



Augmented Reality Based Mobile App for a University Campus

Eman Yaser Daraghmi¹

¹Assistant Professor, Applied Computing Department, e.daraghmi@ptuk.edu.ps

Abstract

In this paper, CAMA, a context-aware mobile application for a university campus is presented. CAMA app provides two main services to the user equipped with a smartphone the context-aware service, and the navigation service. The context-aware service provides illustrative information to users. Students need to know their classrooms, supervisors and public locations, while the university staff may need to know additional information. By connecting our system with the user profile, this service can provide personal information. It can notify both students and teachers of their next classroom, and allow them to download the course material on fly. Moreover, the mobile application includes additional features, such as providing the shortest path via the navigation service, detecting more than one object simultaneously, location sharing, personal context-aware, recommendation, and voice-commanded search. The combination of features that are included in CAMA app, such as providing personalized context-aware services and recommendation, distinguishes it from other applications. All the features of CAMA app are possible with the availability of state-of-the-art AR technologies. CAMA app was tested in a University Campus as one of the technologies to be used later in an intelligent Campus environment.

Keywords: Augmented Reality, context-aware, navigation, recommendation.

1. Introduction

Nowadays, the magnificent computing capabilities in handheld devices, such as smart phones and tablets as well as the new technologies, such as Augmented Reality (AR) and 3D visualization are powering the mobile application sector to the extent of becoming the fastest growing one in information communication technologies. The capability of handheld devices in delivering the user location allows for context-aware information services adopted to the user.

Previous research shows that the new technologies, such as Augmented Reality (AR) and 3D visualization play a significant role in improving the context-aware services. AR is a novel technology that aims at displaying virtual 3D objects that merge seamlessly with the actual video scene captured by the device camera (Delail et al. 2012). As AR is founded on detection of a known marker, as well as its tilt in 3D coordinates (Paredes Figueras & Simonetti Ibanez 2013), when the marker is in a permanent static location, the localization is accomplished.

Following the previous concept, this research introduces an augmented reality based Context-Aware Mobile Application, namely CAMA. Our proposed system provides indoor and outdoor location detection based on a combination of image marker recognition and inertial measurements. This localization service is core to deliver a context-aware information system for a university campus provided via augmented reality. Augmented Reality (AR) is a live, direct or indirect, view of a physical, real-world environment whose elements are augmented (or supplemented) by computer-generated sensory input such as sound, video, graphics or GPS data.

Our proposed system meets the highest standards and provides the most rewarding environments for users (students or university staff) seeking illustrative information inside the university campus.



First, CAMA app provides the context-aware service to users. Students need to know their classrooms, supervisors and public locations available such as the cafeterias, while the staff may need to know additional information. Also, our proposed app play a vital role in handling interrupting students. In a university, students outside the classroom interrupt professors while giving a lesson inside a classroom in order to ask about friends, other professors, other lectures, or even insuring if they enroll in this class or not. CAMA provides all of this information to users outside the class without interrupting the professor. In addition, it provides navigation to other places or people inside the campus, and provides personal recommendation to users. All the services provided by the CAMA app built around an augmented reality software layer. The AR layer informs the user of a close point of interest, overlaying self-explanatory 3D virtual objects related to the location on the real-time video capture. The points of interest nearby are also shown in a 360 degree fashion supported on the device compass readings. Any AR virtual object is “clickable” or “touchable”, such that the user obtains information related thereupon. CAMA system was initially developed for the University campus where the authors work. Nevertheless, the idea can be extended to any scenario without major architectural changes. Section II presents the others related work. Section III presents architecture of the system. Section IV describes with sufficient detail the implementation aspects of each provided service. Section V addresses the evaluation of the system. Finally, Section VI brings the future prospects and conclusions.

2. Related Work

2.1. Augmented Reality

AR refers to a live and real-world image that has been enhanced or diminished by virtual content through a camera interface. AR technology aims to simplify everyday tasks by complementing the user’s perception of and interaction with the real world (Chen et al. 2011). AR is the ability to superimpose digital media on the real world through the screen of a device such as a personal computer or a smart phone, to create and show users a world full of information which has not been possible to conceptualize until now (Wagner & Barakonyi 2006). The most vital part of Augmented Reality is to acquire data from the database. The data about the objects which are used for displaying information after recognizing objects, as well as images captured in different angles are stored in the database. Thus, these information can be displayed virtually through Augmented Reality in an appropriate environment (Zhou et al. 2012; Morhing et al. 2004).

