Dynamic-ICP: Doppler-Aware Iterative Closest Point Registration for Dynamic Scenes

Dong Wang, Daniel Casado Herraez, Stefan May and Andreas Nüchter

***Abstract*—Reliable odometry in highly dynamic environments remains challenging when it relies on ICP-based registration: ICP assumes near-static scenes and degrades in repetitive or low-texture geometry. We introduce Dynamic-ICP, a Doppler-aware registration framework. The method (i) estimates ego translational velocity from per-point Doppler velocity via robust regression and builds a velocity filter, (ii) clusters dynamic objects and reconstructs object-wise translational velocities from ego-compensated radial measurements, (iii) predicts dynamic points with a constant-velocity model, and (iv) aligns scans using a compact objective that combines point-to-plane geometry residual with a translation-invariant, rotation-only Doppler residual. The approach requires no external sensors or sensor–vehicle calibration and operates directly on FMCW LiDAR range and Doppler velocities. We evaluate Dynamic-ICP on three real-world datasets-HeRCULES, HeLiPR, AevaScenes-focusing on highly dynamic scenes. Dynamic-ICP consistently improves rotational stability and translation accuracy over the state-of-the-art methods. To encourage further research, the code is available at: https://github.com/JMUWRobotics/Dynamic-ICP.**

***Index Terms*—Odometry, Mapping, Localization, SLAM, Autonomous Vehicle Navigation**

I. INTRODUCTION

RELIABLE odometry in unknown environments is a fundamental requirement for robust autonomy in ground, aerial, and mobile robots. Decades of research have yielded numerous point cloud registration methods for odometry pipelines, among which the Iterative Closest Point (ICP) [2] [6] has been widely adopted due to its precision and efficiency. ICP aligns a source to a target cloud by alternating correspondence search with transform refinement to minimize the geometric error. While ICP variants can be highly accurate in quasistatic scenes, they implicitly assume that most points are stationary between successive frames, an assumption violated in highly dynamic settings. As a result, moving objects induce spurious correspondences and bias, and performance further degrades in geometrically repetitive, low-texture environments (e.g., tunnels, bridges) where ambiguous matches can lead to drift.

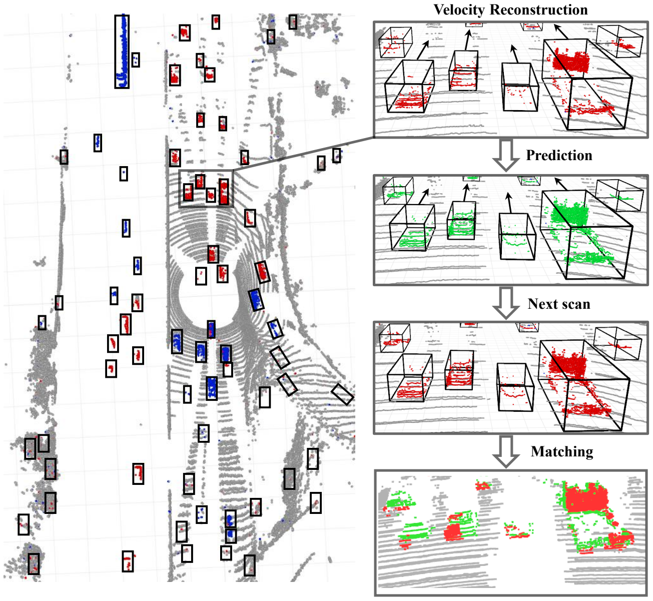

Fig. 1: Workflow of Dynamic-ICP for dynamic objects. Left: FMCW LiDAR scan in highway scenarios. Dynamic points are colored and clustered. Right: Velocity reconstruction, prediction, and matching of dynamic objects. The raw dynamic points are colored red, while the predicted points are colored green. Black arrows and boxes represent the object's velocity and bounding box, respectively.

In recent years, frequency-modulated continuous-wave (FMCW) LiDAR technology has advanced beyond traditional LiDAR systems by providing range and per-point Doppler velocity. These sensors measure direct motion cues in dynamic scenes by comparing phase shifts across the transmitted chirp sequence. However, the use of Doppler for understanding dynamic scenes is still in early stages.

To address these limitations, we propose Dynamic-ICP, a Doppler-aware variant of ICP that explicitly models scene dynamics and fast ego motion. Rather than assuming a fully static scene, Dynamic-ICP reduces sensitivity to dynamics by predicting where points will be in the next frame before establishing correspondences. Concretely, we exploit per-point Doppler velocity measurements (e.g., from FMCW sensors) to cluster moving objects, reconstruct each object's full 3D velocity, and warp points into their next-frame states, as shown in Fig. 1. A distance-adaptive dynamic correlation then guides correspondence weighting, suppressing spurious matches from dynamic objects while preserving informative structure. Compared to approaches that simply reject dynamic points or rely solely on robust loss functions [1] [24] [36], Dynamic-ICP pri-

Manuscript received: October 21, 2025; Revised: January 26, 2026; Accepted: February 23, 2026. This paper was recommended for publication by Editor Javier Civera upon evaluation of the Associate Editor and Reviewers' comments. This work was in parts supported by the Federal Ministry for Economic Affairs and Climate Action (BMWK) on the basis of a decision by the German Bundestag under the grant number KK5150106RL4.

D. Wang (dong.wang@uni-wuerzburg.de) and A. Nüchter are with Julius-Maximilians-Universität Würzburg, Germany. Andreas Nüchter is also with the Zentrum für Telematik e.V., Würzburg and currently International Visiting Chair at U2IS, ENSTA, Institut Polytechnique de Paris, France. S. May is with Nuremberg Institute of Technology Georg Simon Ohm, Germany. D. Casado Herraez is with CARIAD SE and the University of Bonn, Germany.

