ASHESI UNIVERSITY

COMBINING VISUAL BASED LOCALIZATION WITH RECEIVED SIGNAL STRENGTH INDICATOR FOR AN AUGMENTED REALITY INDOOR NAVIGATION

CAPSTONE PROJECT

B.Sc. Electrical and Electronics Engineering

Faisal Aduko Wahabu

2021

ASHESI UNIVERSITY COLLEGE

COMBINING VISUAL BASED LOCALIZATION WITH RECEIVED SIGNAL INDICATOR FOR AN AUGMENTED REALITY INDOOR NAVIGATION

CAPSTONE PROJECT

Capstone Project submitted to the Department of Engineering, Ashesi University College in partial fulfilment of the requirements for the award of Bachelor of Science degree in Electrical and Electronics Engineering.

Faisal Aduko Wahabu

April 2021

DECLARATION

I hereby declare that this capstone is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……………………………………………………………………………………………

Candidate's Name:

FAISAL ADUKO WAHABU
……………………………………………………………………………………………

Date: **May 14, 2021** ……………………………………………………………………………………………

I hereby declare that preparation and presentation of this capstone were supervised in accordance with the guidelines on supervision of capstone laid down by Ashesi University College.

Supervisor's Signature:

……………………………………………………………………………………………

Supervisor's Name:

SIBY SAMUEL
……………………………………………………………………………………………

Date: **May 14, 2021** ……………………………………………………………………………………………

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to Dr. Siby Samuel, whose continuous guidance and efforts helped make this project successful. Dr. Siby Samuel, who supervised this project, was the mastermind behind the project topic and provided me with a lot of literature.

I am also particularly grateful to Dr. Nathan Amanquah, Head of the Department of Engineering at Ashesi University, for his constant advice and reminders that kept me focused.

Abstract

Indoor navigation has recently become an important topic due to the rampant development of complex buildings that make navigation within those structures difficult. The Global Navigational Satellite Systems (GSS), like the Global Positioning System (GPS), currently cannot provide navigation services to indoor environments since they require a direct line of sight between the satellites and the receiving devices. Hence, the need for an indoor positioning system that would serve as the GPS equivalent for indoor environments. There are different kinds of indoor positioning technologies, each with its advantages and shortfalls. This project explored the idea of sensor fusion where two positioning methods; the Received Signal Strength Indicator (RSSI) of WiFi and the Visual Based Localization methods are combined to implement an accurate and more reliable indoor positioning system with an augmented reality navigation experience. The Google ARCore SDK with Simultaneous Visual Inertial Odometry algorithms integrated into it was used to implement the visual base localization, while the Root Mean Square Error algorithm was used to implement the RSSI method. Testing the system on different indoor lighting conditions showed an average deviation of 16.73 cm from the actual position. The results indicated that the lighting condition of an indoor space impacts the Localization of a navigating user in the indoor environment.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction and Background

Our ability to locate new territories within and beyond the globe has increasingly become a vital knowledge being sought for over centuries. The knowledge required to navigate the sea is learned from the earlier seafarers who used primitive techniques to direct their vessels in the open sea. Ancient sea navigation techniques involved studying wind directions and observing celestial bodies like the sun's rising on the equinox and the arrangement of identified stars in the outer space, thus the constellation of the night sky. In our modern world, the Global Navigational Satellites Systems (GNSS) makes it possible for navigational devices to track their position in the world, usually in an outdoor environment[1]. Some of the GNSS technologies include the Global Positioning System (GPS) developed by the United States of America, the Globalnaya Navigazionnaya Sputnikovaya Sistema (GLONASS), a Russian equivalent of the GPS, and Europe's Galileo. The accuracy of the current GNSS technologies is dependent on a clear line of sight with the satellites; hence interference from trees, clouds, tall buildings, and others degrade their accuracy.

Modern navigational techniques have significantly evolved, and their applications have been much needed beyond what could have been imagined. Applications such as finding one's way out in a dense city, a university campus, and even in the forest have realized the integration of geographical maps, the Global Positioning System (GPS), and the evolved computing power for accurate Localization. Unfortunately, navigating oneself in an indoor environment like a complex indoor office remains a challenge due to the limitations associated with heavily dependent technologies like GPS.

The Global Positioning System (GPS) is a navigational technology breakthrough developed in 1973 by the US department of defense[2]. It is a radio navigational system that provides real-time geolocation and time data from four or more satellites to any receiver situated at any part of the globe. Devices that can receive GPS signals from satellites are called GPS Receivers. A GPS receiver must have a direct line of sight with at least four satellites for accurate position and time to be estimated. Pseudo-multi-lateration algorithms achieve GPS localization. Unfortunately, GPS does not work well in indoor environments like inside a building, subway, a cave, or any surrounding where there is obstruction of the high frequency, low power radio signals from the satellites. GPS receivers have noise floor ranging around −131 dBW; however, the power levels required to track GPS signals in indoor environments range from −160 dBW to −200 dBW [3]. This leaves navigation in indoor environments practically impossible with the novel Global Positioning System (GPS).

Indoor navigation embodies routing in an environment where GNSS technologies fail, hence the need for an indoor positioning technology. Indoor positioning technologies are based on using the enormous different kinds of environmental sensors that come with computing devices like today's mobile phone to compare a prebuilt map with the environmental signals received[4]. There have been several strides in trying to produce an equivalent GPS technology for indoor environments. Some of the notable technologies employed for indoor positioning include Vision-based Localization (VBL), Received Signal Strength Indicator(RSSI), Inertial Positioning, Radio Frequency Identification (RFID), Visual Light Communication (VLC), Bluetooth beacons, Ultra-wideband (UWB)[5], Magnetic Field Mapping (MFM) and many others.

1.1 Vision-Based Localization (VBL)

Vision-Based Localization (VBL) is a technique that utilizes visual features of the environment fed by a device's camera, applying computer vision algorithms on the visual landmarks to extract useful information, and determining the position of the user within a fixed framed environment[6]. It encompasses keeping in memory the current visual environment and recognizing it when the device's camera reencounters it. Some vision-based localization solutions use marker detection in their architectural algorithms, comprising fiducial and natural marker tracking systems. *Fiducial marker* tracking systems are built to encode and decode information from an artificially designed tracked marker. The *natural tracking system* uses the natural appearance of objects to track them. A more robust direct vision-based localization that considers the different scenes in a dynamic environment is Simultaneous Localization and Mapping (SLAM)[7].

1.2 Inertial Positioning

Inertial navigation uses the Inertial sensors of mobile devices for pose estimation. Inertial sensors consist of gyroscopes and accelerometers, put together as Inertial Measurement Unit (IMU). The IMU sensors are arranged orthogonally to each other[8]. The gyroscopes measure angular velocity with respect to a reference frame, whereas the accelerometer measures the linear acceleration due to a force acting on the IMU. The output of the Inertial measurement is given by;

$$w_t^b = \left(w_t^{b_x}, w_t^{b_y}, w_t^{b_z}\right)^T \qquad (1)$$

$$f_t^b = \left(f_t^{b_x}, f_t^{b_y}, f_t^{b_z}\right)^T \qquad (2)$$

Where w_t is the third-dimensional angular velocity and f_t is the third dimensional applied force on the IMU measured at time t. The IMU orientation is determined from the angular velocity of the IMU with respect to its framed reference. The acceleration due to gravity and the force acting on the IMU are integrated and projected with the IMU orientation. When the acceleration is mathematically integrated, it will result in velocity, and if further integrated, it would produce the IMU position with respect to its reference frame[9] as indicated below.

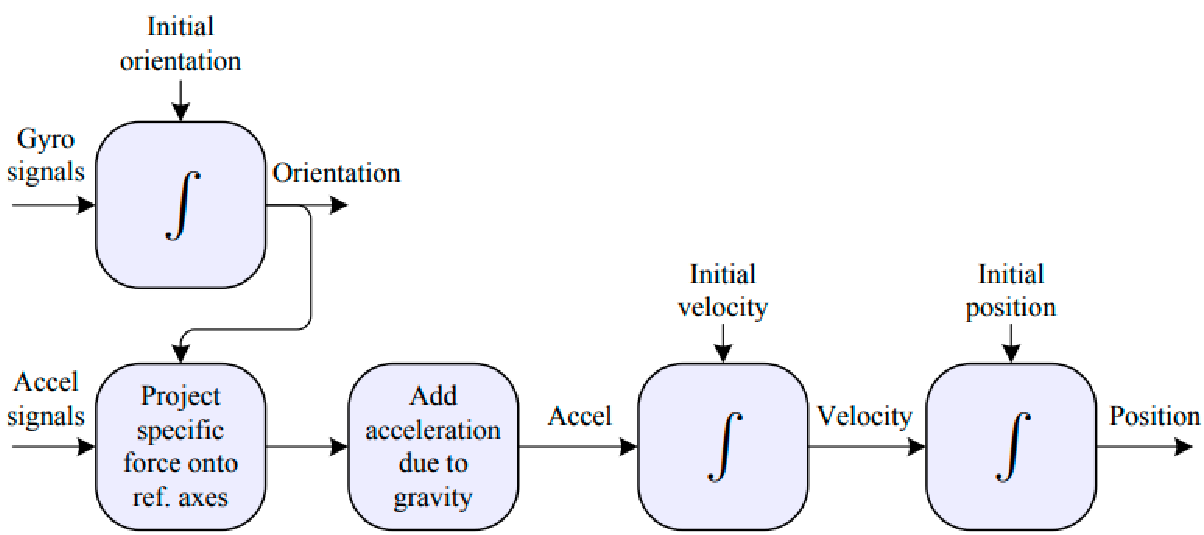


Figure 1.1:Inertial navigation algorithm[21]

1.3 Received Signal Strength Indicator (RSSI)

RSSI utilizes the strength (power) of WiFi signal values received from access points or routers. The strength of the signals varies with distance; thus, the greater the distance from the access points, the lower the received signals' power and vice versa[10]. The power of the received signals from various access points at a time can be used to tell the distance between the receiving device and the transmitter (access points). This will reasonably estimate the position of the device at any time in the indoor environment. The RSSI method has been the most adopted indoor position estimation tool since the revolution to build a GPS equivalent for indoor location mapping. RSSI was the technology used by the first iPhone and iPod touch models when they initially had no inbuilt GPS receivers. RSSI is used internally by all wireless networking devices, including the mobile phone, making the RSSI data acquisition, processing, and pose estimation easier without requiring external hardware.

