UNIVERSITÉ CLERMONT AUVERGNE

École Universitaire de Physique et d'Ingénierie

AprilTag-based Trajectory Tracking for the LIMOS Robot

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

Defense Date:
March 02, 2026

Aubière, Clermont-Ferrand

Title

Subtitle

Author

Abstract

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Contents

List of Figures

Nomenclature

Camera Model

(c_x, c_y) Coordinates of the principal point of the image

(u, v) Pixel coordinates in the image

(x_d, y_d) Distorted image coordinates

(x_n, y_n) Normalized coordinates in the image plane

$\boldsymbol{\epsilon}_i$ Measurement noise associated with corner observation

λ Scale factor (depth Z_c)

$\mathbf{p}_c$ 3D point expressed in the camera reference frame $[X_c\ Y_c\ Z_c]^\top$

$\mathbf{u}_i$ Observed pixel coordinates (measurement)

$\pi(\cdot)$ Perspective projection function $\pi([x, y, z]^\top) = [x/z, y/z]^\top$

Σ_u Covariance matrix of measurement noise

f_x, f_y Focal lengths expressed in pixels

K Intrinsic camera matrix (internal parameters)

k_i, p_i Radial and tangential distortion coefficients

Geometry and Transformations

R Rotation matrix $SO(3)$

$SE(3)$ Special Euclidean group (rigid transformations in 3D)

$SO(3)$ Special Orthogonal group (rotations in 3D)

t Translation vector $\mathbb{R}^3$

${}^A T_B$ Homogeneous transformation matrix from reference frame B to A

AprilTag and Algorithms

$\mathbb{I}\{\cdot\}$ Indicator function (1 if true, 0 otherwise)

$\rho^r(\cdot)$ Bitwise rotation function by 90° increments

$d_H(a, b)$ Hamming distance between two binary codes

H Planar homography matrix

1 Introduction

Being able to follow a path once and repeat it optimally is a practical skill for mobile robots, especially in indoor environments where GPS is not available and tasks are repetitive by nature. Thus, this is the central idea behind Teach and Replay[1], the robot first performs a guided route while it registers the markers looking for a positioning reference of each tag and then uses this record of positions to reproduce the same route optimally.

In this practical exercise, this concept was implemented on a LIMO mobile robot using its integrated RGB cam and AprilTag [2] fiducial markers positioned along a closed circuit with a start and end. During the teach phase, the robot completes a first turn while detecting the markers and saves the information from the positions of the robot in relation to the tags to represent the executed movement. After pressing the Y key of the Xbox controller or another controller's triangle, the system switches to the replay phase, where the robot tries to follow the learned trajectory. The goal is simple: complete the circuit, reducing the path deviation and maintaining a safe distance from the tags, but a restriction is imposed on the user of at least always seeing two tags in the cam.

In order to develop the solution in a controlled way and evaluate its behavior in real conditions, a simulation workflow was made for reality. Therefore, it started in the Gazebo Ignition, where the perception and control components have been tested and validated. We then transfer the same workflow to the real LIMO robot, where external noise such as lighting variation, motion blur, and wheel slippage naturally increase the challenge of detection quality for tracking performance. This report therefore documents not only the final system but also the changes that occurred when moving from simulation to a real platform.

The rest of the report presents the requirements of the problem, justifying that it will be useful a tracking system trajectory, and also details the methodology proposed for the method Teach and Replay, both for the part of the validation in simulation and for the implementation in the real case, defines the evaluation metrics and discusses the results obtained with a direct comparison between simulated and real executions.

However, this work was that it will serve to help students from related areas of programming and robotics at Polytechnique de Clermont-Ferrand to use the system in a practical work, so they can understand how the simple camera perception part, using markers, can be applied directly to mobile robotics, understand the complexity of the transformations of positions used and how, with a simple camera and markers, it is possible to trace a trajectory and, with optimization, arrive at the best possible trajectory.

1.1 Objectives

1.1.1 General objective

To develop and validate a vision-based *Teach and Replay* navigation system for a mobile robot, using an RGB camera and fiducial markers, capable of learning and reproducing a trajectory.

1.1.2 Specific objectives

→ Implement a visual perception pipeline based on AprilTag detection for estimating the relative pose between the robot and multiple observed markers.

- Design and implement the *teach* phase in a closed circuit, recording the guided trajectory of the robot under the operational constraint of simultaneous visibility of at least two markers.
- Develop the *replay* phase, allowing the reproduction of the learned trajectory with minimization of path deviation.
- Validate and debug the perception and control modules in a simulation environment, under controlled conditions, before implementation on real hardware.
- Transfer the same workflow to the LIMO mobile robot and analyze the impact of real-world disturbances.
- Make available a didactic and reusable system on GitHub.

