ORB Visual and WiFi Online RSSI fusion SLAM

YI-HSIEN LU[1], CHIA-CHIHUANG[2], CHIH-CHUNG CHOU[2], CHENG-FU CHOU[1,2]
[1]Department of Graduate Institute of Network and Multimedia,
National Taiwan University,
TAIWAN

[2]Department of Computer Science and Information Engineering,
National Taiwan University,
TAIWAN

Abstract: - Simultaneous Localization and Mapping (SLAM) technologies are indispensable for indoor service robots, enabling them to navigate through and interact with environments. Visual SLAM systems often encounter significant challenges such as dynamic obstacles, variable lighting, feature scarcity, and perceptual aliasing in real-world scenarios. By merging the precise environmental mapping capabilities of visual SLAM with the ubiquity and stability of WiFi signals, our method effectively addresses the limitations typically associated with visual SLAM. Notably, our fusion technique leverages existing WiFi infrastructure, thus providing a cost-effective improvement in spatial awareness without the extensive offline database requirements of WiFi RSSI-based localization. Comparative performance evaluations highlight that our graph optimization-based approach not only surpasses the original ORBSLAM3 method but also significantly outperforms the Extended Kalman Filter (EKF) in terms of accuracy, particularly in environments characterized by poor lighting, feature-less scenes, and significant occlusions. This is evidenced by a reduced Root Mean Square Error (RMSE) in localization: 3.09m for our method versus 4.02m for EKF. This enhancement in precision underscores the potential of our integrated system to advance indoor navigation technologies, making it a crucial development in the field of robotics and automated systems.

Key-Words: - Real-Time, Visual SLAM, WiFi Localization, Robotic Navigation, Spatial Awareness,Sensor Fusion.

Received: August 18, 2023. Revised: May 27, 2024. Accepted: July 13, 2024. Published: September 3, 2024.

1 Introduction

Visual SLAM (Simultaneous Localization and Mapping) is a popular solution for indoor robot localization by feature point extraction and matching to positioning and mapping. Due to its high accuracy, lightweight, low cost, and low power consumption shown in Figure 1.

Compare some previous work.ORBSLAM3, [1], one of the Visual SLAM SoTA methods,which have short-term, mid-term, and long-term data association by ORB descriptor model and adjusted method to feature extraction and feature matching. The ORBslam3 uses quite an efficient and precise way to Visual, but still can't overcome the problem when a robot or device goes to featureless environment would lose track and cause low accuracy.YOLO-SLAM, a kind of improved SoTA Visual SLAM integrated with semantic information supported by deep learning models, it can help robots better perceive their surrounding environment. However, the accuracy of the estimated position is largely dependent on feature correspondences and can be adversely affected by occlusion caused by dynamic objects, featureless scenes, drastic viewpoint changes, and changes in illumination, leading to incorrect estimations due to false tracking correspondences, [2].

In our paper, Figure 16, Figure 18 and Figure 20 illustrate different challenge trajectories. These figures show that without the WiFi, [3] submodule and algorithm added to ORB-SLAM3, the system loses track and its accuracy decreases.

Additionally, loop closure detection is a crucial component of the SLAM system for the relocalization of a robot in a map. Perceptual aliasing, especially in symmetric and repetitive environments such as indoor corridors with similar patterns of doors and lights, can lead to false loops and inaccurate map estimations. Our proposed method can avoid false loop detection by using WiFi RSSI, [3] value outliers, making the system more robust. Figure 19 shows that using ORB-SLAM3 without the WiFi submodule may cause false loop detection, but with our WiFi submodule, false loop detection can be avoided, resulting in a more accurate and robust system.

Moreover, wireless signal-based indoor localization has become increasingly popular in recent years as a reliable method for identifying the locations of IoT (Internet of Things) devices in indoor environments, where GNSS (Global Navigation Satellite Systems) are typically unavailable due to the lack of a direct line-of-sight. This has motivated various research efforts to develop effective techniques for this type of localization.

However, Wireless-signal-based indoor localization [3], approaches have been able to achieve acceptable accuracy, these methods are not compatible with the need for centimeter-level precision. Additionally, such strategies normally require a predefined WiFi radio map which must be maintained and updated regularly, making them incompatible with the idea of SLAM, where a robot can be placed in an unknown environment without prior knowledge.

We present a robust life-long SLAM system that utilizes ORB-SLAM3, [1] as its base Visual SLAM module. This system consists of a Visual SLAM and a WiFi SLAM module, allowing it to address challenges with vision-based localization and navigation. These two modules interactively update both the vision map and WiFi map, with the WiFi SLAM module, [3] consisting of a tracking submodule to locate the robot when vision has difficulty, as well as a mapping submodule that autonomously updates with assistance from Visual SLAM. The advantage of our system over database-based Wireless signal based offline indoor localization methods is its ability to adapt to changing environments. Furthermore, WiFi information is used in the loop detection submodule of Visual SLAM to prevent false loop detection, since WiFi signals are different in two separate places with similar vision scenes. On top of that, a real-time degeneracy detection module is used to detect whenever the vision sensor is degraded, which introduces a mechanism to decide whether to compensate the degradation with WiFi signal information. Our system enables the combination of Visual SLAM and WiFi SLAM to provide reliable and accurate robot localization in dynamic indoor environments. With this, robots can be deployed in unknown environments without prior knowledge, and accurately localize and map areas in real-time.

