defined using geographic coordinates), and information is overlaid onto it. Our goal is to visualize these layers through virtual objects overlaid onto the real environment. Each layer contains a set of POIs, which include information about 2D/3D objects to be displayed as text, 3D models, textures, images, billboards, etc. associated with the real object. Layers can also be organized in a hierarchical model as a tree-like structure. On the root, there is a layer with the most import POIs. Each POI in this layer is connected to other layers that contain less important POIs, and so on. In order to validate our idea, we present also four augmented reality applications developed using our framework: (1) an application that supports the management of failures in electrical power lines; (2); an application designed to collect and analyze of water-level data measured by hydrometric stations in southern Italy (3) an application that supports the management of sensor networks in a smart city; (4) an application for tourists to find historical zones, restaurants, hotels, and other attractions in an unfamiliar city.

The work presented in this paper is based on the work presented in (Capece et al. 2016), where we preliminarily proposed our approach and its supporting software tool. The current paper extends the previous paper as follows:

1. An improved and extended description of the approach and tool used to enable the augmented virtual dressing;
2. A hierarchical model of the layers in order to have a better visualization of the POIs;
3. In addition to the applications (1) and (2), we present the application for the management of sensor networks (3) and an application for tourist (4).

The remainder of this paper is structured as follows. Section 2 provides an overview of related works. Section 3 provides some background information about the software tools that we used to develop the framework. Section 4 presents detailed information about our framework. Section 5 demonstrates the potential applications of our framework in four case studies. Section 6 concludes and suggests future directions for the research.

2 Related work

There are many frameworks that support the development of different AR applications. Here, we focus on location-based AR (LBAR) applications, which are now widely used. The popularity of LBAR has increased with improvements in smartphone functionality and sensor technology. Our system is mainly based on the use of GPS and a digital compass that, with the support of a geographic information system, provide information about a POI. The user points the device's webcam at the surrounding environment, and then, using inbuilt sensors, the system displays information about the POI via an overlaid geospatial tag environment. The user clicks on the tags to view pictures, video, and other media associated with the POI.

One of the main problems of developing LBAR applications is the organization of and interaction with geospatial tags. This is especially true where there are many POIs close to the user's location. Choi et al. (2010) suggests a way to organize geospatial tags within a scene.

Geiger et al. (2014) describes a method for developing an engine for LBAR applications giving profound insights into the design and implementation of such an advanced mobile application. Delić et al. (2014) proposes a method based on LBAR that enhances the user's view of his or her surroundings with information about the land parcels that lie within the visual field of the device. Geiger and Delić aim is to use AR technology for educational purposes, through the development of instructional applications. Their approach highlights another way to use LBAR through the real-time construction of virtual objects on the scene, using the Web Map Service specification defined by the Open Geospatial Consortium (Lupp 2011).

Other software such as Junaio (Madden 2011) shows virtual objects as images and labels, and a radar supports the search for POIs in the user's environment. This system is typically used for route planning. Images and videos can also be saved in a database for other applications (e.g., uploading and sharing on social media). Madden (2011) allows developers to create various features. Users can associate text messages with AR images, make calls that interact with virtual tags, send email, and plan routes.

However, in the utility sector, the need to work with 2D plans and the fact that assets are normally hidden or difficult to reach has limited the interest in AR. Nevertheless, the real environment visited by field workers is still three-dimensional. AR can provide a suitable alternative to 2D plans in the field work superimposing 3D graphics registered with the real world over a field workers view to see where underground or overground infrastructure lies (Schall et al. 2009). In addition, it can provide faster and accurate localization of the assets, thereby reducing the time of intervention simulating X-ray vision (Bane and Hollerer 2004). A work that specifically deals with an AR system for an underground asset is (Schall et al. 2002).

3 Our client–server approach

We propose an approach based on a client–server architecture. The client runs on the smart mobile device and has the goal of tracing the user location, sending it to the server, and rendering the virtual objects received from the server on the display. Conversely, the server receives the user's location, extracts from a database all the nearest POIs, and sends the virtual objects to the client. Figure 1 shows this client–server architecture. The advantage of this architecture is that it can be used in different application contexts. In particular, the client side allows the development of applications for mobile devices that display POIs using AR. The server side allows the development of web applications (web service) that handle requests from the client (AR browser) and return responses after querying a POI database. In more detail:

- The client interacts with a server application (web service) through requests made via representational state transfer (REST) messages.
- When the request is accepted, the web service queries the database and sends the retrieved information to the client as a JavaScript Object Notation (JSON) message.
- The client interprets the response and renders the information in the form of virtual objects, overlaying them onto the real environment being viewed on the screen.

We use several specialized frameworks for different tasks. The REST architecture (Fielding and Taylor 2002) is used for client–server exchanges. It enables the design of network-based applications in such a manner that their functionality is implemented as a set of resources. Each resource can be referred using a Uniform Resource Identifier. One of the advantages of the REST architecture is that it provides a uniform interface that enables client–server separation. The client, rather than the server, handles the user's state and the application interface. Consequently, both the client and the server are completely independent of the technology. The session state is maintained by the client and can be transferred to the server through other services. We use the REST architecture to handle the exchange of messages, specifically the Restlet (Louvel et al. 2012) framework, version 2.3. This framework is composed of two main parts: (1) the Restlet application programming interface (API): a neutral API that supports the principles of the REST architecture, facilitating call management within the client; (2) the Restlet engine: the implementation of the Restlet API. For client-side development, Restlet can interact easily with remote resources through its HTTP connector. The system analyses the representation of resources (in JSON format) and extracts information that needs to be saved in the model's objects.

