

GraffMatch: Global Matching of 3D Lines and Planes for Wide Baseline LiDAR Registration

Parker C. Lusk¹, Devarth Parikh², Jonathan P. How¹

Abstract—Using geometric landmarks like lines and planes can increase navigation accuracy and decrease map storage requirements compared to commonly-used LiDAR point cloud maps. However, landmark-based registration for applications like loop closure detection is challenging because a reliable initial guess is not available. Global landmark matching has been investigated in the literature, but these methods typically use ad hoc representations of 3D line and plane landmarks that are not invariant to large viewpoint changes, resulting in incorrect matches and high registration error. To address this issue, we adopt the affine Grassmannian manifold to represent 3D lines and planes and prove that the distance between two landmarks is invariant to rotation and translation if a shift operation is performed before applying the Grassmannian metric. This invariance property enables the use of our graph-based data association framework for identifying landmark matches that can subsequently be used for registration in the least-squares sense. Evaluated on a challenging landmark matching and registration task using publicly-available LiDAR datasets, our approach yields a 1.7x and 3.5x improvement in successful registrations compared to methods that use viewpoint-dependent centroid and “closest point” representations, respectively.

Index Terms—Localization; Mapping; Recognition

I. INTRODUCTION

ESTIMATING the rigid-body transformation between two sensors is a fundamental component of many mobile robotic systems. Wide-baseline registration is particularly challenging since an odometry signal may not be accurate, or even available, to use as an initial guess. This situation arises in core tasks such as loop closure generation, multi-robot map merging, extrinsic calibration, and global (re)localization. In these cases, the relative rotation and translation between sensors can instead be accurately estimated by matching co-visible features and optimizing for the best feature alignment.

In visual settings, appearance-based descriptors are commonly used for image retrieval or place recognition [1]–[3]. However, appearance-based techniques are often limited due to their sensitivity to illumination, weather, and viewpoint changes. Working with 3D sensors can alleviate these issues because of the geometric nature of the data [4], [5], but this requires storing and processing large point clouds, which can hinder online operation. In the context of LiDAR-based navigation, using geometric landmarks such as lines and

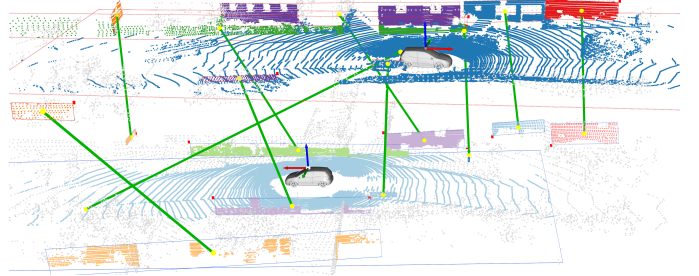


Fig. 1. Successful matching and alignment of line and plane landmarks from two LiDAR scans 180 deg and 14 m apart, without an initial guess. Sensor origins are denoted by the coordinate axes on top of the cars and scans are offset in the z direction for visualization. Poles and planar patches extracted from each LiDAR scan are represented as 3D affine Grassmannian elements. Using the associated Riemannian metric allows for the evaluation of geometric consistency between landmark pairs. Correspondences (shown with green lines connecting landmark centroids) are identified using our GraffMatch algorithm and then used to estimate the rotation and translation between the two sensors, yielding an alignment error of 0.8 deg and 12 cm.

planes has resulted in storage-efficient maps [6]–[8], better scene understanding [9], [10], and low-drift odometry and mapping [11]–[14]. However, existing geometric landmark matching techniques tend to assume 2D motion only [8], [9], use local association strategies given an initial guess from odometry [14], [15], or use a series of heuristic checks [16]–[18] that lead to low matching success rate. SegMatch [19] uses point clusters as landmarks, but the repeatability of cluster segmentation may strongly depend on viewing angle, especially in the presence of partial occlusion or object motion.

Recent successes in graph-based data association [20], [21] have significantly increased the robustness of the correspondence selection process in spatial perception problems by leveraging the notion of pairwise consistency. By matching constellations of objects that have consistent pairwise distances across two views, good correspondences can be identified even in the presence of many bad ones. Global data association techniques do not require an initial registration guess, but rely on a correctly defined pairwise distance that is invariant to transformation; for example, the Euclidean distance between two rigidly-attached points does not change even as those points are rotated and translated. Existing works frequently use Euclidean distance between line or plane pairs, using representations such as the centroid [8], which is not well-defined for infinite lines and planes, or the “closest point” (CP) parameterization [13], which lacks the necessary translation invariance because of its dependence on sensor origin.

Instead, we represent line and plane landmarks naturally as elements of a Grassmannian manifold, which is the space of all linear subspaces. In particular, we utilize the *affine* Grassmannian manifold, which allows for the representation of affine subspaces (i.e., linear subspaces not necessarily contain-

Manuscript received: August, 16, 2022; Revised November, 5, 2022; Accepted November, 28, 2022.

This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers’ comments. This work was supported by the Ford Motor Company.

¹Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. {plusk, jhow}@mit.edu

²Ford Motor Company. dparikh9@ford.com

Digital Object Identifier (DOI): see top of this page.

ing the origin). Thus, invariant distances between geometric primitives can easily be defined in a principled manner using the Grassmannian metric, enabling the use of our robust, global data association framework [20]. Then, the rigid transformation between a pair of candidate loop closure scans can be estimated by solving a line and plane registration problem with known correspondences in the least-squares sense.

This letter presents an evolution of our previous work [22] by providing further validation on additional datasets, an in-depth experimental analysis, and new applications of our method in automatic extrinsic sensor calibration. In summary, our main contributions are:

- the introduction of the affine Grassmannian for global data association of lines and planes, leading to 3D registration without requiring an initial guess;
- a least squares estimator for rigid transformation using line and plane representations directly, leading to accurate rotation and translation estimation;
- experimental evaluation in the context of loop closure, using challenging scan pairs of LiDAR datasets, showing superior recall and accuracy over the state-of-the-art.

We emphasize that this is the first work using the affine Grassmannian manifold for data association, which provides a unifying and principled framework for associating points, lines, planes (or higher dimensional linear objects) in spatial and geometric perception problems encountered in robotics.

II. RELATED WORK

Geometric registration algorithms are either local or global, depending on if an initial guess is required. Local methods are often used in scan matching (e.g., ICP [23]), where consecutive scans typically have small displacement between them. In contrast, global methods do not need an initial guess to succeed and are often preferred in settings like loop closure detection because no good initial guess exists. In point-based registration, global methods first generate candidate point correspondences, typically based on local descriptors [5], [24], [25]. The set of putative correspondences are likely to contain incorrect matches, called outliers, and so robust iterative estimation techniques like RANSAC [26] or graduated non-convexity [27], [28] can be used to select a subset of correspondences that best support the model. Recently, non-iterative robust approaches based on graph theory [20], [21] have been introduced that significantly improve the performance of estimation in the presence of outliers. In this work, we apply our CLIPPER algorithm [20] to the domain of global matching of 3D lines and planes.