2 Background

2.1 Visual SLAM

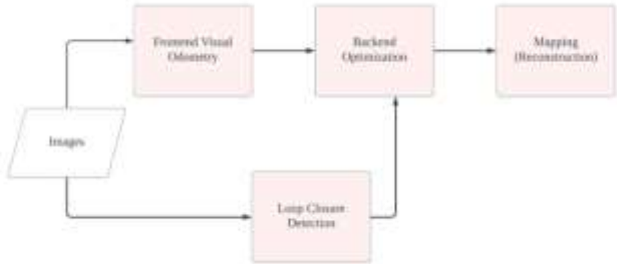

Fig. 1: Typical visual SLAM framework, [1]

With the advantages of sensor configuration simplicity, lightweight, and low cost, visual-based SLAM algorithms are proposed in research. A typical visual SLAM framework consists of frontend visual odometry, backend optimization, loop closure detection, and mapping modules as shown in Figure 2. The frontend visual odometry estimates the motion between input images from sensors and constructs a local map using a feature-based method or direct method. Backend optimization then optimizes the results from visual odometry. Simultaneously, the mapping module constructs and maintains a global map based on the measurements. To combat accumulated error, loop closure detection recognizes previously visited places, relocalizes, and improves mapping accuracy by reducing accumulated drift caused by noise.

ORB-SLAM3 is one of the well-known keyframe-based real-time visual SLAM algorithms, [1] which consists of three main threads: tracking, local mapping, and loop closing. ORB (Oriented FAST and Rotated BRIEF) features are used in this system, which is then transformed into map points after the corresponding frame is selected as a keyframe to construct the map. The tracking thread tracks for unmapped regions using ORB features extracted from images and matches ORB features to map points to perform local bundle adjustment in local mapping thread. In our system, we use ORB-SLAM3 with RGBD cameras as our visual SLAM module and sensors to demonstrate the challenges of visual SLAM and how WiFi signals, [3] can improve them.

2.2 WiFi-based Indoor Localization

The lack of availability of GNSS in indoor environments has led to an increase in demand for indoor localization solutions. One popular solution is based on WiFi fingerprinting, which utilizes the existing infrastructure of WiFi networks. This method has attracted attention from both academia and industry as it is achievable and cost-effective.

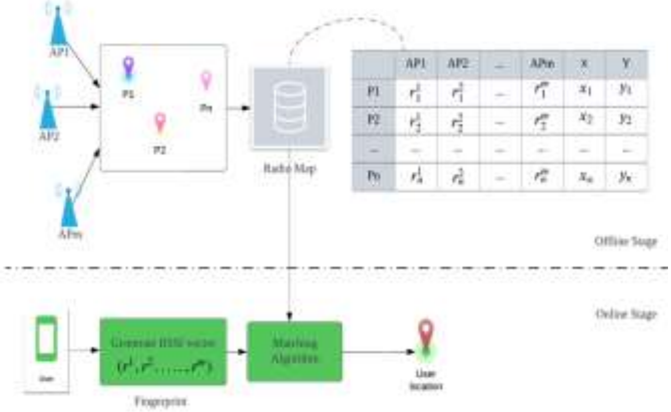

Fig. 2: Typical WiFi-based indoor localization pipeline

A typical WiFi-based indoor localization pipeline is shown in Figure 2. It consists of an offline stage and an online stage. Firstly, a radio map $\Omega \in R^{N \times M}$ construction is done in the offline stage, where N is the number of fingerprints, and M represents the number of access points plus 2 (X and Y to represent locations). A fingerprint is a vector v ∈ RM of RSSI r_m received in a place n with coordinates (x_n, y_n).Secondly, in the online stage, the user's location is estimated by matching the fingerprint of the current place to those on the radio map. Traditional matching algorithms such as K-Nearest Neighbors, Decision Tree, Random Forest, [4], and Support Vector Machine classifiers, [5] have been explored for years. WKNN can be applied to WiFi localization by using the signal strength (RSSI) values from nearby access points as features. Given a set of RSSI measurements from multiple access points, WKNN can determine the k nearest neighbors (based on signal strength similarity) to the query point (the device for which localization is required). Random Forest can also be utilized for WiFi localization. During inference, the trained Random Forest model can predict the location of a device based on its WiFi signal strengths. Generally, machine learning-based solutions achieve higher accuracy than traditional methods, but they can be expensive because training and tuning are required, and as the scale of the model increases, more computational resources are needed. Additionally, data-driven approaches depend heavily on the distribution of training data, so a natural trade-off between accuracy and robustness needs to be considered. Both traditional WiFi fingerprint-based indoor localization and machine learning-based solutions require an offline database, which does not align with the scenarios in a SLAM system, where a robot explores and locates itself without prior knowledge. Therefore, our system proposes a WiFi SLAM solution that can operate without an offline database.

