LiDAR-Based Localization and Environment Recognition

Autonomous Aerial Vehicle Project

Niusha Sedgh
Amirkabir university of technology

Spring 2026

Abstract

This project addresses relative localization and environment recognition of an autonomous aerial vehicle using synchronized LiDAR, camera, and inertial measurement unit (IMU) data. The localization pipeline estimates the six-degree- of-freedom (6-DoF) motion of the vehicle from consecutive LiDAR point clouds using multi-scale Iterative Closest Point (ICP). Visual odometry based on SIFT feature matching and the essential matrix is additionally employed to provide motion information when LiDAR registration is unreliable.

To improve the robustness and temporal smoothness of the localization estimate, an error-state Extended Kalman Filter (EKF) is implemented. The prediction stage uses accelerometer and gyroscope measurements, while LiDAR-based pose estimates are incorporated as measurements during the update stage. The resulting pose estimate is also used as an initial guess for ICP.

The second part of the project focuses on environment recognition. A YOLO-World detector is used to identify trees, traffic lights, and fire hydrants in camera images. The synchronized LiDAR point cloud is projected onto the camera image, allowing LiDAR points associated with each detected object to be selected. A robust median-based procedure is then used to estimate the metric three-dimensional position of each object. Finally, the estimated vehicle pose is used to transform object positions into a global reference frame and construct an environment map.

Contents

1 Introduction

Autonomous vehicles require reliable localization and environment perception for safe navigation. LiDAR-based odometry and visual motion estimation are widely used approaches for estimating vehicle motion from onboard sensor measurements [2,4]. In this project, synchronized LiDAR, camera, and IMU measurements are used to estimate the motion of an aerial vehicle and recognize objects in its surrounding environment.

The localization system estimates the vehicle's relative 6-DoF pose from consecutive LiDAR point clouds using multi-scale ICP. Visual odometry is also used as a complementary source of motion information when LiDAR registration is unreliable. An Extended Kalman Filter (EKF) further combines IMU measurements with pose estimates to improve the robustness of the localization.

The second part of the project focuses on environment recognition. Detected objects are associated with LiDAR points through camera–LiDAR projection, allowing their metric 3D positions to be estimated and transformed into a global reference frame. The resulting object positions are then used to construct an environment map.

2 Problem Statement and Objectives

The project consists of two main tasks.

2.1 Vehicle Localization

The first task is to estimate the relative 6-DoF pose of the aerial vehicle from sequential sensor measurements. The pose consists of three translational components and three rotational components:

$$\mathbf{x} = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T, \tag{1}$$

where ϕ, θ, and ψ denote roll, pitch, and yaw, respectively.

The main requirements are:

- Registration of consecutive LiDAR point clouds.
- Estimation of the relative 6-DoF vehicle motion.
- Construction of the complete estimated trajectory.
- Alignment of the estimated trajectory with the ground truth.
- Quantitative evaluation of position and orientation errors.
- Development of an EKF using IMU data as an optional scoring component.

2.2 Environment Recognition

The second task consists of:

1. Detecting the required environmental objects in camera images.
2. Estimating their metric three-dimensional positions.

3. Transforming these positions into the global reference frame.
4. Visualizing the detected objects on an environment map.

The required object categories are trees, traffic lights, and fire hydrants.

3 Sensor Configuration

The project provides synchronized measurements from a visible camera, LiDAR, and IMU.

3.1 Camera

The camera resolution is 1280×720 pixels. The intrinsic camera matrix provided by the project is

$$\mathbf{K} = \begin{bmatrix} 1109 & 0 & 640 \\ 0 & 1109 & 360 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

No lens distortion is specified.

The camera is mounted approximately 7.5 cm in front of the vehicle center and is physically aligned with the vehicle body.

3.2 LiDAR

The LiDAR has a maximum range of 30 m. Its vertical field of view is 60° with an angular resolution of 1.25°, while its horizontal field of view is 360° with an angular resolution of 0.16°.

The LiDAR is also mounted approximately 7.5 cm in front of the vehicle center.

3.3 IMU

The IMU measurements are provided at a nominal frequency of 100 Hz. The available measurements include

$$\mathbf{a} = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T \tag{3}$$

and

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T. \tag{4}$$

4 Coordinate Systems and Sensor Extrinsics

Correct handling of coordinate systems is critical because the LiDAR, camera, body, and world frames do not necessarily follow the same axis conventions.

The vehicle reference frame is considered aligned with the LiDAR frame, with the LiDAR mounted 7.5 cm forward of the body center. Therefore, the LiDAR-to-body transformation is modeled as

$$
\mathbf{T}_L^B = \begin{bmatrix} 1 & 0 & 0 & 0.075 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5}
$$

The corresponding inverse transformation is

$$
\mathbf{T}_B^L = \left(\mathbf{T}_L^B\right)^{-1}. \tag{6}
$$

For the camera, the implemented camera-to-body rotation is

$$
\mathbf{R}_C^B = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}. \tag{7}
$$

This transformation accounts for the different axis conventions between the OpenCV camera frame and the vehicle body frame.

Because the camera and LiDAR are both specified to be aligned with the body and located at the same longitudinal offset, their relative translation is modeled as zero unless an additional calibration is provided.

5 LiDAR Odometry

5.1 Point Cloud Preprocessing

Each LiDAR measurement is loaded as a point cloud in PLY format. Before ICP registration, the point cloud can be processed using voxel downsampling and statistical outlier removal.

Voxel downsampling reduces the number of points and consequently decreases the computational cost of registration. Statistical outlier removal is used to suppress isolated measurements.

The implemented preprocessing pipeline is illustrated below:

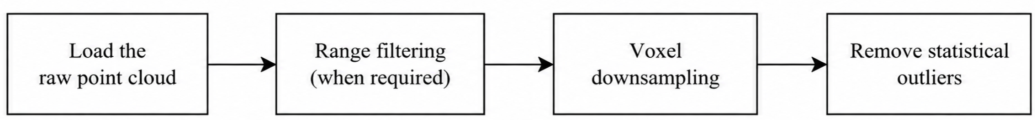


Figure 1: LiDAR point cloud preprocessing pipeline.