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

oritizes Doppler-consistent pairs and performs correspondence search in a motion-compensated domain. The added rotation-only Doppler residual eliminates classic ICP degeneracies and reduces drift in repetitive or low-texture regions. In highly dynamic scenes, per-cluster velocity prediction aligns moving objects across frames. It also stabilizes orientation under strong ego and object motion. All of this is done while retaining the simplicity and real-time efficiency that makes ICP practical.

We evaluate Dynamic-ICP across diverse, highly dynamic scenarios with substantial object motion. Experiments demonstrate consistent gains in rotation/translation accuracy, faster convergence, and improved robustness under severe dynamics. We also report detailed ablations to quantify the effect of Doppler-based clustering, velocity reconstruction, and distance-adaptive correlation on overall performance. Code will be released to facilitate reproduction and extension. To summarize, our contributions are:

- We present Dynamic-ICP, a Doppler-aware dynamic ICP designed for highly dynamic scenes where standard ICP degrades.
- We leverage per-point Doppler velocity to cluster moving objects and reconstruct each object's translational velocity. A distance-adaptive motion correlation then extrapolates object states to the next frame to guide correspondences.
- Our Doppler-consistent matching substantially alleviates the rotation-estimation failure modes of vanilla ICP, improving stability and accuracy under strong ego- and object motion.
- Extensive experiments across diverse datasets show state-of-the-art performance in highly dynamic scenarios.

II. Related Work

In this section, we review state-of-the-art ICP approaches, including traditional frameworks and recent methods that leverage Doppler information. We also discuss techniques that integrate ICP with scene flow estimation and present our method in the context of Doppler-aware ICP for dynamic scenes.

A. Iterative Closest Point and Variants

Iterative Closest Point (ICP) is the canonical method for rigid point cloud alignment, originating from early formulations that alternated between closest-point association and least-squares pose estimation [2] [6]. Efficiency and robustness have been improved through careful choices of metrics, sampling, and weighting [26] [27], probabilistic modeling as in Generalized-ICP [28], and outlier handling via trimmed objectives [7]. Distribution-based registration such as the normal distributions transform, replaces discrete matches with continuous densities to enhance convergence in sparse or noisy settings [3] [23]. To extend ICP to large baselines, global or certifiable methods provide strong initializations and outlier guarantees [34] [37]. Recent variants target real-time and degenerate scenes: Voxelized GICP accelerates and stabilizes GICP with voxel-wise covariance aggregation [19], and KISS-ICP [30] shows that a carefully engineered point-to-point pipeline with simple motion compensation and robust thresholds can be both accurate and broadly applicable across sensors. Despite these advances, most ICP-style pipelines implicitly assume near-static scenes over short horizons, which limits performance under strong ego motion and highly dynamic environments.

B. Doppler Velocity based Matching

FMCW LiDAR sensors provide per-point radial velocity via Doppler shift, offering direct motion cues that can disambiguate correspondences under strong ego and object motion. Previous work has exploited Doppler velocity for instantaneous estimation of ego-motion, using the radial velocity constraint to separate static structures from moving objects and to infer velocity vectors [13] [31]. Incorporating Doppler velocity into registration has been explored by constraining or guiding correspondences based on velocity consistency. This approach is sometimes referred to as Doppler-constrained ICP or motion-compensated matching [5] [10] [11] [33]. In the SLAM framework, the Doppler velocity sensor is integrated with inertial and other exteroceptive sensors to stabilize odometry, reduce drift, and improve data association in challenging conditions, such as rain, low texture, or repeated geometry [35]. However, most existing methods rely on predominantly static environments and treat moving points as outliers. In contrast, our Doppler-aware matching method reconstructs 3D velocities for each object from the Doppler effect and predicts their positions in the next frame. This method yields more reliable correspondences and stronger rotational constraints in highly dynamic scenes.

C. Scene Flow

Scene flow estimates a dense 3D motion field between consecutive observations, generalizing optical flow to three dimensions [29]. Classical formulations solved for per-point 3D motion under smoothness constraints, while modern learning-based methods infer flow directly from point clouds using cost volumes, permutation-invariant layers, and optimal transport objectives [9] [18] [22]. Recent directions relevant to dynamic registration include radar-based cross-modal supervision for 4D radar scene flow [8], ICP-Flow [21], which uses rigid-motion priors and classical ICP to produce consistent object-wise flow without training, and 4D voxel networks that fuse multiple frames for efficient spatio-temporal reasoning [17]. Complementary to these, cross-modal Doppler guidance has been proposed to transfer radar-derived 3D velocities to LiDAR for self-supervised scene flow [15]. However, dense flow estimation can be computationally demanding and may rely on learned priors that do not transfer across sensors or environments. In contrast, Dynamic-ICP provides a lightweight, training-free alternative to dense scene flow that retains real-time efficiency in highly dynamic scenes.

Inspired by previous works Doppler-ICP [11], DoGFlow [15], Flow4d [17], ICP-Flow [21] and Doppler-SLAM [31], we propose Dynamic-ICP, a Doppler-aware registration framework that integrates per-point Doppler velocity into correspondence search. Rather than masking dynamic content or relying on dense scene flow estimation, Dynamic-ICP reconstructs object-wise 3D velocities from Doppler velocity, predicts next-frame positions with a distance-adaptive correlation model, and performs ICP in this motion-compensated space. This design yields more reliable correspondences and improved rotation and translation estimates in high-speed, highly dynamic scenes while preserving the simplicity and efficiency of ICP. Another advantage of our method is that it operates without requiring extrinsic sensor-vehicle calibration compared to Doppler-ICP [11]. In summary, Dynamic-ICP combines Doppler-aware matching and ICP-style optimization to provide lightweight, robust registration for classic ICP's failure modes in dynamic scenes.

III. Dynamic-ICP

We use standard notation: vectors are bold lowercase (e.g., $\boldsymbol{b}$), matrices bold uppercase (e.g., $\boldsymbol{B}$), and frames by superscripts/subscripts (e.g., $\boldsymbol{b}^l$). The rigid transform is $\boldsymbol{T} = [\boldsymbol{R}\,|\,\boldsymbol{t}]$ with $\boldsymbol{R} \in SO(3)$ and $\boldsymbol{t} \in \mathbb{R}^3$.