2.3 Visual SLAM with WiFi

Due to the unique advantages and disadvantages of camera and WiFi sensors, several methods, [3] have been proposed to combine these two sensors to compensate for each other's weaknesses and construct a more robust system.Proposed a system that utilizes WiFi-based positioning methods, [4] for mobile robot-based learning data collection, localization, and tracking in indoor spaces. The system combines the extended Viterbi algorithm, tracking algorithm, odometer information, and a new signal fluctuation matrix to improve the accuracy of robot location tracking and the effectiveness of building a high-quality WiFi Radio Map.

With the help of WiFi information, they select a subset of RGBD images that correspond to the similar location range as the current frame for loop closure detection, thus avoiding the perceptual aliasing problem. In addition, computational complexity can be reduced because of the low computation overhead of determining WiFi similarity, and the number of RGBD images in the database that need to be searched is decreased by filtering loop closure candidates via their WiFi similarity. In our system, we also integrate WiFi with visual SLAM to tackle the false loop closure problem by associating a keyframe with corresponding WiFi information. However, instead of storing the WiFi fingerprint or signature, we store a pose estimated by the WiFi SLAM module in our system. Furthermore, our system not only solves the perceptual aliasing problem but also provides a coarse robot position sup- ported by our WiFi SLAM module to make our system more robust when visual SLAM is out of function.

Both Extended Kalman Filter (EKF), [5], [6] and Graph Optimization are popular techniques used for Simultaneous Localization and Mapping (SLAM) in robotics and computer vision. EKF SLAM uses a state vector to represent the robot's pose and the map's features and estimates the state vector by incorporating sensor measurements such as odometry and range measurements. EKF SLAM is computationally efficient and is widely used in mobile robotics applications. On the other hand, Graph Optimization represents the SLAM problem as a graph, where nodes represent robot poses and landmarks and edges represent constraints between them. Graph Optimization finds the optimal estimate of the robot's trajectory and the map by minimizing the error between the constraints and the estimated values. Although Graph Optimization is computationally more expensive than EKF, it is a global optimization technique that can improve the accuracy of SLAM estimates. In recent years, with the improvement of hardware, graph optimization has become increasingly popular in modern SLAM algorithms.

The closest work to ours is [6], where an EKF-based SLAM using WiFi signal strength is proposed to estimate the pose of the robot and the locations of the access points (APs) in the environment. The pose estimated by WiFi signal can be further used to improve loop closure in visual SLAM and provide a rough localization result. This work estimates the robot pose using a WiFi signal and RGBD images based on an Extended Kalman Filter (EKF), [6]. Graph optimization is only conducted when the last frame is detected, to optimize the pose estimation. In contrast, our system is a full graph optimization-based system. We implement both our visual SLAM and WiFi SLAM modules based on graph optimization due to the advantage of graph optimization that it takes the whole history state into account and is a more accurate approach that can handle non-Gaussian errors, whereas EKF only considers recent states, and the disadvantages of EKF that assumes that the system's error is Gaussian and may lead to inconsistency in highly non-linear systems.

2.4 Degenercy Detection

Sensors have an inevitable degradation problem.For example, a vision sensor may degrade in cases of poor lighting, occlusion, and featureless scenes. Similarly, a Lidar sensor may degrade in scenarios with self-symmetry or fewer geometric constraints. When faced with such degradation, a SLAM system may lose track. To improve the robustness of a SLAM system, A well-known work, [7], [8] proposed a general mechanism to detect degeneracy. This work defines an optimization based state estimation problem as $\arg\min\|Ax - b\|_2$ and a degeneracy factor, $D = \delta d/\delta X_c$, where δX_c represents the maximum amount of shift of an artifact constraint, and δd is the difference between the original estimation result and the estimation result affected by the artifact constraint. After a series of mathematical deductions, the degeneracy factor $D = \lambda_{\min} + 1$, where $\lambda_{\min}$ is the smallest eigenvalue of $A^T A$. With this lemma, we can detect a degeneracy by setting a threshold for the minimum eigenvalue and further integrating sensor data extraction to compensate for the degradation.

3 Method

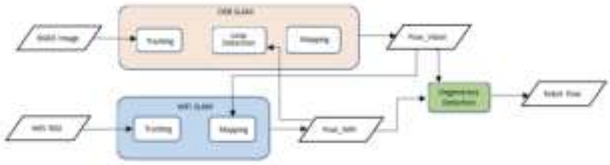

Fig. 3: illustrates the proposed system, [1], [3]

3.1 System Diagram

The input is a pair of RGBD images and WiFi RSSI（Received Signal Strength Indication) values, and the output is the robot poses. The Visual SLAM module, based on ORB-SLAM3, uses RGBD images as input and outputs an estimated robot pose determined by visual information. On the other hand, the WiFi SLAM module utilizes WiFi RSSI values as input and outputs a robot pose estimated using the WiFi signal. If a vision degeneracy is detected, the pose estimated through WiFi is utilized instead.The whole system is shown in Figure 3.

3.2 Graph-based SLAM

A SLAM problem, [1], can be formulated as a MLE (Maximum Likelihood Estimation) problem with a probability model

$$\boldsymbol{X}^* = \arg maxP(\boldsymbol{X} \mid \boldsymbol{Z})$$

where X represents the state and Z represents the observation.