5.2 Multi-Scale ICP

The relative transformation between consecutive LiDAR frames is estimated using the Iterative Closest Point (ICP) algorithm [1]. The source point cloud corresponds to the current frame, while the target point cloud corresponds to the previous frame.

The transformation is initialized using a predicted relative motion. ICP is then performed at multiple spatial scales. The implemented scale factors are

$$
[2.5,\ 1.5,\ 1.0]. \tag{8}
$$

At each scale, the point clouds are voxel-downsampled and a corresponding maximum correspondence distance is selected.

Point-to-plane ICP is used when valid surface normals can be estimated. Otherwise, the algorithm automatically falls back to point-to-point ICP.

For each registration step, the fitness and inlier RMSE reported by Open3D are monitored.

5.3 Motion Representation

The relative transformation is represented using a homogeneous matrix

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{9}$$

where $\mathbf{R} \in SO(3)$ represents rotation and $\mathbf{t} \in \mathbb{R}^3$ represents translation.

The relative motion obtained from ICP is converted between LiDAR and body coordinates using the corresponding extrinsic transformations.

6 Visual Odometry

Visual odometry is used as a complementary motion estimation method, particularly when LiDAR registration is rejected.

6.1 Feature Detection and Matching

SIFT (Scale-Invariant Feature Transform) is used as the primary feature detector because of its robustness to scale and rotation changes [2]. ORB is used as a fallback if SIFT is unavailable.

For each pair of consecutive images, descriptors are extracted and matched using a brute-force matcher. The Lowe ratio test is then applied to reject ambiguous feature correspondences.

A match is accepted when

$$d_1 < 0.75 d_2, \tag{10}$$

where d_1 and d_2 are the distances of the best and second-best matches, respectively.

6.2 Essential Matrix and Relative Pose

The matched image points are used to estimate the essential matrix through RANSAC. The relative camera pose can be recovered from the epipolar geometry using the five-point relative pose formulation [3].

$$\mathbf{E} = \texttt{findEssentialMat}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{K}). \tag{11}$$

The relative rotation and translation direction are then obtained using `recoverPose`.

Since the system uses a monocular camera, the magnitude of the recovered translation cannot be directly determined. Therefore, the translation is normalized and its scale is estimated using the magnitude of the previous accepted LiDAR motion.

6.3 Visual Fallback

When ICP fails a quality or motion-consistency test, visual odometry can be used as a fallback motion estimate.

In addition, the implementation contains a yaw-only visual fallback. In this case, the translation obtained from LiDAR is retained and only the yaw component is replaced using the visual estimate. This approach is useful in situations where visual rotation is reliable but monocular translation is not.

7 Motion Validation

An ICP result is not accepted solely based on successful numerical convergence. The resulting transformation is evaluated using both ICP quality metrics and motion consistency.

The two main ICP quality criteria are fitness and inlier RMSE. In addition, the estimated translation magnitude and yaw change are compared with reasonable motion limits.

If the estimated motion is inconsistent with the expected vehicle motion, the ICP result is rejected and an alternative estimate is considered.

This validation stage prevents a locally converged but physically incorrect ICP solution from corrupting the global trajectory.

8 Extended Kalman Filter

8.1 State Representation

An error-state EKF is implemented to fuse high-rate IMU measurements with lower-rate pose measurements.

The nominal state is

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} & \mathbf{v} & \mathbf{R} & \mathbf{b}_a & \mathbf{b}_g \end{bmatrix}, \tag{12}$$

where

- $\mathbf{p}$ is position,
- $\mathbf{v}$ is velocity,
- $\mathbf{R}$ is the body orientation,
- $\mathbf{b}_a$ is accelerometer bias,
- $\mathbf{b}_g$ is gyroscope bias.

The covariance matrix is a 15×15 matrix corresponding to

$$\delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{p} & \delta\mathbf{v} & \delta\boldsymbol{\theta} & \delta\mathbf{b}_a & \delta\mathbf{b}_g \end{bmatrix}^T. \tag{13}$$

8.2 Prediction Model

The accelerometer measurement is first corrected using the estimated bias:

$$\mathbf{a}_b = \mathbf{a}_m - \mathbf{b}_a. \tag{14}$$

The acceleration in the world frame is

$$\mathbf{a}_w = \mathbf{R}\mathbf{a}_b + \mathbf{g}, \tag{15}$$

where the gravity vector in the ENU coordinate system is

$$\mathbf{g} = \begin{bmatrix} 0 & 0 & -9.81 \end{bmatrix}^T. \tag{16}$$

Position and velocity are propagated according to

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{v}_k \Delta t + \frac{1}{2}\mathbf{a}_w \Delta t^2, \tag{17}$$

and

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \mathbf{a}_w \Delta t. \tag{18}$$

The orientation is propagated using the exponential map on $SO(3)$, following the Lie-group formulation commonly used for robotic state estimation [5]:

$$\mathbf{R}_{k+1} = \mathbf{R}_k \exp\left((\boldsymbol{\omega}_m - \mathbf{b}_g)\Delta t\right). \tag{19}$$

The covariance is propagated using

$$\mathbf{P}_{k+1} = \mathbf{F}_k \mathbf{P}_k \mathbf{F}_k^T + \mathbf{Q}_k. \tag{20}$$

8.3 Pose Update

The LiDAR-based pose is used as an EKF measurement.