A. Problem Statement

We denote the sensor frame for the previous point cloud as the source frame, $\mathcal{F}_S$, and the current point cloud as the target frame, $\mathcal{F}_T$. Let

$$\boldsymbol{u}_i^t \;\triangleq\; \frac{\boldsymbol{p}_i^t}{\|\boldsymbol{p}_i^t\|}, \mathcal{P}_t = \{(\boldsymbol{p}_i^t,\ \boldsymbol{u}_i^t,\ s_i^t)\}_{i=1}^{N_t}, \tag{1}$$

be the point set acquired at time t, where $\boldsymbol{p}_i^t \in \mathbb{R}^3$ is the position of the point in the sensor frame, $\boldsymbol{u}_i^t \in \mathbb{S}^2$ is the unit line-of-sight (LOS) vector from the sensor to $\boldsymbol{p}_i^t$, and $s_i^t \in \mathbb{R}$ is the measured Doppler velocity along $\boldsymbol{u}_i^t$ (positive away from the sensor). Likewise $\mathcal{P}_{t+1}$ is available at time $t{+}1$. Our goals are to:

1) estimate the transformation $\boldsymbol{T}_{t\to t+1} \in SE(3)$ that aligns $\mathcal{P}_t$ to $\mathcal{P}_{t+1}$;
2) achieve registration that remains reliable under strong ego- and object motion.

We write $\boldsymbol{T} = [\boldsymbol{R} \mid \boldsymbol{t}]$ with $\boldsymbol{R} \in SO(3)$ and $\boldsymbol{t} \in \mathbb{R}^3$. A body twist is $\boldsymbol{\xi} = [\boldsymbol{\omega}^\top\ \boldsymbol{v}^\top]^\top \in \mathbb{R}^6$.

B. System Overview

Fig. 2 illustrates the overall system architecture, highlighting four key modules: (i) Ego-Motion Estimation, which estimates the ego velocity from Doppler velocity, (ii) Dynamic Points Clustering, which clusters the dynamic points into individual objects, (iii) Dynamic Points Prediction, predicting the dynamic points in next frame, and (iv) Doppler-aware ICP Matching. The following subsections describe the design and implementation of each module in detail.

C. Ego-Motion Estimation

To estimate the ego-motion of the sensor, we first leverage the static points. For static scene points, the measured Doppler velocity equals the velocity induced by the ego-motion at the point projected onto the LOS:

$$s_i^t = -\,(\boldsymbol{u}_i^t)^\top \big(\boldsymbol{v} + \boldsymbol{\omega} \times \boldsymbol{p}_i^t\big), \tag{2}$$

where $\boldsymbol{v}, \boldsymbol{\omega}$ are the instantaneous linear and angular velocities of the sensor at time t. With our LOS definition $\boldsymbol{u}_i^t$ being the unit vector from the sensor origin to $\boldsymbol{p}_i^t$ (thus $\boldsymbol{u}_i^t = \boldsymbol{p}_i^t/\|\boldsymbol{p}_i^t\|$), the radial projection of the rotational velocity about the sensor origin vanishes, i.e., $(\boldsymbol{u}_i^t)^\top(\boldsymbol{\omega} \times \boldsymbol{p}_i^t) = 0$. Hence, single-scan radial Doppler constrains $\boldsymbol{v}$ but does not directly observe $\boldsymbol{\omega}$. Rotation is estimated in Sec. III-F through scan-to-scan registration using geometric constraints and the Doppler matching term.

We estimate $\boldsymbol{v}$ by robust regression over all points in the scan:

$$\min_{\boldsymbol{v}} \sum_i \rho\big(s_i^t + (\boldsymbol{u}_i^t)^\top \boldsymbol{v}\big)\,, \tag{3}$$

with $\rho(\cdot)$ being a robust penalty (e.g., Huber). Eq. (3) is solved as a linear least squares in the three unknowns of $\boldsymbol{v}$ [32].

Unlike previous ego motion estimators that employ global static assumptions [14] [20], we suppose that a majority of points are static only at initialization. If $t = 0$, we set a neutral prior $\boldsymbol{v}^0 = \mathbf{0}$. For subsequent frames ($t > 0$), we initialize the prediction from ego-velocity obtained from the previous registration result $\hat{\boldsymbol{T}}_{t-1\to t}$. This policy reduces reliance on a persistent "mostly static" assumption beyond the first frame and accelerates convergence of Eq. (3). At $t = 0$, Eq. Eq. (3) benefits from sufficient static support. If the sensor is initialized in a scene dominated by moving objects (e.g., heavy traffic with limited visible background), the initial estimate and thus the velocity filter can degrade for the first frames. With the solution $\hat{\boldsymbol{v}}$, the velocity filter is defined as

$$r_i^t \;=\; s_i^t + (\boldsymbol{u}_i^t)^\top \hat{\boldsymbol{v}}. \tag{4}$$

A point is flagged dynamic if $|r_i^t| > \tau(d_i)$, where $d_i = \|\boldsymbol{p}_i^t\|$ and $\tau(d) = \tau_0 + \kappa d$ is a distance-adaptive threshold. The remaining points form the static set $\mathcal{S}_t$.

D. Dynamic Point Clustering

After computing the ego-velocity from static points, we use the dynamic points to cluster dynamic objects. Let $\mathcal{D}_t = \mathcal{P}_t \setminus \mathcal{S}_t$ be the set of dynamic candidates identified by the velocity filter. We employ spatial and velocity consistency analysis to cluster dynamic points.

