UNIVERSITÉ CLERMONT AUVERGNE

École Universitaire de Physique et d'Ingénierie

An AprilTag-Based Approach to Trajectory Recording and Relocalization for Mobile Robots

Second-year Master's project

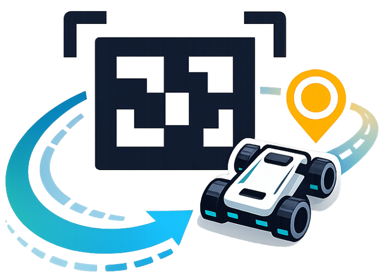
Automation, Robotics track: Artificial Perception and Robotics

Authors:
Joao Pedro MARTINS DO LAGO REIS
Yann Kelvem DA SILVA RAMOS

Supervisor:
Dr. Sébastien LENGAGNE

Evaluator:
Dr. Aufrère ROMUALD

Defense Date:
March 02, 2026



Aubière, Clermont-Ferrand

An AprilTag-Based Approach to Trajectory Recording and Relocalization for Mobile Robots

Abstract

This work presents a ROS 2 pipeline for trajectory recording and trajectory-relative relocalization using AprilTags on the AgileX LIMO platform. The method processes recorded rosbag2 data offline to estimate a marker map and a reference camera trajectory through nonlinear least-squares optimization. During relocalization, new pose estimates are compared with the stored reference in order to compute the distance to the nearest trajectory sample D and the heading deviation relative to the local path direction ϕ.

The pipeline was evaluated using synthetic data, simulation, and real experiments. In simulation, comparison with ground truth yielded a mean trajectory error of 0.051 m and a mean tag-position error of 0.087 m. In real recording experiments, map consistency was assessed through tape measurements from one reference tag to 20 other tags, resulting in absolute errors of 0.001 m, 0.099 m, and 0.039 m for minimum, maximum, and mean values, respectively. In real relocalization tests, the discrepancy between estimated and manually measured distances ranged from 0.0033 m to 0.0102 m across three off-trajectory cases. An additional real recording dataset was processed under a different traversal configuration to further evaluate the estimation procedure.

Keywords: ROS 2, AprilTags, trajectory recording, relocalization, nonlinear least squares, pose estimation, marker map, mobile robotics

Contents

1 Introduction **2**

2 System Requirements **3**

2.1 Operational Context 3

3 System Architecture **5**

3.1 Recording Phase 5

3.2 Relocalization Phase 8

3.2.1 Offline evaluation 9

3.2.2 Online execution 11

4 Experimental Validation **12**

4.1 Recording Phase Validation under Controlled Conditions 12

4.2 Recording Phase Validation in Simulation 13

4.3 Relocalization Validation in Simulation 15

4.4 Recording Phase Validation on the Real Platform 15

4.5 Relocalization Validation on the Real Platform 16

4.6 Second Real Recording Experiment 17

5 Conclusion **18**

6 Difficulties and Use of Artificial Intelligence **19**

List of Appendices **21**

List of Appendices **21**

A Tag Distance Validation Table **21**

B Project Website QR Code **21**

1 Introduction

Mobile robots operating in indoor laboratory environments cannot rely on satellite-based positioning systems and must instead estimate their pose using locally defined spatial references. Reliable localization therefore remains a fundamental requirement for controlled experimentation and autonomous operation, particularly in structured environments where repeatability and spatial consistency are expected [1]. Within this context, the objective of this project is to design a two-phase framework in which a mobile robot first records a reference trajectory during an initial run and subsequently estimates its pose with respect to that same reference.

The purpose of this relocalization stage is to provide the geometric quantities required for the later implementation of a trajectory tracking control law. The scope of the work is thus not the design of the controller itself, but the establishment of a consistent and reproducible spatial framework that enables such control strategies to be developed and evaluated. Thus, the creation of this framework aims to establish a reusable scenario for practical work by students at Polytech, in which they can record a trajectory with the robot and, after positioning it outside of that trajectory, use location data to implement a control law that allows the robot to follow the previously recorded trajectory..

Although a trajectory can be recorded during an initial traversal, this action alone does not ensure that the same spatial path can be consistently referenced at a later time. In practice, odometry based motion estimation accumulates error over time due to sensor noise and unmodeled effects such as wheel slip, which leads to drift even in repeated runs [2]. Consequently, successive executions of the same nominal motion may produce trajectories that are not directly comparable in a common frame. To mitigate this, the recorded motion must be associated with stable spatial references so that the robot can reconstruct a consistent relationship between its current pose and the previously recorded path, which is a recurring requirement in mobile robot localization pipelines [1]. The technical problem therefore lies in defining a trajectory representation that remains coherent over time and in establishing a method to estimate the robots pose relative to that reference during subsequent runs.

Based on these requirements, this work implements a two-phase system with an offline reconstruction stage and a relocalization stage designed for both offline analysis and online execution. Data collected during the initial traversal are processed offline to reconstruct the reference trajectory and to estimate an optimized marker map, since this step involves iterative estimation procedures with higher computational cost. During subsequent runs, relocalization relies on marker detection and pose estimation to compute the robot pose with respect to the reconstructed reference, which makes online deployment feasible.

The system was developed in ROS 2 Humble and validated on the AgileX LIMO platform. AprilTag v3 was selected based on recent evaluations reporting improved detection performance in practical conditions, and the tag36h11 family was used due to its widely adopted coding scheme with a large identifier set and error tolerance properties [3], [4]. As a result, the system provides the reference trajectory and the relocalization outputs needed for trajectory tracking, and it is designed to be coupled with a controller developed as part of subsequent work.

2 System Requirements

2.1 Operational Context

The system is intended for controlled experimental environments in which AprilTag fiducial markers are deployed in advance. In this setting, the robot does not rely on satellite-based positioning and instead uses onboard perception together with a predefined marker reference to support trajectory recording and subsequent relocalization.

Two workflows are provided to support development and validation:

- a real-robot workflow validated on the AgileX LIMO platform,
- a simulation workflow used for repeatable testing and dataset generation.

The implementation is developed in ROS 2 Humble and organized as a modular workspace. The requirements listed in the following subsections specify the expected system functions, delivery constraints, and operational limitations.

Table 2.0: Functional Requirements

ID	Requirement
FR1	The system shall provide workflows for recording robot motion and perception data into rosbag2 datasets for offline processing.
FR2	The system shall process recorded rosbag2 datasets offline to generate a reference trajectory and a spatial map of detected AprilTags.
FR3	The offline processing stage shall export structured output files, including trajectory CSV and tag map YAML/CSV formats.
FR4	The system shall provide an online relocalization node capable of loading a reference trajectory, subscribing to a live pose topic, computing nearest-point distance and heading deviation, and publishing these quantities for controller integration.
FR5	The system shall publish visualization-compatible ROS topics enabling RViz display of reference trajectory, current pose, nearest reference pose, and relocalization error indicators.
FR6	The system shall provide launch files and profile scripts supporting both real and simulation workflows.
FR7	The system shall include documented execution procedures ensuring reproducibility from recorded datasets.

Table 2.1: Non-Functional Requirements

ID	Requirement
NFR1	Offline reconstruction shall be executable independently of live robot operation.
NFR2	Online relocalization shall operate within computational limits compatible with real-time ROS 2 execution.
NFR3	The architecture shall maintain separation between perception, offline reconstruction, and online relocalization components.
NFR4	The system shall ensure consistency of coordinate frames within the ROS 2 TF structure.
NFR5	Dataset files shall remain external to version control to maintain repository portability.

