Triangulation-Enhanced WiFi-Based Autonomous Localization and Navigation System: A Low-Cost Approach

Suleiman Abu Kharmeh
Department of Computer Engineering
An-Najah National University
Nablus, Palestine
sabukharmeh@najah.edu

Emad Natsheh
Department of Computer Engineering
An-Najah National University
Nablus, Palestine
e.natsheh@najah.edu

Rahaf Nasrallah
Department of Computer Engineering
An-Najah National University
Nablus, Palestine
s11923488@stu.najah.edu

Masa Masri
Department of Computer Engineering
An-Najah National University
Nablus, Palestine
s11923986@stu.najah.edu

***Abstract*— This study investigates the use of WiFi signal strength for localization and navigation in low-cost robotic systems, focusing on indoor environments. Mathematical models are developed based on signal strength measurements, integrated with triangulation techniques for dynamic robot positioning. Both logarithmic and linear regression techniques are employed to determine the distances between the robot and access points. Through experimental verification, accurate localization abilities are showcased. Various navigation algorithms, including A*, are evaluated for effectiveness. Overall, this study presents a low-cost system for indoor autonomous positioning and navigation, with potential applications in home care, inventory management, and emergency support.**

***Keywords*— indoor localization; low-cost; triangulation; signal strength indicator; navigation**

I. Introduction and Related Work

The utilization of WiFi signal strength for localization and navigation purposes in low-cost systems is demonstrated in this study. The system can estimate the position of a moving robot dynamically which enables the use of such information in navigating indoor environments. The approach offers a promising solution for location-based services and indoor navigation systems, utilizing existing WiFi infrastructure for cost-effective implementations in robotic systems.

Mathematical models are established which can accurately estimate the distance between transmitter and receiver based on signal strength measurements. This is then integrated to triangulation techniques to correctly calculate the exact location of a moving robot. Through experimental validation, the approach demonstrates effectiveness in providing precise localization.

The study then demonstrated the application of various navigation algorithms for implementing in low-cost robots. Multiple algorithms are investigated and A* is demonstrated to show superior performance and effectiveness.

Overall, the study demonstrates a complete integrated solution for localization and navigation and offers a cost-efficient solution for various applications requiring autonomous positioning and navigation in indoor environments.

The WiFi-fingerprinting method involves two phases: training for generating a fingerprint radio-map and estimation for determining user location. Traditional manual methods for radio-map creation are time-consuming and labor-intensive, especially in large areas. A contemporary solution is the automated robot-based method, where a robot collects RSS data to build the radio map, reducing time and effort [1,2]. Indoor robot-based solutions have diverse applications like home care, warehouse inventory management and emergency support [3,4,5].

In our previous work [6], a robot equipped with multiple WiFi-transceiver nodes to construct an extensive dataset was demonstrated. Nodes collect Received Signal Strength Indicators (RSSI) at various heights, streaming data to an online database for localization analysis. The study focuses on factors like antenna height's impact on WiFi signal strength, aiming for a robust indoor localization system.

Utilizing collected RSSI data and multilateration techniques to accurately determine unknown positions is demonstrated in [7]. It promises a straight-forward approach for location estimation called weighted three minimum distances method (WTM) in which a set of three distances is selected from a wider set of access points. This minimal set is then used in location estimation using triangulation technique. This study uses a similar approach for estimating the location of the robot with varying accuracies depending on many parameters including cell size, grid size, number of known anchor access points and number of collected samples per position to name a few. Different filtering techniques can be used to improve the quality of the collected dataset and subsequently the accuracy of the calculated localization information. Ref. [8] demonstrates the use of extended Kalman filer to ignore any outliers in the collected data samples. The authors of [9] proposes the use of outlier detection methods for removing the effect of such disproportionately erroneous distance estimates in location estimation using the RSSI.

This study demonstrates a complete framework in which a fully autonomous low-cost robot is constructed for location estimation and navigation in indoor environments. First Section II introduces the robot along with its various subsystems, interfaces and sensors, then Section III introduces the selected AI algorithms that can be used in autonomous navigation, then Section IV discusses a localization technique based on triangulation of recorded WiFi signal strengths from a selected set of anchor access points, then Section V discusses the results of the localization and navigation experiments which were executed during this project, finally Section VI presents the conclusions and potential future work.