Spatial consistency: We adopt HDBSCAN [4], a hierarchical density-based clustering method robust to uneven point density and partial occlusion. We apply HDBSCAN to $\mathcal{D}_t$ with minimum cluster size m_{cs} (smallest object) and minimum samples m_{s} (controls density conservativeness and outlier robustness), producing clusters $\{\mathcal{C}_k^t\}$ and an explicit noise set $\mathcal{N}_t$.

Velocity consistency: For each cluster $\mathcal{C}_k^t$, we assume that all points within the cluster belong to a rigidly moving

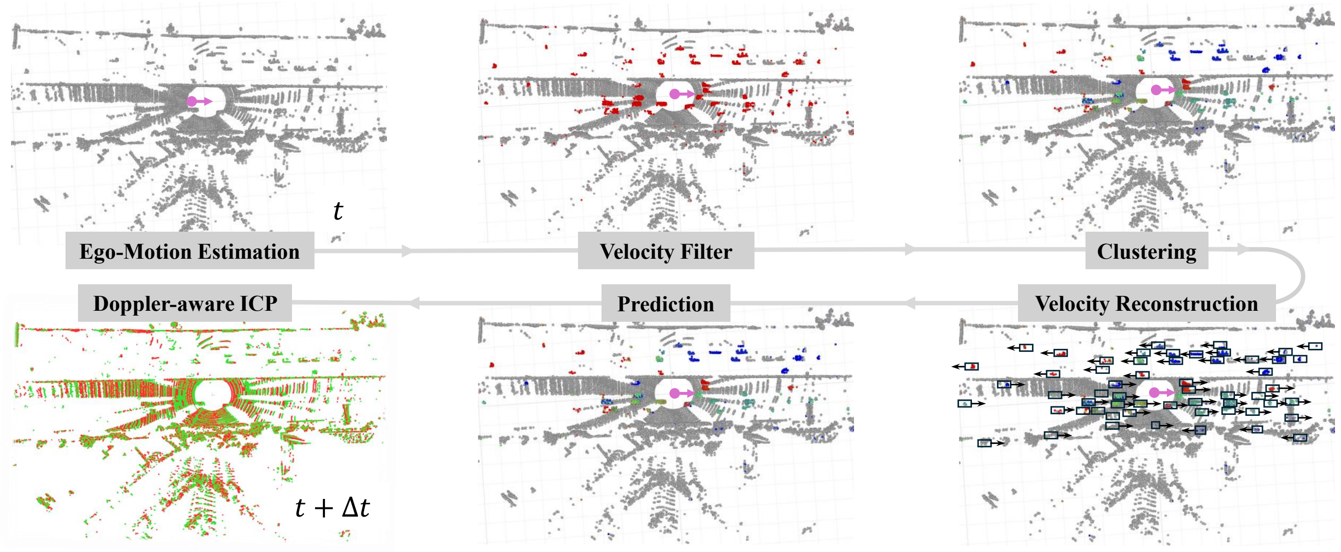

Fig. 2: Pipeline of Dynamic-ICP consists of four main modules: (i) Ego-Motion Estimation (Sec. III-C); (ii) Dynamic Points Clustering (Velocity Filter, Clustering and Velocity Reconstruction) (Sec. III-D); (iii) Dynamic Points Prediction (Sec. III-E); and (iv) Doppler-aware ICP Matching (Sec. III-F). The figure illustrates the workflow on point cloud data: starting from a raw scan, ego velocity (arrows) is estimated from per-point Doppler velocity. The velocity filter distinguishes dynamic points (red) from the static background (gray) and clusters the dynamic set into individual objects (colored). For each cluster, the object velocity is reconstructed from its points' Doppler velocities, yielding a velocity vector and a bounding box (black arrow and box). These velocities are then used to predict object states to the next frame. Finally, the scan (green) at time t (predicted dynamic points, together with the static points) are aligned to the scan (red) at time $t + \Delta t$ via Doppler-aware ICP.

object, as illustrated in Fig. 3. We first form the ego-motion–compensated Doppler velocity $\tilde{s}_i^t$ for each dynamic point $\tilde{s}_i^t$,

$$\tilde{s}_i^t \triangleq s_i^t + (\boldsymbol{u}_i^t)^\top \hat{\boldsymbol{v}}, \tag{5}$$

and recover the cluster velocity $\hat{\boldsymbol{v}}_k^t$ by ordinary least squares:

$$\varepsilon_i = (\boldsymbol{u}_i^t)^\top \hat{\boldsymbol{v}}_k^t - \tilde{s}_i^t, \qquad i \in \mathcal{C}_k^t, \tag{6}$$

where ε_i denotes the per-point velocity residuals and is then filtered by a velocity-adaptive threshold $\tau_v = \lambda \cdot \hat{\boldsymbol{v}}$. We keep only the points satisfying

$$|\varepsilon_i| \leq \tau_v, \tag{7}$$

removing the rest to $\mathcal{N}_t$. If the inlier fraction falls below a threshold $\phi_{\min}$, the entire cluster is discarded. The 3D velocity reconstruction above requires sufficiently diverse LOS directions within a cluster. For distant or compact objects, the LOS vectors can become nearly parallel, making tangential components weakly observable and the least-squares problem ill-conditioned. We therefore assemble

$$\boldsymbol{U}_k^t \triangleq \begin{bmatrix} (\boldsymbol{u}_{i_1}^t)^\top \\ \vdots \\ (\boldsymbol{u}_{i_M}^t)^\top \end{bmatrix} \in \mathbb{R}^{M\times 3}, \qquad \tilde{\boldsymbol{s}}_k^t \triangleq \begin{bmatrix} \tilde{s}_{i_1}^t \\ \vdots \\ \tilde{s}_{i_M}^t \end{bmatrix}, \tag{8}$$