The model contains the classes, with the properties corresponding to the table's tuples. When a HTTP request is made by the client, the server queries the database by using a persistence layer in the form of a data access object. A

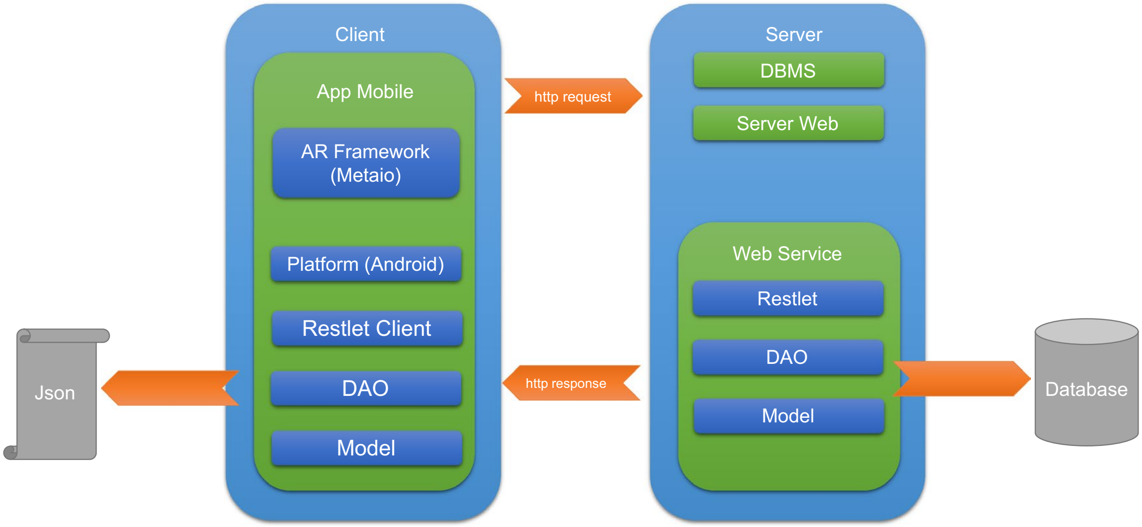


Fig. 1 The architecture of our client–server framework. *AR* augmented reality, *DAO* data access object, *DBMS* database management system

query is built for the correspondent database using such as Hibernate or others. The query's result is transformed into objects whose types are defined by the model's classes, and these objects are then stored in JSON format using libraries such as Jackson API or Gson. The same model is defined on the client side as on the server side; when the JSON response is obtained, the client converts it to the model's objects, which will be used in the application's context. For server-side development, we built additional HTTP connector, listening on port 7080.

Our software architecture uses the HTTP protocol to transmit data by defining architectural constraints, but not the implementation of components (Li 2011). Network resources are used to communicate network components that exchange resource representations. A representation is composed of (1) a sequence of bytes (the content); (2) metadata that describe the content; (3) metadata that describe this metadata (e.g., hash sums) (Wilde and Pautasso 2011). A connector (a client or server) mediates communication between components. It enables the application to interact with a resource, given its identifier and the action to be taken. The application interprets the answer and determines how to represent the information (e.g., as an XML document or JSON). Components are identified by their roles within the application and can be classified as either (1) user agents: these use a client connector to instantiate a request, then receive the response (e.g., a web browser); (2) origin servers: these use a connector server that receives the request and provides the representation of its resources (e.g., Apache Tomcat); (3) intermediate components: these can operate as a client and as a server and support the translation of both requests and responses (e.g., a gateway).

To represent the resource obtained through REST, we used the Metaio framework. In the context of our approach, the implementation is encapsulated, and the developer does not need to know the details of the acquisition, rendering, sensing, and tracking. Metaio uses standard graphics libraries for 3D scenes and rendering, in particular OpenGL ES API. This is a subset of the OpenGL graphics library that was designed for embedded devices such as smartphones (Shreiner and Group 2009).

4 Management of the POIs

A POI represents a specific point characterized by three parameters: latitude, longitude, and altitude. Then it is geolocalized according to location type such as shop, restaurant, hotel, road, or specific address. A POI can also be a location that is of particular interest to specialist users such as companies carrying out relief work or maintenance. In this case, the POI may be a critical area, or it may identify an at-risk area (e.g., of flooding). In our system, POIs are displayed on a mobile device using billboards that are overlaid onto a scene that is captured by the device's camera. The billboard is an icon that takes different shapes and colours based on the POI. Each billboard is associated with a label that displays information such as the distance between the device and the POI. In addition, multimedia can be associated with the POI, such as photos or videos, which are displayed when the user touches the icon on the mobile device. The diagram shown in Fig. 2 provides an overview of the structure of the POI database. The database can be populated through applications such as phpMyAdmin (Delisle 2009) for MySQL (Widenius and Axmark 2002). In the same case, we developed a tool that greatly simplifies the creation of POIs (Fig. 3). This tool enables the creation and management of POIs, the visualization of POIs by using their geographic coordinates, the addition of multimedia associated with the POIs, and the use of Google Maps to add new POIs. The tool we developed allows the user to interface at high level with the database. This tool has been developed in Java and uses Java Data Base Connectivity (JDBC) as a connector, in fact it uses different drivers based on the DBMS type to access and manage the data persistence layer. When the use context and the database are defined following our standards, using the concept of POI as central identity of the Data Base, the developed tool allows to create, visualize and modify the geolocalized data. Furthermore, allows to pair media as photos and videos in addition to textual information. The flexibility of the tool allows it to be used in multiple application domains because it defines an optimal interface for back-end data management and allows performing simulation on the effectivity of Augmented Reality in an application context.

The POI concept is central to our architecture. We need to be able to access information about POIs in situations where there is both good and poor network coverage. We use the concepts of online and offline modes to address these two cases.