II. Autonomous Low-Cost Robot Construction

As part of this work, a completely autonomous robotic system was constructed to demonstrate the localization and navigation techniques proposed in the study. It would first collect all relevant WiFi-hotspot information from a

predefined set of anchor WiFi access points throughout the desired indoor space. That information is then used by the localization algorithm described in Section IV. The localization formulae developed in Section IV along with the autonomous navigation techniques described Section III are then used to develop the embedded control firmware which is then configured on the Robot's microcontroller to enable its function as an autonomous navigation system.
The details of this robot long with the range of sensors, actuators and control sub-units are discussed in the remainder of this section.

A. External Overview

An overview of the robot can be seen in Fig. 1. The robot was housed in a sturdy and compact High-Density Fibreboard (HDF). This enabled the structure to contain all sensors and actuators needed for the robot, along with all control boards, power supplies and wiring while minimizing interference with the WiFi transceivers widely used in the project. The rest of this section outlines the different subsystems that make up the robot, providing a concise description of each.

A set of four motors for controlling the movement of the robot were used for allowing the robot to move in a well-controlled manner, be able to maneuver well and rotate freely in 360-degree angles while in position.

A set of lithium batteries were used to power the motors through an H-Bridge sub-circuit. Three 3.7-volt cells were used in series to provide a collective voltage is 11.1-volt to the H-Bridge to power the motors.

An additional 5-volt power sub-system was used to power the digital part of the system which includes the Arduino control sub-system, the ESP8266 [10] SoC used for localization and all other digital sensors and actuators used in the system. A separate power supply was used in this case to control the digital part of the system to minimize the noise interference usually associated by the motors and their drivers.

The robot was quipped by set of sensors for collecting and storing environment metrics. Those included the water sensor, smoke sensor and humidity and temperature sensor set (DHT) to provide readings of the robot's surrounding conditions that could impact the system or its users. The user can access those readings either by the mobile application developed specifically for interacting with the robot or by the LCD display integrated into the robot. The environment sensors can be used as an early notification mechanism in the case of alarming conditions within the indoor building in which it resides.

An input-output subsystem comprising of a standard LCD display and keypad used for quick control of various functions of the robot from sensing, localization, and navigation.

A camera module was also integrated to enable the control of the robot from an off-site location. For this purpose, the ESP32-CAM sub-system was configured in streaming mode which was viewable by the in-house developed mobile application.

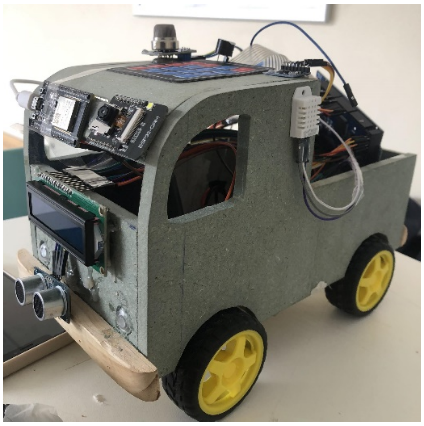

Figure 1 The complete autonomous navigating Robot

This streaming functionality enables users to visualize the path taken by the robot and potentially interfere with the decisions taken by the robot if needed. This would enable the deployment of the robot in various applications including security monitoring, first-aid responding, learning assistance amongst other applications.

A mobile application was developed from which all aspects of the robot can be observed and controlled. This provides an interface for both users and administrators. The application allows users to monitor surrounding conditions, including detecting emergencies such as fires, water spillage, alarming levels of humidity or temperatures, along with other indicators. It also served as a remote display for the camera streaming functionality. Finally, it enabled the control and observation of localization and navigation functionalities of the robot where users can track the location of the robot through the application and instruct the robot to navigate autonomously to a specific position chosen through the app.

B. Schematic Description

The schematic of the overall electronic system is depicted in Fig. 2. The main control board is an Arduino Mega sub-system. The connections from the main control board towards the motor-control sub-circuit are achieved using two main control signals: one for the left-side motors and another for the right-side ones. The motor-drive signals are treated as analog signals using pulse-width modulation (PWD), which is sufficient to drive the robot to the designated position either for data-collection (which is used in development of the positioning algorithm) or for autonomous navigation. This enables the main control board to completely control the speed and direction of movement of the robot.
The use of the Arduino Mega as a main control board was instrumental in creating a powerful and effective system with a multitude of peripherals integrated while maintaining the low-cost aspect of the system.

