Informed, Constrained, Aligned: A Field Analysis on Degeneracy-aware Point Cloud Registration in the Wild

Turcan Tuna[†]
tutuna@ethz.ch

Julian Nubert[†‡]
nubertj@ethz.ch

Patrick Pfreundschuh[§]
patrick.pfreundschuh@ethz-asl.ch

Cesar Cadena[†]
cesarc@ethz.ch

Shehryar Khattak[*]
shehryar.khattak@jpl.nasa.gov

Marco Hutter[†]
mahutter@ethz.ch

Abstract

The ICP registration algorithm has been a preferred method for LiDAR-based robot localization for nearly a decade. However, even in modern SLAM solutions, ICP can degrade and become unreliable in geometrically ill-conditioned environments. Current solutions primarily focus on utilizing additional sources of information, such as external odometry, to either replace the degenerate directions of the optimization solution or add additional constraints in a sensor-fusion setup afterward. In response, this work investigates and compares new and existing degeneracy mitigation methods for robust LiDAR-based localization and analyzes the efficacy of these approaches in degenerate environments for the first time in the literature at this scale. Specifically, this work proposes and investigates *i)* the incorporation of different types of constraints into the ICP algorithm, *ii)* the effect of using active or passive degeneracy mitigation techniques, and *iii)* the choice of utilizing global point cloud registration methods on the ill-conditioned ICP problem in LiDAR degenerate environments. The study results are validated through multiple real-world field and simulated experiments. The analysis shows that active optimization degeneracy mitigation is necessary and advantageous in the absence of reliable external estimate assistance for LiDAR-SLAM. Furthermore, introducing degeneracy-aware hard constraints in the optimization before or during the optimization is shown to perform better in the wild than by including the constraints after. Moreover, with heuristic fine-tuned parameters, soft constraints can provide equal or better results in complex ill-conditioned scenarios. The implementations used in the analysis of this work are made publicly available to the community.[1]

---

1   Introduction

Light Detection And Ranging (LiDAR) sensors provide accurate distance measurements at high rates, which makes them suitable for reliable robot pose estimation and accurate mapping in large-scale environments. Simultaneous Localization and Mapping (SLAM) is a crucial tool to operate robots in challenging and degenerate environments [Ebadi et al., 2022]. While SLAM and robot localization can also be achieved with other sensing modalities [Kazerouni et al., 2022, Khattak et al., 2020b, Nissov et al., 2024a], the field robotics community has been focusing on improving LiDAR-based SLAM techniques for better accuracy and robustness in large-scale environments [Bavle et al., 2023, Cadena et al., 2016, Ebadi et al., 2022, Nair et al., 2024].

Most widely accepted LiDAR-SLAM frameworks leverage the point cloud registration algorithms' efficiency to provide an accurate environment map. These algorithms align pairs of point clouds with sufficient overlap by finding the six Degrees-of-Freedom (DoF) rigid transformation. One well-established point cloud registration method is the Iterative Closest Point (ICP) algorithm, often utilized in computer vision, 3D pose estimation, and robotics [Li et al., 2023]. In a typical LiDAR-SLAM framework [Khattak et al., 2020a, Jelavic et al., 2022], LiDAR scan-to-map registration is performed, which is vital to align the new information to the already generated map of the environment.

Algorithmic limitations   Despite the success of ICP and the algorithm's broad applicability, some limitations exist. These limitations include *i)* the risk of convergence to local minima, *ii)* the sensitivity to inaccurate initial transformation prior, *iii)* being susceptible to point-wise feature extraction noise, and *iv)* robustness against the lack of geometric constraints in the underlying optimization problem [Censi, 2007, Brossard et al., 2020]. Notably, while the first three sources of error can be directly addressed by modifying the LiDAR-SLAM framework, the underlying optimization problem's ill-conditioning (*iv)*) directly stems from the environment's lack of information along specific directions and cannot be overcome easily.

Degenerate point cloud registration   While the robotic community has developed LiDAR-based mapping frameworks [Vizzo et al., 2022, Xu et al., 2022, Jelavic et al., 2022] that function robustly in most environments, point cloud registration and, subsequently, mapping in degenerate environments (without a sufficient amount of distinctive features) still poses a challenge to these frameworks due to a lack of present information in the data. State-of-the-art methods [Tuna et al., 2023, Zhang et al., 2016, Westman and Kaess, 2019] particularly targeted to those scenarios divide the problem into two steps: *i)* analysis and detection of the degenerate directions and, *ii)* mitigation of the adverse effects on the optimization. Notably, while research on the first point (*i)*) received considerable attention from the research community [Nobili et al., 2018, Ramezani et al., 2020, Tagliabue et al., 2020, Liu et al., 2021, Nubert et al., 2022b, Zhen et al., 2017, Zhen and Scherer, 2019, Tuna et al., 2023, Zhang et al., 2016], the number of works salvaging the information in the best possible way given the detected degenerate directions (described in *ii)*) is far more limited [Nashed et al., 2021, Tuna et al., 2023, Zhang et al., 2016]. Latter methods try to mitigate the effect of the degeneracy on the optimization and share the ability to fully benefit from the pose updates, at least in the well-constrained optimization directions, while former methods entirely skip the optimization and rely on the external odometry to propagate the ICP pose estimates. As discussed in [Tuna et al., 2023], skipping the optimization entirely, if ill-conditioned, is a plausible strategy only for short periods, particularly when the registration initial guess is unreliable and external aiding to the optimization problem is beneficial.

Contributions   In response to the research gap mentioned above, this work studies and compares new and existing degeneracy mitigation methods with various simulated and real-world experiments, such as in Figure 1, to investigate the efficacy of degeneracy mitigation in point cloud registration in complex degenerate environments. In addition to the study results, this work introduces three distinct methods to constrain the optimization problem for robust point cloud registration. Lastly, the study's results are put into context with the limitations of the state-of-the-art methods and validated through multiple robotic field experiments and simulated examples. The contributions of this work are as follows:
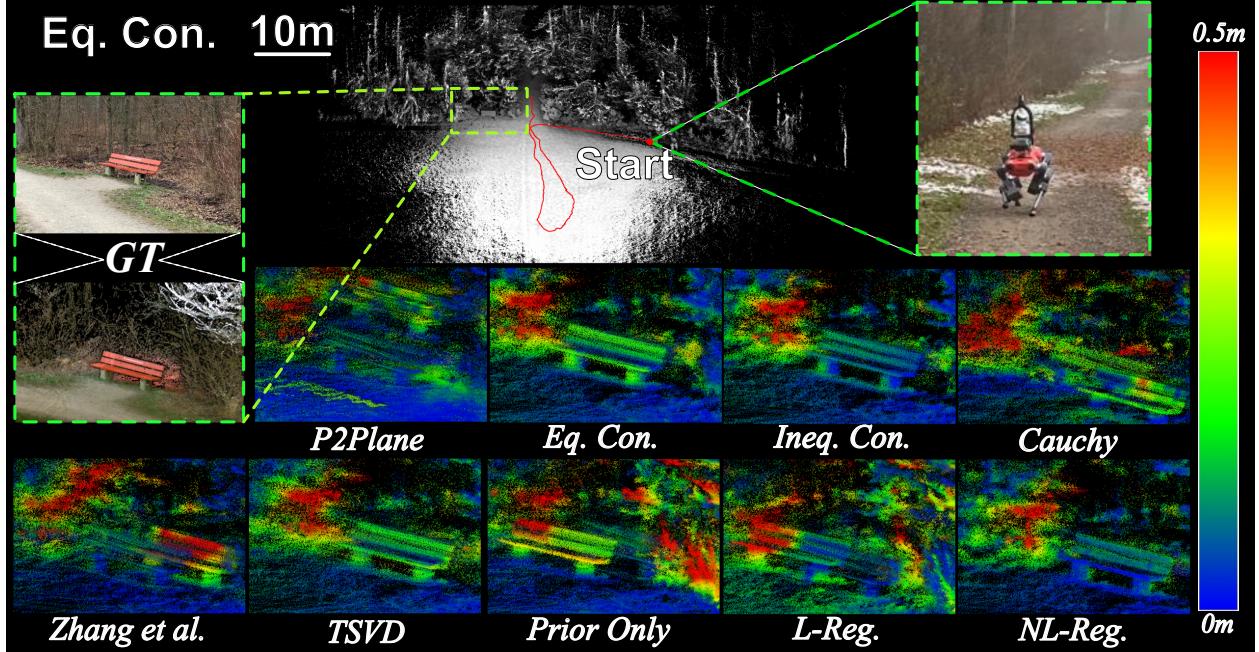
Figure 1: The mapping results of the ANYmal forest open-field degeneracy experiment (c.f. Section 5.3) are shown. **Top:** The robot trajectory generated by the Equality Constraints method is shown next to the ground truth map and images of a region of interest. **Bottom:** The registered error map for each method is shown. The color reflects the point-to-point distances to the ground truth map of the environment.

- A thorough study on the effectiveness of different constraint types for degenerate point cloud registration optimization problem for robot operation in real-world geometrically ill-conditioned environments such as natural open fields, urban tunnels, and complex construction sites.

- The implementation, investigation, and discussion of sub-space Tikhonov regularization, Truncated Singular Value Decomposition, and inequality-constrained ICP methods for the topic of degenerate point cloud registration for the first time in the literature.

- Comparison of the discussed methods in a fair and isolated fashion through simulated examples and multiple thorough robotic field experiments conducted with a legged robot, a walking excavator, and a hand-held setup which are equipped with different LiDARs.

- Open-source implementation and consolidation of all investigated methods in a single framework to foster future research on developing degeneracy-aware LiDAR-SLAM systems.

2 Related work

2.1 (Robust) Point cloud registration

Fast and accurate pose estimation through point cloud registration has been a central research topic in robot perception over multiple decades [Pomerleau et al., 2015]. Different families of solutions have been explored in this field, such as ICP and its variants [Segal et al., 2009, Babin et al., 2021, Censi, 2008, Rusinkiewicz, 2019], feature-based [Zhang and Singh, 2014, Behley and Stachniss, 2018], probabilistic [Biber and Straßer, 2003, Myronenko and Song, 2010, Maken et al., 2020], learned [De Maio and Lacroix, 2022, Bai et al., 2021a, Nubert et al., 2021] or robust global registration [Zhou et al., 2016, Lim et al., 2022] methods.

Despite the bulk of existing approaches, the simpler ICP variants using point-to-point [Besl and McKay, 1992], point-to-plane [Low, 2004], symmetric point-to-plane [Rusinkiewicz, 2019] and G-ICP [Segal et al., 2009] are among the most widely used LiDAR-based point cloud registration algorithms due to their simplicity and practicality [Pomerleau, 2013, Li et al., 2023]. All ICP variations have in common to find the refined six DoF transformation between two point clouds by repeatedly finding sets of corresponding point pairs, followed by iterative minimization of a pre-defined alignment metric.

On the other hand, the susceptibility of the ICP method to local minima encouraged researchers to investigate globally optimal and certifiable algorithms for robust point cloud registration [Yang et al., 2020b, Yang and Carlone, 2019, Carlone et al., 2015, Li et al., 2023, Lim et al., 2022]. These algorithms ensure the estimated solution corresponds to the globally optimal solution, even under outlier dominant cases. Similarly, robust M-estimators have been proposed to deal with heavy noise-dominated point cloud registration problems. These estimators down-weigh the outliers to mitigate their influence on the cost function. The well-known M-estimators such as Welsch, Tukey, Cauchy, Huber, and Geman-McClure have been used by the robotics community [Vizzo et al., 2022, Zhang et al., 2021, Babin et al., 2019]. Recently, Babin *et al.* [Babin et al., 2019] investigated the effectiveness of different robust norms on non-degenerate registration problems and concluded that, if tuned correctly, most robust norms perform similarly by increasing the signal-to-noise ratio in the optimization problem. In addition, to further mitigate the need for manual hand-tuning, researchers proposed adaptive M-estimators [Chebrolu et al., 2021] to be robust against different distributions of inputs and adaptive graduated non-convexity [Yang et al., 2020a, Yang et al., 2020b, Briales and Gonzalez-Jimenez, 2017, Lim et al., 2022] to improve robustness against large perturbations. Moreover, motivated by the success of the ICP algorithm in practical applications, researchers focused on improving the robustness of the algorithm by statistical means [Maken et al., 2020, Jian and Vemuri, 2010], with better sampling of features [Kwok, 2018, Kwok and Tang, 2016, Petracek et al., 2023], focusing on adaptive matching [Vizzo et al., 2022] and optimization acceleration techniques [Zhang et al., 2021].

Yet, none of these methods particularly tackle situations where insufficient information is contained in the underlying point cloud due to the symmetric or self-similar nature of the environment, which is often present in practice in environments such as tunnels, open planes, or narrow corridors [Tuna et al., 2023, Nubert et al., 2022b, Pfreundschuh et al., 2023].

2.2 Detection and mitigation of optimization degeneracy

2.2.1 Optimization degeneracy detection

Registration Uncertainty   Previous methods have been proposed to model the registration uncertainty and capture the degeneracy of the optimization problem by estimating the covariance of the pose estimation process [Censi, 2007, Brossard et al., 2020]. While this can be a good indicator of ill-conditioning, the obtained measures often provide overly optimistic estimates [Bonnabel et al., 2016]. Recently, Talbot *et al.* [Talbot et al., 2023] provided additional insights on accurately modeling the uncertainty of LiDAR point cloud registration information for use in a multi-modal sensor fusion framework as a loosely coupled measurement. Similarly, LOG-LIO2 [Huang et al., 2024] provides insights for modeling the uncertainty of LiDAR range measurements for tighter integration in a LiDAR inertial odometry framework by capturing the optimization errors occurring in point cloud registration.

Learning-based approaches   Recent research has investigated data-driven approaches to identify degeneracy in LiDAR information. The authors in [Gao et al., 2020] introduced a deep learning-based entropy metric by fusing covariance estimation with localizability information of the environment. To avoid complicated monte-carlo sampling during robot operation, Nubert *et al.* [Nubert et al., 2022b] suggested utilizing synthetic data for training while relying solely on the current LiDAR scan to compute a 6-DoF localizability metric. While data-driven methods have the potential to outperform traditional methods by modeling higher-degree dynamics of the problem, the interpretability of these methods still presents a challenge.

Optimization stability based approaches Grounded in the analysis of the optimization variables, Gelfand *et al.* [Gelfand et al., 2003] investigated the geometric stability of the point-to-plane ICP by proposing a sampling-based method to extract the most informative scan points to improve the conditioning of the optimization process. Similarly, [Zhen and Scherer, 2019] proposed directly estimating an environment's localizability using the Hessian eigenspace by measuring the sensitivity of a point and surface-normal pair constraint w.r.t to the optimization states. Comparably [Zou et al., 2023] utilized a similar formulation to generate an information matrix to identify the degenerate directions of the optimization, which is later propagated to a factor graph for accurate robot pose estimation. Other works [Cho et al., 2018, Tagliabue et al., 2020, Ebadi et al., 2021] also propose to use the optimization condition number as a single combined degeneracy metric for the full 6-DoF pose estimation.

Degeneracy awareness Among pioneering methods for degeneracy detection, [Zhang et al., 2016] introduced a detection metric in combination with a degeneracy mitigation technique named *solution remapping*. The metric identifies the degenerate directions of the optimization by analyzing the minimum eigenvalue of the optimization's Hessian matrix. Following a similar formulation, [Hinduja et al., 2019] used the optimization's relative condition number as the degeneracy detection threshold in contrast to the minimum eigenvalue. Further, the authors in [Bai et al., 2021b] proposed to simultaneously utilize the minimum eigenvalue and the condition number while additionally using the graduated non-convexity to improve the conditioning of the optimization for accurate pose estimation. Recently, [Jin and Jiang, 2024] proposed to use the minimum eigenvalue in combination with the partial condition number to improve the robustness of the degeneracy detection for accurate multi-modal sensor fusion. Similarly, DAMS-LIO [Han et al., 2023] investigates the Hessian of the optimization to acquire the degenerate directions for sensor fusion.

Towards fine-grained degeneracy handling, our earlier work X-ICP [Tuna et al., 2023] proposed a correspondence information-based degeneracy detection method to constrain the optimization problem for point cloud registration. Similarly, [Chen et al., 2024] proposes to detect LiDAR degeneracy using the correspondence information in the registration step and later utilizes graduated non-convexity to perform robust pose estimation. Recently, COIN-LIO [Pfreundschuh et al., 2023] proposes to complement the uninformative directions in a LiDAR inertial framework with LiDAR intensity information.

2.2.2 Degeneracy mitigation in robotics

For robot operation in more complex and unstructured environments, numerous approaches have been proposed to mitigate the effect of optimization ill-conditioning for LiDAR-based robot pose estimation methods. These approaches can be categorized as either *Passive* or *Active*, based on how degeneracy mitigation is achieved. More specifically, the categorization is based on whether or not an approach changes the underlying optimization cost function or the immediate relative LiDAR pose estimation of the registration, given the detected degeneracy information. Technical details are provided in Section 4.

