MATTER: International Journal of Science and Technology
ISSN 2454-5880

Varelas et al., 2020

Volume 6 Issue 3, pp. 43-57

Date of Publication: 11th December 2020

DOI- https://doi.org/10.20319/mijst.2020.63.4357

This paper can be cited as: Varelas, T., Pentefoundas, A., Georgiadis, C. & Kehagias, D. (2020). An AR Indoor Positioning System Based on Anchors. MATTER: International Journal of Science and Technology, 6(3), 43-57.

AN AR INDOOR POSITIONING SYSTEM BASED ON ANCHORS

Thomas Varelas
Information Technologies Institute, Centre of Research & Technology – Hellas Site 51, 57001, Thermi, Thessaloniki, Greece
thomvare@iti.gr

Alexandros Pentefoundas
Information Technologies Institute, Centre of Research & Technology – Hellas Site 51, 57001, Thermi, Thessaloniki, Greece
alexandrospedefoudas@iti.gr

Charalabos Georgiadis
Information Technologies Institute, Centre of Research & Technology – Hellas Site 51, 57001, Thermi, Thessaloniki, Greece
ch.georgiadis@iti.gr

Dionysios Kehagias
Information Technologies Institute, Centre of Research & Technology – Hellas Site 51, 57001, Thermi, Thessaloniki, Greece
diok@iti.gr

Abstract

Indoor navigation is a very interesting scientific domain due to its potential use compared with the GPS signals, which are restricted to outdoor environments. This paper describes commonly used methods of Indoor navigation, positioning, and mapping systems using Augmented Reality (AR) techniques. An Indoor navigation system, which is based on an AR application, is a pipelined procedure, which is consisted of three modules. Those are the positioning system, the map, and the *route planning algorithms. In this paper, the emphasis is placed on the positioning system module and the creation of the map. The most notable options concerning the AR positioning systems use markers or detected planes in the environment in order to accurately define the position of the user in it. In this paper, we propose a new method of positioning which is based on anchors and unlike other methods can provide a total marker-less experience to the user. Anchors are a crucial feature of most AR Frameworks and used to add augmented content on top of a feature point. Also, we propose a mapping technique that fully supports the positioning method mentioned previously. Concepts like AR Frameworks, anchors, and feature points are also, deeply discussed. The proposed method for position tracking does not require any special hardware or component other than a smartphone with a camera. The proposed method for map creation is an enhanced version of an existing method of the ARKit framework. Finally, the paper analyzes the new methods in terms of accuracy in the estimated user position and measures the error in a distance calculation module that was developed to support the positioning method.*

Keywords

AR, Anchor, Feature Points, Indoor, Positioning Systems, Mapping System

1. Introduction

The user navigation in an environment, (indoor or outdoor) is a complex and difficult task to accomplish. It consists of three major elements: the positioning system, the map (mapping system), and the route planning algorithm. Due to the inability of the Global Positioning System (GPS) to locate signals in indoor environments, there is a need for new approaches. Many solutions have been proposed and analyzed in the past, including the utilization of special hardware such as Bluetooth beacons, Wi-Fi transmitters, and RFID transmitters. (Pavel Kriz, 2016) The problem associated with the aforementioned solutions is that even though they can achieve high accuracy of user positioning in an interior environment, most of the time, the cost of installation and maintenance of these systems is relatively high, thus limiting their large-scale deployment. For the above-mentioned reasons, there is a need for a solution that is less costly and easier to install in an indoor environment but with an acceptable amount of accuracy.

Several solutions have been proposed to solve the problem of indoor positioning using AR technologies. All of these solutions have been deployed using a set of libraries and tools provided by an AR framework. A significant number of them are using the ARKit framework for iOS devices, which in its latest versions, provides many unique features and utilities for a developer of

