Guido: Augmented Reality for Indoor Navigation Using Commodity Hardware

1st Zafer Tan Çankırı
Department of Computer Engineering
Bilkent University
Ankara, Turkey
tan.cankiri@ug.bilkent.edu.tr

2nd Erdem Ege Maraşlı
Department of Computer Engineering
Bilkent University
Ankara, Turkey
ege.marasli@ug.bilkent.edu.tr

3rd Sait Aktürk
Department of Computer Engineering
Bilkent University
Ankara, Turkey
sait.akturk@ug.bilkent.edu.tr

4th Sinan Sonlu
Department of Computer Engineering
Bilkent University
Ankara, Turkey
sinan.sonlu@bilkent.edu.tr

5th Uğur Güdükbay
Department of Computer Engineering
Bilkent University
Ankara, Turkey
gudukbay@cs.bilkent.edu.tr

***Abstract*—Indoor positioning is one of the difficult problems in current navigation systems. There is an increasing demand for detecting the locations of objects and humans inside closed environments in various fields including surveillance, robotics, and entertainment. Recent works focus on indoor navigation systems using different technologies including Wireless Local Area Network (WLAN), Radio Frequency Identification (RFID), Inertial Measurement Unit (IMU), and Simultaneous Localization and Mapping (SLAM). Research reveals using these technologies alone is inefficient in terms of accuracy and cost. To address this issue, we propose a marker-based Augmented Reality (AR) indoor navigation system with integrated SLAM and IMU. We use Unity's AR Foundation Framework for highly accurate results with minimum hardware requirements.**

***Index Terms*—Augmented Reality, Simultaneous Localization and Mapping, Inertial Measurement Unit, Indoor Navigation**

I. Introduction

There is an increasing demand for indoor navigation in many fields, including surveillance, robotics, and entertainment. Positioning in indoor locations is a challenging task as buildings disallow techniques that function well for outdoors. Recently, a large amount of research has been conducted on indoor navigation systems using different technologies such as Wireless Local Area Network (WLAN), Radio Frequency Identification (RFID), Inertial Measurement Unit (IMU), and Simultaneous Localization and Mapping (SLAM) [12].

We propose an indoor positioning system that enables 2D environment mapping and navigation, with minimum hardware requirements. We make use of SLAM, IMU and marker-based Augmented Reality (AR) technologies together to map the indoor location. We use Unity's [21] AR Foundation Framework [18] that utilizes ARKit [6] for IOS support. ARKit enables utilizing IMU and SLAM together, which results in better accuracy. Additionally, we include 3D humanoid agents inside the AR environment to assist navigation.

Indoor navigation requires recording the room borders and the obstacles, based on a reference image that provides a specific pose for each room. We save predefined locations to use in navigation and generate their visibility graphs. When the user requests navigation for a specific location, the room recordings are fetched for the reference image, to convert the new AR space into the recorded one. Then the pathfinding algorithm uses the generated visibility graph of the corresponding location, to calculate a path from the current position of the user to the desired destination. A 3D humanoid agent directs the user by walking along the calculated path.

In this application we aim for high measurement accuracy with minimum hardware requirements. The accuracy of the approaches that use WLAN and RFID is highly correlated with the capabilities of the special hardware in use [10], [16]. Our goal is to eliminate the requirement for special hardware, while achieving high accuracy; thus we also minimize the cost. The resulting system is able to run on low-end AR supported smartphones.

The organization of this paper is as follows: Section II discusses the related work. Sections III, IV, and V discuss indoor environment mapping, indoor positioning, and indoor navigation, respectively. Section VI includes experimental results, and Section VII includes visual results. Finally, Section VIII concludes with an overall analysis of the resulting system, and discusses possible future extensions.

II. Related Work

Neges et al. [12] compare the technologies that are available for indoor scene navigation, and provide the advantages and limitations of each approach. Wireless Local Area Network (WLAN), Radio Frequency Identification (RFID), and Indoor-Global Positioning System (GPS) require specific infrastructure installation.