III. AUTONOMOUS NAVIGATION

In terms of robotics and grid-based navigation [11], the selection of algorithm can really influence the operational time and functionality of the robot mission. Two widely used methods, A* algorithm [12] and Uniform Cost Search (UCS) [13], have different routes for the consideration of the problem of the path that most optimally connects from the starting point to the goal point on a grid map.

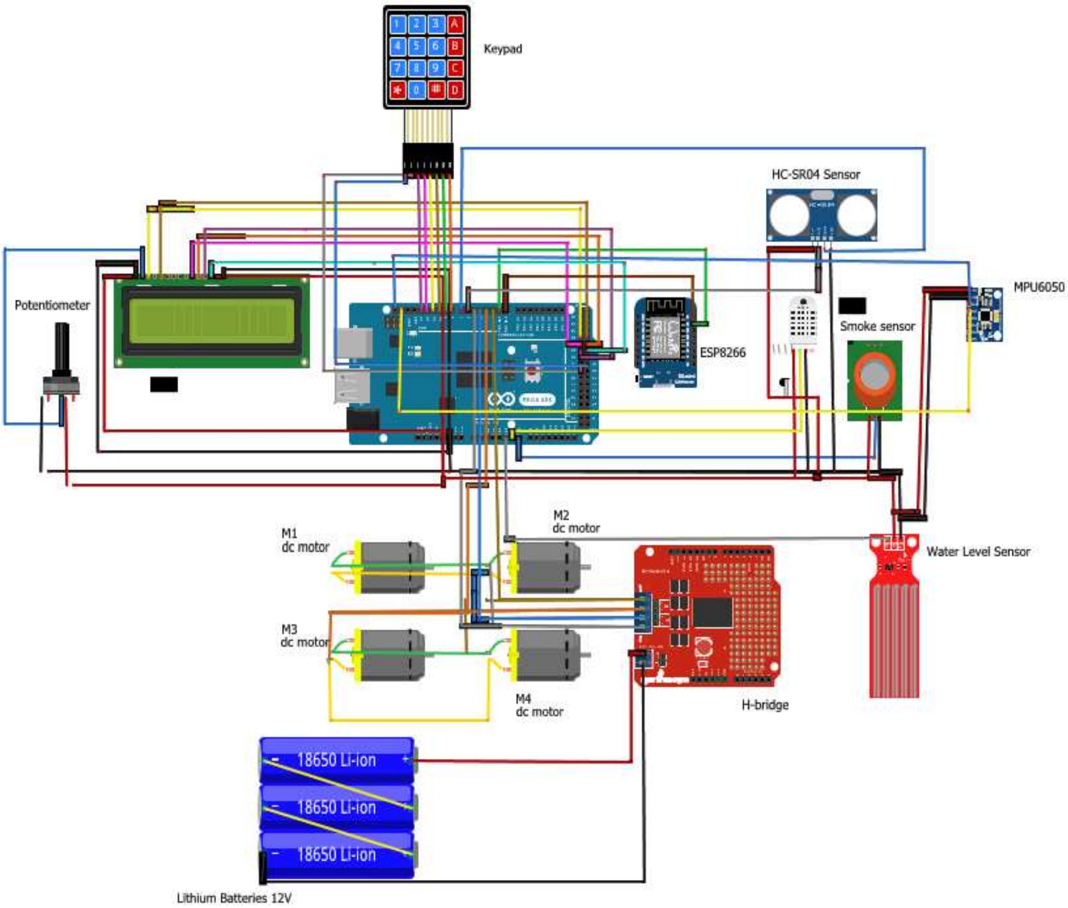

Figure 2. Robot overall circuit diagram

A. *Uniform Cost Search*

Uniform Cost Method, as the name vouches for, is the search strategy that allows exploring the search space by picking the ones with the lowest cumulative cost. The expansion model ignores the cost and information of the goal state only, increasing the number of nodes with the priority determined according to the distance from the start point.

However, this approach drives the process towards the shortest path, which often means that huge amounts of nodes are explored, thus resulting in greater computational complexity.

