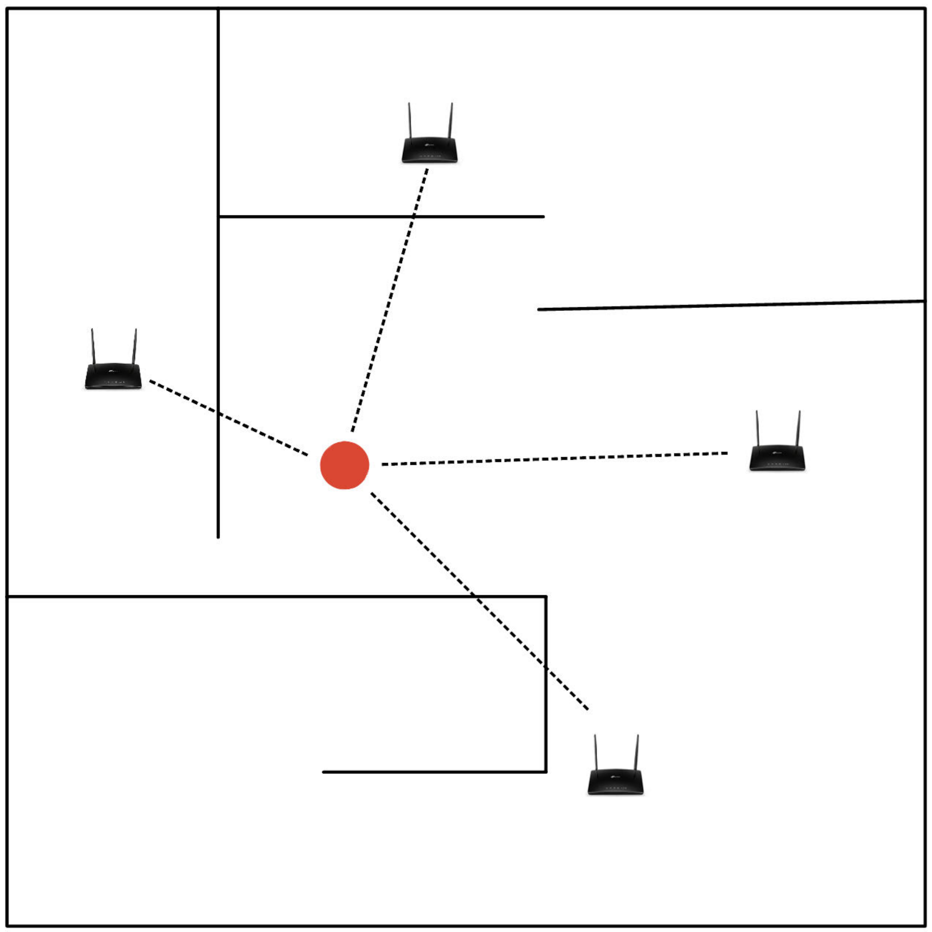
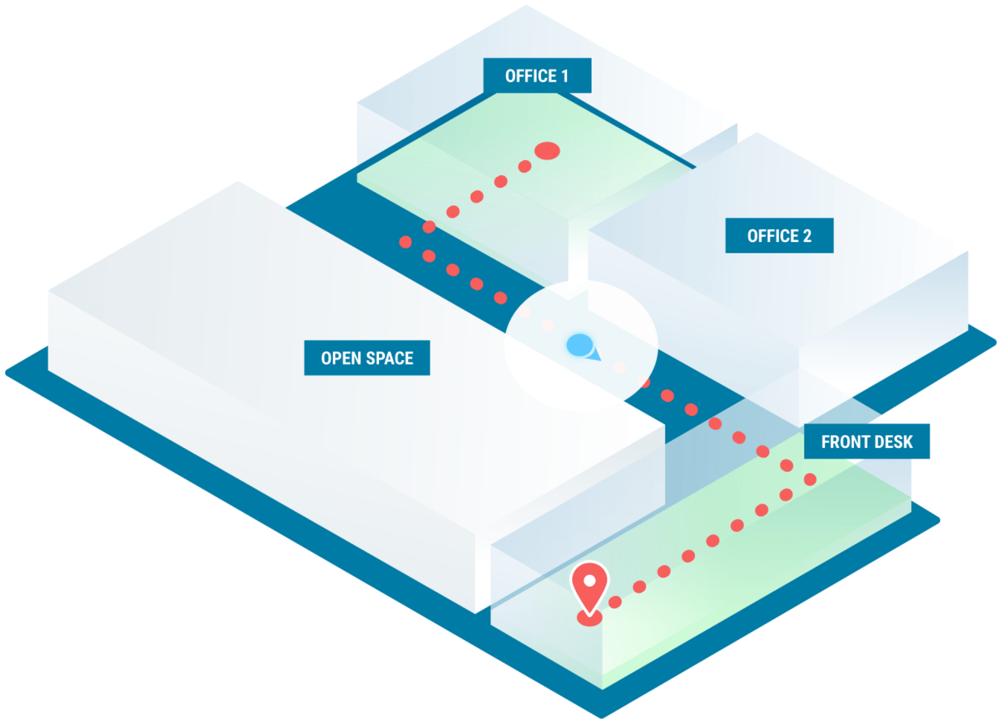
WiFi Assisted Loop Closure Detection