Two main approaches are solving the state estimation problem, while traditional SLAM tends to use filter-based approaches such as Kalman filters and Particle filters, [5], modern graph-based SLAM, [1], [6] uses a least-squares approach, turning a SLAM problem into a least-squares problem and solving it with the optimization algorithm.

In ORB-SLAM, [6], map points $\mathrm{MP}_{\mathrm{wj}} \in$ R3 and robot poses $\mathrm{t}_{\mathrm{iw}} \in$ SE(3), where w stands for the world reference, are optimized minimizing the reprojection error with respect to the matched keypoints $\mathrm{mp}_{\mathrm{ij}} \in$ R2, the error function is:

$$\mathrm{e}_{\mathrm{ij}} = \mathrm{mp}_{\mathrm{ij}} - \pi(\mathrm{t}_{\mathrm{iw}}, \mathrm{MP}_{\mathrm{wj}})$$

where π is the projection function.

In our system, as illustrated in Figure 4, WiFi access points are utilized as landmarks along with Visual SLAM map points. And this is precisely the novelty of our paper.

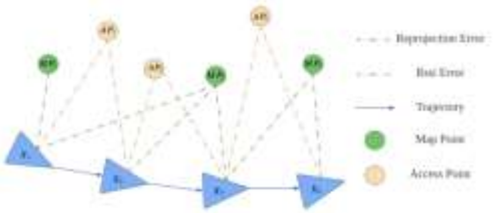

Fig. 4: Graph visualization of our system

3.3 WiFi SLAM Module

3.3.1 Propagation Model

The signal propagation model plays an important role in indoor localization systems based on WiFi-received signal strength indication (RSSI). WiFi RSSI attenuates with distance.Signal propagation model, [8] is described by:

$$P(d) = P(d_0) - \eta 10 \log_{10}\left(\frac{d}{d_0}\right)$$

where d and d0 are the distances from the transmitter, P (d) and P (d0) are the received RSSI(dbm) at distance d and d0, and η is the path loss exponent.

3.3.2 Mapping Submodule

Mapping the submodule of the WiFi module is implemented by estimating access point (AP) positions, [9], [10]. As illustrated in Figure 4, WiFi access points serve as landmarks in our system. The location of these access points is continually estimated and updated to maintain an up-to-date WiFi map. The keyframe class in ORB-SLAM3 has been modified to include the observation of access points APm ($m \leq M$, with M being the number of the access points in the environment) at corresponding location xi, along with their respective RSSI values. When an access point is observed more than α times ($\alpha \geq 3$) and has an RSSI value greater than β(dBm), it will be selected as a candidate node in the graph. Before starting optimization, the status of candidate nodes will be further evaluated to ensure that they have been properly initialized since a good initialization of nodes is crucial for optimal results. To initialize, the average location of all locations where APm was observed will be taken as the initialization value.

3.3.3 Tracking Submodule

With the aid of a WiFi map that is kept up-to-date by the WiFi Mapping module, the tracking submodule can determine the robot's pose, denoted as x_i ,using a similar approach as the mapping submodule.At an uncertain robot's pose x_i, we can receive RSSI values rim from each access point $\mathrm{AP_m}$. By following the same method used in the mapping submodule, we can construct a graph with $\mathrm{AP_m}$ as fix vertices, the difference between the estimated RSSI value and the received RSSI value as edges, and our estimated robot's pose $\mathrm{x}_i$as an estimated vertex. After optimization, the robot's pose, x_i can be estimated. The main difference between the tracking and mapping submodules is that the tracking submodule aims to estimate the robot's pose with known fixed AP positions, while the mapping submodule aims to estimate AP locations with fixed robot's poses.

Algorithm 1 Estimate Access Point (AP) Positions

Input: RSSI values r_{im} of access points AP_m and corresponding location $\mathbf{x_i}$ estimated by Visual SLAM module

Output: Updated positions of the access points

List: CandidateAps
Graph: G
for each AP_m in AP_M **do**
  if RSSI $r_{im} \geq \beta$(dBm) and observed time $\geq \alpha$ **then**
    Add AP_m to CandidateAps
  end if
end for
for each AP_m in CandidateAps **do**
  if AP_m is not Initialized **then**
    Initialize(AP_m)
  else
    G.addVertex(AP_m)
  end if
end for
for each corresponding location $\mathbf{x_i}$ **do**
  G.addFixedVertex($\mathbf{x_i}$)
  G.addEdge($edge_{im}$)
end for
Optimize G with Levenberg − Marquardt algorithm
Return updated positions of the access points