Augmented reality has many applications (Zarzuela et al. 2013; Escobedo et al. 2014; Sannikov et al. 2015; Daraghmi & Yosef 2016; Wu et al. 2005) . First used for military, industrial, and medical applications, it has also been applied to commercial and entertainment areas. For example, in education, augmented reality applications can complement a standard curriculum. Text, graphics, video and audio can be superimposed into a student’s real time environment. Textbooks, flashcards and other educational reading material can contain embedded “markers” that, when scanned by an AR device, produce supplementary information to the student rendered in a multimedia format. The first versions of augmented reality are already installed on some smartphones. Smartphone owners can download an application called Layar that uses the phone’s camera and GPS capabilities to gather information about the surrounding area. Layar app allows you to scan the public streets for vital information, entertainment news, or networking possibilities. The program essentially ‘layers’ the information about the landmarks and businesses on a particular street. Another example is Google Sky Map4. This Android app will appeal to stargazers and astronomers of all varieties. Simply point your phone at the sky and identify a legion of constellations, stars and planets. If you move the camera around, the information will change in accordance with the coordinates (Stanimirovic & Kurz 2014; Billinghamurst 1999; Bell et al. 2002; F. et al. 2002).

2.2 Vuforia Augmented Reality SDK

Vuforia is an AR SDK for handheld devices, such as smartphones and tablets. Vuforia places 2d or 3d object to a real-time video obtained from these devices.

Vuforia depends on the capabilities of the computer vision technology to recognize, detect, and track objects captured by the video camera in real time. As mentioned in Vuforia manual, all the objects captured from the video camera will not be detected, and only a few objects may be tracked and detected because of the handheld devices’ CPU and GPU.

The capability of Vuforia to register images enables developers to position and orient a virtual object, such as 2D, 3D objects or other type of media in the space, related to real world images or video when these are viewed through the camera of the handheld devices. The virtual object then can track the position and orientation of the real image in real time; thus, the viewer's perspective on the object corresponds with their perspective on the real world target (Paredes Figueras & Simonetti Ibanez 2013). In this way, the virtual object appears as if it from another real world object. The Vuforia SDK supports different types of targets, both 2D and 3D, including multi-target configurations, marker less image targets and frame markers.

The SDK have another additional features, such as localized occlusion detections using virtual buttons, image target selection in real time and the ability to reconfigure and create target sets depending on the scenario (Paredes Figueras & Simonetti Ibanez 2013).

Vuforia provides an API6 in Java, C++ and .Net languages through an extension to the Unity3D game engine. Unity is an integrated authoring tool for creating 3D video games or other interactive content such as architectural visualizations or real-time 3D animations. Unity consists of both an editor for developing/designing content and a game engine for executing the final product (Sellers & Jacobs n.d.).

Vuforia SDK supports both native development for iOS and Android, while also enabling the development of AR applications in Unity that are easily portable to both platforms. AR applications developed using Vuforia are therefore compatible with a broad range of mobile devices including the iPhone (4, 4S and 5), iPad, and Android phones and tablets running Android OSversion 2.2 or greater and an ARM8v6 or 7 processor with FPU9 processing capabilities.

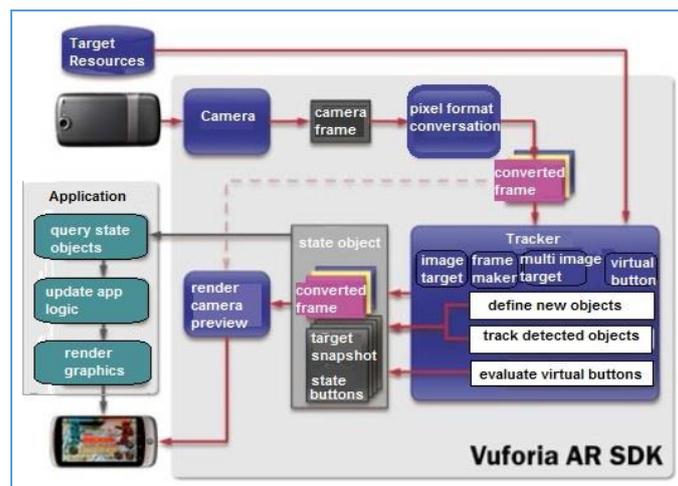


Figure 1: Vuforia SDK Detailed Architecture (Paredes Figueras & Simonetti Ibanez 2013)