AR applications. On the other hand, there is the ARCore framework, which gives the ability to develop applications on smartphones running Android. In this paper, we made use of the ARFoundation framework, a cross-platform solution that is built on top of the ARCore and ARKit frameworks to support the creation of multiplatform experiences. It can be used in the form of a package contained in the Unity3D package library. Most AR frameworks provide some basic features such as marker recognition, plane detection, feature point detection, and face shape recognition (Google Developers ARCore). Furthermore, they can provide information about the smartphone movement and location in third space with a process called simultaneous localization and mapping. However, the use of AR positioning systems is not a universal solution as those systems most of the time lacks inaccuracy. Methods that try to minimize the accuracy error have been proposed and they mostly suggest the use of markers, which try to improve the initial position estimation given by the computer vision algorithms that the AR application relies on. The use of markers is far less costly than e.g. Bluetooth Beacons but, in several environmental conditions, they may be inefficient due to their dependence on prion installation. The question that arises in such use cases is: Is there a way to have a completely marker-less AR positioning system?

Our paper introduces an alternative approach to this problem, through the use of AR, which requires no special hardware other than a smartphone with a camera. We propose a positioning and a mapping method for indoor navigation applications that are solely based on marker-less features of the surrounding environment. The objective of this paper is to describe sufficiently the new method and test it in a real indoor environment. The structure of the paper is presented below.

- Section 2 reviews related work from the literature in the field of AR positioning systems and sets the fundamentals that led to our proposed solution.
- Section 3 describes our proposed anchor-based method for indoor positioning and analyses the methodology used.
- Section 4 presents some preliminary experimental results from the application of our proposed method in a real environment and discusses them.
- Section 5 concludes the paper and suggests future work.

2. AR Positioning Systems - Literature Review

As mentioned above, in this paper we are studying only the AR positioning systems available and not any system utilizing extra hardware, such as Bluetooth beacons or Wi-Fi-based systems. Regarding the use of AR, the first system we will mention is the marker-based AR positioning system (Chee, 2011). Such a system is based solely on the marker recognition feature of an AR framework to provide the user position in the 3D space. By the term marker here, we are referring to any easily distinguishable elements put in the environment so that they can be identified apart from other objects in the same environment (Erkan Bostanci, 2013). Such elements are e.g. QR codes or real 3D objects that have several salient features (G Stavropoulos, 2010). In these systems there are several markers, uniquely identified, so when they are recognized, they show an arrow representing the route that the user needs to follow until he reaches his destination. Every time the user finds a marker, the route planner algorithm computes the link between the current marker and the destination and suggests where the user needs to go from there.

Another approach is an AR positioning system based on marker-less features such as plane detection or feature point detection. A plane is usually a surface that can be distinguished from any object that is placed on, or under it. Planes can be horizontal or vertical and many systems utilizing planes to determine a user's position on the map tend to use them also for tracking surfaces of different types and categorizing them based on their shape (Erkan Bostanci, 2013). Finally, a worth noting approach is the feature point detection feature. There are several papers (Rastislav Cervenak, 2019) that explored this feature's capabilities including this one. Previous work has been done in grouping feature points and creating sets of them so they direct the users to the desired destination. Finally, there is a study where large static objects such as windows and doors were taken as reference points and the method tried to accurately position the user in the environment using distance calculation between the large objects and the camera. (Aoran Xiao, 2018)

3. Anchor Based Proposed Method - Methodology

In this section, we present the steps of our proposed method as indicated in *Figure 1*.