The position residual is

$$\mathbf{e}_p = \mathbf{p}_{meas} - \mathbf{p}_{pred}. \tag{21}$$

For orientation, the rotation error is calculated as

$$\mathbf{R}_{err} = \mathbf{R}_{meas}\mathbf{R}_{pred}^T, \tag{22}$$

and the corresponding small-angle attitude error is obtained using the logarithmic map on $SO(3)$ [5].

$$\delta\boldsymbol{\theta} = \log\left(\mathbf{R}_{err}\right). \tag{23}$$

The complete innovation vector is

$$\mathbf{y} = \begin{bmatrix} \mathbf{e}_p \\ \delta\boldsymbol{\theta} \end{bmatrix}. \tag{24}$$

The Kalman gain is

$$\mathbf{K} = \mathbf{P}\mathbf{H}^T \left(\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R}\right)^{-1}. \tag{25}$$

The estimated error state is then injected into the nominal state.

The covariance update is implemented using the Joseph form:

$$\mathbf{P} = (\mathbf{I} - \mathbf{KH})\mathbf{P}(\mathbf{I} - \mathbf{KH})^T + \mathbf{KRK}^T. \tag{26}$$

8.4 EKF Prediction as an ICP Initial Guess

One important advantage of the EKF is that the predicted pose can be used to initialize the ICP registration.

The predicted global poses at two consecutive LiDAR timestamps are converted into a relative body-frame motion:

$$\mathbf{T}_{prev \to curr} = \left(\mathbf{T}_{global}^{prev}\right)^{-1} \mathbf{T}_{global}^{curr}. \tag{27}$$

This prediction is then converted to the LiDAR frame and supplied as the initial transformation to ICP.

To avoid unstable predictions from becoming harmful initial guesses, limits are imposed on the predicted translation and yaw. If these limits are exceeded, the previous accepted motion is used instead.

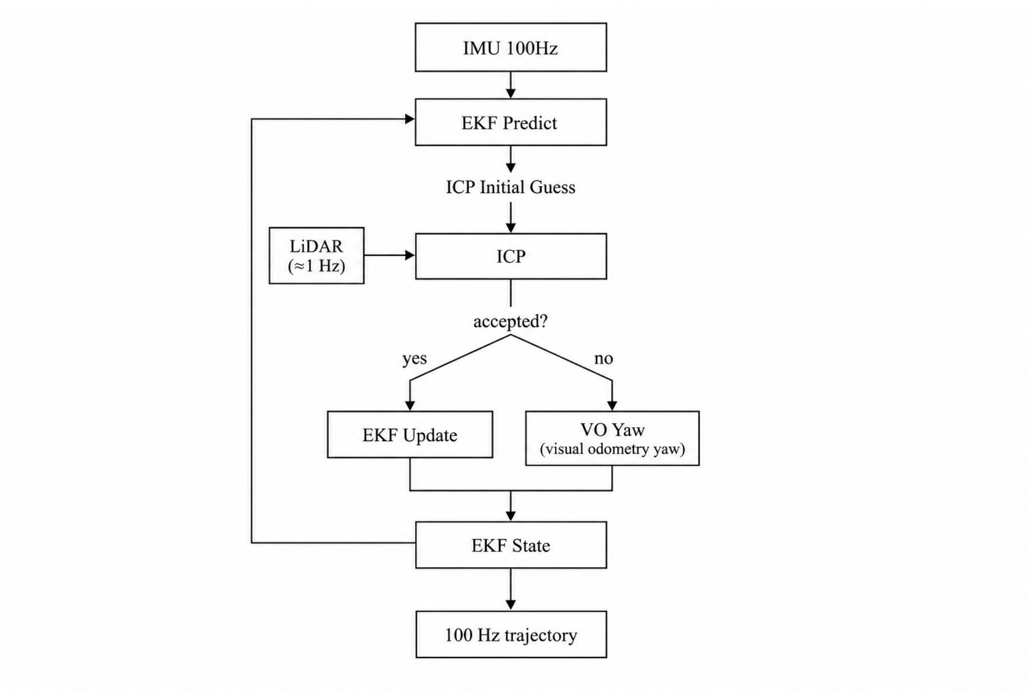


Figure 2: Architecture of the localization pipeline.

9 Trajectory Evaluation

The estimated trajectory is evaluated against the provided ground-truth trajectory. Because the estimated trajectory is relative, it is first aligned with the ground truth using a 4-DoF transformation consisting of yaw rotation and three-dimensional translation.

Let $\mathbf{p}_e$ denote the estimated positions and $\mathbf{p}_g$ the ground-truth positions. The horizontal components are used to estimate the yaw alignment through SVD.

The aligned position is then

$$\mathbf{p}_{aligned} = \mathbf{R}_{yaw}\mathbf{p}_e + \mathbf{t}. \tag{28}$$

The translation is selected such that the centroids of the two trajectories are aligned. After alignment, the position error is

$$\mathbf{e}_p(k) = \mathbf{p}_{aligned}(k) - \mathbf{p}_{gt}(k). \tag{29}$$

The absolute trajectory error (ATE) RMSE is

$$ATE_{RMSE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N} \|\mathbf{e}_p(k)\|^2}. \tag{30}$$

The implementation also calculates MAE, maximum position error, and per-axis RMSE and MAE.

For orientation, the estimated yaw is corrected using the alignment yaw, and angular errors are wrapped to the interval $[-\pi, \pi]$.

The following metrics are therefore reported:

- ATE RMSE
- ATE MAE
- Maximum position error
- X, Y, and Z RMSE
- X, Y, and Z MAE
- Yaw RMSE and MAE
- Pitch RMSE and MAE
- Roll RMSE and MAE

10 Environment Recognition

10.1 Object Detection

The environment-recognition stage uses YOLO-World for open-vocabulary object detection [6]. The detector is configured with the three required object classes:

- Tree
- Traffic light
- Fire hydrant

For every input image, the detector returns the object class, confidence score, and bounding box

$$B = [x_1, y_1, x_2, y_2]. \tag{31}$$

The synchronized image and LiDAR frame are matched using the frame identifier contained in their filenames.

11 LiDAR-to-Camera Projection

To associate metric LiDAR information with image detections, LiDAR points are projected into the camera image.