2 Theoretical foundation

2.1 Pose composition in $SE(3)$

In mobile robotics and computer vision, the geometric relationship between two rigid reference frames is described by a transformation of the special Euclidean group $SE(3)$, which combines rotation and translation into a single homogeneous matrix [3], [4]. Thus, the pose of a reference frame B with respect to a reference frame A is represented by:

$$ {}^{A}T_{B} = \begin{bmatrix} {}^{A}R_{B} & {}^{A}t_{B} \\ 0 & 1 \end{bmatrix}, \qquad {}^{A}R_{B} \in SO(3),\ {}^{A}t_{B} \in \mathbb{R}^{3}, \tag{2.1} $$

In which ${}^{A}R_{B}$ encodes the orientation of B expressed in A, and ${}^{A}t_{B}$ is the position vector of the origin of B expressed in A. Equation (2.1) provides the transformation for a 3D point $\mathbf{p}_{B}$ expressed in B, obtaining its coordinate in A by:

$$ \bar{\mathbf{p}}_{A} = {}^{A}T_{B}\,\bar{\mathbf{p}}_{B}, \qquad \bar{\mathbf{p}} = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix}. \tag{2.2} $$

Thus, the composition of poses follows directly from the product of homogeneous matrices. In particular, if ${}^{A}T_{B}$ represents the pose of B with respect to A and ${}^{B}T_{C}$ the pose of C with respect to B, then the pose of C with respect to A is given as:

$$ {}^{A}T_{C} = {}^{A}T_{B}\ {}^{B}T_{C}, \tag{2.3} $$

which allows chaining relative measurements and absolute reference frames within the same geometric model [3], [4].

Therefore, three reference frames were adopted, the camera as *cam*, the marker as *tag*, and a global reference frame as *world*. Visual estimation directly provides the transformation of the marker with respect to the camera, denoted by ${}^{\text{cam}}T_{\text{tag}}$, that is, the pose of the *tag* expressed in the camera reference frame, in the form of Equation (2.1). When necessary, the pose of the camera with respect to the marker is obtained by the inverse:

$$ {}^{\text{tag}}T_{\text{cam}} = \left({}^{\text{cam}}T_{\text{tag}}\right)^{-1}. \tag{2.4} $$

In addition, transformations between camera, marker, and world are obtained by chaining according to Equation (2.3). For example, if ${}^{\text{world}}T_{\text{tag}}$ is known and ${}^{\text{cam}}T_{\text{tag}}$ is measured by the camera, its pose in the world can be written as:

$$ {}^{\text{world}}T_{\text{cam}} = {}^{\text{world}}T_{\text{tag}}\ {}^{\text{tag}}T_{\text{cam}} = {}^{\text{world}}T_{\text{tag}}\ \left({}^{\text{cam}}T_{\text{tag}}\right)^{-1}. \tag{2.5} $$

This convention will be used to combine successive observations of multiple tags and express all poses in a global reference frame, being necessary to avoid unnecessary calculations and transformations.

2.2 Camera

The geometric modeling presented in Section 2.1 allows expressing 3D points in different reference frames through transformations in $SE(3)$. In particular, a point $\mathbf{P}_w \in \mathbb{R}^3$ expressed in the global reference frame can be converted to the camera reference frame, thus we have that:

$$ \bar{\mathbf{P}}_c = {}^{\text{cam}}T_{\text{world}}\ \bar{\mathbf{P}}_w, \qquad \bar{\mathbf{P}} = \begin{bmatrix} \mathbf{P} \\ 1 \end{bmatrix}, \tag{2.6} $$

In which ${}^{\text{cam}}T_{\text{world}} \in SE(3)$ represents the pose of the *world* reference frame expressed in the camera reference frame, according to the convention of Equation (2.1). Equivalently, explicitly writing rotation and translation,

$$ \mathbf{P}_c = {}^{\text{cam}}R_{\text{world}}\ \mathbf{P}_w + {}^{\text{cam}}t_{\text{world}}, \tag{2.7} $$

the point in the camera reference frame $\mathbf{P}_c = [X_c\ Y_c\ Z_c]^\top$ is obtained, which is the point used by the *pinhole* projective model [3].

Therefore, considering a 3D point expressed in the camera reference frame $\mathbf{P}_c = [X_c\ Y_c\ Z_c]^\top$, with $Z_c > 0$, its coordinates in the normalized image plane are given by:

$$ x_n = \frac{X_c}{Z_c}, \qquad y_n = \frac{Y_c}{Z_c}. \tag{2.8} $$

With this, the conversion of these normalized coordinates to pixel coordinates is performed by the intrinsic matrix K, which incorporates the focal length and the principal point. Thus, the projection in pixels can be written as follows:

$$ \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix}, \qquad K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.9} $$

in which f_x and f_y are the focal lengths in pixels and (c_x, c_y) is the principal point.

Equivalently, combining Eqs. (2.8) and (2.9), the direct projection of a point in the camera reference frame to pixel coordinates can be expressed as follows:

$$ \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}, \qquad \lambda = Z_c, \tag{2.10} $$

Thus, this subsection is directly connected to the formulation used later in *Perspective-n-Point*.

Real lenses introduce geometric distortions that deviate from the ideal projection of the *pinhole* model. In practical applications, these distortions are often modeled by radial and tangential terms, estimated through calibration [5]. The presence of unmodeled or even poorly calibrated distortion systematically shifts image observations, potentially affecting mainly corner localization and, consequently, degrading pose estimation based on reprojection.

For uncertainty analysis, it is common to assume that corner observations in the image are affected by approximately Gaussian additive noise in pixels. Thus, for each corner i one observes:

$$\hat{\mathbf{u}}_i = \mathbf{u}_i + \boldsymbol{\epsilon}_i, \qquad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \Sigma_u), \tag{2.11}$$

in which $\mathbf{u}_i = [u_i \; v_i]^\top$ represents the ideal projection and $\hat{\mathbf{u}}_i$ the measurement in the image. This hypothesis justifies formulating pose estimation as a least-squares problem via reprojection error, since under Gaussian noise the minimization of the sum of squared errors coincides with maximum likelihood estimation [3].