Poles/lines and planes commonly exist in man-made environments and have recently been used as landmarks in LiDAR navigation. The benefits of using higher-order geometric primitives include lower storage and processing requirements since there are fewer pole and plane objects than points [6]–[8], and more accurate odometry because it is infeasible to get exact point-to-point correspondences from sparse LiDAR point clouds [15]. Local methods for landmark-based registration rely on identifying the closest landmarks between two scans, given an initial alignment guess. Nearest neighbor search requires calculating distances between landmarks, which is

dependent on the landmark representation. The most common representations include vector form for lines, Hesse normal form for planes, and the CP vector for lines/planes [29]. The CP representation compactly encodes position and vector orientation in a 3-vector, and the Euclidean distance is often used to find similar landmarks [13], [15]. For lines in vector form or planes in Hesse normal form, both the angle between vectors and the distance between points is used [12]. If points are retained in one of the scans, then the point-to-landmark RMSE can also be used [14]. Other methods [6], [7], [30] project the centroids of the detected landmark onto the ground plane, creating 2D points. However, these methods assume the landmark (infinite line or plane) has a well-defined centroid, the ground plane is known, and that 2D registration is sufficient.

Since descriptors for 3D lines and planes have not been thoroughly explored, most global methods rely on a series of geometric tests to assign correspondences [16]. LiPMatch [18] adopts an interpretation tree for plane matching [17] in loop closure detection, using unary and binary constraints between candidate plane matches to determine the largest set of consistent matches. However, some of these constraints are sensitive to viewpoint change (e.g., centroid, area). ClusterMatch [8], [9] matches poles/lines and planes by iteratively searching for landmark pairs with similar pairwise centroid distance until a large number of matches supports the resulting transformation. A critical drawback of these methods is that, although global methods, the heuristic geometric tests that are used are often heavily view dependent. In contrast, our method performs global registration by searching for a set of correspondences that have consistent intrascan pairwise distances, using a distance definition based on their natural representation as elements of Grassmannian manifolds.

The Grassmannian has been used extensively in subspace learning [31], especially in face recognition [32] and appearance tracking [33] tasks in computer vision. Calinon [34] outlines the use of Riemannian manifolds in robotics and notes the under-representation of the Grassmannian manifold.

A. Preliminaries

We briefly introduce the Grassmannian manifold, but a more comprehensive introduction is provided in [35]. The Grassmannian is the space of k -dimensional subspaces of \mathbb{R}^n , denoted $\text{Gr}(k, n)$. For example, $\text{Gr}(1, 3)$ represent 3D lines containing the origin. An element $\mathbb{A} \in \text{Gr}(k, n)$ is represented by an orthonormal matrix $A \in \mathbb{R}^{n \times k}$ whose columns form an orthonormal basis of \mathbb{A} . Note that the choice of A is not unique. The geodesic distance between two subspaces $\mathbb{A}_1 \in \text{Gr}(k_1, n)$ and $\mathbb{A}_2 \in \text{Gr}(k_2, n)$ is

$$d_{\text{Gr}}(\mathbb{A}_1, \mathbb{A}_2) = \left(\sum_{i=1}^{\min(k_1, k_2)} \theta_i^2 \right)^{1/2} \quad (1)$$

where θ_i are known as the principal angles [36]. These angles can be computed via the singular value decomposition (SVD) of the corresponding orthonormal matrices of \mathbb{A}_1 and \mathbb{A}_2 ,

$$A_1^\top A_2 = U \text{diag}(\cos \theta_1, \dots, \cos \theta_{\min(k_1, k_2)}) V^\top. \quad (2)$$

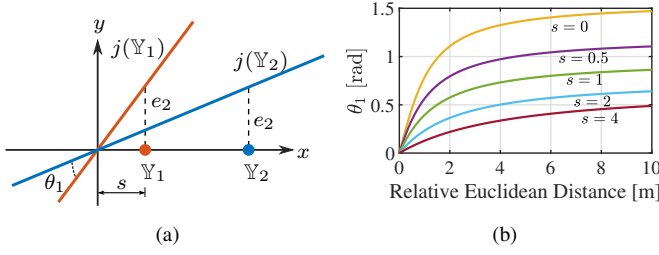


Fig. 2. (a) Example of points in $\text{Graff}(0, 1)$ being embedded as lines in $\text{Gr}(1, 2)$. The principal angle between these two linear subspaces is θ_1 . (b) When applied directly, d_{Graff} is not invariant to global translation s .

We are specifically interested in affine subspaces of \mathbb{R}^3 , e.g., lines and planes that may be at some distance away from the origin. In analogy to $\text{Gr}(k, n)$, the set of k -dimensional affine subspaces constitute a smooth manifold called the *affine Grassmannian* and denoted $\text{Graff}(k, n)$ [37]. We write an element of this manifold as $\mathbb{Y} = \mathbb{A} + b \in \text{Graff}(k, n)$ with affine coordinates $[A, b] \in \mathbb{R}^{n \times (k+1)}$, where $A \in \mathbb{R}^{n \times k}$ is an orthonormal matrix and $b \in \mathbb{R}^n$ is the displacement of \mathbb{A} from the origin. We emphasize that $\text{Graff}(k, n) \neq \text{Gr}(k, n) \times \mathbb{R}^n$. Instead, an element $\mathbb{Y} \in \text{Graff}(k, n)$ is treated as a higher-order subspace via the embedding

$$j : \text{Graff}(k, n) \hookrightarrow \text{Gr}(k+1, n+1), \quad \mathbb{A} + b \mapsto \text{span}(\mathbb{A} \cup \{b + e_{n+1}\}), \quad (3)$$

where $e_{n+1} = (0, \dots, 0, 1)^\top \in \mathbb{R}^{n+1}$ (see [37, Theorem 1]). Fig. 2a shows an example of two points \mathbb{Y}_1 and \mathbb{Y}_2 in \mathbb{R} being embedded as different lines $j(\mathbb{Y}_1)$ and $j(\mathbb{Y}_2)$ in \mathbb{R}^2 .

The Stiefel coordinates of $\mathbb{Y} \in \text{Graff}(k, n)$,

$$Y = \begin{bmatrix} A & b_0/\sqrt{1+\|b_0\|^2} \\ 0 & 1/\sqrt{1+\|b_0\|^2} \end{bmatrix} \in \mathbb{R}^{(n+1) \times (k+1)}, \quad (4)$$

allow for the computation of distances between two affine subspaces using the Grassmannian metric,

$$d_{\text{Graff}}(\mathbb{Y}_1, \mathbb{Y}_2) = d_{\text{Gr}}(j(\mathbb{Y}_1), j(\mathbb{Y}_2)), \quad (5)$$

with principal angles computed via the SVD of $Y_1^\top Y_2$. The vector $b_0 \in \mathbb{R}^n$ is the orthogonal displacement of \mathbb{A} , which is the projection of b onto the left nullspace of A s.t. $A^\top b_0 = 0$.