A LiDAR point is transformed into the camera coordinate system according to

$$\mathbf{p}_C = \mathbf{T}_L^C \mathbf{p}_L. \tag{32}$$

For a camera-frame point

$$\mathbf{p}_C = \begin{bmatrix} X_C & Y_C & Z_C \end{bmatrix}^T, \tag{33}$$

the corresponding image coordinates are

$$u = f_x \frac{X_C}{Z_C} + c_x, \tag{34}$$

$$v = f_y \frac{Y_C}{Z_C} + c_y. \tag{35}$$

Points with invalid depth are rejected. Points projected outside the 1280×720 image are also excluded.

A dedicated projection-debugging procedure was implemented to verify the coordinate convention between the LiDAR and camera before performing object localization.

12 3D Object Position Estimation

For each detected object, LiDAR points whose image projections fall inside the corresponding bounding box are selected.

Using the complete bounding box can introduce background points. Therefore, the central region of the bounding box can be used to obtain a cleaner association with the detected object.

The candidate LiDAR points are first used to calculate a robust median depth:

$$d_{med} = \text{median}\left(d_1, \ldots, d_N\right). \tag{36}$$

Points whose depths differ significantly from this dominant depth are discarded. Specifically, the implemented tolerance is

$$\tau = \max(0.5,\ 0.1 d_{med}). \tag{37}$$

The final object position in the LiDAR coordinate frame is estimated as the coordinate-wise median of the remaining points:

$$\mathbf{p}_{obj}^L = \text{median}\left(\mathbf{p}_1, \ldots, \mathbf{p}_M\right). \tag{38}$$

This median-based procedure reduces the influence of sparse outliers and background points.

13 Transformation to the Global Reference Frame

Once the object position is estimated in the LiDAR frame, it is transformed to the global reference frame using the estimated vehicle pose.

The global LiDAR transformation is

$$\mathbf{T}_G^L = \mathbf{T}_G^B \mathbf{T}_L^B. \tag{39}$$

Therefore,

$$\mathbf{p}_{obj}^G = \mathbf{T}_G^L \mathbf{p}_{obj}^L. \tag{40}$$

The resulting coordinates represent the metric three-dimensional position of the detected object relative to the initial vehicle reference frame.

For every valid detection, the system stores:

- Frame identifier
- Object class
- Detection confidence
- Bounding-box coordinates
- Number of associated LiDAR points
- Estimated depth
- LiDAR-frame position
- Camera-frame position
- Global-frame position

14 Environment Mapping

The estimated global positions of detected objects are accumulated over the sequence.

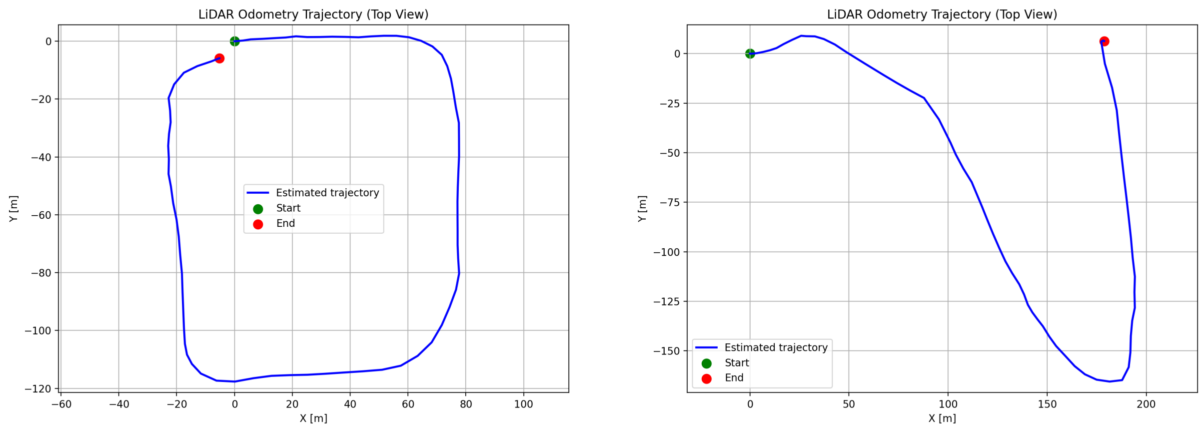
The resulting environment map represents the detected objects in the horizontal reference plane. The vehicle starting position is considered the origin of the reference frame.

Multiple observations of the same object may appear at different locations because of localization and object-position estimation errors. Nevertheless, the resulting map provides a spatial representation of the recognized environment and demonstrates the integration of localization and perception.

15 Experimental Results

15.1 LiDAR-Only Odometry Results

The first experiment evaluates the baseline LiDAR odometry system. In this configuration, consecutive LiDAR point clouds are registered using the multi-scale ICP algorithm without visual odometry or EKF-based initialization.