1.4 Radio Frequency Identification (RFID)

RFID positioning uses wireless radio waves to communicate between a tag and a reader. The radio waves transmit information containing unique codes and data about an object. There are three types of RFID technology including passive, semi passive and active RFID. The tag, also called a transponder, usually attached to a tracked body, contains unique code, and the reader transmits radio waves to read the information on the tag. This method of indoor positioning is expensive to deploy, and the coverage of remote radio waves is limited to small distances(about 3 m) due to interference from external electromagnetic waves and the multipath effect.

1.5 Visual Light Communication (VLC)

VLC relies on pulsing the intensity of light to transmit data to a receiver. They modulate the intensity of light hence depends on a high pulse rate light to prevent flicking. VLC positioning allows devices' cameras to read and understand light fixtures; therefore, light bulbs are programmed to exhibit a unique blinking pattern undetectable by the human eye and correspond to a particular position in an indoor space. VLC can attain submeter accuracy when more light bulbs are used[11]. However, they depend on high pulse rate light, which puts more burden on the receiving device, requiring more processing power.

1.6 Ultra-wideband (UWB)

UWB positioning uses the radio spectrum to transmit information over a wide bandwidth under lower-level energy over a short-range. UWB is beneficial for its ability to detect reflected signals and reject them quickly. It also does not require a direct line of sight with the transmitter making it very ideal for dynamic environments. Unfortunately, it is costly to deploy UWB positioning.

Measurements from each sensor like Bluetooth, Camera, gyroscope, accelerometer, ultrasonic, infrared, and radio sensors come with their constraints, hence the need for sensor fusion. Sensor fusion allows different types of localization sensors to be used together to tackle each sensor's weakness, a process called integration navigation[12]. Incidences, where a single sensor technology has been used to achieve the required results are not known yet, according to Welch and Foxlin[13].

By using the Location Stack model[14], the various indoor positioning methods can be compared in terms of their accuracy and the infrastructure needed to deploy them.

Table 1: Characteristics of different indoor navigation methods

Indoor Positioning Method	**Sensor**	**Measurement**	**Fusion Algorithms**	**Accuracy (cm)**	**Infrastructure Need**
Inertial Positioning	Accelerometer and Gyroscope	Position and Orientation	Mohony, Madgwick & NXP	2 - 10	Medium
Vision-Based Localization	Camera	Image data	Background subtraction & Blob detection	10 -50	Medium
WiFi Positioning	WiFi	RSSI	Deterministic and probabilistic radio map comparison	200 -1000	Medium
Radio Frequency Identification	RFID Sensor	TOA & POA	Least Mean Square, Kalman filter etc.	2 -10	Medium
Ultra-wideband	UWB Sensor	TDOA & AOA	Proprietary	10 - 100	Medium
Ultrasound	Ultrasonic Sensor	Time of flight	Multi-lateration	2 - 10	High
Infrared Positioning	IR Sensor	Containment	none	0.057	Low

1.8 Augmented Reality (AR)

Augmented Reality is a computer program that adds virtual objects generated by the computer into the real-world scene. This is usually achieved by capturing the real environment's image or video with the device's camera and imposing virtual 2D or 3D objects on it. Augmented Reality was first realized in 1968 by Ivan Sutherland, a Harvard professor and computer scientist. He called his device the Sword of Damocles, mounted from the ceiling, which displays computer graphics to make one feel immersed in it when head mounted. Since then, the technology has been further developed until the term "Augmented Reality" was coined by Tom Caudell, a Boeing researcher, in 1990. From 2014, the world had seen an exponential increase in the number of Augmented Reality applications created. Companies like Google, Apple, Wikitude, and many others have developed their Software Development Kit (SDK) to develop Augmented Reality applications.

The foundation of Augmented Reality scores on the following principal operational requirements; World tracking, Scene estimation, and Anchor overlay rendering. Devices such as smartphones that run Augmented Reality use their camera's background to overlay virtual contents. These virtual contents are rendered by the device's Graphical Processor Unit (GPU) and display. Augmented Reality applications use tracking to learn and understand the environment surrounding them. The foundation for AR tracking takes its roots from spatial computing.

Spatial computing refers to an AR application's ability to recognize its location in space, orientation, and movement with respect to its surroundings. The Inertial Measurement Unit (IMU) achieves spatial computing in smartphones. IMU houses gyroscopes and accelerometers, which measure linear acceleration and rotational acceleration, respectively, and a magnetometer for Inertial pos tracking. Reading from IMU is not reliable due to the noise that accompanies the sensor readings and their drift. However, this noise can be filtered using the Kalman filtering technique and employing sensor fusion. Sensor Fusion is a process in which multiple sensors' data predict and track the IMU. To periodically ensure the right predictions, the Simultaneous Localization and Mapping (SLAM) technique is employed to ensure that Augmented Reality tracking is possible.

Computer vision usage in AR solves drift in IMU by referencing tracked feature points as anchors. These anchors serve as the reference localization point in which virtual AR objects are placed. The virtual anchor object becomes part of the environment such that a movement of the AR device will make it behave as if it were a real object. One of the essential elements in scene estimation for AR is lighting. Lighting dramatically affects computer vision computation, pose estimation, and virtual object integration into the physical environment. Hence, multiple viewpoints fed in visual data are encouraged from which better lighting estimation can be reported to the application to ensure user immersion.

1.9 Problem statement

Vision-based localization dependence on camera inputs makes it highly sensitive to lighting conditions. In environments with strong lighting conditions, the Vision-Based Localization method excellently understands the real world and performs accurate pose estimation. However, its usability in environments with poor lighting conditions becomes a nightmare. The pose estimation errors become higher when the device finds it challenging to understand and calibrate the camera feed. This means that indoor navigation applications in situations like in the hospital where there might be poor lighting conditions become obsolete with VBL, hence the need for a more robust and accurate solution.

This project explores sensor fusion to solve the light dependency of the Vision-Based Localization method by integrating the Received Signal Strength Indicator (RSSI) method for a robust indoor navigation application. The project will combine the power of the RSSI and the

Vision-Based Localization methods to create an optimal mobile Augmented Reality indoor navigation solution with improved accuracy in the estimation of pose within all possible usable environmental lightening conditions.

Chapter 2: Literature Review

Enormous efforts have been put into research to develop an accurate indoor navigation system. Most of these researches, dabbed as exploring an indoor GPS equivalent, have explored different options ranging from Bluetooth beacons, WiFi Fingerprinting and trilateration, geomagnetic fingerprinting, computer vision, ultra-wideband (UWB), Radiofrequency identification, infrared, visible light communication, ultrasound, and many more. These technologies come with their advantages and disadvantages, considering cost, environmental obstructions, and processing power required by devices using them.

2.1 Q-Learning Model Using Google ARCore

Jianna et al. [15] implemented an indoor navigation application for a robot's navigation. They claimed to provide a cost-effective and easy framework for autonomous navigation in indoor environments. This system was built on a Q-learning algorithm using Google's ARCore for Augmented Reality applications. ARCore uses Visual Inertial Odometry (VIO), in which the phone's camera is used to understand the environment, perform light estimation, motion tracking, and determining pose. The pose refers to the position and orientation of the phone with respect to its current environment.

On the other hand, Q-learning is a form of reinforcement learning algorithm that is capable of learning by itself its current state and executing commands as required by it. Q-learning algorithm was used to estimate the shortest possible path to a destination when a digitally drawn map in pdf form, describing the indoor environment, is fed to it. The application was implemented using android studio SDK, ARCore SDK Open graphics library (OpenGL) written in Kotlin.

This mode of navigation uses a robot onto which the phone is mounted. The phone does the Localization, and data is sent to control the robot's movement, which intends to direct the user to the desired location. The routing algorithms store a permanent cloud anchor at the starting point, which the robot uses as a reference. From the start, more anchors are placed at various positions along the route to update the algorithm of the direction of the next anchor. The process continues until the destination anchor is reached.

The limitation of this system is that it is always dependent on the lighting conditions of the room. As the room gets dimmer, the errors of Localization worsen. As a result, there is always an increased offset error when routing through long destinations and when it encounters moving persons. This could have been implemented directly on the mobile device with 3D graphics to guide users to navigate instead of adding extra hardware to develop a robot to guide users. Using the robot will most definitely accumulate the robot's mechanical systems' errors to that of the application, further worsening the estimation error.

