Particle Filter-Based Indoor Localization with Learning Based PDR and Monocular Depth-Aided BIM Matching

Pravin Kumar Jaisawal[1,*], Youness Dehbi[2] and Harald Sternberg[1]

[1]Department of Hydrography and Geodesy, HafenCity University, Henning-Voscherau-Platz 1, 20457 Hamburg, Germany
[2]Computational Methods Lab, HafenCity University, Henning-Voscherau-Platz 1, 20457 Hamburg, Germany

Abstract

Modern smartphones offer several sensors that enable indoor pedestrian localization without the need for additional hardware. Pedestrian dead reckoning (PDR) provides a low-cost and efficient solution. However, it suffers from error accumulation and drift over time. Image-based localization methods can mitigate these limitations but are very computationally intensive to run at a higher frequencies. To address this, we propose a hybrid localization framework based on a particle filter, where a frequent, low-cost deep learning based inertial PDR is fused with a less frequent image-based updates to achieve accurate and robust localization. Our method does not require an offline mapping process and instead utilizes Building Information Models (BIM) generated maps. Furthermore, we incorporate recent monocular depth estimation models to generate depth directly from single images, thereby eliminating the need for continuous image streams to generate point clouds. Experimental results show that our proposed method can effectively track pedestrian poses using primarily smartphone sensors and BIM data.

Keywords

Indoor pedestrian positioning, Deep learning, Monocular depth estimation, BIM, Smartphone, ICP

1. Introduction

Indoor localization plays a crucial role in enabling location-aware applications for both pedestrians and robots in complex indoor environments such as airports, train stations, hospitals, and conference centers [1]. These capabilities become critical during emergencies and are also necessary for other applications such as facility monitoring, indoor gaming, and intelligent inventory management.

Modern smartphones are now equipped with multiple sensors, including an accelerometer, gyroscope, barometer, magnetometer, GNSS, camera, and sometimes even a LiDAR. This availability of sensors and widespread usage of smartphones provides a unique opportunity to deliver these services directly to the phone users without the need for additional equipment.

While GNSS is considered the de facto solution for outdoor positioning, it fails indoors due to signal attenuation, multipath effects, and obstructions. Various alternative indoor localization technologies such as ultra-wideband (UWB) [2], WiFi [3], 5G [4], magnetic systems [5], and acoustic systems [6] have been proposed. However, these solutions often require additional infrastructures, which can be costly to install, maintain, and scale, making them less desirable for many practical applications.

Localization technologies that are purely dependent on smartphones are favorable considering low cost and simplicity. Inertial localization, commonly referred to as Pedestrian Dead Reckoning (PDR), offers a low-cost and efficient approach, however, it typically suffers from drift and long-term error accumulation, requiring external corrections to maintain accuracy. The natural candidate for such corrections is smartphone LiDAR, however, its range is limited. For instance, the range of LiDAR in iPhone 16 Pro Max is about 10 m, but in practice we observe an effective range of only around 5 m. Moreover, the generated point cloud from this LiDAR is sensitive to reflective or transparent surfaces

IPIN-WCAL 2025: Workshop for Computing & Advanced Localization at the Fifteenth International Conference on Indoor Positioning and Indoor Navigation, September 15–18, 2025, Tampere, Finland
*Corresponding author.
pravin.jaisawal@hcu-hamburg.de (P. K. Jaisawal); youness.dehbi@hcu-hamburg.de (Y. Dehbi); harald.sternberg@hcu-hamburg.de (H. Sternberg)
0009-0004-3441-0793 (P. K. Jaisawal); 0000-0003-0133-4099 (Y. Dehbi); 0000-0002-1905-2287 (H. Sternberg)