Passive degeneracy mitigation Most passive degeneracy mitigation methods use a complementary sensor modality in a sensor fusion framework [Jin and Jiang, 2024, Nissov et al., 2024b, Wen et al., 2024, Bai et al., 2021b] or rely on an external odometry source [Khattak et al., 2020a, Pfreundschuh et al., 2023, Ebadi et al., 2021, Tagliabue et al., 2020] to propagate the robot pose estimate in case of LiDAR degeneracy. Uniquely, the authors in [Lim et al., 2023] employed an adaptive parameter setting strategy to mitigate the effects of LiDAR degeneracy. In similar reasoning, the authors in [Ferrari et al., 2024] focused on improving the correspondence search of the ICP algorithm to improve ICP quality and tangentially robustness to degeneracy. The authors in [Petracek et al., 2024] proposed a degeneracy-aware information redundancy minimizing information sampling strategy to alleviate the adverse effects of the LiDAR degeneracy.

Active degeneracy mitigation The work in [Floery, 2010] is among the first to perform active degeneracy mitigation by focusing on point cloud registration through additional constraints to reduce the incorrect point cloud registration. Similarly, [Olsson and Eriksson, 2008] concentrates on reducing the error originating from
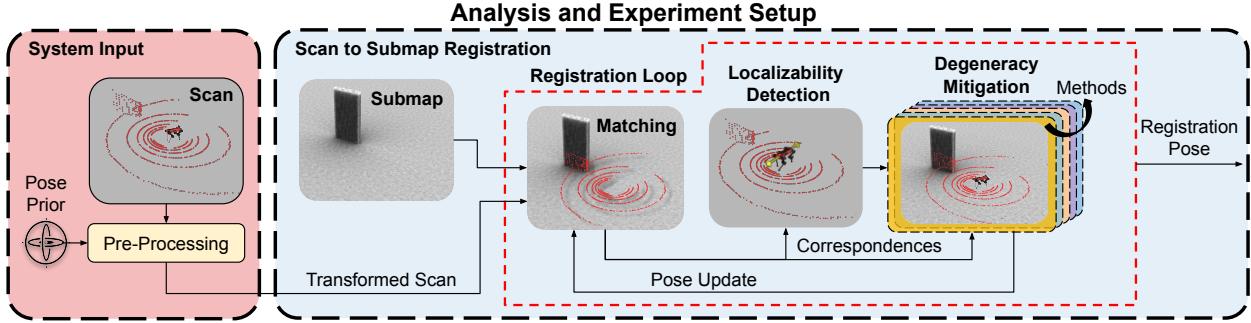
Figure 2: Overview of the analysis framework and experiment setup. The pose prior is used to transform and undistort the input point cloud, fed to the iterative registration optimization loop with the existing point cloud map. The localizability of the optimization at the current iteration is detected based on the method proposed in X-ICP [Tuna et al., 2023] and fed to the analyzed degeneracy mitigation methods (explained in Section 4). The final registration pose within the registration loop is calculated iteratively as in ICP fashion.

the linearization of the rotation variables by introducing non-linear equality constraints in the optimization. Recently, Zhang *et al.* [Zhang et al., 2016] proposed *Solution Remapping* to project the relative LiDAR pose updates from the point cloud registration only along the well-constrained directions. In X-ICP [Tuna et al., 2023], adding hard constraints along the degenerate directions was proposed to reduce the robot pose estimate error. In [Nashed et al., 2021], authors proposed using additional regularization terms in a non-linear least squares problem to mitigate the effect of optimization degeneracy. Furthermore, to the best of the authors' knowledge, [Nashed et al., 2021] is the only existing work that compares different degeneracy mitigating methods. However, the corresponding comparison is limited and does not generalize to the applications of LiDAR-SLAM and point cloud registration. Moreover, the method proposed in [Nashed et al., 2021] does not conduct field experiments and uses an RGBD camera instead of a 3D-LiDAR. As discussed, a considerable body of work focuses on LiDAR degeneracy. Despite this fact, the literature lacks a comprehensive analysis of how to effectively use the LiDAR degeneracy information. This work thoroughly analyzes and compares different methods in simulated and real-world settings to bridge this gap.

3   Problem analysis and preliminaries

This work investigates the advantages and disadvantages of different optimization degeneracy mitigation methods. For an effective analysis, the problem of degenerate point cloud registration is considered under the scope of LiDAR-SLAM in geometrically challenging environments, using only LiDAR point cloud information and external odometry estimates. This section presents the preliminaries on point cloud registration and degeneracy detection.

3.1   Point cloud registration and ICP

The point cloud registration problem aims to align two point clouds and obtain their relative rigid transformation as shown in Figure 2. To achieve this, various techniques and metrics are proposed, as discussed in Section 2. In this work, the ICP algorithm with the point-to-plane [Low, 2004] cost function is utilized as it is widely adopted by the robotics community and many point cloud registration frameworks [Jelavic et al., 2022, Tagliabue et al., 2020, Khattak et al., 2020a, Dellenbach et al., 2022, Pomerleau et al., 2013, Xu et al., 2022].

Specifically, point cloud registration focuses on estimating the homogeneous transformation $\boldsymbol{T}_{\text{ML}} \in SE(3)$ between a *source* and a *reference* point cloud of sizes N_p and N_q respectively, given an initial guess $\boldsymbol{T}_{\text{ML,init}}$.

The *source* point cloud $_{\mathtt{L}}\boldsymbol{P} \in \mathbb{R}^{3 \times N_p}$ is expressed in the LiDAR frame $\mathtt{L}$, whereas the *reference* point cloud $_{\mathtt{M}}\boldsymbol{Q} \in \mathbb{R}^{3 \times N_q}$ is expressed in (sub)map frame $\mathtt{M}$. The homogeneous transformation $\boldsymbol{T}_{\mathtt{ML}} = \left[\boldsymbol{R}_{\mathtt{ML}} \,|\, _{\mathtt{M}}\boldsymbol{t}_{\mathtt{ML}}\right]$, consists of a rotation matrix $\boldsymbol{R} \in SO(3)$, and a translation vector $\boldsymbol{t} \in \mathbb{R}^3$, denoted as $\boldsymbol{t} = [t_x, \, t_y, \, t_z]^\top$. For the N_p *source* points $_{\mathtt{L}}\boldsymbol{p}_i \in \mathbb{R}^3$ in $_{\mathtt{L}}\boldsymbol{P}$, the closest reference point $_{\mathtt{M}}\boldsymbol{q}_i \in \mathbb{R}^3$ in $_{\mathtt{M}}\boldsymbol{Q}$, is found through a K-Nearest Neighbors correspondence search given initial transformation $\boldsymbol{T}_{\mathtt{ML,init}}$ as shown in Figure 2 matching block. This transformation is often provided as an initial guess and applied to all points in $_{\mathtt{L}}\boldsymbol{P}$ to obtain $_{\mathtt{M,init}}\boldsymbol{P}$ to improve the matching process and the optimization convergence characteristics [Brossard et al., 2020] through the iterative process of ICP. In the remainder of the algorithm, all variables are expressed in (sub)map frame $\mathtt{M}$, and for brevity, the notation is simplified to omit the frame. The result of the correspondence search can be summarized as a one-to-many matching $\{\boldsymbol{p}_i, \, \boldsymbol{Q}(\boldsymbol{p}_i)\}$ for $i \in \{1, \ldots, N\}$, where $\boldsymbol{p}_i$ and $\boldsymbol{q}_i := \boldsymbol{Q}(\boldsymbol{p}_i)$ are the $N \leq N_p$ matched point pairs from *source* and *reference* point clouds, respectively. Moreover, $\boldsymbol{n}_i \in \mathbb{R}^3$, $\|\boldsymbol{n}\| = 1$ is the surface-normal vector of reference point $\boldsymbol{q}_j$, as needed for the point to plane [Low, 2004] ICP cost function. The number of matched points N indicates the size of the optimization problem. The cost function of the point-to-plane ICP minimization problem is defined as

$$\min_{\boldsymbol{R}, \, \boldsymbol{t}} \sum_{i=1}^{N} \left\| \left((\boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{t}) - \boldsymbol{q}_i\right) \cdot \boldsymbol{n}_i \right\|_2. \tag{1}$$

The problem shown in Equation (1) can be reformulated as a quadratic cost minimization problem following the derivation of Pomerleau *et al.* [Pomerleau et al., 2015] after introducing the scalar triple product and rotation matrix linearization:

$$
\begin{aligned}
&\min_{\boldsymbol{x} \in \mathbb{R}^6} \boldsymbol{x}^T \underbrace{\left(\sum_{i=1}^{N} \underbrace{\begin{bmatrix} (\boldsymbol{p}_i \times \boldsymbol{n}_i) \\ \boldsymbol{n}_i \end{bmatrix}}_{\boldsymbol{A}^{\Delta}} \underbrace{\left[(\boldsymbol{p}_i \times \boldsymbol{n}_i)^T \; \boldsymbol{n}_i^T \right]}_{\boldsymbol{A}^{\Delta^\top}} \right)}_{\boldsymbol{A}} \boldsymbol{x} \\
&-2\boldsymbol{x}^T \underbrace{\left(\sum_{i=1}^{N} \underbrace{\begin{bmatrix} (\boldsymbol{p}_i \times \boldsymbol{n}_i) \\ \boldsymbol{n}_i \end{bmatrix}}_{\boldsymbol{A}^{\Delta}} \boldsymbol{n}_i^T (\boldsymbol{q}_i - \boldsymbol{p}_i) \right)}_{\boldsymbol{b}} + \text{Const..}
\end{aligned}
\tag{2}
$$

Here, $\boldsymbol{x} = [\boldsymbol{r}^\top, \, \boldsymbol{t}^\top]^\top \in \mathbb{R}^6$ are the optimization variables, where the translational part is $\boldsymbol{t} \in \mathbb{R}^3$. Moreover, the rotation vector $\boldsymbol{r} \in \mathbb{R}^3$ where $\boldsymbol{R} \approx \boldsymbol{r}^\wedge + \boldsymbol{I}$. The $^\wedge$ "hat operator" transforms a vector to its skew-symmetric matrix representation, $\boldsymbol{r}^\wedge \in \mathfrak{so}(3)$. Here, $\mathfrak{so}(3)$ is the Lie algebra of $SO(3)$ group, which consists of skew-symmetric matrices of size 3×3.

Substituting $\boldsymbol{A}$ and $\boldsymbol{b}$ into Equation (2) results in the regular linear least squares optimization formulation

$$E(\boldsymbol{x}) = \min_{\boldsymbol{x} \in \mathbb{R}^6} \left\| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \right\|_2, \tag{3}$$

with $E(\boldsymbol{x})$ indicating the residual error of the optimization and minimization of this cost function results in the Frobenius norm (2-norm) solution. Here, $\boldsymbol{A} \in \mathbb{R}^{6 \times 6}$ denotes the Hessian of the optimization problem, and $\boldsymbol{b} \in \mathbb{R}^6$ contains the constraints imposed by the registration. The optimal translation $\boldsymbol{t}$ and rotation $\boldsymbol{r}$ vectors can be calculated at each linearization point. Due to the errors in linearization and correspondence matching, the matching and minimization operations of ICP are repeated iteratively until convergence.

3.2 Solving the optimization

To solve the problem in Equation (3), as depicted in Figure 2 as the degeneracy mitigation block, can be done with either nonlinear solvers, such as Gauss-Newton or Levenberg Marquardt (LM) optimization [Agarwal et al., 2024], or closed form solvers, such as Singular Value Decomposition (SVD) [Wall et al., 2003] or Lower-Upper (LU)-decomposition, can be used. This work uses SVD unless stated otherwise, as in Section 4.2.6 and Section 4.3.2.

The minimum of Equation (3) is found when the cost function gradient w.r.t. $\boldsymbol{x}$ is zero in all directions. Following the previous definitions,

$$
\begin{aligned}
\frac{\partial E(\boldsymbol{x})}{\partial \boldsymbol{x}} &= \frac{\partial}{\partial \boldsymbol{x}}\left[\boldsymbol{b}^\top \boldsymbol{b} - \boldsymbol{b}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}^\top \boldsymbol{A}^\top \boldsymbol{b} \right. \\
&\qquad \left. + \boldsymbol{x}^\top \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x}\right] \\
&= -2\boldsymbol{A}^\top \boldsymbol{b} + 2\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} = 0,
\end{aligned}
\tag{4}
$$

resulting in the normal equation after re-arranging the terms:

$$
\left(\boldsymbol{A}^T \boldsymbol{A}\right)\boldsymbol{x} = \boldsymbol{A}^T \boldsymbol{b}.
\tag{5}
$$

The inverse of matrix $\boldsymbol{A}^\top \boldsymbol{A} \in \mathbb{R}^{6\times 6}$ is required to acquire the optimal solution $\boldsymbol{x}$. Equation (5) can be solved with Cholesky factorization given that $\boldsymbol{A}^\top \boldsymbol{A}$ is not ill-posed (near-singular). As this work focuses on LiDAR degeneracy and subsequently ill-conditioned optimization problem, using Cholesky factorization is not suitable.

Instead, SVD, a factorization method that produces two orthonormal bases next to the diagonalized original matrix, is used to solve this equation, especially when the optimization problem is ill-conditioned, as SVD solution exists for all (non-square) matrices. For a matrix $\boldsymbol{A}$ of size $m \times n$, the decomposition is defined as

$$
\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top, \quad \boldsymbol{A}^{-1} = \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^\top,
\tag{6}
$$

with $\boldsymbol{U}$ a $m \times m$ rotation matrix, $\boldsymbol{\Sigma}$ a diagonal $m \times n$ scaling and projecting matrix that contains the singular values of the matrix $\boldsymbol{A}^\top \boldsymbol{A}$, and lastly, $\boldsymbol{V}$ the second $n \times n$ rotation matrix. For symmetric and positive (semi)-definite matrices $\boldsymbol{A}$, SVD becomes identical to eigenvalue-decomposition $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, and subsequently

$$
\begin{aligned}
\boldsymbol{A}^\top \boldsymbol{A} &= \left(\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^\top\right)^\top (\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}) \\
&= \boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^\top,
\end{aligned}
\tag{7}
$$

where $\boldsymbol{\Sigma}^2$ denotes a square diagonal matrix having as entries the squared singular values of $\boldsymbol{A}$. The right singular vectors $\boldsymbol{V}$ of $\boldsymbol{A}$ are equivalent to the eigenvectors of $\boldsymbol{A}^\top \boldsymbol{A}$, while the singular values of $\boldsymbol{A}$ are the square root of the eigenvalues of $\boldsymbol{A}^\top \boldsymbol{A}$.

After the SVD decomposition, the resulting orthonormal bases and the diagonal scaling matrix allow for the pseudo-inverse calculation:

$$
\left(\boldsymbol{A}^\top \boldsymbol{A}\right)^{-1} = \boldsymbol{V}^\top \boldsymbol{\Sigma}^{-2} \boldsymbol{V}.
\tag{8}
$$

Applying the pseudo-inversion to Equation (5) results in:

$$
\boldsymbol{x}^* = \boldsymbol{A}^{-1}\boldsymbol{b} = \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^\top \boldsymbol{b}.
\tag{9}
$$

As the SVD is based on the pseudo-inversion described in Equation (8), the optimal $\boldsymbol{x}^*$ can be found regardless of the conditioning of $\boldsymbol{A}$.