B. *A* Algorithm*

The A* algorithm differs from the uniform cost search by incorporating the strengths of both multi-variate search and goal-driven technology. By considering the cumulative node cost and an estimation of the node cost required to reach the desired goal, Algorithm A* strategically prioritizes node selection to efficiently achieve the goal using available resources. This heuristic approach often relies on distance measures such as Manhattan or Euclidean distances to guide movement and determine the next moves, thereby creating potential shortcuts in the search space and enhancing the overall efficiency of the underlying process.

IV. TRILATERATION-BASED LOCALIZATION

Trilateration is a fundamental technique used to determine the exact location of an object by measuring distances from three strategically placed stations, as depicted in Fig. 3. This method is widely employed in the field of robot localization, which has numerous practical applications.

The problem at hand is presented as the challenge of finding the point where three circles intersect. This requires the identification of the solutions to the resulting system of quadratic equations.

$$(x - x1)^2 + (y - y1)^2 = d1^2 \quad (1)$$
$$(x - x2)^2 + (y - y2)^2 = d2^2 \quad (2)$$
$$(x - x3)^2 + (y - y3)^2 = d3^2 \quad (3)$$

To determine the separation between the robot and the anchor access points, a suite of RSSI values is recorded and analyzed and then an estimation of the distance is made using a guided numerical regression where a formula is produced that relates a recorded RSSI value to the distance of a known access point. Those formulae are used at a later stage to enable approximate distances from the known access points to be calculated. Once these distances are calculated, trilateration according to Eqn. 1-3 is utilized to determine the approximate position of the robot within the specified grid coordinates.

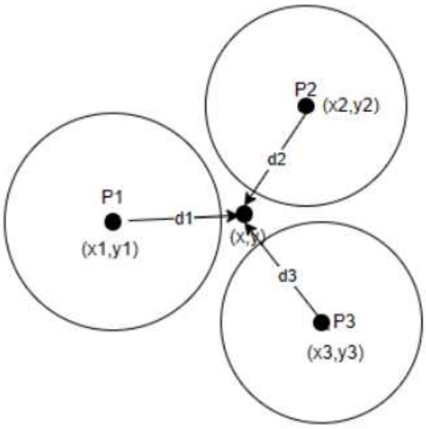

Figure 2. Trilateration-Based Localization

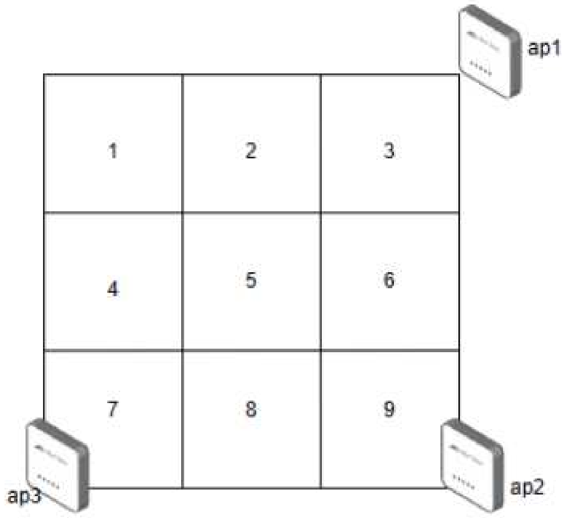

Figure 3. Grid defining the anchor nodes positions

Table 1 Collected RSSI values sample.

Points	RSSI 1	RSSI2	RSSI3	D1	D2	D3
(22,22)	53	61	65	71.47027	115.1217	159.8061
(67,22)	46	67	63	31.82766	131.3697	131.8825
(112,22)	47	49	62	31.1127	159.1006	115.317
(22,67)	60	58	63	95.46203	71.47027	131.8825
(67,67)	53	53	56	70.83784	95.46203	96.16652
(112,67)	52	54	55	70.5195	131.0267	71.7844
(22,112)	61	66	61	131.0267	31.82766	115.317
(67,112)	64	60	56	114.3372	70.83784	71.7844
(112,112)	64	59	51	114.1403	114.3372	32.52691