Students:
Shashank Dammalapati
Faris Hajdarpasic

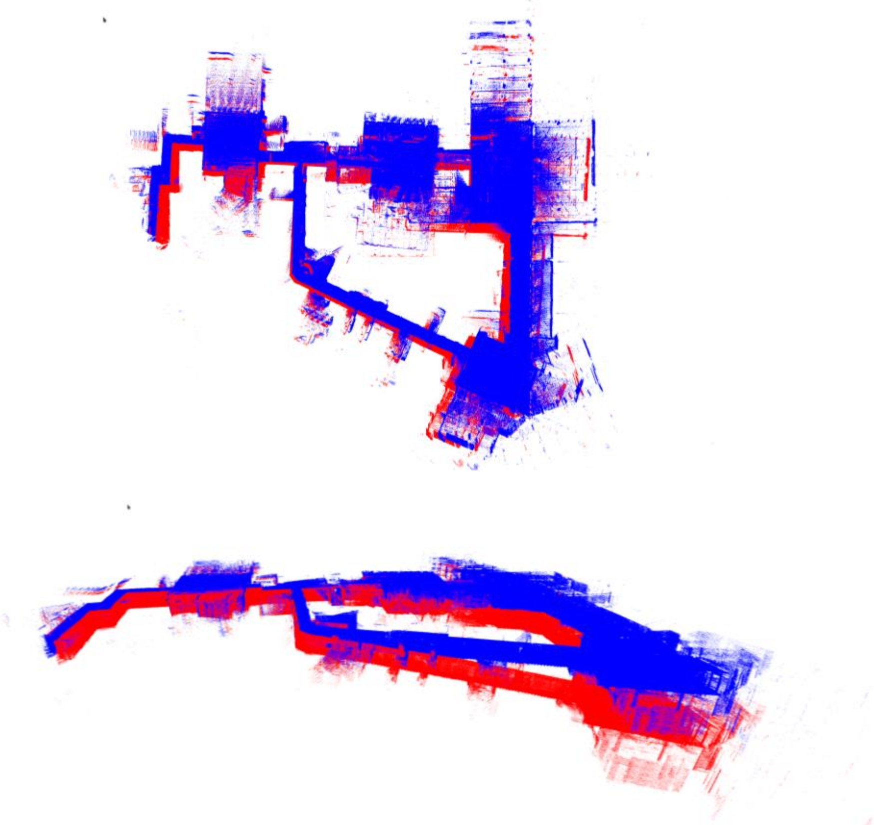
Supervisor:
Jan Quenzel



Introduction

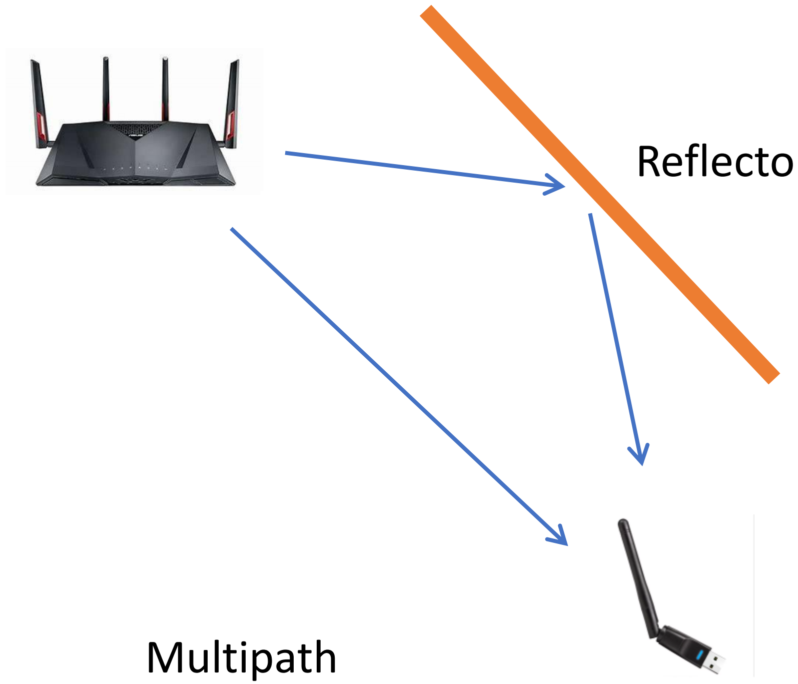


Indoor environments are GPS denied environments

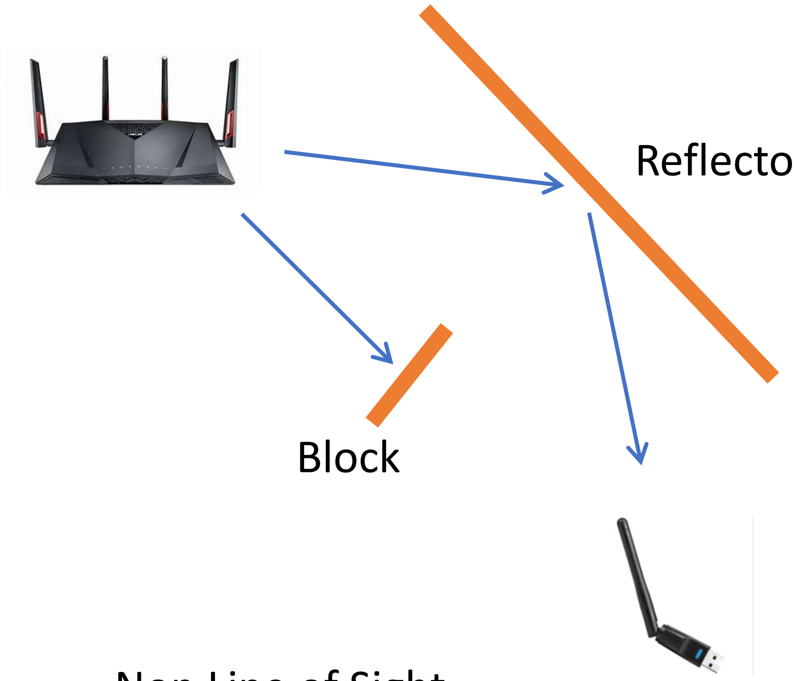


Loop Closure

Challenges



Multipath



Non Line of Sight

CLINS

CLINS: Continuous-Time Trajectory Estimation for LiDAR-Inertial System

Jiajun Lv[1,†], Kewei Hu[1,†], Jinhong Xu[1], Yong Liu[1], Xiushui Ma[2], Xingxing Zuo[1]

***Abstract*— In this paper, we propose a highly accurate continuous-time trajectory estimation framework dedicated to SLAM (Simultaneous Localization and Mapping) applications, which enables fuse high-frequency and asynchronous sensor data effectively. We apply the proposed framework in a 3D LiDAR-inertial system for evaluations. The proposed method adopts a non-rigid registration method for continuous-time trajectory estimation and simultaneously removing the motion distortion in LiDAR scans. Additionally, we propose a two-state continuous-time trajectory correction method to efficiently and efficiently tackle the computationally-intractable global optimization problem when loop closure happens. We examine the accuracy of the proposed approach on several publicly available datasets and the data we collected. The experimental results indicate that the proposed method outperforms the discrete-time methods regarding accuracy especially when aggressive motion occurs. Furthermore, we open source our code at https://github.com/APRIL-ZJU/clins to benefit research community.**

I. Introduction

Multi-sensor fusion plays an essential role in simultaneous localization and mapping (SLAM) algorithms with its complementarity and robustness, and it has been widely deployed in autonomous navigation, scene reconstruction, mixed reality, etc. In this paper, we study the high-accuracy continuous-time trajectory estimation with a fusion of LiDAR and IMU measurements. Most existing methods process LiDAR and IMU measurements in a discrete-time fashion. The LiDAR scan collecting points when the laser heads rotate around a mechanical axis, thus motion distortion is unavoidable when the LiDAR does not keep static in the data collecting process. Discrete-time based methods undistort the LiDAR points into the start time instant of the LiDAR scan by interpolations. IMU measurements are interpolated and integrated to formulate relative poses constraints between discrete LiDAR scans. Discrete-time based methods have several inherent limitations that hurt the estimation. Firstly, in practice, different sensors do not get measurements at the same frequency, let alone the same time instants. Interpolations has to be employed to fuse measurements from different sensors, which introduces non-negligible errors. Secondly, it is unlikely to leverage the raw measurements directly, i.e. the raw LiDAR points and raw IMU measurements. Raw LiDAR points are undistorted into the specific time instants to constitute LiDAR scans, while IMU measurements are assembled to get integrated relative pose measurements. These phenomenons

[1] April Lab, Zhejiang University, Hangzhou, China. (Yong Liu is the corresponding author, email: yongliu@iipc.zju.edu.cn)
[2] NingboTech University, Ningbo, China.
† Jiajun Lv and Kewei Hu contribute equally to this work.

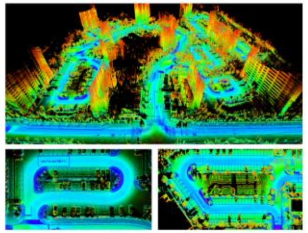


Fig. 1: Dense 3D reconstruction of Kaist-Urban-07 dataset by simply assembling 2D LiDAR scans from SICK LMS-511 with the estimated continuous-time trajectory from CLINS. The global trajectory is estimated with 3D LiDAR, Velodyne VLP-16 and Xsens IMU, MTi-300. 2D LiDAR scans is accumulated for reconstruction due to its high density.

result from the intractable super-high-frequency raw sensor measurements, which requires huge amount of pose variable to be estimated if they are utilized in a direct way. The above-mentioned difficulties can be summarized as the discrete pose representation fails to meet the system's demand of high temporal resolution. Recently, the continuous-time based method, which models the trajectory as a function of time and supports querying poses at any timestamp that naturally solves the problem of integrating asynchronous and high-frequency data. With those sound properties, continuous-time based method has been applied to many areas, such as visual-inertial navigation system [1], [2], event camera [3], rolling-shutter camera [1], actuated LiDAR [4], intrinsic and extrinsic calibration between sensors [5], [6].

This paper proposes a complete continuous-time trajectory estimation framework for LiDAR-inertial system. We summarize the contributions as follows:

- We propose a continuous-time trajectory estimator, which now supports the fusion of 3D LiDAR points and inertial data, and it is easy to expand and fuse data from other asynchronous sensors at arbitrary frequencies.
- We propose a two-stage continuous-time trajectory correction method to efficiently and effectively tackle loop closures.
- The proposed approach is extensively evaluated on

CLINS fuses LiDAR and IMU measurements to produce high-accuracy continuous time trajectory

J. Lv, K. Hu, J. Xu, Y. Liu, X. Ma and X. Zuo, "CLINS: Continuous-Time Trajectory Estimation for LiDAR-Inertial System," 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems

WiFi Reception Tools

- We are trying to get WiFi data for each timestamp
- Tools that we tried for gathering data:
  - iwlist
  - nmcli
  - airodump-ng