In this way, it is assumed that the robot camera is calibrated, such that K and the distortion coefficients are known. Corner detections obtained from AprilTags are treated as pixel observations.

2.2.1 Pose by PnP

Once the four corners of the marker are obtained and its identifier is validated, the pose of the marker with respect to the camera can be estimated because the AprilTag has known planar geometry and defined physical dimensions. In the detector proposed by Olson, the geometry observed in the image is initially related to the marker plane through a homography; the author describes that the method computes the 3×3 homography matrix and that it is obtained by the DLT (Direct Linear Transform) algorithm [2]. However, to recover position and orientation in metric units, the homography alone is not sufficient; pose estimation requires additional information, specifically the camera intrinsic parameters and the physical size of the marker [2].

Modeling the camera using the pinhole model, the relationship between the known 3D points, which are the marker corners in its reference frame, and their 2D observations in the image can be written by Equation (2.12). In this formulation, each marker corner $\mathbf{X}_i$ is transformed to the camera reference frame by a rotation ${}^{\text{cam}}R_{\text{tag}}$ and a translation ${}^{\text{cam}}t_{\text{tag}}$, and then projected onto the image plane via the intrinsic matrix K:

$$\lambda_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = K\Big({}^{\text{cam}}R_{\text{tag}}\mathbf{X}_i + {}^{\text{cam}}t_{\text{tag}}\Big), \tag{2.12}$$

in which λ_i is a scale factor, $\mathbf{u}_i = [u_i \; v_i]^\top$ represents the corner observation in the image, and $\mathbf{X}_i$ represents the corresponding corner in the marker reference frame.

Thus, pose estimation by PnP can then be interpreted as the problem of finding the parameters ${}^{\text{cam}}R_{\text{tag}}$ and ${}^{\text{cam}}t_{\text{tag}}$ that best explain the observations, minimizing the reprojection error. Equation (2.13) formalizes this idea by defining a cost function based on the difference between the observed corners $\mathbf{u}_i$ and the reprojected corners $\hat{\mathbf{u}}_i$, obtained by projecting the points transformed by the geometric model:

$$\begin{aligned} &\min_{{}^{\text{cam}}R_{\text{tag}} \in SO(3),\, {}^{\text{cam}}t_{\text{tag}} \in \mathbb{R}^3} \; \sum_{i=1}^{4} \left\| \mathbf{u}_i - \pi\big(K\left({}^{\text{cam}}R_{\text{tag}}\,\mathbf{X}_i + {}^{\text{cam}}t_{\text{tag}}\right)\big) \right\|^2 \\ &= \min_{{}^{\text{cam}}R_{\text{tag}},\, {}^{\text{cam}}t_{\text{tag}}} \; \sum_{i=1}^{4} \left\| \mathbf{u}_i - \hat{\mathbf{u}}_i \right\|^2, \quad \hat{\mathbf{u}}_i = \pi\big(K\left({}^{\text{cam}}R_{\text{tag}}\,\mathbf{X}_i + {}^{\text{cam}}t_{\text{tag}}\right)\big)\,. \end{aligned} \tag{2.13}$$

Thus, in Eq. (2.13), the function $\pi(\cdot)$ represents the perspective projection, defined as $\pi([x\ y\ z]^\top) = [x/z\ \ y/z]^\top$. This formulation highlights that the quality of the estimated pose depends directly on the accuracy of corner detection $\mathbf{u}_i$, intrinsic calibration K, and the geometric validity of the marker. Therefore, corner quantization errors, motion blur, illumination variations, or inconsistencies in detection degrade the reprojection error and, consequently, the geometric consistency.

2.3 AprilTag

AprilTags are planar fiducial markers widely used as *landmarks* in robotics and computer vision, as they allow recovering the marker identity and estimating its relative pose with respect to the camera from a single image [2], [6]. However, unlike 2D codes designed for high data capacity, the design of AprilTags prioritizes perceptual robustness and a low false positive rate. Therefore, this robustness results from the combination of a high-contrast outer border, suitable for geometric extraction under perspective, and an internal *payload* constructed as a finite family of binary codes with Hamming distance validation [2], [6].

Thus, Figure 2.1 illustrates this functional separation: the border facilitates quadrilateral detection and accurate estimation of the four corners in the image, while the internal region encodes the identifier in a redundant and verifiable manner. In general terms, detection can be understood as a two-stage process, locating geometrically consistent candidates and rectifying the internal region to decode the *payload* [2]. After ID validation, the observed corners provide 2D/3D correspondences with the known square geometry, which allows estimating the relative pose of the marker via the camera projective model.

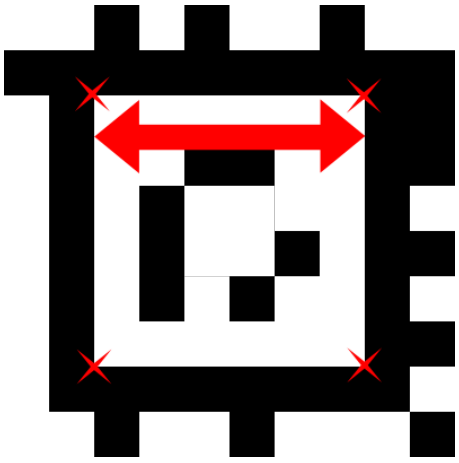


Figure 2.1: Structure of an AprilTag.