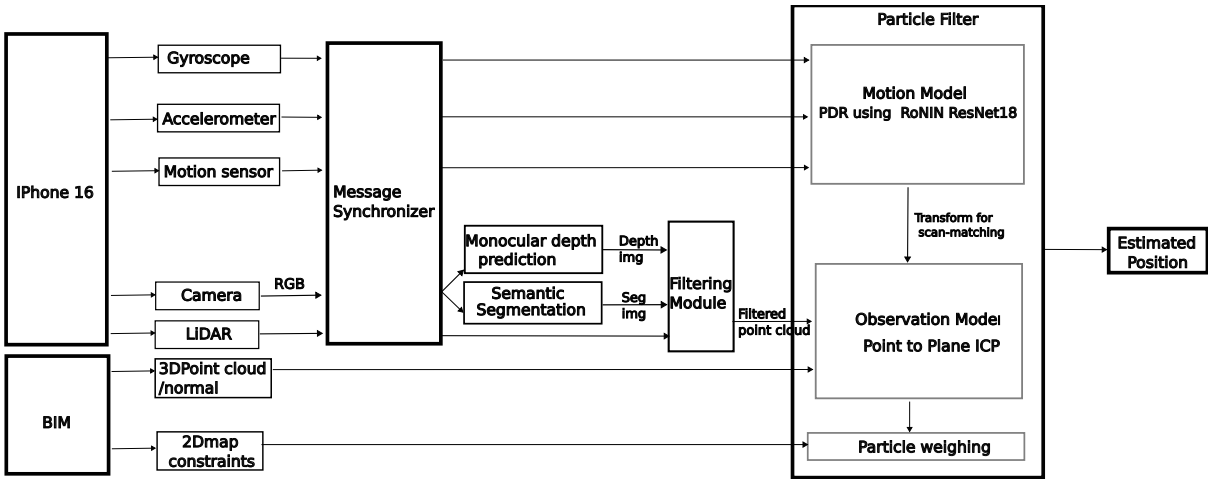


Figure 1: Our proposed workflow for indoor localization using smartphone data

(*e.g.* glass) and edges regions often suffer from noise which are quite common in indoor environment. These constraints reduce the availability of distinctive feature needed for reliable scan matching.

To overcome this limitations, we adopt an image-based approach using the recent advancements in monocular depth estimation (MDE). MDE enables depth prediction directly from a single monocular image by exploiting implicit depth cues, allowing depth estimation well beyond the range of smartphone LiDAR. This approach reduces the need for a continuous stream of camera images, as required in traditional methods such as SLAM or visual-inertial odometry (VIO). As a result, a simpler fusion model can be designed that only needs occasional image input. Moreover, in other vision-based localization approaches such as image retrieval systems, typically geo-referenced image database needs to be maintained. To create such a database, mapping becomes an essential offline task. We eliminate this requirement by utilizing Building Information Models (BIM). BIMs provide not only structural geometry but also semantic information, such as whether an element is a door, wall, slab, *etc.* Image-based corrections can be achieved by aligning the point cloud generated from MDE model directly with the point cloud derived from BIM.

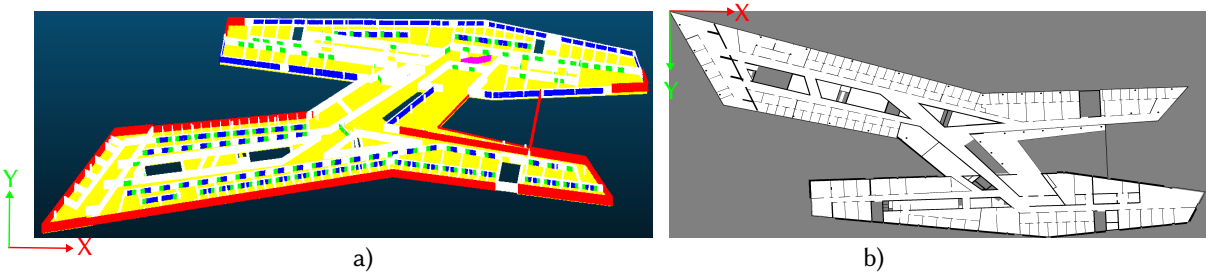
In this paper, we propose a particle filter based hybrid localization method that relies solely on smartphone sensors and BIM to provide robust indoor localization. Our approach does not require additional offline mapping prior to the online localization process. In each iteration of the particle filter, particles are propagated using a PDR-based motion model, followed by map matching and weight update based on image data. The visual correction step is performed by aligning the single image-based point cloud, obtained via monocular depth estimation, with the point cloud derived from BIM using Iterative Closest Point (ICP).

The main contributions of this paper are (1) a single-image-based point cloud generation method, leveraging MDE, that can be directly aligned with BIM-derived point clouds without requiring offline mapping, (2) a particle filter–based hybrid localization framework that fuses frequent, low-cost inertial measurements with occasional, computationally intensive image-based corrections to achieve robust indoor positioning using only smartphone sensors and BIM.