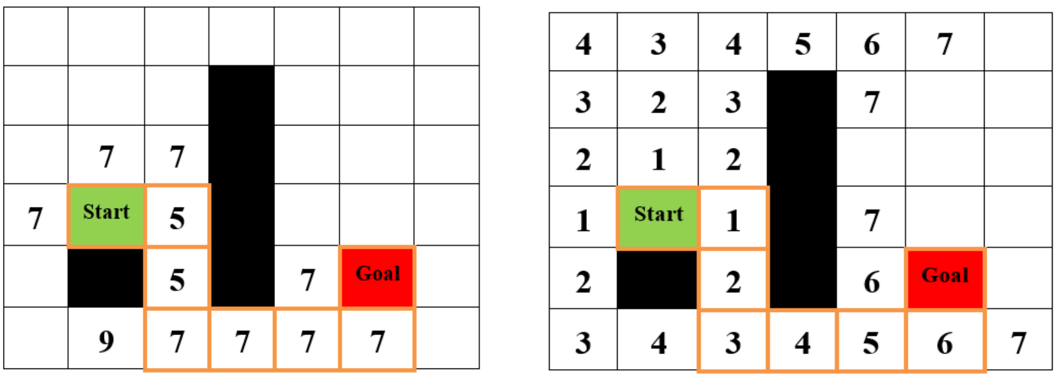

Figure 5. A comparison between A and uniform cost search in grid word (a) A* (b) UCS*

A. Automated collection of RSSI data

To begin our experiment, a set of RSSI information is collected from three access points that were set up using a set of ESP8266 sub-systems that were strategically placed at predetermined locations. Following this, the robot was positioned at a set of known locations within a grid arrangement, as shown in Fig. 4.

The scanning operation is then commenced for RSSI data, repeating the process for a duration of 20 to 30 iterations per location. The purpose of this was to ensure that the data collected at each location remained consistent.

To aid in this process, an additional ESP8266 module was utilized centrally to the robot. Each ESP8266 was assigned a unique SSID and configured to operate on a channel that did not overlap with the others in the system, thereby minimizing the likelihood of collisions occurring. Table 1 presents a sample of the gathered RSSI values.

The data obtained for every individual position was subjected to outlier detection using box plots to detect and emphasize any abnormalities. This method ensures that our calculations are not affected by data points that deviate significantly from the overall set of measurements for each location.

V. Results and Discussion

The results of the various navigation and localization techniques are discussed in this section.

A. *Assessment of Navigation Algorithms*

A decision regarding whether to apply A* algorithm or a Uniform Cost Search algorithm to a robotic platform depends on parameters like the size of the grid, complexity of the environment, and the available computational resources. In cases where the environment is a simple one and the price of displacement between subsequent cells is identical, Uniform Cost Search might bring good competition with a low computational price.

But, the more complex situations where the usual cost of movement varies or where there is a presence of barriers that cannot be avoided, A* algorithm stands apart. It enables this algorithm to narrow down the goal's heuristic information resulting to the best path with just minimal exploration of nodes. This means that in most of the actual usage, A* often shows better results than the Uniform Cost Search algorithm as regards both pathfinding speed and resource utilization.

In the end, in grid navigation of robots, both A* algorithm and Uniform Cost Search represent tool cases that are able to fulfil very specific tasks based on the requirements and constraints of the given case. In any case, A* algorithm stands high chances to outperform in cases where execution time and fault tolerance count the most, thanks to using heuristic-driven mechanism to find its way through complicated settings. Fig. 5 presents a comparative analysis between the two algorithms.

As shown in the figure, a comparative analysis has been conducted between two algorithms in a grid world environment. Both algorithms identified the best route, but A* tended to investigate fewer areas seeking the optimal path. Notably, the numbers in each square relate to how each algorithm works: A* depends on the estimated cost of the cheapest solution through node n (f(n)), while UCS relies on travel costs so far (g(n)).

B. *Evaluation of the Localization Algorithms*

After analyzing logarithmic and linear regression methods to calculate the distances between the robot and the access points, our objective was to identify the more effective approach. Upon analyzing the RSSI data from the anchor access points and the expected distances, the specific regression equations for each access point are numerically solved as demonstrated in Fig. 6 to 8.