In this way, each detection provides the marker identifier and a rigid transformation associated with its relative pose, usually expressed as ${}^{\text{cam}}T_{\text{tag}} \in SE(3)$. This transformation is obtained from the correspondences between the corners in the image and the known corners in the marker reference frame, and can be refined by minimizing the reprojection error. Likewise, Figure 2.2 shows a typical example of detector visualization, in which the estimated quadrilateral, the decoded identifier, and the marker reference frame axes are overlaid on the image.

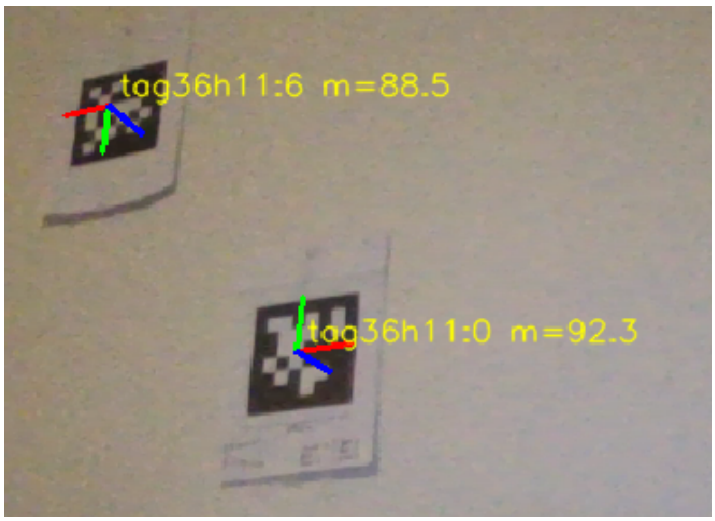


Figure 2.2: Example of output from an AprilTag detector.

2.3.1 Hamming distance

At this stage, AprilTag decoding is deliberately conservative, as it aims to reject ambiguous patterns and reduce false positives. Therefore, the central principle consists of comparing the observed *codeword* with a dictionary of valid codes and accepting a detection only when there is a sufficiently close match.

In the original work, the decision rule is described by a Hamming threshold, in which "If the best match has a Hamming distance less than the user-specified threshold, a detection is reported" [2]. This mechanism makes explicit the trade-off between *recall* and false positive rate: more permissive thresholds increase the chance of accepting degraded readings, but also raise the risk of incorrect detections.

In this case, given two bit strings $a, b \in \{0,1\}^n$, the Hamming distance is defined as

$$d_H(a,b) = \sum_{k=1}^{n} \mathbb{I}\{a_k \neq b_k\}, \tag{2.14}$$

in which $\mathbb{I}\{\cdot\}$ is the indicator function. In AprilTags, however, the orientation of the marker in the image is unknown *a priori*, which requires validation to consider all possible discrete rotations of the pattern. For this, the minimum Hamming distance under rotations is defined as

$$d_H^{\text{rot}}(a,b) = \min_{r\in\{0,1,2,3\}} d_H\big(a, \rho^r(b)\big), \tag{2.15}$$

in which $\rho^r(\cdot)$ represents the rotation of the *codeword* in multiples of 90°.

In the case of AprilTag 2, the authors highlight that payload generation preserves a minimum separation between valid codes by explicitly considering all possible rotations, stating that the system is built by "guaranteeing a minimum Hamming distance between tags under all possible rotations" [6]. In this way, the threshold applied to Equation (2.15) acts as an explicit reliability criterion, reducing the probability of false positives even under noise, partial occlusions, or image degradation. Therefore, a conservative Hamming distance threshold was adopted, prioritizing a low false positive rate at the expense of more permissive readings.

Thus, AprilTag 3 extends the system by allowing the generation of families with flexible *layouts*, while preserving principles associated with low false positive rates. Krogius *et al.* highlight as a central contribution "a flexible layout system whereby users can generate sets of tags with the data bits arranged in a specified shape" [7]. Therefore, to maintain reliability while expanding the pattern

space, the authors introduce "a complexity metric applicable to diverse tag layouts which we use to generate tags with low false positive rates" [7]. Regardless of the chosen family, the projection and pose estimation equations remain valid, as they depend on the planar geometry of the marker and the camera model, and not on the specific binary pattern employed.

3 Methodology

3.1 Pipeline validation

Before integration into simulation in Gazebo Ignition, a preliminary validation stage was carried out with the objective of verifying the geometric consistency of the proposed pipeline. In this case, this validation had an exclusively qualitative character and was conducted in the RViz environment, which allowed inspection of the reference frames, axes, and transformations involved in the problem.

At this stage, a simple script was developed responsible for publishing the main geometric elements of the system in RViz, the camera and the tags, without involving robot dynamics or control. The objective was to ensure that the conventions adopted for pose representation and transformation composition in $SE(3)$ were correct before introducing additional simulation effects[8].

In particular, the script allowed simultaneous visualization of the nominal camera trajectory along a closed circular path, defined directly in the global reference frame, as well as the poses of the fiducial markers also expressed in the anchor reference frame, which was defined as 36h11:0. In addition, the *frames* associated with the camera and the tags were represented by RViz TFs.

Therefore, this visualization made it possible to confirm the correct orientation of the camera and fiducial marker axes, as well as the consistency of the rigid transformations in $SE(3)$ used to express relative and absolute poses.

Thus, immediately after the basic geometric validation, a second stage of qualitative verification was carried out, in which controlled synthetic noise was introduced into the camera and marker poses. The objective of this stage was to visually observe the behavior of the pipeline under small perturbations, simulating typical inaccuracies of visual estimation of AprilTags.