The main component of the client side of any AR based application is the AR SDK. Fig. 1 shows the detailed architecture of the Vuforia SDK. In Vuforia, the camera singleton ensures that every preview frame is captured and passed efficiently to the tracker. Developers only need to tell the camera singleton when capture should start and when it should stop. The camera frame is automatically delivered in a device depending on the image format and size. The pixel format converter singleton converts the camera format to a format suitable for OpenGL ES10 rendering and for tracking (e.g. luminance). This conversion also includes down-sampling to have the camera image in different resolutions available in the converted frame stack. The tracker singleton contains the computer vision algorithms that detect and track real world objects in camera video frames. Based on the camera image, different algorithms is used for detecting new targets or markers, and evaluating virtual buttons. The results are stored in a state object that is used by the video background renderer and can be accessed from application code. The tracker can load multiple datasets, but only one can be active at a time. The video background renderer singleton renders the camera image stored in the state object. The performance of the background video rendering is optimized for specific devices. To develop an AR app, developers should initialize all the above components and perform three key steps. First, query the state object for newly detected targets, markers, then, update the application logic with the new input data. And finally, render the augmented graphics overlay. Target resources are created using the on-line Target Management System.

3. System Overview

Fig. 2 shows an overview of the CAMA system architecture. As shown in Fig. 2, our proposed system based on the client-server architecture. It consists of the client side and the server side. The server side in turns consists of the CAMA server and the Vuforia Cloud server. The components of the client and the server sides are shown in Fig. 3 and Fig. 4.

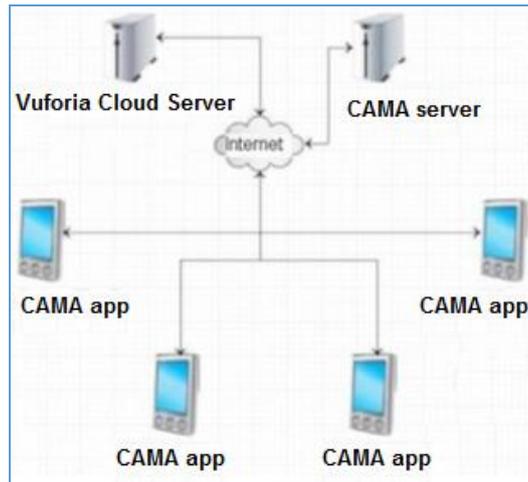


Figure 2: Server-Client System Architecture

CAMA app provides two main important services: the context-aware service and the navigation service. Next section presents those services in details. CAMA app is developed using Qualcomm’s Vuforia SDK with Unity 3D IDE.

The Qualcomm’s Vuforia SDK, which is a back end web application, accomplishes the image recognition and the detection process. Vuforia SDK analyzes the real-time video frames from the smartphone camera to detect well-known markers that should correspond to high-contrast static planar objects which are part of the landscape. Qualcomm’s Vuforia SDK works by delivering the distance and 3D orientation of the image marker (as in (Delail et al. 2012)).

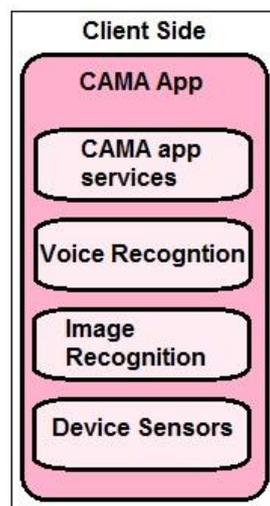


Figure 3: The client side components

The target orientation and posture are further used by the AR layer to overlay 3D virtual objects that merge seamlessly with the scene in the video capture. The markers help the handheld device to place virtual objects on it.

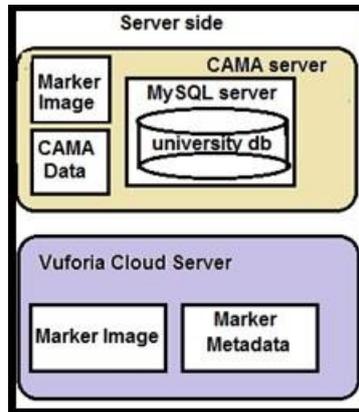


Figure 4: The server side components

To use Vuforia SDK, first, developers have to upload their marker images to the Vuforia Cloud Server, which includes the target manager component. The target manager component converts the uploaded marker image into a feature file using Natural Feature Tracking (NFT). The target manager provides two methods to handle these features points. Fig. 5 shows the feature points of an image.