Fingerprint similarity

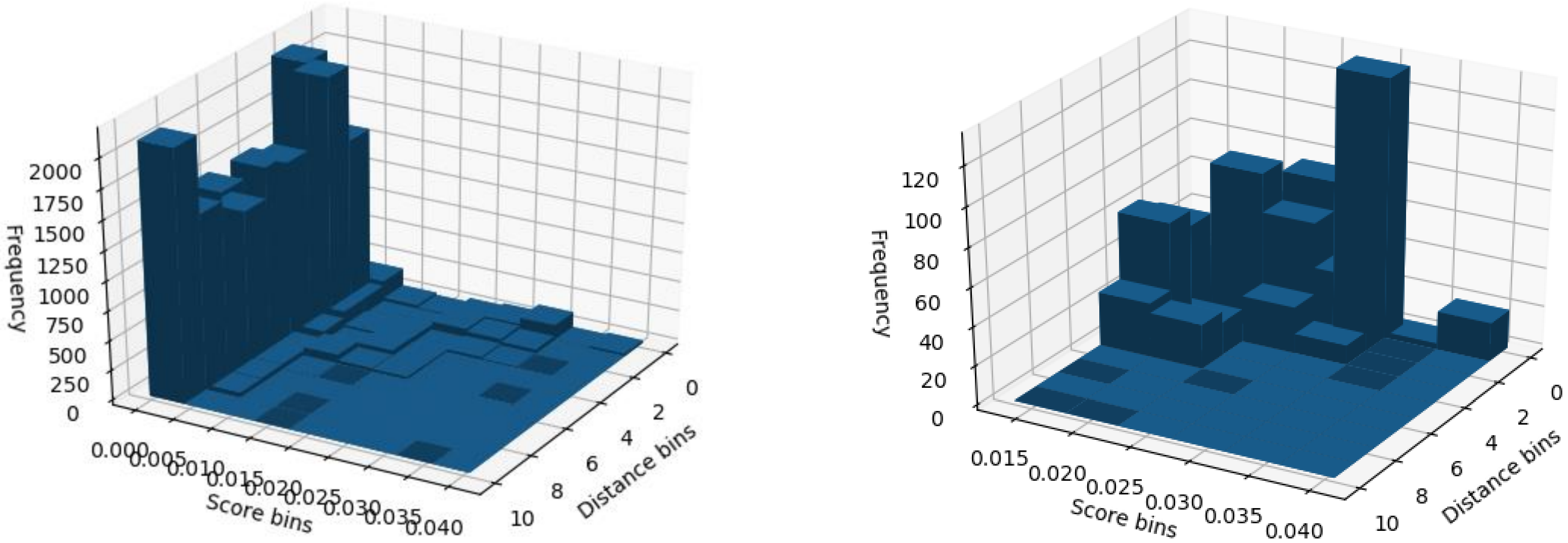
- Each node x_t has a fingerprint assigned to it
- Fingerprint $\boldsymbol{f}_t = (f_1, f_2, \ldots, f_{L_t})$ is a list of RSS values of all MAC addresses visible at timestamp t
- Similarity between fingerprints is calculated like:

$$\text{sim}(f_i, f_j) = \underbrace{\frac{H}{L_i + L_j - H}}_{\text{Detection likelihood}} \cdot \underbrace{\frac{1}{H} \prod_{n=1}^{H} \exp\left(-\frac{(f_{i,n} - f_{j,n})^2}{2\sigma^2}\right)}_{\text{RSS likelihood}}$$

- Where L_i, L_j are total number of visible APs at respective timestamps, and H is number of common APs visible from both timestamps

K. Ismail *et al*., "Efficient WiFi LiDAR SLAM for Autonomous Robots in Large Environments," *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*

Fingerprint Similarity



Original similarity score issue

Fingerprint Similarity

- Alternative similarity between pose x_i and pose x_j, based on their fingerprint can be calculated as:

$$\text{sim}(f_i, f_j) = \left(\prod_{n=1}^{H} \exp\left(-\frac{(f_{i,n} - f_{j,n})^2}{2\sigma^2} \right) \right)^{\frac{1}{H}}$$

- Which makes similarity score less dependent to number of H (i.e. common APs)

Loop Closure with WiFi

- Due to the unpredictability of WiFi signal propagation, same poses, at different timestamps can have different fingerprints
- Therefore, computing only fingerprints similarity is not enough for detecting loop closures

for $i \leftarrow 1$ **to** T **do**

    `// Check accumulated distance and fingerprint similarity`

    for $j \leftarrow 1$ **to** i **with** $acc(\mathbf{x}_i, \mathbf{x}_j) \geq 50$ **and** $sim(\mathbf{f}_i, \mathbf{f}_j) \geq 0.3$ **do**

        $\triangleright$ Compute the relative pose $\mathbf{T}^*$ between $\mathbf{x}_i$ and $\mathbf{x}_j$ according to Equation 3

        if *Average distance computed for* $\mathbf{T}^*$ *is smaller than 3 meters* **then**

            $\triangleright$ Add $< \mathbf{x}_i, \mathbf{x}_j >$ as loop closure

        end

    end

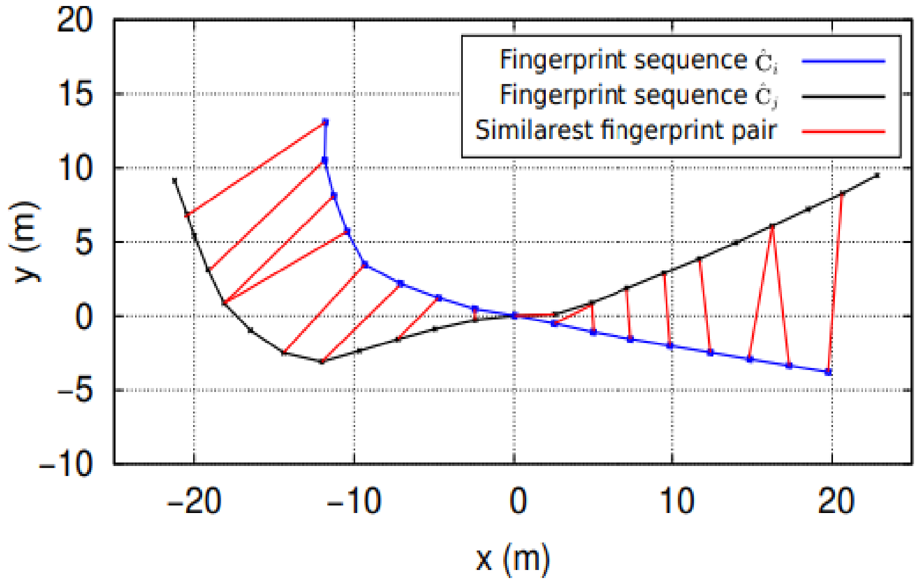
end

K. Ismail *et al*., "Efficient WiFi LiDAR SLAM for Autonomous Robots in Large Environments," *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*

Loop Closure with WiFi

- For finding relative transformation between two poses x_i and x_j, spatial sequence based approach is used
- Sequence of nodes centered around x_i and x_j
- Relative transformation is obtained as following:

$$\mathbf{T}^* = \arg\min_{\mathbf{T}} \frac{1}{w} \sum_{\tau=-\frac{w}{2}}^{\frac{w}{2}} \mathrm{dist}(\mathbf{T}(\mathbf{x}_i^{-1}\mathbf{x}_{i+\tau}), \mathbf{x}_j^{-1}\mathbf{x}_{j^*})$$



- Transformation can be found using ICP with known correspondences

K. Ismail *et al*., "Efficient WiFi LiDAR SLAM for Autonomous Robots in Large Environments," *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*

Loop Closure with WiFi

- $\boldsymbol{x}_j^*$ represents correspondence of the point $x_{i+\tau}$, and it is calculated as following:

$$\mathbf{x}_{j^*} = \frac{1}{\sum_{l=1}^{k} \mathrm{sim}(\mathbf{f}_{i+\tau}, \mathbf{f}_{\pi(l)})} \sum_{l=1}^{k} \mathrm{sim}(\mathbf{f}_{i+\tau}, \mathbf{f}_{\pi(l)}) \cdot \mathbf{x}_{\pi(l)}$$

- where *k* is *k*-nearest neighbours of the point $x_{i+\tau}$ in *j*-sequence, in similarity space

K. Ismail *et al*., "Efficient WiFi LiDAR SLAM for Autonomous Robots in Large Environments," *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*