Table 2.2: Project Deliverables and Constraints

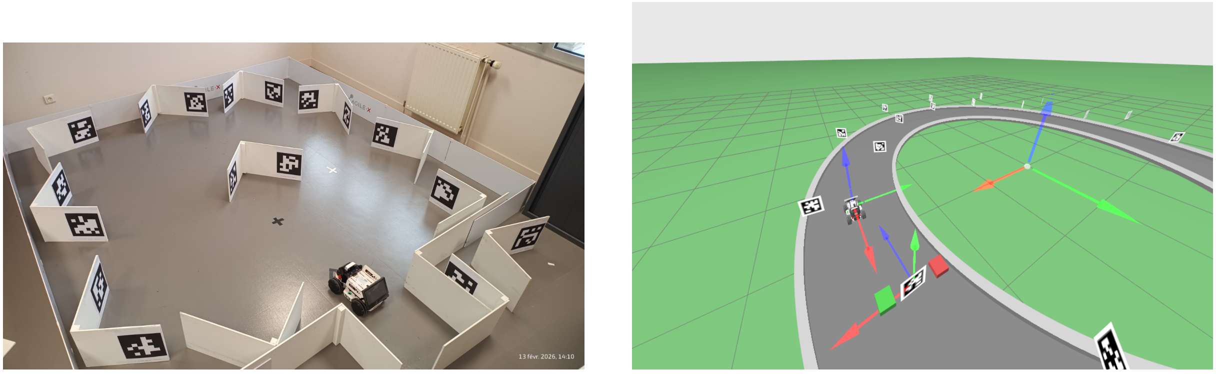
ID	Requirement
PR1	The project shall be delivered as a version-controlled ROS 2 workspace repository containing source code, launch files, scripts, and documentation.
PR2	The repository shall include structured workflows for both real and simulation execution.
C1	The system shall be implemented in ROS 2 Humble.
C2	The perception pipeline shall use AprilTag v3 with the tag36h11 family.
C3	Experimental validation shall be conducted on the AgileX LIMO platform.
C4	The system shall not include full autonomous navigation, SLAM of unknown environments, obstacle avoidance, or mission-level planning.

Table 2.3: Technical Limitations

ID	Limitation
TL1	Accurate and stable pose estimation assumes that at least two AprilTags are simultaneously visible in the camera field of view. While pose can be computed from a single detection, multi-tag observations improve numerical stability and reduce sensitivity to noise and calibration errors.
TL2	Localization precision depends on camera calibration quality, tag size, distance to the tags, and image resolution. The system does not guarantee fixed absolute accuracy under arbitrary environmental conditions.

3 System Architecture

The system is implemented on the AgileX LIMO wheeled platform and uses a calibrated monocular RGB camera to observe AprilTags placed at fixed locations in the workspace. Figures 3.1 illustrate the experimental setup in simulation and in its corresponding real-world deployment. An initial traversal is recorded as a rosbag2 dataset so that AprilTag detections and camera intrinsics can be processed offline to build a geometric reference.



(a) Real-world experimental setup with AgileX LIMO and fixed AprilTags.

(b) Simulation environment used for dataset generation and validation.

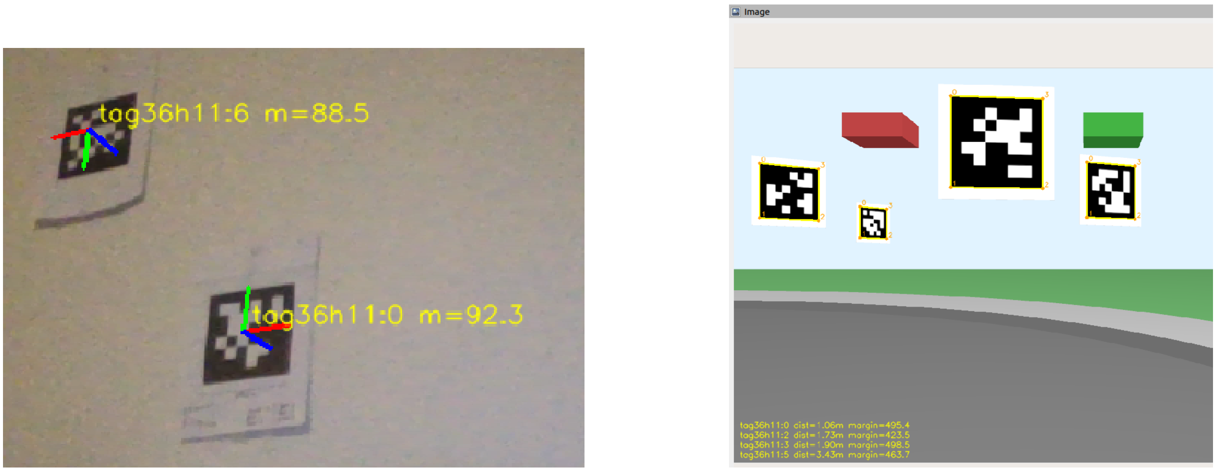
Figure 3.1. Physical and simulated setups used to evaluate the proposed architecture.

The workflow is exercised in two settings, namely a real-robot setup and a simulation environment designed to replicate the perception pipeline under controlled conditions. Although the geometric layouts and operational details differ between scenarios, both settings rely on a rigidly mounted camera observing static AprilTags and follow the same estimation and optimization procedures. The architecture is therefore presented as two consecutive phases. In the recording phase, the traversal is processed to estimate an optimized tag map and a reference trajectory expressed consistently with that map. In the relocalization phase, new pose estimates are compared to the stored reference in order to compute the distance to the trajectory and the heading deviation relative to the local path direction. These quantities are published as outputs and may be used by a trajectory tracking controller implemented outside the scope of this work.

Figure 3.2 illustrates representative camera frames acquired during both real and simulated executions. Detected markers are outlined and identified, and their estimated poses are expressed with respect to the camera frame. These per-frame relative transformations constitute the primary measurements used in the recording phase. Although each detection provides a six-degree-of-freedom estimate, the resulting poses are inherently local and subject to noise. As a result, consistent global structure must be recovered by integrating observations across multiple viewpoints.

3.1 Recording Phase

The recording phase processes the data collected during the initial traversal in order to construct a reference tag map and a corresponding camera trajectory. Figure 3.3 summarizes the computational steps involved. Each block in the pipeline corresponds to a specific estimation stage, which is detailed in the discussion below.



(a) Example of AprilTag detection in the real environment.

(b) AprilTag detection in the simulated environment.

Figure 3.2. Representative detection, each detected tag yields a camera-relative pose estimate obtained through a PnP formulation.

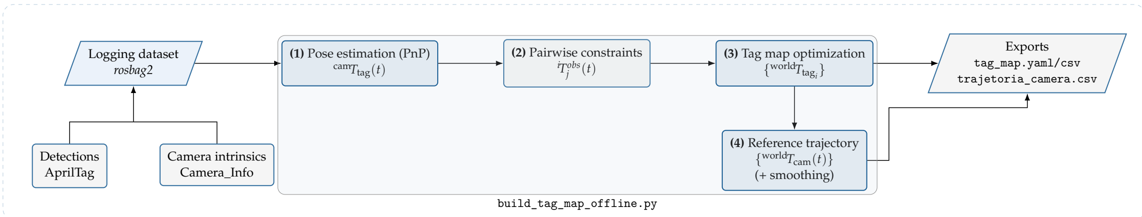


Figure 3.3. Architecture of the offline recording pipeline.

As indicated in block **(1)** of Figure 3.3, the first step of the recording phase estimates camera-to-tag relative poses by solving a Perspective-n-Point problem under calibrated intrinsics [5] using the logged detections.