In online mode, the system queries a remote database (e.g., MySQL) directly, using the available network. The app sends a request to the web service and provides the user's location and the maximum range (in m) within which the user wants to see all POIs. This distance range serves as a filter to prevent the user from being overwhelmed by too many POIs. The filter is implemented by applying the haversine formula as described below. It is also possible to apply additional filters (e.g., to focus on those POIs that are most relevant) by specifying the device's latitude and longitude, and a radius that represents the maximum distance to be used when locating POIs or, if the POI represents a status for monitoring the precise condition of an object, a filter that shows only the most critical POIs (e.g., high temperature or dangerous water level). The response is in JSON format. Whenever the user's location changes, the system sends a new request to the web service with the geographic

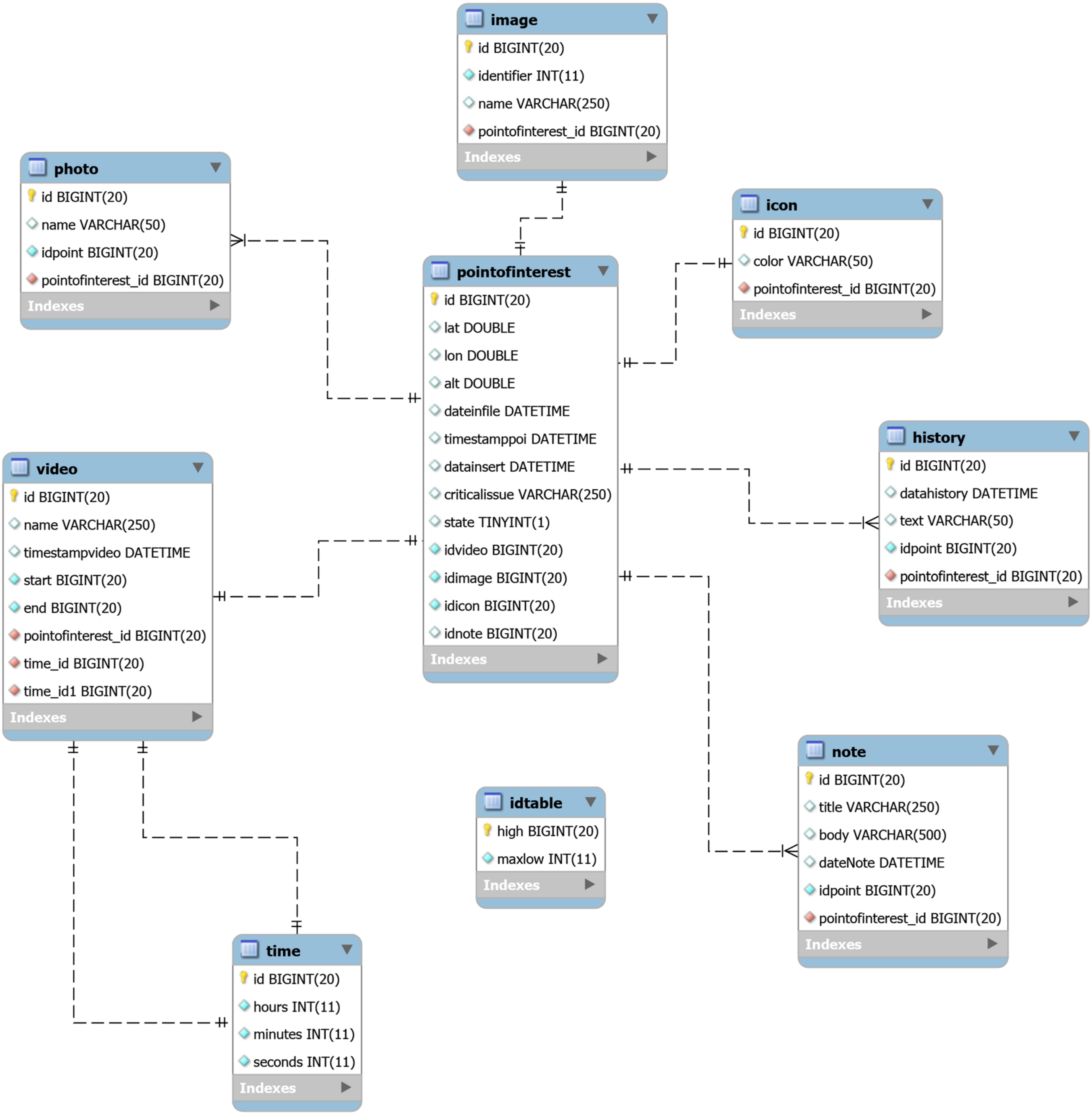
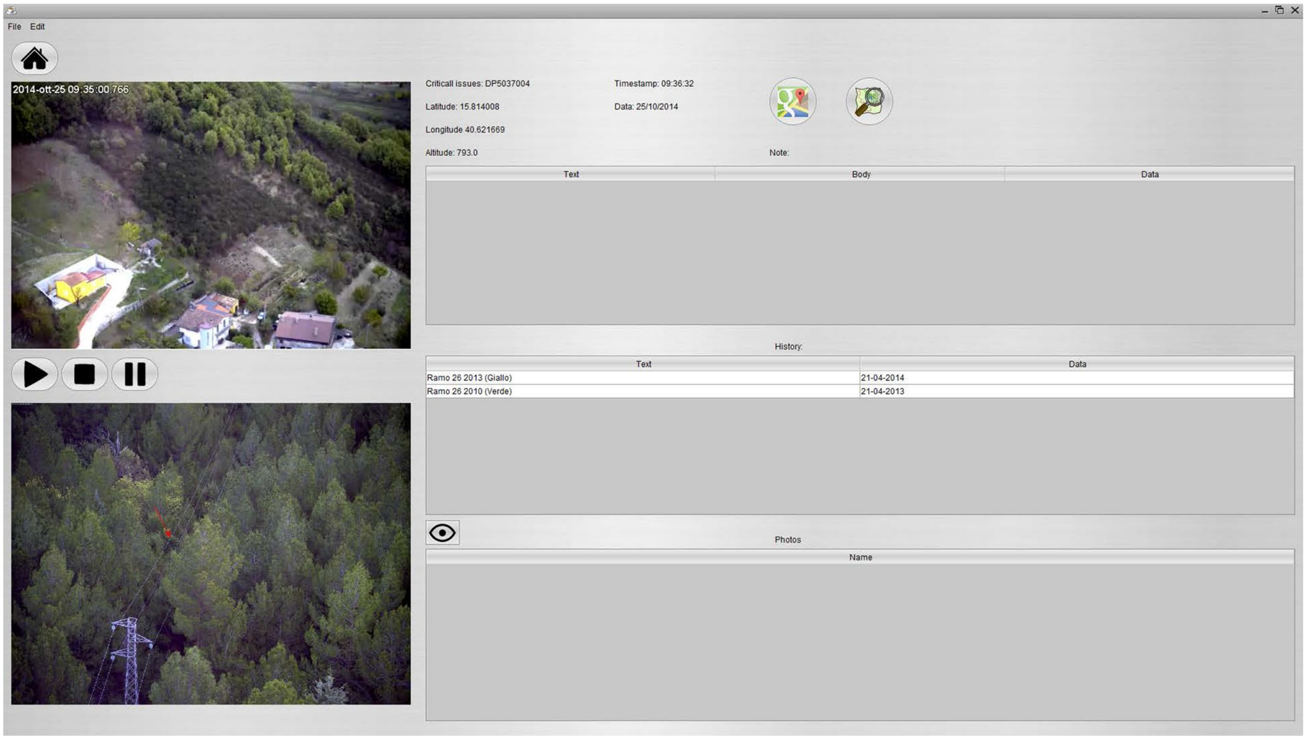