WLAN and RFID technologies require signal measurement at reference points, and their successes depend on the signal coverage of the device for accurate continuous positioning [17]. Although indoor GPS does not require data

978-1-7281-9134-8/20/$31.00 ©2020 IEEE
DOI 10.1109/IV51561.2020.00123

preparation, its success depends on signal coverage for accurate continuous positioning. Additionally, the accuracy of WLAN, RFID, and Indoor-GPS heavily depend on building-specific disruptive factors, such as the material and thickness of walls [13]. Even with good signal coverage, there are Radio Frequency (RF) issues, such as interference with other radio sources, that influence the positioning accuracy [9]. This is because RF-based positioning systems (except for GPS) use the Received Signal Strength Index (RSSI) as an input to calculate the distance between the user and the reference points. In addition to these, RF-based positioning requires special hardware that increases the cost. WLAN and RFID based solutions do not require signal sources since WLAN access points or Bluetooth Low Energy (BLE) beacons use them as reference points for trilateration techniques.

WLAN based solutions use RSSI measurements of WiFi access points and their fixed locations for positioning. With these measurements, it is possible to calculate the distances between each access point and the user, because of the correlation between the measured signal power and the distance [14]. When distances to at least four different fixed locations are known, it is possible to calculate the user's position in relation to these fixed locations using trilateration techniques. Various products, such as Anyplace [1], use WLAN-based positioning; however, the accuracy of this approach is not at the desired level, and it requires post-processing, such as a Gaussian process, to increase the accuracy of RSSI readings [4]. Techniques such as Kalman Filtering [3] could be combined with this post-processing phase as well.

Using BLE beacons for indoor positioning is a new RFID based approach. Energy consumption of the BLE devices is low, thus it is possible to create a grid network of BLE beacons and use them for positioning with trilateration techniques [5]. Despite the low energy consumption of the BLE devices, the accuracy of the positioning system depends on the number of devices [7], which results in high deployment costs.

For indoor GPS, special hardware such as GPS antennas and repeaters are required, as GPS cannot be used for indoor locations directly. This special hardware should be installed in places to provide indoor positioning for the users [23].

Inertial Measurement Unit (IMU) technology is integrated into most of the consumer devices, including cellular phones; and it provides real-time position data using accelerometer, gyroscope, and magnetometer sensors. However, the generated data is relative to the previous position, and errors accumulate and propagate at a high rate. As a result, relying on this technology alone would create erroneous outcomes. In order to reduce the error rate, filtering (such as Kalman Filtering [22]) is required.

Various studies aim to estimate and analyze the steering directions of people navigating in indoor regions. For example, Azizi et al. [2] propose a floor-plan embedding that extracts and utilizes important low-level space semantics and structural information with encoding space utilization, by detecting people's movement and activities inside the space. This method could be useful for showing the crowded areas by generating heat maps that provide density information. This could be used for commercial purposes such as focusing advertisements in crowded areas. However, this method increases the need for computational power since it processes a lot of graph operations and requires deep learning.

Yang et al. [24] propose a pedestrian trajectory extraction method based on one fish-eye camera that functions inside interior spaces, which would be useful for avoiding collision between humans and virtual agents. Polvi et al. [15] introduce a 3D positioning approach for SLAM based handheld AR systems, which utilizes 3D ray-casting and epipolar geometry for virtual object positioning. The approach does not require perfect 3D reconstruction of the environment and virtual depth cues, thus it better handles the hardware limitations of handheld devices and possible restrictions in the environment.

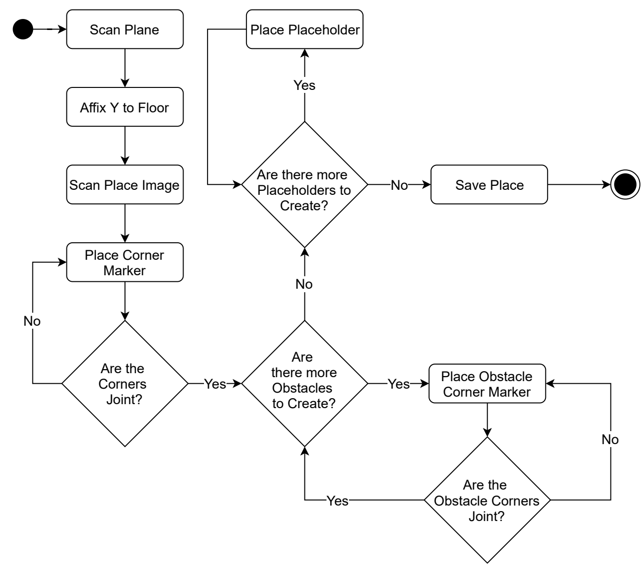


Fig. 1. Indoor environment mapping flow diagram.

