

Development of FTM Automatic Measurement System for Indoor Location

Takashi Ookawahara, Yudai Sunagozaka, Rei Hirata,
Atushi Koizumi, Hiroshi Oguma
Department of Control Information System Engineering
NIT, Toyama College
1-2 Ebie-Neriya, Imizu, Toyama, 933-0239 Japan
oguma@nc-toyama.ac.jp

Koki Nakamura, Takenori Sumi, Yukimasa Nagai
Information Technology R&D Center
Mitsubishi Electric Corporation
5-1-1 Ofuna, Kamakura, Kanagawa, 247-8501 Japan
Nakamura.Koki@ds.MitsubishiElectric.co.jp

Abstract—We conduct measurement experiments using the IEEE802.11-2016 standard with the goal of establishing an indoor location estimation method using Wi-Fi access points (APs). This standard employs the Fine Time Measurement (FTM) protocol to calculate the distance between an AP and a terminal. To measure indoor ranging accuracy, measurements need to be taken at many points. In this study, we create a measurement robot using line tracing and QR codes, and an Android application that automatically collects FTM data, and built an automatic measurement system. This system enables automatic collection of distance measurement data from many points, shortening the measurement time and reducing human cost and human error. Furthermore, to evaluate the usefulness of the proposed system in comparative experiments, we also investigated the impact of AP height on the positioning accuracy.

Keywords—Indoor Location, Fine Timing Measurement, IEEE802.11-2016, Wireless LAN

I. INTRODUCTION

We are engaged in the development of an indoor positioning method using Wi-Fi APs [1]. One of the protocols used to measure the distance between a Wi-Fi AP and a client device is defined in IEEE 802.11-2016 (IEEE 802.11mc) [2], which adopts the Fine Timing Measurement (FTM) protocol. Using this standard, we are conducting research to estimate the location of a device based on distance measurements to multiple APs [3]. In previous studies, evaluation experiments were conducted by manually moving and placing the device at each measurement location. To expand the measurement area, it was necessary to perform measurements at a large number of points. Moreover, to examine the effects of changes in the measurement environment or AP placement, as well as to ensure experimental reproducibility, multiple experimental patterns and repeated measurements were required. As a result, the temporal cost and human labor involved in data collection became enormous, posing a significant challenge to the progress of research. To address this issue, we aim to build an automated FTM measurement system by developing a line-tracing robot with QR code detection and an Android™ application.

II. METHOD

A. System Overview

The automated measurement robot is equipped with a line-tracing sensor and a QR code reader. A predefined measurement path is created using black tape affixed to the floor, and QR codes are placed at designated measurement points. Each QR code encodes the name of the corresponding measurement location as a character string. While detecting and following the black line, the robot scans the QR codes at

each measurement point. Upon successful detection, it reads the embedded information and stops at the location. An overview of the proposed FTM automated measurement robot and the overall system configuration is illustrated in Fig. 1.

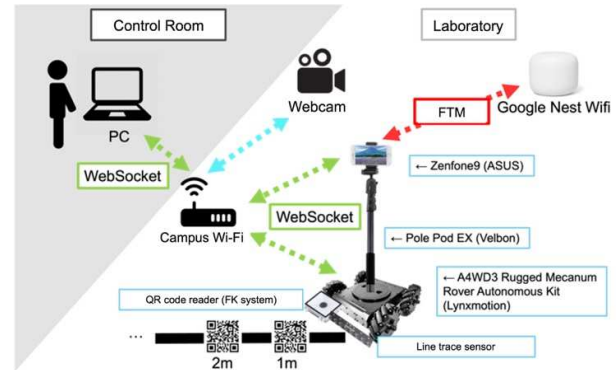


Fig. 1. Overview of the proposed FTM-based automated measurement system.

A monopod stand is mounted on the robot to hold a smartphone, allowing for adjustable measurement heights. The line-tracing sensor is attached to the front of the robot chassis, while the QR code scanner is mounted on the underside. For FTM-based ranging, a Zenfone 9 smartphone is used as the Initiator (mobile device), and a Google Nest Wi-Fi unit supporting IEEE 802.11mc is used as the Responder (APs). The components used for constructing the robot are listed in Table I.