Fig. 2 Diagram of the structure used to manage the database of points of interest

coordinates of the current location of the device. Based on these parameters, the service returns all POIs that fall within the specified radius. When new POIs are found, the view is updated with the new information in real time. Additionally, as the user approaches a POI, the label displays the distance between it and the user, which is updated in real time. The web service receives the request and queries the database with the parameters given in the URL. The query is executed using a connector or drivers. Note that the database must initially contain at least one table recording POIs, and will eventually contain other tables relating to other types of information to be displayed.

In offline mode (e.g., when network coverage is poor) the system uses information that has previously been stored in a JSON file. The information to be displayed is provided in advance. The JSON file is automatically generated and lists all POIs. The user can also choose what information to display according to his or her needs. This requires the user



Criticall issues: DP5037004

Timestamp: 09:36:32

Latitude: 15.814008

Data: 25/10/2014

Longitude 40.621669

Altitude: 793.0

Note:

Text	Body	Data

History:

Text	Data
Ramo 26 2013 (Giallo)	21-04-2014
Ramo 26 2010 (Verde)	21-04-2013

Photos

Name

Fig. 3 Screenshot of the tool simplifying management of the point-of-interest database. It focuses on interactivity and is specifically designed to be used with the augmented electrical lines application (Sect. 5.1). In the example shown, an aerial video and photo are associated with a newly added POI

to have previously downloaded information about the POI (when the user had access to a mobile network). The user will also need to have downloaded any multimedia (images, videos, etc.) associated with the POIs. These data are saved locally in JSON format. The user can then view the available information without using the network and, when necessary, update the database by refreshing the information when a network becomes available.

4.1 Haversine formula

To visualize on the device's display only POIs in a certain range, we use the haversine formula (Sinnott 1984) to measure the distance between the user's location and each POI. The haversine formula is a method in spherical geometry to calculate the distance between two points on a curved surface such as Earth, given their longitudes and latitudes. These coordinates define a unique great circle divided into two arcs. The distance d between the two points is the length of the shorter arc. The haversine formula is defined as:

$$\mathrm{d} = 2r\arcsin(\sqrt{h(\vartheta_2 - \vartheta_1) + \cos(\vartheta_1)\cos(\vartheta_2)h(\lambda_2 - \lambda_1)}),$$

where

$$h(\theta) = \sin\left(\frac{\vartheta}{2}\right)^2 = \frac{1-\cos(\vartheta)}{2}.$$

Here, λ_1 and λ_2 are the longitudes and ϑ_1 and ϑ_2 are the latitudes of the first point and second point, respectively, and r is the radius of Earth. In our case, although we take into account short distances, the haversine formula is numerically better conditioned for short distances (Sinnott 1984). By using the haversine formula, querying the database to find all the nearest POIs that are within a certain distance from the user's location is straightforward.

5 Case studies

In this section, we describe four applications that have been developed using our client–server framework.

5.1 Augmented electrical lines

Most of the faults that occur in electrical transmission lines can cause a flow of heavy current associated with short circuits. These increase the chance of fire, damage to electrical equipment, and imbalance of the power supply. When a fault occurs, technical specialists must be deployed to the geographic area to locate and repair the electrical line. Sometimes these operations are difficult because electrical lines tend to be located in areas of dense vegetation and rugged terrain. In some cases the interventions must be carried out with the aid of an equipped helicopter.