2. Related Works

Image-based localization methods can be broadly categorized into marker-based and markerless approaches. Marker-based localization relies on fiducial markers, such as AprilTag [7] or ArUco [8], placed throughout the indoor environment to track the camera's position. While these systems can provide accurate localization, their main drawbacks include the need for physical installation and maintenance of markers, which may be impractical or unwelcome in many buildings.

Markerless localization methods typically maintain a database of natural features or landmarks within



a) b)

Figure 2: a) 3D point cloud generated from the 4th floor, with semantic information, b) 2D raster map showing navigable (white), non-navigable (black) and unknown (gray) regions

the environment. These methods generally fall into two main categories: Simultaneous Localization and Mapping (SLAM) or Structure from Motion (SfM) based approaches, and image retrieval-based systems. In SLAM or SfM [9], a 3D map of the environment is first constructed. Subsequently, the camera's 3D pose can be estimated through stereo triangulation or feature matching against the map. Image retrieval approaches [10], in contrast, require a pre-collected database of geo-referenced images. Localization is then performed by identifying the most visually similar image from this database and inferring the camera pose accordingly. One limitation of both markerless approaches is the necessity of creating and maintaining detailed maps or image databases of the indoor environment, which can be resource-intensive and limits scalability. In contrast, our method leverages prior knowledge [11] (i.e. Building Information Model), which are often already available for many buildings, thereby eliminating the need for extensive pre-mapping or database maintenance.

In BIM based approaches, prior work mostly use LiDAR for aligning point cloud with maps generated from BIM [12, 13]. Unlike these approaches, we use solely smartphone camera to create dense point cloud for aligning with point cloud derived from BIM.

3. Methodology

Figure 1 illustrates the proposed framework for particle filter-based indoor localization using smartphone sensor data and BIM. The framework primarily relies on IMU data and integrates periodic (if available) position corrections from camera based observation model within particle filter. The 2D map, derived from BIM, is used for particle collision detection to remove particles that violates structural constraints. All elements of the workflow are discussed briefly in following subsections.

3.1. Data Acquisition

We use iPhone 16 Pro Max to collect data on the fourth floor of our university building. The *SensorLog* [14] application was used to record accelerometer, gyroscope, motion sensor, and barometer data at 100 Hz, while another application *Record3D* [15] was used to record RGB images, depth images (based on LiDAR), depth confidence maps, and camera intrinsics at 6 Hz. The smartphone is connected to a PC/laptop via a USB-C cable, allowing sensor data transmission over TCP. We chose to record camera data at 6 Hz as a trade-off between synchronization and data efficiency. Since our downstream operations require only one image per second, capturing camera data at 6 Hz allows, however, enough alignment with IMU data. Additionally, given the slow motion of pedestrians, visual change at this pace are not drastic, ensuring no significant loss of detail.

3.2. BIM to Map Generation

We require a target 3D point cloud for scan matching using the ICP algorithm and a 2D map for detecting wall collision during the particle filter propagation step. Both 2D and 3D maps are generated from BIM directly using the *IfcOpenShell* library, without relying on any external software. *IfcOpenShell* provides access to both structural and semantic information of each BIM component. For 3D map generation,

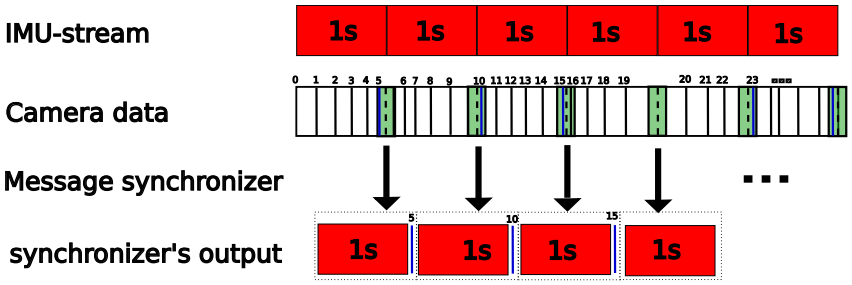


Figure 3: Message synchronization: the green box represents the permissible time interval, and the blue line in the camera data indicates the data stream selected for the fusion process.

components are first floor-wise separated. We then generate point cloud by applying Poisson Disk sampling on faces of each component, with density of 100 $points/m^2$. For large components such as slabs (in BIM, slabs refer to floor or roof elements), sometimes computationally prohibitive dense point cloud could be generated. In such cases, we tile these components into smaller, manageable regions and generate point cloud for each tile individually. 3D map with semantic information for the fourth floor is shown in Figure 2 a).

