DungeonAssistant: A Scalable Indoor Localization System

Haoyun Qin
qhy@seas.upenn.edu

University of Pennsylvania

Abstract

This paper presents DungeonAssistant, a *scalable* indoor localization system based on WiFi signal and LiDAR 3D reconstruction that provides significant operational benefits. We provide a systematic approach for data collection, evaluation, and fingerprint refinement on complex indoor localization settings and scenarios. We enabled the construction of such a system using merely an Android device and an iPad/iPhone device with LiDAR capability by a single person within 30 minutes. Our evaluation shows that the median distance error of our localization system is less than 4 meters within 2 days after the original data collection, which is already usable in production. We further evaluate the system and justify the need for such a painless and low-cost data collection system, by investigating the evolution of prediction error as the time being.

1 Introduction

Indoor localization is a crucial technology that enables a myriad of applications, including indoor navigation, asset tracking, emergency response, and location-based services. Despite the abundance of outdoor localization systems, such as GPS, indoor localization remains a challenging problem due to the complex and dynamic nature of indoor environments. The multipath effect, signal attenuation, and environmental changes all contribute to the difficulty of achieving accurate and reliable indoor localization.

Various technologies have been proposed to tackle the indoor localization problem, including infrared, ultrasonic, RFID, UWB, and WiFi, and etc. The approaches can also be categorized into Monitor Based Localization (MBL) and Device Based Localization (DBL). Survey [18] presents a detailed landscape of the current indoor localization system universe.

In this paper, we present DungeonAssistant, a scalable indoor localization system that addresses the challenges associated with the fingerprinting approach. Our main localization methodology follows the *RADAR* system [3]. We take the WiFi RSSI (Received Signal Strength Indicator) based MBL approach, meaning that we would not need any special devices on either user or receiver end, but can just leverage the extensive WiFi access points in the building to get accurate localization. Such localization typically employs a database of WiFi signal strength, also known as the fingerprint database, which is established for different locations in the indoor environment. However, the cost of establishing and maintaining the fingerprint database is substantial, as it requires the collection of WiFi signal strength at each location, which is a labor-intensive and time-consuming process. This problem is mostly not solved or is only partially solved in the past systems [6, 8, 11, 12, 14–17]. The existing methodologies typically presuppose familiarity with the floor plan or require data collection from multiple devices or agents. Alternatively, there is a compromise between accuracy and the utilization of the original approach. The novelty of DungeonAssistant lies in its integration of WiFi signal collection and LiDAR sensing, leveraging readily available devices such as iPads (or iPhones) and Android phones. This approach significantly reduces the cost of establishing the fingerprint database, as it allows for the simultaneous collection of WiFi signal strength and corresponding spatial data using a single person within 30 minutes, and simultaneously gives us the floor plan and the 3D reconstruction of the whole environment. Moreover, it provides a systematic approach for data collection, evaluation, and fingerprint refinement, enhancing the scalability and operational benefits of the system.

The main contributions of this paper are as follows: (1) we propose a WiFi-based indoor localization system that incorporates LiDAR sensing for efficient fingerprint database construction; (2) our system provides huge operational benefits such that a single person can create an indoor localization system under a complex indoor setting using an Android phone and an iPhone/iPad with LiDAR capability within 30 minutes; (3) we provide a detailed description of the system design, including the roles of system maintainer and user, the fingerprinting approach, and the nearest neighbor localization method; (4) we present an implementation of the system and an evaluation of its performance in a complex indoor environment; and (5) we published all the code and dataset involved in this project.

The remainder of this paper is organized as follows. Section 2 provides a detailed overview of the DungeonAssistant system design. Section 3 discusses some practical issues regarding the data collection process. Section 4 describes the implementation of the system, including the WiFi signal capturer, MultiScan modification, fingerprint dataset construction, and indoor localization demo app. Section 5 presents the evaluation of the system in a complex indoor environment, detailing the experimental setup and discussing the results. Section 6 shows some related works in the past and identifies our novelty. Section 7 discusses the limitations of the system and proposes directions for future work. Finally, we conclude the paper in Section 8.

2 Localization System Design

2.1 System Model

In DungeonAssistant, we distinguish two primary roles: the system maintainer and the user. The system maintainer is tasked with the initial setup and ongoing maintenance of the localization system, while the user is the entity that employs the system for indoor localization purposes.