3.3 LiDAR degeneracy detection

When deploying robots in challenging featureless environments, point cloud registration can become ill-conditioned, and hence, a method to detect the degenerate directions of the optimization is required. This work uses the degeneracy detection method described in X-ICP [Tuna et al., 2023] for all analyzed methods for a fair comparison, shown in Figure 2 as a localizability detection block. Following the derivation from [Tuna
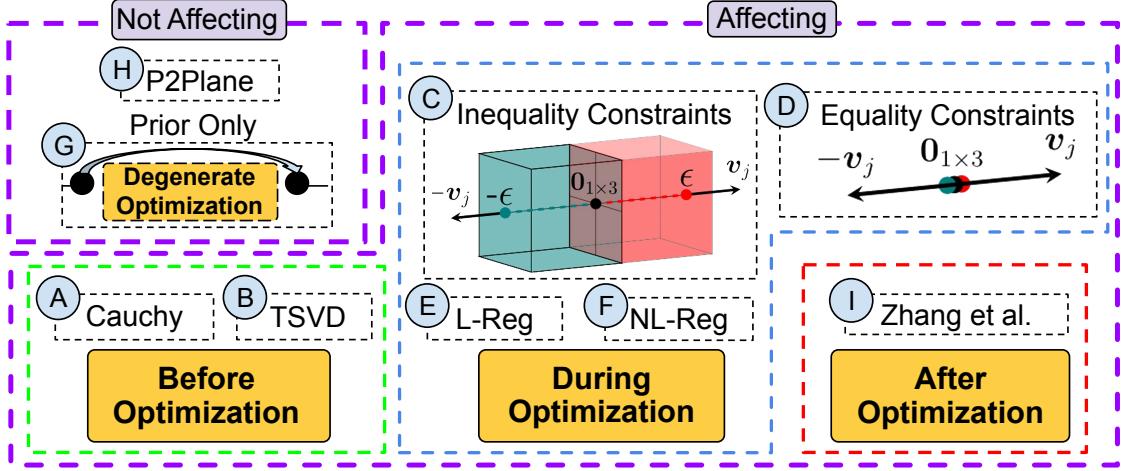
Figure 3: The overview diagram of the analyzed methods and at which stage these methods affect the point cloud registration is depicted. Methods-(**A-B**) affect before the optimization occurs. Methods-(**C-F**) directly affect or change the underlying optimization, while Method-(**I**) alters the solution after the optimization. The Method-(**G**) directly uses the external pose prior in the event of degeneracy, while the baseline Method-(**H**) directly solves Equation (3), ignoring degeneracy. A common property of affecting methods is that these methods alter the cost function of the optimization or add additional constraints to the optimization. Meanwhile, non-affecting methods do not affect the optimization.

et al., 2023], the optimization Hessian is divided into sub-matrices to identify the eigenvectors corresponding to the rotational and translational parts r and t of the optimization variables x:

$$A = \begin{bmatrix} A_{rr} & A_{rt} \\ A_{tr} & A_{tt} \end{bmatrix}_{6 \times 6}.$$

The factorized Hessians $A_{rr} \in \mathbb{R}^{3 \times 3}$ and $A_{tt} \in \mathbb{R}^{3 \times 3}$ correspond to rotation and translational information, respectively. Furthermore, A_{rr} and A_{tt} are utilized for degeneracy detection as it is not trivial to treat t and r jointly due to differences in scale and physical meaning. For the corresponding sub-matrices, the eigenvalue-decomposition can be written as:

$$A_{tt} = V_t \Sigma_t V_t^\top, \quad A_{rr} = V_r \Sigma_r V_r^\top.$$

The matrices $V_t \in \mathbb{R}^{3 \times 3}$ and $V_r \in \mathbb{R}^{3 \times 3}$ are the eigenvectors in matrix form, and $\Sigma_t \in \{\text{diag}(v) : v \in \mathbb{R}^n_{\geq 0}\}$ and $\Sigma_r \in \{\text{diag}(v) : v \in \mathbb{R}^n_{\geq 0}\}$ are the scaling matrices with the eigenvalues of A_{tt} and A_{rr} as the diagonal entries, respectively. For the remainder of the work, the null space approximation of V_t and V_r are assumed to be known through this analysis in combination with their associated eigenvalues from Σ_t and Σ_r. If an eigenvector, $v_j \in \{V_t, V_r\}$, is within the null space of V_t or V_r, it is denoted as *degenerate* (non-localizable), otherwise *non-degenerate* (localizable). In addition, the mapping $V \Leftrightarrow \{V_r, V_t\}$ is assumed to be known, as a subset of methods described later in Section 4 expect degenerate eigenvectors in $\mathbb{R}^{3 \times 3}$, while others expect them in $\mathbb{R}^{6 \times 6}$.

4 Methodology and implementation

In this section, the theoretical and implementation details of the methods compared in Section 5 are laid out. Motivated by the discussion in Section 2, the methods are classified under three categories, namely, *i) Baseline, ii) Active Degeneracy Methods* and *iii) Passive Degeneracy Methods*.

4.1 Baseline: point to plane ICP

Point to Plane ICP (P2Plane) refers to the ICP formulation using the point to plane [Low, 2004] cost function as previously detailed in Section 3. P2Plane, shown in Figure 3 as method-H, acts as a baseline and shows the adverse effects of LiDAR degeneracy on point cloud registration and LiDAR pose estimation if no degeneracy-aware actions are taken to mitigate the effects of optimization ill-conditioning. The optimization cost function of this method is defined in Equation (3).

4.2 Active degeneracy mitigation

The active degeneracy mitigation class contains the methods that utilize the detected degeneracy information to alter the optimization cost function or the optimization variables. These methods require a separate accurate degeneracy detection algorithm and often perform additional computations to mitigate the effects of degeneracy. By definition, all of the active degeneracy mitigation methods affect the optimization, as shown in Figure 3.

4.2.1 Equality constraints

Equality Constraints (Eq. Con.), as recently discussed in X-ICP [Tuna et al., 2023], utilize constrained optimization techniques to add additional equality constraints along the degenerate directions, thus preventing optimization of the solution along these directions during iterative refinement. This method is shown in Figure 3 as method-D. In the formulation of the equality constraints, the degenerate eigenvectors $\boldsymbol{v}_j \in \{\boldsymbol{V}_t, \boldsymbol{V}_r\}$ where $\boldsymbol{V}_{r,t} \in \mathbb{R}^{3 \times 3}$ will be used as constraint directions. Given the eigenvectors, the constraint formulation is

$$
\begin{aligned}
\boldsymbol{v}_j \cdot (\boldsymbol{t} - \boldsymbol{t}_0) &= 0, \quad \text{if } \boldsymbol{v}_j \in \boldsymbol{V}_t, \\
\boldsymbol{v}_j \cdot (\boldsymbol{r} - \boldsymbol{r}_0) &= 0, \quad \text{if } \boldsymbol{v}_j \in \boldsymbol{V}_r,
\end{aligned}
\tag{10}
$$

where $\boldsymbol{t}_0$ and $\boldsymbol{r}_0$ are the translation and rotation constraint values, respectively. In this work, the Eq. Con. constraint values are set to $\boldsymbol{t}_0 = \boldsymbol{r}_0 = \boldsymbol{0}$ to prevent motion updates along the degenerate directions.

To utilize these constraints in the unconstrained cost function shown in Equation (3), the equations are augmented with zero vectors to be applied to a 6-DoF optimization problem. The resulting augmented constraints are:

$$
\begin{aligned}
\left[\boldsymbol{0}_{1\times3}, \boldsymbol{v}_j\right] \cdot \boldsymbol{t} &= \boldsymbol{0}, \quad \text{if } \boldsymbol{v}_j \in \boldsymbol{V}_t, \\
\left[\boldsymbol{v}_j, \boldsymbol{0}_{1\times3}\right] \cdot \boldsymbol{r} &= \boldsymbol{0}, \quad \text{if } \boldsymbol{v}_j \in \boldsymbol{V}_r.
\end{aligned}
\tag{11}
$$

The constraints are then stacked into a matrix of form $\boldsymbol{C}\boldsymbol{x} = \boldsymbol{d}$ and integrated into the ICP problem formulation, via

$$
\underbrace{\begin{bmatrix} \boldsymbol{0}_{m_r \times 3} & \boldsymbol{v}_j \\ \vdots & \vdots \\ \boldsymbol{v}_j & \boldsymbol{0}_{m_t \times 3} \end{bmatrix}}_{\boldsymbol{C}_{(m_r+m_t)\times6}} \boldsymbol{x} = \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{d}_{(m_r+m_t)\times1}},
\tag{12}
$$

where the number of equality constraints is denoted as m_t translational and m_r rotational ones, resulting in a total number of $c = m_t + m_r$ constraints.

The unconstrained ICP cost function shown in Equation (3) can then be re-written as a constrained optimization:

$$
\begin{aligned}
\min_{\boldsymbol{x} \in \mathbb{R}^6} \quad & \left\| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \right\|_2, \\
\text{s.t.} \quad & \boldsymbol{C}\boldsymbol{x} - \boldsymbol{d} = 0.
\end{aligned}
\tag{13}
$$

Introducing Lagrangian Multipliers can convert an equality-constrained optimization problem into an unconstrained optimization problem [Tuna et al., 2023]. The Lagrangian of the minimization problem of Equation (13) is defined as

$$\mathbb{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{x}^\top \mathbf{A}^\top \mathbf{A} \boldsymbol{x} - 2\boldsymbol{b}^\top \mathbf{A} \boldsymbol{x} \\ + \boldsymbol{\lambda}^\top (\mathbf{C} \boldsymbol{x} - \boldsymbol{d}) + \boldsymbol{b}^\top \boldsymbol{b}, \tag{14}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^c$ are the c Lagrangian multipliers. Interestingly, the effect of the constraints can be inferred by the magnitude of the Lagrangian multiplier it is paired with. As the effect of the constraint gets larger, the magnitude of the Lagrangian multiplier increases accordingly. Similarly, if the magnitude of the Lagrangian multiplier is close to zero, the constraint has a minimal effect on the optimization. The effect of the constraints can be seen by comparing Equation (4) and Equation (14), with the only difference being the added constraint terms. Re-formating the Lagrangian into the matrix form reveals the familiar least squares formulation

$$\min_{\boldsymbol{x} \in \mathbb{R}^{6+c}} \underbrace{\begin{bmatrix} 2\boldsymbol{A}^\top \boldsymbol{A} & \boldsymbol{C}^\top \\ \boldsymbol{C} & \boldsymbol{0} \end{bmatrix}}_{A'} \underbrace{\begin{bmatrix} \boldsymbol{x}^* \\ \boldsymbol{\lambda}^* \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} 2\boldsymbol{A}^\top \boldsymbol{b} \\ \boldsymbol{d} \end{bmatrix}}_{b'}.$$

Here, $\boldsymbol{C} \in \mathbb{R}^{c \times 6}$ is the constraint matrix, relating the optimization variables $\boldsymbol{x}$ to the constraints $\boldsymbol{d} \in \mathbb{R}^{c \times 1}$. Furthermore, the augmented optimization vector $\boldsymbol{x}$ constitutes the optimization variables and the Lagrangian multipliers. This augmented cost function can be solved using SVD, similarly to the standard point-to-plane ICP cost function in Equation (3)

4.2.2 Inequality constraints

While equality constraints provide a straightforward way to deal with degenerate directions, the system's systematic errors, such as eigenvector calculation errors, sensor noise, and the iterative nature of ICP, might require relaxation to escape local minima. In this scenario, Inequality Constraints (Ineq. Con.) method, shown in Figure 3 as method-C, provides a simple yet powerful alternative to equality constraints with a potentially larger convergence basin:

$$\min_{\boldsymbol{x} \in \mathbb{R}^6} \quad \left\| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \right\|_2, \\ \text{s.t.} \quad -\boldsymbol{\epsilon} \leq \boldsymbol{L}\boldsymbol{x} \leq \boldsymbol{\epsilon}. \tag{15}$$

The upper and lower bounds of the inequality constraints can differ, but for simplicity, both are set to the same magnitude vector, $\boldsymbol{\epsilon}$. Importantly, to account for the scale differences between translation and rotation components, the amplitude of the rotational components are set to $\epsilon/2$ and ϵ for translation counterpart, hence $\boldsymbol{\epsilon} = [\epsilon/2, \epsilon/2, \epsilon/2, \epsilon, \epsilon, \epsilon]^\top$. Here, ϵ is selected after heuristic tuning on the Ulmberg Tunnel dataset, shown in Figure 9c, and set to $\epsilon = 0.0014$. Moreover, the constraint matrix $\boldsymbol{L}$ is generated as a row-major matrix based on the optimization's degenerate directions, with rows indicating the degenerate directions.

It is not trivial to solve inequality-constrained optimization problems as it is for equality constraints. Thus, to find the solution, the above formulation is converted to a quadratic program (QP):

$$\min_{\boldsymbol{x} \in \mathbb{R}^6} \quad \frac{1}{2} \boldsymbol{x}^\top \boldsymbol{F} \boldsymbol{x} + \boldsymbol{f}^\top \boldsymbol{x} \\ \text{s.t.} \quad -\epsilon \leq \boldsymbol{L}\boldsymbol{x} \leq \epsilon. \tag{16}$$

Here, $\boldsymbol{F} = 2\boldsymbol{A}^\top \boldsymbol{A}$ and $\boldsymbol{f} = -2\boldsymbol{A}^\top \boldsymbol{b}$. The QP problem formulation suits cases where $\boldsymbol{F}$ might be ill-posed and additionally can handle different types of constraints (equality and inequality at the same time); however, for the sake of clear comparison, only inequality constraints are used. To solve such a QP problem, the open-source QP library QPmad [Schreppel and Brembeck, 2020] is utilized, internally employing the Goldfarb and

Idnani dual primal method [Goldfarb and Idnani, 2006]. The minimum of the objective function, subject to the current active constraint set, is calculated at every iteration [Schreppel and Brembeck, 2020]. The additional inequality constraints can be removed from the active constraint set if the constraint is no longer active, which relaxes the constrained optimization. The state of the inequality constraint can be inferred from its dual variables(lagrangian multipliers).

4.2.3   Solution remapping [Zhang et al., 2016]

Solution Remapping denotes the optimization degeneracy mitigation method proposed by Zhang *et al.* [Zhang et al., 2016], shown in Figure 3 as method-I. In this method, a solution projection matrix is constructed from the eigenvalue decomposition of the optimization Hessian, which is used to project the degenerate optimization solution x to only the well-constrained directions. First, this method generates an augmented eigenvector matrix V_{aug} based on the degeneracy detection by setting the corresponding eigenvectors in V to 0. Using the introduced definitions in Section 3, the reprojection matrix is calculated as

$$S = V_{aug}{}^{-1} V. \tag{17}$$

After minimizing Equation (3), the reprojection matrix is applied to the solution x^* via

$$x_{aug}^* = S x^*.$$

For a detailed derivation, the readers are referred to the original work of Zhang *et al.* [Zhang et al., 2016].

4.2.4   Truncated SVD

Truncated SVD (TSVD) is a linear algebra method to handle ill-conditioned matrices with high-condition numbers to reduce their rank effectively and to improve their computational stability [Hansen, 1990, Draisma et al., 2018]. The TSVD of a symmetric and semi-positive definite matrix involves truncating the smallest eigenvalues while retaining only the largest eigenvalues. As a result, producing a k-rank approximation of a matrix where k refers to the top-k retained eigenvalues of the matrix. TSVD requires the eigenvalue decomposition for the Hessian defined in Equation (3), which is given as

$$A = U \Sigma V^\top.$$

Next, the diagonal scaling matrix Σ is truncated by setting the eigenvalues σ_i, $\{i \in \{1 \ldots 6\}\}$ in the diagonal entries are overwritten based on whether the paired eigenvector of the eigenvalue is along a degenerate direction. The truncated diagonal entries are denoted with σ_i^{tr}, $\{i \in \{1 \ldots 6\}\}$. If an eigenvector is along a degenerate direction, then $\sigma_i^{tr} = 10^{-4}$, not 0 for numeric stability, and if not then $\sigma_i^{tr} = {1.0}/{\sigma_i}$.

As a result, the truncated diagonal scaling matrix Σ_{tr} omits the under-constrained directions from the decomposition. The truncated Hessian can directly be used to find the solution x^*

$$x^* = A_{tr}^{-1} b$$
$$A_{tr}^{-1} = V \Sigma_{tr} U^\top.$$

More specifically, Σ_{tr} is defined as follows

$$\Sigma_{tr} = \begin{bmatrix} \sigma_1^{tr} & & 0 \\ & \ddots & \\ 0 & & \sigma_6^{tr} \end{bmatrix}$$

By truncating the eigenvalues, the constraints to mitigate the motion along the degenerate directions are directly integrated into the Hessian. By design, A_{tr} is rank-deficient; however, as SVD exists for all matrices, Equation (6) to Equation (9) can be used to acquire the solution.