where $\{i_m\}_{m=1}^M$ denotes the indices of the points in $\mathcal{C}_k^t$. We then estimate the cluster velocity by solving the linear system $\boldsymbol{U}_k^t \hat{\boldsymbol{v}}_k^t \approx \tilde{\boldsymbol{s}}_k^t$. To assess whether this system is numerically stable, we compute the singular values of $\boldsymbol{U}_k^t$. Let $\sigma_{\max}$ and $\sigma_{\min}$ be the largest and smallest singular values, and define the condition number $\kappa(\boldsymbol{U}_k^t) = \sigma_{\max}/\sigma_{\min}$. When $\sigma_{\min}$ is very small (nearly rank-deficient) or $\kappa(\boldsymbol{U}_k^t)$ is large (ill-conditioned), the tangential velocity components become unreliable. In these cases, we discard the cluster. This conservative filtering ensures that prediction is applied only to validated clusters, while the ego pose is estimated by scan-to-scan registration in Sec. III-F. After evaluating the spatial and velocity consistency of the clusters, the remaining points form refined clusters $\{\widetilde{\mathcal{C}}_k^t\}$. These clusters, along with their estimated velocities $\{\hat{\boldsymbol{v}}_k^t\}$, are used in subsequent prediction and matching stages.

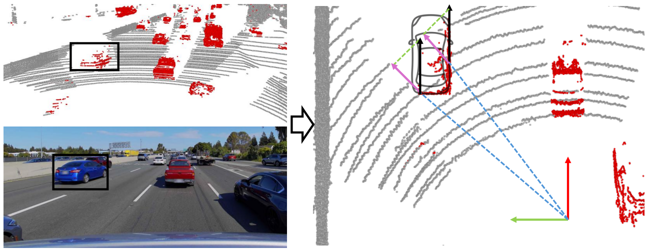

Fig. 3: Velocity reconstruction of dynamic objects. Left: FMCW LiDAR scan with dynamic points highlighted in red and synchronized camera view. Right: Top-down view of LiDAR scan. Dynamic points on the moving rigid body (car) satisfy the Doppler velocity consistency condition. This means that the velocity component of the object (black arrow) in the direction of the line-of-sight (blue dashed line) equals the compensated Doppler velocity (purple arrow) at that point.

E. Dynamic Points Prediction

We predict only points belonging to dynamic clusters using a constant-velocity motion model. For each point i in the refined dynamic clusters $\{\widetilde{\mathcal{C}}_k^t\}$ with translational velocities $\{\hat{\boldsymbol{v}}_k^t\}$,

$$\tilde{\boldsymbol{p}}_i^{t+1} \;=\; \boldsymbol{p}_i^t \;+\; \hat{\boldsymbol{v}}_k^t \Delta t\,, \tag{9}$$

which assumes cluster-wise constant translational velocity over Δt. The constant-velocity model sets the cluster's angular rate to zero during prediction. If an object rotates with angular velocity $\boldsymbol{\omega}_k$, the unmodeled per-point displacement (relative to the cluster centroid $\boldsymbol{c}_k^t$) is

$$\delta \boldsymbol{p}_i^{\mathrm{rot}} \;\approx\; \big(\boldsymbol{\omega}_k \times (\boldsymbol{p}_i^t - \boldsymbol{c}_k^t)\big)\,\Delta t. \tag{10}$$

Because Doppler provides only radial velocity, $\boldsymbol{\omega}_k$ is not directly observable. We therefore (i) keep Δt short, so $\|\delta \boldsymbol{p}_i^{\mathrm{rot}}\|$ remains small and unbiased at the centroid, (ii) use a slightly enlarged, distance-aware correspondence threshold for larger objects, and (iii) rely on the ICP point-to-plane term to absorb the residual rotational misalignment during pose refinement. In practice, this preserves accurate centroid prediction while bounding per-point errors by object size and rotation rate. Points not in dynamic clusters are considered stationary during the sampling interval and thus are not predicted and remain at $\boldsymbol{p}_i^t$. The source set passed to matching is thus

$$\widetilde{\mathcal{P}}_t = \Big\{\boldsymbol{p}_i^t \mid i \in \mathcal{S}_t\Big\} \;\cup\; \bigcup_k \Big\{\tilde{\boldsymbol{p}}_j^{t+1} \mid j \in \widetilde{\mathcal{C}}_k^{t+1}\Big\}, \tag{11}$$

providing next-frame predictions for moving objects while leaving static structure unchanged for the subsequent Doppler-aware ICP alignment.

F. Doppler-aware ICP Matching

Given the predicted source set $\widetilde{\mathcal{P}}_t = \{(\boldsymbol{p}_i, \boldsymbol{u}_i, s_i)\}_{i=1}^{N_t}$ and the target set $\widetilde{\mathcal{P}}_{t+1} = \{(\boldsymbol{q}_j, \boldsymbol{n}_j, \boldsymbol{u}_j, s_j)\}_{j=1}^{N_{t+1}}$, we estimate the rigid motion $\boldsymbol{T} = [\boldsymbol{R} \,|\, \boldsymbol{t}] \in SE(3)$ with a two-term objective that combines geometry residual and Doppler residual. Here, $\boldsymbol{n}_j \in \mathbb{R}^3$ is the unit surface normal at $\boldsymbol{q}_j$ with it's neighbors, obtained by local plane fitting and normalized so that $\|\boldsymbol{n}_j\| = 1$.