The system maintainer's duties encompass the gathering of the critical data necessary for the system's operation. This data comprises pairs of signal data and corresponding spatial location, commonly referred to as the ground truth. The signal data, denoted by $S = \{S_1, S_2, \cdots, S_m\}$, is a multi-dimensional vector encapsulating the signal strength from various sources at a given location, where m denotes the number of overall signal access points. Each dimension of S correlates with the signal strength of a particular source. The spatial location, denoted by L, is a three-dimensional vector indicating the physical space where the signal data is collected. The pair (S, L) constitutes an entry in the fingerprint database D, which is a collection of such pairs. Formally, we have $D = \{(S_1, L_1), (S_2, L_2), ..., (S_n, L_n)\}$, where n represents the total number of entries in the database. Notice that we do not need to know the locations of signal access points of the indoor settings beforehand.

In practice, the system maintainer may also record a timestamp T for each pair in the database, which is crucial for synchronizing the data collection process across different devices. The timestamp is a scalar value representing the time at which the signal data is collected. Therefore, each entry in the database can be extended to a tuple (S, L, T).

Contrastingly, the system user lacks access to the ground truth location data, collects signal data S_u at an unknown location L_u. The user's objective is to estimate the unknown location L_u based solely on the collected signal data S_u and the fingerprint database D provided by the system maintainer.

2.2 Fingerprinting Approach

The fingerprinting approach in DungeonAssistant is grounded on an empirical observation from RADAR [3] systems, suggesting that a signal vector S can be instrumental for location identification. This is based on the premise that the signal strength from various sources, when collected at a particular location, forms a distinct pattern or "fingerprint". While it is not entirely accurate to state that a unique signal vector can identify a distinct location, it is plausible to associate a signal vector with a range of locations, thus defining a region with similar signal characteristics.

In static conditions, the fingerprinting approach can prove highly effective as the signal patterns remain relatively consistent. However, in dynamic environments where either the environment or signal sources may fluctuate, the efficacy of the fingerprinting approach may be impacted. This is attributed to the multipath effect, a phenomenon where signals traverse multiple paths from the source to the receiver, causing variations in the received signal strength. The multipath effect is a prevalent challenge in indoor localization systems and will be further addressed and evaluated in the section 5 and 7.

In contrast to the fingerprinting approach, triangulation methods are also frequently employed for location estimation. These methods, including lateration and angulation, estimate the user's location by determining the distance or angle from multiple known points. Nevertheless, these methods are not immune to the multipath effect, particularly in dynamic environments. Additionally, they necessitate precise knowledge of the signal access points' locations, which adds another layer of complexity and may not be feasible in all scenarios.

In the context of DungeonAssistant, the fingerprinting approach offers a practical and scalable solution for indoor localization. It capitalizes on the readily available WiFi signals and LiDAR sensing data, and the computation is relatively lightweight. Despite its limitations, this approach can deliver satisfactory performance for a broad spectrum of indoor localization applications when integrated with the proposed system design and data collection methodology.

2.3 Nearest Neighbour Localization

Given the preceding discussion, following RADAR [3], in DungeonAssistant, we employ the k-nearest neighbour ($k-$NN) algorithm as the core component of location prediction. The $k-$NN algorithm is a type of instance-based learning that infers the location from the proximity of its k nearest neighbours in the fingerprint database.

Mathematically, the process can be formalized as follows. Given a user's signal data S_u, we first identify the set of k nearest neighbours in the fingerprint database D. This is done by calculating the distance between S_u and every signal data S_i in D, then selecting the k entries with the smallest distances.

Let $\mathrm{NN}_k(S_u)$ denote the set of k nearest neighbours of S_u

in D. Formally, we have,

$$\mathrm{NN}_k(S_u) = \underset{\{(S_i,L_i,T_i)\}\subseteq D, |(S_i,L_i,T_i)|=k}{\arg\min} \sum_{i=1}^{k} d(S_u, S_i)$$

where $d(\cdot,\cdot)$ is the distance function, i.e. a function with the property of non-negativity, symmetry, triangular inequality, and identity. A common family of distance function is Minkowski distance, defined as:

$$d(x,y) = \left(\sum_{j=1}^{m} |x_j - y_j|^p \right)^{\frac{1}{p}}$$

The predicted location L_u is then given by the average of the locations of the k nearest neighbours:

$$L_u = \frac{1}{k} \sum_{(S_i,L_i,T_i)\in \mathrm{NN}_k(S_u)} L_i$$

By using the k-NN algorithm, DungeonAssistant can provide a robust and efficient location prediction, leveraging the rich and diverse signal data collected from accessible and inexpensive devices.