4.2.5   Tikhonov regularized linear least squares

Regularization, commonly employed in optimization techniques such as Levenberg Marquardt (LM), offers an easy way to escape singularities and improve optimization characteristics. Motivated by this idea, Linear optimization with Tikhonov Regularization (L-Reg.), shown in Figure 3 as method-E, is employed for a sub-space of the linear least squares minimization based on the null-space approximation provided by the localizability information. The previously defined cost function in Equation (3) can be re-written as a regularized linear least squares problem as follows:

$$\min_x \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{L}\boldsymbol{x}\|_2^2$$
$$= \min_x \left\| \begin{pmatrix} \boldsymbol{A} \\ \sqrt{\lambda}\boldsymbol{L} \end{pmatrix} \boldsymbol{x} - \begin{pmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{pmatrix} \right\|_2^2 .$$

Here, λ is the linear regularization parameter, and $L \in \mathbb{R}^{c \times 6}$ is the constraint matrix, with c denoting the number of degenerate directions. Following a similar derivation to Equation (5), the normal equations of the regularized minimization are

$$\left(\boldsymbol{A}^\top \boldsymbol{A} + \lambda \boldsymbol{L}^\top \boldsymbol{L}\right) \boldsymbol{x} = \boldsymbol{A}^\top \boldsymbol{b}. \tag{18}$$

Immediately, λ can be identified as the regularization scaling parameter to the additional cost matrix $\boldsymbol{L}^\top \boldsymbol{L}$ induced by the regularization term. This term provides the penalization needed to minimize the projection of the solution onto the null-space approximation provided to the optimization (by improving the conditioning of the problem (cf. Section 3)). The regularization parameter $\lambda > 0$ is not known a priori and has to be determined based on the problem data. In this work, λ is selected after heuristic tuning on the Ulmberg Tunnel dataset similar to Ineq. Con. method, shown in Figure 9b, and set to $\lambda = 440$. The constraint matrix $\boldsymbol{L}$ is set as a row-major matrix consisting of the degenerate directions of the optimization, $\boldsymbol{v}_j \in \{\{\boldsymbol{V}_t, \boldsymbol{V}_r\} \in \mathbb{R}^{3 \times 3}\}$. Equation (18) can be solved with SVD as described in Section 3.

Understanding the Tikhonov regularization   It is important to understand the effect of the regularization on the underlying optimization. An analysis for the general case of $\boldsymbol{L} \neq \boldsymbol{I}$ requires the solution to the generalized SVD of $(\boldsymbol{A}, \boldsymbol{L})$. Hence, for the brevity of understanding of the effects, the constraint matrix is set as $\boldsymbol{L} = \boldsymbol{I}$ for the following explanation. Similar to Equation (9), given the SVD of $\boldsymbol{A}$, the steps to obtain the solution $\boldsymbol{x}_\lambda$ are as follows:

$$(\boldsymbol{V}\boldsymbol{\Sigma}^\top \underbrace{\boldsymbol{U}^\top \boldsymbol{U}}_{=\boldsymbol{I}} \boldsymbol{\Sigma}\boldsymbol{V}^\top + \lambda \underbrace{\boldsymbol{I}}_{=\boldsymbol{V}\boldsymbol{V}^\top})\boldsymbol{x}_\lambda = \boldsymbol{V}\boldsymbol{\Sigma}^\top \boldsymbol{U}^\top \boldsymbol{b}. \tag{19}$$

Similar terms are bundled together and re-arranged, which reveals how the regularization affects the Hessian. Multiplying and setting $\boldsymbol{y} = \boldsymbol{V}^\top \boldsymbol{x}_\lambda$, the equation can be simplified to

$$\left(\boldsymbol{\Sigma}^\top \boldsymbol{\Sigma} + \lambda \boldsymbol{I}\right) \boldsymbol{y} = \boldsymbol{\Sigma}^\top \boldsymbol{U}^\top \boldsymbol{b}. \tag{20}$$

Since $\boldsymbol{V}$ is orthogonal, the 2-norm of $\boldsymbol{y}$ is same as $\boldsymbol{x}_\lambda$. The solution to the regularized problem can then be derived using the dyadic decomposition:

$$\boldsymbol{x}_\lambda = \sum_{i=1}^{6} z_i \frac{\left(\boldsymbol{u}_i^\top \boldsymbol{b}\right)}{\sigma_i} \boldsymbol{v}_i , \quad z_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda}. \tag{21}$$

Here z is the filter coefficient damping the eigenvalues $\sigma_i < \lambda$ of $\boldsymbol{A}$ and subsequently eigenvalues $\sigma_i^2 < \lambda$ of $\boldsymbol{A}^\top \boldsymbol{A}$. Hence, the regularization mainly affects the degenerate directions and effectively changes the Hessian's eigenvalues.

Table 1: Non-linear Optimizer configuration for the NL-Reg. method.

	Value	Default
Parameter Tolerance	10^{-3}	10^{-8}
Function Tolerance	10^{-3}	10^{-6}
Gradient Tolerance	10^{-6}	10^{-10}

4.2.6 Non-Linear optimization with Tikhonov regularization

This section introduces the non-linear optimization-based regularization technique, Non-linear Optimization with Tikhonov Regularization (NL-Reg.), shown in Figure 3 as method-F. Previous works such as [Fitzgibbon, 2003, Nashed et al., 2021] utilize non-linear cost functions to improve the ICP behavior. Particularly, Nashed *et al.* [Nashed et al., 2021] proposed to incorporate the null space of the optimization as a regularization term for the non-linear cost function. Motivated by this, Tikhonov regularization is employed here with the non-linear cost function as

$$E(\boldsymbol{x})_{\mathrm{NL}} =$$
$$\min_{\boldsymbol{R},\,\boldsymbol{t}} \sum_{i=1}^{N} \left\| \underbrace{\left((\boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{t}) - \boldsymbol{q}_i\right)\cdot \boldsymbol{n}_i}_{e_i} \right\|_2 + \lambda_D \|\boldsymbol{L}\boldsymbol{x}\|_2^2, \tag{22}$$

where λ_D is the degeneracy regularization parameter and $\boldsymbol{L}$ is the row-major constraint matrix consisting of the degenerate directions of the optimization. Similar to previous methods, the value of λ_D is selected after heuristic tuning on the Ulmberg Tunnel dataset, shown in Figure 9a, and set to $\lambda_D = 675$. The well-known LM algorithm is used for the minimization of the cost function in Equation (22). The Jacobian and Gradient and Hessian of cost $E(\boldsymbol{x})_{NL}$ with respect to $\boldsymbol{x}$ is defined as

$$\boldsymbol{J}_{\mathrm{NL}} = \frac{\partial e}{\partial \boldsymbol{x}}$$
$$\nabla E(\boldsymbol{x})_{\mathrm{NL}} = 2\boldsymbol{J}_{\mathrm{NL}}^\top \boldsymbol{e_i} + 2\lambda_D \boldsymbol{L}^\top \boldsymbol{L}\boldsymbol{x},$$
$$\boldsymbol{H}_{\mathrm{NL}} = \frac{\partial^2 E(\boldsymbol{x})_{\mathrm{NL}}}{\partial \boldsymbol{x}^2} = 2\boldsymbol{J}_{\mathrm{NL}}^\top \boldsymbol{J}_{\mathrm{NL}} + 2\lambda_D \boldsymbol{L}^\top \boldsymbol{L}.$$

Utilizing these definitions, the update rule of the LM algorithm is derived as follows:

$$\boldsymbol{x}_{\mathrm{new}} = \boldsymbol{x} - (\boldsymbol{H}_{\mathrm{NL}} + \lambda_s \boldsymbol{I})^{-1} \nabla E(\boldsymbol{x})_{\mathrm{NL}}. \tag{23}$$

Here, λ_s refers to the smoothness parameter of the LM algorithm, ensuring the stability of the matrix inversion. As discussed in the Ceres documentation [2], the magnitude of the state correction might lead to a non-convergent algorithm. Hence, a trust region constraint is added to the solution. The reader is referred to the documentation of Ceres for more implementation details on robust non-linear least squares. For clarity, the altered optimizer parameters are provided in Table 1. The high-level ICP registration optimization termination criteria are identical for all methods, and the non-linear optimization termination is determined by the parameters partly defined in Table 1.

4.3 Passive degeneracy mitigation

Passive degeneracy mitigation refers to methods not inherently (*actively*) affected by the degeneracy detection. Instead, these methods either use this information passively or rely on algorithms that are supposedly less affected by disturbances and degeneracy.

---

[2] http://ceres-solver.org/

4.3.1 External prior

In this approach, if optimization ill-conditioning is detected, the initial guess $\boldsymbol{T}_{\mathrm{ML,init}}$ is directly used for point cloud registration. Consequently, this operation renders the point cloud registration ineffective by skipping the optimization process altogether. This method relies on external odometry estimates during degeneracy to propagate the solution; hence, it is denoted as *Prior Only*. This method is shown in Figure 3 as method-G.

4.3.2 Globally optimal point cloud registration

In Quatro [Lim et al., 2022], the authors define *degeneracy* in globally optimal point cloud registration context as the *absence of inlier correspondences* in the matching process. This definition differs from the LiDAR-degeneracy definition used in this work (Section 3), which focuses on the inherent lack of geometric information the environment provides along specific directions. Quatro [Lim et al., 2022] is a robust globally optimal point cloud registration method estimating 4-DoF transformation while requiring only less than three correspondences. Fast Global Registration (FGR) [Zhou et al., 2016] is a well-known global registration method that does a single-stage objective minimization to find the transform between two overlapping surfaces. Section 5.1.2 highlights that these globally optimal point cloud registration frameworks are not able to compensate for the lack of geometric constraints provided by the environment; inlier point correspondences also do not constrain the optimization along symmetric and self-similar directions of motion.

4.3.3 M-Estimators

To analyze the efficacy of robust outlier rejection methods in point cloud registration, this work investigates the Cauchy M-Estimator with a Median of Absolute Deviation (MAD) as the scale estimator [Babin et al., 2019]. In the remainder of this work, this method is referred to as *Cauchy* for simplicity and shown in Figure 3 as method-A. A non-convex robust cost function can be used in an Iterative Re-weighted Least Squares (IRLS) [Bergström and Edlund, 2014] fashion, adapting the point cloud registration cost function defined in Equation (3) as

$$\min_{\boldsymbol{x} \in \mathbb{R}^6} \rho \left(\left\| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \right\|_2 \right). \tag{24}$$

As Babin *et al.* [Babin et al., 2019] found the Cauchy robust cost function $\rho\left(\cdot\right)$ to be effective against the presence of outliers, it is selected as the cost function studied in this analysis. The corresponding function and the weight $w\left(\cdot\right)$ are formulated as follows:

$$\rho\left(e\right) = \frac{\kappa^2}{2} \log\left(1 + \left(\frac{e}{\kappa}\right)^2\right), \quad w\left(e\right) = \frac{1}{1 + \left(e/\kappa\right)^2}. \tag{25}$$

Here, e refers to the error term induced by each correspondence pair in the iterative process of ICP, and κ refers to the function tuning parameter. For details of the MAD scale estimator and how to solve the IRLS problem, the readers are referred to Babin *et al.* [Babin et al., 2019]. To mitigate the effect of selecting the proper M-Estimator parameter, the proposed value of $\kappa = 1$ has been selected as suggested in [Babin et al., 2019].

4.4 Implementation details

All methods described in this section are implemented in the *libpointmatcher* [Pomerleau et al., 2013] open-source registration library and are integrated into the *Open3d SLAM* [Jelavic et al., 2022] framework as the scan-to-submap registration module. *Open3d SLAM* runs at the LiDAR rate of $10hz$, and utilizes external odometry pose estimates as the registration prior $\boldsymbol{T}_{\mathrm{ML,init}}$, enabling scan-to-submap nearest-neighbor search to identify the point correspondences. The surface normal of a point is calculated using k-nearest-neighbor search and principal component analysis, where the neighborhood is set as $k = 10$. The LiDAR scans are cut
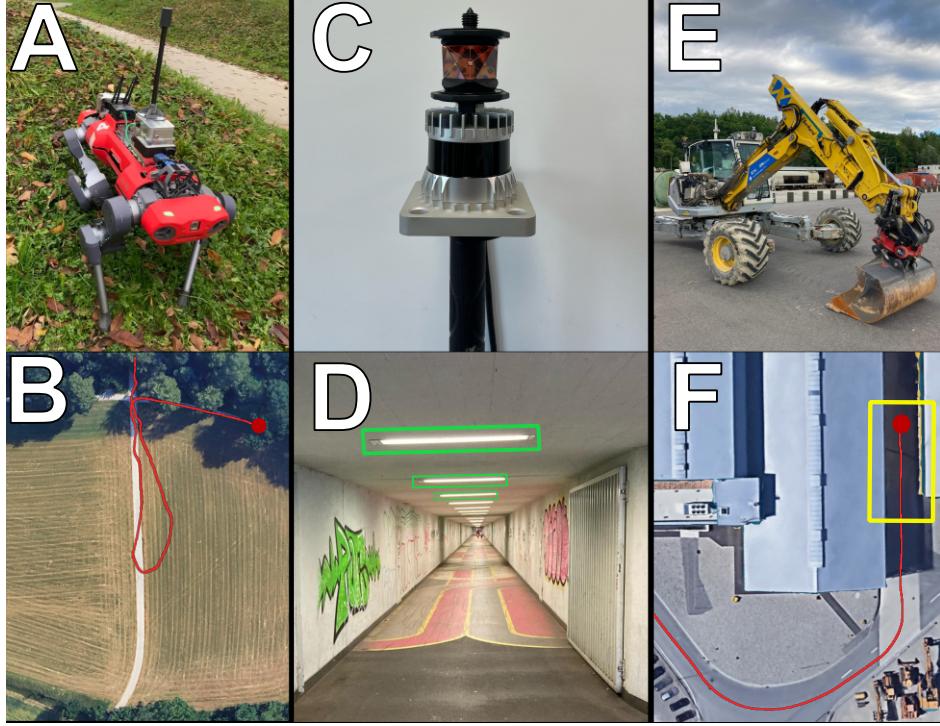
Figure 4: An illustration of three real-world deployments with environmental LiDAR degeneracy. **A:** The ANYmal robot in the deployed environment. **B:** The robot's GPS trajectory (in red) over a satellite image of the deployed environment is shown. The shown open field is subject to 3-axes LiDAR degeneracy. **C:** The handheld sensor payload of the Ulmberg bicycle tunnel [Pfreundschuh et al., 2024] experiment is shown. **D:** The degenerate bicycle tunnel is shown where the sunken LED light housings are highlighted. **E:** The HEAP [Jud et al., 2021] walking excavator. **F:** The construction environment for the excavator experiment is shown with the trajectory of HEAP overlaid as a red line. The LiDAR degenerate narrow-passage (highlighted in yellow) at the start (red dot) poses a significant challenge to LiDAR-SLAM solutions.

off at a $70m$ maximum distance and processed without downsampling. Furthermore, the trimmed outlier filter is utilized in all methods except the Cauchy method to replicate a realistic system. The methods requiring heuristic parameter setting, L-Reg., NL-Reg., and Ineq. Con. are tuned based on the performance in the Ulmberg bicycle tunnel experiment (Section 5.2). All evaluations were performed on a desktop computer with an *Intel i9 13900K CPU*, and to guarantee repeatability and deterministic execution, all operations are performed in a single-threaded fashion.

5 Experiments and evaluation

In this section, the different degeneracy mitigation methods are analyzed in depth against simulated and real-world scenarios to identify each method's advantages and weaknesses.

First, a standalone degenerate point cloud registration experiment is done in Section 5.1 to analyze each method's feasibility and low-level effects. Later, a simulated global point cloud registration study is discussed in Section 5.1.2, showing the impact of global point cloud registration methods in mitigating LiDAR degeneracy. Finally, in Section 5.1.3, the compared methods are tested on simulated data with Open3D-SLAM [Jelavic et al., 2022] in the loop.

Moreover, the methods are tested in different real-world deployments as shown in Figure 4, such as natural

open fields(Section 5.3), urban tunnels(Section 5.2), and construction sites(Section 5.4). In combination with diverse robotic platforms and sensor payloads, *i)* the ANYmal legged robot [Hutter et al., 2017] equipped with a Velodyne VLP-16 LiDAR (Section 5.3), *ii)* the HEAP [Jud et al., 2021] excavator equipped with an Ouster OS0-128 LiDAR (Section 5.4), and *iii)* the ENWIDE hand-held sensor pack [Pfreundschuh et al., 2024] equipped with an Ouster OS1-128 LiDAR (Section 5.2).

Each experiment has a unique subset of challenges, posing real challenges for existing point cloud registration frameworks, allowing for a detailed and insightful analysis of the compared methods.

Importantly, some of these unique challenges render the feasibility of recovering the true point cloud map of the environment. As an example, in the Ulmberg tunnel experiment(Section 5.2), the LiDAR degeneracy is continual for more than 80% of the dataset, making it infeasible for LiDAR-based solutions to provide a reasonable map. In this kind of scenario, the performance is mainly driven by better usage of the external motion prior to and mitigation of this LiDAR degeneracy. On the other hand, the experiments discussed in Section 5.3 and Section 5.4 either have mild LiDAR degeneracy where some information is still present or a short duration of LiDAR degeneracy where then, a scan-to-submap based registration can recover from.

5.1 Simulation studies

A set of simulated experiments is conducted to understand the applicability and effectiveness of each method. First, a standalone single registration experiment is conducted in Section 5.1.1 to isolate the point cloud registration from the mapping framework. The registration includes two point clouds with known perturbation and transformation initial guesses. Secondly, Section 5.1.2 analyzes the feasibility of globally optimal point cloud registration using the same setup. Finally, a simulated ANYmal [Hutter et al., 2017] experiment is conducted with *Open3d-SLAM* in the loop to simulate a robotic application to perform LiDAR-SLAM.