III. Indoor Environment Mapping

Indoor environment mapping consists of three main stages: (i) projecting 3D AR Session space to a 2D plane by affixing y coordinates of the objects in the space to y coordinates of the reference image on the ground, (ii) identifying the pose of the reference image, and (iii) specifying the borders of the area, the corners of the obstacles, and the locations of specific objects. Figure 1 depicts the indoor environment mapping process.

A. Projecting the AR Session Space to the Ground Plane

We project 3D AR Session space to the ground plane, where the static reference image is placed since indoor navigation does not require 3D mapping. This also helps to keep the accuracy of the system high by limiting the mapping area. However, this could be problematic with multi-floor buildings, and in such cases, using multiple reference poses on each floor separately would be the solution. We detect the ground plane as AR Foundation's *Plane Within Polygon* type, which is a plane within a polygon by ray casting with *AR Raycast Manager* [19]. Detecting the planes in polygons is a slow process; as a result, once we detect the floor, we use the plane's y coordinate to place all 3D objects including the reference image in the AR Session space on this ground plane.

B. Detecting the Reference Image

A reference pose is required for fetching the saved mappings of a place. We use a static physical image on the ground to serve as a reference pose. The relative pose according to the AR Session of the reference image is gathered from AR Foundation's *Tracked Image Manager*. We discuss the reason for this requirement in Section IV.

C. Identifying Objects on The Ground Plane

We record the poses of objects such as room borders, obstacles, and specific target points (*placeholders*) to be used in indoor positioning and navigation. We use ray casting to identify their relative poses in the AR Session space to save the objects. We use the first hit of *AR Raycast Manager* [19] to detect the pose of the desired object. We repeat this procedure for each object in the place.

IV. Indoor Positioning

In this section, we discuss the details of the indoor positioning system in two parts: *Fetching the Place Data* and *Detecting the User Pose*. The flow diagram of the indoor positioning framework is visible in Figure 2.

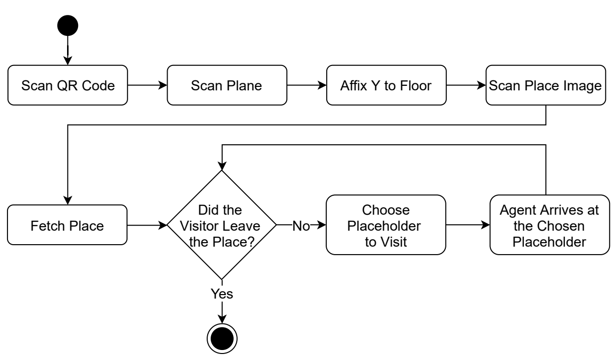


Fig. 2. Flow diagram of the indoor positioning framework.

A. Fetching the Place Data

We use Quick Response (QR) codes for determining the visited places. The data of the currently visited place needs to be fetched to enable user interaction. After fetching the place data, we project the positions of the objects in the place to the ground plane, as discussed in Section III. AR Session spaces formed by AR Foundation use the current pose of the smartphone as the origin when the camera is launched. Because the initial pose at camera launch is not predetermined, a reference pose is required to transform the new space into the fetched one.

We use a predefined image in the physical space to obtain the reference pose so that we can calculate the required transformation operations to convert the new space into the fetched one. We are able to position the fetched objects to the new space since the location of the reference image is fixed inside the physical world. We use *Make Content Appear At* [20] method of AR Foundation to perform these transformations.

Make Content Appear At translates all objects, including the camera, in the new detected space, so that the position of the reference pose becomes the position of reference pose in the stored space. The relative positions of the objects in the new detected space to the reference pose should be preserved; therefore, all transformations applied to the reference pose is also applied to other objects. First, the reference pose is translated to the origin, and then it is translated to the position of the reference pose in the stored space. Additionally, the rotation of the reference pose in the new detected space is transformed into the rotation of the reference pose in the stored space. To achieve this, we first apply the inverse of its rotation and then apply the rotation of the reference pose in the stored space. All these rotation transformations are also applied to the other objects including the camera in the new detected space to keep their relative rotations to the reference pose. After all these transformations, the new detected space becomes the stored space as it can be seen in Figure 3 and Algorithm 1.