Geometry residual: the point-to-plane term penalizes displacement along the target normal, encouraging the transformed source points to lie on the target surface and yielding fast, stable convergence in rigid registration [27]:

$$r_{g,ij} \;=\; \boldsymbol{n}_j^\top \big(\boldsymbol{R}\tilde{\boldsymbol{p}}_i + \boldsymbol{t} - \boldsymbol{q}_j\big). \tag{12}$$

Doppler residual: the Doppler term enforces agreement between the rotated source Doppler velocity and the target Doppler velocity along the line-of-sight, as illustrated in Fig. 4. It provides a complementary, rotation-focused cue that is independent of translation:

$$r_{v,ij} \;=\; \boldsymbol{u}_j^\top \, \boldsymbol{R}\big(s_i\, \boldsymbol{u}_i\big) \;-\; s_j. \tag{13}$$

We minimize a robust iteratively reweighted least squares objective

$$[\boldsymbol{R} \,|\, \boldsymbol{t}] = \min \sum_{(i,j)} \Big[\, (1-\lambda_v)\rho_g\big(r_{g,ij}^2\big) \;+\; \lambda_v\, \rho_v\big(r_{v,ij}^2\big) \,\Big], \tag{14}$$

where ρ_g, ρ_v are robust kernels (e.g., Huber) and $0 \le \lambda_v \le 1$ balances the two terms. Since $r_{v,ij}$ depends only on $\boldsymbol{R}$ (its derivative with respect to $\boldsymbol{t}$ is zero), it supplies a translation-invariant rotational constraint that (i) remains informative in repetitive or low-texture geometry (e.g., tunnels, bridges), (ii) is insensitive to depth ambiguity along the viewing ray (Fig. 4), and (iii) directly penalizes rotation errors when point-to-plane becomes weak or is affected by residual dynamics.

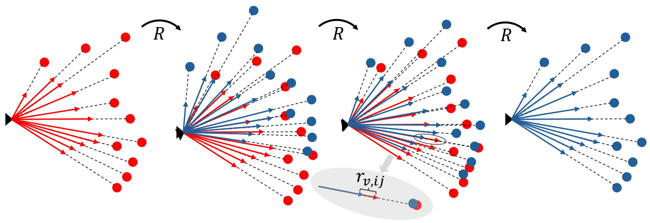

Fig. 4: Doppler residual of Doppler-aware ICP matching. Red points and arrows denote the source frame, and blue points and arrows denote the target frame. Arrows show line-of-sight directions whose lengths are proportional to Doppler (radial) velocity. From left to right, the diagram illustrates the matching process: source Doppler rays are rotated to align with target rays, after which correspondences are established by combining point-to-plane geometry with the Doppler residual.

IV. Experimental Results

A. Implementation Details

We implement Dynamic-ICP by extending the point-to-plane ICP pipeline from Open3D [38] (KD-tree search, point-to-plane Jacobians) with an additional Doppler residual and its Jacobian. LOS directions $\boldsymbol{u}$ are stored separately from surface normals $\boldsymbol{n}$ to keep the geometric term strictly point-to-plane, while the Doppler term uses only LOS information. To limit the effect of outlier correspondences, we use Tukey loss on both terms, with tuning constants $\rho_g = 0.5$ for the point-to-plane geometry residual (meters) and $\rho_v = 0.3$ for the Doppler residual. We fix the balance to $\lambda_v = 0.2$ across all experiments (chosen empirically). Dynamic clustering uses HDBSCAN with $m_{\mathrm{cs}} = 30$ and $m_{\mathrm{s}} = 10$, which control the smallest admissible object and density conservativeness, respectively. We keep these HDBSCAN parameters fixed across all datasets to avoid environment-specific tuning and to maintain a fully automatic and reproducible pipeline. Here, m_{cs} mainly controls the minimum object size (in number of points after voxelization), while m_{s} controls the conservativeness of clustering. We do not use additional sensors (IMU, GNSS, or wheel odometry). The initial pose is derived solely from the FMCW LiDAR: range supports geometric alignment, while per-point Doppler provides (i) an ego translational velocity estimate for dynamic filtering and (ii) an additional rotation-stabilizing cue through the Doppler matching term in Sec. III-F.

B. Experimental Evaluation

We evaluate on four Doppler-capable datasets, focusing on highly dynamic segments:

- HeLiPR [12]: driving sequences, recorded by an FMCW LiDAR.
- HeRCULES [16]: diverse routes with fast ego movement and moving traffic, recorded by an FMCW LiDAR.

- AevaScenes [25]: highway and city scenes with two FMCW LiDARs.
- DICP [11]: curved walls (feature-less scenario) in CARLA Town05 with one simulated FMCW LiDAR.

We use the raw point clouds and raw per-point Doppler measurements as provided, without additional deskewing. For AevaScenes, which only supplies ego-motion compensated point clouds, we use these compensated point clouds directly. Moreover, since AevaScenes provides object-level velocity ground truth, we additionally validate the reconstructed object velocities against this reference.

C. Comparison to State-of-the-Art Methods

We benchmark against five state-of-the-art baselines under identical pre-processing (voxel grid, target normals) and correspondence radius. The baselines are: point-to-point ICP [2], which minimizes Euclidean distances between corresponding points; point-to-plane ICP [27], which projects errors onto target surface normals; Generalized-ICP [28], a probabilistic point-to-plane formulation with local covariances; KISS-ICP [30], a lightweight point-to-point ICP with adaptive correspondence thresholds; and Doppler-ICP [11], which integrates Doppler velocity into point-to-plane ICP and requires sensor–vehicle extrinsic calibration. In contrast, our method does not require calibration between the sensor and the vehicle and is the only approach that performs cluster-wise dynamic prediction. Since we focus on registration accuracy, we report relative pose error (RPE) with respect to reference trajectories provided by the dataset at the frame gap, decomposed into rotation (RRE) and translation (RTE). The best and second best results are **bolded** and underlined, respectively. Qualitative trajectories for the best-performing methods are shown in Fig. 5.