5.1.1 Static simulation - single registration

In this experiment, each method is provided with source and reference point clouds sampled from a simulated and perfectly cylindrical and symmetric mesh, shown in Figure 5, the reference point cloud is perturbed with a known transformation and an identity initial guess associates point clouds. The purpose of this experiment is two-fold: first, to analyze the performance of each method in a rotation-only (rotation around the Z-axis) degenerate setting and, secondly, to inspect the evolution of the solutions per iteration of the ICP algorithm.
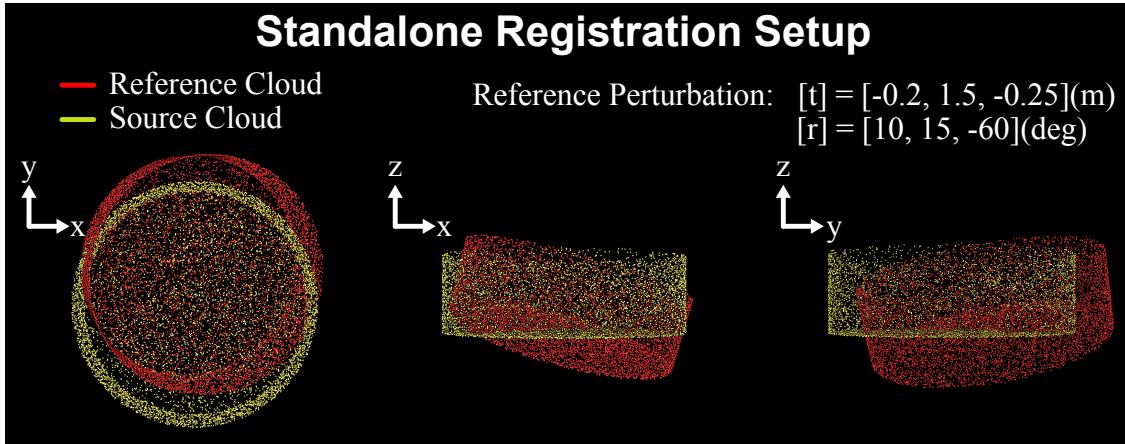

Figure 5: Setup of the static simulation experiment is shown. The red reference point cloud is shifted by r and t relative to the source point cloud. The registration goal is to find the best-aligning transformation between the two.
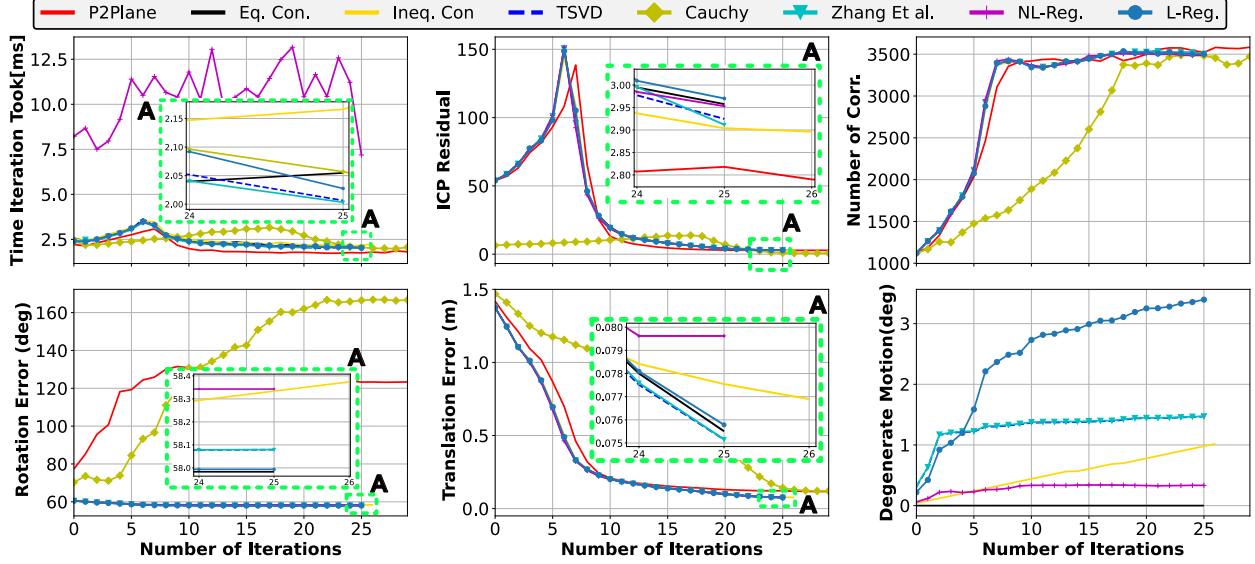
Figure 6: Analytics of the degenerate point cloud registration for the static single registration experiment (Section 5.1.1). **Top Left:** Computational time took for each iteration. **Top Middle:** The residual cost of the ICP formulation at the time of each iteration. **Top Right:** The number of correspondences at the time of each iteration. **Bottom left/middle:** The rotation and translation errors of the registration. **Bottom Right:** The motion residual along the degenerate direction during the registration process.

As this experiment focuses on the registration task, the *Prior Only* method is skipped. The methods are employed for a single ICP registration task with a maximum iteration count of 30, with collected statistics as shown in Figure 6.

Firstly, all the methods except NL-Reg. are comparable in computational complexity and suited for real-time deployment (top left). Secondly, the ICP residual is minimized for all methods to a similar magnitude after a spike in the beginning (top middle) for all methods except Cauchy due to a rapid increase in correspondence inliers (top right). However, this minimized cost does not necessarily reflect convergence to the actual transformation; it only indicates a successful residual minimization. Notably, this can be seen in the increasing rotation errors of P2Plane and Cauchy (bottom left). On the other hand, all active degeneracy mitigation methods correctly utilize the degeneracy information and prevent further increases in rotational error while decreasing the translation error (bottom middle). Interestingly, comparing the total motion residual in the degenerate direction (rotation around Z) among the active degeneracy mitigation methods reveals a difference in internal dynamics. As expected, Eq. Con. does not produce any motion residual as the degeneracy constraints are utilized as equality constraints. The out of the soft-constrained methods, NL-Reg. allows some motion until iteration 10 while Ineq. Con. allows a linear *creep* motion as the optimization can move along the degenerate direction in every registration step. On the other side of the spectrum, L-Reg. allows for motion in the degenerate direction, possibly due to a lower regularization parameter. Finally, TSVD and Zhang *et al.* perform very similarly since these methods both manipulate the eigenvectors of the optimization through eigenvalues either with k-rank approximation or solution projection.

5.1.2 Static simulation - global registration

In this experiment, the global registration methods Quatro [Lim et al., 2022] and FGR [Zhou et al., 2016], discussed in Section 4.3.2, are employed for a similar task as in Section 5.1.1. Moreover, similar to Section 5.1.1, a 1-axis rotational degenerate point cloud is used as shown in Figure 7 with a different perturbation to the reference point cloud for rotation around Y and X axes as Quatro can only estimate 4-DoF transforms. The purpose of this experiment is to highlight the fact that the degenerate point cloud registration problem is

not an outlier problem but, in fact, an absence of data problem. As this is an absence of data problem, the expected output from these methods is not the recovery of the true transform but, in fact, mitigation of further adverse effects of the degeneracy.

As seen in Figure 7, both methods fail to register correctly around the degenerate axis (c.f rotation around Z axis) as illustrated by the mismatched colors of the point clouds. In contrast, both methods recovered the correction translation very well. This finding supports that global registration methods are not *drop-in* solutions to the problem of degenerate point cloud registration as the LiDAR degenerate point cloud registration problem is inherently ill-defined, and there are many global-minima to recover from. Multiple configurations have been employed for both methods to ensure objective comparison, and the best-performing configuration is used in the generation of Figure 7.
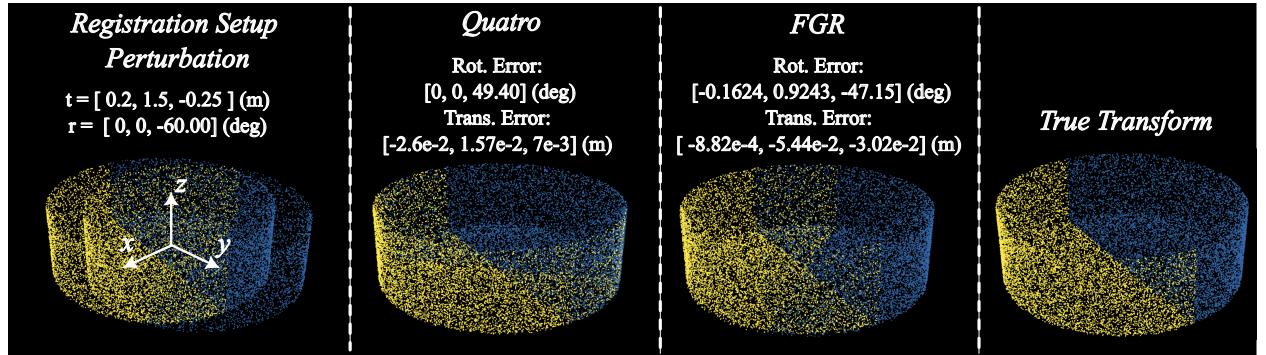

Figure 7: **Top-Row:** The target point cloud and the registration setup is provided with the expected transform. **Bottom-Row:** The results of Quatro [Lim et al., 2022] and FGR [Zhou et al., 2016] are shown with the resulting transformation errors. The point clouds are colored to visualize the final transformation, with the ideal outcome depicted in the **top right**.
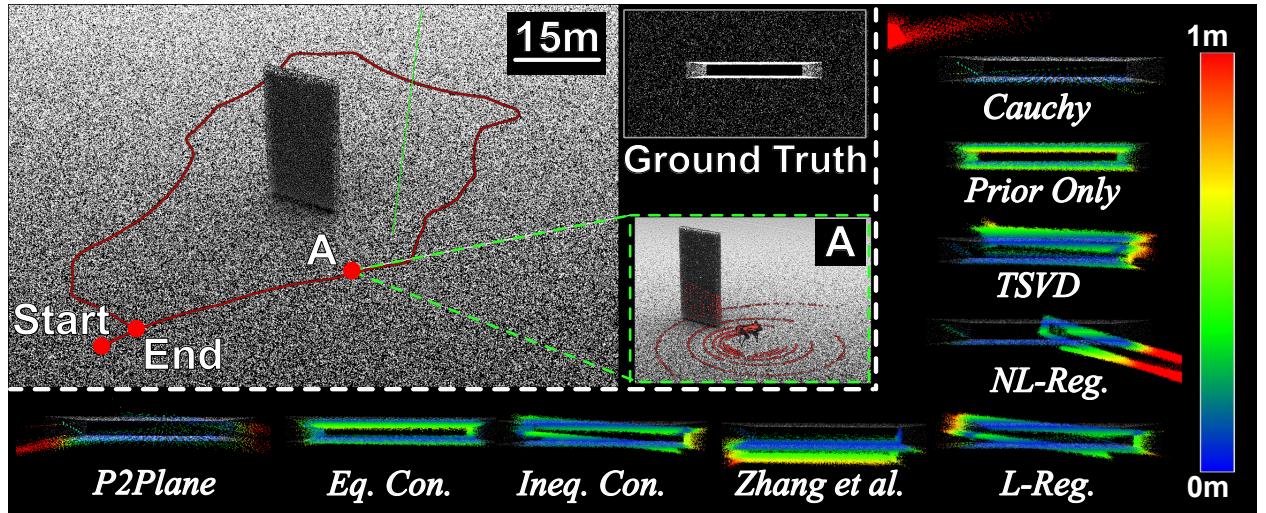

Figure 8: The mapping results of the dynamic ANYmal simulation experiment with *Open3d SLAM* in the loop are shown. **Top:** The robot trajectory is overlaid with the simulated environment ground truth map. **Bottom:** The registered maps (with the ground removed) for each method are shown next to an error color bar. Points in the maps are colored according to the point-to-point distances to the ground truth map of the environment.

5.1.3 ANYmal Simulation

In this experiment, the ANYmal robot equipped with a Velodyne VLP-16 LiDAR is simulated within a 3-axis LiDAR degenerate environment as shown in Figure 8. The robot traverses around a rectangular pillar on a planar surface, challenged with degeneracy in translation XY-axes and Z-axis rotation. To leverage the simulated environment and analyze the robustness of each algorithm, a uniformly distributed noise is added to the otherwise perfect external pose prior, which is used as an initial guess in the point cloud registration for all methods. The noise is sampled from normal distributions $\mathcal{N}(\mu_t, \sigma_t{}^2)$ and $\mathcal{N}(\mu_r, \sigma_r{}^2)$ where the distribution variables are $\mu_t = 0\,\text{cm}$, $\sigma_t = 0.05\,\text{m}$, $\mu_r = 0\,\text{rad}$, $\sigma_r = 0.01\,\text{rad}$. The mapping results of each method and the ground truth point cloud of the pillar are shown in Figure 8.

To highlight the mapping quality degradation due to LiDAR degeneracy, the ground plane is cut from the results of each method. As seen in Figure 8-A, the sparsity of the LiDAR scan and the size of the environment creates another layer of complexity since only a limited amount of points fall onto the pillar. Besides, uniquely, this experiment starts where the LiDAR is degenerate, and as a result, the degenerate scan-to-submap registration approximates degenerate scan-to-scan registration. This further cripples the optimization as the sensitivity to outliers increases since the number of correspondences is limited at this stage.

The mapping results shown in Figure 8 indicate a good performance by Eq. Con. and Prior Only methods. As expected, the P2Plane baseline and Cauchy generated a broken map. This is due to the severe LiDAR degeneracy composition of the experiment, where the robot starts and stops the simulated experiment in the presence of multi-directional LiDAR degeneracy. Morever, the TSVD, L-Reg. and Ineq. Con. methods generated a similar map with moderate drift in rotation and translation. The drift of L-Reg. and Ineq. Con. can be explained due to not well-tuned parameters, and the drift of TSVD might indicate the importance of mitigating the degeneracy during or after the optimization as this is the only active degeneracy mitigation method acting before the optimization. Lastly, the NL-Reg. method performs worse than the other active degeneracy mitigation methods but better than the passive counterparts; similar reasoning with parameter selection might explain the performance.

Table 2: APE and RPE (per 1 m distance traveled) metrics are provided for the dynamic ANYmal simulation experiment where the best is in **bold** and the second best is <u>underlined</u>.

	APE		RPE	
	Translation $\mu(\sigma)[m]$	Rotation $\mu(\sigma)[\text{deg}]$	Translation $\mu(\sigma)[m]$	Rotation $\mu(\sigma)[\text{deg}]$
Eq. Con.	<u>0.0835</u>(<u>0.0555</u>)	**0.3739**(<u>0.2066</u>)	<u>0.0224</u>(<u>0.0183</u>)	**0.3621**(**0.2413**)
InEq. Con.	0.651(0.6654)	1.900(1.610)	0.0463(0.0318)	0.4943(0.2860)
Zhang *et al.*	0.6576(0.3469)	<u>0.5212</u>(0.3251)	0.0413(0.0401)	0.4278(0.3291)
TSVD	2.05(1.23)	1.2245(0.642)	0.0391(0.0352)	0.4393(0.2892)
P2Plane	2.485(1.299)	8.31(635)	0.155(0.209)	0.9405(1.4511)
Prior Only	**0.0182**(**0.0093**)	0.5826(**0.2010**)	**0.017**(**0.007**)	0.77(0.295)
NL-Reg.	1.1699(0.9127)	3.8112(2.502)	0.0463(0.0555)	0.471(0.8862)
L-Reg.	1.1059(1.1141)	3.1277(2.4344)	0.0435(0.0422)	0.5862(0.8013)
Cauchy	1.210(0.802)	4.6278(4.1965)	0.0774(0.1323)	0.7352(1.7570)

Quantitative comparison of the methods is presented in Table 2 as relative and absolute error metrics [Sturm et al., 2012] and calculated using the EVO evaluation package[3]. The metrics show comparable performance between Eq. Con., Prior Only and Zhang *et al.* with a lead from Eq. Con. over Zhang *et al.* Since the Prior Only method utilizes the noised pose prior in the presence of degeneracy, the observed errors originate

---

[3] https://github.com/MichaelGrupp/evo

from the numeric errors of registration and mapping as well as the added noise. Interestingly, none of the tunable methods allow for constraint relaxation, Ineq. Con., L-Reg. and NL-Reg. performs comparably to the best-performing method. Intuitively, constraint relaxation indicates relying more on point cloud registration instead of the external pose prior, and as the experiment consists of multiple snippets of severe degeneracy, relying on the registration does not result in better performance. Instead, methods that do not allow relaxation perform better. On the other hand, the passive degeneracy mitigation methods show LiDAR slip and broken map.

