Where Am I? Indoor Localization and Navigation

Where Am I? Indoor Localization and Navigation

Árni Bjarnsteinsson[1], Ross Roessler[1], Matej Mrazek[1], Fabian Blatter[1]   Supervisors: Tong Wei[2], Daniel Barath[1]
[1]ETH Zurich; [2]CTU Prague

1 Introduction

Common localization techniques like GPS tracking are not well suited for indoor environments. We design an application that can localize the user in a known indoor environment using image data, visualizes the surroundings, and contains basic navigation capabilities.

2 Method Overview

- The Magic Leap 2 uses the front camera to obtain a picture. It queries an API running HLoc [2] (Section 3)
- The API responds with an estimate of the user position and rotation, in the absolute coordinates of a known scene. We use the ETH HG building from the Lamar [1] dataset.
- Our application uses the pose data to preview the surroundings, show the perceived location of the user inside the entire location, and allow for navigation (Section 4)

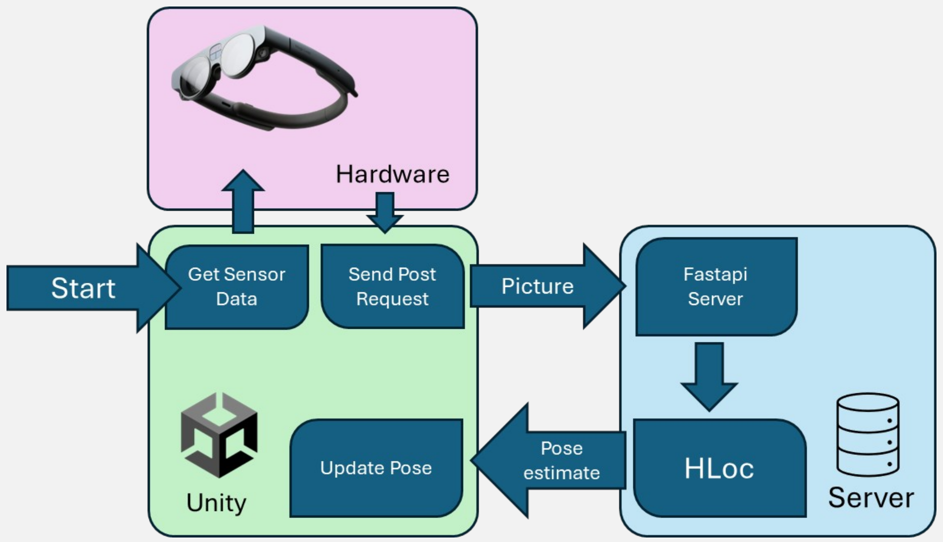
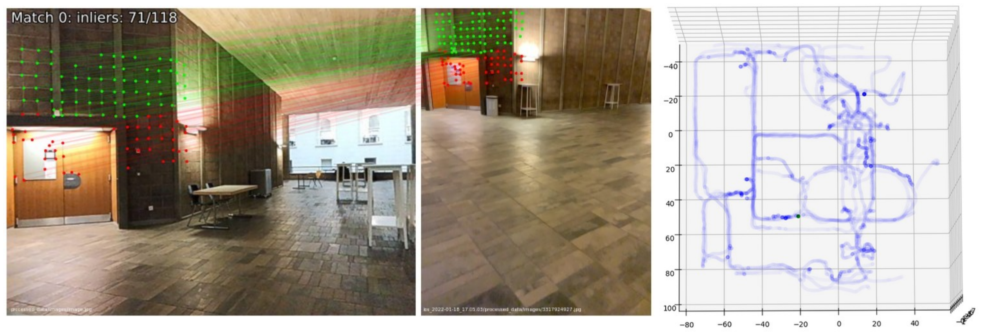


Diagram of client server connection

3 Hierarchical Localization

Steps:

1) **Preprocessing.** Extract SuperPoint and NetVLAD [3] features from LaMAR image dataset of 10,000 images. These are used for two different matching algorithms.
2) **Process Query.** Extract same features from query image.
3) **NetVLAD**. Fast image matching. Returns top 40 matches from database.
4) **Detailed Matching**. Using SuperGlue, match the keypoints from the query image to the 40 database matches.
5) **Load 3D Data**. Load, rescale, and shift depth image. Filter to valid depth points.
6) **COLMAP [4]**. Using RANSAC, estimate the camera position from all 3D points.
7) **Transform**. Convert position to minimap global frame.