Under this condition, RViz allowed visualization of whether pose compositions remained geometrically plausible over time, that is, whether small perturbations resulted in smooth and consistent frame displacements, without causing discontinuities, axis inversions, or degenerate behaviors.

Therefore, with the consistency of the geometric model verified in these preliminary stages, the same set of transformations, reference frame conventions, and assumptions began to be used in the simulation environment in Gazebo Ignition.

3.2 Teach Phase

At this stage, the term *Teach Phase* refers to the process of systematic acquisition of geometric observations of the environment while the robot traverses a known circuit containing multiple fiducial markers. Teaching the robot, in this context, means providing sufficient information to geometrically describe the environment through the transformations associated with the markers observed along the path, without yet performing any form of global optimization or error correction.

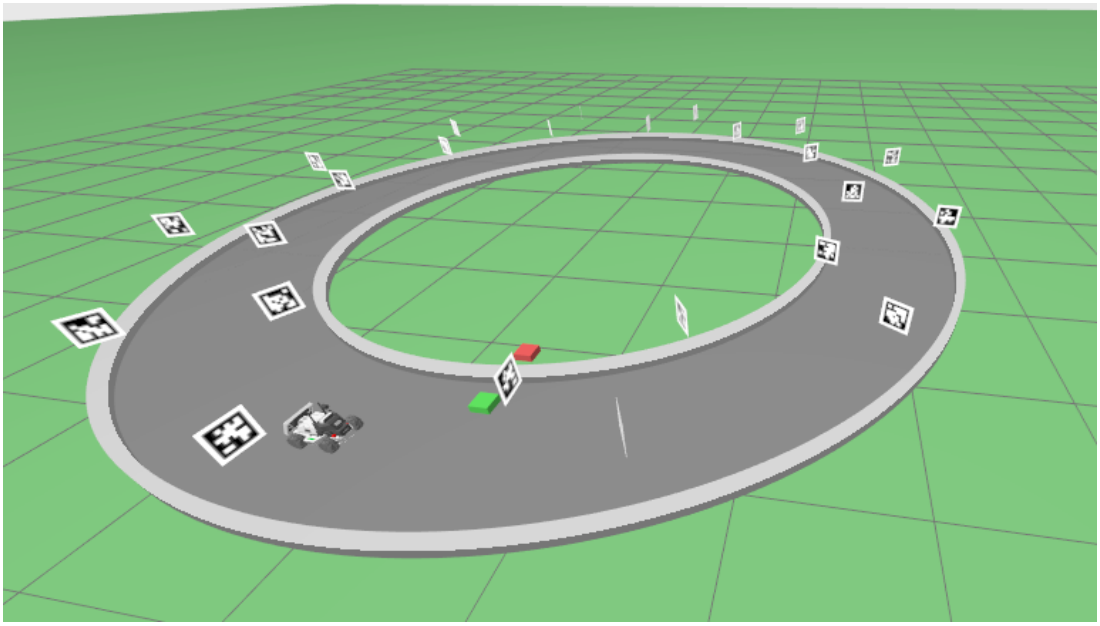


Figure 3.1: Circuit in Gazebo Ignition and distribution of AprilTags.

Thus, the simulation environment contains a closed circuit with 24 AprilTags distributed along the trajectory, as shown in Figure 3.1. Among these, only one is treated as the absolute reference of the system. The pose of the *tag zero* is known *a priori* and acts as the map anchor, defining the global reference frame. The physical size of the markers, the camera intrinsic parameters, and its calibration are assumed to be known from the beginning of the experiment. In this way, the global map is initialized exclusively from the anchor, with no prior knowledge of the position of the remaining tags.

In this case, at the beginning of the Teach Phase, the system does not know the pose of the other 23 tags, the trajectory to be traversed, nor the global pose of the camera over time. Moreover, these elements are progressively inferred from the visual observations recorded during motion. In this manner, at each instant, the robot observes a subset of the markers present in the environment, and it is imposed as an operational criterion that, whenever possible, at least two tags are simultaneously within the camera field of view. This condition favors geometric consistency between successive observations and allows establishing spatial relationships between different markers.

Furthermore, during the Teach Phase, the robot is manually guided by the operator along the circuit. The motion is conducted with approximately constant velocity and without abrupt accelerations, in order to preserve the quality of visual observations and reduce degradations associated with motion blur. Thus, the trajectory is closed, starting and ending in the region of the anchor tag, so that the global reference is observed both at the beginning and at the end of execution.