To construct a globally consistent representation of the environment, the local camera-to-tag measurements must be expressed in a common reference frame. In this work, rigid-body transformations are denoted by ${}^{A}T_{B} \in SE(3)$, representing the pose of frame B with respect to frame A [6]. Each detection provides a relative transformation ${}^{\text{cam}}T_{\text{tag}_i}(t)$ at time t. Since the camera pose varies along the traversal, each measurement relates a tag pose to a different camera coordinate system. If one marker is fixed as the map reference (gauge anchor), the camera pose follows from the inverse of the observed camera-to-tag transformation, as written in:

$$ {}^{\text{world}}T_{\text{cam}}(t) = {}^{\text{world}}T_{\text{tag}_0} \left({}^{\text{cam}}T_{\text{tag}_0}(t)\right)^{-1} \tag{3.1} $$

Equation (3.1) highlights both the utility and the limitation of single-frame detections. The optimized tag map provides a globally consistent geometric reference for expressing camera poses across time. In this work, the map frame (denoted as world) is internally defined by the selected gauge anchor rather than by an absolute ground-truth origin. The tag map is therefore estimated as the set of poses $\{{}^{\text{world}}T_{\text{tag}_i}\}$ that best explains the pairwise constraints accumulated along the traversal. Whenever the anchor tag is visible, a world-referenced camera pose estimate follows immediately from Eq. (3.1).

As illustrated in block **(2)**, frames containing multiple visible tags allow the construction of pairwise constraints. When two tags i and j are simultaneously detected, their relative transformation can be computed from the camera-centered estimates. Using the composition and inversion properties of rigid transformations [5], [6], the observed relation is given by :

$$ {}^{i}T_{j}^{obs}(t) = \left({}^{\mathrm{cam}}T_{\mathrm{tag}_i}(t)\right)^{-1}\, {}^{\mathrm{cam}}T_{\mathrm{tag}_j}(t) \tag{3.2} $$

Each observation in Eq. (3.2) induces a constraint between the unknown world poses ${}^{\mathrm{world}}T_{\mathrm{tag}_i}$ and ${}^{\mathrm{world}}T_{\mathrm{tag}_j}$. In an ideal noise-free setting, the observed relative pose would match the relative pose implied by the world estimates :

$$ {}^{i}T_{j}^{pred} = \left({}^{\mathrm{world}}T_{\mathrm{tag}_i}\right)^{-1}\, {}^{\mathrm{world}}T_{\mathrm{tag}_j} \tag{3.3} $$

In practice, detection uncertainty, calibration inaccuracies, and viewing geometry variations lead to inconsistencies across multiple measurements of the same tag pair. Collecting all pairwise constraints along the traversal yields a network of relative pose relations between tags.

As shown in block **(3)**, estimating a consistent global map from such constraints is posed as a nonlinear least-squares problem, following the general principles used in geometric least-squares estimation and graph-based formulations [7], [8]. Let $\mathcal{E}$ denote the set of observed tag pairs. The tag poses are obtained by minimizing the sum of residuals over $\mathcal{E}$, as stated in :

$$ \min_{\{{}^{\mathrm{world}}T_{\mathrm{tag}_i}\}} \sum_{(i,j)\in\mathcal{E}} \left\| \mathrm{err}_{ij} \right\|^2 \tag{3.4} $$

In Eq. (3.4), err_{ij} quantifies the discrepancy between the observed relative transformation in Eq. (3.2) and the relative transformation predicted by the current map estimate in Eq. (3.3). In practice, this discrepancy is decomposed into translational and rotational components.

To make the residual in Eq. (3.4) explicit, the predicted and observed relative transformations are written as ${}^{i}T_{j}^{pred} = (R_{ij}^{pred}, t_{ij}^{pred})$ and ${}^{i}T_{j}^{obs} = (R_{ij}^{obs}, t_{ij}^{obs})$. The translational component is defined as :

$$ r_{t,ij} = t_{ij}^{pred} - t_{ij}^{obs} \tag{3.5} $$

For the rotational component, the relative rotation between prediction and observation is computed by composition,

$$ R_{err,ij} = \left(R_{ij}^{pred}\right)^{\top} R_{ij}^{obs} \tag{3.6} $$

The matrix $R_{err,ij}$ is then mapped to a minimal three-parameter representation $r_{R,ij} \in \mathbb{R}^3$ using an axis–angle parametrization, which is consistent with the pose representation adopted for optimization [6]. The complete residual used in Eq. (3.4) is formed by stacking the translational and rotational components, $\mathrm{err}_{ij} = [\, r_{t,ij}^{\top} \;\; r_{R,ij}^{\top} \,]^{\top}$.

Each tag pose ${}^{\mathrm{world}}T_{\mathrm{tag}_i}$ is parameterized using a minimal six-dimensional representation consisting of a three-parameter rotation vector and a three-dimensional translation. The optimization variables therefore correspond to the stacked pose parameters of all non-anchored tags.

The minimization in Eq. (3.4) can be generalized to include a robust cost function,

$$ \min_{\{{}^{\mathrm{world}}T_{\mathrm{tag}_i}\}} \sum_{(i,j)\in\mathcal{E}} \rho\left(\left\| \mathrm{err}_{ij} \right\|^2\right) \tag{3.7} $$

where $\rho(\cdot)$ denotes a robust loss function. In contrast to a purely quadratic penalty, robust losses reduce the influence of large residuals that may arise from occasional misdetections, partial occlusions, or inaccurate corner localization. Such formulations belong to the class of M-estimators commonly used in robust least-squares estimation [9].

In this implementation, a soft-ℓ_1 loss is employed. This loss behaves quadratically for small residuals, preserving the local convergence properties of standard least squares, and transitions to approximately linear growth for larger errors. As a result, residuals with large magnitude contribute less aggressively to the total cost, preventing them from dominating the optimization process. This choice is consistent with the presence of occasional inconsistent tag observations while maintaining sensitivity to well-conditioned measurements.

The solution is obtained iteratively by updating the pose parameters so as to decrease the total residual error. One tag pose is held fixed throughout the optimization in order to eliminate the global gauge freedom discussed previously. The nonlinear least-squares problem is solved using an iterative trust-region method, as implemented in standard numerical optimization libraries. At each iteration, the pose parameters are updated so as to reduce the total robust cost defined in Eq. (3.7). This procedure follows the classical GaussNewton framework for nonlinear least squares, adapted to the chosen minimal pose parametrization [7].

Finally, as indicated in block **(4)**, the optimized tag map is used to reconstruct the camera trajectory. For each observation of a tag i at time t, a world-referenced camera pose estimate is obtained according to:

$$
{}^{\text{world}}T^{(i)}_{\text{cam}}(t) = {}^{\text{world}}T_{\text{tag}_i} \left({}^{\text{cam}}T_{\text{tag}_i}(t)\right)^{-1} \tag{3.8}
$$

When multiple tags are detected in the same frame, several estimates $\{{}^{\text{world}}T^{(i)}_{\text{cam}}(t)\}$ are obtained. In the absence of noise, these estimates would coincide; in practice, small discrepancies arise due to measurement uncertainty and residual map errors. A single pose per frame is obtained either by averaging the rigid transformations or by solving a local nonlinear least-squares problem that minimizes the residual error between predicted and observed camera-to-tag transformations.

The resulting sequence $\{{}^{\text{world}}T_{\text{cam}}(t)\}$ defines the reference trajectory recorded during the initial traversal. Since frame-wise estimation may introduce high-frequency variations, the trajectory is optionally regularized in time prior to export.

The temporal regularization is applied directly to the discrete sequence $\{{}^{\text{world}}T_{\text{cam}}(t)\}$ and penalizes large pose differences between consecutive timestamps while preserving the spatial structure imposed by the optimized tag map. This operation improves temporal consistency without altering the underlying map estimation. The final smoothed trajectory constitutes the geometric reference used in the subsequent *relocalization phase*, where live pose estimates are compared against this stored reference in order to compute distance and heading deviations.

3.2 Relocalization Phase

The relocalization phase evaluates the geometric deviation between an estimated camera pose and the optimized reference trajectory obtained in the recording phase. In this stage, the reference trajectory $\{{}^{\text{world}}T_{\text{cam}}(t_k)\}$ is assumed to be fixed and previously computed, while the tag map may be used upstream by the localization source that provides live poses. Figure 3.4 summarizes the computational structure adopted for relocalization, separating offline analysis from online execution.