Data Collection

Tiago - Nimbro

OSDome
Hemisphere

field of view
180º

range
20 m (10%)

min range
0.5m

weight
470 g

precision
up to ± 1.0 cm

64 Channel Lidar
+
IMU

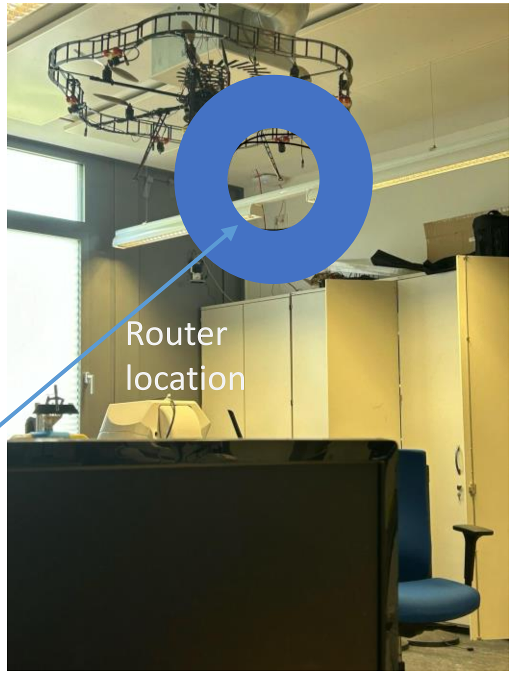
Brio fisheye camera

USB WIFI Receiver

Mobile Platform

Collected multiple bags files with in
Friedrich-Hirzebruch-Allee 8, 53115 Bonn

Exploring WIFI Data in CLINS Map



- Sphere describing the possible locations for the router
- In the trajectory,

Green indicates high signal strength,
Blue indicates low signal strength

Strength – Distance Relationship

FSPL (dB) = 20log10(d) + 20log10(f) + K

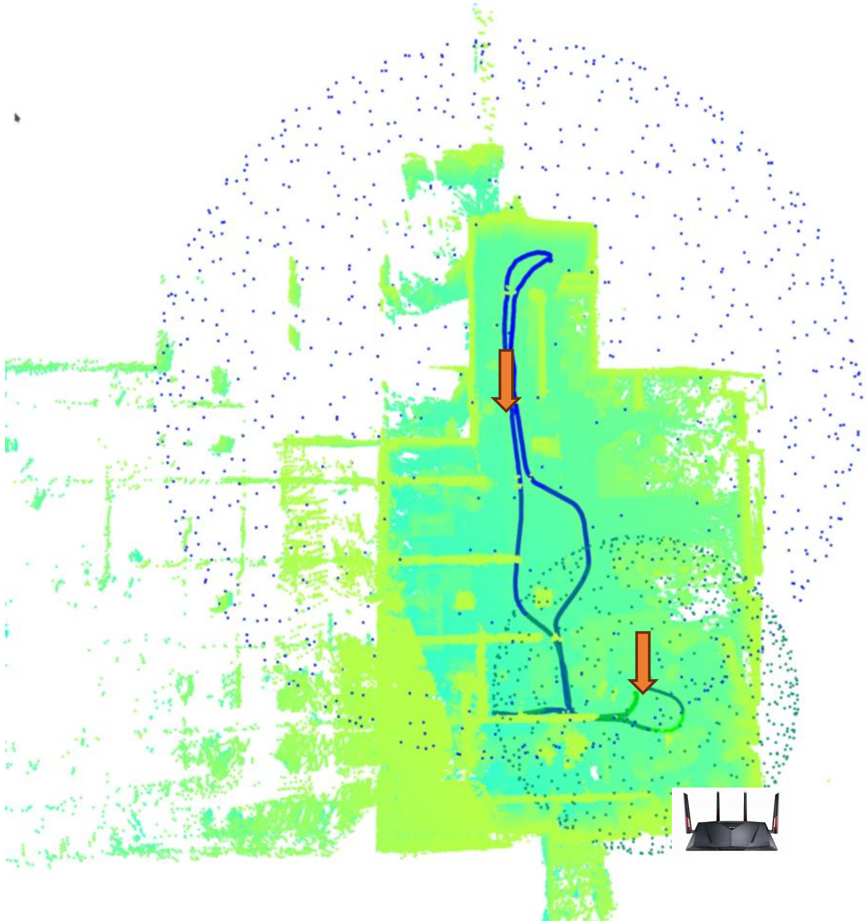
Where,

d = distance
f = frequency
K= constant that depends on the units used for d and f

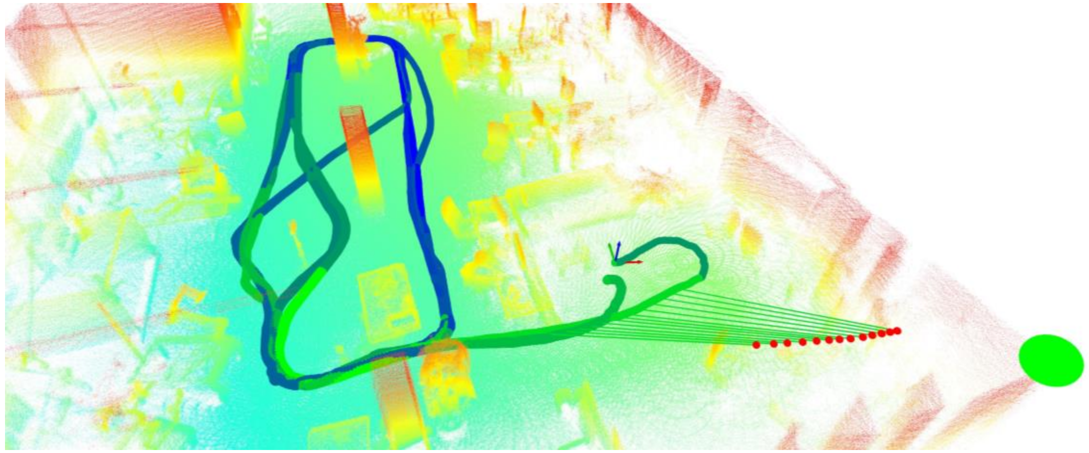
If d is measured in kilometers, f in MHz, the formula is:

FSPL (dB) = 20log10(d)+ 20log10(f) + 32.44

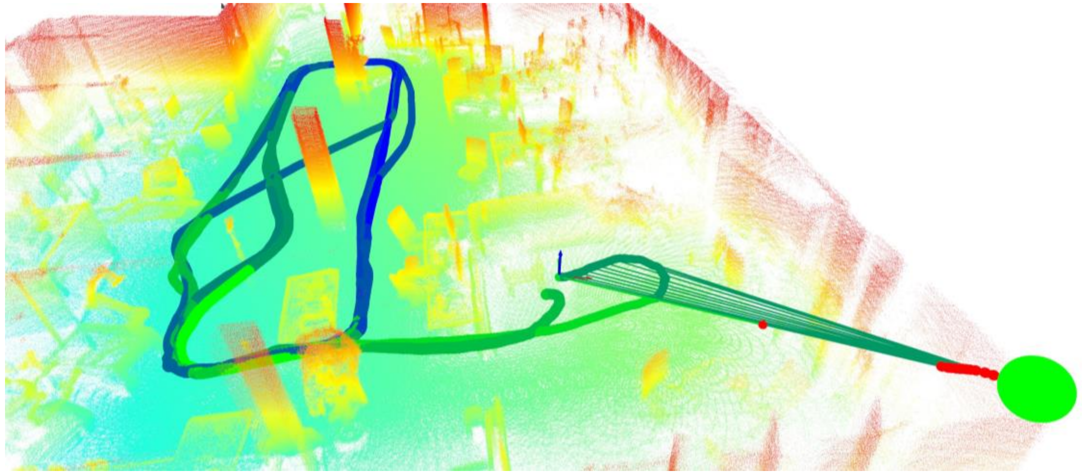
Distance Estimated using FSPL



Spheres describing the possible locations for the router

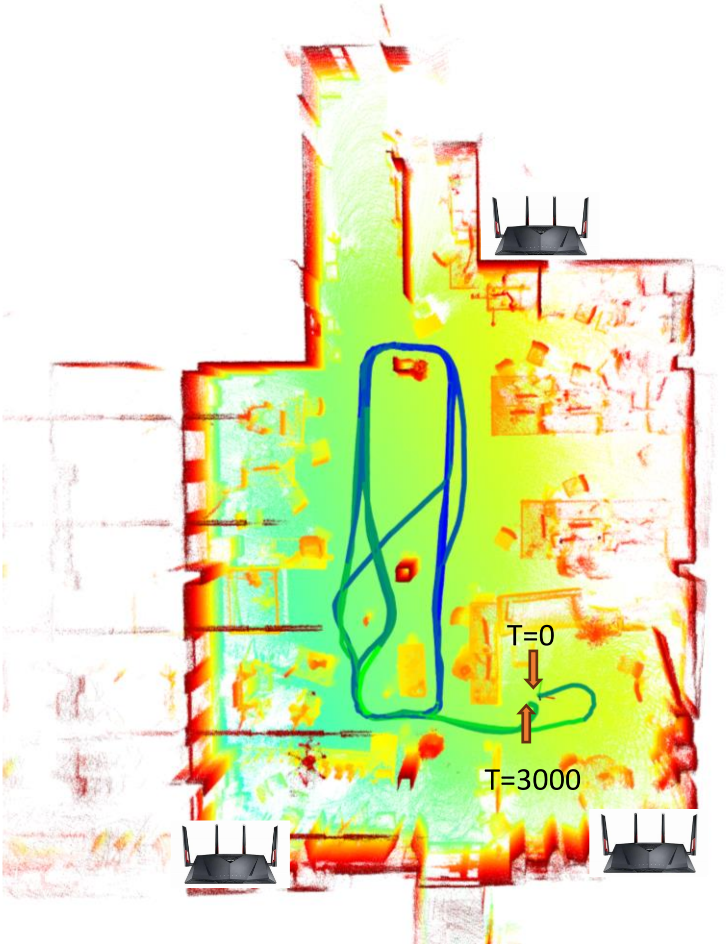


Distance estimates for time stamps 0-10 seconds

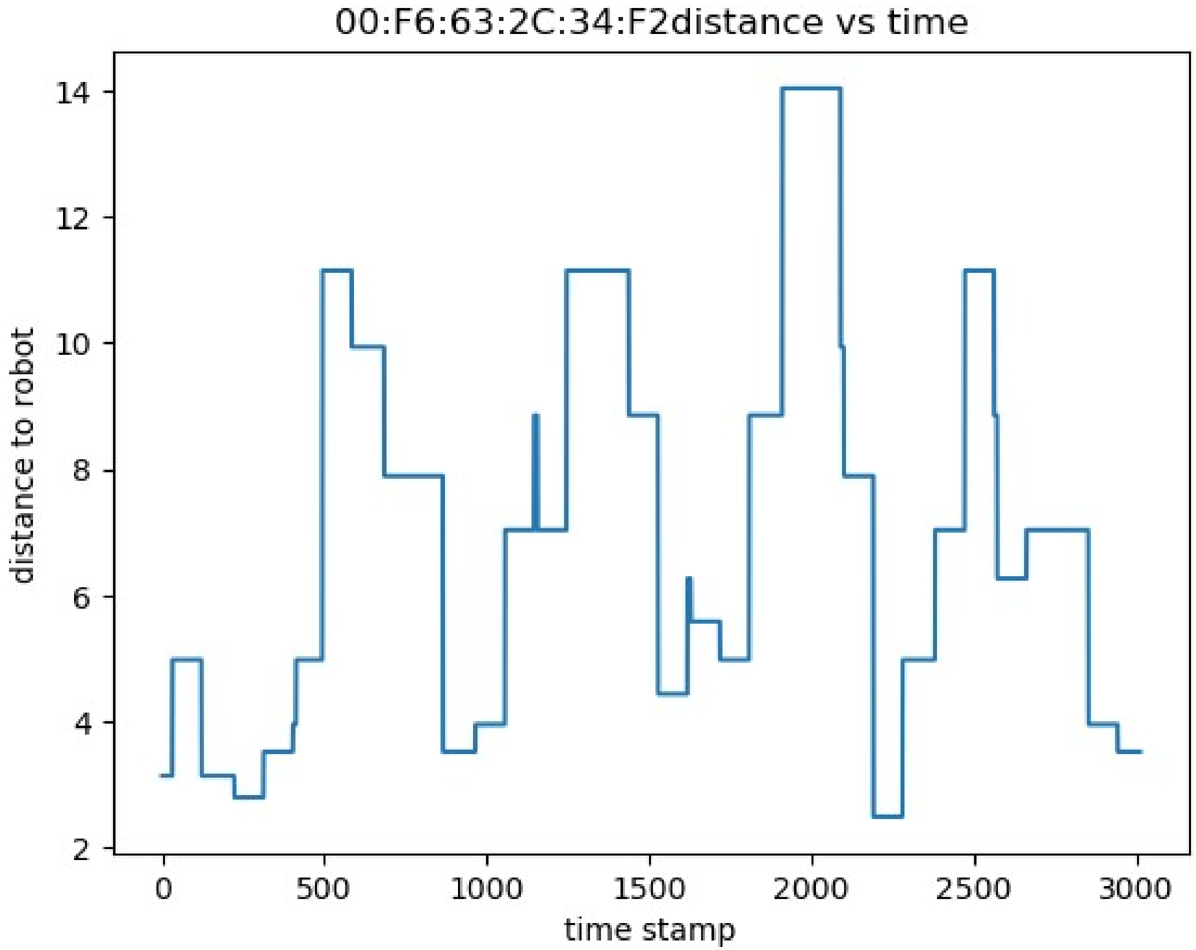


Distance estimates for time stamps 12-20 seconds

Signal Strength of a given AP over time

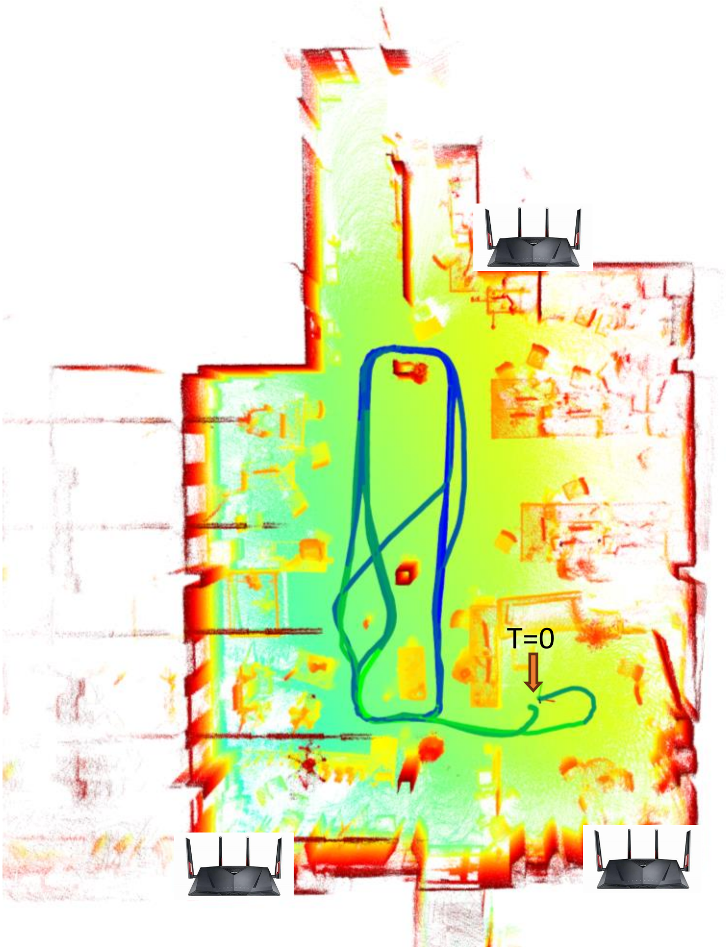