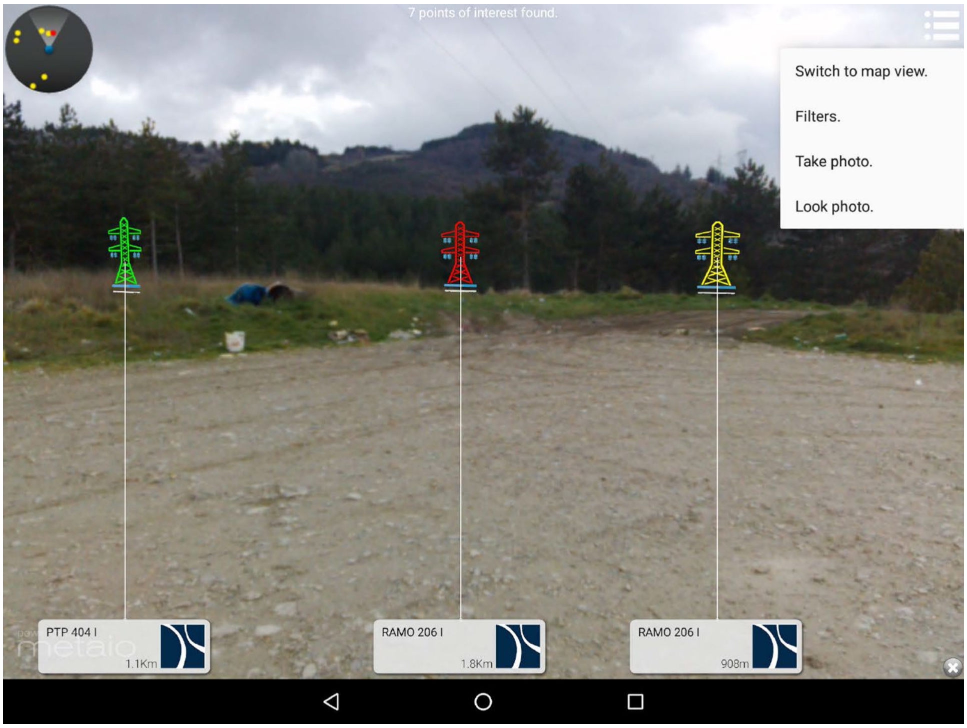
The first application designed with our framework supports the management of failures in electrical power lines. The idea is to identify in near real-time the network anomalies linked to their geographic coordinates. In particular, the aim is to provide information useful for locating high-voltage pylons and for producing an asset inventory of the power line components that require maintenance. These pylons are displayed in AR as billboards that are oriented toward the pylons. The POI is the pylon. Information is shown about the problem (e.g., an ID number or the date it was reported) and its location. Each POI can be associated with multimedia such as photos and aerial footage video obtained from helicopters.

As shown in Fig. 4, each billboard label provides a summary of important information: the ID of the problem, the distance from the user's current location to the pylon, and a colour indicating the operation's emergency level. In addition, the application visualizes a radar display (top left of figure) showing the direction of the pylon to help the user rotate the device in the correct direction. The user can interact with POIs (pylons) by pressing on the billboards. A popup (Fig. 5) provides the user with a number of options to perform further actions for each pylon, such as view more information or add notes.

5.2 Augmented waters

Floods are among the most devastating natural hazards in the world, claiming more lives and causing more property damage than any other natural phenomenon. According to the World Resources Institute (2015), the number of people at risk within flood-prone areas will rise from 21,000,000 in 2015 to 54,000,000 in 2030. Despite the increase in flood protection investments during the last two decades, populations tend to grow in socio-economically attractive flood-prone areas, and it is therefore likely that the vulnerability level of such areas will continue to increase. During a crisis, responding organizations have to deal with large uncertainties in making critical decisions. They need to gather situational information (e.g., state of civil, transportation, and information infrastructures) and information about available resources (e.g., medical facilities and rescue and law enforcement units). Timely and accurate information can radically transform the ability of these organizations to gather, manage, analyse, and disseminate information when responding to flood events. Technologically advanced systems can support emergency actions and mitigation.

We developed a mobile application that uses AR to help field workers choose the fastest paths to reach critical areas during an extreme event. It helps them to communicate the state of structures and infrastructures to the operating centre in real time and to identify the most suitable mitigation measure. In particular, the mobile application is designed to support the collection and analysis of water-level data



7 points of interest found.

Fig. 4 The augmented reality application for electrical power lines. The points of interest visualize the position of three pylons, with the colours indicating the status of each pylon. The yellow circle in the radar display (top left) indicates the positions of pylons, and the blue circle is the mobile device. (Colour figure online)

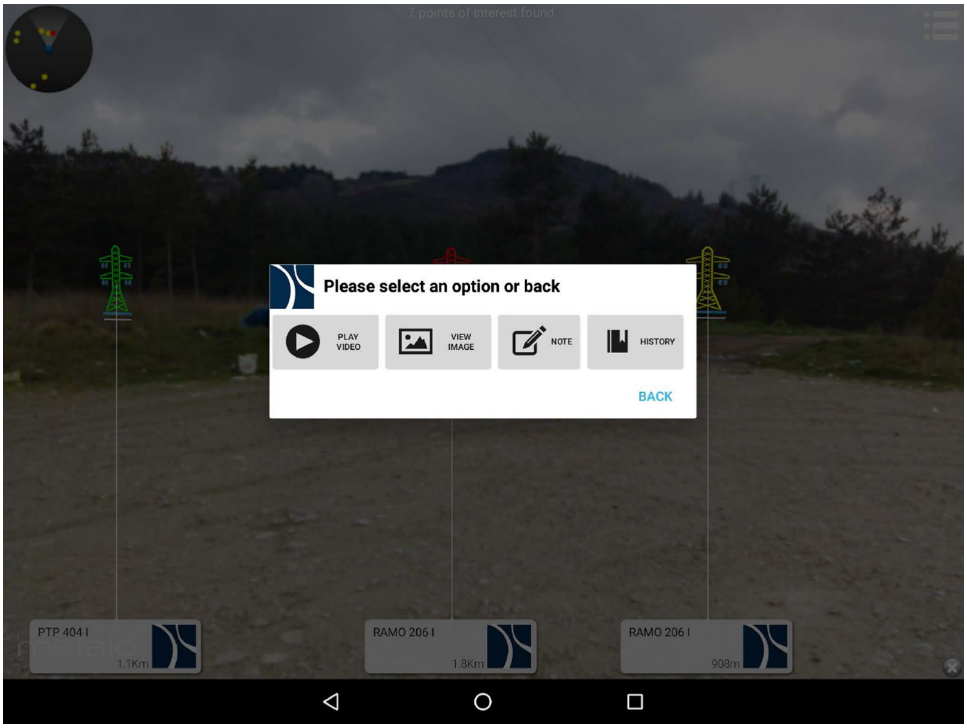


Fig. 5 Popup showing user options for the point of interest (pylon): view an image associated with the pylon; view an aerial video; view the pylon's history; add notes such as an intervention report