2.2 Received Signal Strength Indicator (RSSI)

Yoshihiro et al. [16] proposed an indoor navigation system for the visually impaired, in which received signal strength values from radio frequencies transmitted by WiFi access points are used. The proposed system allows a mobile device to communicate with access points within the range in triangulating the distance between the user node and the access points. Firstly, the user node recognizes the access points network and connects to them. The access points then send the received signal values with their SSID name to the user node, which sends this information to an external server for triangulation.

The distance computation is done by using the relationship.

$$d = 10^{\frac{-(RSSI+A)}{10N}} \qquad Where: d = distance \quad (3)$$

$$A = Radio\ wave\ intensity \text{ (- dBm)}$$

$$N = Route\ quality\ factor$$

$$RSSI = RSSI\ value$$

The accuracy of the estimated distance depends on the number of access points available. The higher the density of access points, the higher the accuracy and vice versa. The error of distance estimation is 0.7 m for the maximum RSSI values from three access points.

Using RSSI values to estimate location is not always accurate, as demonstrated in this proposal. This is because of the noise resulting from the non-uniform propagation of the radio signals[17]. In a dynamic environment, the RSSI values of an access point change within the same position changes, introducing more errors.

2.3 Combination of Visual Inertial Odometry, Beacons, and Magnetic Field Map.

In their paper, Rajagopal et al. [18] explored the idea of sensor fusion in which they combined Visual Inertial Odometry (VIO) and beacons to construct a magnetic field map of an indoor environment. They exhibited that with a 6DOF tracking from VIO, a 3D magnetic field map can be obtained and integrated to estimate the orientation of a user in an indoor space. The magnetic field vector map is constructed by the user moving around in the indoor space using the device in build magnetic sensor. Beacons are then used to localize the device with magnetic sensors by comparing it with the already mapped magnetic field. This, they claim, reduces the "vision-based search uncertainty." In one of the indoor environments in which their proposed method was tested, about 80% localization errors of 46 cm and 27 cm respectively was recorded.

These errors were generally a result of the inefficiencies associated with magnetic field indoor mapping. Most Indoor environments have metals that significantly affect the magnetic field maps; hence its application in indoor spaces introduces more challenges.

The above proposed indoor navigation models come with their respective limitations. However, there exists a common relationship between the limitations of the methods. The RSSI method is less accurate in a dynamic environment but can be used in any environmental conditions with radio signals from access points. On the other hand, the Visual Inertial Odometry method has higher accuracy in brighter environments but works poorly in dimmer surroundings. Therefore, this project will capitalize on the power of both methods in developing an accurate indoor navigation application that can be used in all environments. The Google ARCore shall be used to implement the visual Inertial Odometry for the Augmented Reality experience. A pattern recognition algorithm like the K Nearest classification techniques shall also be explored with the Root Mean Square Error localization algorithms for the RSSI method to reduce the position estimation error and reduce the processing power, and memory requirement needed[19].

Chapter 3: System Requirements and Design

This chapter explores the various requirements, goals, and functionality that the proposed design should meet in the end. These requirements comprise of what the technical system's details should be made up with and how it should help achieve the project's objectives. The requirements also comprise what the users of the projects should be able to achieve, the user journey in using the project, the convenience and accuracy that users can achieve with the application to navigate themselves in a dynamic indoor environment.

3.1 System Requirements

Table 3.1: System requirements

Number	**Requirement** *The system should be able to:*	**Justification of Requirement**
1	Be used in environments of varying lighting conditions.	This is to make the system universally useful for all kinds of indoor environments, irrespective of the state of their lighting conditions
2	Track the position of the user in the indoor environment with the highest possible accuracy.	Accuracy in Localization is important in ensuring the right information is available to navigate a user without ambiguities.
3	Display Augmented Reality 3D graphics to aid a user in navigating.	This will allow the user to match routing instructions with the real

		environment for an immersive experience
4	Compatible with all Android six and above mobile devices.	Android, six is the minimum version for the Augmented Reality experience; hence this application should be usable in all compatible android devices.
5	Accurately orient the user in the right direction.	In a vast indoor space, orienting the user ensures that the user positions themselves in the right direction for the routing.
6	Multiuser experience.	A multiuser experience will allow more than one person to use the application and in different indoor environments.
7	Automatically locate the position of a user using RSSI	This is to avoid the user from having to know their current position in an indoor space since the user might be a stranger in the environment
8	Require less processing power and memory by the target devices.	A less power requirement will ensure the device can be used for longer times during routing without running

		down. Also, less memory requirement will increase the device's performance and prevent the application from frequent crashing.
9	Communicate with a remote server in real-time.	A remote server will host user details, feature maps, and a radio map of indoor space. This will ease the burden on the device running the application.
10	Be Inexpensive implementing.	The cost of implementing this project should not be high since this is just a prototype

3.2 Users Requirements

Table 3.1: User requirements

Number	**User Requirements** *The user should be able to:*	**Justification of Requirement**
1	Experience real-time rendering of direction anchors with less lagging.	This is to ensure the user is receiving the right navigating instructions at the right time.
2	Start navigating in an already hosted environment without registration.	This is to reduce the steps involved in using the

		application and also reduce the tendency of users' anxiety in sharing their details
3	Save, edit, and discard hosted routes.	This will allow the admin to have control over the mapping of their indoor space.
4	Choose their desired signal networks(SSIDS) .	This will not only be used for Localization but will also serve as unique fingerprints associated with a specific indoor space. Hence a user's current location will be easily identified without scanning any special marker.
5	Incur less cost deploying the technology.	The cost of deploying the application for an indoor space should be less.

Chapter 4: Project Design

This project implements an indoor navigation system that enables a user to navigate within an indoor environment, irrespective of the lighting conditions in the indoor space. The system's design is an application for use in a mobile phone that will require less processing power, storage, and less battery consumption. With the inbuilt capability of modern smartphones, inbuilt sensors like cameras, gyroscopes, accelerometers, and magnetometers provide the needed architecture for Augmented Reality applications and determine the device's pose. The design of this project closely flows with the *Location stack model*.

4.1 The Location Stack Model

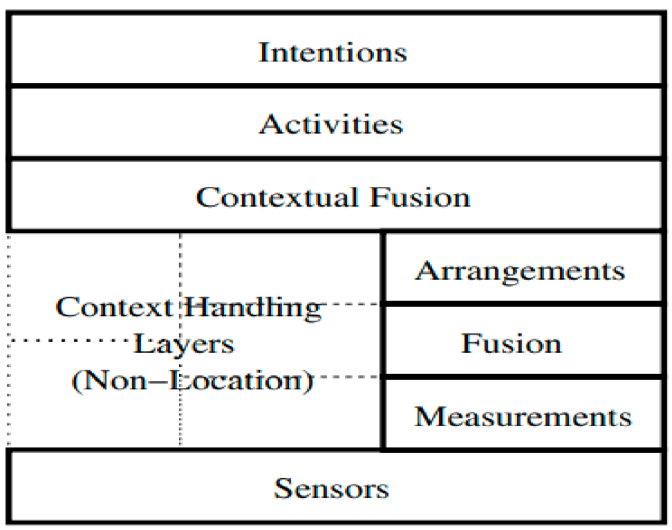


Figure 4.1: The seven-layer location stack model[14]

The Location Stack model is an abstraction proposed by Hightower et al. for location-based computing. The model is set to serve as the standard infrastructure from which location-based computing is built upon, encouraging a community in which designers can have a single protocol to evaluate and cooperate. It follows the same model as the Open system interconnect

(OSI) model in computer networks. "The Location Stack is multiple sensing technologies supporting multiple applications with reusable middle" [14], as shown in figure 2 above. The first three layers in the location slack model describe the services built on top of location-based applications.

In comparison, the last bottom three layers summarize the mechanisms used in location-based applications. The sensor layer describes the physical sensors used, whiles the measurement layer describes the measurements achieved by the sensors used. However, the fusion layer introduces computer algorithms that capitalize on the obtained measurements to predict a tracked object's position and orientation. Deterministic fusion algorithms such as multi-lateration and multi-angulation are applied to single sensor measurements to measure multiple distances and angle of arrival[20]. Multiple sensor measurements use the Bayesian algorithms for sensor fusion since multiple sensor measurements come with different sampling rates from each sensor with their unique associated errors. Bayesian filters measures both certainty and uncertainty of the position of a tracked object, thereby considering the errors associated with each measurement taken by each sensor[21]. This makes multiple sensor fusion techniques very resilient than single sensor techniques.

The arrangement layer in the location stack model depicts the association between two or more tracked objects. In contrast, the contextual fusion layer describes the location data merger with information that has no location elements. This ensures proper organization of the process in a location-based application. The activities layer is a system that classifies contextual information into activities using models like machine learning. Finally, the intensions layers make use of the user's intentions and implement the user's desires.

4.1.1 Bayesian filters

The quest to fuse multiple sensors together in localization applications has made Bayesian filtering very desirable in addressing the uncertainties accompanied by the measured signals from sensors. In the case of a dynamic environment where there is noise , Bayesian filters allow for certainty to be established with the measured signal values using probability[22]. In a dynamic environment, the state x, of a measured signal, given the noisy measurement, z, at time t, can be expressed with the Bayesian filter as.

$$Bel(x_t) = p(x_t | z_1, \dots., z_t) \quad (4)$$

This is interpreted as the probability of a measured state of a signal given that there is noise associated with the measured signal.