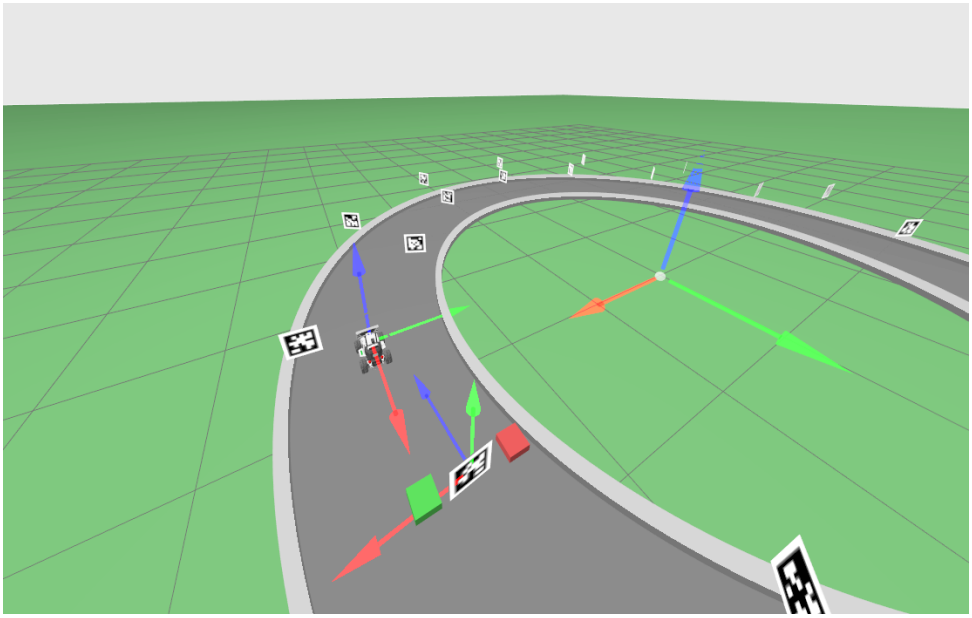


Figure 3.2: Reference frames used: *world*, *Limo*, and *tag*.

Thus, as the robot moves, each marker detection provides the rigid transformation between the camera reference frame and the reference frame of the observed tag, that is, ${}^{\text{cam}}T_{\text{tag}}(t)$. When the anchor tag is observed, the global pose of the camera can be estimated by composition in $SE(3)$:

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

In this way, for the remaining tags, the recorded observations provide relative relationships over time, which will later be used for offline optimization of their trajectory.

Figure 3.3 summarizes the flow executed in the Teach Phase, highlighting detection acquisition, pose estimation, and temporal logging in *rosbag*.

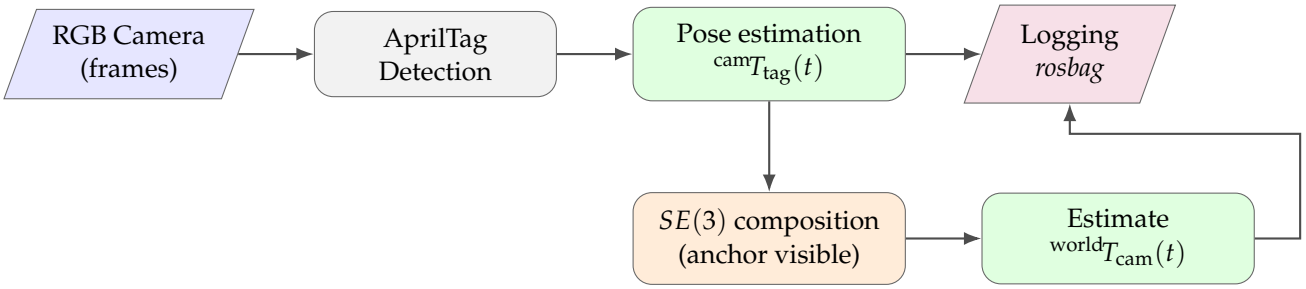


Figure 3.3: Processing flow in the Teach Phase.

It is worth noting that all observations acquired during the Teach Phase are stored in log files in the *rosbag* format. In this way, these records include tag detections, the associated geometric transformations, and the temporal information necessary to reconstruct the sequence of events; in summary, the teach phase collects all camera and tag TFs, as can be seen in the pseudocode in Algorithm 1, to be later optimized and used in the replay phase.

Algorithm 1: Teach Phase: acquisition and logging of geometric observations

Input: Known anchor pose ${}^{\text{world}}T_{\text{tag}_0}$; intrinsics K; physical size of the tags
Output: Temporal log: detections, ${}^{\text{cam}}T_{\text{tag}}(t)$, and (when possible) ${}^{\text{world}}T_{\text{cam}}(t)$

Initialize *rosbag* and storage structures
Define global reference frame *world* from the anchor (tag 0)
while *Teach Phase active (robot manually guided)* **do**
  Receive camera frame at time t
  Detect set of visible tags $\mathcal{S}(t)$ (IDs and corners)
  if $|\mathcal{S}(t)| < 2$ **then**
    Mark state as "low observability" (without aborting)
  foreach *tag* $j \in \mathcal{S}(t)$ **do**
    Estimate relative pose ${}^{\text{cam}}T_{\text{tag}_j}(t)$ from observed corners and known geometry
    Log $\left(t,\ j,\ {}^{\text{cam}}T_{\text{tag}_j}(t)\right)$ to the *rosbag*
  if *anchor tag* $0 \in \mathcal{S}(t)$ **then**
    Compute ${}^{\text{world}}T_{\text{cam}}(t)$ using composition in $SE(3)$:
      ${}^{\text{world}}T_{\text{cam}}(t) \leftarrow {}^{\text{world}}T_{\text{tag}_0} \left({}^{\text{cam}}T_{\text{tag}_0}(t)\right)^{-1}$
    Log $\left(t,\ {}^{\text{world}}T_{\text{cam}}(t)\right)$ to the *rosbag*
  Update counters/statistics (e.g., number of frames, number of detections)
Finalize and save the *rosbag*

4 Results

4.1 Results in RViz

This subsection presents the results of the *Teach Phase* in a controlled scenario, visualized in RViz, before the application of any optimization stage. The objective here is to verify the geometric consistency of the pipeline: the estimated trajectory must be compatible with the distribution of the markers and, since it is a closed circuit, the shape of the path must be preserved.

