A Fully Automated Multi-Stage Localization Pipeline for Indoor Floorplan-Constrained Trajectory Estimation

Abstract

A fully automated localization pipeline for indoor trajectory estimation under floorplan constraints is presented. The method is designed for settings where full trajectory prediction, high temporal coverage, and strict automation are all mandatory. The system combines a visual–inertial and floorplan-aware base localizer with three progressively more selective refinement modules: a joint posegraph overlay, a local depth-ray refinement stage, and sequential floorplan refinement with automatically generated same-floor reference-track anchors. Each refinement stage is allowed to replace a trajectory only when fixed machine-evaluated acceptance rules are satisfied, including floorplan-consistency gains, bounded geometric deformation, and non-regression of wall-crossing behavior. The final method produces the complete competition package without manual trajectory editing.

1 Introduction

Indoor localization in the Hilti–Trimble challenge requires the prediction of complete trajectories rather than isolated poses. This makes the problem substantially different from single-image localization: in addition to local accuracy, the method must preserve continuity, global shape, and temporal coverage throughout the entire sequence. The challenge also imposes an important methodological constraint: fully automated solutions are preferred, while human-in-the-loop correction is not acceptable for prize-eligible submissions.

These two constraints lead to a practical design tradeoff. A method can be aggressive and attempt broad trajectory rewrites, but such systems often introduce hard-to-control geometric failures. Alternatively, a method can be conservative and accept modifications only when several independent signals indicate that the correction is both geometrically plausible and floorplan-consistent. The second strategy is adopted here.

The final system is built as a staged pipeline. A robust automated base localizer first produces complete trajectories for all runs. Several automated refinement families are then applied sequentially. Each family operates only on a machine-selected cohort of weak trajectories and may replace a trajectory only if a fixed acceptance rule is satisfied. This design keeps the entire procedure reproducible and avoids the prohibited pattern of preserving only those individual trajectories that appear favorable after manual review.

2 Method Overview

The final method consists of four consecutive stages:

1. base visual–inertial and floorplan-aware localization;
2. joint localization and posegraph auto-overlay;

Sensor bags + initial poses → base visual–inertial localization → floorplan-aware candidate scoring → joint posegraph overlay → local depth-ray refinement → sequential floorplan refinement → anchor-supported sequential refinement → validated final trajectory package

Figure 1: High-level structure of the final automated localization pipeline. Every transition in the pipeline is implemented by scripts and governed by fixed acceptance rules.

3. local depth-ray refinement;
4. sequential floorplan refinement with and without automatically generated anchor support.

All stages operate on complete trajectories in the global map frame. Later stages never assume that an alternative candidate should replace the current solution by default. Instead, they evaluate a restricted candidate cohort and retain the previous trajectory whenever the proposed correction fails the stage-specific acceptance rule.

3 Base Localization Pipeline

The base stage estimates a complete map-frame trajectory for each sequence by combining visual–inertial odometry with floorplan-aware hypothesis selection. Let $z_{1:T}$ denote the sensor stream of a run and let $T_0 \in SE(3)$ denote the provided initial pose prior. A visual–inertial estimator first produces a raw trajectory in its native frame. This trajectory is then transformed into the map frame using the calibrated sensor-to-camera extrinsic and the initial pose prior. If the raw estimate is unavailable or fails the automatic plausibility checks imposed by coverage and trajectory-quality diagnostics, the system falls back to one or more automatically generated same-floor reference templates. These templates are aligned to the target run by floor identity, start-pose normalization, and temporal resampling. The base stage therefore selects from a finite candidate set $\mathcal{C} = \{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(K)}\}$ rather than assuming that a single raw trajectory must be retained.

Each candidate $\mathbf{X} \in \mathcal{C}$ is evaluated against the floorplan by geometric diagnostics that measure free-space agreement, corridor-orientation agreement, outside-of-floorplan rate, wall-crossing rate, temporal support, local step regularity, and spatial spread. The nominal base trajectory is chosen by

$$\mathbf{X}^{\star}_{\text{base}} = \arg\max_{\mathbf{X} \in \mathcal{C}} \mathcal{S}_{\text{base}}(\mathbf{X}), \tag{1}$$

where $\mathcal{S}_{\text{base}}$ is the ranking objective

$$\begin{aligned} \mathcal{S}_{\text{base}} = {} & 120\, r_{\text{free}} + 20\, h_{\text{corr}} - 140\, r_{\text{out}} - 180\, r_{\text{cross}} \\ & - 90\, \max(0, \bar{s} - 0.25) - 0.8\, d_{\text{start}} - 0.3\, d_{\text{end}} \\ & - 0.02\, \max(0, d_{\text{bbox}} - 80) + 0.002\, \min(N, 8000). \end{aligned} \tag{2}$$