TABLE I. COMPONENTS USED

Product Name	Model Number	Manufacturer
A4WD3 Rugged Mecanum Rover Autonomous Kit	RB-Lyn-1135	Lynxmotion
Raspberry Pi 4 Model B	SC0194	Raspberry Pi Ltd
High-Precision AD/DA Module for Raspberry Pi	WAVESHARE-11010	WAVESHARE
Line-Tracing Sensor Module	AE-NJL5901AR-8CH	Akizuki Denshi Tsusho Co., Ltd.
Embedded 2D Code Reader	F820GS-U	FKsystem

A Raspberry Pi 4 is mounted on the robot to control the system by acquiring voltage values from the line-trace sensors, reading QR code information, and performing bidirectional communication with a smartphone. An overview of the communication system is shown in Fig. 2.

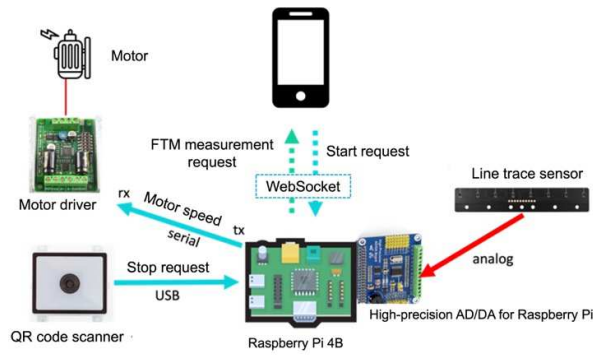


Fig. 2. Overview of the communication system.

The voltage values from the line-trace sensors are read using a high-precision AD/DA converter for Raspberry Pi, and the data is transmitted via SPI (Serial Peripheral Interface) communication. The QR code scanner is connected via USB and recognized as a USB HID (Human Interface Device). The motor driver receives motor rotation speed commands through serial communication. For bidirectional communication with the smartphone, wireless communication is performed using socket communication over Wi-Fi. A system flowchart of the smartphone and robot is shown in Fig. 3.

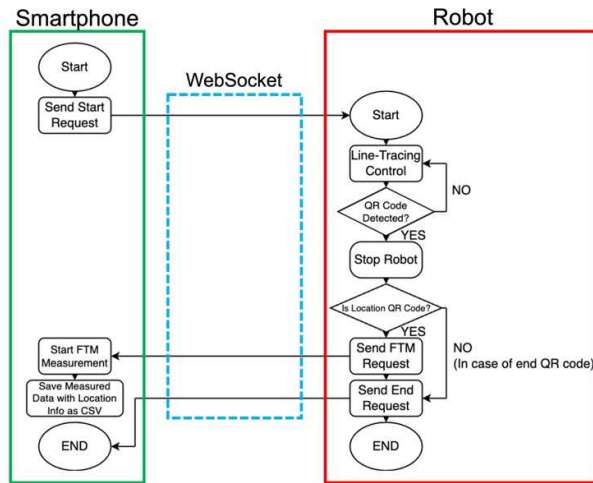


Fig. 3. System flowchart of the smartphone and robot.

The smartphone acts as a client, while the robot functions as a server. A WebSocket is used to send a line-tracing request from the smartphone to the robot. Upon receiving the request, the robot performs line tracing until it detects a QR code, at which point the robot stops. If the QR code contains location information for measurement, the robot sends the location information and a request for FTM to the smartphone. Upon receiving the FTM request, the smartphone performs FTM measurements for a fixed duration. The measured data is saved as a CSV file named with the location information obtained from the QR code. The smartphone then sends another line-tracing request to the robot. This process is repeated until the QR code indicates a termination point, at which time the robot stops.

B. Overview of the Robot

The robot platform is based on the A4WD3 Rugged Mecanum Rover Autonomous Kit by Lynemotom. Figs. 4–6 show the internal structure, bottom view, and full view of the robot, respectively.

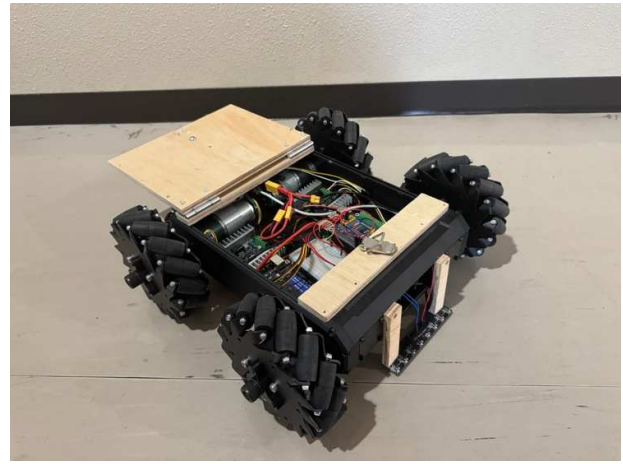


Fig. 4. Internal structure of the robot.