For convenience, the line $\mathbb{Y}^\ell \in \text{Graff}(1, 3)$ may also be represented in vector form as $\ell = [A; b] \in \mathbb{R}^6$, and a plane $\mathbb{Y}^\pi \in \text{Graff}(2, 3)$ may be represented in Hesse normal form as $\pi = [n; d] \in \mathbb{R}^4$ where $n = \ker A^\top$ and $d = \|b_0\|$. Under a rigid transformation $T = (R, t) \in \text{SE}(3)$, the transformation law of lines and planes can be written

$$\ell' = f_\ell(\ell, R, t) := [RA \quad Rb + t]^\top \quad (6)$$

$$\pi' = f_\pi(\pi, R, t) := T^{-\top} \pi. \quad (7)$$

III. METHOD

Given a set $\mathcal{S}_i = \{\mathbb{Y}_1^\ell, \dots, \mathbb{Y}_{l_i}^\ell, \mathbb{Y}_1^\pi, \dots, \mathbb{Y}_{p_i}^\pi\}$ with l_i lines and p_i planes, we refer to the a -th landmark as $s_{i,a} \in \mathcal{S}_i$. Our method is comprised of the following steps: (i) constructing a consistency graph based on pairwise landmark distances, (ii) identifying landmark correspondences via the densest complete subgraph in the consistency graph, and (iii) estimating a rigid transformation based on the identified correspondences.

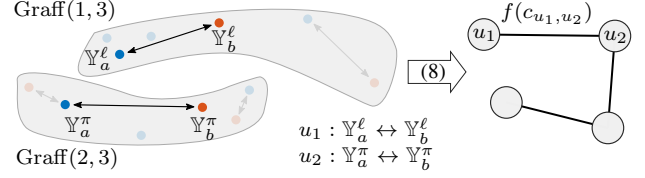


Fig. 3. Construction of a consistency graph. Using d_{Graff} , the distance between a line and a plane in set \mathcal{S}_i (•) is compared to the distance between the two corresponding landmarks in set \mathcal{S}_j (•). The consistency of these two distances is evaluated using (8) and the edge (u_1, u_2) is so weighted.

A. Consistency Graph Construction

A consistency graph for two sets $\mathcal{S}_i, \mathcal{S}_j$ is an undirected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$. Each vertex $u \in \mathcal{V}$ represents a potential correspondence between landmarks of the same type, denoted $s_{i,a} \leftrightarrow s_{j,b}$. An edge $(u_p, u_q) \in \mathcal{E}$ between vertices $u_p, u_q \in \mathcal{V}$ exists if the correspondence pair is *consistent*; the edge-weighting function $w : \mathcal{E} \rightarrow [0, 1]$ indicates the level of consistency. The correspondence pair u_p, u_q is consistent if the distance between the underlying landmarks satisfies

$$c_{u_p, u_q} \stackrel{\text{def}}{=} |d(s_{i, u_p^i}, s_{i, u_q^i}) - d(s_{j, u_p^j}, s_{j, u_q^j})| < \epsilon, \quad (8)$$

where d is some distance function and, by some abuse of notation, u_p^i denotes the a -th landmark of \mathcal{S}_i involved in the correspondence u_p . Note that the two distances in (8) are between landmarks *internal* to sets \mathcal{S}_i and \mathcal{S}_j , respectively. If a pair of correspondences are deemed consistent, the corresponding edge is attributed the weight $w(u_p, u_q) \stackrel{\text{def}}{=} f(c_{u_p, u_q})$, for some choice of $f : \mathbb{R}_+ \rightarrow [0, 1]$ that scores very consistent pairs close to 1. In this paper, we choose $f(c) \stackrel{\text{def}}{=} \exp(-c^2/2\sigma^2)$ for simplicity, although other appropriate functions could be used. Given a consistency graph, correspondences are selected that maximize consistency, further explained in Section III-B.

The distance function d must be carefully chosen to ensure accuracy of graph-based data association. In particular, we desire (8) to hold when $s_{j, u_1^b}, s_{j, u_2^b}$ are the transformed versions of $s_{i, u_1^a}, s_{i, u_2^a}$, respectively. This invariance property leads to subgraphs of the consistency graph that indicate a set of landmark matches.

Definition 1. A distance $d : X \times X \rightarrow \mathbb{R}$ is *invariant* if $d(x_1, x_2) = d(x'_1, x'_2)$, where $x'_1, x'_2 \in X$ are the transformation of $x_1, x_2 \in X$ under $T \in \text{SE}(3)$, respectively.

We establish the invariance of the metric d_{Graff} to rotation and, under careful application, translation.

Proposition 1. For elements $\mathbb{Y}_1 \in \text{Graff}(k_1, 3)$, $\mathbb{Y}_2 \in \text{Graff}(k_2, 3)$ with affine coordinates $[A_1, b_1]$ and $[A_2, b_2]$, the affine Grassmannian metric d_{Graff} is invariant if the affine components are first shifted to the origin, i.e., if both b_1 and b_2 are first translated by $-b_1$.

Proof. See Appendix A. \square

The intuition of Proposition 1 can be understood from Fig. 2. As \mathbb{Y}_1 and \mathbb{Y}_2 are together translated further from the origin, the principal angle between $j(\mathbb{Y}_1)$ and $j(\mathbb{Y}_2)$ decreases to zero in the limit. However, the distance between the affine components of \mathbb{Y}_1 and \mathbb{Y}_2 remains the same, no matter the translation. By first shifting the affine components, we remove the dependence of the absolute translation in the computation

of the principal angle, while maintaining the dependence on the *relative* translation between \mathbb{Y}_1 and \mathbb{Y}_2 .

A remaining challenge is to address the insensitivity of d_{Graff} to the Euclidean distance between landmarks' affine components. The yellow curve ($s = 0$) in Fig. 2b represents the principal angle between $\mathbb{Y}_1, \mathbb{Y}_2 \in \text{Graff}(0, 1)$ after shifting them as per Proposition 1, as a function of the relative translation between \mathbb{Y}_1 and \mathbb{Y}_2 . Observe that after a distance of approximately 2 m, the curve quickly asymptotes towards $\frac{\pi}{2}$. This nonlinearity leads to poor discrimination between pairs of correspondences whose internal landmarks are far apart in the Euclidean sense. To combat this when calculating pairwise affine Grassmannian distances, we first scale the affine component of each \mathbb{Y}_i by a constant parameter ρ so that the affine coordinates of \mathbb{Y}_i become $[A_i, b_i/\rho]$. The choice of ρ depends on the average Euclidean distance between landmarks in the environment and its effect is to bring principal angles into the linear regime. The selection of ρ is discussed further in Section IV-B.

With Proposition 1 and the scaling parameter ρ in hand, a consistency graph between landmarks in \mathcal{S}_i and \mathcal{S}_j can be constructed. We establish initial correspondences between each landmark in \mathcal{S}_i with each landmark of \mathcal{S}_j so long as the landmarks are of the same dimensions k (i.e., we do not allow lines to be associated to planes). Given additional information such as color, scan intensity, planar patch area, or pole radius, this initial set of correspondences could be refined, but would rely on accurately segmenting lines and planes across potentially wide baselines. While we restrict landmark correspondences to be of the same dimension, the machinery we have developed allows for computing the consistency of two correspondences whose internal pair of objects have differing dimension, thereby aiding in subgraph selection. Fig. 3 illustrates evaluating the consistency of a correspondence pair using the affine Grassmannian.