5.2 Ulmberg bicycle tunnel experiment: one directional degeneracy

One of the common degenerate cases for LiDAR-SLAM systems is the one-directional degeneracy typical in corridor or tunnel-like structures. Ulmberg bicycle tunnel dataset [Pfreundschuh et al., 2023], depicted in Figure 4-D, represents an example of such an environment. The handheld sensor payload used in this experiment, shown in Figure 4-C, is equipped with an Ouster OS0-128 LiDAR. Importantly, as the handheld payload does not have kinematic measurements, the previously used leg-odometry estimator is not used as

(a) Effect of the regularization weight on the error metrics of NL-Reg. method for is shown. The regularization weight values with low errors are highlighted in red.

(b) Effect of the regularization weight on the error metrics of L-Reg. method is shown. The regularization weight values with low errors are highlighted in red.

(c) Effect of the upper and lower bound of the inequality constraints for the Ineq. Con. method is shown. The ϵ bound value with low errors is highlighted in red.
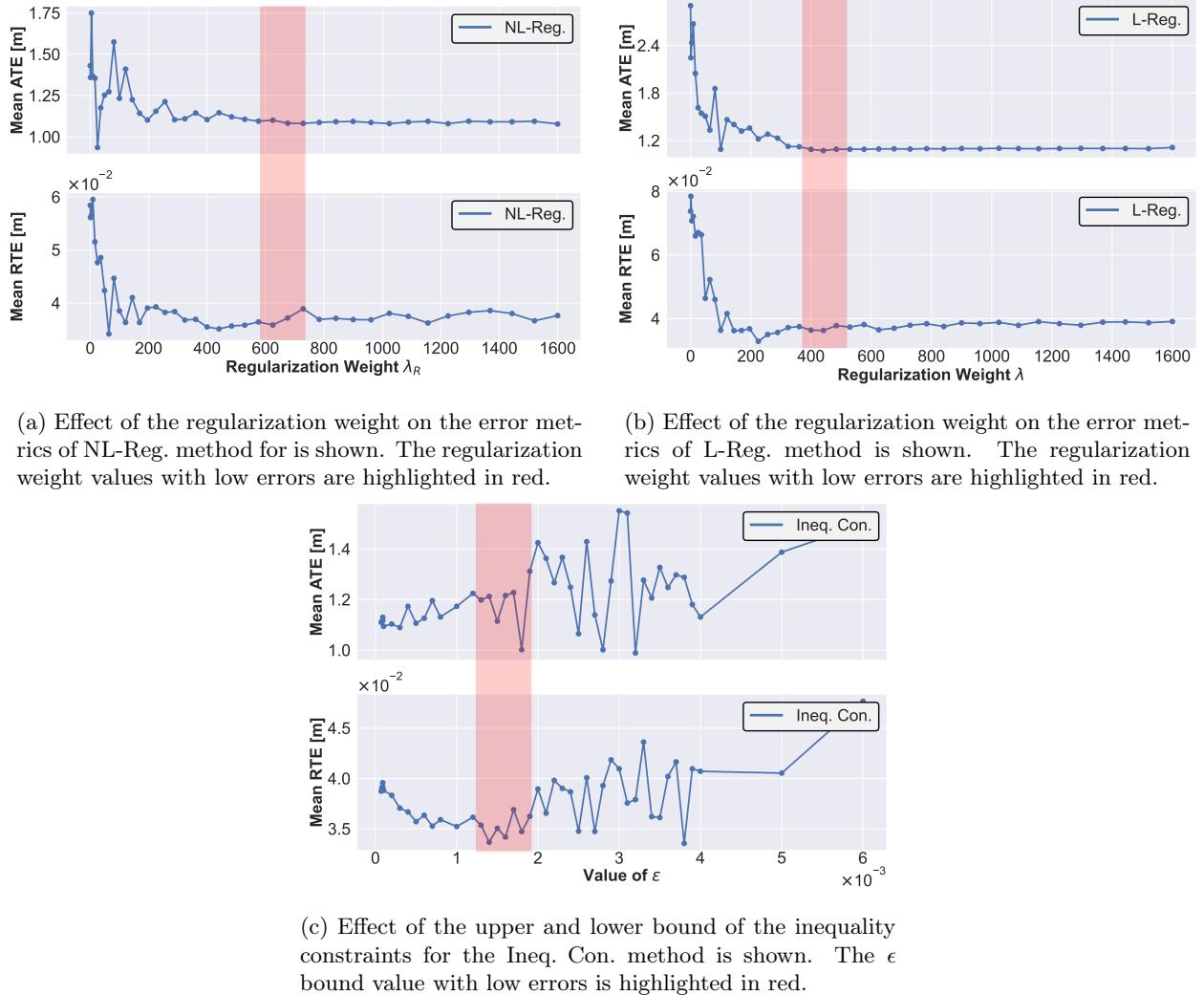
Figure 9: The effect of parameter tuning is illustrated for L-Reg., NL-Reg. and Ineq. Con. methods for the Ulmberg Bicycle tunnel experiment.

the odometry source. Instead, the intensity-based odometry estimation from COIN-LIO [Pfreundschuh et al., 2023] is a very-well suited alternative to such environments as this method utilizes the visual ques of the environment in the intensity spectrum and does not rely on the spatial features of the degraded environment (c.f. surface normals). Hence, the pose estimation of COIN-LIO [Pfreundschuh et al., 2023] is used to propagate the previous scan-to-submap registered pose during optimization degeneracy for all methods.

5.2.1 Parameter tuning

The methods Ineq. Con., L-Reg., and NL-Reg. are sensitive to parameter selection. To address this problem, the performance of these methods is analyzed using the data from this experiment. Figure 9c shows the effect of the upper and lower bound selection for the Ineq. Con. method on the Relative Translation Error (RTE) and Absolute Translation Error (ATE) metrics. When the bounds are numerically close to 0, the RTE error increases, indicating local errors; however, the decreasing ATE suggests that the constraint's increased tightness helps prevent large-scale drift. Accordingly, the bound parameter is set as $\epsilon = 0.0014$, indicating a lower bound of $-\epsilon$ and the upper bound of ϵ. This value is selected as it falls to the saddle region where the error metrics are, on average, low.

Similarly, the effect of the regularization parameter for the methods L-Reg., and NL-Reg. are provided in Figure 9b and Figure 9a, respectively, in relation to the RTE and ATE errors. Interestingly, as the regularization weight increases, both methods reach a minimum error region for the RTE error; however, further increments increase the RTE error. On the other hand, increments in the regularization weight decrease the ATE error. Based on this analysis, $\lambda = 440$ for L-Reg. and $\lambda_R = 675$ for NL-Reg. are selected and used for all the experiments in this work.

5.2.2 Global point cloud registration

A standalone registration analysis is done with real-world point clouds from the Ulmberg bicycle tunnel data to evaluate the global registration methods. The registration setup consists of two point clouds extracted from the experiment data while maintaining approximately 70% overlap between the scans. As shown in the top row of Figure 10, the source point cloud is transformed and provided as input to the global registration methods Quatro [Lim et al., 2022] and FGR [Zhou et al., 2016]. As seen from the bottom row of Figure 10, both methods fail to find the true global minima without distinguishable geometric features. Furthermore,
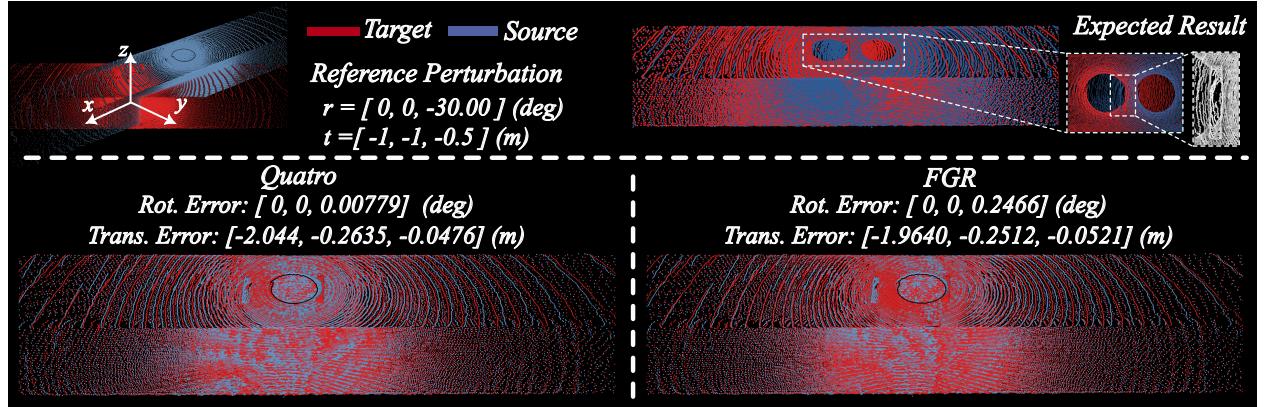

Figure 10: **Top:** The expected registration result is shown alongside the highlighted light housing of the tunnel; for better visualization of the edges, the points are in shaded gray. **Middle:** The point cloud registration setup is provided with the perturbed reference cloud. **Bottom:** The registration results from the Quatro [Lim et al., 2022] and FGR [Zhou et al., 2016] methods are shown with the associated final transformation errors (particularly in translation).

for this problem setup, the empty spots in the scan act as a false global minima, which the methods cannot distinguish from the true global minima, which can be inferred from the roof light structures illustrated in the top-row of Figure 10.

5.2.3 SLAM in the loop registration

For each method, the point cloud map of the (upon return) twice-observed stairs region is provided in Figure 11. Moreover, the ground truth map of the non-degenerate section, collected again with Leica BLK2GO scanner [Del Duca and Machado, 2023], and the real-life picture of the environment are provided in Figure 11. The mapping results show the effectiveness of active degeneracy mitigation methods, as all methods of this category generate a point cloud map with minimal or no drift. Importantly, Ineq. Con., L-Reg., and NL-Reg. methods are tuned for this environment hence, it is expected of these methods to perform well. The P2Plane and Prior only methods show map duplication along the direction of the tunnel and the Cauchy method shows the most drift among all methods.

The change in RTE per 1 m traversed distance and ATE are shown in Figure 12b and Figure 12a, respectively. As seen in Figure 12b the active degeneracy mitigation methods have lower variation in error compared to the baseline P2Plane and Cauchy methods. In comparison, the Prior Only method has a bad estimate peak around $t = 180$ s, which is sufficient to generate a misaligned map. Interestingly, a comparison of Ineq. Con. and NL-Reg. reveals that despite both methods performing well, the error variation in Ineq. Con. is higher, which results in a duplicated map at the end of the trajectory. This result might indicate stability and robustness are preferable to better local accuracy.

On the other hand, as seen in Figure 12a, all methods accumulate errors over the trajectory of the platform. The source of this error is the aleatoric uncertainty of the data and the epistemic uncertainty of Open3D-SLAM [Jelavic et al., 2022] as the map curves through the trajectory. It is also interesting to see that between $t = 40$ s and $t = 80$ s, the Cauchy method outperforms all other methods until $t = 90$ s, where it starts accumulating errors due to misregistration of the scans. Indicating that robust norms are powerful tools that might benefit accuracy; however, they still suffer from the absence of inliers.

Furthermore, statistics of these metrics are provided alongside the ICP computational cost of the ICP registration in Table 3. In the comparison of all methods, Ineq. Con. methods produce the best mean error in RTE while Eq. Con. has the least variation. Among the not fine-tuned methods, Zhang *et al.* performs the best in terms of mean RTE error. Moreover, NL-Reg. performs the best in terms of ATE error, and Prior Only has the least variation. As NL-Reg. is fine-tuned to this environment, the performance is not surprising. However, the computational cost of this method makes it less applicable to compute limited
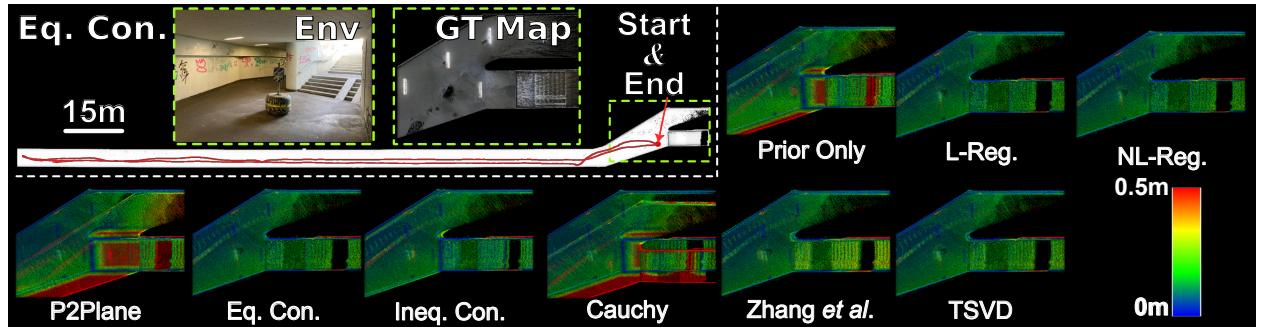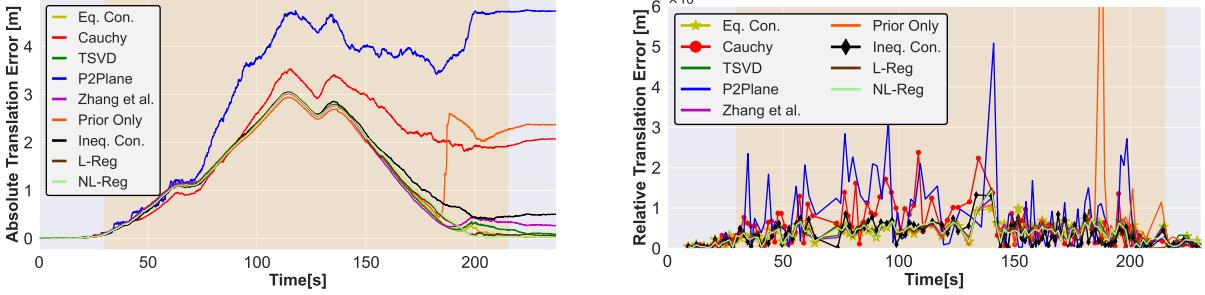

Figure 11: The mapping results from the Ulmberg bicycle tunnel experiment are shown. **Top:** The ground truth map of the tunnel and the real-world image of the region of interest are shown. **Bottom:** The registered maps for each method are shown next to an error color bar. Points in the maps are colored according to the point-to-point distances to the ground truth map of the environment.

(a) The ATE for each method is provided over the duration of the experiment. The LiDAR degenerate region is highlighted in orange. The common increase in error at the middle of the trajectory is mainly due to the bending of the map due to errors in surface normal extraction.

(b) RTE for each method is provided over the duration of the experiment. The LiDAR degenerate region is highlighted in orange. The active degeneracy mitigation methods consistently prevent higher errors.

Figure 12: The ATE and RTE error metrics are provided for the Ulmberg bicycle tunnel experiment.

Table 3: Error metrics and ICP registration computational time for the Ulmberg bicycle tunnel experiment are shown (best in **bold**), and the second best is underlined.

	RTE $\mu(\sigma)[m]$	ATE $\mu(\sigma)[m]$	Computational Cost $[ms]$
Eq. Con.	0.039(**0.020**)	1.10(1.04)	18.95(3.18)
Ineq. Con.	**0.033**(0.026)	1.21(0.97)	19.13(**2.98**)
Zhang *et al.*	0.0383(<u>0.0210</u>)	1.125(<u>0.983</u>)	18.90(3.03)
TSVD	0.0388(0.0211)	1.11(1.01)	18.95(3.00)
P2plane	0.073(0.082)	2.90(1.78)	16.96(3.50)
Prior Only	0.056(0.1075)	1.54(**0.947**)	**10.24**(9.28)
NL-Reg	<u>0.035</u>(0.022)	**1.097**(1.031)	80.14(29.04)
L-Reg	0.0358(0.0236)	<u>1.099</u>(1.030)	22.17(4.18)
Cauchy	0.050(0.046)	1.76(1.090)	18.33(3.52)

platforms. Among the not fine-tuned methods, Eq. Con. method has the least ATE error.