Path – 4 Clock Wise Loops

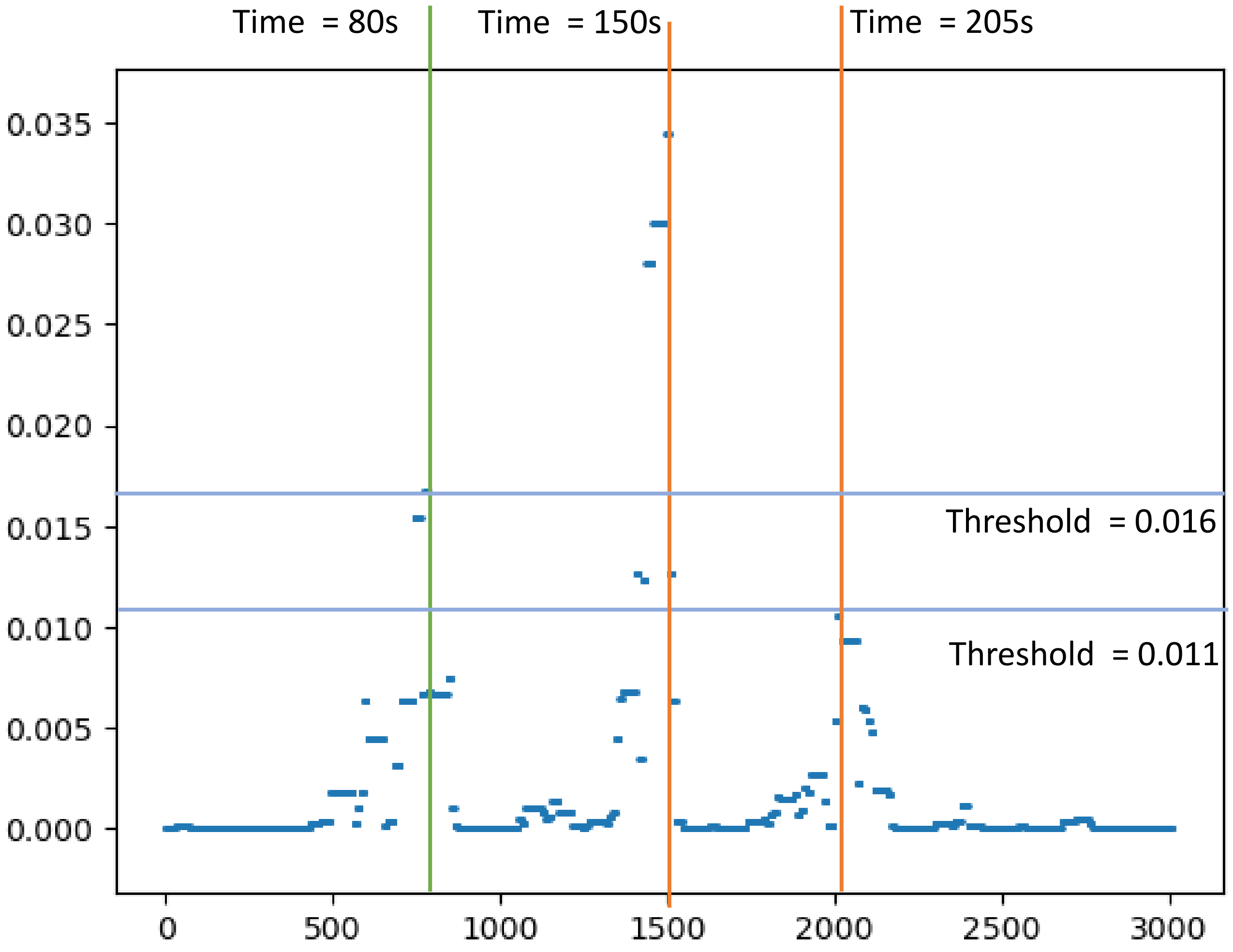


Distance calculated using FSPL formula

Similarity over time



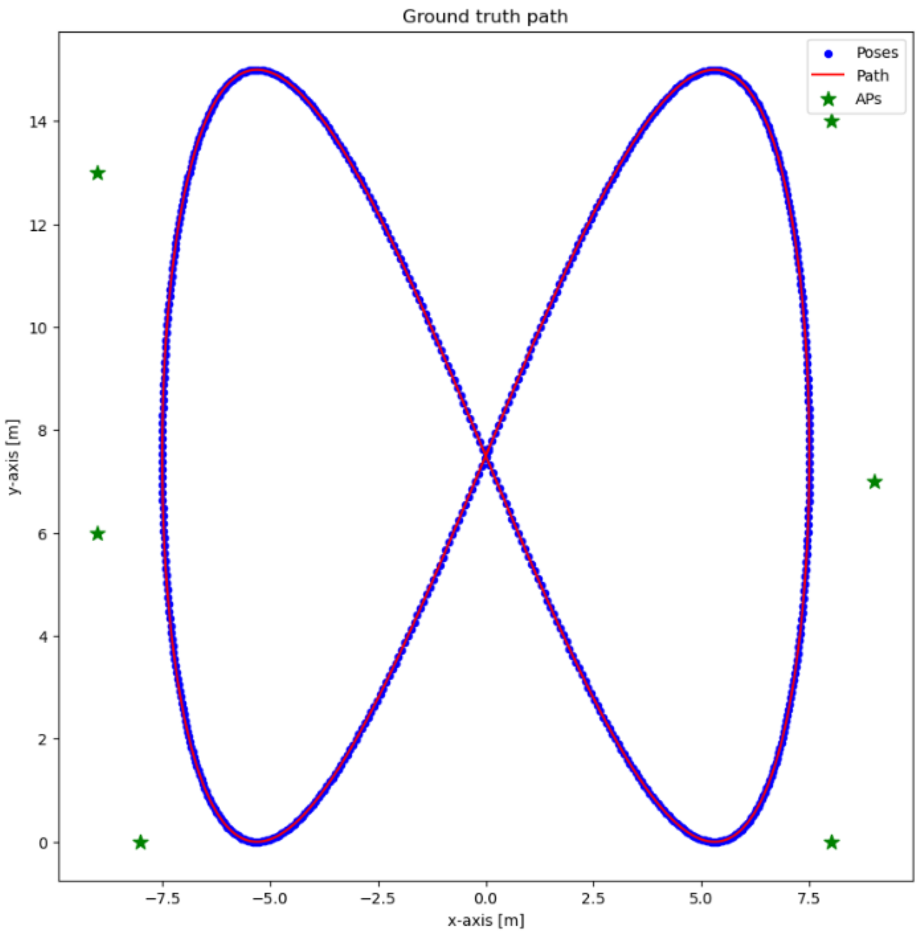
Path – 4 Clock Wise Loops

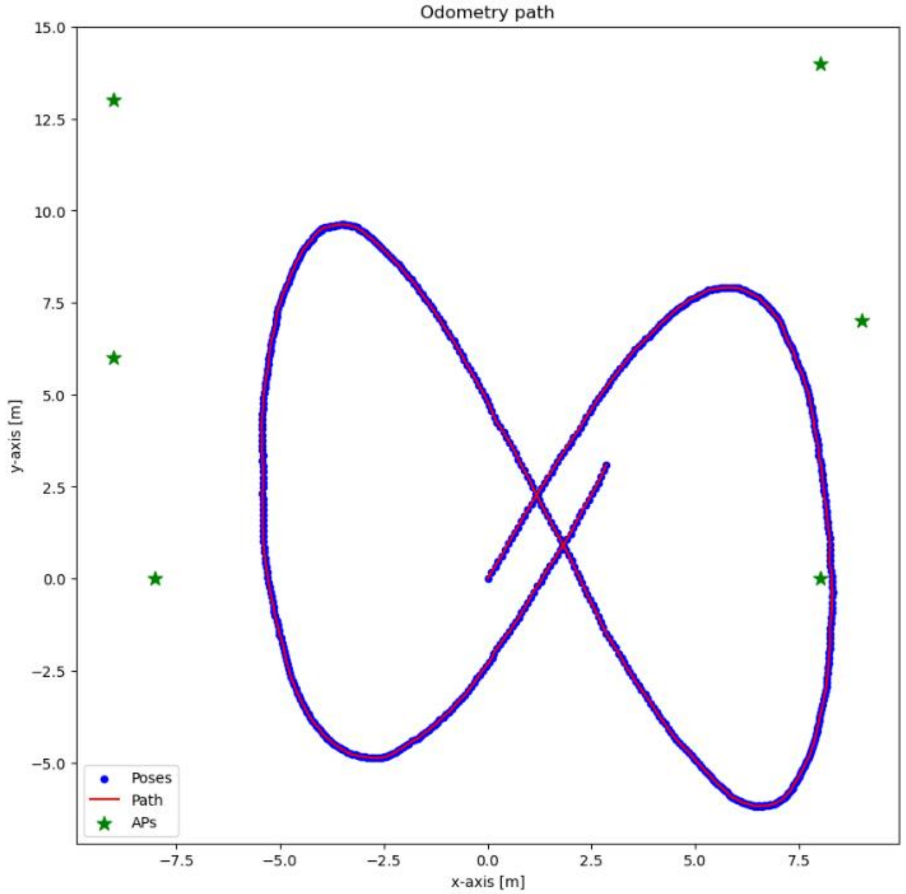


Similarity score between timestamp 1500 and all other time stamps

K. Ismail *et al*., "Efficient WiFi LiDAR SLAM for Autonomous Robots in Large Environments," *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*