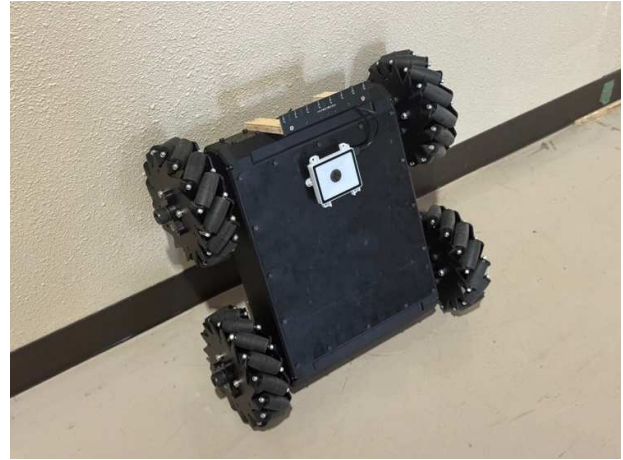


Fig. 5. Bottom view of the robot.



Fig. 6. Full view of the robot.

To facilitate battery replacement and maintenance, a custom top cover was made by processing a wooden board. The top section of the robot can be opened and closed, and a monopod is mounted to secure a smartphone. A line-trace sensor is attached to the front of the robot, and it is mounted at a height of less than 5 mm from the ground so that it can accurately detect black lines. A QR code reader is installed on the underside of the robot to scan QR codes.

C. FTM Automatic Measurement Android Application

The user interface (UI) of the FTM automatic measurement application on the smartphone is shown in Fig. 7.



Fig. 7. User interface of the FTM automatic measurement application.

The left screen shows the standby mode, the center shows the measuring mode, and the right shows the moving mode. The screen transitions according to the system state. The interface includes a button for scanning available APs, input forms for setting the measurement duration and interval at each location, and a connection button to communicate with the robot. When the AP scan button is pressed, the application scans for APs that support IEEE 802.11mc and registers them for use [4][5]. Pressing the connection button establishes a WebSocket connection with the robot. Upon receiving a measurement request from the robot, the application performs distance measurements at specified intervals for the designated duration.

D. Robot Monitoring and Control Application

To automate FTM measurements, the system is designed to allow monitoring of the robot's status without requiring a person to be present at the measurement location. An online camera is installed in the laboratory, enabling remote observation of the robot at the measurement site. A snapshot of the online camera footage is shown in Fig. 8.

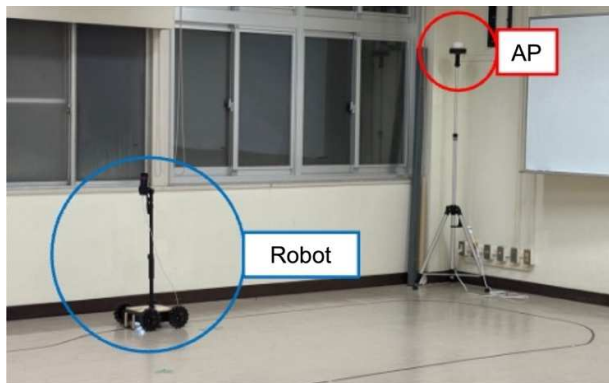


Fig. 8. Online camera view of the robot in the laboratory.

Since a WebSocket server runs on the robot's Raspberry Pi 4, the robot's status can be monitored through a web application by accessing the Raspberry Pi's IP address and

server port via a browser. The developed robot monitoring web application is shown in Fig. 9.

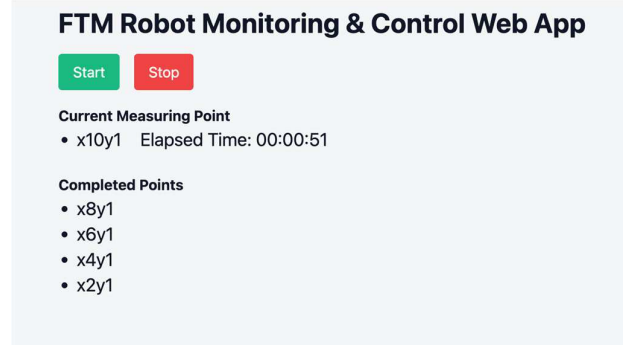


Fig. 9. Web application for monitoring and controlling the robot.