In contrast to the recording phase, no map optimization or trajectory reconstruction is performed here. Instead, the system receives a query pose ${}^{\text{world}}T^{q}_{\text{cam}} \in SE(3)$ and computes its geometric relationship with respect to the discrete reference trajectory. The resulting quantities are a distance-to-trajectory metric and a heading deviation, which are defined formally in the following subsection.

From a methodological standpoint, the online relocalization node is intentionally limited to geometric comparison against a fixed reference trajectory. Its inputs are the stored trajectory $\mathcal{T}_{ref}$ and a live query pose stream; the map YAML is not consumed directly in this stage. The map remains part of the upstream localization source that estimates the live pose, while the relocalization node computes only the association and error quantities defined by Eqs. (3.12), (3.13), and (3.15). This separation preserves a clear boundary between pose estimation and trajectory-deviation evaluation.

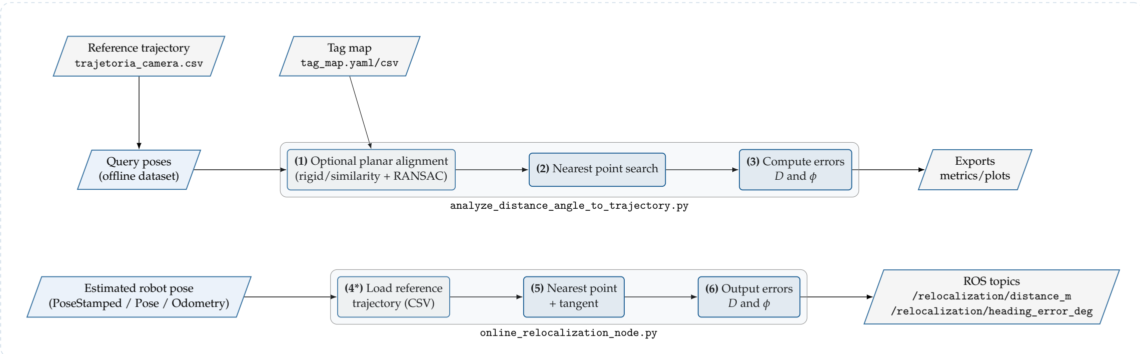


Figure 3.4. Architecture of the relocalization pipeline with separated offline analysis and online execution.

3.2.1 Offline evaluation

As presented in block **(1)**, the first operation in the relocalization workflow optionally performs a planar alignment between the query dataset and the stored reference trajectory. This step compensates for possible coordinate discrepancies arising from independent reconstructions or scale differences during map generation.

Let $\{p^{ref}_k\} \subset \mathbb{R}^2$ denote the planar positions extracted from the reference trajectory $\{{}^{\text{world}}T_{\text{cam}}(t_k)\}$ and let $\{p^{q}_{\ell}\} \subset \mathbb{R}^2$ denote the planar positions of the query dataset. When a set of common tags is available in both datasets, a transformation is estimated so that the transformed query positions best match the reference positions in a least-squares sense.

In the most general case considered here, the alignment is modeled as a planar similarity transformation,

$$p^{q,aligned}_{\ell} = sRp^{q}_{\ell} + t \tag{3.9}$$

where $R \in SO(2)$ represents a planar rotation, $t \in \mathbb{R}^2$ is a translation vector, and $s \in \mathbb{R}^+$ is a uniform scale factor. When scale correction is not required, the model reduces to a rigid transformation with $s = 1$.

The optimal parameters (R, t, s) are obtained by minimizing the squared alignment residual over the set of matched tag correspondences:

$$\min_{R,t,s} \sum_{m \in C} \left\| p^{ref}_{m} - \left(sRp^{q}_{m} + t\right) \right\|^2 \tag{3.10}$$

where $\mathcal{C}$ denotes the set of tag identifiers observed in both datasets. Problems of this form admit closed-form solutions based on singular value decomposition and correspond to the classical orthogonal Procrustes formulation [10].

In practice, inconsistencies may arise due to partial overlap or incorrect detections. To reduce the influence of such outliers, a RANSAC scheme [11] is applied prior to the final least-squares estimation. The transformation computed from inlier correspondences is then applied to all query poses. If no valid correspondences are available, or if alignment is explicitly disabled, the query poses are evaluated in their original coordinate frame. Once both trajectories are expressed in a common planar reference, the relocalization task reduces to determining the geometric association between each query pose and the reference trajectory.

As illustrated in block **(2)**, once the query and reference trajectories are expressed in a common planar frame, the relocalization problem reduces to establishing a geometric association between a query pose and the discrete reference trajectory.

Let the reference trajectory exported in the recording phase be represented by the set:

$$\mathcal{T}_{ref} = \left\{ {}^{\text{world}}T_{\text{cam}}(t_k) \right\}_{k=1}^{N} \tag{3.11}$$

and let $p_k^{ref} \in \mathbb{R}^2$ denote the planar position extracted from each pose. Given a query pose ${}^{\text{world}}T_{\text{cam}}^{q}$ with planar position $p^q \in \mathbb{R}^2$, the nearest reference association is defined as:

$$k^* = \arg \min_{k \in \{1,\ldots,N\}} \left\| p^q - p_k^{ref} \right\|_2 \tag{3.12}$$

Equation (3.12) defines the reference sample whose planar position is closest to the query position in the Euclidean sense. The selected index k^* serves as the geometric anchor for evaluating the deviation between the live pose and the stored reference trajectory. Based on this association, the relocalization phase proceeds to quantify the positional and angular discrepancy with respect to the selected reference pose.

As presented in block **(3)**, once the nearest association index k^* is available, the relocalization pipeline evaluates two geometric error metrics defined with respect to the associated reference sample.

The positional error is defined as the Euclidean distance between the query planar position $p^q \in \mathbb{R}^2$ and the associated reference planar position $p_{k^*}^{ref} \in \mathbb{R}^2$:

$$D = \left\| p^q - p_{k^*}^{ref} \right\|_2 \tag{3.13}$$

To compute an angular error, a reference direction is defined locally along the discrete reference trajectory. Let $p_k^{ref} = (x_k^{ref}, z_k^{ref})$ denote the planar reference samples. The reference heading at index k is obtained from the local tangent using a centered difference:

$$\theta_k^{ref} = \operatorname{atan2}\left(z_{k+1}^{ref} - z_{k-1}^{ref},\ x_{k+1}^{ref} - x_{k-1}^{ref} \right) \tag{3.14}$$

with a one-sided variant at the endpoints. Let θ^q denote the query planar heading extracted from the query pose orientation in the same plane. The heading error is then defined as:

$$\phi = \operatorname{wrap}_{(-\pi,\pi]}\left(\theta^q - \theta_{k^*}^{ref} \right) \tag{3.15}$$

where $\operatorname{wrap}_{(-\pi,\pi]}(\cdot)$ maps the angular difference to the principal interval.

Equations (3.13)–(3.15) specify the outputs computed at this stage for each evaluated query pose. These values are then forwarded to the reporting stage in the offline workflow.

As presented in block **(3)** of the offline branch, the computed values of D and ϕ are exported for quantitative assessment. Given a sequence of query poses $\{{}^{\text{world}}T^{q}_{\text{cam}}(t_\ell)\}$, Equations (3.13) and (3.15) are evaluated for each timestamp. This produces two discrete time series:

$$\{D(t_\ell)\}_{\ell=1}^{M} \qquad \{\phi(t_\ell)\}_{\ell=1}^{M} \tag{3.16}$$

where M denotes the number of evaluated query samples.