Fig. 5: Estimation Access Point Algorithm

Explained the above method more clearly. We employed techniques from Visual SLAM (Simultaneous Localization and Mapping) to initially estimate the positions of WiFi Aps and our algorithm about estimating WiFi Aps is shown below in Figure 5 (Algorithm 1). Once these positions are approximated, they are used in conjunction with WiFi RSSI (Received Signal Strength Indicator) values to enhance pose estimation. The algorithm for pose estimation is shown below in Figure 6 (Algorithm 2). This hybrid approach leverages the strengths of both Visual SLAM and WiFi signal analysis. By estimating WiFi locations first, the system can use these locations as additional data points for more accurate pose estimation than would be possible by relying solely on WiFi RSSI values for localization. This method provides a more robust and precise navigation framework by systematically refining both the map of the environment and the robot's understanding of its position within it.

```
Algorithm 2 Estimate Robot Pose
Input: WiFi map and received RSSI values r_im of access points AP_m
Output: Robot pose estimation
Graph: G
G.addVertex(x_i)
for each received RSSI value r_im of AP_m do
    if RSSI r_im ≥ β(dBm) and AP_m is Initialized then
        G.addFixedVertex(AP_m)
        G.addEdge(edge_im)
    end if
end for
Optimize G with Levenberg − Marquardt algorithm
Return robot pose estimation
```

Fig. 6: Estimation Access Point Algorithm

3.4 Visual SLAM Module

3.4.1 Base Visual SLAM Algorithm

ORB-SLAM3, [1] is a famous open-source and well-structured visual SLAM framework, which is used as a research tool by many students and researchers. We choose ORB-SLAM3 as the based visual SLAM algorithm in our system, finding out its weakness and improving it by integrating WiFi as an extra sensor, [9].

3.4.2 Loop Detection Submodule

To address the issue of false loop closures, we enhance the loop detection mechanism in the visual SLAM module by incorporating WiFi signals. We assume that WiFi RSSI values received in different locations from the same access points should be distinct and obtained in different locations should be distinguishable. This helps to rectify false loop closures that arise due to similar appearances in two distinct locations shown in Figure 7.

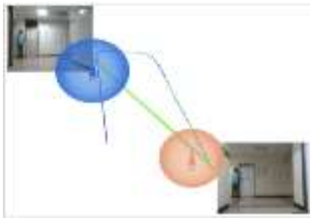

Fig. 7: Loop detection, [1], [8]

Associated with Parts 3.3 and 3.4, our system combines the advantages of both WiFi and Visual SLAM to achieve a more robust navigation solution. While WiFi-based indoor positioning techniques typically offer meter-level accuracy, they lack the precision of centimeter-level accuracy. On the other hand, relying solely on Visual SLAM can lead to instability in environments with insufficient features. By fusing these two technologies in our system, we create a synergy akin to an ensemble method in deep learning. Incorporating multiple sensors and optimizing their positioning allows us to achieve more robust and precise navigation outcomes.

4 Experiment

4.1 Dataset Setup of Experiments

There are a bunch of well visual SLAM benchmarks such as TUM dataset [11], EuRoC [1] dataset and KITTI dataset, [7], however, according to the best of our knowledge, there is not a SLAM dataset consists of RGB-D images and WiFi signal that can be used in our experiments to determine our system performance. As a result, we constructed our dataset on the fifth floor of the college of Electrical Engineering and Computer Science Building (CSIE), National Taiwan University. We use an RGB-D camera, Realsense D435i, produced by Intel to collect RGB-D images and an ASUS Zenbook pro15 laptop with WiFi 6E(802.11ax) network card to collect RSSI signal from access points in the environment shown in Figure 8.

Fig. 8: Dataset Setup of Experiments

CSIE 5F is a typical corridor environment. The total area of our experimental space is approximately 870 square meters, and it consists of around 326 access points, including WiFi 2.4G (802.11b) and WiFi 5G (802.11ac). For a visual representation of the access points detected during our data collection process, please refer to Figure 6.

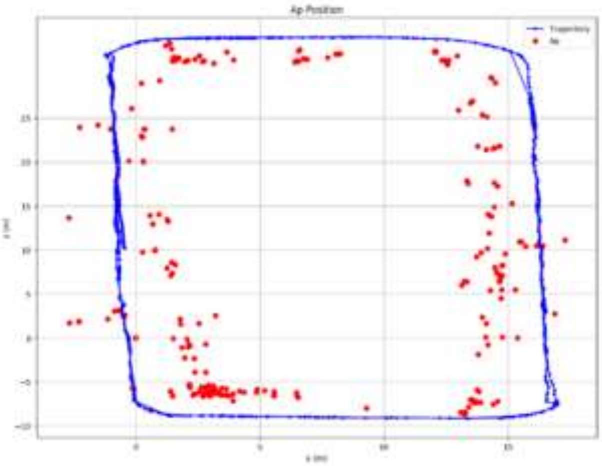

Fig. 9: Access Points Distribution

4.2 Data Preprocessing

Fig. 10: Data preprocess

Due to the inherent characteristics of the devices, the RGB-D image data is captured at a frequency of 15 frames per second (fps), while the WiFi RSSI data is collected at a considerably lower frequency of only 1 fps. To ensure accurate data preprocessing, it is essential to conduct a data association and synchronization process to align and harmonize the two inputs. For this purpose, we leverage the capabilities of ROS (Robot Operating System), a versatile and open-source framework widely adopted in robotics for developing and programming robotic systems. ROS provides a rich collection of tools and libraries that enable us to associate the RGB-D images and WiFi data by aligning their timestamps, ensuring synchronization, and merging them into a unified data stream that seamlessly integrates into our system.