5 Conclusion

Results:

- Great experience when correctly localized
- *Localization:* 10 seconds   *Accuracy:* 2 meters
- Challenges: Limited depth and global trajectory data, many similar areas in HG, speed vs. accuracy tradeoff

Possible Improvements:

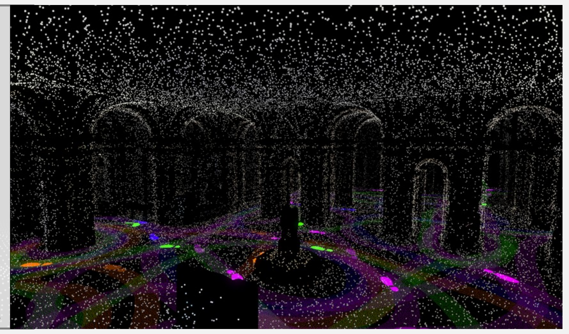
- Depth and trajectory interpolation so we can use more data
- Smooth localization estimates with Pose Graph Optimization

4 Visualization

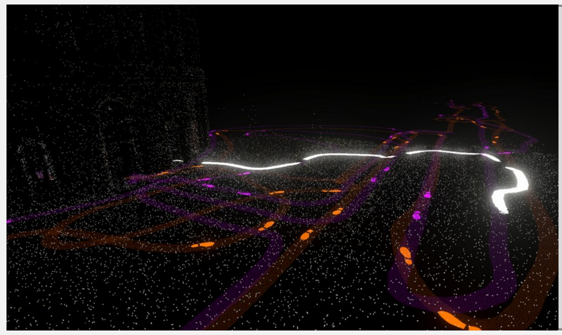
We design an application in Unity that renders an overlay to be shown over the real world. We have three main components:

Point Visualization

We show a subset of the points from the Lamar dataset. We modify the rendering process to hide points occluded by geometry - everything seen as black in the rendering should be transparent on the headset.



Point cloud showing the fountain near the main ETH entrance



Trajectories

We show the trajectories taken when our dataset was collected. This informs the user where data for localization is available and where it is not.

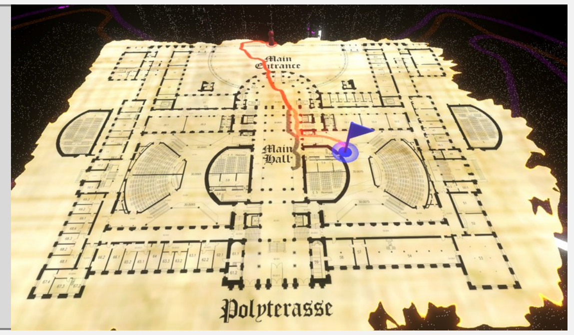
The white trajectory shows the planned navigation path.

The trajectories near the main entrance of the HG building.

Minimap

Shows the current location of the user (red pawn) and the navigation target (blue flag) in the entire building.

The currently planned path is shown in red, the path currently being planned based on user input is shown in brown.



A preview of the map inside the app

References

1. Sarlin P E, Dusmanu M, Schönberger J L, et al. Lamar: Benchmarking localization and mapping for augmented reality. ECCV 2022.
2. Sarlin P E, Cadena C, Siegwart R, et al. From coarse to fine: Robust hierarchical localization at large scale. CVPR 2019
3. Arandjelovic, Relja, et al. NetVLAD: CNN architecture for weakly supervised place recognition. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
4. Schonberger, Johannes L., and Jan-Michael Frahm. Structure-from-motion revisited. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.