3 Data Collection

In DungeonAssistant, the process of data collection is a critical component of the overall framework. To optimize the utilization of available resources and ensure the efficient collection of both WiFi signal strength and LiDAR data, we adopt a distributed approach wherein data collection tasks are divided across different devices. This strategy allows us to leverage the unique capabilities of each device effectively, enhancing the overall quality and accuracy of the collected data.

However, the distributed nature of our data collection process introduces the need for time synchronization across the devices involved. This is crucial to ensure that the signal strength data and corresponding spatial data are accurately associated, forming valid entries in the fingerprint database.

To address this, we perform a time synchronization process before the start of each data collection round. This process aligns the clocks on the devices involved, establishing a common temporal reference that enables the accurate pairing of WiFi signal strength and spatial data.

It is worth noting that the assumption is made that the timestamp will not drift significantly during the course of each round of data collection. This assumption is reasonable given the relatively short duration of data collection rounds and the inherent clock stability of modern devices.

In the following subsections, we delve into the specifics of the WiFi RSSI collection and position collection processes, discussing the methodologies employed and their significance in the DungeonAssistant system.

3.1 WiFi RSSI Collection

The WiFi Received Signal Strength Indicator (RSSI) is a crucial parameter in our indoor localization system, DungeonAssistant, as it forms the basis of the signal data S used for location prediction. RSSI is a measure of the power level received by a device from a WiFi signal source. It is typically expressed in decibels (dBm), with larger values indicating stronger signals.

Given that iOS and iPadOS devices do not permit direct access to WiFi RSSI values, we leverage an Android device for this purpose. Android provides a well-documented API that allows for the collection of WiFi RSSI values, making it an ideal choice for our system [2].

We use the `WifiManager.startScan()` method to initiate a WiFi scan, which triggers an asynchronous scan for access points. The results of this scan, which include the RSSI values, can be retrieved using the `WifiManager.getScanResults()` method. This method returns a list of `ScanResult` objects, each of which encapsulates the details of an access point, including its RSSI value and MAC addresses.

It is worth noting that the default scan rate on Android devices is approximately once every 30 seconds. However, to obtain a higher refresh rate of once every 2 seconds, one need to enable the "WiFi scan throttling" option in the developer settings. This allows us to collect more granular data, enhancing the accuracy of our localization system [2].

The use of these methods necessitates specific permissions in the Android application manifest. The two location permissions, `ACCESS_COARSE_LOCATION` and `ACCESS_FINE_LOCATION`, are required to retrieve the WiFi scan results. Additionally, the permissions to access (`ACCESS_WIFI_STATE`) and change (`CHANGE_WIFI_STATE`) the WiFi state are necessary to initiate a WiFi scan [1].

3.2 Position Collection

Overview. LiDAR (Light Detection and Ranging) technology has emerged as a powerful tool for 3D reconstruction, providing highly accurate, dense point clouds of the environment. It operates by emitting laser beams and measuring the time taken for the light to bounce back after hitting an object. This time-of-flight (ToF) information is then used to calculate the distance between the LiDAR sensor and the object, thereby enabling the creation of 3D representations of the surroundings.

In the framework of our indoor localization system, DungeonAssistant, the integration of LiDAR-based 3D reconstruction is multifaceted. Primarily, it enriches the spatial data collected, thereby refining the precision of the fingerprint database. Additionally, the 3D models derived from the LiDAR data offer an intuitive visualization of the environment, which is invaluable for system maintainers during both the initialization and upkeep phases. Furthermore, these 3D reconstructions significantly contribute to the development of user-end applications. The comprehensive environment information captured by the 3D models can be utilized to create interactive and user-friendly interfaces, thereby enhancing the user experience. This is particularly beneficial for informing users of their exact location within the mapped environment, offering a seamless and intuitive navigation experience. See section 7 for more discussion.