These sequences are stored together with the associated timestamps and reference indices, enabling post-processing and statistical evaluation. Summary quantities such as mean, median, percentile levels, and maximum deviation are derived directly from the exported series. Additionally, planar visualizations are generated to illustrate the spatial relationship between the query samples and the reference trajectory. This offline reporting stage provides a quantitative characterization of relocalization performance over an entire traversal.

3.2.2 Online execution

As depicted in block **(4*)**, the online branch starts by loading the reference trajectory generated in the recording phase. The file `trajetoria_camera.csv` provides the discrete sequence $\mathcal{T}_{ref} = \{{}^{\text{world}}T_{\text{cam}}(t_k)\}_{k=1}^{N}$.

In online execution, $\mathcal{T}_{ref}$ is treated as fixed data. The node parses the file once and stores the planar reference samples $\{p_k^{ref}\}_{k=1}^{N}$ together with their associated local headings $\{\theta_k^{ref}\}_{k=1}^{N}$, computed from the discrete trajectory tangent as in Eq. (3.14). This initialization establishes the reference set required to evaluate incoming pose estimates without modifying the stored trajectory. Once the reference trajectory is available in memory, each incoming pose is processed by evaluating the nearest-neighbor association in Eq. (3.12) and subsequently computing D and ϕ.

As presented in block **(5)**, the online node performs a nearest-sample association between the current pose estimate and the fixed reference set. At each time t, the node receives a query pose ${}^{\text{world}}T^{q}_{\text{cam}}(t)$ and extracts its planar position $p^q(t)$. The association index is then obtained by directly applying the nearest-neighbor criterion in Eq. (3.12), yielding:

$$k^*(t) = \arg\min_{k\in\{1,\ldots,N\}} \left\| p^q(t) - p_k^{ref} \right\|_2 \tag{3.17}$$

The reference sample indexed by $k^*(t)$ provides the corresponding stored heading $\theta^{ref}_{k^*(t)}$, which anchors the frame-by-frame evaluation of the geometric errors defined in Eqs. (3.13) and (3.15). The resulting association $\left(p^q(t), p^{ref}_{k^*(t)}\right)$ is therefore the immediate input to the error output stage described next.

As summarized in block **(6)**, the online execution evaluates the geometric errors for each incoming pose and makes them available as real-time outputs. Given the association index $k^*(t)$, the positional and angular deviations are computed. These quantities are evaluated independently at each time step and therefore define two time-varying signals, $D(t)$ and $\phi(t)$, describing the instantaneous geometric discrepancy between the current pose estimate and the stored reference trajectory.

The online node publishes these values as scalar outputs, together with visualization elements that indicate the current pose, the associated reference sample, and the geometric connection between them. This structure enables direct inspection of relocalization behavior during execution and provides the signals required for potential control integration. The experimental section evaluates these outputs quantitatively by analyzing their temporal evolution under different traversal conditions, thereby assessing the consistency between live pose estimates and the reference trajectory established during the recording phase.

4 Experimental Validation

4.1 Recording Phase Validation under Controlled Conditions

Before integrating the relocalization stage in simulation and real-world experiments, the map optimization procedure described in Section 3.1 was validated under controlled conditions. The objective of this validation was to isolate the nonlinear least-squares formulation and verify its ability to recover consistent tag poses from noisy relative observations.

A synthetic dataset was constructed in which tag positions were manually defined in a known global frame and camera poses were sampled along a reference trajectory. From this ground-truth configuration, camera-to-tag relative observations were generated from rigid transformations and subsequently perturbed by controlled noise injection. This setup enabled direct comparison between the optimized tag poses and the known ground-truth solution, allowing quantitative evaluation of estimation error.

Initially, the system was evaluated using noise-free detections. Under this condition, the residual error was zero within numerical precision, indicating consistency between the optimized solution and the ground-truth configuration. The corresponding error statistics are reported in Table 4.1.

Metric	Mean Error	Minimum	Maximum
Camera Trajectory	0.0000 m / 0.00°	0.0000 m / 0.00°	0.0000 m / 0.00°
Tag Poses	0.0000 m / 0.00°	0.0000 m / 0.00°	0.0000 m / 0.00°

Table 4.1. Residual errors in the ideal scenario (no noise).

Figure 4.1 illustrates the alignment between the optimized solution and the ground-truth configuration. The estimated tag poses coincide with the reference positions, and the reconstructed camera trajectory overlaps the ground-truth circular path.

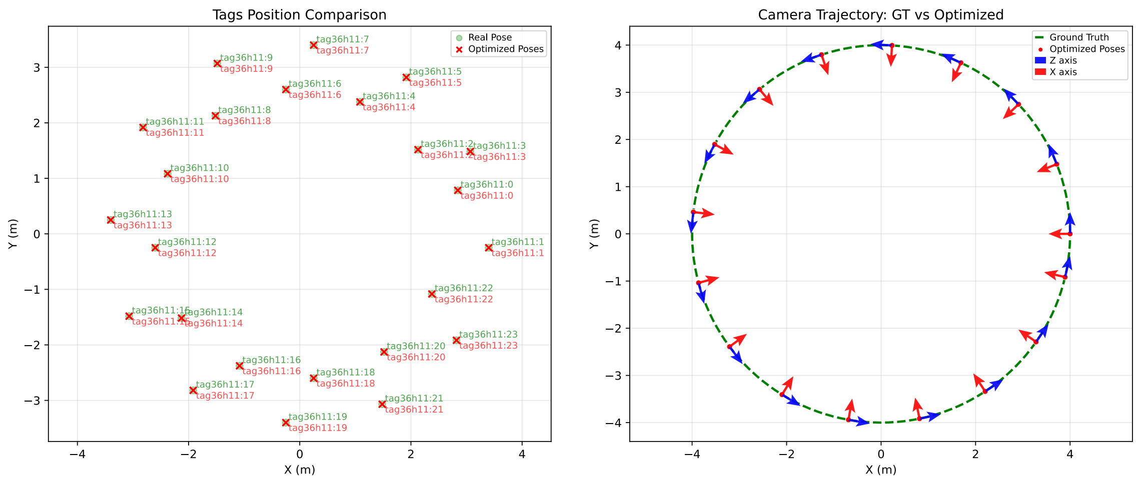


Figure 4.1. Optimized tag map and reconstructed camera trajectory in the noise-free scenario.

To evaluate the behavior under perturbation, zero-mean Gaussian noise with standard deviation of 10 cm was injected into the translational components of the synthetic camera-to-tag observations. The resulting optimization errors are reported in Table 4.2.

Metric	Mean Error	Minimum	Maximum
Camera Trajectory	0.0584 m/0.97°	0.0126 m/0.11°	0.1485 m/2.67°
Tag Poses	0.0364 m/0.73°	0.0156 m/0.73°	0.0583 m/0.73°

Table 4.2. Error statistics under 10 cm Gaussian noise.

The identical angular values reported for Tag Poses in Table 4.2 are due to rounding at the displayed precision for this synthetic configuration.

Figure 4.2 presents the corresponding visual comparison. Although the measurements were perturbed, the optimized tag map preserves the geometric structure of the environment and the reconstructed trajectory remains consistent with the underlying ground-truth configuration within the error bounds reported in Table 4.2.