Thus, Figure 4.1 shows the reference trajectory (*ground-truth*) in the XY plane, used as a basis for comparison. Next, Figure 4.2a presents the estimated trajectory obtained from successive compositions with the anchor and the observed transformations. In this case, it is observed that the estimate preserves the general shape of the circuit and maintains spatial coherence with the arrangement of the tags, indicating that the reference frame conventions and the composition in $SE(3)$ are correct. Small local differences are expected, since each observation contributes directly to the chaining process, and local errors propagate along the sequence.

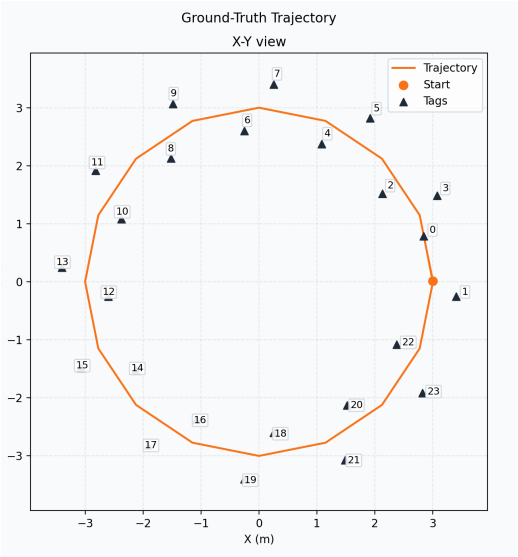
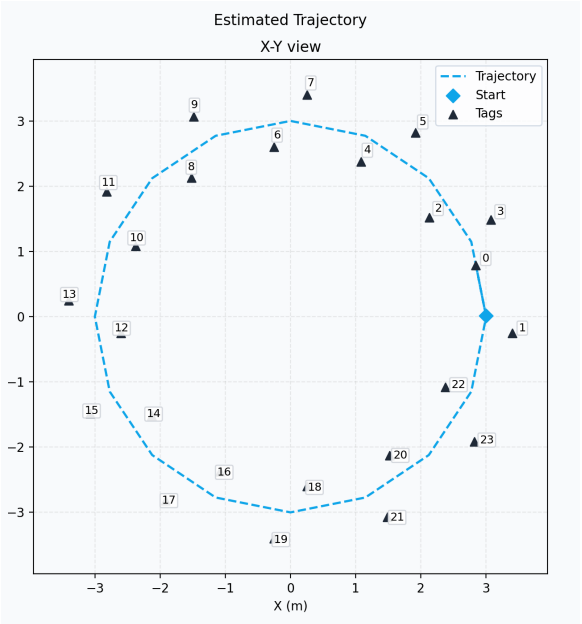
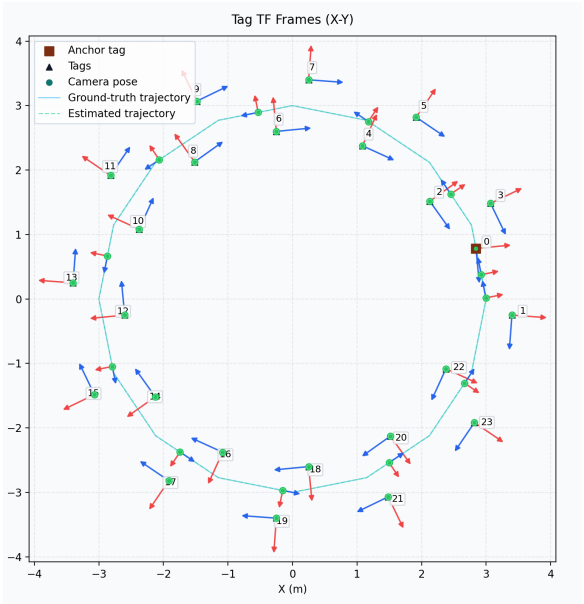


Figure 4.1: Reference trajectory (*ground-truth*) in the XY plane.



(a) Estimated trajectory in the XY plane.



(b) Visualization of the *frames* (*TF*).

Figure 4.2: Results in the XY plane.

To complement the interpretation, Figure 4.2b presents the *TF frames* associated with the tags in the XY plane, together with samples of the camera pose along the trajectory. This visualization is particularly useful for identifying axis inversions or convention inconsistencies.

4.2 Simulation in Gazebo Ignition

For the Gazebo Ignition simulation, unlike the controlled case in RViz, the system operates under conditions closer to real execution: the robot moves in the simulated world, the camera produces images at discrete time steps, and pose estimation depends directly on the quality of AprilTag detections. Thus, effects such as temporal discretization, perspective variations, noise in corner estimation, and small dynamic inconsistencies are reflected in the inferred trajectories and poses.

Therefore, Figure 4.3 shows the estimated camera trajectory in the XY plane during the Teach Phase. It can be observed that the global shape of the circuit is preserved, indicating that the chaining of transformations maintains geometric consistency along the path. However, the trajectory exhibits local irregularities and segments with greater dispersion, reflecting the cumulative nature of error when no global refinement is applied.