Figure 5: Image target with visible features

The first method allows developers to upload an image to target manager and retrieve a dataset file that should be included in the developed application as a local marker database. The drawback of this method is when registering a new marker, the local marker database must be updated. The second method allows developers to upload a marker image to target manager and then the Vuforia Cloud Server will store the dataset file and keys will be given to developers in order to retrieve the dataset. There are four given keys: the Server Access Key, the Server Secret Key, the Client Access Key and the Client Secret Key. The Server Access Key and the Server Secret Key will be required to access the Vuforia Cloud Server and to register a new marker, while the Client Access Key and Client Secret Key is required to request a recognition service from the client application. The second method to handle the features points is used to implement our CAMA app.

After the marker images is submitted to Vuforia Target Manager, its features will be identified. These features are basically pointed edges detected from the image of the marker and written into a file. This file will be stored in the cloud server of Vuforia Cloud Recognition Service along with a txt file associated with it as the metadata. In CAMA app, the metadata file is txt file containing a name and an id of the marker image. Therefore, when the application detects an image marker, it collects the features of the image and send them to the Vuforia server to be compared with the features points stored in the database assigned to the application.

When the feature sent by the application matches one of the features in the database, the server will send the associated metadata file to the client application. This metadata file is received by the client application as an identifier to allow users to requests the user location from the CAMA server and thus, providing the app services. To update the user location, the detection of a new marker will obviously trigger the location update. If the user moves away from that location, his/her location is updated with the device sensors that register motion (accelerometer) and direction (compass). The client side also has the voice-commanded recognition module which will be used to provide the navigation service. The speech recognition module is optimized to recognize a set of voice commands issued by the user. The recognition is done using the API of iSpeech SDK [10] that offers the ability to provide the speech recognition with a list of words, allowing the recognition of text that is not available in the dictionary nor the English language, such as Arabic names. It also achieves very good accuracy at recognizing different accents.

The second component of the proposed system is the CAMA Server. CAMA server is a private server that handles every WWW request from the client application. Clients will only contact Vuforia server directly for a recognition request. Any other requests done by the client application are directed to this server. The server is responsible for storing the data. The server also has a connection with the university database. Therefore, when receiving the request from a specific user, the server decides the type of information that should be returned. For example, in a University, students need to know their classrooms, supervisors and public locations, while the staff may need to know additional information. By making the decision and data customization on the server, which may be hosted on the Internet or locally on the Intranet, enables an intelligent behavior of the campus.

The last component is also a very important part of the system, the Vuforia Cloud Server. This server is owned by Vuforia, and available as a service for Vuforia Application Developers. Client application is allowed to request 1000 recognitions per month. This is a cumulative number for the entire client accessing the game. However, the 1000 recognition restriction is only applied to free users of Vuforia Cloud Service. The steps that we followed to implement the CAMA application are shown in Fig.6 and Fig.7. Fig.6 shows the implementation steps of the Vuforia Cloud Target Recognition System, while Fig.7 shows the implementation steps performed in unity environment.

As mentioned before, for the development of the CAMA application Unity3D is used.

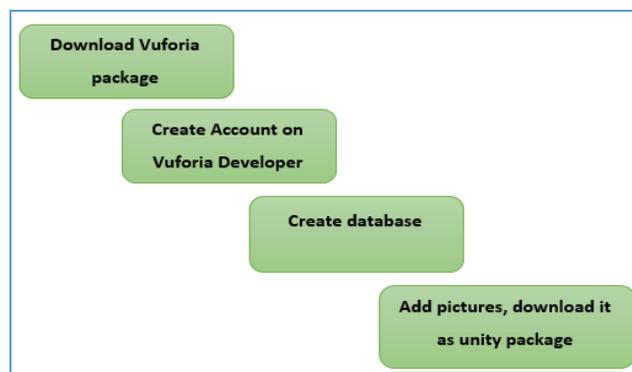


Figure 6: The implementation steps of CAMA app (Vuforia Cloud Target Recognition System side)

Unity is a fully integrated development engine that provides rich out-of-the-box functionality to create games and other interactive 3D content (Fernandez 2011). Several functionalities are provided by unity, such as assemble art and assets into scenes and environments; add lighting, audio, special effects, physics and animation; simultaneously play test and edit games. Unity has the capability to publish to any chosen platforms, such as desktop computers, iOS, Android, etc. In the other hand, the Vuforia extension for Unity enables detection and tracking functionality in the Unity IDE and allows to easily creating AR applications and games.