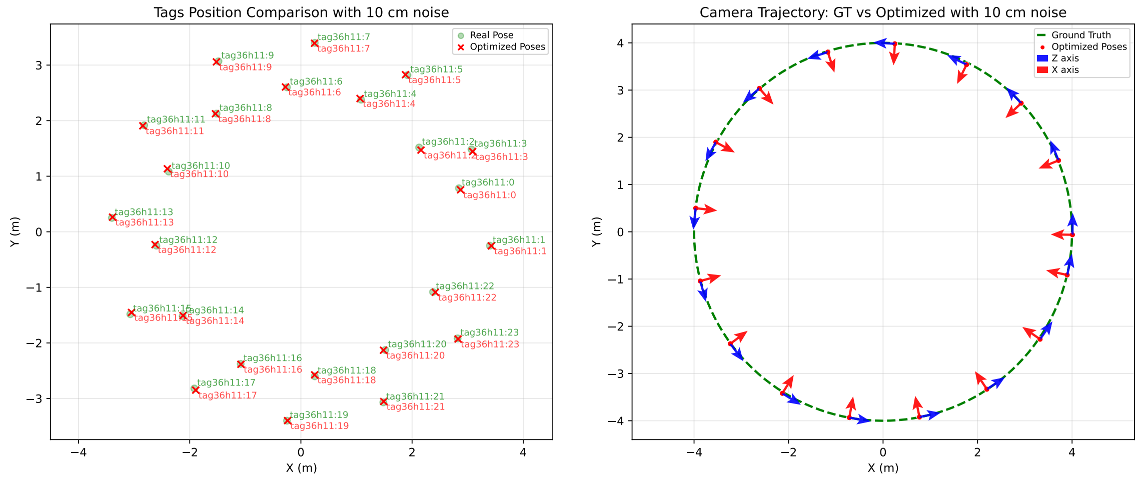


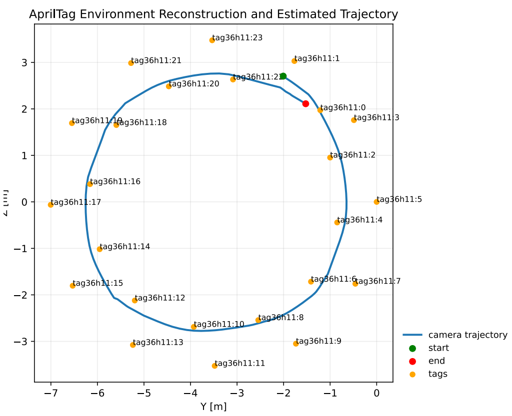
Figure 4.2. Optimized tag map and reconstructed trajectory under 10 cm Gaussian noise.

4.2 Recording Phase Validation in Simulation

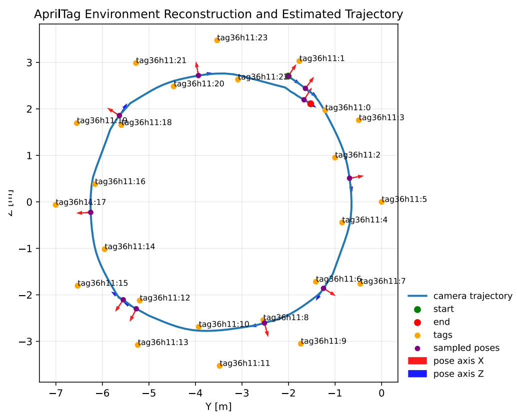
The recording pipeline was evaluated in the Ignition Gazebo environment to analyze the behavior of the pose estimation and nonlinear optimization stages under simulated sensing conditions with known ground truth. In this configuration, AprilTags were placed at predefined positions in the virtual world, and the robot executed a circular traversal while detections were logged.

Since the simulator provides ground-truth camera and tag poses, the reconstructed quantities obtained from the optimization procedure described in Section 3.1 can be directly compared against the reference configuration. This setup enables evaluation of the geometric consistency of the map estimation stage independently of the relocalization process.

The reconstructed trajectory in Fig. 4.3(a) follows the circular path imposed in the simulated experiment. This behavior reflects the accumulation of pairwise constraints defined in Eq. (3.2) and their reconciliation through the nonlinear least-squares formulation in Eq. (3.4).



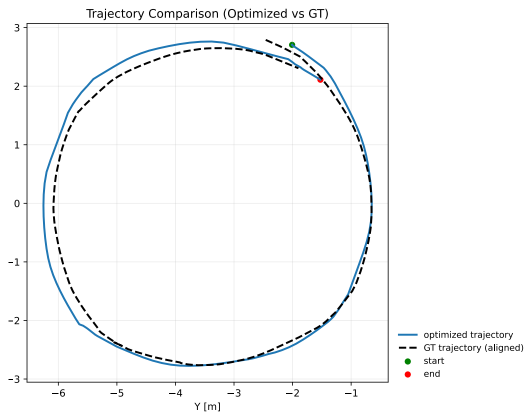
(a) Reconstructed trajectory



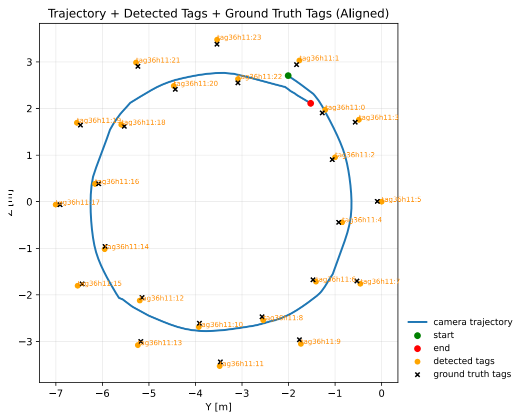
(b) Reconstructed trajectory with local frames

Figure 4.3. Reconstructed camera trajectory obtained after map optimization.

The local frames shown in Fig. 4.3(b) exhibit smooth orientation variation along the path, consistent with the joint minimization of translational and rotational residuals described in Eqs. (3.5)–(3.6).



(a) Optimized vs Ground Truth trajectory



(b) Optimized trajectory and tags vs Ground Truth

Figure 4.4. Comparison between optimized reconstruction and ground-truth configuration.

A direct comparison with the simulator reference is provided in Fig. 4.4(a). The trajectory error was computed as the 3D distance between each optimized pose and its nearest ground-truth counterpart in Euclidean space. Over the complete traversal, the mean error was 0.0510 m, with RMSE of 0.0566 m and a 95th percentile of 0.0902 m.

The tag reconstruction can be assessed in Fig. 4.4(b), where optimized tag poses are shown together with their ground-truth positions. For the $N = 24$ detected tags, the translational error relative to ground truth yielded a mean of 0.0873 m, a median of 0.0908 m, and an RMSE of 0.0879 m, consistent with the global solution obtained from the robust least-squares formulation in Eq. (3.7).

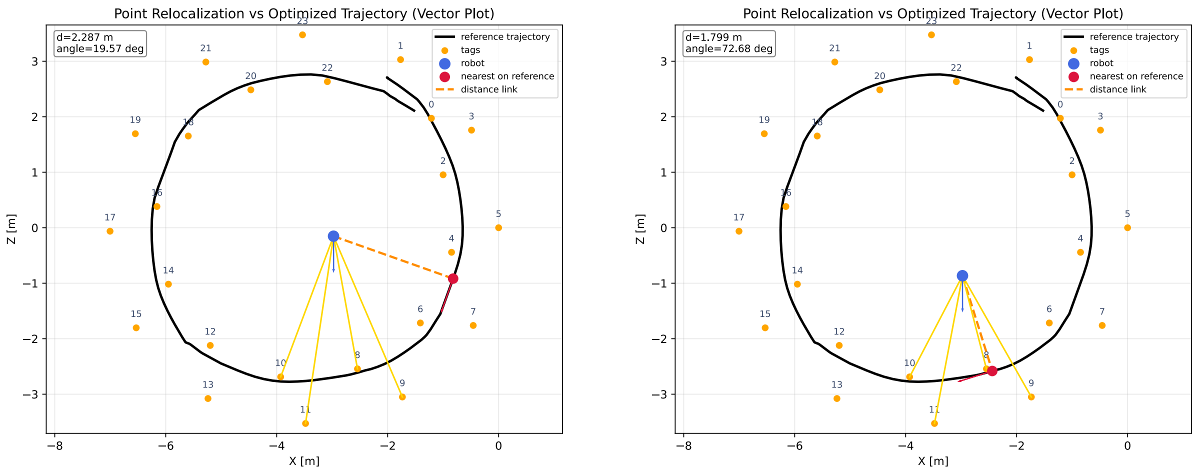
Taken together, the reported trajectory and tag errors indicate that the reconstruction obtained in simulation remains geometrically coherent with the imposed environment while following the same estimation process defined in the recording phase. Although the simulator provides ground-truth poses, the pipeline operates exclusively on image-based detections and solves the PnP problem using the simulated camera intrinsics, followed by nonlinear least-squares optimization of relative constraints. Consequently, the solution minimizes residuals between observed tag pairs rather than absolute deviation from ground truth. The resulting mean trajectory error of approximately 5 cm and mean tag error of approximately 8 cm reflect the behavior of the relative formulation under simulated sensing. At the scale of the circular trajectory and tag distribution considered in this experiment, these deviations remain small relative to the environment dimensions and do not modify the recovered global structure.