B. Graph-based Global Data Association

Given a consistency graph, the task of matching objects from two scans is reduced to identifying the densest clique of consistent correspondences, formalized as the problem

$$\begin{aligned} & \underset{u \in \{0,1\}^m}{\text{maximize}} && \frac{u^\top M u}{u^\top u} \\ & \text{subject to} && u_i u_j = 0 \quad \text{if } M(i, j) = 0, \quad \forall i, j, \end{aligned} \quad (9)$$

where $M \in [0, 1]^{m \times m}$ is the weighted adjacency matrix (i.e., from w as defined in Section III-A) with ones on the diagonal, and $u \in \{0, 1\}^m$ indicates a consistent set of correspondences. Note that we choose to maximize the *density* of correspondences rather than the cardinality (i.e., maximum clique) as our previous work has found this objective to produce more accurate results [20]. Problem (9) is NP-hard, therefore we solve a particular relaxation which yields high accuracy solutions via our efficient CLIPPER algorithm (see [20] for more details).

C. Transformation Estimation

Given pairwise correspondences between landmarks in \mathcal{S}_i and \mathcal{S}_j , consider finding the best rigid transformation to

simultaneously align matched lines and planes by solving the optimization problem

$$\min_{\substack{R \in \text{SO}(3), \\ t \in \mathbb{R}^3}} \sum_{i=1}^p \|\pi'_i - f_\pi(\pi_i, R, t)\|^2 + \sum_{i=1}^l \|\ell'_i - f_\ell(\ell_i, R, t)\|^2. \quad (10)$$

This problem can be solved in closed-form by first solving for the rotation via SVD, then solving for the translation via least squares, similar to Arun's method for point cloud registration [38]. Note that in the case of only parallel planes or only collinear lines, (10) will be degenerate (i.e., the correlation matrix used in SVD will be rank deficient). These cases can be numerically detected via the condition number κ of the correlation matrix. The benefit of using the line and plane geometry directly, as opposed to a point parameterization, is twofold. First, it allows the use of the full information present in the infinite plane or line, i.e., distance from origin as well as orientation. Second, it does not require assumptions about where the "centroid" of the plane or line is, which is undefined for infinite planes and lines and requires consistent segmentation of landmarks from point clouds. Together, these benefits lead to a more accurate rigid transformation estimate when aligning line and plane landmarks.

IV. EXPERIMENTS

We evaluate our method, called GraffMatch, using LiDAR scans from three datasets: KITTI [39] sequences 00, 02, 05, 08; KITTI-360 [40] sequences 00, 04, 06, 09; NCLT [41] sessions 2012-04-29, 2012-05-11, 2012-12-01. We include comparisons with CPMatch [15], BruteForceRMSE [14], ClusterMatch [8], and LiPMatch [18], all of which are used for geometric landmark registration in state-of-the-art pipelines. CPMatch uses nearest neighbor search on CP vectors and BruteForceRMSE exhaustively calculates the point-to-line/plane RMSE between each potential landmark match, followed by the selection of correspondences with low RMSE. ClusterMatch creates putative correspondences using pairwise landmark centroid distances followed by inlier validation, and LiPMatch uses an interpretation tree to find plane correspondences with similar geometric properties. We also include comparisons using our CLIPPER algorithm, using Euclidean distance of centroids and CP vectors to score consistency, both of which lack the required invariance for lines and planes. The algorithms are implemented using Python and C++ and executed on an i9-7920X CPU with 64 GB RAM. The parameters used for GraffMatch (see (8)) are $\epsilon = 0.2$ and $\sigma = 0.05$. In our comparisons, we show that GraffMatch successfully matches more landmarks, leading to more successful registrations.

A. Dataset Preparation

Motivated by place recognition and loop closure detection applications, we sample poses along the ground-truth trajectory of each dataset sequence with a stride of 2 m to create *places* (i.e., a pose with the LiDAR scan at that pose), following [4]. For each place p , true *revisits* are found by identifying any previously visited place p' such that the path length between p and p' is greater than 50 m and the Euclidean

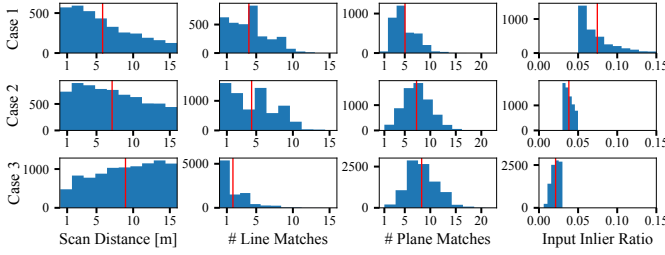


Fig. 4. Statistics of the 20k loop closures in the test dataset, divided into three cases according to each loop closure’s input inlier ratio. The distance between sensor scans and input inlier ratio are correlated, with the most difficult case (Case 3) having longer sensor baselines. On average, there are more correct plane landmark matches than correct line landmark matches.

distance between the places is less than 16 m. Ground-truth landmark matches between revisited places are generated by aligning the landmarks using ground truth and solving a linear-sum assignment problem based on d_{Graff} distances. Valid place pairs are selected as *loop closures* from revisits having more than 4 true landmark matches with registration conditioning $\kappa < 1e3$ and having registration error less than 5 deg and 1 m using the true landmark matches. This setup provides a scene matching dataset similar to 3DMatch [5], but having line and plane landmarks instead of point features.

Lines are detected in point clouds by first selecting pole-like points found by clustering the LiDAR range image, as in [42]. For KITTI, pole-like points are selected using semantic information from SemanticKITTI [43]. These points are clustered using DBSCAN [44] implemented in Open3D [45] and PCA is used to estimate lines from pole-like clusters. Planar patches are extracted from the LiDAR scan using our own implementation¹ of [46]. Because planar patches are bounded, there may be multiple planar patches that correspond to the same infinite plane, so we merge planes that are similar. The statistics of correct line and plane landmark matches in the dataset are shown in Fig. 4, separated into three cases according to input inlier ratio (IIR), which is defined as the number of correct matches out of the number of possible matches and indicates the difficulty of each place pair. Cases 1, 2, and 3 consist of place pairs with IIRs of 0.05+, 0.03 to 0.05, and 0 to 0.03, respectively. These IIR ranges are chosen to expose the relationship between IIR and sensor baseline distance and to ensure enough place pairs exist in each case.

B. Selection of Scaling Parameter

The scaling parameter ρ (see Section III-A) is chosen so that the pairwise affine Grassmannian distance lies in the linear regime and is therefore more sensitive when scoring consistencies. The Velodyne HDL-64E used in KITTI has a range up to 120 m, with an average point range in the KITTI dataset of approximately 80 m. Additionally, the average Euclidean distance between landmark centroids in KITTI is 26 ± 16 m, as shown in Fig. 5. Therefore, we select $\rho = 40$ so that relative Euclidean distances of 80 m will be scaled to 2 m, which is at the end of the linear regime (see Fig. 2b). However, GraffMatch is not extremely sensitive to this choice and yields similar matching results for a range of ρ values while holding other parameters constant, as shown in Fig. 5.