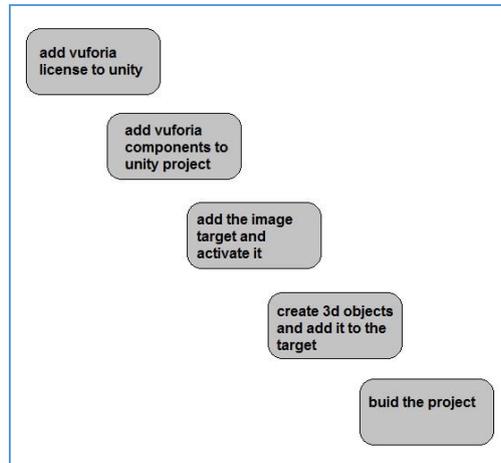


Figure 7: CAMA app implementation steps (unity side)

The scope of this research is shown in Fig. 8. As shown in the figure, the research scope is indicated by a blue box (top box). The features or processes that are located inside the top box are covered in the development of this application, while the rest is managed by Vuforia. Uploading marker image to Vuforia Target manager is done through a request from the mobile application to a private server and then using the Vuforia Web Service; the image is forwarded to Vuforia Target Manager. The system is scalable, and it can be extended to include multiple buildings, new locations and detectable markers without major architectural software modifications. The App was tested in Sony Z2 mobile phone with the official android SDK (Software Development Kit). Following sections shows the services provided by the CAMA system.

4. CAMA app in action

CAMA app provides two main services which are the context-aware service and the navigation service (Fig. 9 shows the login screen of the CAMA app, while Fig. 10 shows the services of the CAMA app).

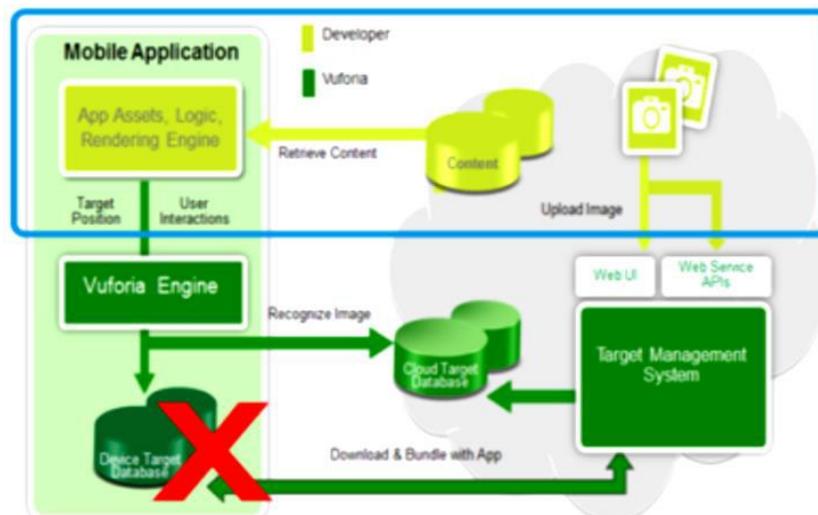


Figure 8: The research scope (Paredes Figueras & Simonetti Ibanez 2013)



Figure 9 : Login Screen for the CAMA app

4.1 Context-aware Service

CAMA App provides indoor and outdoor context-aware services when getting the location of the user. The context-aware service is important for both students and university staffs since students need to know their classrooms, supervisors and public locations available such as the cafeteria, while the staff may need to know additional information.

After the user log in, an interface with two options appear to the user (context-aware service or navigation service). Note that, logging in is vital in our app to link the user personal profile stored in the university database (such as the user timetable) with the detected marker to provide personal context-aware information.



Figure 10: The CAMA app services

When selecting the context-aware service, the app opens the handheld device camera and finds an image marker to start the localization process. Each marker has a permanent location, thus, when a marker is detected,

its unique identifier is used to search the database for the location of that marker. Since the user is close to the marker, this location can be assumed to be the user location. Getting the location of the user can be used further to provide context-aware information via Augmented Reality (AR), which is based on Vuforia SDK. Fig. 11 shows a snapshot of a 3D AR objects (the room number of the detected marker and a question mark virtual button) overlaid on the actual scene captured by the camera.

The information of the detected marker delivered by Vuforia used to render the 3D object in the “right” place. The 3D object data can be downloadable from the server, this makes the system portable and easily reconfigurable, as the 3D data is obtained from the server on the fly. Moreover, “clickable” virtual buttons, provided also by the Vuforia SDK, were implemented as they allow for visual interaction with the detected image. In our proposed app, the question mark “?” presents the virtual button. When a virtual button is pressed, personal context-aware information is displayed. Detected markers can be classrooms, office rooms, labs, faculty buildings, building aisle, or public locations such as a cafeteria.