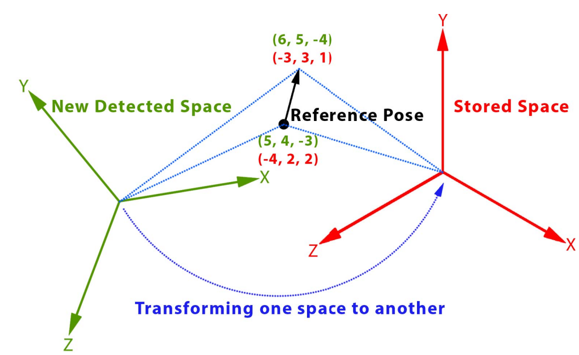


Fig. 3. Space Transformation with a Static Reference Pose

Algorithm 1: Space Transformation with a Static Reference Pose

$norm_{pos} \leftarrow -(Reference \in Space_{New\ Detected})_{pos}$;
$norm_{rot} \leftarrow (Reference \in Space_{New\ Detected})_{rot}^{-1}$;
$disp_{pos} \leftarrow (Reference \in Space_{Stored})_{pos}$;
$disp_{rot} \leftarrow (Reference \in Space_{Stored})_{rot}$;
foreach $object \in Space_{New\ Detected}$ **do**
  $object_{pos} \leftarrow object_{pos} + norm_{pos} + disp_{pos}$;
  $object_{rot} \leftarrow norm_{rot} \times disp_{rot} \times object_{rot}$;
end

B. Detecting the User Pose

The position of the user is required to interact with the AR objects in the place. The position of the camera in the AR Session space changes when the user moves within the place. Since the camera is also in the AR Session space, the transformations applied by *Make Content Appear At* [20] also transform its position. In addition to these transformations, we also project the position of the camera to the ground plane by the affixed y coordinate. As a result, we can get the camera position relative to the objects around it on the ground. To determine the looking direction of the user, we use ray casting. The first ray hit of AR Foundation's *Make Content Appear At* on the ground is used to get the position of a point in front of the camera. The looking direction vector of the user can be calculated from the position of this point and the position of the camera (see Figure 4).

V. Indoor Navigation

The indoor navigation system creates a dynamic path, starting from the user position to the position of the desired placeholders. We utilize *Extremity Pathfinder* Python library [11] to create visibility graphs of the places for indoor navigation. We generate visibility graphs using the saved place border and obstacle polygons. *Extremity Pathfinder* makes use of Lee's Visibility Graph Algorithm [8]. The purpose of this algorithm is to detect the visible corners of a polygon from different corners of other polygons; thus, we obtain the line segments that connect the corners of different polygons without crossing any obstacles. We store the endpoint vertices in the visibility graph. The edges of the visibility graph carry the Euclidean distances of the points. Therefore, we can run a shortest-path algorithm on this graph to find a suitable path for navigation. We use A* search algorithm with Euclidean distance between the current position and the destination as the heuristic. The total time complexity of this procedure is $\mathcal{O}(n \log n)$ [8].

VI. Experimental Results

We conduct two different experiments to analyze the accuracy of our approach. The first experiment is on 1D measurements, and the second one focuses on 2D measurements.

TABLE I
1D Measurement Experiment Results. The first column is the length in the actual physical world and the second column is the length measured by our application. The error rate is the percentage error between the measured and actual lengths.

Line Length (cm)	*Measured Length (cm)*	*Error rate (%)*
20	20.3	1.5
40	40.3	0.75
60	59.7	0.5
80	79.4	0.75
100	99.1	0.9
200	203.2	1.6
500	495.3	0.94

In the first experiment, where we focus on 1D measurement, we compare the actual length of objects to the measured AR length. The aim of this experiment is to find out the 1D measurement accuracy of the approach since the positioning system of our solution is based on such measurements. We list actual lengths and corresponding measured AR lengths of different samples in Table I. We calculate the error rate of each sample using the following equation:

$$\textit{Error Rate} = \frac{|l_1 - l_2|}{l_1} \times 100,$$

Where l_1 is the actual length, and l_2 is the measured AR length of the corresponding sample.

The average measurement error rate is *0.99%*. We did not encounter any correlation between the measured lengths and the corresponding error rates, we believe the error is mainly user-dependent. Based on this experiment, our approach is *99.01%* accurate with 1D measurements.

TABLE II
2D Measurement Experiment Results. The first column is the actual (x, z) coordinate in the physical world. The second column is the (x, z) coordinate in AR Session world. We calculate the error according to the Euclidean distance between the Physical and AR Coordinates.