Simulated Data

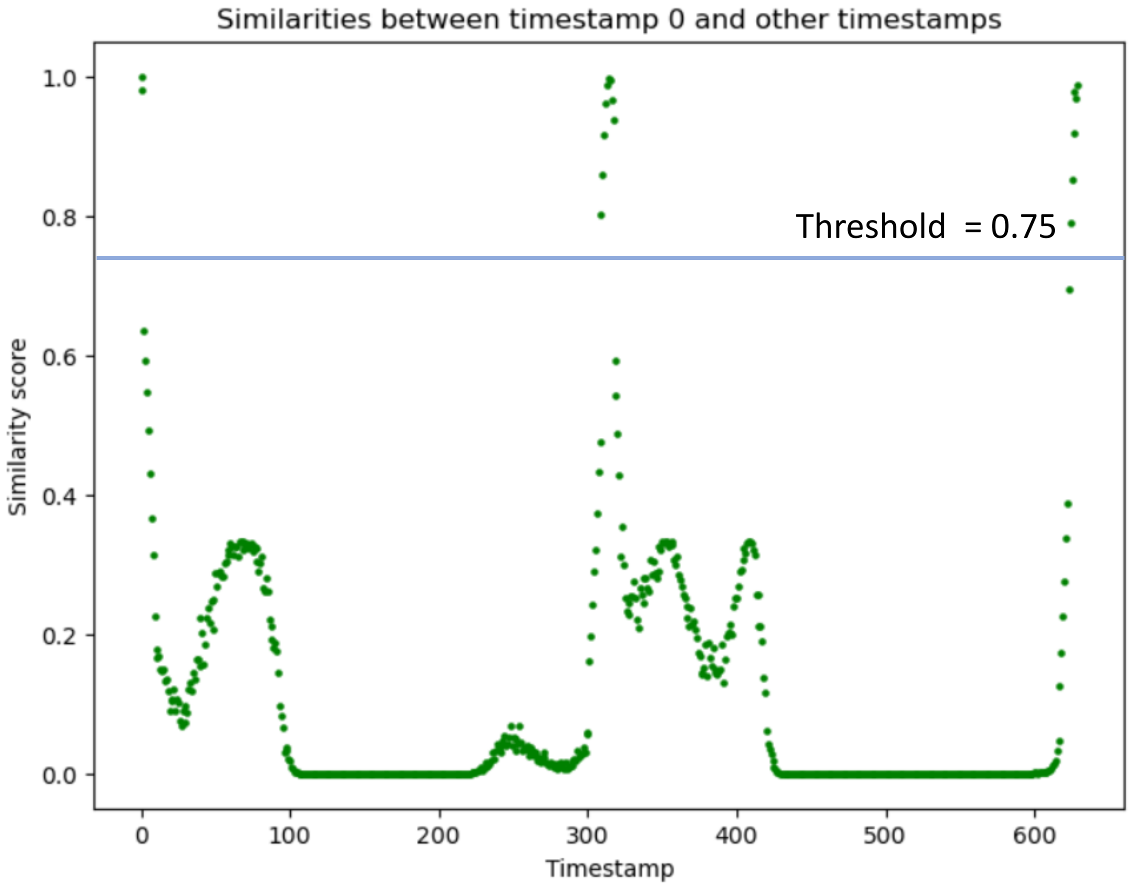




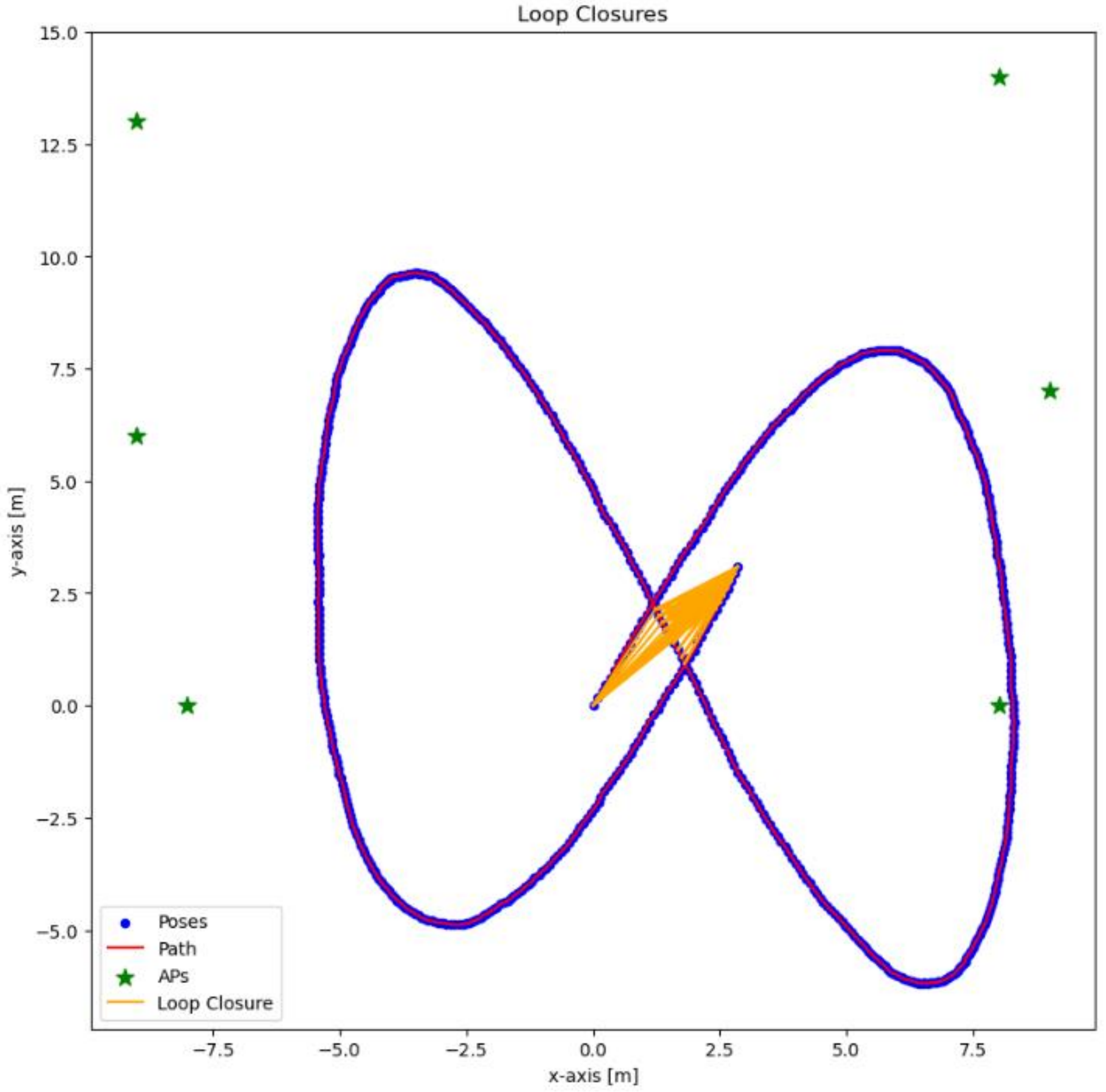
Simulated Data
- Ground Truth Odometry
- Known WIFI router locations

Simulated Data with Noise - Assumptions
- Noise and Bias Odometry
- Noisy WIFI Data (Gaussian) – Line of sight

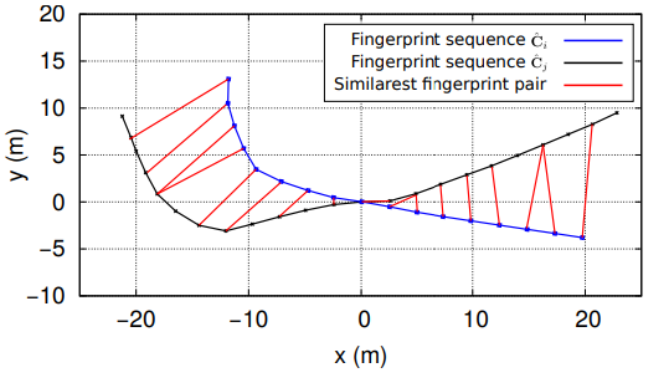
Loop Closures in Simulated Data



Conditions for a loop closure candidate
Similarity Threshold = 0.75
Min dist travelled = 20m