4.3 Relocalization Validation in Simulation

The relocalization stage was evaluated in the Ignition Gazebo environment using the reference trajectory generated during the recording phase. Since the simulator provides ground-truth poses, the computed geometric quantities can be directly compared against analytical values.

To verify the numerical consistency of the distance computation D, we selected synthetic test points between tag pairs in the YZ plane and compared algorithm outputs with simulator ground truth.

Across the eight tested points, the observed absolute distance error remained at numerical-precision scale (order of 10^{-7} m), which confirms that the implemented distance formulation is consistent with the simulator geometry.



(a) Point 1

(b) Point 8

Figure 4.5. Representative synthetic points used to validate the computation of D in the YZ plane.

In these synthetic examples, the displayed heading ϕ is consistent with the expected angle because the test orientations were intentionally defined during point generation. This validation therefore confirms both the distance metric and the angle convention before full relocalization experiments.

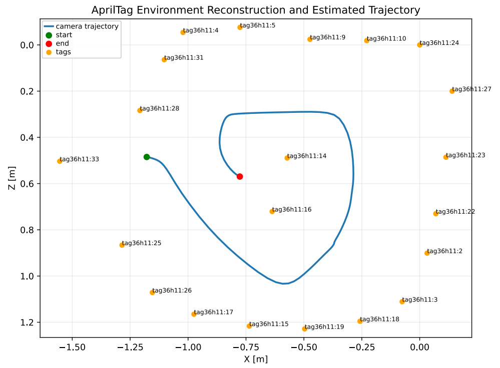
4.4 Recording Phase Validation on the Real Platform

Notwithstanding, as the algorithms were validated with synthetic data and simulation, the next step was to use them in a real scenario. In Figure 4.6, the real robot was controlled with a remote controller in an environment full of tags, to avoid points where the robot would not be able to detect markers. It is worth emphasizing that ground markers were placed for the beginning, a triangle, and the end, a black X, of the executed trajectory.

For the validation of the first recording stage, the robot was driven while the detection node was responsible for extracting all detections over time and storing them in a rosbag. With the rosbag as database, it was possible to extract the YAML file and the images with detections and, thus, estimate the tags map and the trajectory performed by the robot camera.

Notwithstanding, the files with the detections of the tags must be sent to a computer with computational power greater than that of the LIMO robot, since, to perform the optimization of all tags and also of all trajectory positions per frame, we increase the number of parameters to a level that would be costly for the LIMO robot to compute.

(a) Real-world experimental setup for the recording phase.



(b) Reconstructed map and estimated camera trajectory from the recorded rosbag.

Figure 4.6. Real recording setup and corresponding reconstructed trajectory result.

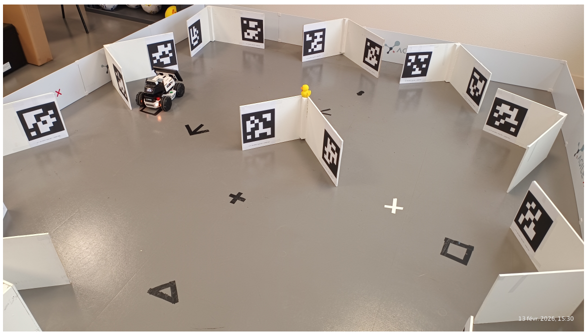
In Figure 4.6, one can observe the shape of tag map reconstructed in a way very close to the real one when compared with the image captured at the moment of the experiment and with the plotted graph. In the graph, the tags are in coherent relative positions, both for those that are on the same board and for those that are on different boards. The optimized trajectory performed by the robot also became similar to the real trajectory, in exclamation format, as can be verified in the experiment video.

For the real scenario, we do not have full ground truth to verify whether the tags are exactly at their physical placements. To evaluate map consistency, tape-measure measurements were performed from tag 16 to the other tags. The complete comparison between estimated and measured distances is provided in Table A.1.

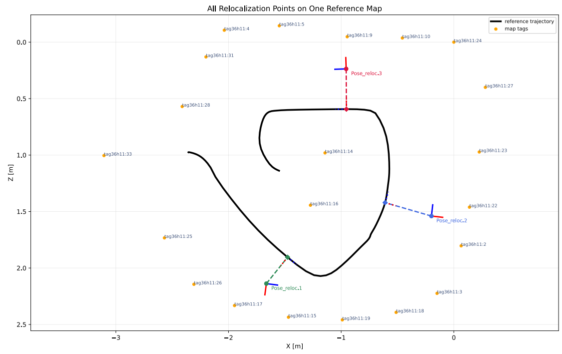
Analyzing Table A.1, a maximum error of 0.099 m is observed between tags 16 and 15, a minimum error of 0.001 m between tags 16 and 14, and a mean error of 0.039 m. These values indicate centimeter-level consistency overall. The minimum error for tags 16 and 14 is coherent with their frequent joint visibility, while the largest error can be related to observation angle, fewer supporting frames, and manual measurement uncertainty.

4.5 Relocalization Validation on the Real Platform

For the relocalization stage on the real robot, we do not have global pose ground truth. Therefore, we defined a local metric reference using tape-measure measurements between each robot test pose (outside the recorded trajectory) and the nearest point on the recorded path. To perform these measurements, markers were placed on the floor. As shown in Figure 4.7, arrow markers and the white X marker represent poses on the recorded trajectory, while the square marker and the figure-eight marker represent poses outside the trajectory.



(a) Real-world setup for relocalization evaluation with marked test poses.



(b) Measured and estimated relocalization distances for three real test cases.

Figure 4.7. Real relocalization validation setup and corresponding distance comparison.

Thus, for this trial, three points were marked on the trajectory and three points in which the robot was placed during the relocalization phase. The distances between points measured in the real environment and the estimated distances for the same poses with optimized data are in Table 4.3.

Relocalization Case	Estimated D [m]	Measured D [m]	Error [m]	ϕ [deg]
1	0.2998	0.3100	0.0102	-31.38
2	0.4284	0.4200	0.0084	-10.34
3	0.3567	0.3600	0.0033	-1.63

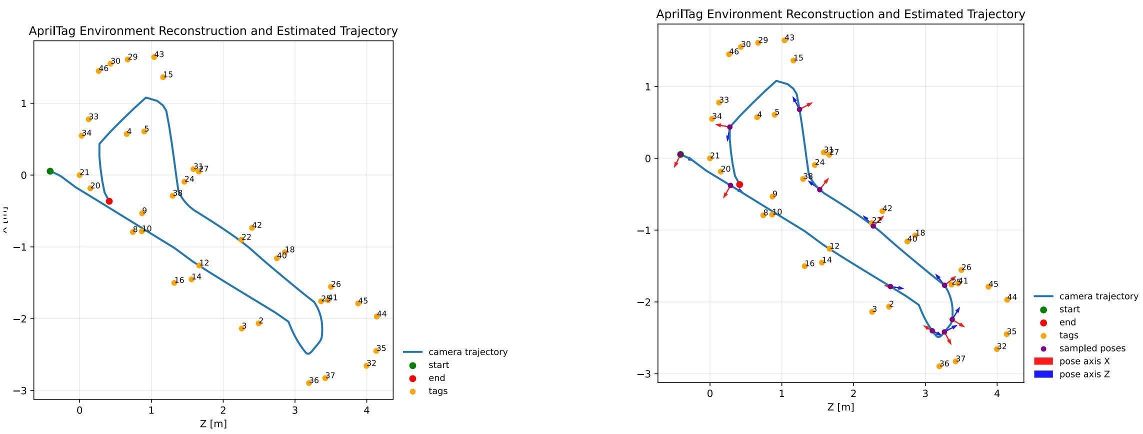
Table 4.3. Summary table with estimated versus measured distance to the recorded trajectory and the estimated heading offset ϕ.