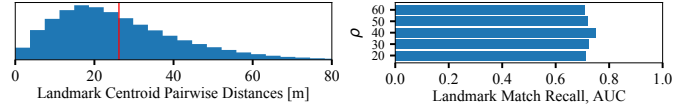


Fig. 5. (left) Pairwise distances of landmark centroids in KITTI. The mean is 26 ± 16 m. Using this data, we choose the scaling parameter as $\rho = 40$. (right) GraffMatch is not extremely sensitive to this choice of scaling. Other values of ρ yield similar matching results in KITTI, indicated by the AUC.

C. Evaluation Metrics

We evaluate both landmark matching ability and registration quality. Correctness of landmark matching is evaluated via landmark-match recall (LMR) [25], which measures the percentage of place pairs that can be registered with high confidence given landmark matches. LMR is defined as

$$\text{LMR} = \frac{1}{N} \sum_{s=1}^N \mathbb{1} \left(\left[\frac{1}{|\mathcal{C}_s|} \sum_{(i,j) \in \mathcal{C}_s} \mathbb{1}(d_{ij} < \tau_d) \right] > \tau_{\text{OIR}} \right), \quad (11)$$

where N is the number of place pairs, \mathcal{C}_s is a set of landmark correspondences between place pair s , and d_{ij} is the d_{Graff} distance between landmark matches after registering landmarks using ground truth. The inlier distance threshold τ_d controls how close two landmarks must be (after registering the landmarks using the ground truth) to be considered an inlier correspondence. The output inlier ratio (OIR) measures the precision of an algorithm’s correspondence set \mathcal{C}_s and the threshold τ_{OIR} is used to examine the percentage of place pairs an algorithm can recover with at least the specified OIR. Point-based registration works have evaluated performance with an OIR threshold as low as 0.05 [25], arguing that RANSAC can be effective even with only a 5% inlier ratio, even though many iterations would be required. In contrast, we set $\tau_{\text{OIR}} = 0.8 \gg 0.05$ to emphasize the robustness of GraffMatch and we do not use RANSAC for inlier refinement. The area under the LMR curve (AUC) generated by sweeping $\tau_{\text{OIR}} \in [0, 1]$ and holding $\tau_d = 6$ deg is used as a summary measure, with an AUC of 1 being ideal.

Registration quality is evaluated via rotation, translation error and registration recall. Error is calculated with respect to the ground truth transformation $(\mathbf{R}^*, \mathbf{t}^*)$ as $\arccos((\text{Tr}(\hat{\mathbf{R}}^T \mathbf{R}^*) - 1)/2)$ and $\|\hat{\mathbf{t}} - \mathbf{t}^*\|$. Registration recall is the percentage of successfully registered place pairs (i.e., loop closures). A successful registration has an estimation error within 5 deg and 1 m to the ground truth transformation.

D. Landmark Matching and Registration Results

Data association is attempted on each place pair, after which the landmark matches are used to estimate the relative rotation and translation. Fig. 6 plots algorithms’ LMR curves for the NCLT dataset. GraffMatch is able to match approximately 80% of place pairs with an OIR of 0.8 or more, while other methods are only able to match less than 40% of place pairs for the same inlier regime. For correspondence sets having such a high inlier ratio, additional methods like RANSAC could be used to further refine the correspondence set if desired. However, we show that even without this extra step, GraffMatch is able to successfully select the most correct correspondences. Fig. 6 also includes line-only and plane-only variants of GraffMatch to highlight the value of creating a

¹<https://github.com/plusk01/pointcloud-plane-segmentation>

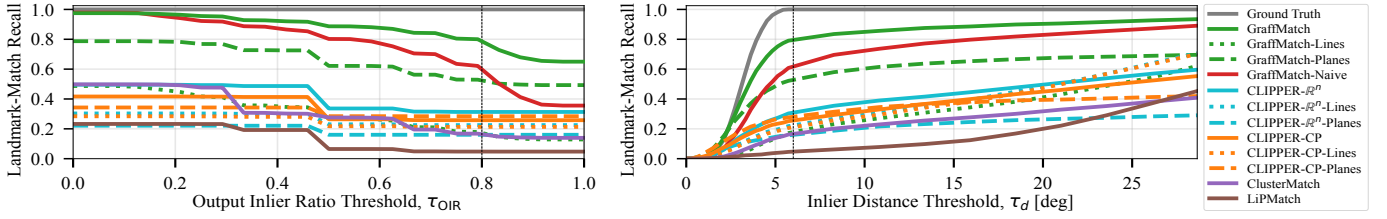


Fig. 6. (left) Landmark-match recall (LMR) varied over τ_{OIR} , indicating for each algorithm the percentage of place pairs having selected correspondences with at least the given OIR. (right) The maximum distance a pair of ground-truth registered landmarks can be while considered an inlier match.

TABLE I. Results for each dataset, divided into three cases based on the feature inlier ratio. We compare each algorithm on recall rate and landmark-match recall AUC and the highest for each dataset case is bolded. The average rotation and translation errors of the successful registrations are also listed.

	N	CPMatch [15]				BruteForceRMSE [14]				ClusterMatch [8]				LiPMatch [18]				CLIPPER- \mathbb{R}^n				CLIPPER-CP				GraffMatch			
		Recall [%]	LMR AUC	R_e [deg]	t_e [cm]	Recall [%]	LMR AUC	R_e [deg]	t_e [cm]	Recall [%]	LMR AUC	R_e [deg]	t_e [cm]	Recall [%]	LMR AUC	R_e [deg]	t_e [cm]	Recall [%]	LMR AUC	R_e [deg]	t_e [cm]	Recall [%]	LMR AUC	R_e [deg]	t_e [cm]	Recall [%]	LMR AUC	R_e [deg]	t_e [cm]
KITTI	Case 1 1392	24	0.47	1.1	21	9	0.33	1.2	62	35	0.34	1.7	18	16	0.30	1.1	19	62	0.70	1.5	24	42	0.56	1.3	21	81	0.91	1.1	20
	Case 2 4560	19	0.53	1.2	23	3	0.34	1.3	69	16	0.18	1.5	17	15	0.32	1.2	22	46	0.55	1.5	27	22	0.38	1.4	25	52	0.78	1.1	19
	Case 3 2329	7	0.42	1.5	29	<1	0.23	2.1	66	6	0.10	1.5	21	6	0.23	1.7	30	20	0.30	1.7	31	7	0.17	1.7	33	21	0.59	1.2	24
KITTI-360	Case 1 182	17	0.51	1.6	29	3	0.34	1.5	28	21	0.28	1.4	26	8	0.23	1.8	35	16	0.32	1.5	26	10	0.35	1.7	32	53	0.85	1.6	32
	Case 2 1555	10	0.43	1.6	27	1	0.26	2.4	20	8	0.14	1.3	25	5	0.19	1.7	32	12	0.26	2.0	31	6	0.28	1.7	30	41	0.74	1.8	36
	Case 3 7143	4	0.27	1.8	28	<1	0.14	2.6	20	2	0.06	1.7	31	2	0.11	2.0	36	6	0.14	2.2	36	1	0.14	2.1	34	21	0.55	1.9	42
NCLT	Case 1 1854	6	0.33	2.0	30	1	0.23	2.6	81	21	0.36	2.9	30	4	0.16	2.5	41	25	0.42	2.7	35	14	0.39	2.7	35	60	0.91	2.7	35
	Case 2 676	5	0.38	2.0	25	1	0.20	2.8	89	7	0.17	2.9	34	1	0.09	1.9	44	20	0.36	2.9	39	9	0.32	2.8	36	43	0.78	2.8	38
	Case 3 150	4	0.32	2.5	29	<1	0.14	2.6	76	2	0.15	3.2	19	0	0.07	-	-	12	0.31	3.2	43	2	0.23	2.4	42	16	0.62	2.9	36