The probability distribution allows Bayesian filters to estimate a tracked object's true position and the amount of uncertainty associated with the calculated position. The different types of Bayesian filters include the particle filters, grid-based filters, multi hypothesis filters, linearized and extended Kalman filters, and Kalman filters[21]

This project consists of three major parts, the Visual Inertial Odometry, where computer vision algorithms in conjunction with Inertial sensors will be used to determine the pose of a mobile device, the Received Signal Strength Indicator, where the signal strength from routers/ access points will be used for positioning, and finally, merging the two methods, VIO and RSSI together. For a better user experience, an Augmented Reality experience is incorporated for use in a mobile device.

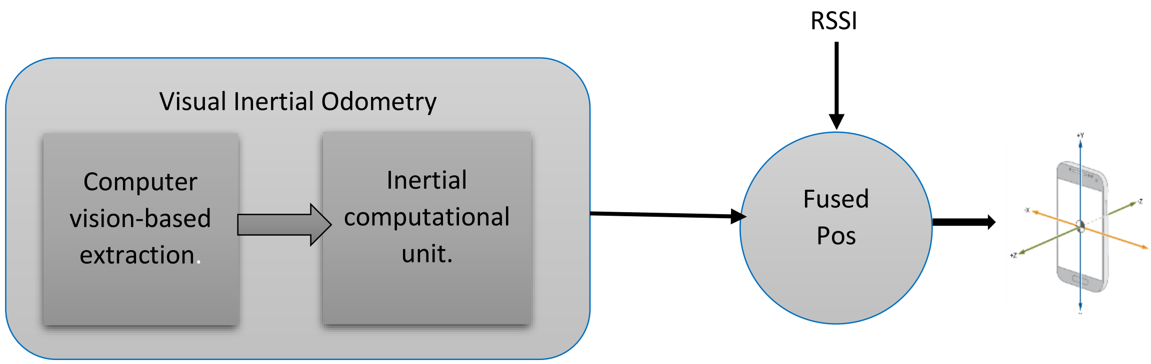


Figure 4.2: Block diagram of summarized project design

The system's design scopes three operational stages of the mobile application, including environment mapping, Localization, and resolving the mapped environment for multiuser access.

4.2 Environment Mapping

Environment mapping involves collecting the visual feed within the indoor environment needed for localizing using the video feed from a mobile phone's camera. The mapping also includes collecting RSSI values to create a radio database for the localization algorithms. The visual feed with orientation data and the RSSI radio map will serve as the reference information for relocalization.

4.2.1 Vision-Based Environmental Mapping

The ability of the system to understand the environment and create a 3D map without any prior knowledge of the environment can be greatly achieved with Simultaneous Localization and Mapping (SLAM)[23].

4.2.1.1 Simultaneous Localization and Mapping (SLAM)

Simultaneous Localization and Mapping (SLAM) is a localization method that maps and understands the surrounding environment and can recognize the environment at a later encounter. With computer vision, a video feed from a device camera is analyzed to identify peculiar scenes and feature points (planes) used to establish the spatial awareness of the environment. One problem associated with SLAM is the errors accumulated due to the device's movement [24]. These errors are minimized with the incorporation of Inertial sensors. Visual Simultaneous Localization and Mapping (VSLAM), which is the technology that uses SLAM, allows the camera as the only sensor to visually map the environment and returns the position and orientation of any tracked object within the environment. A real-time implementation of SLAM introduces computational complexity, data association, and environmental representation challenges[25]. Improving the computational efficiency of SLAM algorithms involves using optimization techniques that seek to optimize the computational power needed and produce localization estimates whose covariance represents the full-fledged SLAM algorithms. Improving the computational efficiency of SLAM also involves applying conservative algorithms that result in higher covariance estimates with reduced computational complexity compared to the optimal algorithms. Sub mapping, State augmentation, sparsification, and partitioned updates are some of the algorithms that can help reduce the computational requirements of SLAM. The issue of data association can be tackled with multi hypothesis, assigning appearance signatures, and batch validation. On the other hand, environmental representation can be tackled with nongeometric landmarks, delayed mapping, and trajectory estimation methods[24].

The implementation of a three-dimensional SLAM further introduces complexities into modeling the system. However, 3D SLAM can be achieved with three techniques.

i) An implementation of a 2D SLAM with additional mapping in the third dimension

ii) An extension from 2D to 3D SLAM with feature mapping.

iii) Aligning pose estimates with 3D scan of the environment.

This project will implement an extension of 3D SLAM from 2D using feature extraction mapping for accurate 6-degrees-of-freedom (6DoF) pose estimation.

Davison et al. [26] used the probabilistic estimates of the current environment feature map using a device camera, in which pose information is established with the camera's current state. Using the extended Kalman filters, an initialized state of a mapped environment is dynamically updated until the mapping process is ended. Extended Kalman filters allow for the state of the mapped camera feed to be determined probabilistically whiles updating these states when the camera is in motion. This enables the states of the mapped feature points to constantly get updated when new mapped features in the environment are added. Since the created map is based on probability, its faith over time depends on both the certainty(best) of the estimated camera and features states and the number of deviations that might accompany the estimated states. The number of deviations associated with the states is a first-order distribution; hence, mathematically, the obtained map can be expressed as the certain states of the camera and feature points and the uncertain covariance as shown below.

$$\hat{X} = \begin{pmatrix} \hat{x}_v \\ \hat{y}_1 \\ \hat{z}_2 \\ \vdots \end{pmatrix}, P = \begin{bmatrix} P_{xx} & P_{xy_1} & P_{xy_2} & \cdots \\ P_{y_1x} & P_{y_1y_1} & P_{y_1y_2} & \cdots \\ P_{y_2x} & P_{y_2y_1} & P_{y_2y_2} & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix} \quad (5)$$

From the expressions above, $\hat{X}$ is the state vector representing the best measured states of the Camera and feature points in the environment. P represent the measured probabilistic deviation (covariance) matrix . The state of the camera, x_v in the state vector describes the 3D vector position of the camera, its orientation quaternion, the velocity, and angular velocity vectors. The features states y_i in the state vector, describe the location of the feature points in the environment. The camera frame vectors with respect to the world frame geometry in figure 4 below depicts how the camera maps and understands its position, and orientation in the physical world.

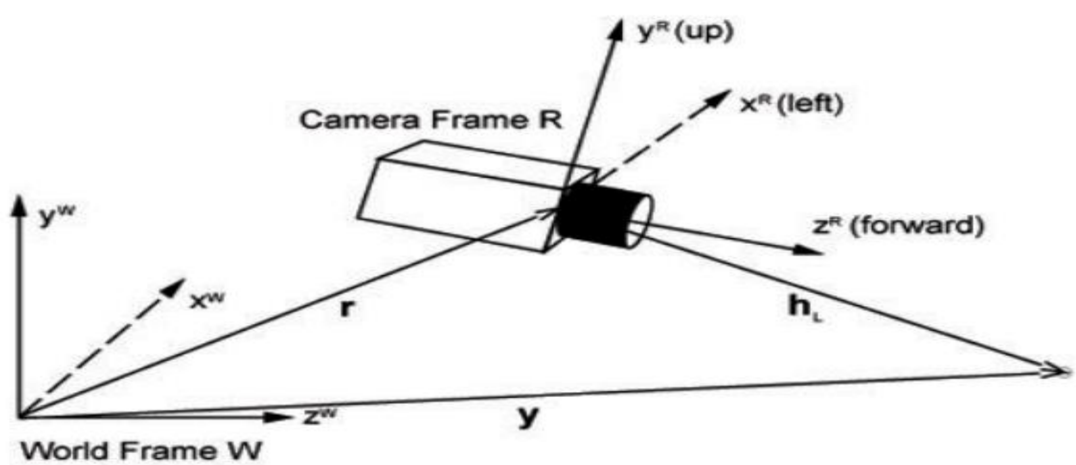


Figure 4.3: Coordinate frames and vectors of camera and feature states geometry

4.2.2 RSSI Environment Mapping

The environmental mapping for the received signal strength indicator values for this application will be achieved by scanning within the environment to collect the signal strength values from various access points. The application will require a storage medium like a database which will be hosted in a remote server. The received signal strength indicator values will only constitute values from routers and or access points. This is to reduce the error of pose estimation due to the blockage of the radio waves from cell towers which might encounter many obstacles. Besides, access points provide flexibility due to the already established access points infrastructure in areas where this application will be used.

4.3 Localization

Localization refers to the process of estimating the position and orientation of a tracked body in space. Localization helps in space awareness and compares the relative positions between two points within an environment. This project utilizes localization algorithms to understand the geometric coordinates of an individual in an indoor space.

4.3.1 Localization with Vision-Based

The vision-based localization design is composed of the integration of visual Localization and Inertial measurements from the Inertial Measuring unit (IMU) of mobile phone devices. This integration is called Concurrent Inertial Odometry and Mapping or Simultaneous Visual Inertial Odometry.

4.3.1.1 Simultaneous Visual Inertial Odometry

Simultaneous Visual Inertial Odometry tracks the motion of the camera of the device from which visual data from the environment is fed, combined with the device's gyroscope and accelerator data for accurate 6-degrees-of-freedom (6DoF) pose estimation. The gyroscope data indicates the device's position and orientation, while the accelerometer measures its movement with respect to the physical surroundings. Accounting for the force acting on the device while in motion helps reduce latency and ensure that the correlation among estimated variables(position and orientation) and their noise characteristics are discarded.