measured by stations in southern Italy. The application can show the location of stations and their hydrometric levels. More importantly, it can graphically display daily hydrometric values to provide a quick preview of changes in water levels. The POIs in this context are hydrometric stations, and associated information relates to the water level, the date of the last reading, and a graph showing changes in water level.

We evaluated the performance of the application in the middle valley of the Bradano river, in the south-eastern Basilicata region (Italy), both in a real case study and in a simulated one. The real case study included visiting the Bradano basin and locating POIs, for example, gauging stations, buildings, or bridges. Because in this application it is necessary to maintain contact with the real environment and identify and reach these POIs more rapidly, their locations are overlaid onscreen as coloured symbols (Fig. 6). For example, gauging stations are represented graphically as coloured droplets. This allows the user not only to visualize clearly the position of each POI, relative distance, and direction from the user, but also to know the level of criticality of the flood-prone areas at the station itself. Whenever the water level changes, the colour of the symbol changes according to the status of the danger. For example, if the values of water depth acquired by a gauging station have reached the level of ordinary, moderate, or high criticality, the droplet symbol will be yellow, orange, or red, respectively. In addition, to help a workforce to control a river and flood plain during an extreme event, the application can update the hydrometric level and show the trend of the flow

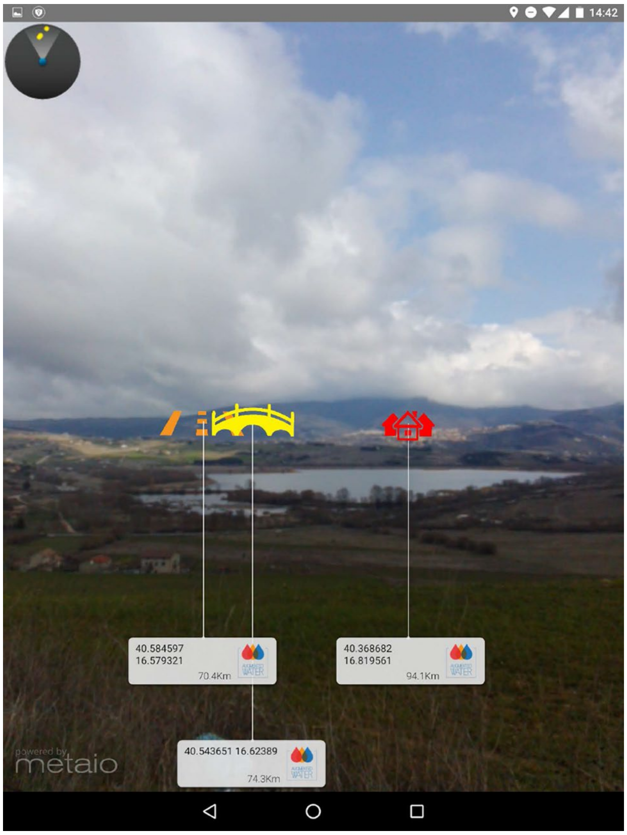


Fig. 6 The augmented reality application for hydrometric measurements. The three points of interest indicate the status of a street, a bridge, and a group of houses in a flood-prone or flooded area. Colours indicate the criticality of water levels. (Colour figure online)

depth recorded during the day. Finally, the application, using Google Maps, shows the direction to reach the location of the hydrometric station or that of the structure or infrastructure at risk of flooding.

5.3 Augmented sensors

A smart city uses digital technologies to enhance the quality and performance of urban services to reduce costs and resource consumption, and to engage more effectively and actively with its citizens. Sensors are the building blocks of smart cities because they gather data and send these data to a central server that refines and makes the data available to the general public. As cities move toward becoming smarter, demand for sensors has increased to monitor important parameters such as noise, traffic, light, car parks, waste management, and pollution levels. As a result, sensor management is playing an increasingly crucial role in data gathering (Lützenberger et al. 2015).

We developed a prototype application that enables system administrators to monitor a network of sensors when moving around the city. The application helps to locate system sensors, visualize their current status and data gathered about them, and to flag any sensors that have an abnormal status. However, the application is designed for experts and not for end users. In this context, the POIs are the sensors installed in the city, and the colour of each POI indicates the status of the sensor. More information can be obtained using a Bluetooth connection when the administrator is close enough to the sensor. For example, as shown in Fig. 7, the application can indicate the location and the status of a sensor monitoring traffic and another sensor monitoring light.

5.4 Augmented travel

The tourism industry offers many potential uses for AR, including offering more information to visitors, but AR is still underutilized in this industry. Tourists tend to be in unfamiliar environments such as a new city in search of hotels, restaurants, museums, and shops and have no idea how to get there or discover more.