Figure 11: a snapshot of a 3D AR objects overlaid on the actual scene captured by the camera

Upon detecting a classroom (Fig. 12), the context-aware service shows: the lecture currently given in the classroom or will be given next (if there is no running lecture when detecting the room), the topic of the lecture, the name of the teacher gives the lecture, and the name of the students enrolled in that lecture. Providing all of these information is possible since the CAMA server is connected with the university database. Moreover, if the user who uses the app to detect the classroom is an enrolled students, a recommendation will appear to the student to take a seat inside the classroom and to not miss the class (note that, if the user is the teacher who should give the lecture also a notification will appear to him to go to the class). Also, enrolled students or the responsible professor will be given an options to download the course material of that lecture. A user also can see the list of lectures to be given inside the detected room during the day.

It is clear that by linking the system with the database and providing all the mentioned information, the problem of interrupting the lecture to ask about others will be decreased. The interrupting person can know all of the information via the app only with one click without knocking the classroom door.

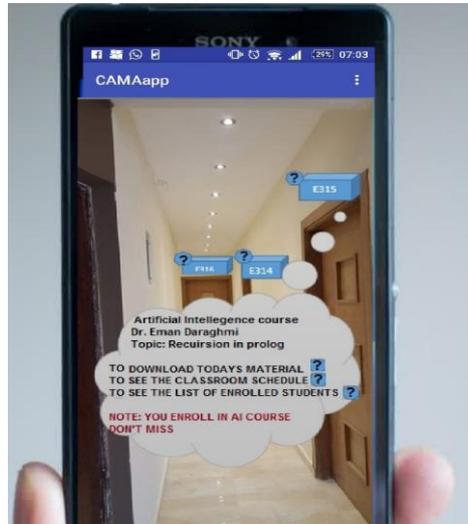


Figure 12: Context-aware information for a classroom

If the detected object is an aisle (Fig. 13), the app is able to detect all the rooms located in that aisle (detecting more than one object simultaneously). The number of each room located in that aisle will be merged with the video capture. All the appeared objects are clickable. If a user click on one detected room, all the information previously mentioned will be displayed (Fig. 12).

If the detected object is an office room (Fig. 14), the service returns the faculty name, his/her office hours, the courses taught by him, his research interest, and his contact information.

If the detected object is an outside building, the service shows the locations of important rooms located inside the detected building (Fig. 15).

Note that, locations of app will be stored in the database upon using any service provided by the app in order to allow others searching for them as will be explained in next section.



Figure 13: An aisle with 3 classrooms



Figure 14: Context-aware service

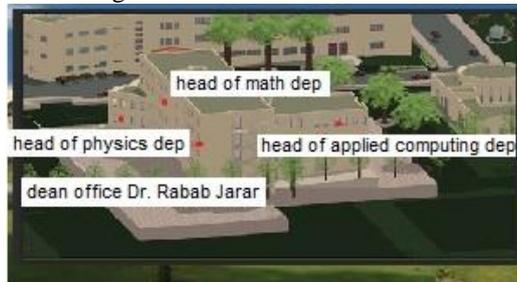


Figure 15: context-aware for outside building

4.2. Navigation Service



Figure 16: Navigation service to room E214

This service allows users (students or university staff) to search for locations (another building, the cafeteria, lab, class room), users (other students or university staff), or even courses (the classroom in which the course is given in or will be given in). This service is provided by either typing the search request in the search field, or requesting the search with the voice recognition engine. For example, the user can type “Dr. Eman Daraghmi office”, or give a command via voice recognition engine “Where is Dr. Eman Daraghmi?” Since the system is connected to the university database, the user also can search “Where is my next lecture?”. The app will retrieve the user timetable from the university database and detect the student next course. Moreover, any user who detected a marker can share his location which will be updated via the smartphone internal sensors and stored in the database. Therefore, searching for people is possible with the navigation service.



Figure 17: Showing the name of the given class when clicking the “?” virtual button

To detect the current user location, an image marker must be detected as explained previously. After detecting the user location, the user can search for other locations. Showing the path from the user location to the destination is the link to providing indoor navigation (Fig. 16, Fig. 17). In order for the system to search for the path, intersections within the building are added to the location database with links that identify which locations can be directly connected. This data can be used to search for the shortest distance path between any two locations and A* algorithm is the best and most efficient algorithm to use for this type of search. The algorithm guarantees finding the shortest path in the fastest possible way without the need of visiting or testing all locations.

5. System Evaluation

5.1. System Comparison

In order to evaluate our system, we compare CAMA app with other AR context-aware apps. Table 1 shows a comparison of features available in CAMA to other mobile applications that provide augmented reality and context-aware services.