Physical Coordinate (cm)	*AR Coordinate (cm)*	*Error rate (%)*
(20, 20)	(20.2, 19.9)	0.79
(40, 40)	(40.3, 40.2)	0.64
(60, 60)	(59.5, 61)	1.32
(80, 80)	(79.4, 79.5)	0.69
(100, 100)	(99.2, 100.4)	0.63
(200, 200)	(202, 199.9)	0.71
(500, 500)	(495.4, 501.7)	0.69

In the second experiment, we focus on 2D measurements to detect the error rate for positioning. For this purpose, we scan the same rectangular place with the same camera positioned at different physical coordinates of the place. We compare the physical coordinates to the measured AR coordinates for different samples in Table II.

We calculate the error rates based on the differences between the actual and measured distances, using the following equation:

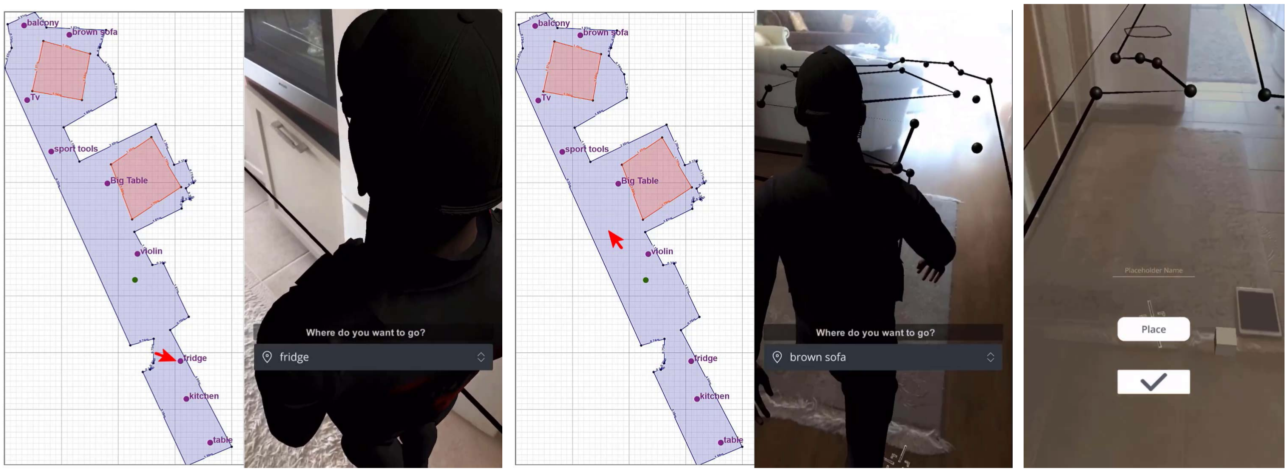
$$\frac{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}{\sqrt{x_1^2 + y_1^2}} \times 100,$$

Where (x_1, y_1) is the 2D coordinates of the physical location, and (x_2, y_2) is the corresponding AR coordinates.

We find no correlation between the physical coordinates and the corresponding error rate. We believe the error is mainly user-dependent, and also caused by the sensor calibration errors. We conclude that the proposed indoor positioning system has an average error rate of *0.78%* and the accuracy of the system is *99.2%*.

VII. Visual Results

In this section, we provide visual results from our application that uses the discussed approach and provide details about our implementation.



(a) Real-time rotation tracking. (b) Real-time position tracking. (c) Object augmentation.

Fig. 4. Visuals from the application: (a) On left, the administrator portal visualizes the user direction using a red arrow towards the fridge. On right, the application screen shows that the camera is directed towards the fridge. (b) On left, the administrator portal visualizes the user's current location within the environment in real-time. On right, the application screen shows a 3D humanoid walking towards the desired destination "brown sofa", starting from the user's current position. (c) Scanning for indoor mapping: synthetic objects for place boundaries and placeholders are augmented into the live feed.

We test our proposed solution by developing a mobile application with the name *Guido*, and an accompanying web-based administration portal. In the application screen, the user sees the environment through the camera, with a 3D humanoid that moves towards the desired destination (see Figure 4). On the administration portal, the position and the direction of the camera (the user) are visualized with a red arrow in real-time, while the user is moving inside the scanned environment (see Figure 4).

Figure 5 shows the measurements of the scanned place, including the measurements of the place edges, obstacles, and points of interest.