By clicking the “Start” button, the robot can be remotely instructed to begin line tracing, and pressing the “Stop” button halts the robot. The application also displays the current measurement location and the elapsed time since the start of the measurement in real time. Once a measurement is completed, the location is added to a list of completed measurement points. Thus, the system enables both remote control and monitoring of the robot.

III. AUTOMATIC DATA COLLECTION USING THE ROBOT

In this experiment, we verify whether the developed robot can autonomously collect ranging data at multiple locations. Traditionally, it has been difficult to experimentally evaluate the impact of environmental changes on positioning accuracy through manual measurements. This study also investigates whether the proposed system is effective for conducting such comparative experiments. Specifically, we vary the height of the APs to analyze how this environmental change affects positioning accuracy. Based on the collected data, we estimate the device's location and evaluate the estimation accuracy at each measurement position and under each environmental condition.

A. Experimental Method

The experimental environment is shown in Fig. 10. The experiment was conducted in the Monozukuri Laboratory on the first floor of Building 1 at the National Institute of Technology, Toyama College. A Zenfone 9 was used as the device, and four Google Nest Wi-Fi APs were installed at the four corners of the laboratory. The laboratory floor was divided into a 2-meter grid, and measurements were taken at a total of 32 locations. The experimental conditions are summarized in Table II. FTM-based ranging was performed repeatedly at 400 ms intervals, with each location measured for one minute. The height of the device was fixed at 1 meter, while the AP height was varied under three conditions: 0 m, 1 m, and 2 m.

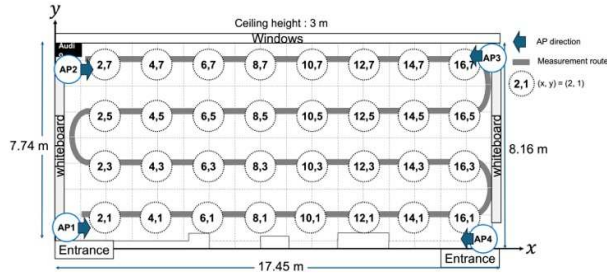


Fig. 10. Experimental environment.

TABLE II. EXPERIMENTAL CONDITIONS

Item	Condition
Measurement duration	1 minute per point
Measurement interval	400 ms
AP installation height	0 m, 1 m, 2 m
Device height	1 m

B. Position Estimation Method

The estimated coordinates of the device are then calculated by finding the centroid of the intersection point of the three lines.

In FTM measurement, the straight-line distances from the device to each of the four APs are obtained. Using this ranging data, the position of the device can be estimated geometrically [6]. Let the coordinates of the n -th AP be (x_n, y_n) , and the distance from the AP to the device be d_n . Then, the following four circle equations centered at each AP with radius d_n can be written:

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = d_1^2 \\ (x - x_2)^2 + (y - y_2)^2 = d_2^2 \\ (x - x_3)^2 + (y - y_3)^2 = d_3^2 \\ (x - x_4)^2 + (y - y_4)^2 = d_4^2 \end{cases} \quad (1)$$

Subtracting the first equation from each of the remaining three yields a system of three linear equations:

$$\begin{cases} 2(x_2 - x_1)x + 2(y_2 - y_1)y = d_1^2 - d_2^2 - x_1^2 - y_1^2 + x_2^2 + y_2^2 \\ 2(x_3 - x_1)x + 2(y_3 - y_1)y = d_1^2 - d_3^2 - x_1^2 - y_1^2 + x_3^2 + y_3^2 \\ 2(x_4 - x_1)x + 2(y_4 - y_1)y = d_1^2 - d_4^2 - x_1^2 - y_1^2 + x_4^2 + y_4^2 \end{cases} \quad (2)$$

The estimated coordinates of the device are then calculated by finding the centroid of the intersection point of the three lines.

C. Results

Using the robot, all measurement points were successfully measured without requiring a human to manually move the device. Since each of the 32 locations was measured for one minute, including travel time between points, the entire measurement process was completed in approximately 35 minutes. Based on the collected data, the Euclidean distance errors between the measured and estimated positions were calculated for each location. Figs. 11–13 show the cumulative distribution functions (CDF) of the distance errors for each AP height condition. In addition,

Figs. 14–16 present heatmaps of the percentage of measurement data with distance errors within 2 meters.