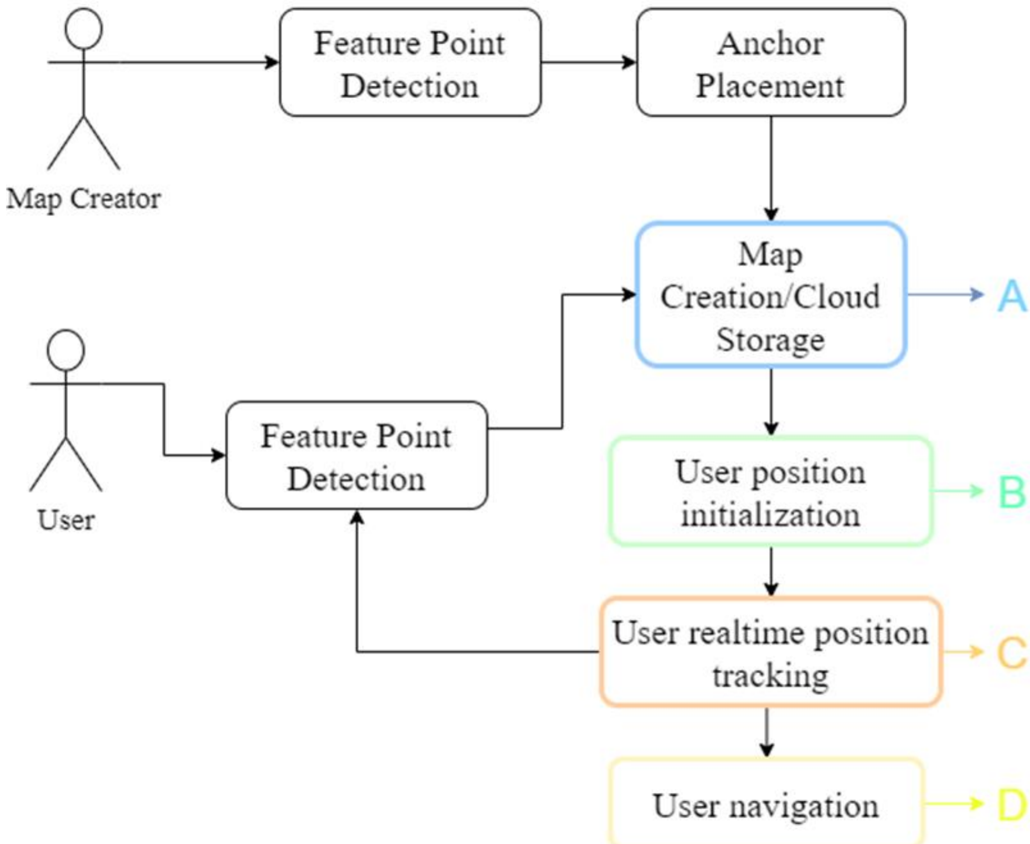

Figure 1: *Proposed Method Schema. The Colored Steps A, B, C, and D refer to the Steps Described Below*

3.1 Creating the Map - Step A

As already mentioned, this paper proposes a new method based on Anchors and specifically Cloud Anchors, which is a relatively new feature of the ARCore framework. We use anchors to determine the exact user position in the environment without the use of any marker. First, to determine the exact position of the user we need a positioning metric based on a map of the indoor environment. In our case, a special type of user that we will call the Map Creator from now on needs to create a map of the indoor place. This procedure can be performed by anyone with a smartphone utilizing a single camera and it is different from the positioning procedure. The Map Creator needs to start walking around the indoor environment capturing as much of the place as possible, with the smartphone's camera. In this case, the more details are captured from the environment, the better and faster will be the initialization of the positioning system, as explained in what follows. As long as the camera keeps capturing the space around the user, more feature points will be detected, as shown in *Figure 2*. Feature points are represented as yellow dots above objects of the real environment. The presence of a feature point means that at that exact point there is a distinguishable detail. E.g. in *Figure 2* feature points are placed in the corners of the corridor. All those feature points together form a point cloud. Point clouds are useful because they can recreate a 3D space amongst other functional properties. While the feature point cloud is created by simply moving the camera around and with the help of real-time feature point recognition provided by the ARCore framework, the Map Creator is responsible to place Anchor objects in front of him while walking. The Anchor objects are placed by a simple tap on a detected feature point. On every tap, the Map Creator places more than one Anchor object. This is done to simplify the Trilateration process of the position initialization, which will be presented below. The additional Anchor objects are not visible to the Map Creator. It is a good practice to place an Anchor approximately every 2 meters in order to simplify the initialization step, which is described below. Alongside the anchors that the Map Creator places in the walking path, there is another type of anchor, which is placed in a feature point of every point of interest (POI). With the term POI here, we are referring to every destination available in the indoor environment. The anchors that are placed in POIs have different shapes and designs as shown in *Figure 3*. When the Map Creator places an anchor, the anchor's corresponding feature point alongside some of its neighbor feature points is saved in a cloud hosting service. For each anchor, we save a unique identifier, its corresponding feature point position, the type of anchor, and the positions of all its neighbors. The amount of neighbor feature points that we need to save is around 5000, according to prior testing. This gives a good anchor recognition speed in the next step. For the purposes of the current paper, we used the Cloud Anchor service provided by the Google Cloud Anchor platform. With this platform, we were able to save the cloud anchors directly without using our cloud infrastructure.