4.3 Evaluation and Comparison

In Wifi SLAM module,we use RSSI signal strength. Only an RSSI value greater than β(dBm) will be considered as valid data. To determine the optimal threshold, we conducted tests using various RSSI values ranging from -100 dBm to -40 dBm. After careful evaluation, we have decided to set the threshold at -60 dBm. This threshold demonstrated lower error and maintained an adequate number of valid access points, making it a suitable choice for our system. The experimental result is depicted in Figure 11.

Metric/Threshold (Meter/dbm)	Non	-80	-60	-40
Rmse	11.11	9.53	6.15	More accurate but the number of Aps is not enough.
Median	9.28	7.18	4.58	
std	5.52	4.89	2.98	
Min	0.71	0.45	0.59	
Max	25.65	24.64	13.57	

Fig. 11: RSSI Threshold Table

As our system aims to create a robust and sustainable solution that can continuously update the vision mapping and WiFi mapping information, we conducted a test to simulate the long-term operation of a robot in the environment. As depicted in Figure 9 and Figure 10, our WiFi SLAM module demonstrates the capability to improve its accuracy in real-time without requiring manual interventions shown in Figure 12 and Figure 13.

Metric/Run (Meter)	1	2	3	4	5
Rmse	6.95	4.76	3.98	3.55	3.09
Median	5.83	3.68	2.68	2.91	2.22
std	3.56	2.54	2.32	1.94	1.67
Min	0.40	0.51	0.39	0.19	0.13
Max	15.57	9.36	12.03	9.54	5.70

Fig. 12: Longterm Accuracy Table

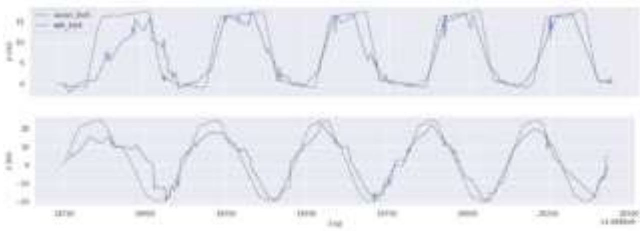

Fig. 13: Longterm Accuracy Graph

Compared with Offline Methods, WKNN (Weighted k-Nearest Neighbors) [11] and Random Forest, [4] are commonly employed techniques for WiFi localization. Conversely, Wi-Fi DSAR [12], [13] is a machine learning-based approach that utilizes an auto-encoder. In Figure 11 the error comparison between our method and these approaches is depicted. The results indicate that despite the challenge of online updating without prior knowledge, which is crucial for a SLAM system, our method maintains an acceptable level of performance when compared to these offline database-dependent methods.

Metric/Method (Meter)	WKNN	Random Forest	WiFi DSAR	Our Method
RMSE	2.63	3.52	2.09	3.09

Fig. 14: Comparison with Offline Methods

Compared with EKF, the filter-based approach has, in theory, lower accuracy compared to the graph optimization approach. In practical terms, the error comparison between our method and EKF (Extended Kalman Filter) depicted in Figure 14 and Figure 15 shows that our method utilizing graph optimization exhibits higher accuracy than the EKF approach.

Metric/Method (Meter)	EKF	Our Method
RMSE	4.02	3.09

Fig. 15: Comparison with EKF

In the Visual SLAM module, to showcase the robustness of our system against common challenges such as lighting variations, occlusion, and featureless environments, and to address the issue of false loop detection caused by similar visual scenes, we performed a series of experiments specifically designed to simulate these scenarios.

As depicted in Figure 13, we intentionally created an environment with insufficient lighting to observe the behavior of the system. In the case of pure visual SLAM, the system experienced track loss, resulting in an incorrect trajectory. However, when we integrated the WiFi SLAM module to compensate for the challenging lighting conditions, the trajectory remained correct. This demonstrates the effectiveness of the WiFi SLAM module in improving robustness and ensuring accurate trajectory estimation even in challenging lighting situations. Figure 14 provides additional insight into the performance of the two modules. When the visual SLAM module lost track, the minimum eigenvalue associated with it approached 0, indicating a degenerated state. In contrast, the minimum eigenvalue of the WiFi SLAM module remained higher than 300. This demonstrates that even when the visual information degrades, the WiFi signal can still provide reliable measurements without suffering from degeneracy.

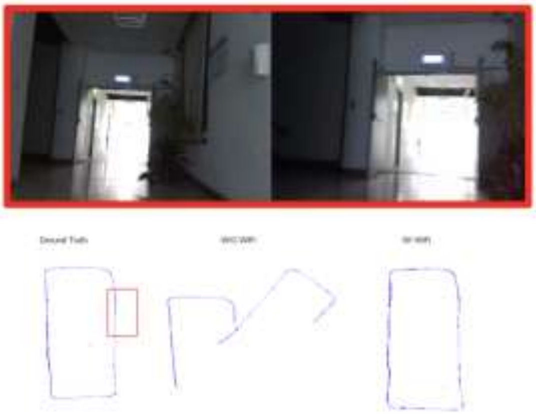

Fig. 16: Lighting Challenge Trajectories

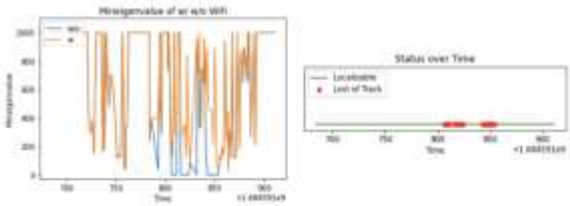

Fig. 17: Minimum Eigenvalue Comparison (Lighting)