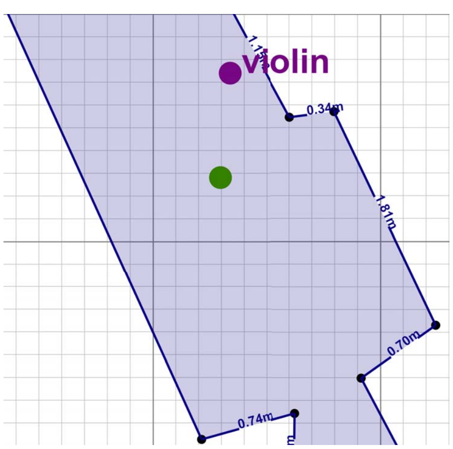


Fig. 5. The measurements of the scanned environment.

In our AR-based indoor mapping system, we augment synthetic objects on top of the live camera feed for the user to see the points of interest, as well as the boundaries of the augmented world. An example of these synthetic objects is shown in Figure 4 (c).

VIII. Conclusion and Future Work

In our proposed solution, we make use of Augmented Reality (AR) Technology, which has the capability of producing 2D plans of indoor environments. Our indoor positioning system utilizes the produced 2D plans. We also provide 3D humanoid agents for realistic navigation scenarios within the generated AR environment. We utilize Unity's AR Foundation framework to create precise 2D plans and localization for indoor places. To create 2D mappings of indoor places, our approach divides the place into sub-elements: the place borders as a polygon, static obstacles as a polygon list, and placeholders as a point list.

For dynamic pathfinding, we create unique visibility graphs of the places using the idea of the polygon with holes. For indoor localization, we fetch the saved 2D plan of places and complete the positioning of AR space using the reference image pose. We construct the dynamic shortest path between the user position and the selected placeholder using the A* algorithm, which utilizes the place visibility graph. We use a 3D humanoid agent to guide the user along the desired path for a better user experience.

With this approach, our goal is to implement an accurate indoor mapping and navigation system with minimum hardware requirements. The precision of early indoor positioning systems, such as WLAN and RFID, heavily depend on the hardware quality and quantity. Our approach facilitates indoor positioning and mapping with minimal hardware requirements, and without compromising precision. Our solution requires an AR supported smartphone on the user's side, and the quality of the camera and the sensors could have a potential influence on the precision and performance of the application.

We believe that augmented reality will become an inseparable part of our daily lives, and indoor mapping and navigation will become essential for collaborative works, multiplayer AR games, and interactive experiences.

This work could be enhanced by utilizing a multi-room based setup using a connected graph for modular rooms. The 3D plan of places could be mapped by utilizing the floor height and additional elements such as windows. To avoid collision with dynamic objects in the environment, moving objects could be detected and their trajectories could be used to improve pathfinding [8].

Acknowledgment

We would like to thank Mr. Kaan Ünlü and Mr. Doğa Oruç, for their help in 3D agent creation and customization processes.

References

[1] AnyPlace. AnyPlace - Indoor Information Service. https://anyplace.cs.ucy.ac.cy, 2020. Accessed on 2020-06-19.

[2] Vahid Azizi, Muhammad Usman, Samarth Patel, Davide Schaumann, Honglu Zhou, Petros Faloutsos24, and Mubbasir Kapadia. Floorplan embedding with latent semantics and human behavior annotations. In *Proceedings of the 11th Annual Symposium on Simulation for Architecture*, SimAUD'20, 2020.

[3] Matthias Bachtler. Kalman filter supported WiFi and PDR based indoor positioning system. Technical report, Bachelor's Thesis, University of Bern, Bern, Switzerland, 2018.

[4] Andrei A. Golovan, Alexey A. Panyov, Vasily V. Kosyanchuk, and Alexey S. Smirnov. Efficient localization using different mean offset models in Gaussian processes. In *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 365–374. IEEE, 2014.

[5] Carles Gomez, Joaquim Oller, and Josep Paradells. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors*, 12(9):11734–11753, 2012.

[6] Apple Inc. Understanding world tracking. https://developer.apple.com/documentation/arkit/world_tracking/understanding_world_tracking, 2020. Accessed on 2020-06-18.