Analyzing Table 4.3, the distance error in relocalization with respect to the recorded trajectory ranges from 0.003 m to 0.010 m between estimated and measured values. This centimeter-level consistency is coherent with multi-tag observations acquired while the robot is stationary at each test pose. The reported ϕ values are consistent estimated heading offsets for these cases; however, without external angular ground truth, they should not be interpreted as absolute orientation error metrics.

4.6 Second Real Recording Experiment

To complement the first real recording result, an additional real dataset from the second scenario was processed using the same offline recording pipeline. This trajectory was acquired under different practical conditions and used to assess whether the same estimation procedure produced consistent outputs when the traversal shape and marker observations changed.

As shown in Fig. 4.8, the reconstructed path follows the executed route, and the orientation frames evolve smoothly along the trajectory samples. This complementary case indicates that the same pairwise-constraint accumulation and nonlinear least-squares optimization described in Section 3.1 produce consistent map-and-trajectory estimates under more than one real recording condition, without modifying the estimation formulation.



(a) Second scenario reconstructed trajectory

(b) Second scenario trajectory with local frames

Figure 4.8. Additional real recording result processed with the same offline pipeline.

5 Conclusion

This work implements a ROS 2 pipeline for trajectory recording and trajectory-relative relocalization using AprilTags on the AgileX LIMO platform. The workflow is organized in two phases. The first phase processes a recorded rosbag2 traversal offline to estimate an optimized tag map and an associated reference trajectory. The second phase performs relocalization in offline analysis and in online execution by comparing new pose estimates to the stored reference and computing two control-oriented quantities, namely the distance to the nearest trajectory point and the heading offset with respect to the local trajectory direction. The implementation is limited to geometric reference construction and metric computation, while trajectory tracking control is treated as external to this work.

The pipeline was evaluated progressively using synthetic data, simulation, and real experiments. In controlled synthetic tests, the map optimization stage produced solutions consistent with the expected behavior under ideal observations and under injected noise. In simulation, comparison against ground truth yielded mean errors of 0.051 m for the reconstructed trajectory and 0.087 m for the reconstructed tag positions, with bounded deviations at the scale of the simulated environment.

In real recording experiments, where full global ground truth is not available, map consistency was assessed by comparing tape-measure distances from tag 16 to 20 other tags against the corresponding distances computed from the estimated map. The resulting absolute error ranged from 0.001 m to 0.099 m, with a mean of 0.039 m. In real relocalization tests performed off the reference trajectory, the distance output was compared against manual measurements, producing discrepancies between 0.0033 m and 0.0102 m. The heading offsets reported by the system in these trials ($\phi = -31.38°$, $-10.34°$, and $-1.63°$) are internal relocalization outputs and are not treated as absolute angular errors due to the absence of an external orientation reference.

These results support the internal consistency of the recording and relocalization stages under the available references, while also delimiting the scope of the evaluation. Future work should extend the set of real-world trials, include an external orientation reference for validating ϕ, and integrate a trajectory tracking controller that consumes the validated relocalization outputs.

6 Difficulties and Use of Artificial Intelligence

During the development of the project, several practical difficulties were encountered. A primary limitation was the incompatibility between the ROS version installed on the Polytech robots and the ROS distribution supported by the AprilTag package used in this work. This required adjustments in the software environment and constrained direct reuse of certain components across platforms.

Another limitation was the restricted time available for extensive experimental validation. Although additional trials in different environments were planned, including experiments at PAVIN, these tests could not be conducted due to unfavorable weather conditions and the proximity of the report submission deadline. Consequently, the experimental campaign was limited to the scenarios described in this document.

Artificial intelligence tools were used during the project in a supportive capacity. They assisted in identifying implementation issues in the code and in suggesting alternative design approaches that were subsequently evaluated and refined by the authors. AI tools were also used for language revision and translation, as the initial drafts of several sections were written in the author's native language.

References

[1] I. Ullah, D. Adhikari, H. Khan, M. S. Anwar, S. Ahmad, and X. Bai, "Mobile robot localization: Current challenges and future prospective," *Computer Science Review*, vol. 53, p. 100 651, 2024, ISSN: 1574-0137. DOI: `https://doi.org/10.1016/j.cosrev.2024.100651`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S1574013724000352`.

[2] S. A. S. Mohamed, M.-H. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen, and J. Plosila, "A survey on odometry for autonomous navigation systems," *IEEE Access*, vol. 7, pp. 97 466–97 486, 2019. DOI: `10.1109/ACCESS.2019.2929133`.

[3] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, Available at: `https://april.eecs.umich.edu/pdfs/olson2011icra.pdf`, 2011, pp. 3400–3407. [Online]. Available: `https://april.eecs.umich.edu/pdfs/olson2011icra.pdf`.

[4] M. Krogius, A. Haggenmiller, and E. Olson, "Flexible layouts for fiducial tags," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1898–1903. DOI: `10.1109/IROS40897.2019.8967787`.

[5] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Second. Cambridge University Press, ISBN: 0521540518, 2004, Available at: `http://www.robots.ox.ac.uk/~vgg/hzbook/hzbook2/HZbook-2nd-Ed.pdf`.

[6] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.

[7] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment — a modern synthesis," in *Vision Algorithms: Theory and Practice*, B. Triggs, A. Zisserman, and R. Szeliski, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 298–372, ISBN: 978-3-540-44480-0.

[8] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010. DOI: `10.1109/MITS.2010.939925`.

[9] P. J. Huber, "Robust Estimation of a Location Parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964. DOI: `10.1214/aoms/1177703732`. [Online]. Available: `https://doi.org/10.1214/aoms/1177703732`.

[10] P. H. Schoenemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966. DOI: `10.1007/BF02289451`.

[11] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981, ISSN: 0001-0782. DOI: `10.1145/358669.358692`. [Online]. Available: `https://doi.org/10.1145/358669.358692`.

Appendices

A Tag Distance Validation Table

Source tag	Target tag	Estimated [m]	Measured [m]	Absolute error [m]
16	2	1.385	1.380	0.005
16	3	1.368	1.360	0.008
16	4	1.728	1.820	0.092
16	5	1.613	1.690	0.077
16	9	1.527	1.450	0.077
16	10	1.691	1.680	0.011
16	14	0.481	0.480	0.001
16	15	1.011	1.110	0.099
16	17	1.115	1.090	0.025
16	18	1.216	1.220	0.004
16	19	1.054	1.040	0.014
16	22	1.412	1.410	0.002
16	23	1.571	1.560	0.011
16	24	1.925	1.900	0.025
16	25	1.328	1.300	0.028
16	26	1.249	1.220	0.029
16	27	1.871	1.800	0.071
16	28	1.436	1.470	0.034
16	31	1.608	1.660	0.052
16	33	1.886	1.870	0.016
Mean	-	-	-	**0.039**
Max	-	-	-	**0.099**
Min	-	-	-	**0.001**

Table A.1. Comparison between estimated and measured distances from tag 16.

B Project Website QR Code

Figure B.1. QR code linking to the project website and complete media/results page.