Similarly, we designed a scenario where a person continuously walked around the environment, leading to tracking loss due to occlusion. As illustrated in Figure 16 and Figure 17, the integration of the WiFi SLAM module, [14], proved beneficial as it helped overcome the occlusion challenge, leading to accurate final results. Furthermore, Figure 17 displays a comparison of the minimum eigenvalues. It demonstrates that the WiFi SLAM module provides a non-degenerate constraint, improving the system's robustness when the visual SLAM module experiences degradation.

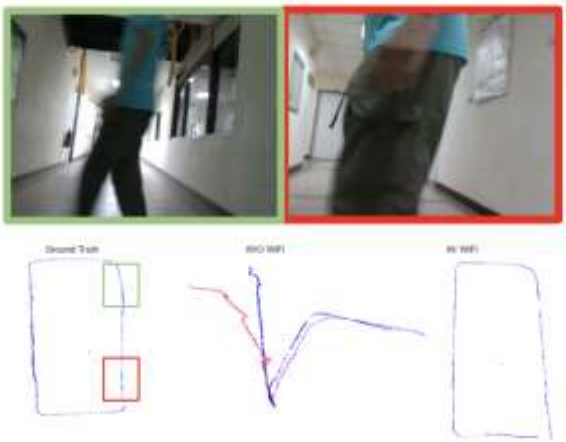

Fig. 18: Occlusion Challenge Trajectories

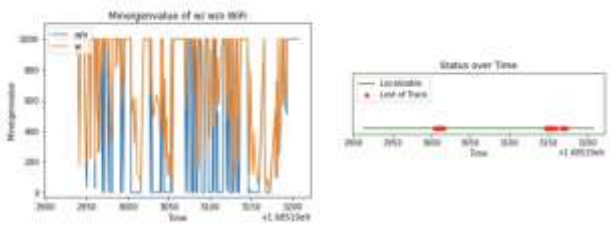

Fig. 19: Minimum Eigenvalue Comparison (Occlusion), [12]

Finally, as part of our evaluation, we intentionally designed two featureless scenes within the environment to assess the performance of our system. The results, depicted in Figure 18 and Figure 19, unequivocally demonstrate the significant contribution of WiFi integration in enabling the system to effectively handle featureless scenes. The WiFi integration proves to be a valuable asset in overcoming the challenges posed by the absence of distinct visual features, ultimately enhancing the system's performance and reliability.

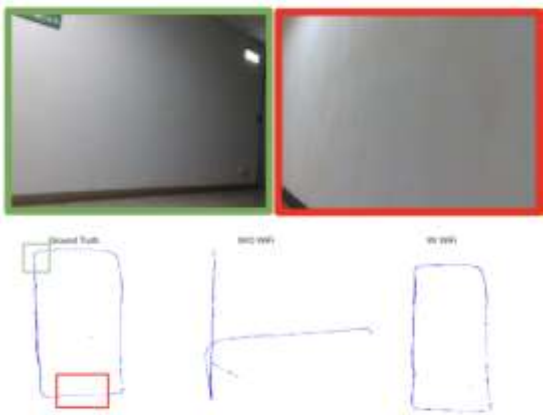

Fig. 20: Feature-less Scene Challenge Trajectories

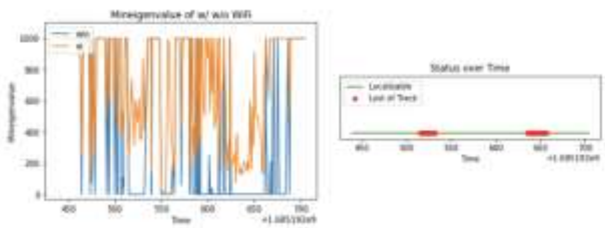

Fig. 21: Minimum Eigenvalue Comparison (Featureless), [12]

To assess the system's capability to eliminate false loop detection caused by two visually similar scenes in different locations, [15], we deliberately designed visually similar environments at two distinct places.

The result depicted in Figure 20, Figure 21 and Figure 22 demonstrates that the original ORB-SLAM3 fails to differentiate between these two locations, leading to a false loop detection. As a consequence, the system corrects the trajectory based on this false loop detection, resulting in an incorrect trajectory.

In contrast, our system incorporates WiFi information to filter the loop detection process. Consequently, these two visually similar places are not identified as a loop, preventing the system from making incorrect trajectory corrections based on false loop detections. If the Visual extraction is recognized as the same place (but it is not), our Wifi fingerprint system will prevent it from false loop detections by RSSI value outlier removal.