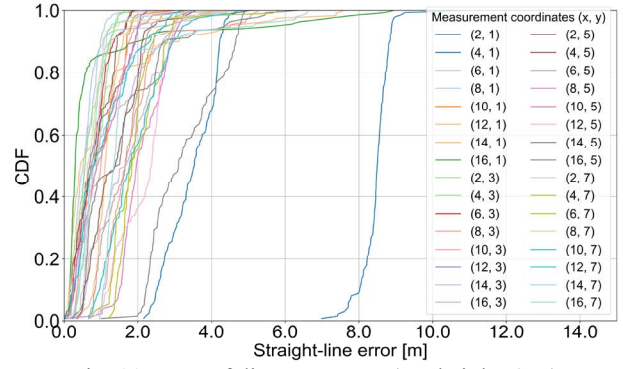


Fig. 11. CDF of distance errors (AP height: 0 m)

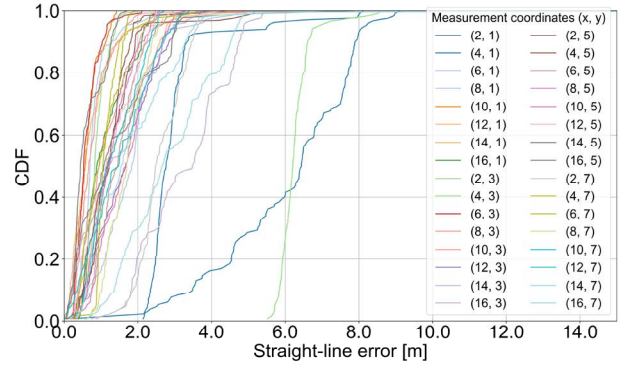


Fig. 12. CDF of distance errors (AP height: 1 m)

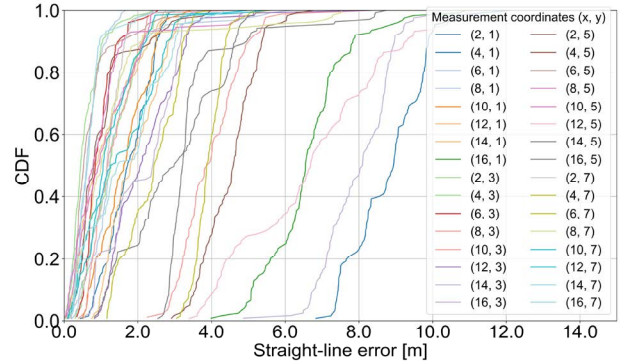


Fig. 13. CDF of distance errors (AP height: 2 m)

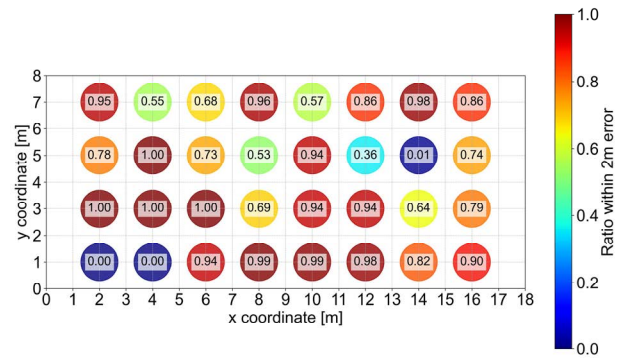


Fig. 14. Heatmap of 2 m error rate (AP height: 0 m)

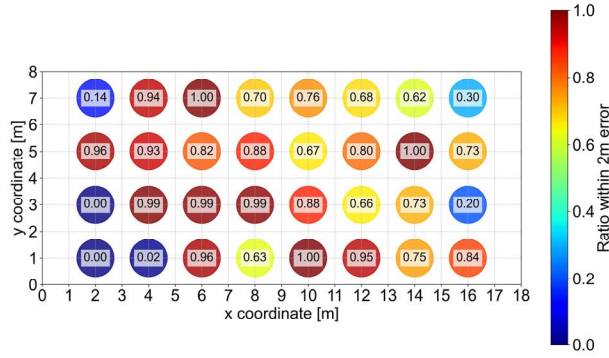


Fig. 15. Heatmap of 2 m error rate (AP height: 1 m)

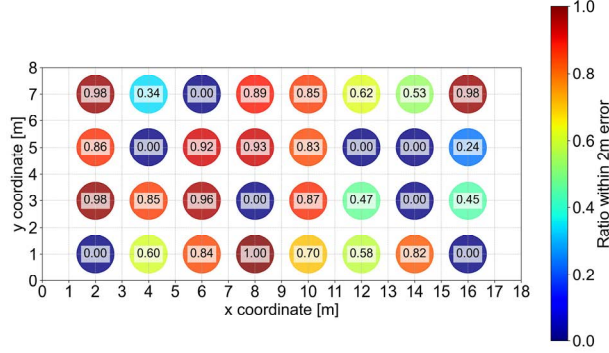


Fig. 16. Heatmap of 2 m error rate (AP height: 2 m)