For the sake of accessibility and cost-effectiveness, we opted for the iPhone/iPad platform for LiDAR data collection. Some recent models of these devices are equipped with LiDAR scanners, making them a readily available and practical choice for our system. Moreover, the high-quality sensors and robust software ecosystem of these devices ensure reliable and accurate data collection.

To facilitate the process of 3D reconstruction, we employ the MultiScan project [13]. MultiScan is a scalable RGBD scanning framework specifically designed for 3D environments with articulated objects. It leverages Apple's ARKit [9] on the iPad/iPhone, obtaining camera intrinsic parameters, depth maps, RGB images, and camera extrinsic parameters at a rate of 60fps during the data collection phase via a dedicated application. The data collected is then processed on a server using Open3D [21], a modern library for 3D data processing, featuring powerful 3D algorithms and data structure in C++ and a Python binding, to generate the 3D models.

Trajectory Extraction. The trajectory of the device is a vital component of DungeonAssistant. It essentially represents the spatial path traversed by the device, which is captured in the form of a sequence of camera extrinsic in the world coordinate system. This sequence of positions, denoted by $L = \{L_1, L_2, \cdots, L_p\}$, where p represents the number of collected positions, is extracted from the transformation matrix provided by Apple's ARKit [9].

In the context of ARKit's coordinate system, the transformation matrix encapsulates the position and orientation of the camera in the world coordinate system at a given point in time [10]. The matrix is a 4x4 homogeneous transformation matrix $\mathbf{M}$, with the upper-left 3x3 sub-matrix $\mathbf{R}$ representing rotation and the upper-right 3x1 sub-vector $\mathbf{T}$ representing translation, denoted as

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix}$$

To extract the trajectory, we introduce a middleware in the MultiScan's processing server. This middleware intercepts the transformation matrix provided by ARKit, and extracts the translation vector. The extracted translation vectors form the trajectory T, which is then used in conjunction with the collected WiFi RSSI values to construct the entries in the fingerprint database D.

Post Closure Optimization.

Though MultiScan [13] can help us with most of the indoor reconstruction settings, is not without its challenges. One such challenge arises from the inherent limitations of ARKit's [9] loop closure detection capabilities, or rather, the lack thereof. This oversight is particularly evident in large and complex indoor environments, where the accumulated error in the ARKit visualization often results in the inability to close loops. This issue is further reflected in the extrinsic data collected by the MultiScan [13] application, leading to discrepancies in the 3D models generated by Open3D [21]. To address this, we introduce a *Post Closure Optimization* process, as visualized in Figure 1.

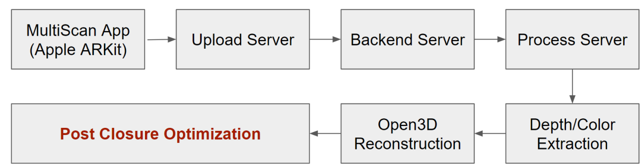

Figure 1: Post Closure Optimization Diagram

The *Post Closure Optimization* process is grounded in the method proposed by Choi *et al.* [7]. Following this method, we first use an overlapping sliding window to crop our point cloud data $\mathcal{P}$ into $\mathbf{P} = \{P_1, P_2, \cdots, P_k\}$, then we incorporate our design with multiway registration using Point2Plane Iterative Closest Point (ICP) [5] and Pose Graph Optimization.

The Point2Plane ICP [5] is a variant of the traditional ICP [4] algorithm, which minimizes the distance between corresponding points in two point clouds. It tries to iteratively find the transformation matrix $\mathbf{T}$ by minimizing

$$\sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{K}} \| \mathbf{p} - \mathbf{T}\mathbf{q} \|^2$$

where $\mathcal{K}$ is the correspondence set between target point cloud $\mathbf{P}$ and source point cloud $\mathbf{Q}$ (notations taken from [21]). While in Point2Plane ICP, the distance minimized is between points in one point cloud and the planes defined by points in the other point cloud as following,

$$\sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{K}} \| (\mathbf{p} - \mathbf{T}\mathbf{q}) \cdot \mathbf{n}_{\mathbf{p}} \|^2$$

This modification allows for faster convergence, which is crucial in large-scale environment mapping.

With pairwise Point2Plane ICP [5], Pose Graph Optimization is performed, which is a technique used to refine the poses (i.e., the orientation and translation of each point cloud). It works by creating a graph where the nodes refer to point clouds, and edges connect point clouds that overlap. The optimization process seeks to adjust the poses to minimize the discrepancy between the observed and expected constraints.