4.3.2 Localization with RSSI Based

The two principal methods can achieve the implementation of RSSI in localization estimation. One of the approaches is converting the received signal strength indicator values into distance models like the radio wave propagation model. The resulting distance values serve as input into a localization algorithm for pose estimation. The second approach called the power-based method, directly uses the received signal strength indicator values as the input for a localization algorithm. This project will use the power base approach to reduce the implementation complexities and allow the system to easily adjust to the removal or addition of new access points in the environment. The various localization algorithms used in RSSI modeling compose of deterministic algorithms and probabilistic algorithms. Probabilistic algorithms have been proven to be less prone to noise and are does not require an online database to store RSSI values. Probabilistic algorithms, however, suffer from increased computational requirements. Examples of probabilistic algorithms include the Bayesian regression[8], nonlinear multilateration. The deterministic algorithms, on the other hand, require less computational power and easier to implement. Examples of deterministic algorithms include support vector machines, K nearest neighbor (KNN), and Artificial neural network (ANN)[9], Root Mean Square Error (RMSE).

4.3.2.1 Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) is a measure of the standard deviation between a predicted RSSI value and the actual value. This project will use the Root Mean Square Error (RMSE) method for Localization. In the dynamic environment where this project shall be used, the RSSI measurements received by a device will always change over time and will not produce stable values. Hence the use of RMSE will allow the system to measure the RSSI values with the least deviation between the observed values and the recorded values. Root Mean Square Error is easier to implement compared to the other methods and can combine the various induced errors over time into a "single measure of predictive power" [27]. The mathematical modeling of RMSE is shown below.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)^2}{N}} \quad (6)$$

The Actual RSSI values will be stored in a database after scanning the environment for RSSI values. The predicted values are the RSSI values observed by the application while trying to localize an individual in real-time.

4.4 Resolving

Resolving in this application is the processes involve in making localization information of the current environment available to a user in the environment. This localization information is a result of the mapped information of the environment hosted on a local server. The resolving information will consist of the visual feature points, Inertial pose data, and a radio map of Mapped RSSI values in an indoor space. The set of resolving information will be stored in a database and be accessible to any user identified in the indoor space where the formation was taken. Sharing routing information among different users and allowing them to interact it is referred to as a multiuser experience.

4.4.1 Multiuser Experience

One of the objectives of this project is to allow different users to share routing information and make it available for others to use. Therefore, this comprises an *administrator* that will host the various routes in a particular indoor space, such that those routes can be readily retrieved by a navigating user in the same environment. Every unique indoor space must have only one administrator. The administrator can host a route, edit, or discard the route. A route is a point between the current location of a user in an indoor environment and the destination of interest. Example of a route is navigating from office A to the washroom.

To keep track of the different **admin** users for different indoor spaces, each admin is provided a unique session with the help of a database. The database will store the unique email address, password, and name of the admin user and provide storage to host the admin's unique routes and unique WiFi SSIDs, which will serve as the identity of that indoor space. A radio map of Received Signal Strength Indicator values in the environment will be stored for the admin. The password and email address are to enable secure access to the hosted information of an indoor space. The unique SSIDs tags an indoor space and help position users in the right direction for accurate Localization.

The **navigating user** on the other hand, will be made available routing information depending on the indoor space in which the user is localized. The application uniquely matches the navigating user with the admin's routing information which that unique SSIDs tag. This will ensure that the associated cost of hosting the login details for navigating users is reduced since the navigating user is not required to uniquely login, thereby easing the steps for the navigation.

The finalized design consists of scanning phase, where a camera SLAM will capture an indoor space's visual feature points. The RSSI values of the space are scanned to form a radio map. Simultaneous Visual Inertial Odometry algorithms are applied on the visual feed data while the Root Mean Square Error is applied in the radio mapped data for Localization. The localization formation is stored in a database and is shared among devices for a multiuser experience. Below is a summarized architecture of the proposed design.

4.5 Summarized Architecture of proposed Design

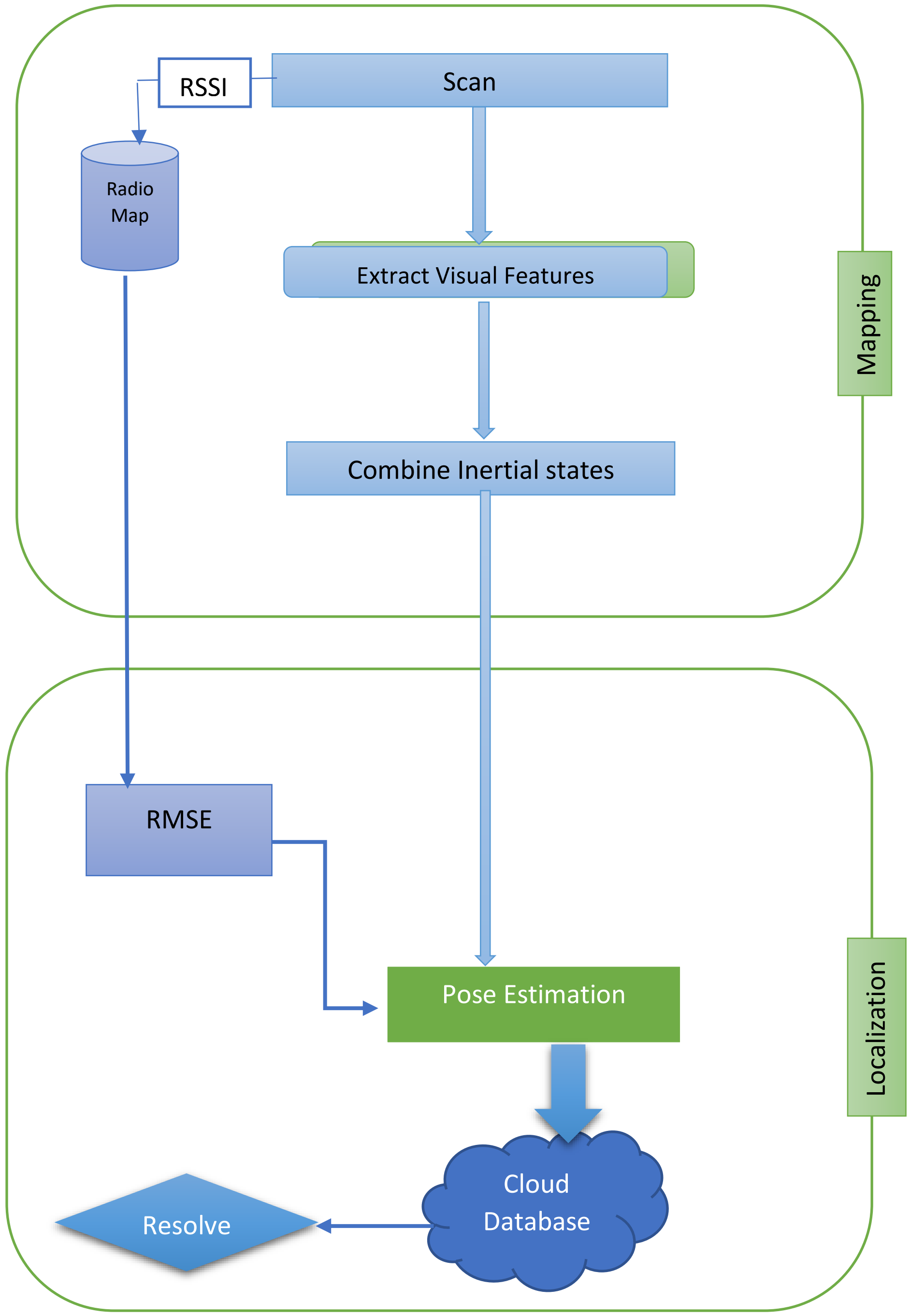


Figure 4.4: Structure of proposed design

Chapter 5: Implementation

The project's implementation is also divided into three categories, the vision-based tracking using Google's ARCore, the Root Mean Square Error implementation for RSSI, and the fusing of the two methods using Unity, an Augmented Reality enabled SDK. The mapped visual Odometry features, and radio map of the indoor environment are pre deployed in a remote database called firebase. The use of firebase will make multiuser experience possible and reduce the processing power required to run the application.

ARCore is a software development kit for Augmented Reality applications developed by Google and released in 2018. ARCore provides the capability of using the device's camera to understand feature points in the environment and track their movements with time. These data combined with the device's Inertial sensors (gyroscope and accelerometer) are used to determine the device's position and orientation with respect to its physical environment. ARCore also provides light estimation capability and motion tracking. The light estimation capability means ARCore can measure the light intensity of a particular environment. This is particularly important to this project as knowledge of indoor lighting conditions will help the localization algorithms adjust and make the application useful in all lighting conditions. Therefore, the Google ARCore SDK has the visual-based Localization and Inertial positioning algorithms ingrained into it as one package for deployment to mobile devices. Enhancing the user experience with Augmented Reality is achieved by digitally adding to the graphical user interface 2D or 3D objects to show direction and other important information to the user, increasing the user's immersion in the physical environment. The Unity SDK achieves this.

Unity is a game development cross-platform software that enables 2D and 3D experiences like Augmented Reality and Virtual Reality. With Unity AR Foundation framework, developers can deploy Augmented Reality content across various mobile and wearable devices. AR foundation provides the platform; hence implementing AR content requires platform-specific plugins. In this project, the application will be deployed to an android device; therefore, ARCore XR Plugin is chosen to implement both the visual Inertial Odometry and the Augmented Reality experience.