The combination of features that are included in CAMA, such as linking the CAMA system with the user personal profile to provide recommendations and personalized context-aware service distinguishes it from other applications.

Table 1: Comparison among CAMA app and other applications

feature	CAMA	(Delail et al. 2012)	(Liapis et al. 2011)
Augmented reality	yes	yes	yes
3D AR Objects	yes	yes	yes
Points of interest	yes	yes	yes
Information sharing	yes	yes	yes
User Profile	yes	yes	no
Friends	yes	yes	no
Detection several objects simultaneously	yes	no	no
Personal recommendation	yes	no	no
Providing personal information	yes	no	no
Outdoor context-aware services	yes	no	no
Sharing location	yes	no	no

5.2. Usability Evaluation

The usability of CAMA app was evaluated by designing a questionnaire that based on the principal factors proposed by Benbunan-Fich (Ma et al. 2012; Benbunan-Fich 2001).

Ten questions were included in the questionnaire based on factors of appeal, content, ease of use, performance, and support. The questionnaire questions are listed in Table 2 with 6 (0–5) agreement degrees ranging from “strongly disagree” to “strongly agree.” This questionnaire was used to survey 210 users (30 faculty members, and 180 students) who were randomly chosen from the university campus. The App was tested in Sony Z2 mobile phone with the official android SDK (Software Development Kit) inside the faculty of Science and Arts at Palestine Technical University-Kadoori. To ensure the active involvement of participants in the questionnaire, an extra prize was provided to them.

Table 2: The usability evaluation

category	question	Average degree of agreement
appeal	I think the user interface is suitable to be displayed in smartphones screens	4.89
appeal	I am willing to use this system frequently	4.76
content	All the functions are well-integrated in the system	4.65
Ease of use	This system is easy to use	4.55
Ease of use	I don't need professional help when using the system	4.23
Ease of use	I can easily find the buttons that I need every time	4.36
performance	The processing speed is fast	4.5
performance	The results of the provided services are accurate	4.29
support	The system can do what I expect it could do	4.28
support	This app makes my life convenience	4.67

As shown in Fig. 18 and table 2, the overall degree of agreement is satisfactory. The appeal category obtained the highest degree of agreement with 4.825, while the ease of use category has the lowest degree of agreement with 4.38.

In addition, Fig. 19 shows the Cronbach α of the questionnaire; the performance category has the smallest α (0.919), and the overall α is 0.912.

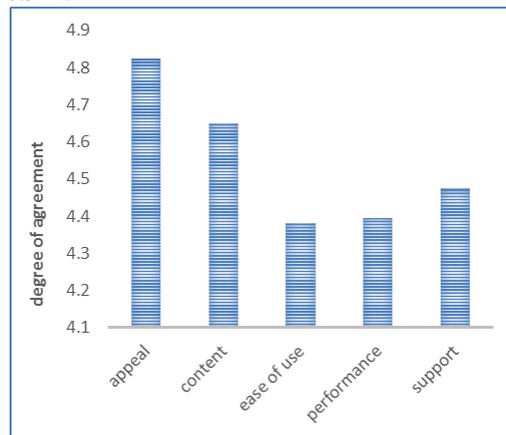


Figure 18: Average degree of Agreement

6. Conclusion

In this paper, we propose a system that can be used efficiently to provide users with personal context-aware information, and navigation service in a University campus. CAMA app meets the highest standards and provides the most rewarding environments for users seeking personalized context-aware information. The system allows users to locate classrooms, teachers and friends. The user profile retrieved from the university database is used to notify both students and teachers of their next classroom. The app provides all the information that the user need to know what inside a closed classroom without interrupting the teacher while giving the lecture. The app also provides the navigation service. This service allows users (students or university staff) to search for locations (another building, the cafeteria, lab, class room), users (other students or university staff), or even courses (the classroom in which the course is given in or will be given in). This service is provided by either typing the search request in the search field, or requesting the search with the voice recognition engine. All the features of CAMA app are possible with the availability of state of the art AR technologies. CAMA app was tested in Palestine Technical University as one of the technologies to be used later in an intelligent university environment. Results show that the overall degree of agreement is satisfactory. In future, we plan to extend our proposed system and improve the user experience by supporting multi locations and university buildings.