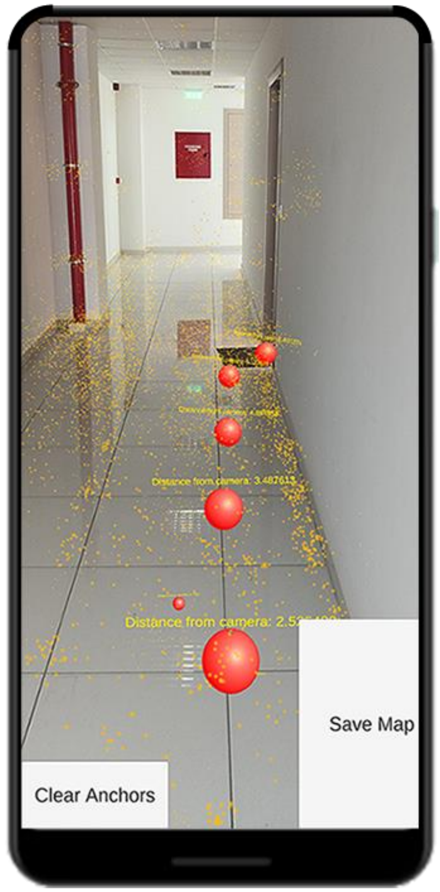

Figure 2: *Anchors (red balls) with Text Above them Showing the Current Distance from the camera. Feature points (Yellow Dots)*

3.2 User Starting Position – Step B

After the creation of the map, a set of anchors and feature points are placed, which form a virtual environment with only the necessary amount of information to recreate the original environment in its proper dimensions. When a user needs to find his way in a mapped indoor environment, he has to open the app and see the image of the AR camera in front of him. At that point, we need to determine what the exact user position in the environment is. To accurately locate a user we need to know if he is currently viewing an anchor. This has been done using the previously referred Cloud Anchor service. This service can locate similarities in the structure and shape between the surrounding point cloud of an anchor, as we saved it, and the detected point cloud from the camera as it is being captured in real-time. If we find two point clouds with a high value of similarity then the corresponding anchor of the point cloud is placed in this area. The rest of the anchors from the saved set is placed based on the pose of the first detected anchor. With the term pose, we refer to the position and rotation of the anchor-based on the position and rotation of the corresponding feature point. At this point, the positioning system knows how to place the previously created map, recreated from the set of anchors and feature points forming a point cloud. This way we end up with a set of anchors that are placed very close to the original positions that the Map Creator saved the anchors.

Figure 3*: Anchor showing the Destination to the User*

3.3 Determine the User Position Real-Time – step C

In this phase, we know there is a map and how we correctly match the map to the user viewing position and angle. Now, we need to determine the exact user position as he moves around the indoor environment and provide the correct navigation. This task is accomplished with the use of the Trilateration method (Fang, 1986). The Trilateration method can find the position of the camera in space given the known distances of the anchors viewed by the camera. In order to use the Trilateration method, we had to create a tool that calculates the exact distance between the camera and an anchor and the distance between two anchors. As described in (Malik, 2012) we can calculate the distance between the camera and the desired point $p(i, j)$ with the following formula.

$$Z = \sqrt{h^2 + Y^2 + X^2} \quad (1)$$

Where h is the height of the camera relative to the ground (known from the sensors of the smartphone). *X* and *Y* are given by the equations (2) and (3), respectively.