By incorporating these methods into the DungeonAssistant system, we can effectively mitigate the loop closure issues inherent in ARKit, thereby enhancing the accuracy and reliability of the 3D reconstructions. This, in turn, improves the overall performance of the indoor localization system.

4 Implementation

The implementation of DungeonAssistant involves the development of several key components, each of which is designed to fulfill a specific role within the system. The components are meticulously integrated, ensuring seamless data flow and efficient operation of the system. Below, we provide an overview of each component and discuss their respective functionalities.

4.1 WiFi Signal Capturer

The WiFi Signal Capturer is an Android application developed to collect WiFi signal strength and timestamp pairs. The application leverages Android's WiFi API to initiate WiFi scans and retrieve the resulting scan data, which includes the RSSI values of detected access points. Along with each RSSI value, the application records a timestamp, which is crucial for data synchronization across different devices. Other than the required functionalities, we also record information such as SSID, frequencies, capabilities, and other access point properties might be useful for further research of the access point deployment of a specific indoor setting. The WiFi Signal Capturer application is open-source and available at https://github.com/JeffersonQin/WiFiSignalCapturer.

4.2 MultiScan Modification

We made substantial modifications to the MultiScan project to enable it to scale and process large scene data (up to 100m $\times$ 120m). The modifications primarily focus on enhancing the data processing capabilities of MultiScan, ensuring its performance in large and complex indoor environments. We incorporated overlapping sliding window cropping, pipelining mechanism and other engineering efforts to scale up the system on a single node cluster with single A6000 GPU. The modified MultiScan project is open-source and available at https://github.com/JeffersonQin/multiscan-modified.

4.3 Fingerprint Dataset Construction

We also implemented a collection of scripts developed for point cloud registration (see section 5 for more detail) and post-closure optimization. The scripts leverage advanced algorithms, including Point2Plane ICP and Pose Graph Optimization (see section 3.2 for more detail), to refine the poses of point clouds and mitigate loop closure issues. These scripts play a crucial role in enhancing the accuracy and reliability of the 3D reconstructions generated by the system. The code is open-source and available at https://github.com/JeffersonQin/DungeonAssistant.

4.4 Indoor Localization Demo App

We developed a demo application to showcase the indoor localization capabilities of DungeonAssistant. The application is designed for the first floor of the School of Applied Science and Engineering at the University of Pennsylvania and features a floor plan extracted from the reconstructed 3D scene. Users can utilize the application to determine their location within the mapped environment, offering a practical demonstration of the system's capabilities. See Figure 2 for the user interface. The demo application is open-source and available at https://github.com/JeffersonQin/DungeonAssistant/tree/master/android. A demo video can also be found here https://youtu.be/XT8v7n0ZuOk.

Figure 2: The user interface of the indoor localization demo app for the first floor of the Engineering Building of the University of Pennsylvania. The red dot indicates the user's location.

Each of these components plays a vital role in the operation of DungeonAssistant. Their integration forms a comprehensive system that is capable of collecting rich and diverse data, processing this data to generate accurate 3D models, and utilizing these models for efficient and reliable indoor localization.

5 Evaluation

Our evaluation is conducted in the first floor of the Engineering Quad Building at the University of Pennsylvania, which features a diverse range of WiFi access points and a complex indoor environment. The evaluation aims to assess the system's performance in terms of localization accuracy and the necessity of an efficient data collection technique.

For reproducibility, we have also published the dataset here: https://huggingface.co/datasets/gyrojeff/DungeonAssistant.

5.1 Experimental Setup

The experimental setup for the evaluation involves the collection of LiDAR data and WiFi RSSI values from 258 distinct WiFi access points within the first floor of the building. The data collection process is conducted using an iPad Pro (11-inch) (4th generation) and an Android phone (Xiaomi MIX 4), as described in Section 4. We simply use tape to attach two devices. The collected data is then processed to construct the fingerprint database D.

For the localization prediction, we employ the k-nearest neighbour ($k-$NN) algorithm with $k = 1$ and Euclidean distance function. This means that for a given signal data S_u, the system predicts the location L_u based on the location associated with the closest signal data in D on Euclidean distance.