[7] Myungin Ji, Jooyoung Kim, Juil Jeon, and Youngsu Cho. Analysis of positioning accuracy corresponding to the number of BLE beacons in indoor positioning system. In *Proceedings of the 17th International Conference on Advanced Communication Technology*, ICACT'15, pages 92–95. IEEE, 2015.

[8] Der-Tsai Lee. *Proximity and reachability in the plane*. PhD thesis, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign,, 1978.

[9] Filip Lemic, Vlado Handziski, Adam Wolisz, Timotheos Constambeys, Christos Laoudias, Stephan Adler, Simon Schmitt, and Yuan Yang. Experimental evaluation of RF-based indoor localization algorithms under RF interference. In *Proceedings of the International Conference on Location and GNSS*, ICL-GNSS '15, pages 1–8. IEEE, 2015.

[10] Kriangkrai Maneerat and Kamol Kaemarungsi. RoC: Robust and low-complexity wireless indoor positioning systems for multifloor buildings using location fingerprinting techniques. *Mobile Information Systems*, Vol. 2019, Article no.5089626, 22 pages, 2019.

[11] Jannik Michelfeit. extremitypathfinder. https://extremitypathfinder.readthedocs.io. Accessed on 2020-06-19.

[12] Matthias Neges, Christian Koch, Markus König, and Michael Abramovici. Combining visual natural markers and imu for improved ar based indoor navigation. *Advanced Engineering Informatics*, 31:18–31, 2017.

[13] Andrey I. Panychev and Anastasia A. Vaganova. Modeling of ray refraction of WLAN signals on the structural elements of the building. In *Proceedings of the International Siberian Conference on Control and Communications*, SIBCON'16, pages 1–6, 2016.

[14] Alexey Panyov, Andrey A. Golovan, and Alexey S. Smirnov. Indoor positioning using Wi-Fi fingerprinting pedestrian dead reckoning and aided INS. In *Proceedings of the International Symposium on Inertial Sensors and Systems*, ISISS'14, pages 1–2. IEEE, 2014.

[15] Jarkko Polvi, Takafumi Taketomi, Goshiro Yamamoto, Arindam Dey, Christian Sandor, and Hirokazu Kato. Slidar: A 3d positioning method for slam-based handheld augmented reality. *Computers & Graphics*, 55:33–43, 2016.

[16] Abdelrahman M. Sawaby, Hesham M. Noureldin, Mayada S. Mohamed, Maryam O. Omar, Nahed S. Shaaban, Noha N. Ahmed, Samar M. ElHadidy, Reham S. Hussein, Ali H. Hassan, and Hassan Mostafa. A smart indoor navigation system over BLE. In *Proceedings of the 8th International Conference on Modern Circuits and Systems Technologies*, MOCAST'19, pages 1–4. IEEE, 2019.

[17] Silverio C Spinella, Antonio Iera, and Antonella Molinaro. On potentials and limitations of a hybrid WLAN-RFID indoor positioning technique. *International Journal of Navigation and Observation*, Vol. 2010, Article no. 397467, 11 pages, 2010.

[18] Unity Technologies. About AR Foundation. https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@2.2/manual/index.html, 2020. Accessed on 2020-06-18.

[19] Unity Technologies. Class ARRaycastManager. https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@2.1/api/UnityEngine.XR.ARFoundation.ARRaycastManager.html, 2020. Accessed on 2020-06-18.

[20] Unity Technologies. Class ARSessionOrigin. https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@2.2/api/UnityEngine.XR.ARFoundation.ARSessionOrigin.html, 2020. Accessed on 2020-06-19.

[21] Unity Technologies. Unity. https://unity.com, 2020. Accessed on 2020-06-18.

[22] Jian Wang, Andong Hu, Xin Li, and Yan Wang. An improved pdr/magnetometer/floor map integration algorithm for ubiquitous positioning using the adaptive unscented Kalman filter. *ISPRS International Journal of Geo-Information*, 4(4):2638–2659, 2015.

[23] Rui Xu, Wu Chen, Ying Xu, Shengyue Ji, and Jianye Liu. Improved GNSS-based indoor positioning algorithm for mobile devices. *GPS Solutions*, 21(4):1721–1733, 2017.

[24] Lie Yang, Guanghua Hu, Yonghao Song, Guofeng Li, and Longhan Xie. Intelligent video analysis: A pedestrian trajectory extraction method for the whole indoor space without blind areas. *Computer Vision and Image Understanding*, 196, Article no. 102968, 12 pages, 2020.