For 2D map generation, we slice the building model and project the geometry within the slice to represent non-navigable regions. Since door locations are known from BIM, we exclude their projection from the map to account for possible passages during particle propagation. For navigable areas, we use the projection of the top surface of floor slabs. The resulting 2D raster map encodes this information with distinct colors: white represents movable regions, black indicates non-movable regions, and gray denotes unknown areas, as can be in Figure 2 b).

3.3. Message Synchronization

Message synchronization is an important step for performing accurate fusion operations. In this work, we adopt the synchronization policy illustrated in Figure 3. A key parameter in this process is the permissible time interval, which is used to align the data streams. Synchronizer accepts 1 second segments of IMU data and individual camera frames. For each IMU segment, the last timestamp is used to search for first camera frame that falls within the permissible interval around that timestamp. The matched camera frame may occur slightly before or after the last IMU's last timestamp, as long as it lies within the defined interval. If multiple camera frames are found, the earliest one is selected. If no camera frame lies within the permissible interval, only the IMU data is used for further processing.

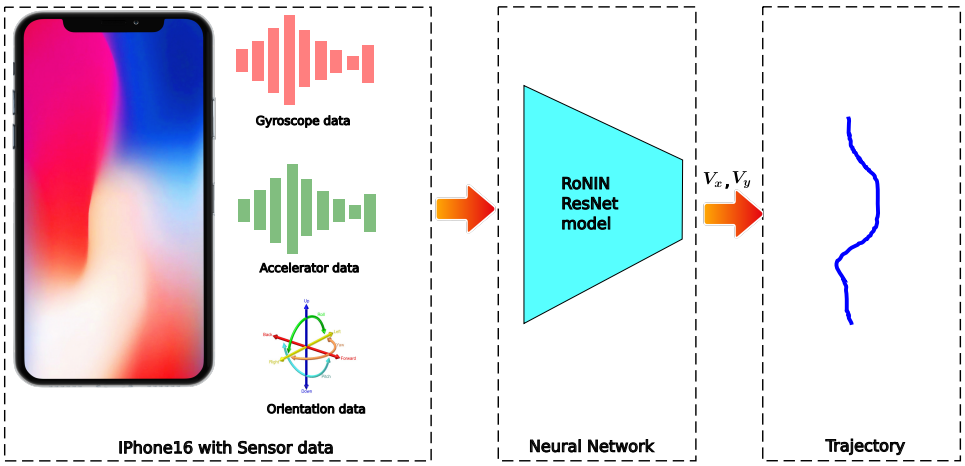


Figure 4: Inertial localization using RoNIN ResNet18 model

3.4. Neural Network Based Pedestrian Dead Reckoning (PDR)

We employ a pre-trained ResNet-based neural architecture from the RoNIN framework [16] for inertial navigation. The network accepts a 200×6 input tensor comprising of 3-axis gyroscope and 3-axis accelerometer data, sampled at 200 Hz, to produce a planar velocity vector (V_x, V_y), as shown in Figure 4. Since the raw data are recorded at 100 Hz, IMU and motion sensor data are upsampled to 200 Hz using 1D interpolation to match the required frequency. The upsampled smartphone data also needs to be transformed into a heading-agnostic coordinate frame (HACF), in which the Z-axis is aligned with gravity, using the orientation (attitude) data from the motion sensor. The network uses this prepared sequence to predict a velocity vector, which represents displacements over each one-second interval.

3.5. Point Cloud Generation from Camera Image

3.5.1. Neural Network Based Monocular Depth Estimation

For monocular depth estimation, we use the pre-trained UniK3D [17] model with the large variant of Vision transformer (ViT-L) [18] as its backbone. This model was chosen due to its superior results in comparison to prior methods and its ability to integrate camera intrinsic directly, enabling the prediction of more precise metric depth image, 3D point cloud, along with associated confidence values, all without the need for domain-specific tuning.