consistency graph utilizing both landmark types. In the NCLT dataset, there are fewer detected poles than planes, leading to a gap between the line-only and plane-only variants; however, using both landmarks results in a greater number of place pairs having a higher OIR. Additionally, a version denoted GraffMatch-Naive is included that does not first shift landmark pairs (as in Proposition 1), and so does not use a distance that is transformation invariant. The gap between GraffMatch and GraffMatch-Naive highlights the importance of using an invariant distance function when scoring landmark association consistency. In the right of Fig. 6, we see how the LMR curve changes with the inlier distance threshold τ_d . By using d_{Graff} for these geometric landmarks, GraffMatch is able to achieve an LMR curve most similar to the ground truth. Due to space limitations, the LMR curves for each dataset are not shown, but the AUC for each dataset and case is listed in Table I. In all datasets and cases, GraffMatch successfully matches more landmarks and has higher registration recall than other methods. As expected, the local methods CPMatch and BruteForceRMSE perform worse than global methods because of wide sensor baselines present in the datasets. However, with the exception of GraffMatch, the global methods leverage landmark properties that are view dependent. ClusterMatch and CLIPPER- \mathbb{R}^n use landmark centroids, which are not well-defined for infinite geometries and may shift depending on the detection. Likewise, LiPMatch uses centroids as well as properties such as the area and the extent of planar patches. CLIPPER-CP uses CP vectors, which are not invariant to translation and cannot be shifted as in Proposition 1 because the CP vector is undefined for landmarks at the origin. These results underscore the importance of using view-independent geometric representations for lines and planes, e.g., the affine Grassmannian manifold used in the GraffMatch framework.

Fig. 7 shows the alignment error of all attempted place pair registrations from KITTI as a grid of density heatmaps, where columns correspond to algorithms and rows (from

top to bottom) correspond to Cases 1, 2, and 3. The first column shows the alignment error using ground truth landmark matches with respect to ground truth alignment, indicating the best achievable landmark-based registration without a point-based refinement step (e.g., using ICP). GraffMatch is the only data association method that consistently scores in the low-translation, low-rotation error regime.

Runtime statistics are shown in Fig. 8. Compared to CLIPPER- \mathbb{R}^n , GraffMatch incurs an additional 50 ms on average due to the calculation of d_{Graff} , but is still capable of real-time at 10 Hz LiDAR rate. GraffMatch runtime could be reduced by encoding prior knowledge in the putative associations, e.g., ground planes should be matched, or large planar patches are not likely to be matched to small patches.

Although storing and processing raw points can be resource intensive, directly operating on point clouds is a typical means of registration and loop closure constraint generation. Thus, we compare GraffMatch with existing point-based methods, with results in Table II. As a baseline, we use FPFH [24] local descriptors (downsampled to 0.35 m) with RANSAC (max 100,000 iters.) implemented in Open3D. Scan context [4], which is nominally used for place recognition, is also included. Different from other point cloud place recognition methods, it provides an initial guess for yaw that can be used to initialize an ICP-based registration step. For a fair comparison, Table II also reports GraffMatch solutions refined by ICP, but only using points of matched landmarks, thereby reducing storage and processing requirements. Because scan context is only rotation invariant, it achieves relatively low recall since ground truth place pairs can have up to 16 m of translation. Performing ICP increases the success rate, but only marginally when compared to GraffMatch without ICP refinement (see Table I). These results on this dataset indicate the translation sensitivity of scan context, which limits it to small-baseline settings. Similarly performing ICP on GraffMatch solutions leads to high success and accuracy, indicating that the landmark matches

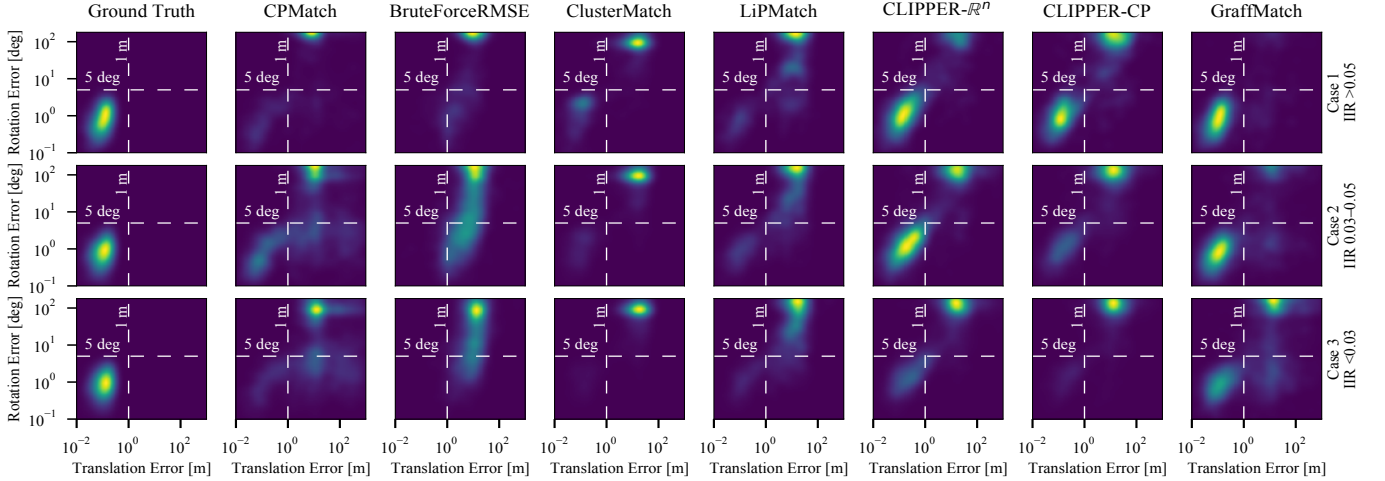


Fig. 7. Alignment error for place pairs, visualized as likelihood-normalized density plots. Columns correspond to algorithms. The first column shows the best alignment error achievable when registering landmarks using the true landmark matches. From top to bottom, each row corresponds to Cases 1, 2, and 3, with Case 1 being easiest and Case 3 being hardest. In each case, GraffMatch produces the greatest number of registrations with low alignment error.