5.3 ANYmal forest experiment: open field degeneracy

During this experiment, the ANYmal robot, shown in Figure 4-A, starts next to a forest and navigates to an open field, does rapid rotations, and returns to the same passage it previously observed after walking for 107 m. In the open field, shown in Figure 4-B, the optimization is expected to have 3-axes degeneracy. Namely, translation on the X-Y plane and rotation around the normal of this plane. However, as the robot is close to a forest region, the tree canopy provides unstructured yet sufficient information for optimization. As a result, LiDAR degeneracy only occurs when the LiDAR loses line of sight to the forest canopy during motion. In this experiment, the ANYmal's kinematic leg odometry estimator (TSIF) [Bloesch et al., 2017] propagates the previous scan-to-submap registered pose along the degenerate direction. The ground truth trajectory is acquired through Global Navigation Satellite Systems (GNSS) positioning and used to analyze the global consistency of the estimated robot translation, as the GNSS measurements only provide position information. Moreover, the ground truth point cloud of the region of interest is collected by the hand-held Leica BLK2GO [Del Duca and Machado, 2023] scanner. Interestingly, even this commercial scanner failed

to register in the degenerate open field; hence, only the points around the re-visited area are kept for error metrics computation. The mapping results of this experiment and the ground truth map of the re-visited are provided in Figure 1. Qualitatively, the methods NL-Reg., Ineq. Con., Eq. Con. and TSVD produce a good map, inferred by inspecting the man-made bench from the environment. All these methods perform active degeneracy mitigation and act on the degeneracy before or during the optimization. It is important to note that the methods NL-Reg. and Ineq. Con. are fine-tuned for the Ulmberg bicycle tunnel experiment yet performed adequately in this experiment as well.

Table 4: Error metrics and ICP-loop computational time for the ANYmal forest experiment, (best in **bold**, slowest in red).

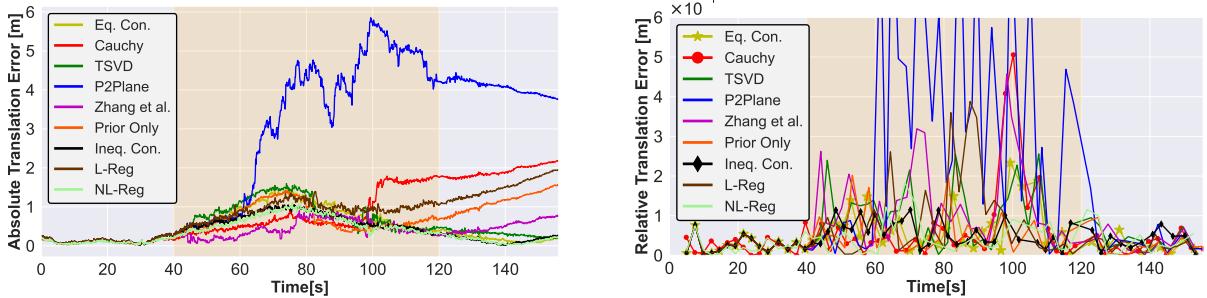|	RTE $\mu(\sigma)[m]$	ATE $\mu(\sigma)[m]$	Computational Cost $[ms]$
Eq. Con.	0.049(0.048)	0.490(0.403)	9.67(3.71)
Ineq. Con.	0.045(**0.031**)	0.405(0.302)	11.92(4.29)
Zhang et al.	0.083(0.094)	0.372(**0.224**)	10.45(4.76)
TSVD	0.068(0.066)	0.542(0.423)	10.43(4.83)
P2plane	0.212(0.281)	2.62(1.99)	11.45(6.53)
Prior Only	0.0439(0.045)	0.662(0.436)	**9.02**(5.39)
NL-Reg	**0.039** (0.037)	**0.364** (0.290)	59.86(30.20)
L-Reg	0.062(0.086)	0.843(0.517)	11.91(4.93)
Cauchy	0.051(0.086)	0.882(0.738)	8.82(**3.98**)

Out of the methods that do not require parameter tuning, Eq. Con. and TSVD perform better, possibly thanks to the degeneracy mitigation before or during the optimization. As the P2plane and Cauchy methods are blind to LiDAR degeneracy, the map generated by these methods is broken, as seen by the duplication. The Prior Only method relies on the odometry prior, and since the leg odometry estimator does not perform great on soft terrain, the generated map is also sub-optimal. Differently, Zhang *et al.* and L-Reg. methods, despite actively mitigating degeneracy, show LiDAR drift in the generated maps.

The performance of Zhang *et al.* might be explained by the difference in how the method addresses degeneracy. Since this is the only method to act on the degeneracy after the optimization.

Moreover, the performance of L-Reg. can be explained by a sensitivity to parameter tuning, as the performance of this method relies on the value of the degeneracy regularization parameter.

RTE per 2 m traversed distance and ATE metrics are provided alongside the ICP computational cost in Table 4. Quantitatively, the best method in terms of RTE and ATE mean error is NL-Reg. followed by Ineq. Con. in RTE and Zhang *et al.* in ATE. It should be noted that the good performance of Zhang *et al.* is shown in statistical metrics, but the drifted map shown in Figure 1 indicates a sub-optimal estimation in rotation as the provided metrics only measure the translation errors. To ensure an in-depth understanding of the performance differences of the compared results, the RTE and ATE errors of each method are also provided throughout the robot trajectory as shown in Figure 13b and Figure 13a. In both plots, at time $t = 100$ second, multiple methods' performance considerably changes. This point in time corresponds to the 3-axes degeneracy, including the rotation around the ground plane normal. The ATE performance of Cauchy and Zhang *et al.* is better than that of the other methods until the degeneracy at time $t = 100$ second is reached. This suggests that these methods might be sensitive to the severity of the degeneracy.

Regarding computational cost, all methods, except the best performing NL-Reg. method, behave similarly to the baseline *Prior Only* taking the lead as this method skips the registration step. However, NL-Reg. is approximately 6 times more computationally expensive as the underlying non-linear optimization takes more time to solve despite the same degeneracy constraints provided as with other methods.

(a) ATE over time of each method for the duration of the experiment. The approximate LiDAR degenerate region is highlighted in orange. Different methods accumulate different amounts of ATE.

(b) RTE over time of each method for the duration of the experiment. The LiDAR degenerate region is approximately highlighted in orange. The per 1 m traveled distance error increases on average in the degenerate region.

Figure 13: The ATE and RTE error metrics are provided for the ANYmal Forest experiment.

5.4 HEAP excavator experiment: short burst degeneracy

In this experiment, a construction site in Oberglatt, Switzerland, is traversed by a total of 170 m by the HEAP [Jud et al., 2021] autonomous excavator. As previously shown by Nubert *et al.* [Nubert et al., 2022a], the environment contains a GNSS-denied narrow passage between two buildings. HEAP and the construction environment are shown in Figure 4. The purpose of this experiment is to analyze the performance of the methods against short and severe LiDAR degeneracy while providing a good registration initial through COIN-LIO [Pfreundschuh et al., 2023]. As shown in Figure 4-F, only the between-building section of the environment is LiDAR degenerate. However, the global consistency of the LiDAR map is still compromised if degeneracy is not mitigated. The mapping result of this experiment is shown in Figure 14. The results indicate that all active degeneracy prevention methods perform comparably, given a good initial guess, regardless of the severity of the degeneracy. The accuracy of the registration initial guess can be inferred from the performance of the Prior Only method. Since the map from Prior Only method is consistent with the real-world geometry, the odometry from COIN-LIO is inferred to be good. On the other hand, the P2Plane and Cauchy methods show LiDAR slip alongside the degenerate direction as these methods do not use the degeneracy information actively in the registration process.
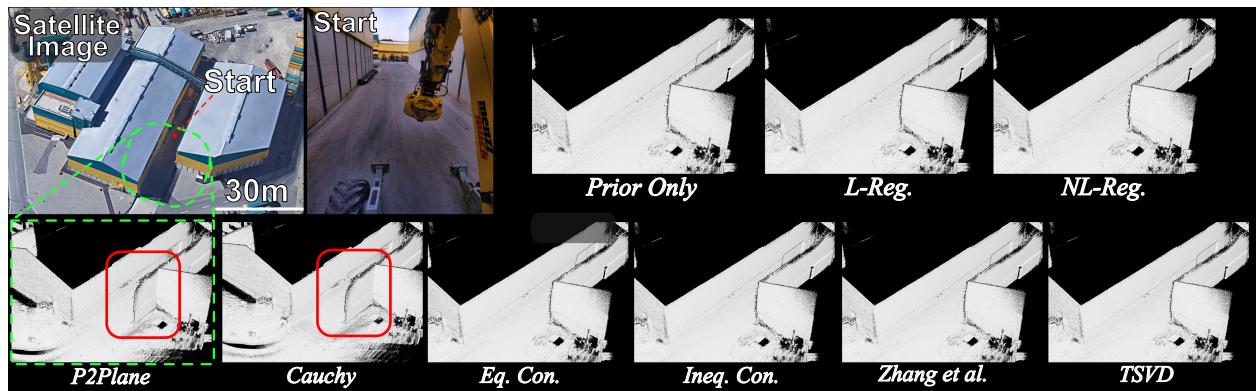

Figure 14: The mapping results of the Excavator narrow-passage experiment are shown. **Top:** The experiment site's satellite imagery is shown alongside HEAP's start location. **Bottom:** The maps from each method are shown. The effect of the LiDAR degeneracy is outlined in red for the P2Plane and Cauchy methods.

6 Practical matters and lessons learned

In this section, the authors provide insights and lessons learned on robust point cloud registration with a focus on field deployments in LiDAR degenerate environments.

Best practices for registration in field robotics   Often, field robotics implies a functional set of modules interacting with each other. These modules should be tailored to each other for the best performance, such as communication delays, sensor calibrations, and computation. From these modules, the availability of an external odometry module (e.g., Visual-Inertial odometry [Pfreundschuh et al., 2023], kinematic-inertial odometry [Bloesch et al., 2017], radar odometry, etc.) is crucial, as the authors found that reliable external odometry prior is crucial to ensure the robustness of the robot pose estimate through subsequent point cloud registration. This has been illustrated in the experiments presented in Section 5.4, as all degeneracy-aware methods performed equally given a good motion prior. Tightly-coupled degeneracy-aware solutions could mitigate this need. However, this is a topic of future research. Moreover, hardware and, subsequently, the deployed sensor setup play a crucial role. For example, a limited field of view of a LiDAR can create LiDAR degeneracy in one environment while a different sensor placement successfully localizes in the same environment. This phenomenon is realized in Section 5.3 as the VLP-16 LiDAR of ANYmal was blocked. Similarly, the handheld setup of Ulmberg Tunnel experiment Section 5.2 showed an OS0-128 is sufficient to create a degeneracy-aware solution to traverse a man-made tunnel up to the motion prior accuracy.

Importance of the software setup   Different than the robot hardware configuration, the SLAM back-end and degeneracy-aware registration framework are similarly important. Degeneracy mitigation alone is inefficient unless coupled with the correct degeneracy detection, as this facilitates degeneracy-aware point cloud registration. The detection method should be robust and generalizable to changes in environments. Furthermore, excellent and reliable spatial feature extraction (c.f. surface normals) is necessary for correct degeneracy detection. While this work does not focus on selecting the LiDAR degeneracy detection method, it is vital to mitigating its adverse effects. Solutions such as robust norms, statistical filters, and semantic filters can help mitigate noise in feature extraction. This is observed in Section 5.2 Figure 12a where the Cauchy method performed better than all other methods until LiDAR degeneracy originating error accumulated.

Besides the data and detection quality, the mapping back-end plays a core role in robust scan-to-submap point cloud registration, as the submap acts as the reference point cloud and the system's understanding of the world. As discussed in Section 5.2 Figure 12a, all active degeneracy prevention methods drift in the global scale, as a common epistemic error originating from the mapping back-end affects all the methods. Finally, the computational complexity of the system must be real-time capable and preferable at the rate of the LiDAR sensor and minimal latency to ensure the usability for downstream tasks such as navigation. Throughout the Section 5, the computational cost of NL-Reg. is highlighted as, while accurate, this method would have to skip measurements in computationally limited systems.

How to mitigate LiDAR degeneracy   Experiments in various real-world scenarios showed that the active degeneracy mitigation methods perform more accurately and reliably than the passive degeneracy methods in the presence of LiDAR degeneracy. Among the active degeneracy mitigation methods, Eq. Con., TSVD and NL-Reg. perform robustly in all experiments, as shown in Table 5, and showing good mapping performance with minimal final ATE error according to the results discussed in Section 5. Common to these methods: the constraints or the degeneracy regularization are added to the optimization *before* or *during* the optimization but not after.

Furthermore, the heuristic parameter dependent L-Reg. and Ineq. Con. methods are found to be more sensitive to parameter tuning than NL-Reg.. For instance, the best parameters identified for the Ulmberg tunnel experiment do not necessarily perform similarly in other environments, as shown in Table 5. Between different constraint types, it is hard to designate one method better than the other. Eq. Con. (equality

Table 5: Performance summary of the compared degeneracy mitigation methods. ✔ indicates no visible drift, ↔ indicates minimal drift, and ✗ indicates a broken point cloud map.

	ANYmal Forest Exp.	Ulmberg Tunnel Exp.	HEAP Excavator Exp.	ANYmal Simulation Exp.
Hard-Constrained				
Eq. Con.	↔	✔	✔	✔
Zhang *et al.*	✗	↔	✔	↔
TSVD	↔	↔	✔	↔
Soft-Constrained				
Ineq. Con.	✔	↔	✔	↔
NL-Reg	✔	✔	✔	✗
L-Reg	✗	✔	✔	↔
Unconstrained				
P2plane	✗	✗	✗	✗
Prior Only	✗	✗	✔	✔
Cauchy	✗	✗	✗	✗

constraints) method is more consistent in all environments while Ineq. Con. (inequality constraints) can perform more accurately, albeit with higher variation in estimation consistency. The TSVD method provides a good alternative solution due to the ease of implementation, minimal computation overhead, and overall consistent performance in all environments with minimal error, as seen in Table 5.

Considerations and recommendation Regardless of the performance of LiDAR degeneracy mitigation methods, some additional aspects, such as computational complexity and ease of usage, are equally important to consider. In terms of computation, all methods except NL-Reg. are close to each other and capable of doing the optimization at rates higher than the LiDAR. As the NL-Reg. method uses non-linear optimization techniques; it takes more computing power, it can fall behind the LiDAR rate, and consequently drop measurements in exchange for increased accuracy and robustness. Another consideration to take into account is the need for tuning. The methods NL-Reg., L-Reg. and Ineq. Con. require parameter tuning. As discussed in Section 5.2.1, the performance of these methods changes based on the set parameter. There are applications such as continuous inspection and surveillance where a robot would need to operate in the same environment multiple times. For such applications, the ability to fine-tune and overfit to an environment can become a benefit instead of a drawback. Similarly, in some applications, such as metric-semantic digital twin generation, the accuracy and robustness of the registration can have priority over ease of use, parameter tuning, or computational requirements.

The summary provided in Table 5 shows that Eq. Con., TSVD and Ineq. Con. did not fail in any of the experiments. Meanwhile, the P2Plane and Cauchy methods always generated a broken map. Despite failing in the ANYmal dynamic simulation experiment, NL-Reg. method generated accurate results. The method of Zhang *et al.* performed comparably with only failure in the ANYmal forest experiment.

As a recommendation, if computational resources and fine-tuning are available, NL-Reg. method is a good option. If deployed on a computationally limited system, with fine-tuning availability Ineq. Con. method has great potential since this method (thanks to the formulation of the QP problem) can also accept equality constraints. On the other hand, if fine-tuning is not an option, Eq. Con. method performed reliably throughout the experiments with good accuracy. However, if the freedom of the optimization is of utmost importance, the TSVD method showed the same reliability in exchange for accuracy but without hard-constraining the optimization.

7 Conclusion and future work

This work analyzed the efficacy of degeneracy mitigation methods in various real-world scenarios and simulated setups. New methods such as Ineq. Con., TSVD, and L-Reg. are proposed and introduced to the field of degeneracy-aware point cloud registration. In addition, previously introduced methods such as Eq. Con. [Tuna et al., 2023], Solution Remapping [Zhang et al., 2016], and Cauchy robust norm [Babin et al., 2019] are compared.

The results discussed in Section 5 and summarized in Table 5 suggest active degenerate mitigation is crucial and beneficial for LiDAR degeneracy. Furthermore, it is shown that the severity of degeneracy and the availability of a good motion prior play a crucial role in selecting the right method. Among the active degeneracy mitigation methods, solutions such as TSVD, Eq. Con., and Ineq. Con. perform more consistently. Compared to methods such as Zhang *et al.*, L-Reg. and NL-Reg.. The robust norm method, Cauchy, improved the performance of the SLAM system in the absence of LiDAR; however, it consistently failed among all the datasets when LiDAR degeneracy was present. Lastly, globally optimal point cloud registration methods are studied for degenerate point cloud registration problem. As shown in Figure 10 and Section 5.1.2, global registration methods are not to be a drop-in solution for LiDAR degeneracy, as the assumption of sufficient inliers is often violated. These results supported the claim that LiDAR degeneracy is not an outlier removal problem but, in fact, the absence of an inlier problem.