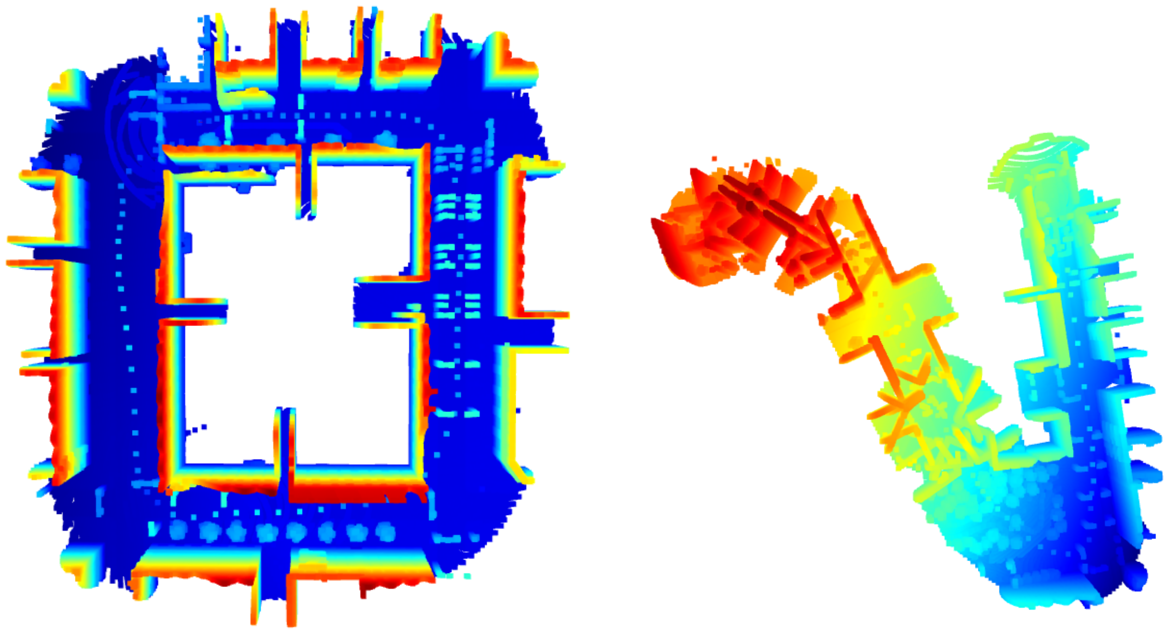


(a) Scenario 1.

(b) Scenario 2.

Figure 3: LiDAR-only estimated trajectories compared with ground truth after 4-DoF alignment.

The merged LiDAR point clouds obtained from the ICP registration are shown in Fig. 4.



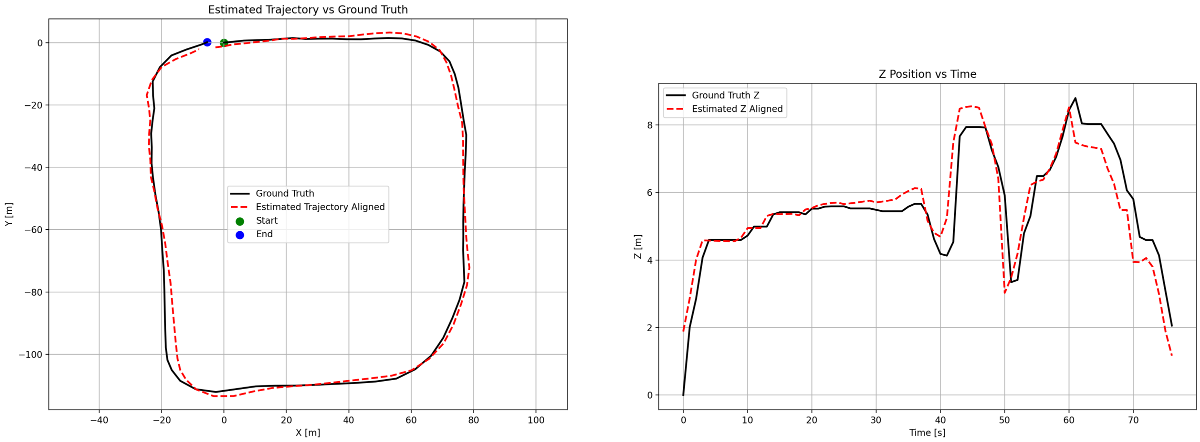
(a) Scenario 1.

(b) Scenario 2.

Figure 4: Merged LiDAR point clouds obtained using LiDAR-only ICP odometry.

15.2 Enhanced LiDAR Odometry with VO and EKF Initialization

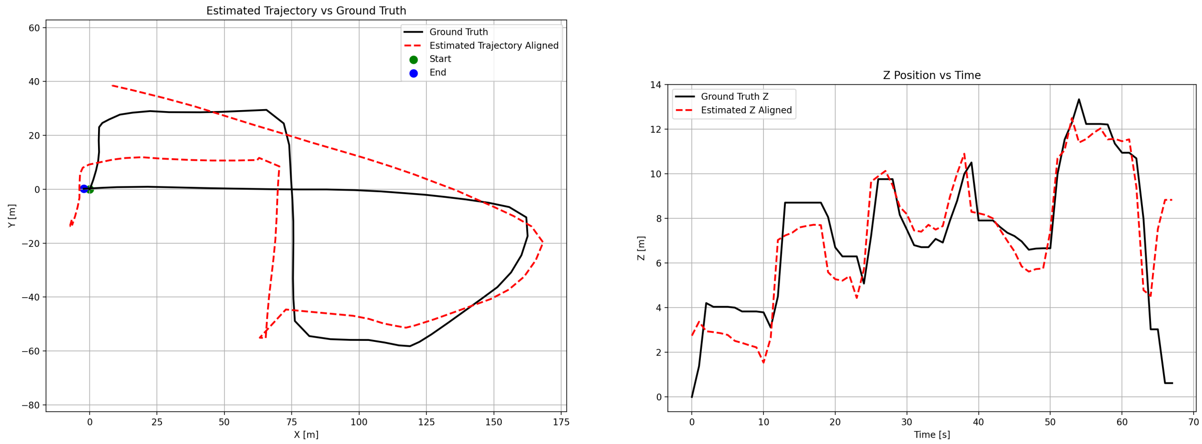
The estimated LiDAR trajectory is compared with the ground-truth trajectory after 4-DoF alignment.



(a) Estimated trajectory and ground truth.

(b) Trajectory comparison in z axis.

Figure 5: LiDAR odometry results after 4-DoF alignment.