To account for the temporal variations in the WiFi signals, we introduce an aggregation time of 1 second. This means that the system averages the signal data collected within each 1-second interval before using it for location prediction.

For the part of post closure optimization, we use 6000 frames (i.e. 100 seconds under 60fps recording rate) as the overlapping length for the sliding window.

5.2 3D Reconstruction

Figure 3 presents the 3D reconstruction of the first floor of the Engineering Quad Building at the University of Pennsylvania, generated using the data collected by DungeonAssistant after concatenation and closure optimization (section 3.2). The 3D model accurately represents the physical layout of the building, including the locations of walls, doors, and other structural features.

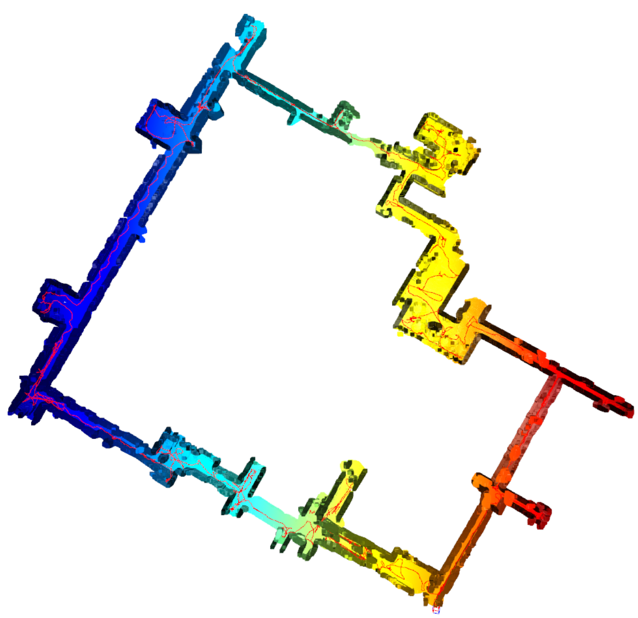

Figure 3: 3D reconstruction of the first floor of the Engineering Quad Building at the University of Pennsylvania. The red line indicates the trajectory of the agent collecting data

5.3 Localization Accuracy

The accuracy of DungeonAssistant is evaluated by comparing the predicted locations with the ground truth locations. The ground truth locations are obtained from the LiDAR data collected during the data collection phase. The predicted locations are derived from the WiFi signal data using the $k-$NN algorithm.

Table 1 presents the distribution of the 3D coordinate prediction errors over two distinct time periods: approximately 1 day and 2 days after the system setup. The prediction error is defined as the Euclidean distance between the predicted locations and the ground truth locations. Also, since the ground truth data we collect and the original dataset are potentially in two different worlds coordinates, we also need to perform a global registration to align the point clouds and thus transform the data collected into the dataset world coordinates. Here we employ the technique of downsampling to deal with the scaling issue, and leverage the Fast Global Registration (FGR) [20] method to perform the registration.

As the first two rows shown in the table, the mean prediction error is around 4 meters for the first two days after the original dataset is collected, indicating overall precision of our method. The standard deviation of the prediction errors is less than 3.5 meters, suggesting a relatively tight distribution of the errors around the mean value. The maximum prediction error is around 20 meters, which is reasonable given the complexity of the indoor environment and the inherent variability of WiFi signals. The median prediction error is less than the mean, indicating that the majority of the prediction errors are on the lower side. The 25th and 75th percentiles further confirm this observation.

For the third row, we collected the data again about 10 days after the original dataset collection. We can see that as the time elapses, the fingerprint will no longer be accurate and the prediction will drift dramatically. The median error increased from 3.24 meters to more than 10 meters. This further indicates the need of a painless data collection system such that we could update the fingerprint dataset constantly with low cost.

In summary, the evaluation results indicate that DungeonAssistant delivers robust and reliable indoor localization performance. Despite the inherent challenges of indoor environments and the variability of WiFi signals, the system is able to provide accurate location predictions, making it a practical solution for a wide range of indoor localization applications.

6 Related Work

NeRF^2. NeRF^2 proposes a Neural Radio-Frequency Radiance Field (NeRF^2) that models RF signal propagation in complex environments using a neural network [19]. With a small amount of signal measurements, NeRF^2 can predict signals at any location given the transmitter's position. However, while it could potentially reduce the amount of data collection and be integrated with our system, the approach of NeRF2 is fundamentally different from ours, focusing on the physical layer of RF signals rather than the application layer. Also, we might need to know the locations of AP around the settings to employ the NeRF^2 technique.