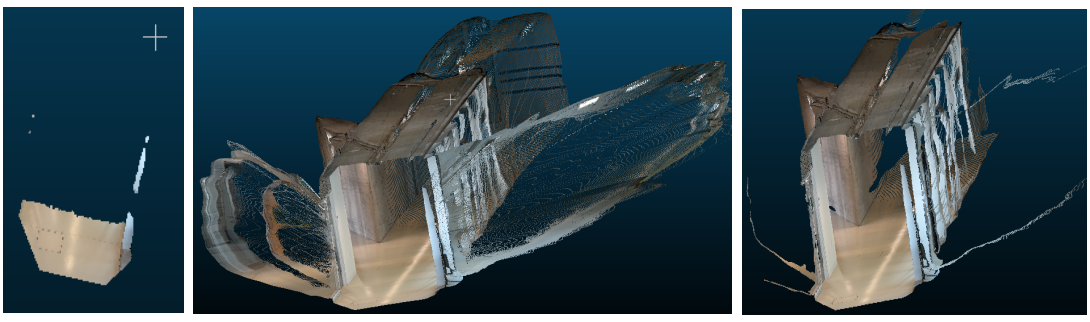
Although the model claims to predict metric depth, we observed a scale mismatch between predicted depth and real world measurements. This discrepancy is likely due to domain shift, in other words, the test data is different from the original training data. To address this issue, we use the partial depth data from the iPhone's LiDAR sensor to correct the scale. The scale factor is estimated using Equation 1. Once the scale is determined, the predicted depth can be corrected by simply multiplying it with this scale as shown in Equation 2.

$$\text{Scale} = \frac{\sum_{i=1}^{n} \mathrm{d}_i^{\text{LiDAR}}}{\sum_{i=1}^{n} \mathrm{d}_i^{\text{Predicted}}} \tag{1}$$

$$\text{Corrected Depth} = \text{Scale} \times d^{\text{Predicted}} \tag{2}$$

where d_i represents depth at pixel i and n represents the number of pixels where LiDAR depth is available.

Figure 5 b) shows an example illustrating the increase in depth range by using MDE approach compared to original iPhone LiDAR (see Figure 5 a)). However, the scale-corrected output still contains inaccuracies due to reflections on glass surfaces, which are quite common in indoor environments. To mitigate this, we use the predicted confidence map, where the confidence value represents the predicted error in the log scale. To exclude points with unreliable depth, we apply a thresholding approach. The threshold is determined by selecting an average value such that more than 80% of the pixels are retained. The final filtered depth is shown in Figure 5 c).



a) b) c)

Figure 5: a) Snapshot of point cloud from iPhone's LiDAR, b) Predicted depth after scale correction, c) Predicted depth after scale correction and filtering

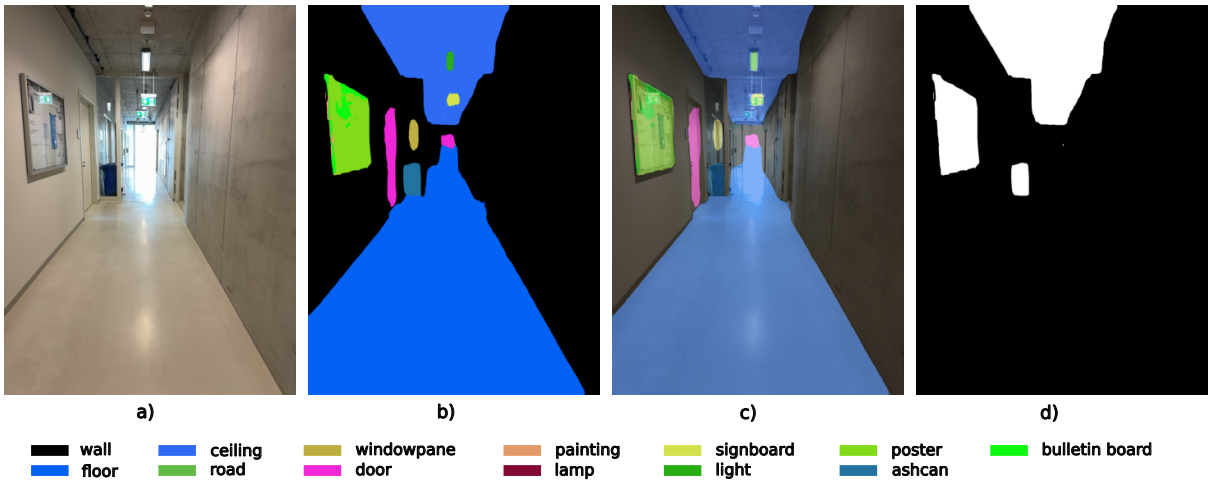


Figure 6: Semantic segmentation on RGB data. a) Input RGB image, b) Predicted Segmentation labels, c) Predicted label overlayed on RGB image, d) Binary mask showing pixels to be filtered