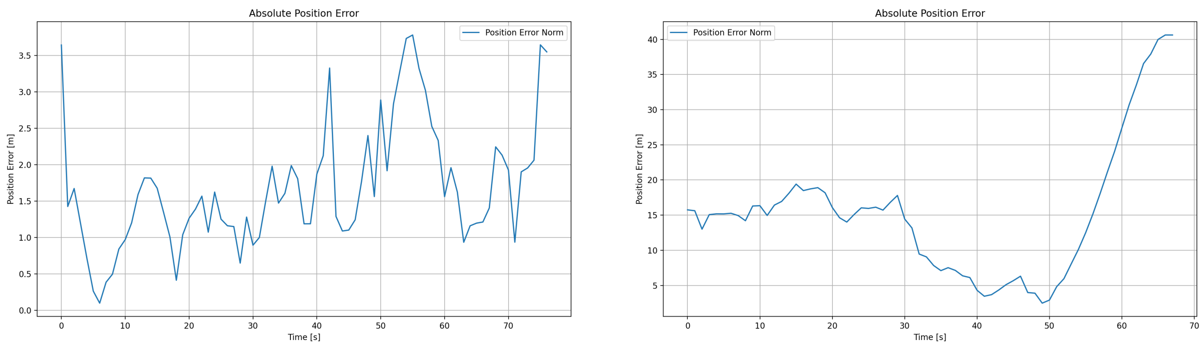


(a) Estimated trajectory and ground truth.

(b) Trajectory comparison in z axis.

Figure 6: LiDAR odometry results after 4-DoF alignment.

The position-error profiles are shown in Fig. 7.



(a) Scenario 1.

(b) Scenario 2.

Figure 7: Absolute position error over time.

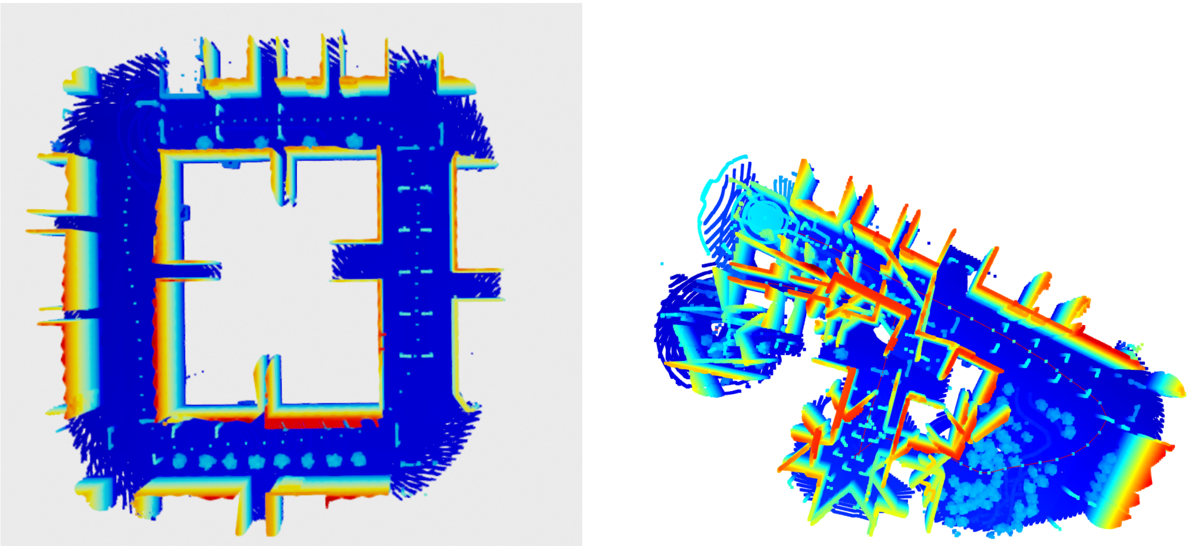
The quantitative localization results are summarized in Table below.

Table 1: performance metrics– Scenario 1.

Metric	Value	Unit
Alignment Yaw	2.946160	deg
ATE RMSE	1.884257	m
ATE MAE	1.681680	m
ATE Maximum	3.781535	m
X RMSE	1.245654	m
Y RMSE	1.179420	m
Z RMSE	0.779575	m
X MAE	0.986527	m
Y MAE	0.960435	m
Z MAE	0.511374	m
Yaw RMSE	6.320220	deg
Pitch RMSE	1.838404	deg
Roll RMSE	1.150946	deg
Yaw MAE	4.812395	deg
Pitch MAE	1.063993	deg
Roll MAE	0.854518	deg

Table 2: performance metrics– Scenario 2.

Metric	Value	Unit
Alignment Yaw	10.193172	deg
ATE RMSE	17.751059	m
ATE MAE	15.099129	m
ATE Maximum	40.621042	m
X RMSE	5.726239	m
Y RMSE	16.688460	m
Z RMSE	1.950793	m
X MAE	4.995098	m
Y MAE	13.676506	m
Z MAE	1.314381	m
Yaw RMSE	12.779068	deg
Pitch RMSE	4.162382	deg
Roll RMSE	1.601460	deg
Yaw MAE	10.561322	deg
Pitch MAE	2.404032	deg
Roll MAE	1.333960	deg



(a) Scenario 1.

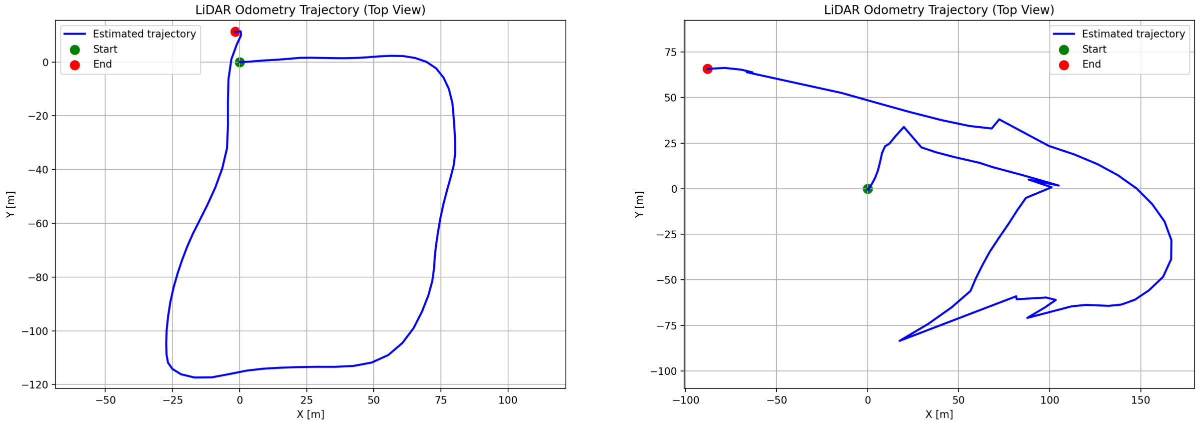
(b) Scenario 2.