To address this, we developed a prototype application that exploits the features of our framework to visualize POIs using a hierarchical representation. The rationale behind this approach is to avoid visualizing a large amount of unstructured information that could confuse tourists. In addition, this representation matches a typical behaviour of a tourist who first explores the most important location of the city and then searches around that location for a specific service such as a hotel, restaurant, and shop. In this case, the POIs are shown as simple circular billboards. In the top level of the hierarchy, we have the most important POIs of the city, such as town squares or historical zones. In the second level, we have hotels located around the top-level POIs. In the third level, we have restaurants and shops located around the previous POIs. Other POIs can be considered subsequently, such as museums, churches, and monuments. Figure 8 shows a screenshot of our application. The user can expand each level separately by touching the POIs, and the distance of each POI from the user is indicated.

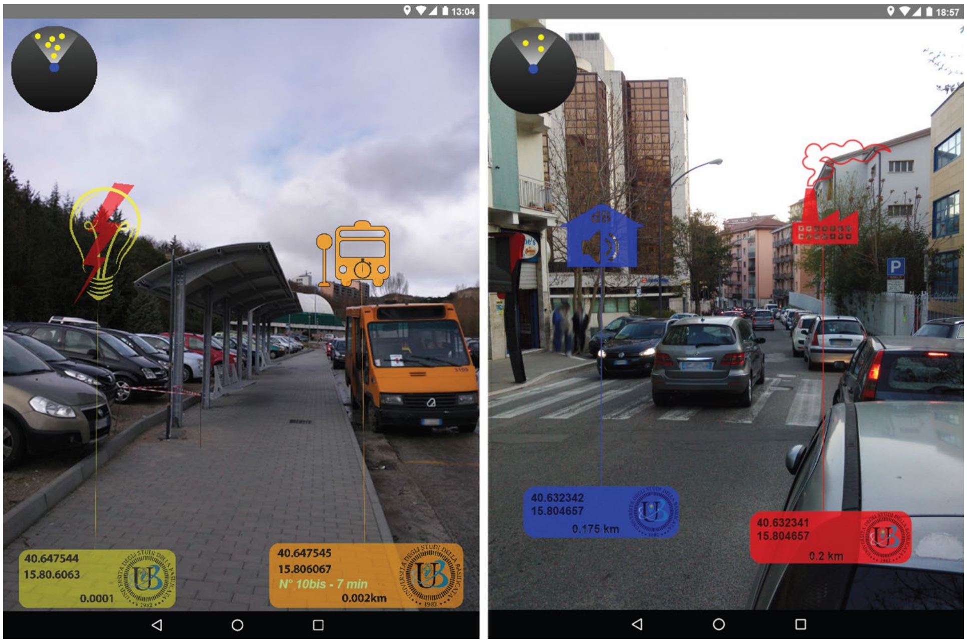


Fig. 7 The augmented reality application for smart city sensor management. Left: two points of interest are a light monitoring sensor and a bus traffic sensor. Right: two points of interest are noise and air pollution sensors

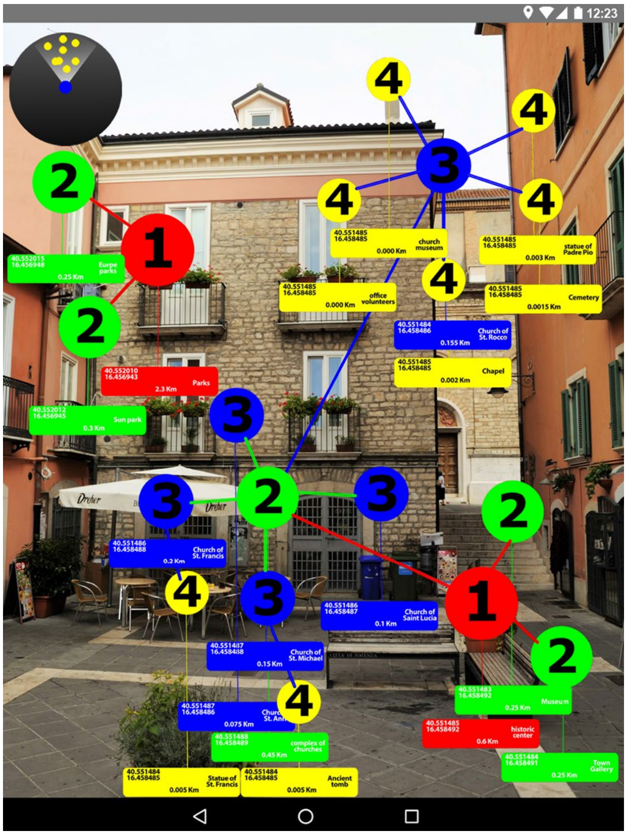


Fig. 8 The augmented reality application for travellers. A hierarchical display is used. Red icons are top-level points of interest, e.g., town squares, green icons are second-level, e.g., hotels, blue icons are third-level, e.g., shops and restaurants, and yellow icons are other significant places, e.g., museums, churches, monuments. (Colour figure online)

6 Conclusion and future work

In this paper, we have presented a scalable software architecture with a clear division between client- and server-side technology for developing geo-located AR applications for smart mobile devices. The server does not provide details of graphics, which can be customized independently according to the needs of the client. The server simply provides the POI information requested by the client, such as the distance from the client device to each POI, and the orientation of POIs. We intend to continue our work on the proposed infrastructure, notably by contextualizing POIs to the type of information the user would like to display. We plan to integrate the geographic information system inside our architecture. We are confident that several types of geographic information can be brought into our architecture relatively easily by using the idea of layers.

One of the major goals of our research is to provide a rapid prototyping framework, within which new applications can be prototyped and tested easily. Our proposed framework is flexible, generalizable, and the infrastructure can provide solutions to problems in many different domains. Our case studies illustrate four applications: management of failures in electrical power lines, support for hydrogeological monitoring, management of sensor networks, and geo-localization of tourist attractions. The broad spectrum of applications already developed with our framework demonstrates that this goal is achievable.