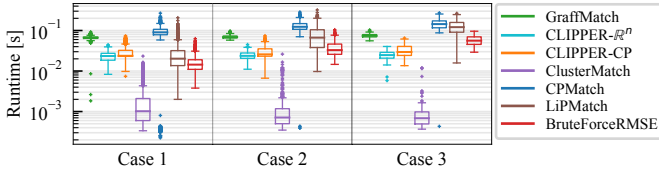


Fig. 8. Runtime statistics. GraffMatch is capable of running in real-time at the typical 10 Hz LiDAR rate. The runtime of GraffMatch could be reduced using landmark descriptors to decrease the number of putative associations.

TABLE II. Point-based registration methods are computationally demanding and do not necessarily lead to higher accuracy or success in wide-baseline settings. Refining GraffMatch with ICP (only using points underlying landmarks) leverages the positive aspects of landmark and point-based registration.

	GraffMatch+ICP				FPFH+RANSAC				ScanContext [4]				ScanContext+ICP			
	Recall [%]	R_e [deg]	t_e [cm]	t [ms]	Recall [%]	R_e [deg]	t_e [cm]	t [ms]	Recall [%]	R_e [deg]	t_e [cm]	t [ms]	Recall [%]	R_e [deg]	t_e [cm]	t [ms]
Case 1	93	0.5	13	119	16	1.7	46	1029	8	0.9	54	25	29	0.5	21	142
Case 2	76	0.4	7	125	12	1.8	44	807	4	0.9	59	25	31	0.5	25	138
Case 3	49	0.3	8	156	5	2.2	51	660	<1	1.0	52	37	14	0.6	32	140

are of high quality (i.e., high LMR AUC in Table I), but there may be cases where the landmark-only least squares registration was ill-conditioned. The FPFH baseline performs poorly due to non-uniform LiDAR sampling and low point cloud overlap, and its high computational cost highlights the potential difficulties of processing point clouds directly.

E. Automatic LiDAR-LiDAR and Camera-Depth Calibration

High-quality extrinsic calibration is a crucial prerequisite for many autonomous systems. Calibration methods typically require data correspondence across sensors and state-of-the-art multimodal calibration often achieve this by requiring fiducial markers on calibration targets [47]. Instead, unlabeled geometric landmarks extracted from each modality can be matched and then registered using GraffMatch. Fig. 9 shows two instances of extrinsic calibration using GraffMatch to globally match planes. In Fig 9a, many chess boards are held in the field-of-view of the LiDARs and planes are extracted from the two point clouds (red and blue). GraffMatch correctly matches 10 planes in 75 ms with calibration error of 1.1 deg and 8 cm. In Fig 9b, an example of multi-modal calibration

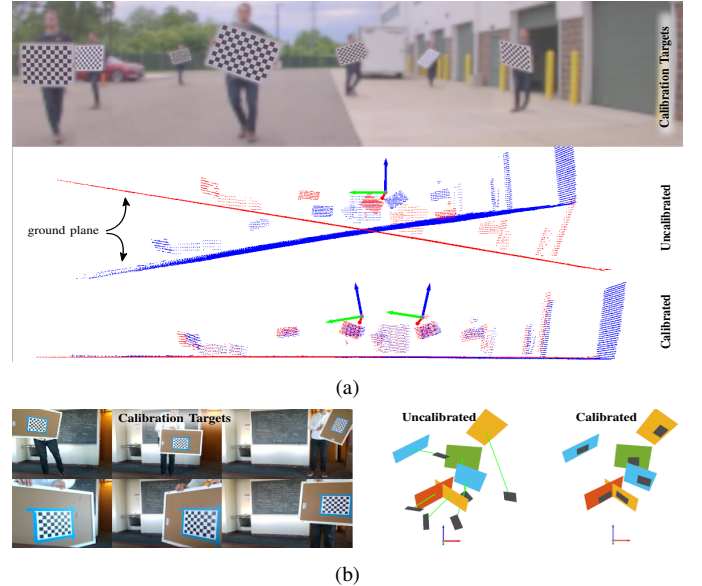


Fig. 9. Automatic extrinsic calibration using GraffMatch. (a) 3D planes are extracted from two Ouster LiDAR scans, illustrated with red and blue point clouds. Plane-to-plane correspondences are identified, allowing the sensors to be calibrated without an initial guess. (b) 3D plane detections of calibration targets are extracted from depth sensor and from intrinsically-calibrated RGB sensor using PnP. RGB and depth data collected from Intel D435i are arbitrarily transformed to simulate uncalibrated sensors.

using is presented, where 3D planes are extracted from chess boards in the image frame and matched to planes extracted from the depth sensor. GraffMatch correctly matches 7 planes in 23 ms resulting in a calibration error of 0.9 deg and 5 cm.

V. CONCLUSION

We presented GraffMatch, a global method for matching and registering 3D lines and planes from two landmark sets without an initial alignment guess. Our main contribution is the representation and data association of lines and planes as elements of the affine Grassmannian manifold. By naturally representing these geometric landmarks as affine subspaces, we leverage the Grassmannian metric to calculate the distance between two landmarks. We prove that affine Grassmannian distance is invariant to rotation and translation provided a shift

operation is first applied. This invariance property enables the use of efficient and robust graph-theoretic data association.

Future research includes reducing runtime and avoiding symmetries by developing landmark descriptors for generating putative matches (i.e., instead of an all-to-all hypothesis), and estimating lines and planes directly via subspace tracking methods and using manifold-based optimization techniques to perform online bundle adjustment of affine Grassmannian landmarks within a SLAM framework.

APPENDIX

A. Proof of Invariance

Proof. Suppose $\mathbb{Y}_1, \mathbb{Y}_2$ are shifted by $-b_1$ such that $b_1 = 0$. This implies that the orthogonal displacement of \mathbb{Y}_1 is $b_{01} = 0$ and the inner product of the Stiefel coordinates of $\mathbb{Y}_1, \mathbb{Y}_2$ is

$$Y_1^\top Y_2 = \begin{bmatrix} A_1^\top A_2 & \frac{1}{\eta_2} A_1^\top b_{02} \\ \frac{1}{\eta_2} b_{01}^\top A_2 & \frac{1}{\eta_1 \eta_2} (b_{01}^\top b_{02} + 1) \end{bmatrix} = \begin{bmatrix} A_1^\top A_2 & \frac{1}{\eta_2} A_1^\top b_{02} \\ 0 & \frac{1}{\eta_1 \eta_2} \end{bmatrix},$$

with $\eta_i \stackrel{\text{def}}{=} \sqrt{\|b_{0i}\|^2 + 1}$. Recall that computing $d_{\text{Graft}}(\mathbb{Y}_1, \mathbb{Y}_2)$ uses the SVD of $Y_1^\top Y_2$. Given $T = (R, t) \in \text{SE}(3)$, let $\bar{\mathbb{Y}}_1, \bar{\mathbb{Y}}_2$ be the transformations of $\mathbb{Y}_1, \mathbb{Y}_2$, with affine coordinates

$$\mathbb{Y}_i : [A_i, b_i] \xrightarrow{T} \bar{\mathbb{Y}}_i : [RA_i, Rb_i + t].$$