Figure 8: Merged LiDAR point clouds obtained using the enhanced localization pipeline with EKF-based ICP initialization and visual-odometry fallback.

15.3 EKF Trajectory Results

The third experiment evaluates the trajectory produced by the EKF itself. This result is presented separately from the enhanced LiDAR odometry in order to analyze the behavior of the inertial state estimator independently.

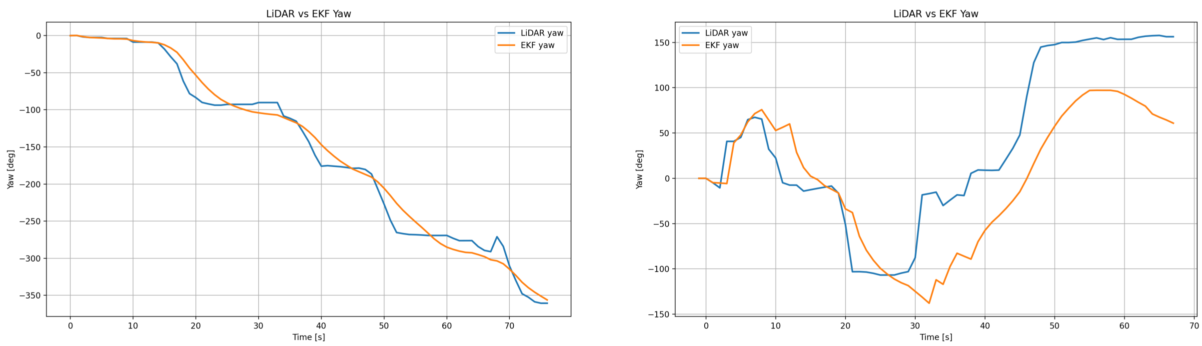
The EKF uses IMU measurements during the prediction stage and incorporates available pose measurements during the update stage. Its output provides a high-rate estimate of the vehicle position and orientation.



(a) Scenario 1.

(b) Scenario 2.

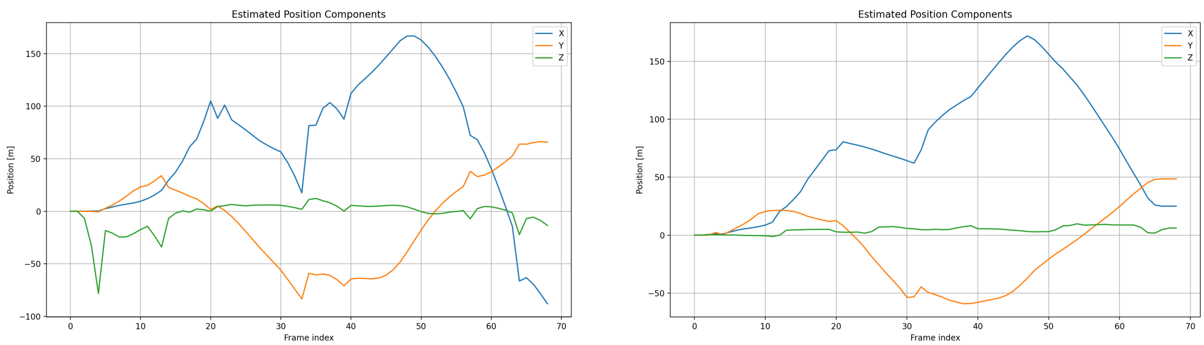
Figure 9: EKF trajectory compared with ground truth.



(a) Scenario 1.

(b) Scenario 2.

Figure 10: Comparison of LiDAR and EKF yaw estimates.



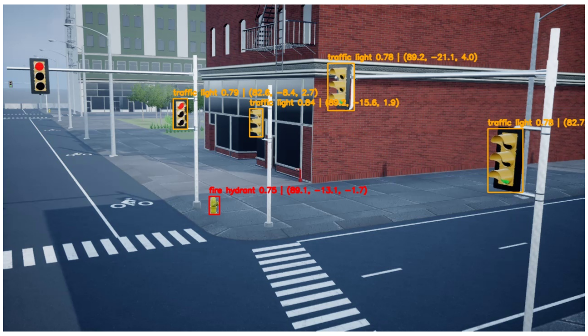
(a) EKF position estimates.

(b) LiDAR position estimates.

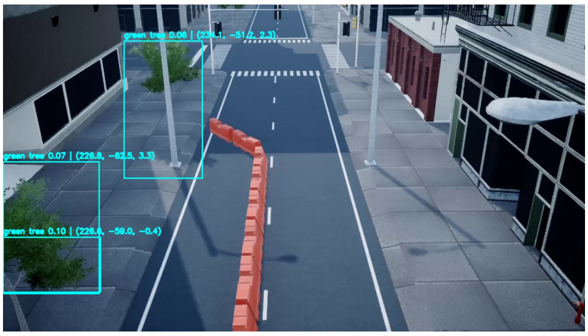
Figure 11: Comparison of LiDAR and EKF position estimates along the X, Y, and Z axes.

16 Environment Recognition Results

The environment-recognition pipeline successfully combines image-based object detection with LiDAR-based metric localization.