Here r_{free} and h_{corr} reward consistency with traversable free space and corridor structure, while r_{out} and r_{cross} penalize occupancy outside the admissible region and intersections with walls. The terms involving $\bar{s}$, d_{start}, d_{end}, and d_{bbox} suppress candidates with unstable local motion, delayed support relative to the sensor window, truncated temporal coverage, or implausible spatial expansion. The final term mildly favors sustained trajectory support without allowing trajectory length alone to dominate the ranking. Equation (2) is not intended as a probabilistic likelihood; it is an operational ranking functional for selecting among automatically generated hypotheses under floorplan constraints.

The selected nominal trajectory is then passed to the floorplan backend, which applies constrained trajectory-level correction only when its internal quality criterion improves. The resulting output is post-processed to enforce monotonic timestamps, normalized quaternions, and full competition-compliant trajectory coverage.

4 Floorplan and Sequential Modeling

Two internal components drive the later refinement stages.

4.1 Floorplan Observation Model

The floorplan module converts a trajectory into quantitative geometric observations. These include free-space occupancy ratio, corridor-heading consistency, outside rate, crossing rate, path length, and trajectory bounding-box area. These signals serve two roles: they describe the nominal trajectory and they define the gates that any corrected trajectory must satisfy.

4.2 Sequential Floorplan Backend

The sequential backend performs a constrained $SE(2)$ correction of a nominal trajectory using sparse keyframes, a beam-search lattice, and floorplan-based observation scoring. Its local observation score combines:

- floorplan ray-signature compatibility;
- local free-space compatibility;
- corridor-heading compatibility;
- yaw-hint consistency; and
- attraction to anchor observations when anchors are available.

Transition penalties discourage motion inconsistency, large step-length deviation, excessive yaw change, and wall crossings. After beam search, the sparse keyframe corrections are interpolated to the full trajectory. An optional post-optimization stage refines the result against the floorplan, and a quality guard rejects any candidate whose internal objective deteriorates.

5 Automated Refinement Stages

5.1 Joint Posegraph Overlay

The first refinement stage combines the base localization with a posegraph-derived alternative trajectory. The overlay module accepts a replacement only when the correction improves floorplan consistency while preserving global trajectory scale.

Its acceptance logic requires:

- a sufficient increase in floor-consistency score;
- a positive loop-improvement signal;
- bounded change in path length and occupied area; and

- a minimum amount of internal candidate support.

In the final pipeline, this stage activated on a single floor-5 sequence. The retained correction improved floorplan consistency while leaving the trajectory shape essentially unchanged.

5.2 Local Depth-Ray Refinement

The second stage applies a local geometry-based correction family to weak runs on top of the posegraph-refined base. It proposes small local modifications and evaluates them with a selector that combines internal candidate score, free-space gain, floorplan stability, and geometric deformation constraints.

The acceptance logic requires:

- a selector score above threshold;
- negligible increase in outside rate;
- negligible increase in wall-crossing rate;
- non-negative free-space improvement; and
- near-unit path and area preservation.

In the final method, this stage activated on one weak underground sequence, where the accepted correction was small in magnitude and primarily improved local geometric agreement.

5.3 Sequential Floorplan Refinement

The third stage introduces a weak-run selector based on current per-run trajectory quality while excluding the released public-ground-truth sequences. The goal is to reserve the sequential floorplan optimizer for trajectories that are both weak enough to justify intervention and well supported by visual observations.

The acceptance logic requires:

- membership in an automatically defined weak-run cohort;
- a minimum amount of visual support;
- successful completion of the sequential backend;
- a non-trivial change relative to the current trajectory;
- a positive internal quality gain; and
- no regression in outside or crossing behavior.

This stage activated on one weak floor-7 sequence in the final system.