$$X = Y \times \tan \varphi \quad (2)$$

$$Y = h \times \tan \theta \quad (3)$$

φ and θ are the vertical and the rotation angles, respectively and they are calculated using the formulas (4) and (5),

$$\varphi = \left(i - \frac{W}{2}\right) \times \left(\frac{FOV_H}{W}\right) \quad (4)$$

$$\theta = \omega + \left(\frac{H}{2} - j\right) \times \left(\frac{FOV_V}{H}\right) \quad (5)$$

where W and H are the image width and height, respectively and FOV is the field of view of the camera. Finally, ω is the angle of the camera relative to the ground. After calculating the distances between the camera and the anchors in the camera field of view, we need two other points with known distances to apply the Trilateration method and accurately find the user's position in the global space. For this purpose, every anchor AR object that is placed consists of three parts. The two parts are invisible to the end-user, but they exist to help the Trilateration method achieve a more accurate result in difficult circumstances, where there are not enough anchors on the camera's field of view. The only visible part of an anchor is the main part, which takes the form of an arrow pointing to the direction of the destination. At this point, we need to mention that for the sake of simplicity of the user interface and to tackle the occlusion issue in AR, we only show the anchors as arrows, when the user is relatively close to them (Manisah Mohd Shah, 2012). The distance calculation tool has been tested and has a centimeter-level of accuracy as shown in *Figure 4*. The tests have been performed on optimal lighting conditions and using a variety of different objects.

3.4 Route Planning – Step D

The final step for a complete indoor navigation system is the Route planning part that is responsible to find the best route from the user exact location to the desired destination. In this paper there was little interference with that subsystem, meaning that we used already known algorithms such as an enhanced version of A* presented in (Dexiang Yao, 2019) and the classic version of the Dijkstra algorithm. The results favored the use of A* because of its slightly more convenient way of path creation.

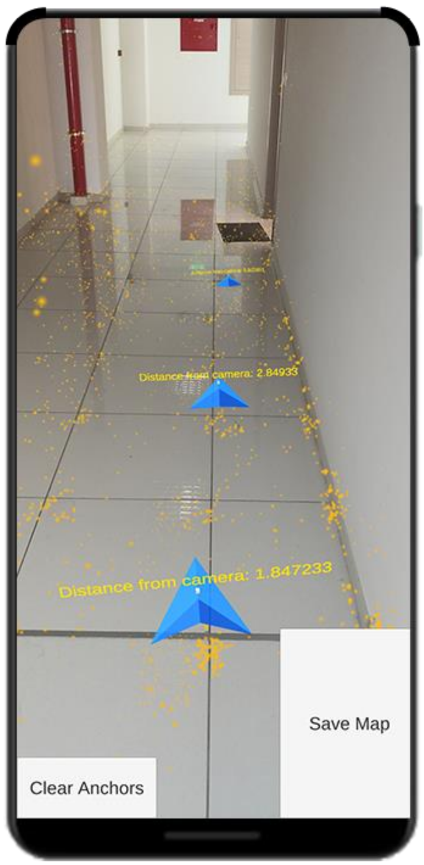

Figure 4: *Arrow Anchors Pointing to the Destination With Text Above them Showing Distance From the Camera*

4. Performed Tests and Benchmarking

For the purpose of this paper, we developed some modules, which helped us to test our method in the context of the position accuracy of the results. One module that we used is the distance calculation module. For this module, we built an executable that detects feature points across the entire camera field of view and with a simple tap places an Anchor on the corresponding feature point. To build this module we used Unity3D and ARFoundation with the ARCore extension for Android. We performed the tests on an Android smartphone with a dual-camera and the accuracy we accomplished was less than 1cm on distances below 2 meters. The accuracy decreases beyond 4 meters in the order of 5-10cm. The anchor cannot be recognized beyond 10 meters so there is no point to calculate the error further than that distance. The accuracy error in distance calculation is shown in Figure 5. This accuracy proves that this method is significantly better than the accuracy in (Aoran Xiao, 2018).

Figure 5: *Plot showing the Error in cm in Relation to Distance in Meters*

Another module that was developed for the purpose of this paper was the Trilateration module. We tested this module using the same set of tools as in the Distance calculation module. The Trilateration module is used to estimate the exact position of the user in the indoor environment. The results strongly depend on the initial Anchor that is detected. If there is enough amount of feature points for the Cloud Anchor service to match the detected point cloud with one of the previously stored point clouds the pose of the first anchor is detected with a higher level of accuracy. In Figure 6 we present the accuracy in relation to the number of detected feature points.