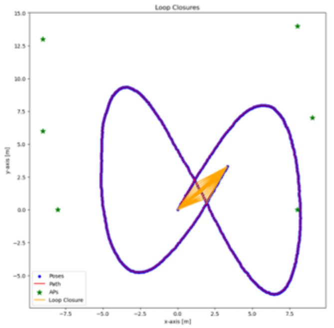


Results for Simulated Data

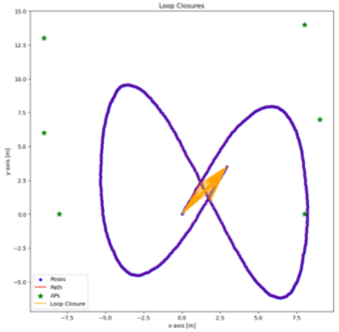


Two snippets of the trajectory isolated for a loop closure

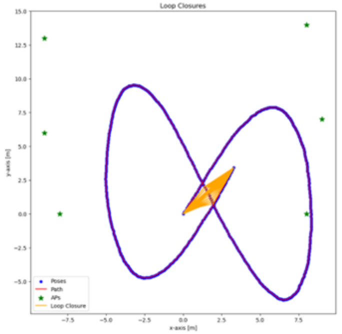
Window Size	Average Error (GT – Estimated transformation)	N(Loop Closures)
6	2.25m	61
12	1.65m	51
20	1.13m	84
30	1.14m	61



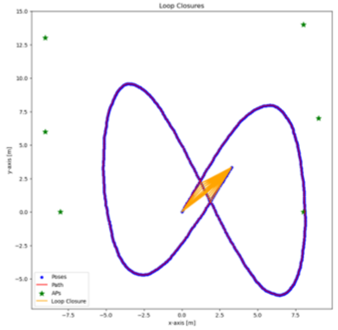
Window Size = 6



Window Size = 12

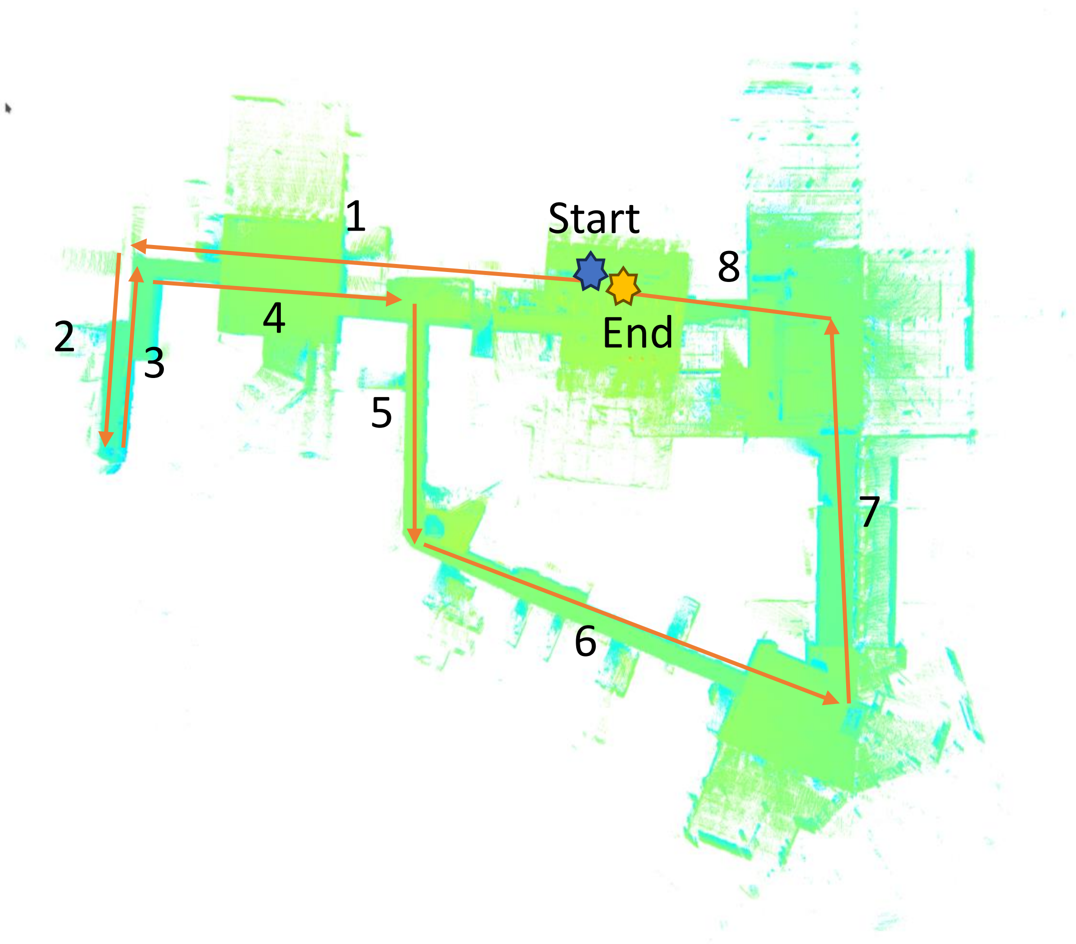


Window Size = 20



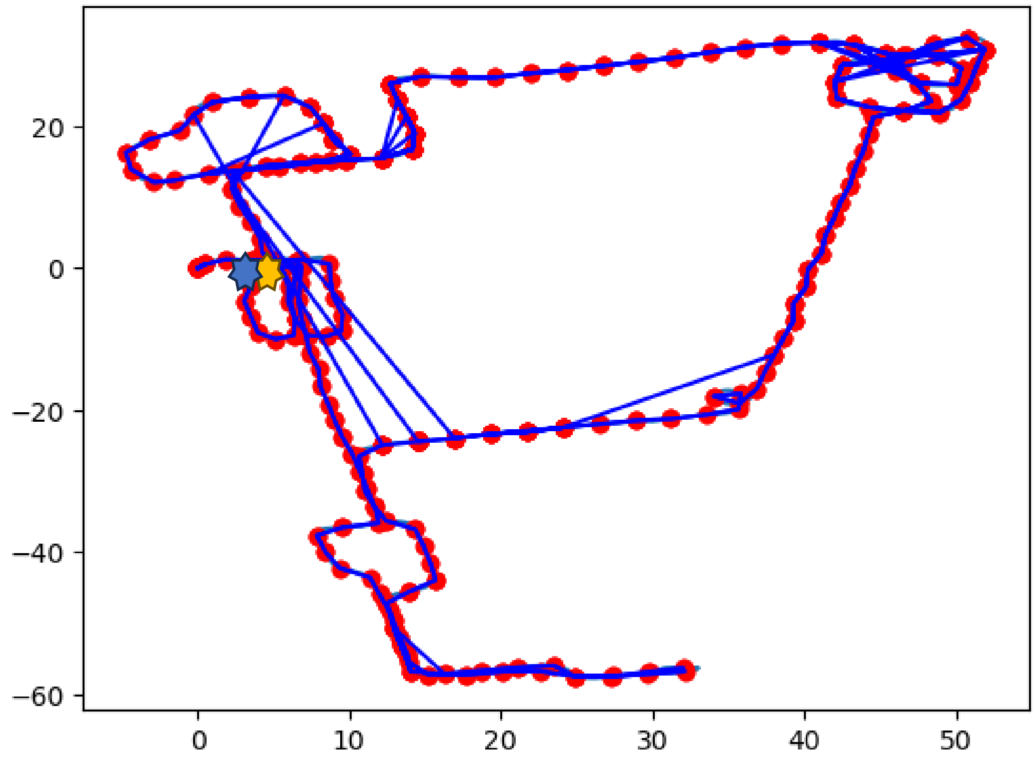
Window Size = 30

Data for the ground floor



Results for the ground floor data

Similarity score between timestamp 0 and all other time stamps



Loop Closure candidates detection using wifi fingerprint similarity

Next

- Finish optimizing the pose graph using WiFi loop closure data.
- Assess the accuracy and reliability of WiFi-based loop closures.
- Study real-world WiFi data and analyze the similarity score's behavior using parameters such as similarity threshold and window size.
- Incorporate this wifi-loop closure into an existing SLAM system, such as CLINS.

Next

- Finish optimizing the pose graph using WiFi loop closure data.
- Assess the accuracy and reliability of WiFi-based loop closures.
- Study real-world WiFi data and analyze the similarity score's behavior using parameters such as similarity threshold and window size.
- Incorporate this wifi-loop closure into an existing SLAM system, such as CLINS.