5.4 Anchor-Supported Sequential Refinement

The final refinement stage extends the sequential backend with automatically generated same-floor reference-track anchors. This is the most restrictive component of the final method. It uses the same weak-run-selection principle as the previous stage but introduces additional anchor-support and shape-preservation constraints.

For each target run, the anchor-generation procedure automatically

Stage	Principal automated acceptance conditions
Joint posegraph overlay	minimum floor gain, positive loop signal, bounded path and area deformation, minimum internal candidate support
Depth-ray refinement	selector score above threshold, negligible outside-rate increase, negligible crossing-rate increase, non-negative free-space gain, bounded geometric deformation
Sequential refinement	weak-run cohort membership, sufficient visual support, successful sequential optimization, positive internal quality gain, no outside/crossing regression
Anchor-supported sequential refinement	weak-run cohort membership, sufficient visual support, sufficient anchor support, successful sequential optimization, positive quality gain, bounded path/area deformation, no outside/crossing regression

Table 1: Summary of the fixed automated acceptance rules used in the final pipeline.

1. selects same-floor reference trajectories using floor identity, reference availability, current run quality, and temporal proximity;
2. exports the necessary target and reference artifacts;
3. generates anchor observations from the reference tracks; and
4. injects these anchors into the sequential observation stream.

The acceptance rule requires:

- weak-run cohort membership;
- sufficient visual support;
- sufficient anchor support;
- successful completion of the sequential backend;
- a real trajectory change;
- positive quality gain;
- bounded path-ratio deformation;
- bounded area-ratio deformation; and
- no regression in outside or crossing behavior.

This final stage activated on one weak floor-3 sequence. Importantly, other candidates with improved internal quality were still rejected when their path or area deformation exceeded the allowable bounds. This conservative rejection behavior is fundamental to the method: it prefers stability over aggressive correction when the geometric evidence is ambiguous.

6 Automated Decision Structure

Table 1 summarizes the main acceptance logic used by the refinement stages.

7 Experimental Protocol and Validation

Validation was performed at two levels.

First, every generated package was checked syntactically: all trajectory files had to remain monotonic in time, use normalized quaternions, and satisfy the expected TUM-format structure. Coverage-preserving padding was applied automatically when needed so that a refinement stage could not silently destroy temporal support.

Second, the released public-ground-truth split was used as a proxy validation set. The final method preserved performance and full coverage on this released split, which served as a safeguard against uncontrolled regressions during development.

The final refinement family was also rerun on top of the resulting base after the last accepted correction. That rerun selected no additional trajectories, providing a clean stopping condition and reducing the risk of ad hoc post-hoc adjustments.

8 Automation Compliance

The final submission satisfies the competition preference for fully automated methods in a strict operational sense:

- the full trajectory package is generated by scripts;
- candidate cohorts are defined from machine-readable signals rather than manual inspection;
- trajectory replacement decisions are made by fixed acceptance rules;
- reference trajectories and anchors are selected automatically from machine-generated assets; and
- no final trajectory is the result of manual pose editing.

This distinction is important. During development, many research branches were explored and rejected. The final competition artifact, however, is not a manual blend of favorable cases. It is the output of a single fixed pipeline whose retained substitutions all satisfy uniform automated rules.

9 Discussion

The main strength of the final method lies in the way the stages are composed. The base localizer provides broad coverage and a stable starting point. Later stages intervene only where the current trajectory is weak and only when the available evidence is strong enough to survive explicit geometric safeguards.

This design has two practical consequences. First, it reduces the risk that a locally convincing correction damages the global path. Second, it yields a method that is easier to defend scientifically under the competition rule, because every accepted correction can be traced to a fixed decision rule rather than a manual exception.

The main limitation is that a conservative strategy may leave some weak runs unimproved when their evidence is insufficient to satisfy the acceptance rule. In the final system, that tradeoff was intentional: the primary goal was not maximal theoretical correction capacity, but a reliable fully automated final submission.

10 Conclusion

This report described a fully automated multi-stage localization pipeline for indoor floorplan-constrained trajectory estimation. The method combines a visual–inertial base localizer with progressively more selective refinement families: posegraph overlay, local depth-ray refinement, and sequential floorplan refinement with automatically generated same-floor anchors. The defining design choice is conservative automated replacement under explicit geometric and floorplan-based safeguards. This enabled the construction of the final competition submission without manual trajectory correction and in a form consistent with the requirement that prize-eligible methods remain fully automated.