In the first experiment, we analyze the performance of our system and compare it to ICP-based methods on six HeRCULES sequences (Table I). Dynamic-ICP delivers the best or tied-best performance across all scenes in both translation (RTE) and rotation (RRE) errors. It clearly improves rotation accuracy on challenging, repetitive geometry such as *Bridge 01* and *River Island 01*, while also reducing translation error (e.g., *Library 01*). On *Stream 01*, our method attains the lowest RTE and ties DICP for the best RRE. The only case without the top rotational score is *Parking Lot 02*, where DICP and point-to-point outperform the other methods. The reason is that this is the only low ego-motion, near-static scenario, so our method degenerates into point-to-plane approach. This further demonstrates the advantage of our method in highly dynamic scenarios. Overall, Doppler-aware prediction and matching yield consistent orientation gains and competitive translation across diverse scenes. On HeLiPR dataset (Table II), Dynamic-ICP achieves the best translation accuracy on five of six sequences and ties the remaining one, while consistently ranking first or second in rotation across all cases. It is particularly effective on scenes with weak geometric features or evolving appearance: *Bridge 02* contains many moving vehicles with slow ego motion, *Kaist 05* includes numerous pedestrians and long-term appearance changes, and *Riverside 05* features many dynamic objects with long-term differences. Across these settings, motion prediction and Doppler-aware matching preserve reliable correspondences and stabilize orientation when static structure is limited or changing. On the AevaScenes high-speed benchmark (Table III), we evaluate Dynamic-ICP on the full set of 44 highway sequences and 43 urban road sequences, and report the averaged frame-to-frame errors over all sequences. Overall, Dynamic-ICP remains accurate and stable across both highway and city driving, achieving the best performance in both translation and rotation on average. In addition to real-world data, we further conducted experiments on a simulation environment with degraded geometry (*Curved Walls*), which mimics tunnel-like conditions with weak and repetitive structure. In this setting, Doppler-ICP achieves the best performance, which we attribute to its translation-coupled Doppler constraint that directly strengthens translation estimation under geometry degeneracy. Dynamic-ICP remains competitive and delivers consistently strong rotation accuracy, highlighting that the proposed Doppler-aware matching and motion prediction remain effective not only in highly dynamic traffic scenarios but also under challenging geometry. We additionally validate reconstructed cluster velocities on AevaScenes using object-level ground-truth annotations, obtaining low forward-axis error (MAE $0.53\,\mathrm{m/s}$, median relative error 2.87%) and moderate lateral error (MAE $1.08\,\mathrm{m/s}$), supporting reliable short-horizon prediction in real traffic. Table IV summarizes the degree of scene dynamics across representative sequences by reporting the ratio of dynamic candidates and the fraction of dynamic points that are clustered and retained for prediction. Overall, a substantial portion of the dynamic candidates can be consistently grouped and validated, indicating that the proposed dynamic clustering and refinement stage provides reliable support for motion prediction in real traffic scenes.

D. Ablation Studies

We conduct additional experiments to investigate the individual benefits of three core components of Dynamic-ICP: velocity filter (VF), dynamic points prediction (DPP), and the Doppler residual (DR) by deactivating each while holding all other settings fixed. As shown in Table V, disabling the velocity filter means we no longer separate dynamic from static points and, consequently, do not run dynamic points prediction. Dynamic points remain in the "static" set and corrupt correspondences, increasing both RTE and RRE. Removing dynamic points prediction means we still detect dynamics with the velocity filter, but discard them and register using only static points. Dropping Doppler residual reduces the objective to standard point-to-plane ICP, removing the rotation-only cue. Orientation becomes less stable in repetitive or low-texture scenes, increasing RRE even when geometry is well aligned. The full model (VF + DPP + DR) consistently yields the lowest errors.

Table VI compares efficiency and robustness across methods. The *convergence rate* is defined as the percentage of registrations that terminate within $I_{\max} = 100$ iterations and satisfy the threshold $\epsilon = 10^{-5}$ (with a maximum point

Method	Bridge 01		Library 01		Parking Lot 02		River Island 01		Stream 01		Street 01	
	RTE [m]	RRE [°]	RTE [m]	RRE [°]	RTE [m]	RRE [°]	RTE [m]	RRE [°]	RTE [m]	RRE [°]	RTE [m]	RRE [°]
ICP (point-to_point)	1.636	0.562	0.641	0.601	0.090	**0.280**	1.098	0.600	1.033	0.619	0.126	0.150
ICP (point-to-plane)	1.586	0.361	0.516	0.523	**0.051**	0.282	0.899	0.454	0.919	0.471	0.089	0.160
GICP	1.639	0.741	0.849	0.994	0.220	0.510	1.151	0.627	1.112	0.733	0.208	0.293
KISS-ICP	1.637	0.716	0.696	1.214	0.293	0.539	1.327	0.685	1.150	0.833	0.260	0.294
Doppler-ICP	1.024	0.689	0.178	0.511	0.069	0.281	0.380	0.334	0.465	**0.216**	0.071	0.150
Dynamic-ICP	**1.023**	**0.248**	**0.177**	**0.285**	**0.051**	0.282	**0.379**	**0.189**	**0.464**	**0.216**	**0.050**	**0.140**

TABLE I: Comparison of ICP based methods on the HeRCULES dataset.

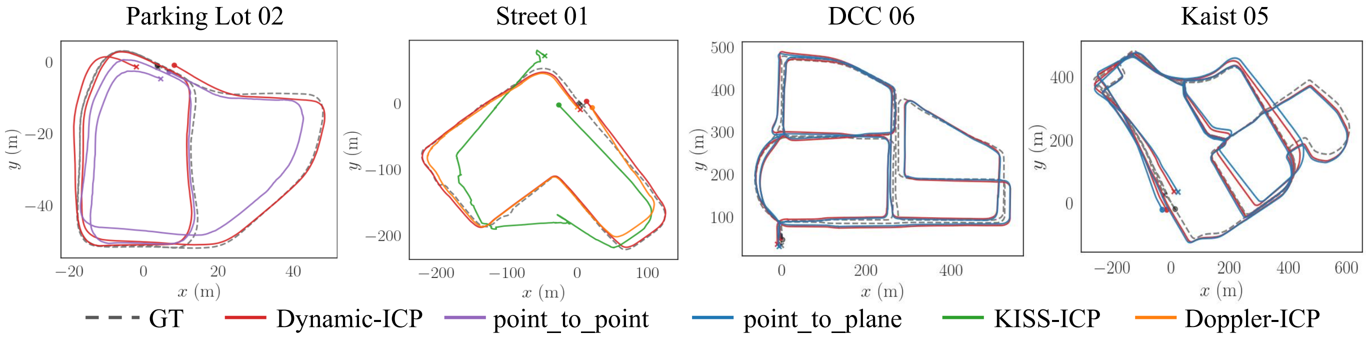

Fig. 5: Qualitative comparison. For clarity, we plot only the two best-performing methods per sequence, including our Dynamic-ICP.