Accurate Indoor Localization With Zero Start-up Cost. This work presents Ubicarse, an indoor localization system for commodity mobile devices that requires no specialized infrastructure or fingerprinting [12]. Ubicarse uses a formulation of Synthetic Aperture Radar (SAR) to emulate large

Time Elapsed	Mean (m)	Std (m)	Median (m)	Max (m)	Min (m)	25th (m)	75th (m)
$\sim$ 1 Day	**4.06**	**2.97**	3.48	**17.19**	0.11	1.76	**5.42**
$\sim$ 2 Days	4.28	3.45	**3.24**	20.85	**0.05**	**1.69**	6.33
$\sim$ 10 Days	12.05	8.07	11.02	41.23	0.15	5.65	17.37

Table 1: 3D Coordinate Prediction Error Distribution

antenna arrays and combines RF localization with stereo-vision algorithms to localize common objects without RF sources. However, it does not provide a floor plan and the reconstruction of the environment, which limits its usability in some applications. Interestingly, it uses a similar technique to geotag small objects in the scene.

Indoor Localization Without the Pain. The work in [6] presents an indoor localization system that requires no pre-deployment effort. The system relies on WiFi signal strength measurements and occasional GPS location fixes from mobile devices. However, it still requires knowledge of the building's floor plan, which can be a significant limitation in many scenarios.

ARIEL. ARIEL is a room localization system that automatically learns room fingerprints based on occupants' indoor movements [11]. It uses WiFi signatures to identify occupancy hotspots and inter-zone correlations. Despite its high accuracy, ARIEL focuses on room localization rather than precise position prediction. Furthermore, it requires knowledge of the floor plan, which is not always available.

7 Discussion and Future Work

3D Reconstruction Accuracy for Evaluation. While the 3D reconstruction of the indoor environment provided by DungeonAssistant is generally accurate, it is still far from perfect. The current system relies on ARKit and post closure optimization for loop closure, which may not always yield optimal results in complex indoor environments. On the other hand, our evaluation would also yield inaccurate results since the accuracy of registration between two trials of data collection is limited by the varying reconstruction results themselves. Regarding this issue, potentially we might want to explore the use of partial 3D affine transformation technique upon deformable parts such as corridors to achieve a better fit between two point clouds.

Improved 3D Registration. To address the limitations in 3D reconstruction, future work could explore the use of advanced registration techniques. For instance, partial 3D affine transformations of deformable parts could be used to achieve a better fit between the point clouds. This could potentially enhance the accuracy of the 3D models and, consequently, the localization performance.

Incorporation of Inertial Sensing. The current system updates the user's location at a rate of 1/2 Hz. To provide a smoother user experience, future implementations could incorporate inertial sensing data from the mobile device. This could allow for more frequent updates and smoother transitions between locations.

Kalman Filter for Smoothing. The user experience could potentially be improved by incorporating a Kalman filter for smoothing. The Kalman filter could help to reduce the noise in the signal data and provide more accurate location predictions.

User Interface Improvements. Future work could also focus on improving the user interface of the system. For instance, a 3D visualization of the indoor environment could be provided to help users better understand their location and surroundings. This could significantly enhance the user experience.

Robot Platform for Data Collection. To ease the burden of data collection, a robot platform could be incorporated into the system. The robot could be programmed to navigate through the indoor environment and collect data automatically, potentially during off-peak hours. This could allow for more frequent updates of the fingerprint database and reduce the need for manual data collection, which would further leverage the operational benefit of our system.

Comprehensive Data Analysis. As more data is collected, comprehensive data analysis could be conducted to gain insights into the system's performance and the characteristics of the indoor environment. This could provide valuable information for further improving the system.

8 Conclusion

In this paper, we presents DungeonAssistant, a *scalable* indoor localization system that harnesses the power of WiFi signal strength and LiDAR data to deliver precise and trustworthy location predictions. The operational advantage of DungeonAssistant lies in its scalability and cost-effectiveness, utilizing common devices such as iPads and Android phones for data gathering. We described the system's architecture, including the roles of the system maintainer and user, and detailed the fingerprinting approach and nearest neighbor localization method used for location prediction.