By Figure 20 displayed, we can discover the trajectory can recognize it is not in the same place so that it will not cause false loop detection which is displayed in the bottom right corner of the image. If we just use pure Visual SLAM (ORBSLAM3), it will cause false loop detection and thus the accuracy would drop very sharply.

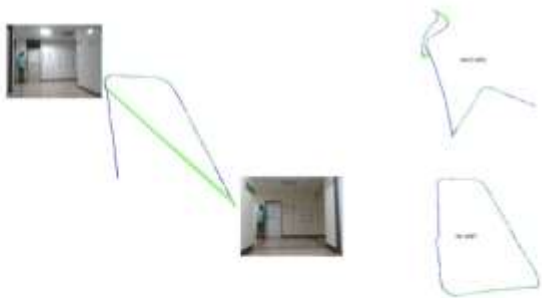

Fig. 22: False Loop Detection, [1], [4]

5 Conclusion

Our contribution reproduced a novel structure combine Visual SLAM and Wifi real-time interactive framework positioning system and mutually helps each other drawback. Our research leverages data from both WiFi and visual sensors, along with degeneracy detection techniques which are more robust then ORBSLAM3, [1]. This framework effectively enhances the robustness of visual SLAM by addressing challenges such as lighting variations, occlusion, and featureless scenes. Additionally, our proposed solution successfully eliminates the issue of false loop detection. By combining WiFi and visual information and implementing advanced detection mechanisms, our framework offers an innovative approach to improving the performance and reliability of SLAM systems.

However, SLAM and Wifi positioning still have some limitations. Although it can have centimeter-level accuracy, we cannot order them to do some task simply by this framework. With the rise of multimodal research like VLM(Visual Language Model) and LLM (Large Language Model) we can not only position but also navigate or order instructions to robot.This will become our future work to research to improve our system.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used ChatGPT in order to in order to improve the readability. After using this tool/service, the authors reviewed and edited the content as needed and takes full responsibility for the content of the publication.

References:

[1] C. Campos, R. Elvira, J. J. Gomez, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multi-map SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[2] C.-C. Chou. *Enhance SLAM Performance with Tightly-Coupled Camera and Lidar Fusion*. PhD thesis, National Taiwan University, Taipei, 2021.

[3] M. Abbas, M. Elhamshary, H. Rizk, M. Torki, and M. Youssef. Wideep: Wifi-based accurate and robust indoor localization system using deep learning. *In 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp.1–10. IEEE, 2019.

[4] G. Biau and E. Scornet. A random forest guided tour. *Test*, Springer Test, 25:197–227, 2016.

[5] A. H. Salamah, M. Tamazin, M. A. Sharkas, and M. Khedr. An enhanced wifi indoor localization system based on machine learning. *In 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp.1–8, 2016.

[6] G. Welch, G. Bishop, et al. *An introduction to the kalman filter*. 1995.

[7] J. Zhang, M. Kaess, and S. Singh. On degeneracy of optimization-based state estima- tion problems. *In 2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp.809–816. IEEE, 2016.

[8] G. Lee, B.-C. Moon, S. Lee, and D. Han. Fusion of the slam with wi-fi-based positioning methods for mobile robot-based learning data collection, localization, and tracking in indoor spaces. *Sensors*, 20(18):5182, 2020.

[9] C.-Z. Sun, B. Zhang, J.-K. Wang, and C.-S. Zhang. A review of visual slam based on unmanned systems. *In 2021 2nd International Conference on Artificial Intelligence and Education (ICAIE)*, pp.226-234. IEEE, 2021.

[10] A. H. Salamah, M. Tamazin, M. A. Sharkas, and M. Khedr. An enhanced wifi indoor localization system based on machine learning. *In 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp.1–8, 2016.

[11] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. *In Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[12] Y.-H. Wang, T.-W. Yang, C.-F. Chou, and I.-C. Chang. Wi-fi dsar: Wi-fi based indoor localization using denoising supervised autoencoder. *In 2021 30th Wireless and Optical Communications Conference (WOCC)*, pp.188-192, 2021.

[13] Yongbo Chen, Liang Zhao , Ki Myung Brian Lee. Broadcast Your Weakness: Cooperative Active Pose-Graph SLAM for Multiple Robots *In 2020 IEEE Robotics and Automation Letters*,

[14] Morgan Quigley,Ken Conley,Brain PGerkey.ROS: an open-source Robot Operating System. *ICRA 2029 Workshop on Open Source Software*.

[15] W. Xue, W. Qiu, X. Hua, and K. Yu. Improved wi-fi rssi measurement for indoor localization. *IEEE Sensors Journal*, 17(7):2224–2230, 2017.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

- Yi-Hsien Lu, carried out the Wifi and Visual fusion experiment preprocess and data collection.
- Chia-Chi Huang, implement C++ code to Wifi and Visual fusion ROS code
- Chih-Chung Chou, give instruction and suggestion
- Cheng-Fu Chou, give instruction and suggestion.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

The research is based on work supported by the National Science and Technology Council, Taiwan, under Grant number NSTC 112-2221-E-008-059-MY2, 112-2221-E-002 -118 -, 113-2221-E-002 -201.

Conflict of Interest

The authors have no conflicts of interest to declare.