5.1 Environment Mapping Implementation with ARCore

ARCore can identify peculiar features of the environments in which the camera sees. These landmarks are referred to as feature points. The feature points are then transformed into planes, tracking horizontal surfaces, vertical, or every surface. To visualize parts of the physical environment that is tracked, a special shader is used to indicate the tracked surface being transformed into a plane. The process of transforming tracked surfaces into planes is called **Meshing.** Meshing ensures that the formed planes are do not get destroyed even when the camera is moved away from that tracked surface. The generated planes of the environment serve as the map of the environment for Localization. The visual tracked plane data is called an Anchor. In Unity, the "ARPlane" class is a representation of the tracked planes. The "AR Plane Mesh Visualizer" creates the visual representation on the planes. The implementation of environment mapping is as shown below.

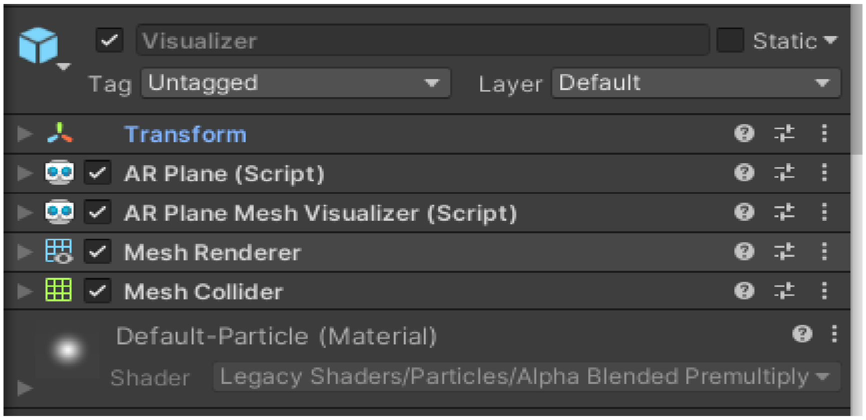


Figure 5.1: Implementation of environment mapping in Unity with ARCore

5.2 Motion Tracking Implementation with ARCore

Motion tracking occurs along with the environmental understanding implementation. Motion tracking, however, allows the device to understand the location of the tracked planes in the environment. Motion tracking uses Visual Inertial Odometry for determining the Localization of feature points. The VIO of ARCore usually combines the visual camera feed(tracked planes) combined with the inertial measuring unit of a device to remember the position and orientation of the device in the physical environment where a particular feature point was tracked, hence the location of the tracked plane. The visual data with the Inertial measurement of a tracked plane forms a 3D of the environment called **Point Cloud**. Point clouds ensure that the moving device may not mistake one tracked plane for another, further increasing the robustness of the map of an indoor environment. The 3D Vector position of a point cloud starts from zero as the ARCore session starts. In this project, smaller red cubes are used to visualize the tracked point clouds in the environment. This is achieved by the point cloud manager class in Unity.

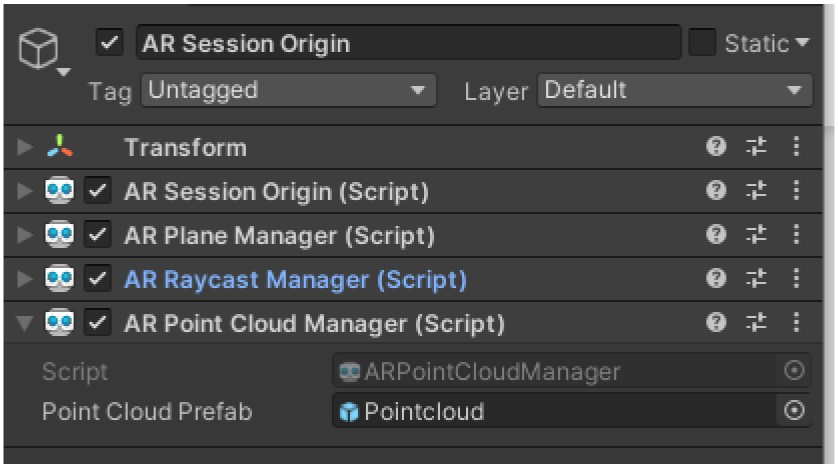


Figure 5.2: Implementation of motion tracking in Unity with ARCore

5.3 Light Estimation Implementation with ARCore

To mimic the real world in a virtual scene, ARCore applies light estimation algorithms on virtual objects, so they can look nearly real with the same lighting conditions as the physical world. This is achieved by analyzing the image data of the current frame and then apply the calibrated lighting estimation to the rest of the frames in the current environment. In this project, however, the light estimation is valuable in indicating whether the Localization and resolving of cloud points would be possible. Hence if the light estimation registers a value at which the plane detection can no longer work, then that value becomes the indicator for Localization with the received signal strength indicator method.

5.4 Anchors and Trackable

The Google ARCore allows virtual content to be placed in the AR world by the user. This is possible if an anchor is established of the area in the environment in which the virtual content will be placed, allowing ARCore to track the placed virtual content over time. The virtual content is instantiated based on the position and orientation of the anchor. All established anchors are updated in the environment over time (trackable), to ensure that ARCore is returning the right pose of the tracked features and the instantiated virtual content. In this project, a user in an indoor environment will have to place virtual directional arrows from the beginning of a route to the destination. These instantiated arrows with their respective position and orientation as well as the visual features points in the indoor environment, are stored in a remote database. This data is then associated with the route and is made available to other users to be relocalized and aid in routing in the same indoor environment. The user that will place the virtual arrows in an indoor environment is the **Host**, whiles the one that uses the hosted data for relocalization is the **Resolver**.

In an indoor environment, the name of the current location of the user and the destination the of a route is use as the id for all the anchors in the specified route. Firebase real-time database is used for storing and retrieving anchors. As such, the Host user must be registered, and authenticated to access previously hosted routes, edit routes, add more routes, or delete routes. The Host also selects the SSIDs of access points within the indoor environment to associate with all the hosted routes of that indoor environment. This is to ensures that the hosted routes of indoor space are publicly available to anyone within that indoor environment. The implementation of dynamically determining if a user is within a hosted indoor environment is achieved by storing the SSIDS and the received signal strength of the access points hosted by the Host user in a firebase real-time database.

However, the resolver user needs no authentication but is instead localized in an indoor environment using the stored SSIDs and their received signal strength values. The Root Mean

Square Root algorithms evaluate the difference between the mean value of the RSSI values registered by their device and the actual values stored in the database.

5.5 Radio Map Construction.

A radio map is a database containing the received signal strengths of access points within a particular geographical area. The information that comes with the signal received from an access point includes the name of the access point (SSID), the RSSI value in dB, and the link quality indicator (LQI). However, the SSID and the received signal strength indicator value uniquely identifies each access point [28]. The radio map is created by measuring the WiFi signals at reference positions and recording the respective RSSI values of the SSIDS. To increase the accuracy of the positioning model, the WiFi signals at each reference point are measured several times, and the average value of all measured RSSI values is assigned to the reference position. By clicking on the "Scan" button on the application and indicating the reference point for each scan, the radio map generated is in form below.

$$\text{RSSI} = \begin{bmatrix} P_{AP1(1)} & P_{AP1(2)} & P_{AP1(N)} \\ P_{AP2(1)} & P_{AP2(2)} & P_{AP2(N)} \\ P_{APK(1)} & P_{APK(2)} & P_{APK(N)} \end{bmatrix} \quad (7)$$

From the set of all recorded RSSI values, $P_{APK(N)}$ represents the signal strength measured at a reference position N, from the Kth access point. Hence, N represents the name of the reference position indicated by the user, and K is the name of the SSID from which the RSSI value was measured and averaged. The matrix is formed from K by N matrix *(number of access points* x *number of reference points*).

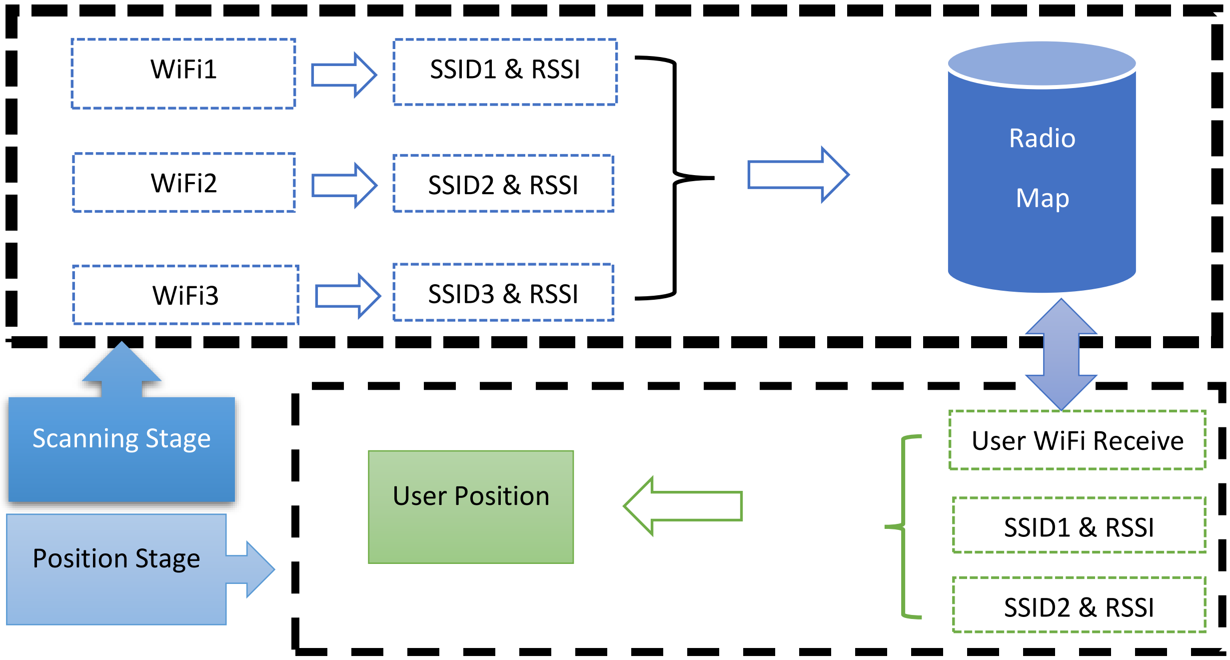


Figure 5.3: Overview of radio map construction

The application scans four times every two minutes. This allows the user to take multiple RSSI values from each reference point and allows the user to move around the reference point. Moving around the reference point is important because it allows the application to measure an average RSSI value that truly represents the reference point since RSSI values can easily change within a dynamic environment.