Using these formulae, the distance between every access point and the robot could be calculated at any position. Table 2 illustrates the distances calculated through logarithmic regression. Table 3 illustrates the distances calculated through linear regression. Table 4 provides the estimated locations based on the distance estimations obtained from both regression methods after applying the triangulation technique. The calculations using either regression showed similar results. However, when compared to the actual distance as shown in Table 1, there is some error, and that is due to the randomness that can occur in the RSSI values and environmental elements that could affect the values like reflections, obstacles, and moving objects within vicinity.

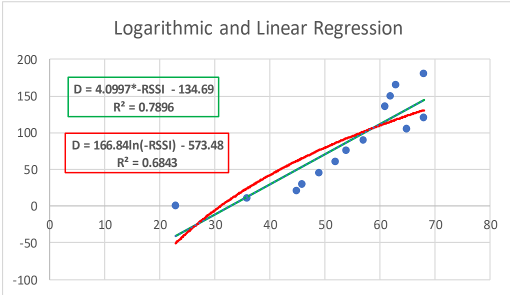

Figure 6. Logarithmic and Linear Regression for distance estimation between Robot and access point one.

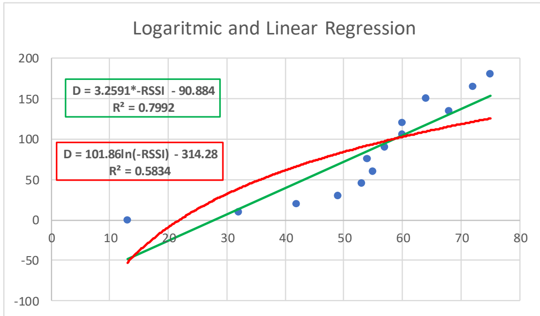

Figure 7. Logarithmic and Linear Regression for distance estimation between Robot and access point two.

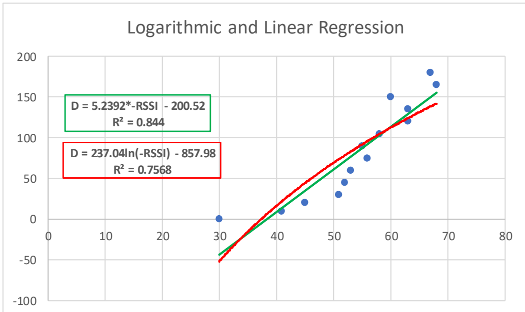

Figure 8. Logarithmic and Linear Regression for distance estimation between Robot and access point three.

Table 2 Calculated distances using logarithmic regression.

Points	Calculated D1	Calculated D2	Calculated D3
(22,22)	88.93465532	104.4407568	131.5208677
(67,22)	65.30145642	113.9969194	124.1127273
(112,22)	68.88958611	82.12846438	120.3199872
(22,67)	109.6317977	99.30398604	124.1127273
(67,67)	88.93465532	90.12140264	96.19329519
(112,67)	85.756624	92.02534032	91.92216703
(22,112)	112.3895729	112.4651958	116.4655738
(67,112)	120.3995078	102.7571201	96.19329519
(112,112)	120.3995078	101.0451857	74.02376617

Table 3 Calculated distances using linear regression.

Points	Calculated D1	Calculated D2	Calculated D3
(22,22)	82.6	107.919	140.035
(67,22)	53.9	127.473	129.557
(112,22)	58	68.811	124.318
(22,67)	111.3	98.142	129.557
(67,67)	82.6	81.847	92.884
(112,67)	78.5	85.106	87.645
(22,112)	115.4	124.214	119.079
(67,112)	127.7	104.66	92.884
(112,112)	127.7	101.401	66.689

Table 4 Calculated location using the distance estimation from both regressions.

Cell	Points	Calculated points using log regression	Cell using log regression	Calculated points using linear regression	Cell using linear regression
1	(22,22)	(43.8, 55.6)	4	(38.01, 44.97)	1
2	(67,22)	(58.6, 44.2)	2	(65.51, 31.8)	2
4	(22,67)	(46.98, 76.8)	5	(41.01, 75.04)	4
5	(67,67)	(63.3, 78.9)	5	(60.4, 78.2)	5
8	(67,112)	(72.3, 100.3)	8	(76.1, 108.07)	8
9	(112,112)	(85.02, 110.05)	8	(89.1, 119.2)	8

VI. CONCLUSION

The study has provided valuable insights into the utilization of WiFi signal strength for localization and navigation in low-cost robotic systems operating in indoor environments. Through the utilization of mathematical models and the integration of triangulation techniques, we have successfully facilitated the dynamic positioning of robots. Our localization capabilities have been validated through experimental verification, showcasing a good accuracy and precision of our approach.