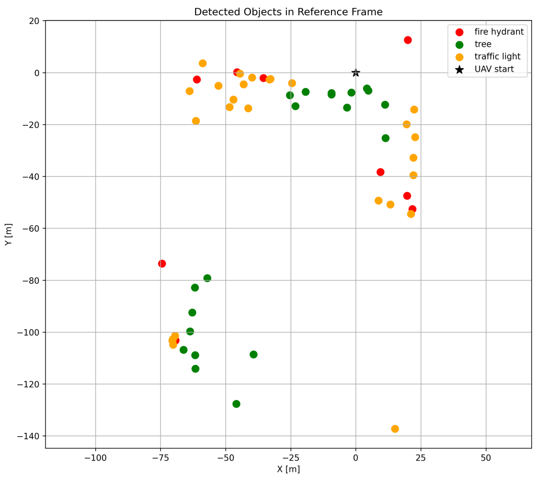
(a) Object detection in the first frame.



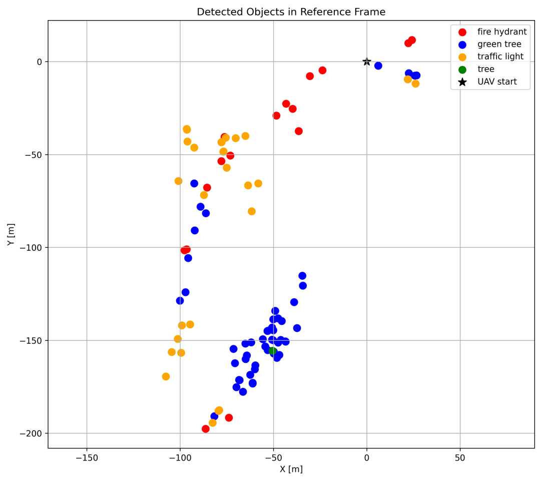
(b) Object detection in the second frame.

Figure 12: Examples of detected environmental objects and estimated positions.

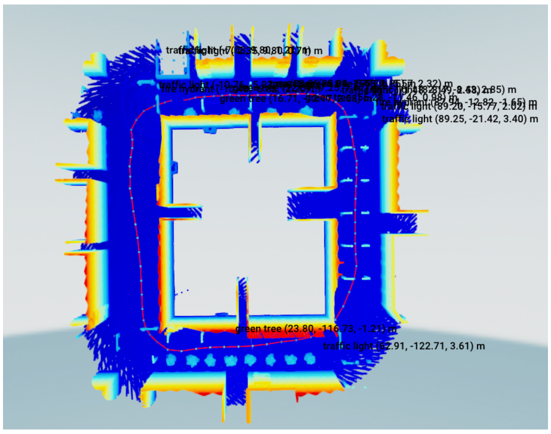
The resulting environment map is shown in Figure below.



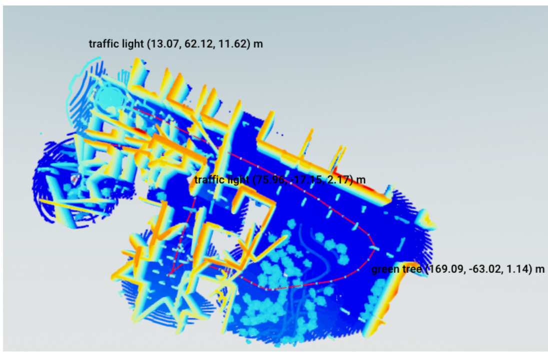
(a) 2D map – Scenario 1.



(b) 2D map – Scenario 2.



(c) 3D map – Scenario 1.



(d) 3D map – Scenario 2.

Figure 13: 2D and 3D representations of the detected objects in the global reference frame.

17 Conclusion

This project presented a complete multi-sensor pipeline for relative localization and environment recognition of an autonomous aerial vehicle.

For localization, consecutive LiDAR point clouds were registered using multi-scale ICP. Point-to-plane registration was used whenever reliable surface normals were available, with point-to-point ICP used as a fallback. Visual odometry based on SIFT features, essential-matrix estimation, and relative pose recovery was incorporated as a complementary source of motion information and as a fallback when LiDAR registration was rejected.

An error-state EKF was also implemented to integrate high-rate IMU measurements with pose observations. The filter estimates position, velocity, orientation, accelerometer bias, and gyroscope bias. Its prediction provides a high-rate motion estimate and can also be used to initialize ICP.

For environment recognition, YOLO-World was used to detect trees, traffic lights, and fire hydrants. The synchronized LiDAR point cloud was projected onto the camera image using the calibrated camera model. LiDAR points inside the detected bounding boxes were then filtered using a robust median-depth procedure to obtain metric three-dimensional object positions. Finally, these positions were transformed into the global reference frame using the estimated vehicle pose and visualized as an environment map.

Overall, the project demonstrates how LiDAR, camera, and IMU measurements can be integrated to address both localization and semantic environment understanding in an autonomous robotic system.

References

1. BESL, Paul J. and Neil D. MCKAY. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1992, vol. 14, no. 2, pp. 239–256. DOI 10.1109/34.121791.

2. LOWE, David G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*. 2004, vol. 60, no. 2, pp. 91–110. DOI 10.1023/B:VISI.0000029664.99615.94.

3. NISTÉR, David. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2004, vol. 26, no. 6, pp. 756–777. DOI 10.1109/TPAMI.2004.17.

4. ZHANG, Ji and Sanjiv SINGH. LOAM: Lidar odometry and mapping in real-time. In: *Proceedings of Robotics: Science and Systems*. Berkeley, CA, USA, 2014. DOI 10.15607/RSS.2014.X.007.

5. SOLA, Joan, Jérémie DERAY and Dinesh ATCHUTHAN. A micro Lie theory for state estimation in robotics. 2018. arXiv:1812.01537. Available from: https://arxiv.org/abs/1812.01537.

6. CHENG, Tianheng, Lin SONG, Yixiao GE, Wenyu LIU, Xinggang WANG and Ying SHAN. YOLO-World: Real-Time Open-Vocabulary Object Detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 16901–16911. DOI 10.1109/CVPR52733.2024.01599.