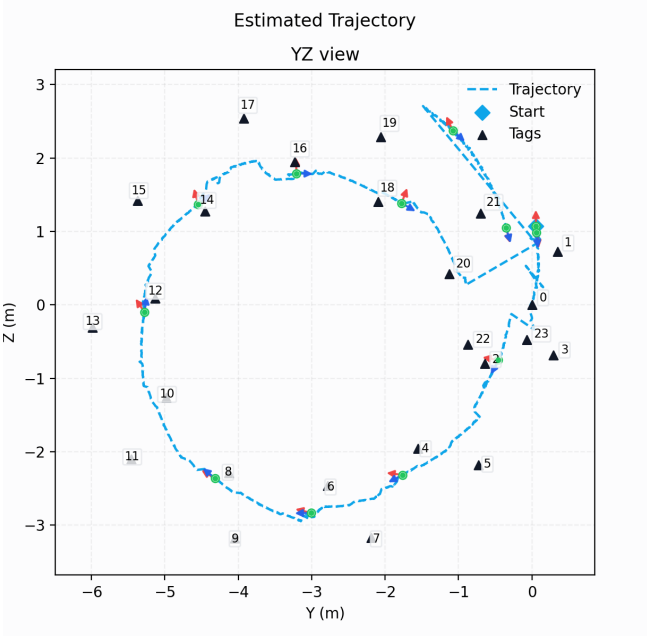


Figure 4.3: Estimated camera trajectory in the XY plane.

In this way, to evaluate how these uncertainties impact the local map of the markers, Figure 4.4 compares the estimated tag positions with the reference positions in the simulated world. It is noted that some tags present more pronounced displacements, especially in regions where the trajectory includes abrupt variations or where detection quality is potentially lower. Thus, this behavior is expected in a purely observational Teach Phase: each local estimate contributes directly to the chaining process and, without fusion of multiple observations, the discrepancy tends to vary from tag to tag.

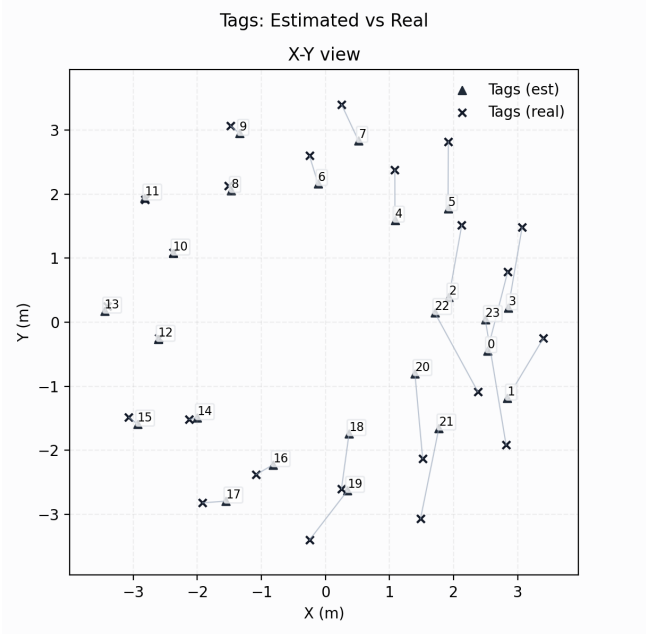


Figure 4.4: Comparison between estimated positions and reference positions of AprilTags in XY.

Finally, Figure 4.5 presents a visualization of the *TF frames* of the tags and samples of the camera pose along the trajectory. This figure is useful for diagnosing reference frame consistency: even with noise and dispersion, the axes remain coherent and no systematic orientation inversions are observed.

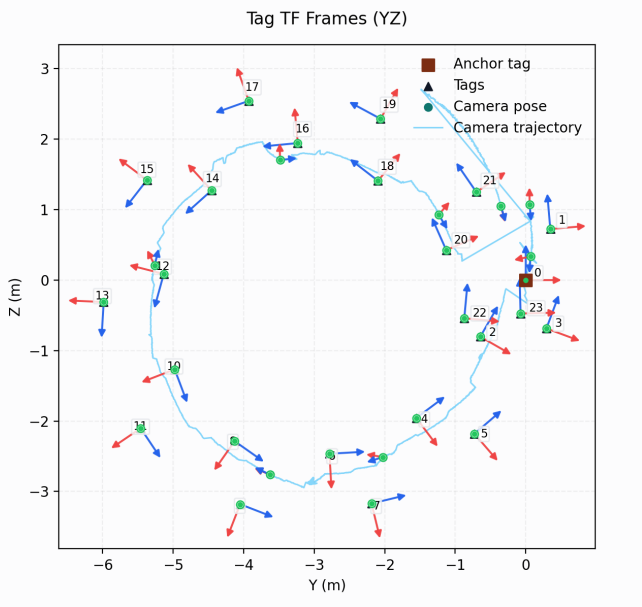


Figure 4.5: Visualization of the *TF frames*.

In summary, the simulation results highlight the expected behavior of a Teach Phase without optimization: the global structure of the circuit is preserved, but local errors progressively accumulate along the trajectory. These limitations directly motivate the introduction of the optimization stage, in which multiple observations of the same tag can be combined and the camera trajectory smoothed to reinforce the global consistency of the map.

5 Discussion

6 Conclusion

References

[1] P. Nourizadeh, M. Milford, and T. Fischer, *Teach and repeat navigation: A robust control approach*, 2024. arXiv: 2309.15405 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2309.15405.

[2] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2011, pp. 3400–3407.

[3] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

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

[5] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000. DOI: 10.1109/34.888718.

[6] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2016.

[7] M. Krogius, A. Haggenmiller, and E. Olson, "Flexible layouts for fiducial tags," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2019.

[8] *About tf2*, ROS 2 Documentation (Humble), Accessed 2026-01-22. [Online]. Available: https://docs.ros.org/en/humble/Concepts/Intermediate/About-Tf2.html.