Finally, our framework takes a significant step toward enabling not only programmers but also designers to work with geo-located AR technology. Rapid prototyping and early experience testing is crucial for developing good user experience and for exploiting the wide range of potential applications in the many different sectors already looking into the possibilities of using AR-based solutions to boost their industries.

References

Azuma RT (1997) A survey of augmented reality. Presence Teleoper Virtual Environ 6(4):355–385

Bane R, Hollerer T (2004) Interactive tools for virtual X-ray vision in mobile augmented reality. In: Proceedings of the 3rd IEEE/ACM international symposium on mixed and augmented reality. Washington, DC, USA: IEEE Computer Society, pp 231–239

Capece N, Agatiello R, Erra U (2016) A client–server framework for the design of geo-location based augmented reality applications. In: 2016 20th international conference information visualisation (IV), pp 130–135

Carotenuto F, Erra U, Scarano V (2014) Slam map application for tracking lights on car dashboards. In: De Paolis LT, Mongelli A (eds) Augmented and virtual reality: first international conference, AVR 2014, Lecce, Italy, September 17–20, 2014, revised selected papers. Springer, Cham, pp 413–420

Choi J, Jang B, Kim GJ (2010) Organizing and presenting geospatial tags in location-based augmented reality. Pers Ubiquit Comput 15(6):641–647

Das P, Zhu M, McLaughlin L, Bilgrami Z, Milanaik RL (2017) Augmented reality video games: new possibilities and implications for children and adolescents. Multimodal Technologies and Interaction 1.2. http://creativecommons.org/licenses/by/4.0/

Delić A, Domančić M, Vujević P, Drljević N, Botički I (2014, Sept) Augeo: a geolocation-based augmented reality application for vocational geodesy education. In: Elmar (elmar), 2014 56th international symposium, pp 1–4

Delisle M (2009) Mastering phpMyAdmin 3.1 for effective MySQL Management. Packt Publishing, Birmingham

Erra U, Colonnese V (2015) Experiences in the development of an augmented reality dressing room. In: De Paolis LT, Mongelli A (eds) Augmented and virtual reality: second international conference, AVR 2015, Lecce, Italy, August 31–September 3, 2015, proceedings. Springer, Cham, pp 467–474

Fielding RT, Taylor RN (2002) Principled design of the modern web architecture. ACM Trans Internet Technol 2(2):115–150

Furht B (2011) Handbook of augmented reality. Springer, Berlin

Geiger P, Schickler M, Pryss R, Schobel J, Reichert M (2014) Location-based mobile augmented reality applications: challenges, examples, lessons learned. In: 10th int'l conference on web information systems and technologies (Webist 2014), special session on business apps, pp 383–394

Geroimenko V (2012) Augmented reality technology and art: the analysis and visualization of evolving conceptual models. In: 16th international conference on information visualisation, IV 2012, Montpellier, France, July 11–13, 2012. IEEE Computer Society, pp 445–453

Global Industry Analysts (2015) Mobile augmented reality (MAR) market trends. Global Industry Analysts. http://www.strategyr.com/MarketResearch/Mobile_Augmented_Reality_MAR_Market_Trends.asp

Hincapie M, Caponio A, Rios H, Mendivil E (2011) An introduction to augmented reality with applications in aeronautical maintenance. In: Transparent optical networks (ICTON), 2011 13th international conference on, pp 1–4

Li H (2011, Sept) Restful web service frameworks in Java. In: Signal processing, communications and computing (ICSPCC), 2011 IEEE international conference on, p 1–4

Louvel J, Templier T, Boileau T (2012) Restlet in action: developing restful web apis in Java. Manning Publications Co, Greenwich

Lupp M (2011) Open geospatial consortium. In: Shekhar S, Xiong H (eds) Encyclopedia of GIS. Springer, Berlin, p 815

Lützenberger M, Masuch N, Küster T, Freund D, Voß M, Hrabia CE, Albayrak S (2015) A common approach to intelligent energy and mobility services in a smart city environment. J Ambient Intell Hum Comput 6(3):337–350

Madden L (2011) Professional augmented reality browsers for smartphones: programming for junaio, layar and wikitude, 1st edn. Wiley, Oxford

Metaio DOC doc description [Computer software manual] (2015). http://dev.metaio.com/sdk/documentation/. Accessed 2 Feb 2015

Schall G, Mendez E, Kruijff E, Veas E, Junghanns S, Reitinger B, Schmalstieg D (2009) Handheld augmented reality for underground infrastructure visualization. Pers Ubiquit Comput 13(4):281–291

Schall G, Schmalstieg D, Junghanns S (2002) The use of augmented reality, GPS and INS for subsurface data visualization. In: XXII international congress of the FIT. Washington, DC, USA, pp 1–12

Shreiner D, Group TKOAW (2009) OpenGL programming guide: the official guide to learning OpenGL, versions 3.0 and 3.1, 7th ed. Addison-Wesley Professional

Sinnott RW (1984) Virtues of the haversine. Sky Telesc 68(2):158

Wang M, Callaghan V, Bernhardt J, White K, Peña-Rios A (2017) Augmented reality in education and training: pedagogical approaches and illustrative case studies. J Ambient Intell Hum Comput pp 1–12

Widenius M, Axmark D (2002) Mysql reference manual, 1st edn. O'Reilly & Associates Inc., Sebastopol

Wilde E, Pautasso C (2011) Rest: from research to practice, 1st edn. Springer, Berlin

Wingfield N, Isaac M (2016) Pokémon go brings augmented reality to a mass audience. The New York Times. https://www.nytimes.com/2016/07/12/technology/pokemon-go-brings-augmented-reality-to-a-mass-audience.html?_r=0. Accessed 2 Feb 2015