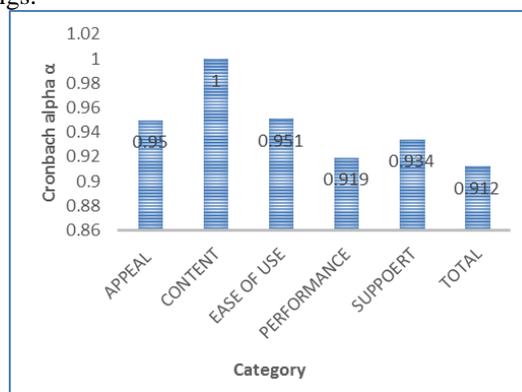


Figure 19: Cronbach alpha



References

- [1]. Bell, B., Tobias, H. & Steven, F., 2002. An Annotated Situation-Awareness Aid for Augmented Reality. In 15th annual ACM symposium on User interface software and technology.
- [2]. Benbunan-Fich, R., 2001. Using protocol analysis to evaluate the usability of a commercial Web site. *Information Management*, 39, pp.151–163.
- [3]. Billinghamurst, H.K. and M., 1999. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *IEEE and ACM International Workshop on Augmented Reality*.
- [4]. Chen, D. et al., 2011. Mobile Augmented Reality For Books on a Shelf. In *IEEE International Conference on Multimedia and Expo*. pp. 1–6.
- [5]. Daraghmi, E.Y. & Yosef, R., 2016. PseudoEye : An Interactive Augmented Reality Assistant App for Electronic Engineering Students. *International Journal of Computer Science and Mobile Applications*, 4(10).
- [6]. Delail, B. Al, Weruaga, L. & Zemerly, M.J., 2012. CAViAR : Context Aware Visual indoor Augmented Reality for a University Campus. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. pp. 286–290.
- [7]. Escobedo, L. et al., 2014. Using Augmented Reality to Help Children with Autism Stay Focused. *IEEE Pervasive Computing*, 13(1), pp.38–46.
- [8]. F., L. et al., 2002. Multimedia Augmented Reality Interface for E-learning (MARIE). *World Transactions on Engineering and Technology Education*, p.173.
- [9]. Fernandez, F.J., 2011. Game Programming Using Unity 3D. In pp. 1–17.
- [10]. Liapis, D., Vassilaras, S. & Yovanof, G.S., 2011. Implementing a Low-Cost , Personalized and Location Based Service for Delivering Advertisements to Mobile Users. , pp.133–137.
- [11]. Ma, X. et al., 2012. Design and Implementation of a Toolkit for Usability Testing of Mobile Apps. *Mobile Networks and Applications*, 18(1), pp.81–97. Available at: <http://link.springer.com/10.1007/s11036-012-0421-z> [Accessed May 27, 2014].
- [12]. Morhing, M., Lessig, C. & Bimber, O., 2004. Video See-Through AR on Consumer Cell Phonestle. In *IEEE/ACM international Symposium on Mixed and Augmented Reality*. pp. 252–253.
- [13]. Paredes Figueras, J. & Simonetti Ibanez, A., 2013. Vuforia v1.5 SDK. Analysis and evaluation of capabilities.
- [14]. Sannikov, S. et al., 2015. Interactive Educational Content Based on Augmented Reality and 3D Visualization. *Procedia Computer Science*, 66, pp.720–729. Available at: <http://www.sciencedirect.com/science/article/pii/S1877050915034316>.
- [15]. Sellers, M. & Jacobs, S., Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#.
- [16]. Stanimirovic, D. & Kurz, D., 2014. [Demo] Smartwatch-Aided Handheld Augmented Reality. In *IEEE International Symposium on Mixed and Augmented Reality*. pp. 373–374.
- [17]. Wagner, D. & Barakonyi, I., 2006. Augmented Reality Kanji Learning. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*. pp. 3–4.
- [18]. Wu, Y., Wu, Z. & Li, J., 2005. Personalized intelligent question answering algorithm in e-learning. In *International Conference on Machine Learning and Cybernetics*. pp. 18–21.
- [19]. Zarzuela, M.M. et al., 2013. Mobile Serious Game Using Augmented Reality for Supporting Childrens Learning About Animals. *Procedia Computer Science*, 25 , pp.375–381. Available at: <http://www.sciencedirect.com/science/article/pii/S1877050913012519>.
- [20]. Zhou, F., Duh, H. & Billinghamurst, M., 2012. Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. pp. 193–202.

A Brief Author Biography

1st Eman Daraghmi works as an assistant professor at the department of applied computing, Palestine Technical University Tulkarm (PTUK). She received her B.S degree in communication and information technology from Al Quds Open University in 2008, her MS degree in Computer Science from National Chiao Tung University, Taiwan in 2011, and her PhD degree in Computer Science and Engineering from National Chiao Tung University, Taiwan in 2015. Her current research interests include Cloud computing, distributed systems, internet technologies and designing.