Shifting $\bar{\mathbb{Y}}_1, \bar{\mathbb{Y}}_2$ by $-\bar{b}_1 \stackrel{\text{def}}{=} -(Rb_1 + t)$ leads to the affine coordinates $\bar{\mathbb{Y}}_1 : [RA_1, 0]$ and $\bar{\mathbb{Y}}_2 : [RA_2, Rb_2]$ so that

$$\bar{Y}_1^\top \bar{Y}_2 = \begin{bmatrix} A_1^\top A_2 & \frac{1}{\eta_2} A_1^\top b_{02} \\ 0 & \frac{1}{\eta_1 \eta_2} \end{bmatrix} = Y_1^\top Y_2,$$

implying that $d_{\text{Graft}}(\mathbb{Y}_1, \mathbb{Y}_2)$ is invariant to (R, t) . \square

REFERENCES

- [1] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2015.
- [2] S. Garg, T. Fischer, and A. Milford, "Where is your place, visual place recognition?" in *IJCAI*, 8 2021, pp. 4416–4425.
- [3] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE T-RO*, pp. 1188–1197, 2012.
- [4] G. Kim, S. Choi, and A. Kim, "Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1856–1874, 2022.
- [5] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *IEEE CVPR*, 2017, pp. 1802–1811.
- [6] A. Schaefer, D. Büscher, J. Vertens, L. Luft, and W. Burgard, "Long-term urban vehicle localization using pole landmarks extracted from 3-d lidar scans," in *IEEE ECMR*, 2019, pp. 1–7.
- [7] J. Kümmerle, M. Sons, F. Poggendorf, T. Kühner, M. Lauer, and C. Stiller, "Accurate and efficient self-localization on roads using basic geometric primitives," in *IEEE ICRA*, 2019, pp. 5965–5971.
- [8] B. Cao, C.-N. Ritter, D. Göhring, and R. Rojas, "Accurate localization of autonomous vehicles based on pattern matching and graph-based optimization in urban environments," in *IEEE ITSC*, 2020.
- [9] B. Cao, R. C. Mendoza, A. Philipp, and D. Göhring, "Lidar-based object-level slam for autonomous vehicles," in *IEEE/RSJ IROS*, 2021.
- [10] H. Bavl, J. L. Sanchez-Lopez, M. Shaheer, J. Civera, and H. Voos, "Situational graphs for robot navigation in structured indoor environments," in *IEEE/RSJ IROS*, 2022.
- [11] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems*, Berkeley, CA, 2014.
- [12] M. Kaess, "Simultaneous localization and mapping with infinite planes," in *IEEE ICRA*, 2015, pp. 4605–4611.
- [13] P. Geneva, K. Eickenhoff, Y. Yang, and G. Huang, "Lips: Lidar-inertial 3d plane slam," in *IEEE/RSJ IROS*, 2018, pp. 123–130.
- [14] L. Zhou, G. Huang, Y. Mao, J. Yu, S. Wang, and M. Kaess, "Plc-lislam: Lidar slam with planes, lines and cylinders," *IEEE RA-L*, 2022.
- [15] L. Zhou, S. Wang, and M. Kaess, "π-lsam: Lidar smoothing and mapping with planes," in *IEEE ICRA*, 2021, pp. 5751–5757.
- [16] K. Pathak, A. Birk, N. Vaskevicius, and J. Poppinga, "Fast registration based on noisy planes with unknown correspondences for 3-d mapping," *IEEE T-RO*, vol. 26, no. 3, pp. 424–441, 2010.
- [17] E. Fernández-Moral, W. Mayol-Cuevas, V. Arevalo, and J. Gonzalez, "Fast place recognition with plane-based maps," in *IEEE ICRA*, 2013.
- [18] J. Jiang, J. Wang, P. Wang, P. Bao, and Z. Chen, "LiPMatch: Lidar point cloud plane based loop-closure," *IEEE RA-L*, pp. 6861–6868, 2020.
- [19] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "SegMatch: Segment based place recognition in 3d point clouds," in *IEEE ICRA*, 2017.
- [20] P. C. Lusk, K. Fathian, and J. P. How, "CLIPPER: A graph-theoretic framework for robust data association," in *IEEE ICRA*, 2021, code: <https://github.com/mit-acl/clipper>.
- [21] J. Shi, H. Yang, and L. Carlone, "Robin: a graph-theoretic approach to reject outliers in robust estimation using invariants," in *ICRA*, 2021.
- [22] P. C. Lusk and J. P. How, "Global data association for slam with 3d grassmannian manifold objects," in *IEEE/RSJ IROS*, 2022.
- [23] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *IEEE TPAMI*, vol. 14, no. 2, 1992, pp. 239–256.
- [24] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *IEEE ICRA*, 2009, pp. 3212–3217.
- [25] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *IEEE/CVF ICCV*, 2019, pp. 8958–8966.
- [26] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, 1981.
- [27] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *European conference on computer vision*. Springer, 2016.
- [28] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE T-RO*, vol. 37, no. 2, pp. 314–333, 2020.
- [29] Y. Yang and G. Huang, "Observability analysis of aided ins with heterogeneous features of points, lines, and planes," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1399–1418, 2019.
- [30] D. Wilbers, C. Merfelds, and C. Stachniss, "Localization with sliding window factor graphs on third-party maps for automated driving," in *IEEE ICRA*, 2019, pp. 5951–5957.
- [31] J. Hamm and D. D. Lee, "Grassmann discriminant analysis: a unifying view on subspace-based learning," in *ICML*, 2008.
- [32] Z. Huang, R. Wang, S. Shan, and X. Chen, "Projection metric learning on grassmann manifold with application to video based face recognition," in *IEEE CVPR*, 2015, pp. 140–149.
- [33] S. Shirazi, M. T. Harandi, B. C. Lovell, and C. Sanderson, "Object tracking via non-euclidean geometry: A grassmann approach," in *IEEE WACV*, 2014, pp. 901–908.
- [34] S. Calinon, "Gaussians on riemannian manifolds: Applications for robot learning and adaptive control," *IEEE RAM*, pp. 33–45, 2020.
- [35] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM SIMAX*, pp. 303–353, 1998.
- [36] A. Björck and G. H. Golub, "Numerical methods for computing angles between linear subspaces," *Mathematics of computation*, 1973.
- [37] L.-H. Lim, K. S.-W. Wong, and K. Ye, "The grassmannian of affine subspaces," *Foundations of Computational Mathematics*, 2021.
- [38] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE TPAMI*, no. 5, pp. 698–700, 1987.
- [39] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE CVPR*, 2012.
- [40] Y. Liao, J. Xie, and A. Geiger, "Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d," *IEEE TPAMI*, 2022.
- [41] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of michigan north campus long-term vision and lidar dataset," *IJRR*, vol. 35, no. 9, pp. 1023–1035, 2016.
- [42] H. Dong, X. Chen, and C. Stachniss, "Online range image-based pole extractor for long-term lidar localization in urban environments," in *IEEE ECMR*, 2021.
- [43] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *IEEE/CVF ICCV*, 2019.
- [44] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.
- [45] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.
- [46] A. M. Araújo and M. M. Oliveira, "A robust statistics approach for plane detection in unorganized point clouds," *Pattern Recognition*, 2020.
- [47] J. Beltrán, C. Guindel, F. García *et al.*, "Automatic extrinsic calibration method for lidar and camera sensor setups," *IEEE T-ITS*, 2022.