3.5.2. Semantic Segmentation

We use the SegFormer [19] network, pre-trained on the ADE20K [20] dataset, for semantic segmentation. ADE20K contains 150 labels, among which several correspond to BIM-related components such as walls, floors/roofs (slabs element), windows, doors, and stairs. The pre-trained model effectively detects these BIM related elements. All non-BIM components, *e.g.* furniture, kitchen, lamps, people, *etc.*, are used to filter out irrelevant regions from scale-corrected point cloud generated from the monocular depth estimation network. Additionally, we remove roof points, as our BIM-derived point cloud does not include roof points. Figure 6 shows an example of semantic segmentation on RGB image. Figure 6 d) represents the binary mask that will be used in filtering module to exclude non-BIM components.

3.6. ICP based Point Cloud Registration

We perform point cloud registration using point-to-plane ICP algorithm [21]. The filtered source point cloud from the camera image (see Section 3.5) is first downsampled with a voxel size of 0.1 to match the spatial resolution of target point cloud, which is derived from the BIM. To reduce the computational cost, the target point cloud is cropped using bounding box of ± 15 meter around initial estimate. Registration is then carried out with maximum correspondence distance of 0.2 (twice the voxel size). Larger values often lead to incorrect alignment due of similar parallel structures in our indoor environment. To ensure reliability of the registration, we accept the ICP results only if there is at least 60% overlap between source and target point cloud and the inlier root-mean-square-error (RMSE) is below 0.2 m.

3.7. Map Constraints

We employ a Signed Distance Field (SDF)-based approach to detect collisions between particles and non-navigable regions in the map. SDF value at each pixel encodes the distance to nearest obstacle, providing continuous representation of free space. During the propagation step of particle filter, a particle is allowed to move if its motion norm is smaller than SDF value at its current location. If not, SDF-guided ray marching is performed along the motion vector to account for edge cases such as motion parallel to wall, where proximity alone does not imply a collision. To limit computational overhead, ray marching is invoked only when a particle's motion norm is larger than the local SDF value, uses adaptive step sizes derived from the SDF to avoid redundant checks, and terminates early upon detecting a collision. Given the small displacements typical in pedestrian motion, our method takes negligible runtime cost relative to overall particle filter update. If a collision is detected, the corresponding particle is discarded.

3.8. Particle Filter

Initializing particle is a crucial step, as it influences the overall localization performance. Since, PDR only provides relative motion estimates from an initial state, any error in initialization is propagated throughout the trajectory. Furthermore, ICP-based observation model is sensitive to initialization, so large discrepancy between true state and initial guessed state can lead to registration failure. A good guess could be found using the ICP algorithm based on first synchronized camera image and course manual guess. The resulting ICP metrics (*e.g.* fitness score) could be used to detect poor initialization. Once a reliable initial pose (x_0, y_0, θ_0) is found, particles are sampled using Gaussian distribution around it, with equal initial weights.

Message synchronizer module provides inputs to particle filter for updating positions. During the motion update step, PDR outputs planar velocity estimate (V_x, V_y). Since each IMU stream corresponds to $1s$ of data, this planar velocity also represents displacements (dx, dy) along X-axis and Y-axis. The change in orientation $d\theta$ is computed from angular difference between the last predicted and current heading. These values (dx, dy, $d\theta$) are used to propagate the particles. Before updating each particle, we check if it collide with walls using map constraints (see Section 3.7). A prior state $(x_{prior}, y_{prior}, \theta_{prior})$ is then computed based on the distribution of active particles and is used to initialize ICP transformation.

If a synchronized camera image is available, we perform an observation update. The 2D prior estimate is converted to 3D pose by setting $Z = 1.5$ (approximate phone location in meter) and zero pitch and roll. While user height and posture can vary over time, these values only corresponds to initial transform for ICP computation. In principle, Z-value from last ICP iteration could be reused, but a fixed value was chosen for simplicity without degrading performance. Using this initial 3D transformation, ICP is performed between the filtered source and target point clouds. If registration results lie within acceptable limit, particle weights are updated based on their distance to estimated ICP-based pose. The updated weight are then used to estimate the current position from the particle filter $(x_{est}, y_{est}, \theta_{est})$ The particles are then resampled, if necessary. If no synchronized camera image is available or ICP results are not within acceptable limit, localization proceeds using only the motion update. The motion and observation steps are repeated until new data arrives from the message synchronizer.