During routing by a resolver user, the Root Mean square error algorithm is used to measure the Euclidean distance between the RSSI value of an access point (AP) in the radio map and that received by the user device from the same access point in real-time.

$$distance(j) = \sqrt{\sum_{i=1}^{n} (AP_i - AP_{Ri})^2} \quad (8)$$

Where AP_{Ri} is the RSSI value measured from an access point with SSID name i in real-time and the RSSI value of the same access point with SSID i from the radio map. The value of $distance(j)$ is the sum of the smallest deviations of each access point the user's device connects to in real-time and their corresponding access point stored in the radio map. The true deviation is therefore the average of the sum of deviations from all **n** access points. Hence, the RMSE value.

$$deviation(true) = \frac{distance(j)}{n} \qquad (9)$$

The bigger the difference between the true deviation and the Euclidean distance($distance(j)$) of the smallest deviation of a particular access point, the higher likelihood that the user is in the location associated with that access in the radio map. The computer algorithms for the RSME implementation is described below.

Algorithm 1: Overview Implementation of RMSE

Initialize system;

foreach detected SSID **do**

  Compare detected SSID with all SSIDs from database;

  if a detected SSID matches with SSIDs from database **then;**

    Compute the Euclidean distance between the detected SSID and the matched SSID in database;

    Return the SSID from database which resulted as the smallest Euclidean distance;

  end

end

Compute the average of the smallest Euclidean distance of matched access points from database;

Return the Location associated with matched AP from database with biggest deviation from above compute result;

Chapter 6: Testing and Results

This chapter discusses the experiment conducted to test the accuracy of the system and the results obtained.

6.1 Experimental Set-Up

A 351cm by 351 cm, indoor room was used for testing under different lighting conditions. Ten circles, each with a diameter of 10 cm were drawn on the floor at an interval of 30cm between each other. They were drawn in a straight line to signify the route from one point to another. The setup is as shown below.

Figure 6.1: Setup room with drawn lines for the experiment

By varying the lighting conditions of the room, cloud anchors, exactly the size of the circles were spawned to fit each circle. The lighting conditions were varied by leaving the main door of the room *fully opened*, *half-opened* and *fully closed*. The application is then switched to the resolving section where each stored anchor under different conditions is supposed to reappear at the same position in the circle from which they were spawned and hosted. The experiment was conducted five times within the same illumination from the outdoor environment. The time taken for the relocalization of anchors in each were also recorded. The various anchor deviations were recorded as shown the table under the appendix.

6.2 Results

A summarized version of the deviations calculated from the different lighting conditions is shown below.

Table 6.1: Summarized obtained results

Door position	Error (cm)	Relocalization time (s)
Fully Opened Door	19.6	7
Partially Closed Door	7.12	4
Fully Closed Door	23.46	10

The *fully closed*-door setup resulted in an accumulated error of 23.6cm. It is the largest error among all the setups. This error was primarily due to the inability of the vision-based algorithms to track feature points in the dark environment; hence most of the anchors were drifting away from the actual positions over time. Also, the *half-opened* door experiment performed better than the fully opened setup. This was an unusual result because the computer vision algorithms work better when there are better lighting conditions. However, better lighting conditions do not always mean a high-intensity lighting condition. The *fully opened* setup allowed a lot of light rays into the room, thereby increasing the amount of light reflected by the room's floor. The reflection consequently made the floor assume the same texture and color; hence the visual feature points became indistinguishable and resulted in poor tracking by the VIO algorithms. The difficulty with quickly identifying feature points meant the time taken for the relocalization to start increases. Therefore, the higher the relocalization time, the higher the errors associated with the placed anchors. Below is a summarized visual depiction of the obtained results.

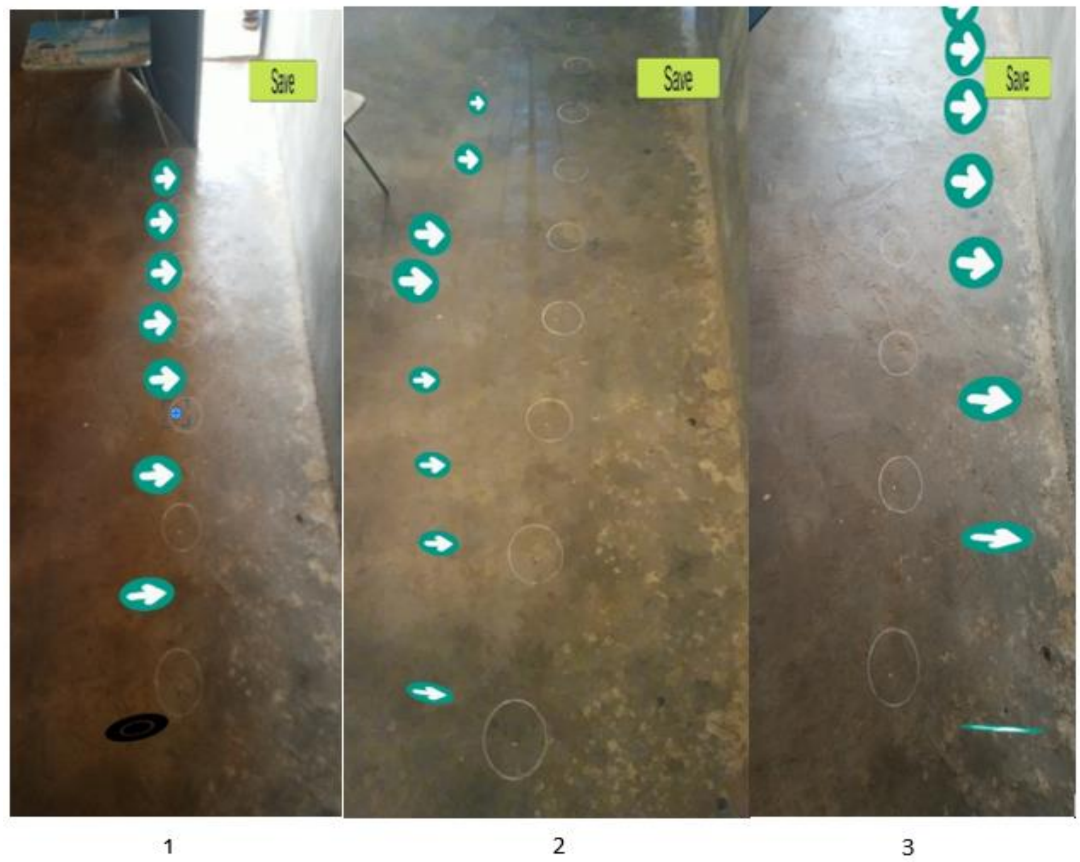


Figure 6.2: Partially opened door(1), fully closed door(2), fully opened door(3)

Chapter 7: Conclusion

The quest to develop an indoor equivalent for GPS was steered into considering the various indoor positioning methods, how some of them work, and the advantages and limitations they present. It became apparent that combining the power of two or more indoor positioning has greater benefits beyond the limits of individual methods, hence the objective of this project. Various vision-based localization algorithms combined with the Root Mean Square Error algorithms for the Received Signal Strength Indicator values from Access points presented the possibility of creating a robust and more accurate indoor positioning system for use in all environments irrespective of the lighting conditions. The result of the system showed the vision-based Localization achieved a high accuracy level with an average error of 16.73 cm. There have been, however, several challenges faced in the implementation of this project and below, I will discuss the limitations of this project and future work suggestions that could further improve the project.

7.1 Limitations

One of the main limitations encountered with the implementation of this project is the integration of the Received Signal Strength Indicator algorithms and the Visual Inertial Odometry algorithms by the Google ARCore framework. The ARCore SDK does not make the trackable data available for developers to integrate other methods to improve the tracking. As a result, the RSSI integration could not be achieved for anchor localization. However, the integration was useful in localizing the user in an indoor space and serving as a signature for each indoor space, allowing a navigating user to identify with the hosted routes of an indoor space.

In addition to the above challenge, the application developed can only be run on selected android devices with API Levels above 14. This requirement comes with the Google ARore SDK determined by the CPU of the device, the screen size, the camera, and motion sensors of the device. A device that is not certified by Google and is not compatible with ARCore will not be able to use this application.

7.2 Future work

The implementation of the proposed design has covered most of the objectives set out. However, there are certain areas of the design and implementation that could be improved or added. Hence for future work, I recommend the following.

