Map-Based Sensor Fusion for Indoor Localization

Rinkal Zadafiya Adarsh Koshiya

OTH Amberg-Weiden | Master Artificial Intelligence

Abstract

Accurate indoor localization is challenging because GPS is unavailable and wireless signals are strongly affected by multipath and attenuation. This project investigates a **map-based particle filter** for pedestrian localization that fuses inertial measurements (IMU), step detection, and BLE beacon RSSI on top of a known floor plan. A Monte Carlo localization framework tracks a cloud of weighted particles that represent possible user positions and headings; at each step, particles are propagated using a motion model and reweighted using beacon signal strength. A floor plan is used to eliminate particles that cross walls or leave walkable areas, which significantly improves robustness in cluttered indoor environments. The system is implemented in Python, evaluated on recorded sensor data, and visualized using Matplotlib and Pygame; all filter steps are exported as plots and videos for qualitative analysis.

Motivation

Indoor positioning enables applications such as **wayfinding in complex buildings**, asset tracking, and context-aware mobile services, but remains difficult to solve reliably. Pure IMU-based dead reckoning accumulates drift, while BLE-based localization alone suffers from noisy RSSI and environmental changes. Map-unaware approaches may easily produce trajectories that walk through walls or enter forbidden regions. This project explores whether **combining IMU, step detection, and BLE RSSI in a map-aware particle filter** can yield accurate and robust pedestrian localization indoors, using only commodity sensors and a pre-defined floor plan.

System Overview and Data

The overall system integrates sensor data collection, particle filter localization, and rich visualization.

- **Sensors and Hardware:** Smartphone / wearable IMU (accelerometer, gyroscope, magnetometer) and BLE beacons deployed in the building.
- **Step Detection:** Peaks in accelerometer magnitude are used to detect steps and estimate step length, providing discrete motion events.
- **BLE RSSI:** Each beacon periodically broadcasts; the received signal strength (RSSI) is logged and later used as a noisy proxy for distance.
- **Recorded Dataset:** Raw sensor measurements and reference trajectory are stored in `Data.csv`, collected using Arduino and mobile devices.
- **Software Environment:** All algorithms, experiments, and plots are implemented in the Jupyter notebook `main.ipynb`.

Method – Particle Filter Architecture

Indoor localization is formulated as a **state estimation problem** solved with a Monte Carlo (particle) filter. **State and Models:**

- **State:** Each particle encodes the pedestrian's position (and optionally heading) on the 2D floor plan.
- **Motion Model:** Upon each detected step, particles are propagated along their heading with sampled step length and orientation noise to capture uncertainty.
- **Measurement Model:** For each beacon, expected RSSI is approximated from the distance to the particle. The difference between expected and measured RSSI defines the likelihood used to update particle weights.
- **Resampling:** Systematic or multinomial resampling reduces particle degeneracy by focusing computation on high-weight particles.

The estimated trajectory is obtained from the weighted mean or maximum a posteriori particle at each time step.

Map Constraints and Particle "Killing"

A central idea of this project is to integrate a **floor plan** directly into the filter.

- The environment is represented as a binary occupancy map distinguishing **walkable** vs. **non-walkable** cells (walls, obstacles, outside regions).
- After each motion update, particles that move into non-walkable cells are **invalidated ("killed")** and removed from the particle set.
- This constrains the hypothesis space and prevents trajectories that cut through walls or leave the building outline.
- Both **map-less** and **map-based** modes are implemented, enabling direct comparison of the impact of map constraints on localization quality.

Implementation and Software Structure

The full workflow is implemented and documented in `main.ipynb`.

- **Data Handling:** `Data.csv` is loaded, synchronized, and preprocessed; sensor streams are aligned to a common timeline.
- **Filter Core:** Modular functions implement initialization, motion update, measurement update, weight normalization, resampling, and map-based particle killing.
- **Visualization:** Matplotlib and Pygame render the floor plan, ground-truth trajectory, estimated path, and particle cloud for each time step.
- **Output Organization:** Plots, animation frames, and videos are written to the `Graph and Video/` directory for later inspection.
- **Reproducibility:** All parameters (number of particles, noise levels, map resolution, beacon layout) are documented in the notebook for repeatable experiments.

Experiments and Results

The project evaluates how map constraints and sensor fusion affect trajectory accuracy and stability.

- **Map-less vs. Map-based:** In the map-less case, the filter can roughly follow the path but may drift substantially and traverse walls. With map constraints, particle spread is reduced and the estimated trajectory stays in valid corridors.
- **Effect of BLE RSSI:** Including RSSI measurements helps to disambiguate similar-looking areas (e.g., long corridors, symmetric layouts), especially when heading estimates from the IMU are noisy.
- **Qualitative Visualizations:** Videos and frame sequences show how incorrect hypotheses are pruned as particles hit walls, and how the cloud converges towards the true path.
- **Overall Observation:** Map-based sensor fusion yields **more realistic, smoother, and physically consistent** trajectories than either IMU-only dead reckoning or map-less filtering.

Limitations and Future Work

- **RSSI Variability:** BLE measurements are highly noisy and environment-dependent; more sophisticated propagation or fingerprinting models could further improve robustness.
- **User-Dependent Step Length:** A fixed or simple distribution for step length may not generalize across users, walking speeds, or phone placements.
- **Map Fidelity:** Localization quality depends on the accuracy and alignment of the floor plan; automatic calibration or SLAM-based refinement is a promising direction.
- **Real-Time Deployment:** The current implementation processes recorded data; porting the algorithm to run in real-time on mobile devices is a natural next step.

Conclusion

This project demonstrates that a **map-aware particle filter** combining IMU, step detection, and BLE RSSI is a viable approach to indoor pedestrian localization. By explicitly enforcing floor-plan constraints and fusing heterogeneous sensor modalities, the system reduces drift and avoids physically impossible trajectories. The Python implementation and visualizations provide a flexible foundation for further research on robust indoor positioning in real-world settings.