4. Experiments and results

4.1. Set-up

The data sequence was collected using an iPhone 16 pro Max on the fourth floor of HafenCity University building. The total duration of data sequence is 468 seconds, covering a travel distance of over 450 meters. As our proposed workflow incorporates few neural network-based components, localization computations were conducted on a moderately powerful laptop equipped with an NVIDIA 6G RTX 3060 GPU and 16 GB RAM memory.

4.2. Localization Results

Figure 7 shows the estimated pedestrian trajectories. It is evident that using RoNIN-based PDR without map correction (marked in blue color) leads to large error. Integrating map-based constraints to PDR significantly reduces errors, yet when two passages are in close proximity on the map, this can lead to incorrect choice of path as highlighted in Figure 7. Furthermore, while PDR with map matching mitigates some drift, it stills suffers from long-term error accumulation due to the absence of map based absolute corrections. Such problems can be mitigated by integrating image-based localization, in our case, using ICP to align point cloud generated from camera image with the 3D point cloud derived from BIM. This enhanced trajectory is referred to as particle filter estimate in the figure, and it appears to correct estimate from PDR with map correction.

During our experiment, we noticed that the ICP in indoor environments is sensitive to discrepancies between the BIM (as-designed model) and the real environment (as-is model). This inconsistency causes ICP to fail or produce result outside our acceptable range. As a result, the particle filter only receives

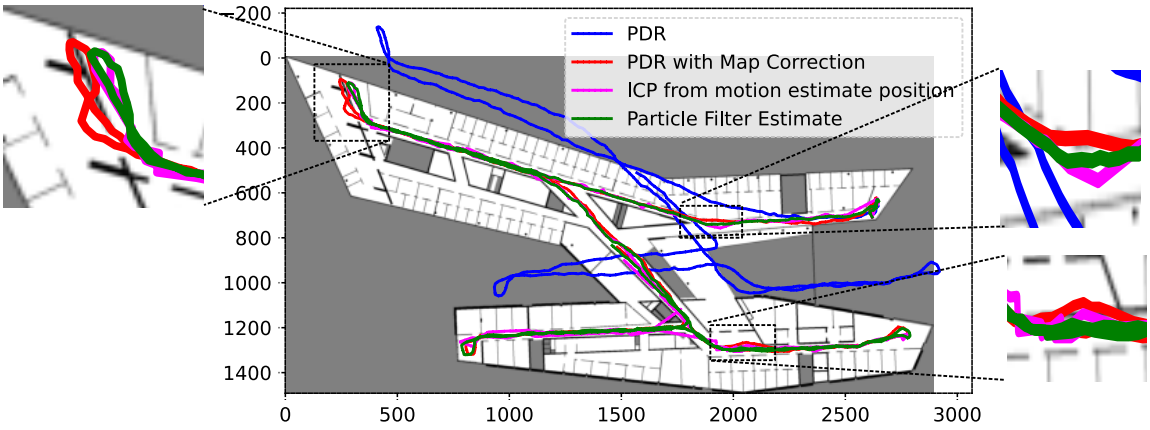


Figure 7: Predicted pedestrian trajectories overlayed on 2D map (resolution = 20 pixels/meter). Highlighted regions show localization errors when only PDR with map correction (in red color) is used and better particle filter estimate (in green color) when ICP is integrated.

occasional corrections. This behaviour can be sometimes observed on the map, where ICP-based positions (marked in pink color) sometimes jumps across walls. Nevertheless, our system remains robust and delivers reliable position.

5. Conclusions

In this paper, we proposed a particle filter based hybrid localization framework, in which more frequent, low cost inertial measurements are fused with less frequent but computationally intensive image-based correction updates. By leveraging only smartphone sensors, BIM data, and monocular depth estimation, we eliminate the need for external infrastructure, offline mapping step, and continuous image sequence.

While we did not conduct quantitative evaluation due to the lack of ground-truth data, observed qualitative results suggest that our method achieves robust and drift-free localization in a complex indoor environment. The current implementation is not yet optimized for smartphone deployment due to the use of large models and high resolution images, future work will address this limitation within an optimization process. We also plan to perform quantitative evaluation using UWB-based ground-truth data and integrate barometer readings to enable accurate indoor localization in multi-floor settings.