With this test, we try to answer the question of how much feature points we need in order to accurately find the initial position of the user. To perform this test, we put the camera of the Map Creator in a predefined starting position and scanned the area for feature points. The Map Creator is placed than an anchor in a position with a known distance between that position and the predefined position in which the camera first opened. After that, a user started scanning from the same position that the Map Creator began the feature point detection. Throughout the whole feature point detection process on the user side, we calculated the distance between the anchor and the predefined position in which the camera first opened. The results are presented in Figure 6. It is clear that, if the initial anchor is placed accurately enough, then the Trilateration module's job is easy. In a typical environment of an office building the accuracy that was accomplished was 10-50cm depending solely on the initial Anchor detection accuracy. One more issue, in this case, is the lighting conditions of the indoor environment. There is a need for sufficient lighting in the environment in order to properly detect feature point clouds that are based on real distinguishable details of the environment. E.g. we don't want feature points detected on shadows of objects due to bad lighting conditions.

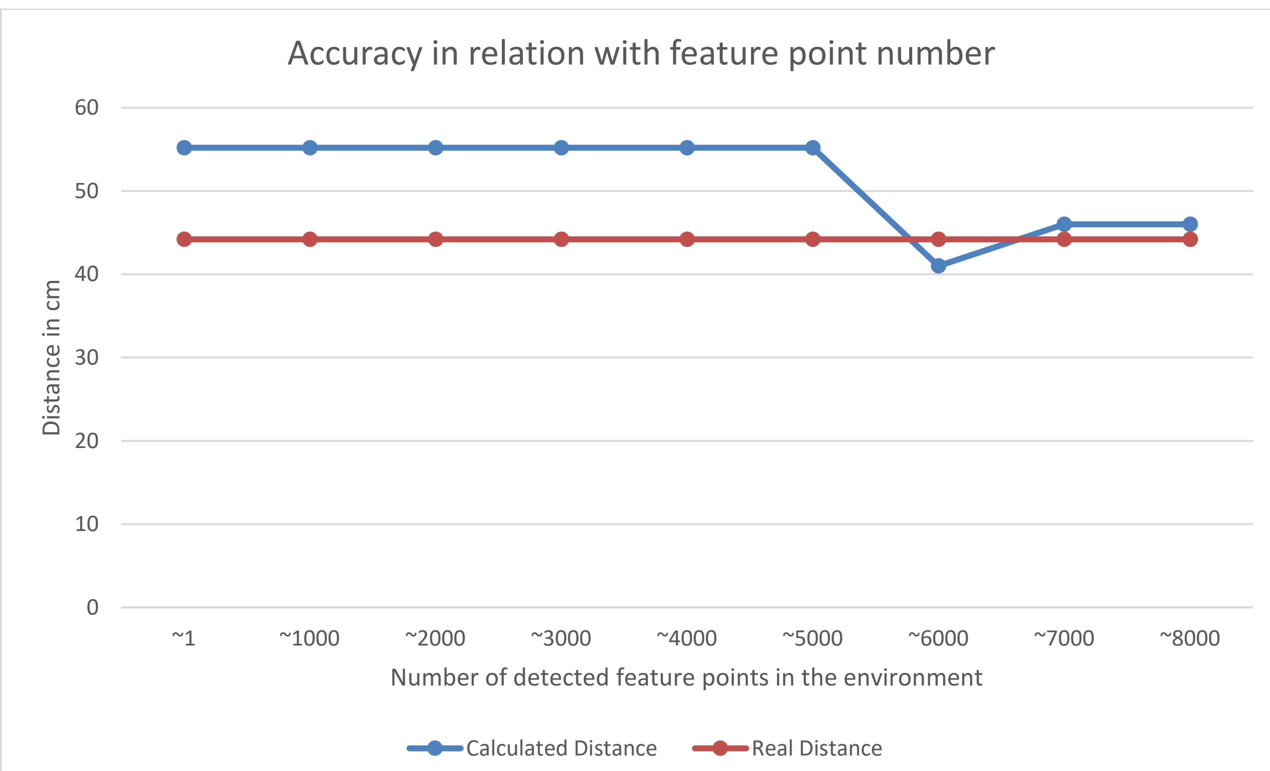

Figure 6: *This Plot Shows the Calculated Distance from a Known Position Based on the Amount of Feature Points Detected in the Scene*