From Figs. 11–13, it is observed that when the AP height is set to 0 m, over 80% of the measurement points exhibit positioning errors within 2 meters. Moreover, the results indicate that as the AP height decreases, the positioning accuracy improves. While the minimum errors at many locations remain within 2 meters for the 0 m AP height condition, an increase in points with errors exceeding 2 meters is evident when the AP height is 2 m. This may be due to the fact that the APs used in this experiment were Google Nest Wi-Fi devices, which are designed for home environments and optimized for low mounting positions.

From Figs. 14–16, it is evident that more than 70% of the measurement points in the central area of the laboratory achieved errors within 2 meters. However, in areas close to the APs, errors tended to increase, ranging from 2 to 8 meters, indicating a decline in estimation accuracy. In particular, point (2, 1) showed a significant spread in the estimated positions. This is likely due to a Non-Line-of-Sight (NLOS) condition between point (2, 1) and AP4. The spatial relationship is illustrated in Fig. 17.

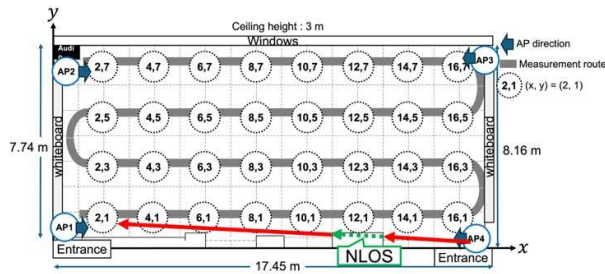


Fig. 17. Spatial relationship between point (2, 1) and AP4.

The laboratory has irregular wall structures, and AP4 is not visible from point (2, 1), resulting in an NLOS condition. The FTM ranging data obtained from each AP at this point is summarized in Table III.

TABLE III. FTM RANGING DATA FOR EACH AP AT POINT (2, 1)

AP	FTM Ranging [m]	True Distance [m]	Error [m]
AP1	1.26	1.84	0.58
AP2	5.55	6.24	0.69
AP3	16.11	16.49	0.38
AP4	17.55	14.93	2.62

The FTM ranging errors for AP1 to AP3 are within 1 meter of the ground truth. However, the ranging result for AP4 is 2.62 meters greater than the true distance. This discrepancy is likely due to the NLOS condition, where FTM ranging tends to overestimate the distance. As a result, an error occurred between the measured and estimated coordinates when calculating the centroid.

IV. CONCLUSION

In this study, we developed an automated FTM measurement system by developing a line-tracing robot with QR code detection and an Android™ application, aiming to automate FTM measurements at multiple locations. The experimental results demonstrated that the system successfully collected FTM data at multiple points without requiring manual movement of the device. Furthermore, the system facilitated data collection for evaluating the impact of environmental changes and AP placement on positioning accuracy, which had previously been difficult to perform manually. In the present experiment, the proposed system was employed for comparative evaluations under varying environmental conditions. The results revealed that changes in AP height affected positioning accuracy, and that the accuracy degraded at locations where APs were NLOS. Since the system eliminates the need for manual device relocation and minimizes human error, it proves to be highly effective for conducting comparative experiments and repeated trials aimed at validating reproducibility.

REFERENCES

- [1] R. Hirata et al., “FTM Implementation and Ranging Performance Characterization for Indoor Localization”, IEEE Global Conference on Consumer Electronics, POS3B, Oct. 2024
- [2] IEEE 802.11-2016, IEEE-SA, Dec. 2016.
- [3] Y. Sunagozaka et al., “Comparison of FTM Ranging Accuracy in Dual- / Single-Band Operation”, IEICE Society Conference, B-17-02, Mar. 2025.
- [4] Android Open Source Project, “Wi-Fi RTT”, <https://source.android.com/docs/core/connect/wifi-rtt?hl=ja>, Accessed May 2024.
- [5] Google Developers, “Wi-Fi RTT”, <https://developer.android.com/develop/connectivity/wifi/wifi-rtt?hl=ja>, Accessed May 2024.
- [6] “How to get one-meter location-accuracy from Android devices (Google I/O '18).” YouTube, <https://youtu.be/vywGgSrGODU>, Accessed May 2024.