In future work, the detailed analysis of the effect of fine-tuning methods L-Reg., NL-Reg., and Ineq. Con. and ways to automate the selection of these parameters will be investigated. Since these methods have the potential to perform better if fine-tuning is possible. Lastly, as a continuation of this work, tight integration of active degeneracy mitigation methods in sensor fusion frameworks such as Graph-MSF [Nubert et al., 2022a] will be focused.

References

Agarwal, S., Mierle, K., and Team, T. C. S. (2024). Ceres Solver.

Babin, P., Dandurand, P., Kubelka, V., Giguère, P., and Pomerleau, F. (2021). Large-scale 3d mapping of subarctic forests. In *Field and Service Robotics*, pages 261–275. Springer.

Babin, P., Giguere, P., and Pomerleau, F. (2019). Analysis of robust functions for registration algorithms. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1451–1457. IEEE.

Bai, X., Luo, Z., Zhou, L., Chen, H., Li, L., Hu, Z., Fu, H., and Tai, C.-L. (2021a). Pointdsc: Robust point cloud registration using deep spatial consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15859–15869.

Bai, X., Wen, W., and Hsu, L.-T. (2021b). Degeneration-aware outlier mitigation for visual inertial integrated navigation system in urban canyons. *IEEE Transactions on Instrumentation and Measurement*, 70:1–15.

Bavle, H., Sanchez-Lopez, J. L., Cimarelli, C., Tourani, A., and Voos, H. (2023). From slam to situational awareness: Challenges and survey. *Sensors*, 23(10):4849.

Behley, J. and Stachniss, C. (2018). Efficient surfel-based slam using 3d laser range data in urban environments. In *Robotics: Science and Systems*, volume 2018, page 59.

Bergström, P. and Edlund, O. (2014). Robust registration of point sets using iteratively reweighted least squares. *Computational optimization and applications*, 58:543–561.

Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie.

Biber, P. and Straßer, W. (2003). The normal distributions transform: A new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 3, pages 2743–2748. IEEE.

Bloesch, M., Burri, M., Sommer, H., Siegwart, R., and Hutter, M. (2017). The two-state implicit filter recursive estimation for mobile robots. *IEEE Robotics and Automation Letters*, 3(1):573–580.

Bonnabel, S., Barczyk, M., and Goulette, F. (2016). On the covariance of icp-based scan-matching techniques. In *2016 American Control Conference (ACC)*, pages 5498–5503. IEEE.

Briales, J. and Gonzalez-Jimenez, J. (2017). Convex global 3d registration with lagrangian duality. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4960–4969.

Brossard, M., Bonnabel, S., and Barrau, A. (2020). A new approach to 3d icp covariance estimation. *IEEE Robotics and Automation Letters*, 5(2):744–751.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332.

Carlone, L., Rosen, D. M., Calafiore, G., Leonard, J. J., and Dellaert, F. (2015). Lagrangian duality in 3d slam: Verification techniques and optimal solutions. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*.

Censi, A. (2007). An accurate closed-form estimate of icp's covariance. In *Proceedings 2007 IEEE international conference on robotics and automation*, pages 3167–3172. IEEE.

Censi, A. (2008). An icp variant using a point-to-line metric. In *2008 IEEE International Conference on Robotics and Automation*, pages 19–25. Ieee.

Chebrolu, N., Läbe, T., Vysotska, O., Behley, J., and Stachniss, C. (2021). Adaptive robust kernels for non-linear least squares problems. *IEEE Robotics and Automation Letters*, 6(2):2240–2247.

Chen, Z., Chen, H., Qi, Y., Zhong, S., Feng, D., Jin, W., Wen, W., and Liu, M. (2024). Release: Resilient localization with enhanced lidar odometry in adverse environments. *arXiv preprint arXiv:2402.18934*.

Cho, H., Yeon, S., Choi, H., and Doh, N. (2018). Detection and compensation of degeneracy cases for imu-kinect integrated continuous slam with plane features. *Sensors*, 18(4):935.

De Maio, A. and Lacroix, S. (2022). Deep bayesian icp covariance estimation. *arXiv preprint arXiv:2202.11607*.

Del Duca, G. and Machado, C. (2023). Assessing the quality of the leica blk2go mobile laser scanner versus the focus 3d s120 static terrestrial laser scanner for a preliminary study of garden digital surveying. *Heritage*, 6(2):1007–1027.

Dellenbach, P., Deschaud, J.-E., Jacquet, B., and Goulette, F. (2022). Ct-icp: Real-time elastic lidar odometry with loop closure. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5580–5586. IEEE.

Draisma, J., Ottaviani, G., and Tocino, A. (2018). Best rank-k approximations for tensors: generalizing eckart–young. *Research in the Mathematical Sciences*, 5(2):27.

Ebadi, K., Bernreiter, L., Biggie, H., Catt, G., Chang, Y., Chatterjee, A., Denniston, C. E., Deschênes, S.-P., Harlow, K., Khattak, S., et al. (2022). Present and future of slam in extreme underground environments. *arXiv preprint arXiv:2208.01787*.

Ebadi, K., Palieri, M., Wood, S., Padgett, C., and Agha-mohammadi, A.-a. (2021). Dare-slam: Degeneracy-aware and resilient loop closing in perceptually-degraded environments. *Journal of Intelligent & Robotic Systems*, 102(1):1–25.

Ferrari, S., Di Giammarino, L., Brizi, L., and Grisetti, G. (2024). Mad-icp: It is all about matching data–robust and informed lidar odometry. *arXiv preprint arXiv:2405.05828*.

Fitzgibbon, A. W. (2003). Robust registration of 2d and 3d point sets. *Image and vision computing*, 21(13-14):1145–1153.

Floery, S. (2010). Constrained matching of point clouds and surfaces (ph. d. thesis). *Technische Universitt Wien.*

Gao, Y., Wang, S. Q., Li, J. H., Hu, M. Q., Xia, H. Y., Hu, H., and Wang, L. J. (2020). A prediction method of localizability based on deep learning. *IEEE Access*, 8:110103–110115.

Gelfand, N., Ikemoto, L., Rusinkiewicz, S., and Levoy, M. (2003). Geometrically stable sampling for the icp algorithm. In *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings.*, pages 260–267. IEEE.

Goldfarb, D. and Idnani, A. (2006). Dual and primal-dual methods for solving strictly convex quadratic programs. In *Numerical Analysis: Proceedings of the Third IIMAS Workshop Held at Cocoyoc, Mexico, January 1981*, pages 226–239. Springer.

Han, F., Zheng, H., Huang, W., Xiong, R., Wang, Y., and Jiao, Y. (2023). Dams-lio: A degeneration-aware and modular sensor-fusion lidar-inertial odometry. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2745–2751. IEEE.

Hansen, P. C. (1990). Truncated singular value decomposition solutions to discrete ill-posed problems with ill-determined numerical rank. *SIAM Journal on Scientific and Statistical Computing*, 11(3):503–518.

Hinduja, A., Ho, B.-J., and Kaess, M. (2019). Degeneracy-aware factors with applications to underwater slam. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1293–1299. IEEE.

Huang, K., Zhao, J., Lin, J., Zhu, Z., Song, S., Ye, C., and Feng, T. (2024). Log-lio2: A lidar-inertial odometry with efficient uncertainty analysis. *arXiv preprint arXiv:2405.01316*.

Hutter, M., Gehring, C., Lauber, A., Gunther, F., Bellicoso, C. D., Tsounis, V., Fankhauser, P., Diethelm, R., Bachmann, S., Blösch, M., et al. (2017). Anymal-toward legged robots for harsh environments. *Advanced Robotics*, 31(17):918–931.

Jelavic, E., Nubert, J., and Hutter, M. (2022). Open3d slam: Point cloud based mapping and localization for education. In *Robotic Perception and Mapping: Emerging Techniques, ICRA 2022 Workshop*, page 24. ETH Zurich, Robotic Systems Lab.

Jian, B. and Vemuri, B. C. (2010). Robust point set registration using gaussian mixture models. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1633–1645.

Jin, Z. and Jiang, C. (2024). Range-aided lidar-inertial multi-vehicle localization and mapping in degenerate environments. *IEEE Transactions on Intelligent Vehicles*.

Jud, D., Kerscher, S., Wermelinger, M., Jelavic, E., Egli, P., Leemann, P., Hottiger, G., and Hutter, M. (2021). Heap-the autonomous walking excavator. *Automation in Construction*, 129:103783.

Kazerouni, I. A., Fitzgerald, L., Dooly, G., and Toal, D. (2022). A survey of state-of-the-art on visual slam. *Expert Systems with Applications*, 205:117734.

Khattak, S., Nguyen, H., Mascarich, F., Dang, T., and Alexis, K. (2020a). Complementary multi–modal sensor fusion for resilient robot pose estimation in subterranean environments. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1024–1029. IEEE.

Khattak, S., Papachristos, C., and Alexis, K. (2020b). Keyframe-based thermal–inertial odometry. *Journal of Field Robotics*, 37(4):552–579.

Kwok, T.-H. (2018). Dnss: Dual-normal-space sampling for 3-d icp registration. *IEEE Transactions on Automation Science and Engineering*, 16(1):241–252.

Kwok, T.-H. and Tang, K. (2016). Improvements to the iterative closest point algorithm for shape registration in manufacturing. *Journal of Manufacturing Science and Engineering*, 138(1).

Li, J., Shi, P., Hu, Q., and Zhang, Y. (2023). Qgore: Quadratic-time guaranteed outlier removal for point cloud registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Lim, H., Kim, D., Kim, B., and Myung, H. (2023). Adalio: Robust adaptive lidar-inertial odometry in degenerate indoor environments. In *2023 20th International Conference on Ubiquitous Robots (UR)*, pages 48–53. IEEE.

Lim, H., Yeon, S., Ryu, S., Lee, Y., Kim, Y., Yun, J., Jung, E., Lee, D., and Myung, H. (2022). A single correspondence is enough: Robust global registration to avoid degeneracy in urban environments. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8010–8017. IEEE.

Liu, Y., Wang, J., and Huang, Y. (2021). A localizability estimation method for mobile robots based on 3d point cloud feature. In *2021 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pages 1035–1041. IEEE.

Low, K.-L. (2004). Linear least-squares optimization for point-to-plane icp surface registration. *Chapel Hill, University of North Carolina*, 4(10):1–3.

Maken, F. A., Ramos, F., and Ott, L. (2020). Estimating motion uncertainty with bayesian icp. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8602–8608. IEEE.

Myronenko, A. and Song, X. (2010). Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275.

Nair, A. D., Kindle, J., Levchev, P., and Scaramuzza, D. (2024). Hilti slam challenge 2023: Benchmarking single+ multi-session slam across sensor constellations in construction. *arXiv preprint arXiv:2404.09765*.

Nashed, S. B., Park, J. J., Webster, R., and Durham, J. W. (2021). Robust rank deficient slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6603–6608. IEEE.

Nissov, M., Khattak, S., Edlund, J. A., Padgett, C., Alexis, K., and Spieler, P. (2024a). Roamer: Robust offroad autonomy using multimodal state estimation with radar velocity integration. In *2024 IEEE Aerospace Conference*. IEEE.

Nissov, M., Khedekar, N., and Alexis, K. (2024b). Degradation resilient lidar-radar-inertial odometry. *arXiv preprint arXiv:2403.05332*.

Nobili, S., Tinchev, G., and Fallon, M. (2018). Predicting alignment risk to prevent localization failure. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1003–1010. IEEE.

Nubert, J., Khattak, S., and Hutter, M. (2021). Self-supervised learning of lidar odometry for robotic applications. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9601–9607.

Nubert, J., Khattak, S., and Hutter, M. (2022a). Graph-based multi-sensor fusion for consistent localization of autonomous construction robots. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10048–10054.

Nubert, J., Walther, E., Khattak, S., and Hutter, M. (2022b). Learning-based localizability estimation for robust lidar localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

Olsson, C. and Eriksson, A. (2008). Solving quadratically constrained geometrical problems using lagrangian duality. In *2008 19th International Conference on Pattern Recognition*, pages 1–5. IEEE.

Petracek, P., Alexis, K., and Saska, M. (2023). Rms: Redundancy-minimizing point cloud sampling for real-time pose estimation in degenerated environments. *arXiv preprint arXiv:2312.07337*.

Petracek, P., Alexis, K., and Saska, M. (2024). Rms: Redundancy-minimizing point cloud sampling for real-time pose estimation. *IEEE Robotics and Automation Letters*.

Pfreundschuh, P., Oleynikova, H., Cadena, C., Siegwart, R., and Andersson, O. (2023). Coin-lio: Complementary intensity-augmented lidar inertial odometry. *arXiv preprint arXiv:2310.01235*.

Pfreundschuh, P., Oleynikova, H., Cadena, C., Siegwart, R., and Andersson, O. (2024). Enwide dataset.

Pomerleau, F. (2013). *Applied registration for robotics: Methodology and tools for ICP-like algorithms*. PhD thesis, ETH Zurich.

Pomerleau, F., Colas, F., Siegwart, R., et al. (2015). A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends® in Robotics*, 4(1):1–104.

Pomerleau, F., Colas, F., Siegwart, R., and Magnenat, S. (2013). Comparing icp variants on real-world data sets. *Autonomous Robots*, 34(3):133–148.

Ramezani, M., Tinchev, G., Iuganov, E., and Fallon, M. (2020). Online lidar-slam for legged robots with robust registration and deep-learned loop closure. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4158–4164. IEEE.

Rusinkiewicz, S. (2019). A symmetric objective function for icp. *ACM Transactions on Graphics (TOG)*, 38(4):1–7.

Schreppel, C. and Brembeck, J. (2020). A qp solver implementation for embedded systems applied to control allocation. *Computation*, 8(4):88.

Segal, A., Haehnel, D., and Thrun, S. (2009). Generalized-icp. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012). A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE.

Tagliabue, A., Tordesillas, J., Cai, X., Santamaria-Navarro, A., How, J. P., Carlone, L., and Aghamohammadi, A.-a. (2020). Lion: Lidar-inertial observability-aware navigator for vision-denied environments. In *International Symposium on Experimental Robotics*, pages 380–390. Springer.

Talbot, W., Nash, J., Paton, M., Ambrose, E., Metz, B., Thakker, R., Etheredge, R., Ono, M., and Ila, V. (2023). Principled icp covariance modelling in perceptually degraded environments for the eels mission concept. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10763–10770. IEEE.

Tuna, T., Nubert, J., Nava, Y., Khattak, S., and Hutter, M. (2023). X-icp: Localizability-aware lidar registration for robust localization in extreme environments. *IEEE Transactions on Robotics*, 40:452–471.

Vizzo, I., Guadagnino, T., Mersch, B., Wiesmann, L., Behley, J., and Stachniss, C. (2022). Kiss-icp: In defense of point-to-point icp–simple, accurate, and robust registration if done the right way. *arXiv preprint arXiv:2209.15397*.

Wall, M. E., Rechtsteiner, A., and Rocha, L. M. (2003). Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer.

Wen, T., Fang, Y., Lu, B., Zhang, X., and Tang, C. (2024). Liver: A tightly coupled lidar-inertial-visual state estimator with high robustness for underground environments. *IEEE Robotics and Automation Letters*.

Westman, E. and Kaess, M. (2019). Degeneracy-aware imaging sonar simultaneous localization and mapping. *IEEE Journal of Oceanic Engineering*, 45(4):1280–1294.

Xu, W., Cai, Y., He, D., Lin, J., and Zhang, F. (2022). Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 38(4):2053–2073.

Yang, H., Antonante, P., Tzoumas, V., and Carlone, L. (2020a). Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. *IEEE Robotics and Automation Letters*.

Yang, H. and Carlone, L. (2019). A quaternion-based certifiably optimal solution to the wahba problem with outliers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1665–1674.

Yang, H., Shi, J., and Carlone, L. (2020b). Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333.

Zhang, J., Kaess, M., and Singh, S. (2016). On degeneracy of optimization-based state estimation problems. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 809–816. IEEE.

Zhang, J. and Singh, S. (2014). Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, pages 1–9. Berkeley, CA.

Zhang, J., Yao, Y., and Deng, B. (2021). Fast and robust iterative closest point. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhen, W. and Scherer, S. (2019). Estimating the localizability in tunnel-like environments using lidar and uwb. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4903–4908. IEEE.

Zhen, W., Zeng, S., and Soberer, S. (2017). Robust localization and localizability estimation with a rotating laser scanner. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6240–6245. IEEE.

Zhou, Q.-Y., Park, J., and Koltun, V. (2016). Fast global registration. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 766–782. Springer.

Zou, J., Shao, L., Tang, H., Chen, H., Bao, H., and Pan, X. (2023). Lmapping: tightly-coupled lidar-inertial odometry and mapping for degraded environments. *Intelligent Service Robotics*, 16(5):583–597.