We implemented DungeonAssistant and conducted an evaluation in a complex indoor environment. The results demonstrated the system's robust performance, with a mean prediction error of around 4 meters. Furthermore, we discussed the limitations of the current system and proposed several directions for future work, including improved 3D registration, incorporation of inertial sensing, use of a Kalman filter for smoothing, user interface improvements, and the use of a robot platform for data collection.

In conclusion, DungeonAssistant represents a scalable yet promising solution for indoor localization. Its use of WiFi signals and LiDAR data, combined with a simple yet effective prediction algorithm, allows it to provide accurate location predictions in a cost-effective manner. While there are areas for improvement, we believe that DungeonAssistant provides a solid foundation for future research and development in the field of indoor localization. As we continue to refine and expand the system, we look forward to seeing its potential applications in a wide range of indoor environments.

References

[1] Android permissions. https://developer.android.com/reference/android/Manifest.permission. Accessed: 2023-12-15.

[2] Android wi-fi scanning overview. https://developer.android.com/develop/connectivity/wifi/wifi-scan. Accessed: 2023-12-15.

[3] P. Bahl and V.N. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, volume 2, pages 775–784 vol.2, 2000.

[4] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[5] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 2724–2729 vol.3, 1991.

[6] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N. Padmanabhan. Indoor localization without the pain. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, MobiCom '10, page 173–184, New York, NY, USA, 2010. Association for Computing Machinery.

[7] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[8] Matteo Cypriani, Frederic Lassabe, Philippe Canalda, and Francois Spies. Open wireless positioning system: A wi-fi-based indoor positioning system. In *2009 IEEE 70th Vehicular Technology Conference Fall*, pages 1–5, 2009.

[9] Apple Inc. Apple arkit documentation. https://developer.apple.com/documentation/arkit. Accessed: 2023-12-15.

[10] Apple Inc. Arcamera - arkit | apple developer documentation. https://developer.apple.com/documentation/arkit/arcamera/2866108-transform, 2023. Accessed: 2023-12-15.

[11] Yifei Jiang, Xin Pan, Kun Li, Qin Lv, Robert P. Dick, Michael Hannigan, and Li Shang. Ariel: Automatic wi-fi based room fingerprinting for indoor localization. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, page 441–450, New York, NY, USA, 2012. Association for Computing Machinery.

[12] Swarun Kumar, Stephanie Gil, Dina Katabi, and Daniela Rus. Accurate indoor localization with zero start-up cost. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, page 483–494, New York, NY, USA, 2014. Association for Computing Machinery.

[13] Yongsen Mao, Yiming Zhang, Hanxiao Jiang, Angel X Chang, and Manolis Savva. Multiscan: Scalable rgbd scanning for 3d environments with articulated objects. In *Advances in Neural Information Processing Systems*, 2022.

[14] Eladio Martin, Oriol Vinyals, Gerald Friedland, and Ruzena Bajcsy. Precise indoor localization using smart phones. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, page 787–790, New York, NY, USA, 2010. Association for Computing Machinery.

[15] Anindya S. Paul and Eric A. Wan. Rssi-based indoor localization and tracking using sigma-point kalman smoothers. *IEEE Journal of Selected Topics in Signal Processing*, 3(5):860–873, 2009.

[16] Deepak Vasisht, Swarun Kumar, and Dina Katabi. Decimeter-level localization with a single wifi access point. In *Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation*, NSDI'16, page 165–178, USA, 2016. USENIX Association.

[17] Sunkyu Woo, Seongsu Jeong, Esmond Mok, Linyuan Xia, Changsu Choi, Muwook Pyeon, and Joon Heo. Application of wifi-based indoor positioning system for labor tracking at construction sites: A case study in guangzhou mtr. *Automation in Construction*, 20(1):3–13, 2011. Global convergence in construction.

[18] Faheem Zafari, Athanasios Gkelias, and Kin K. Leung. A survey of indoor localization systems and technologies. *IEEE Communications Surveys Tutorials*, 21(3):2568–2599, 2019.

[19] Xiaopeng Zhao, Zhenlin An, Qingrui Pan, and Lei Yang. Nerf2: Neural radio-frequency radiance fields. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, ACM MobiCom '23. ACM, October 2023.

[20] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 766–782. Springer, 2016.

[21] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing, 2018.