4.1 Discussion on the Results

As shown in the tests we performed above, the idea of using anchors instead of large objects significantly improved the accuracy in the distance calculation module, which is useful in the trilateration procedure. This is a very reasonable outcome because the distance calculation procedure is quicker and more accurate when it concerns an anchor and the camera in relation to the distance calculation between two objects on the same frame as in (Aoran Xiao, 2018). On the other hand and unlike (Aoran Xiao, 2018) an anchor cannot be recognized when its distance from the camera is more than 10 meters because the device cannot recognize feature points correctly in such a distance and place the anchor correctly.

The results of the second test are more interesting because they show that the error in distance measured between an anchor that is placed after the initialization process and the camera, is significantly smaller after the recognition of more than five thousand feature points from the camera. This simply means that the threshold of feature points needed in order to accurately place an anchor is around five thousand feature points.

5. Conclusion

This paper presented a new method for locating the position of the user in an indoor environment. This method comprises several modules that we implemented separately. Their combination results in an accurate integrated indoor positioning system. The different modules have been tested and the accuracy of a few centimeters was achieved. Anyone can create a map following the principles described above and produce a point cloud of the surrounding space in order to easily navigate other users who find themselves in the same indoor environment. The great advantage is that this can be achieved without any extra sensors, GPS, or markers. The only drawback of our method is the slow initialization of the user position, however, this is traded off by the good accuracy it finally achieves. Specifically, the first time that the user tries to find out his location, he might need to move the smartphone around a lot, so the camera captures as much of the space as possible in order to detect the five thousand feature points as mentioned above. Another issue that may limit the accuracy of the positioning system, is its poor performance in dark places, where it is difficult for the camera to distinguish feature points. We expect that this will be overcome by the introduction of more advanced low-light cameras that have started appearing in the latest smartphones.

5.1 Future Research

Finally, future research should focus on the amount of feature points that surround an anchor and on improving the runtime performance of algorithms for comparing their shape and mean distance. If an algorithm such as ICP is implemented and modified appropriately it might lead to quicker feature point cloud shape comparison and might make unnecessary the use of Google Cloud Anchor service that is now needed for that cause.

Acknowledgements

This research has been co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship, and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code: T1EDK-04609)

REFERENCES

Aoran Xiao, R. C. (2018). An Indoor Positioning System Based on Static Objects in Large Indoor Scenes by Using Smartphone Cameras. sensors. https://doi.org/10.1109/UPINLBS.2018.8559754

Chee, L. &. (2011). Augmented reality-based indoor positioning navigation tool.

Dexiang Yao, D.-W. P.-O. (2019). Development of Augmented Reality Indoor. KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL. 13, NO. 9,.

Erkan Bostanci, N. K. (2013). User Tracking Methods for Augmented Reality. International Journal of Computer Theory and Engineering, Vol. 5, No. 1. https://doi.org/10.7763/IJCTE.2013.V5.654

Fang, B. T. (1986). Trilateration and extension to global positioning system navigation. https://doi.org/10.2514/3.20169

G Stavropoulos, P. M. (2010). 3-D model search and retrieval from range. IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 12, NO. 7 https://doi.org/10.1109/TMM.2010.2053023

Google Developers ARCore. (n.d.). Retrieved from https://developers.google.com/ar/discover/concepts#environmental_understanding

Malik, Y. S. (2012). DEPTH AND GEOMETRY FROM A SINGLE 2D IMAGE USING TRIANGULATION. 2012 IEEE International Conference on Multimedia and Expo Workshops.

Manisah Mohd Shah, H. A. (2012). Occlusion in Augmented Reality.

Pavel Kriz, F. M. (2016). Improving Indoor Localization Using BluetoothLow Energy Beacons. Hindawi Publishing Corporation. https://doi.org/10.1155/2016/2083094

Rastislav Cervenak, a. P. (2019). ARKit as indoor positioning system. 11th International Congress on Ultra-Modern Telecommunications and Control Systems and Workshops (ICUMT). https://doi.org/10.1109/ICUMT48472.2019.8970761