Moreover, our evaluation of uniform cost search and A* navigation algorithms, notably A*, has shed light on their effectiveness in guiding robots through indoor spaces.

In addition to the findings reported, one exciting path for future research is the use of neural network models to improve triangulation procedures. By using the power of artificial intelligence, we have the ability to improve the accuracy and efficiency of our localization systems. Incorporating neural networks into the triangulation process may allow the system to learn and adapt to changing environmental conditions, resulting in more robust and consistent localization outcomes. This approach has the potential to advance the capabilities of low-cost robotic systems in indoor navigation.

VII. REFERENCES

[1] Yan, H.; Peng, T.; Liu, H.; Ding, Y. Indoor Position Method of Industrial Robot Based on WiFi Fingerprint Position Technology. In Proceedings of the 2019 1st International Conference on Industrial Artificial Intelligence (IAI), Shenyang, 22–26 July 2019; pp. 1–6.

[2] Luo, R.; Hsiao, T. Dynamic Wireless Indoor Localization Incorporating With an Autonomous Mobile Robot Based on an Adaptive Signal Model Fingerprinting Approach. *IEEE Trans. Ind. Electron.* 2019, *66*, 1940–1951. https://doi.org/10.1109/tie.2018.2833021.

[3] Li, M.; Zhao, L.; Tan, D.; Tong, X. BLE fingerprint indoor localization algorithm based on eight-neighborhood template matching. *Sensors* 2019, *19*, 4859.

[4] González, J.L.S.; Morillo, L.M.S.; Álvarez-García, J.A.; Ros, F.E.D.S.; Ruiz, A.R.J. Energy-efficient indoor localization WiFi-fingerprint system: An experimental study. *IEEE Access* 2019, *7*, 162664–162682.

[5] Morar, A.; Moldoveanu, A.; Mocanu, I.; Moldoveanu, F.; Radoi, I.E.; Asavei, V.; Gradinaru, A.; Butean, A. A comprehensive survey of indoor localization methods based on computer vision. *Sensors* 2020, *20*, 2641.

[6] Abu Kharmeh S.; Natsheh E.; Sulaiman B.; Abuabiah M.; Tarapiah S.; Indoor WiFi-Beacon Dataset Construction Using Autonomous Low-Cost Robot for 3D Location Estimation. Applied Sciences. 2023; 13(11):6768. https://doi.org/10.3390/app13116768

[7] Du, J.; Yuan, C.; Yue, M.; Ma, T. A Novel Localization Algorithm Based on RSSI and Multilateration for Indoor Environments. Electronics 2022, 11, 289. doi: 10.3390/electronics11020289

[8] Ben Kilani M.; Raymond A. J.; Gagnon F.; Gagnon G.; Lavoie P., RSSI-based indoor tracking using the extended Kalman filter and circularly polarized antennas, 2014 11th Workshop on Positioning, Navigation and Communication (WPNC), Dresden, Germany, 2014, pp. 1-6, doi: 10.1109/WPNC.2014.6843305

[9] Chuku, N.; Nasipuri, A. RSSI-Based Localization Schemes for Wireless Sensor Networks Using Outlier Detection. J. Sens. Actuator Netw. 2021, 10, 10. doi: 10.3390/jsan10010010

[10] Espressif Systems, ESP8266 WiFi module, https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

[11] Russell, S.J.1.; Norvig, P.; Davis, E. (2010). Artificial intelligence: a modern approach. 3rd ed. Upper Saddle River, NJ, Prentice Hall.

[12] Xiang, L.; Daoxiong, G. A comparative study of A-star algorithms for search and rescue in perfect maze," 2011 International Conference on Electric Information and Control Engineering, Wuhan, 2011, pp. 24-27

[13] Mansour, A.G.; Eid, M.S.; Elbeltagi, E.E. Construction facilities location selection for urban linear infrastructure maintenance projects using uniform cost search method. Soft Comput. 2022, 26, 1403–1415