Acknowledgments

The project is supported by the Federal Ministry of Transport and Digital Infrastructure (BMVI), grant number 19OI22008A. The authors would also like to thank Hossein Shoushtari, Son Nguyen and Georg Fjodorow for their valuable discussion and insightful feedback throughout the course of this work.

Declaration on Generative AI

During the preparation of this work, the author used generative AI tools in order to: Grammar and spelling check as well as Paraphrase and reword. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] O. Czogalla, S. Naumann, Pedestrian indoor navigation for complex public facilities, in: 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2016, pp. 1–8. doi:10.1109/IPIN.2016.7743672.

[2] B. Van Herbruggen, B. Jooris, J. Rossey, M. Ridolfi, N. Macoir, Q. Van den Brande, S. Lemey, E. De Poorter, Wi-pos: A low-cost, open source ultra-wideband (uwb) hardware platform with long range sub-ghz backbone, Sensors 19 (2019) 1548.
[3] X. Cao, Y. Zhuang, X. Yang, X. Sun, X. Wang, A universal wi-fi fingerprint localization method based on machine learning and sample differences, Satellite Navigation 2 (2021) 1–15.
[4] X. Zhou, L. Chen, Y. Ruan, R. Chen, Indoor localization with multi-beam of 5g new radio signals, IEEE Transactions on Wireless Communications 23 (2024) 11260–11275. doi:10.1109/TWC.2024.3380737.
[5] K. Shao, Z. Li, M. Shu, Q. Guo, Q. Wu, Smartphone-based multi-mode geomagnetic matching/pdr integrated indoor positioning, in: 2023 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN), IEEE, 2023, pp. 1–8.
[6] S. P. Tarzia, P. A. Dinda, R. P. Dick, G. Memik, Indoor localization without infrastructure using the acoustic background spectrum, in: Proceedings of the 9th international conference on Mobile systems, applications, and services, 2011, pp. 155–168.
[7] E. Olson, Apriltag: A robust and flexible visual fiducial system, in: 2011 IEEE international conference on robotics and automation, IEEE, 2011, pp. 3400–3407.
[8] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, M. J. Marín-Jiménez, Automatic generation and detection of highly reliable fiducial markers under occlusion, Pattern Recognition 47 (2014) 2280–2292.
[9] E. Royer, M. Lhuillier, M. Dhome, J.-M. Lavest, Monocular vision for mobile robot localization and autonomous navigation, International Journal of Computer Vision 74 (2007) 237–260.
[10] T. Sattler, T. Weyand, B. Leibe, L. Kobbelt, Image retrieval for image-based localization revisited., in: BMVC, volume 1, 2012, p. 4.
[11] L. Lucks, L. Klingbeil, L. Plümer, Y. Dehbi, Improving trajectory estimation using 3d city models and kinematic point clouds, Transactions in GIS 25 (2021) 238–260.
[12] H. Yin, Z. Lin, J. K. Yeoh, Semantic localization on bim-generated maps using a 3d lidar sensor, Automation in Construction 146 (2023) 104641.
[13] R. Hendrikx, P. Pauwels, E. Torta, H. P. Bruyninckx, M. Van De Molengraft, Connecting semantic building information models and robotics: An application to 2d lidar-based localization, in: 2021 IEEE international conference on robotics and automation (ICRA), IEEE, 2021, pp. 11654–11660.
[14] B. Thomas, Sensorlog version 6.1, 2024. URL: https://apps.apple.com/de/app/sensorlog/id388014573.
[15] M. Simonik, Record3d — 3d videos version 1.10.5, 2025. URL: https://apps.apple.com/us/app/record3d-3d-videos/id1477716895?ls=1.
[16] S. Herath, H. Yan, Y. Furukawa, Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods, in: 2020 IEEE international conference on robotics and automation (ICRA), IEEE, 2020, pp. 3146–3152.
[17] L. Piccinelli, C. Sakaridis, M. Segu, Y.-H. Yang, S. Li, W. Abbeloos, L. Van Gool, UniK3D: Universal camera monocular 3d estimation, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2025.
[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, ICLR (2021).
[19] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, P. Luo, Segformer: Simple and efficient design for semantic segmentation with transformers, Advances in neural information processing systems 34 (2021) 12077–12090.
[20] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, A. Torralba, Scene parsing through ade20k dataset, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 633–641.
[21] Y. Chen, G. Medioni, Object modelling by registration of multiple range images, Image and vision computing 10 (1992) 145–155.