- The integration of a map interface representing an indoor space will be useful as it will give a navigating user a fair idea of the whole indoor space and communicate the distance between the user's current position and the intended destination. The map could be displayed on one side of the screen whiles the augmented reality scene is displayed on the other side. The map could also be integrated into the global outdoor navigation maps like Google Maps to indicate the position of the indoor space relative to the global map.
- Also, further work can be done on integrating Shortest Path algorithms like Dijkstra. The shortest path algorithms will calculate the shortest possible path for a navigating user from the current location to the intended destination. This is particularly useful for complex indoor spaces like the mall, where there might be more than one route to a particular destination.

- Finally, I recommend further work to make the Root Mean Square Error algorithm more robust and accurate. In a dynamic environment where access points and routers are not guaranteed to remain permanent in an indoor space, the localization algorithms must constantly adapt to newly added or removed access points. This can be achieved by implementing a Machine learning model that will make the system learn the dynamics of a particular indoor space and easily predict a navigating user's position.

Appendix

A. Data Collected from Experiment

Table A.1: Results for when the door is partially opened.

Anchor	Deviation(cm)				
	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
Anchor 1	20	3	21	16	22
Anchor 2	13	9	12	10	12
Anchor 3	10	11	10	9	6
Anchor 4	9	3	5	7	6
Anchor 5	5	3	3	2	5
Anchor 6	1	3	3	4	2
Anchor 7	2	4	6	5	5
Anchor 8	3	5	5	4	6
Anchor 9	4	6	7	8	8
Anchor 10	2	9	9	5	8
Total deviation(cm)	6.9	5.6	8.1	7.0	8.0

Table A.2: Results for when the door is fully opened.

Anchor	Deviation(cm)				
	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
Anchor 1	25	18	18	20	22
Anchor 2	23	15	16	21	21
Anchor 3	17	16	15	22	16
Anchor 4	23	16	17	19	16
Anchor 5	19	15	17	13	15
Anchor 6	21	17	18	22	20
Anchor 7	13	16	15	17	15
Anchor 8	13	16	14	15	16
Anchor 9	9	15	14	16	18
Anchor 10	4	14	13	18	18
Total deviation(cm)	16.7	15.8	15.7	18.3	14.1

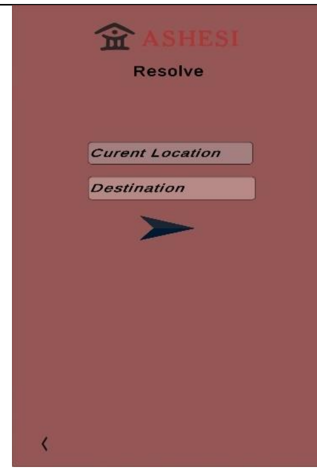
Table A.3: Results for when the door is fully closed.

Anchor	Deviation(cm)				
	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
Anchor 1	14	9	10	13	14
Anchor 2	15	6	23	20	1
Anchor 3	23	9	22	21	23
Anchor 4	35	13	24	23	25
Anchor 5	36	40	38	33	36
Anchor 6	25	37	33	23	32
Anchor 7	19	38	31	33	25
Anchor 8	17	27	26	27	26
Anchor 9	16	39	22	35	15
Anchor 10	23	17	19	22	18
Total deviation(cm)	22.3	23.7	24.8	25.0	21.5

Application User Interface

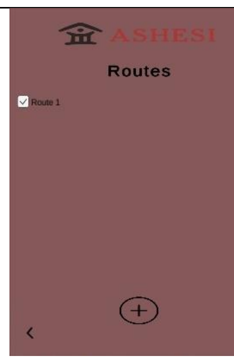
HomePage

Navigation clicked



Admin Clicked

Login Clicke d



Add Rout e

Create Account Clicked

Reference

[1] S. Gleason and D. Gebre-Egziabher. GNSS applications and methods. *Artech House*, 2009.

[2] Yunck, T., Liu, C. H., & Ware, R. (1999). A history of GPS sounding.

[3] G. Dedes and A. Dempster. Indoor GPS: Positioning challenges and opportunities. *In Proceedings of the 62nd IEEE Vehicular Technology Conference* (VTC 2005-Fall), pages 412–415, 2005.

[4] Keisanen, O. (2018). Indoor Signal Mapping Using ARCore. School of Sciences. *Aalto University*.

[5] Kunhoth, J. Karkar, A., Al-Maadeed, S., & Al-Ali, A. (2020). Indoor Positioning and Wayfinding Systems: a survey. *Humancentric Computing and Information Sciences*. https://doi.org/10.1186/s13673-020-00222-0.

[6] Kulich, T. (2019). Indoor navigation using vision-based Localization and Augmented Reality. *Department of Information Technology, Uppsala University*. Retrieved from http://diva-portal.org/smash/record.jsf?pid=diva2%3A1345994&dswid=-9311.

[7] Piasco, N., Sidibé, D., Demonceaux, C., & Gouet-Brunet, V. (2018). A survey on Visual-Based Localization: On the benefit of heterogeneous data. Pattern Recognition, *74, 90–109*. doi:10.1016/j.patcog.2017.09.013 .

[8] Qin J., Sun S., Deng Q., Liu L., Tian Y. Indoor trajectory tracking scheme based on Delaunay triangulation and heuristic information in wireless sensor networks. *Sensors*. 2017;17:1275. doi: 10.3390/s17061275.

[9] D. Titterton and J. Weston. Strapdown Inertial Navigation Technology. *The American Institute of Aeronautics and Astronautics*, 2nd edition, 2004.

[10] O. Pathak, "WiFi Indoor Positioning System Based on RSSI Measurements from WiFi Access Points –A Tri-lateration Approach," *International Journal of Scientific & Engineering Research*, Volume 5, Issue 4, April-2014 ISSN 2229-5518.

[11] Yang, Z., Wang, Z., Zhang, J., Huang, C., & Zhang, Q. (2015, May). Wearables can afford: Light-weight indoor positioning with visible light. In *Proceedings of the 13th annual international conference on mobile systems, applications, and services* (pp. 317-330).

[12] P. Groves. Principles of GNSS, Inertial, and multisensor integrated navigation systems Artech House, 2008.

[13] G. Welch and E. Foxlin. Motion tracking: No silver bullet, but a respectable arsenal. *IEEE Computer Graphics and Applications*, 22:24–38, 2002.

[14] J. Hightower, B. Brumitt, and G. Borriello. The location stack: A layered model for location in ubiquitous computing. *In Proceedings of the 4th IEEE Workshop on MobileComputing Systems and Applications* (HotMobile 2002), pages 22–28, 2002.

[15] Corotan, A., & Irgen-Gioro, J. J. Z. (2019, April 1). An Indoor Navigation Robot Using Augmented Reality. *IEEE Xplore*. https://doi.org/10.1109/ICCAR.2019.8813348

[16] Taniura, Y., & Oguchi, K. (2017, July 1). Indoor location recognition method using RSSI values in system with small wireless nodes. *IEEE Xplore*. https://doi.org/10.1109/TSP.2017.8075935

[17] Yilmaz, A., & Gupta, A. (2016, October 1). Indoor positioning using visual and Inertial sensors. *IEEE Xplore*. https://doi.org/10.1109/ICSENS.2016.7808526

[18] Rajagopal, N., Miller, J., Kumar, K. K. R., Luong, A., & Rowe, A. (2019). Improving Augmented Reality relocalization using beacons and magnetic field maps. In 2019 *International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (pp. 1-8). *IEEE.*

[19] Wu, Z., Fu, K., Jedari, E., Shuvra, S. R., Rashidzadeh, R., & Saif, M. (2016). A Fast and Resource Efficient Method for Indoor Positioning Using Received Signal Strength. *IEEE Transactions on Vehicular Technology*, *65*(12), 9747–9758. https://doi.org/10.1109/TVT.2016.2530761

[20] A. Bensky. Wireless positioning technologies and applications. *Artech House*, 2008.

[21] Woodman, O., & Harle, R. (2008, September). Pedestrian localisation for indoor environments. In *Proceedings of the 10th international conference on Ubiquitous computing* (pp. 114-123).

[22] Simon, D. Optimal state estimation: Kalman, H∞, and nonlinear approaches. *Wiley*, 2006.

[23] A. J. Davison. Real–time simultaneous localization and mapping with a single camera. *In Proceedings of the Ninth IEEE International Conference on Computer Vision*, volume 2,

[24] Teixeira, L., Alberto , B., & Gattass, M. (2013). Indoor Localization using SLAM in parallel with a Natural Marker Detector. *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, 27-33. Retrieved from http://webserver2.tecgraf.puc-rio.br/~abraposo/pubs/SAC2013/Lucas_SAC2013.pdf.

[25] Bailey, T., & Durrant-Whyte, H. (2006). Simultaneous Localization and mapping (SLAM): Part II. *IEEE robotics & automation magazine*, *13*(3), 108-117.

[26] Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM. *IEEE transactions on pattern analysis and machine intelligence*, *29*(6), 1052-1067.

[27] Maduskar, D., & Tapaswi, S. (2017). RSSI based adaptive indoor location tracker. *Scientific Phone Apps and Mobile Devices*, *3*(1), 1-8.

[28] M. M. Atia, A. Noureldin, and M. J. Korenberg, "Dynamic onlinecalibrated radio maps for indoor positioning in wireless local area networks," *IEEE Trans. Mobile Comput*., vol. 12, no. 9, pp. 1774–1787, Sep. 2013.