Method	Bridge 02		DCC 06		Kaist 05		Riverside 05		Roundabout 01		Town 01	
	RTE [m]	RRE [°]	RTE [m]	RRE [°]	RTE [m]	RRE [°]	RTE [m]	RRE [°]	RTE [m]	RRE [°]	RTE [m]	RRE [°]
ICP (point-to_point)	0.671	0.252	0.381	0.302	0.324	0.278	0.923	0.366	0.386	0.249	0.370	0.167
ICP (point-to-plane)	0.534	0.195	0.063	**0.085**	**0.037**	0.102	0.636	0.232	**0.105**	0.101	0.195	0.105
GICP	0.562	0.180	0.468	0.366	0.445	0.406	0.926	0.372	0.428	0.281	0.424	0.243
KISS-ICP	0.558	**0.100**	0.237	0.204	0.215	0.266	1.143	0.682	0.222	**0.095**	0.155	0.335
Doppler-ICP	0.748	0.373	0.519	0.361	0.585	0.568	0.959	0.502	0.479	0.391	0.498	0.416
Dynamic-ICP	**0.370**	0.110	**0.062**	**0.085**	**0.037**	**0.101**	**0.592**	**0.150**	0.155	0.131	**0.152**	**0.102**

TABLE II: Comparison of ICP based methods on the HeLiPR dataset.

Method	Highway 44		City 43		Curved Walls	
	RTE [m]	RRE [°]	RTE [m]	RRE [°]	RTE [m]	RRE [°]
ICP (point-to-point)	0.974	0.168	0.703	0.160	0.555	0.226
ICP (point-to-plane)	0.996	0.164	0.270	0.067	0.263	0.067
GICP	1.052	0.159	0.777	0.241	0.543	0.181
KISS-ICP	0.985	0.256	0.254	0.109	0.511	0.225
Doppler-ICP	1.258	0.797	0.858	0.210	**0.011**	**0.040**
Dynamic-ICP	**0.705**	**0.143**	**0.209**	**0.060**	0.063	0.061

TABLE III: Comparison of ICP based methods across sequences on AevaScenes dataset and DICP dataset.

Sequence	Dyn./All		Clustered/Dyn.	Refined/Dyn.
	Max	Mean	Mean	Mean
highway_day_1	35.7%	22.3%	71.1%	95.8%
city_day_1	33.9%	17.6%	70.3%	93.4%
Bridge 02	29.2%	3.9%	71.0%	89.6%
Street 01	16.8%	7.6%	77.4%	93.1%

TABLE IV: Dynamic-scene statistics on HeRCULES, HeLiPR, and AevaScenes. Dyn./All: dynamic candidates over all scan points; Clustered/Dyn.: clustered dynamics over dynamic candidates; Refined/Dyn.: validated dynamics used for prediction. Max/Mean are computed over frames in each sequence.

Method	River Island 01		Riverside 05		Highway 44	
	RTE [m]	RRE [°]	RTE [m]	RRE [°]	RTE [m]	RRE [°]
w/o VF	0.871	0.680	0.686	0.235	1.078	0.268
w/o DPP	0.411	0.346	0.633	0.237	0.899	0.172
w/o DR	0.900	0.552	0.595	0.154	0.971	0.161
Full	**0.379**	**0.189**	**0.592**	**0.150**	**0.705**	**0.143**

TABLE V: Ablation evaluation on Dynamic-ICP.

Method	Avg. iters ↓	Conv. rate ↑	FPS ↑
ICP (point-to-point)	22.94	98.00%	13.02
ICP (point-to-plane)	17.20	99.70%	**15.10**
GICP	23.65	98.90%	5.68
Doppler-ICP	13.54	**99.80%**	14.20
Dynamic-ICP	**13.50**	**99.80%**	13.34

TABLE VI: Runtime and convergence statistics on HeRCULES.

correspondence distance of 0.3 m for all methods). Overall, Dynamic-ICP converges in fewer iterations than classic ICP variants and similarly to Doppler-ICP, while achieving the highest throughput and matching the best convergence rate. These results indicate that Doppler cues and dynamic point prediction stabilize optimization, reduce iterations, and enable real-time performance without sacrificing reliability.

V. CONCLUSION

We presented Dynamic-ICP, a training-free, Doppler-aware registration framework that employs scene motion as an additional signal for scan registration. Our approach (i) estimates ego translational velocity from per-point Doppler and builds a velocity filter, (ii) clusters moving points with HDBSCAN and reconstructs object-wise 3D velocities, (iii) predicts dynamic points with a constant-velocity model, and (iv) aligns scans using a compact objective that combines point-to-plane geometric residual with a translation-invariant, rotation-only Doppler residual. Unlike Doppler-ICP variants that require sensor–vehicle extrinsic calibration, our approach operates without external sensors or calibration, relying solely on FMCW LiDAR range and Doppler.

Across three Doppler-capable datasets, HeRCULES, HeLiPR, and AevaScenes, Dynamic-ICP delivers state-of-the-art or competitive accuracy, particularly in high-speed, highly dynamic scenes and geometrically repetitive environments. Ablations confirm that the velocity filter, dynamic point prediction, and Doppler residual are complementary and together account for the largest gains in both RTE and RRE. To encourage community adoption and reproducibility, we release our implementation of Dynamic-ICP as open source.

References

[1] J. Behley and C. Stachniss. Efficient surfel-based slam using 3d laser range data in urban environments. In *Proc. of Robotics: Science and Systems (RSS)*, 2018.

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

[3] P. Biber and W. Straßer. The normal distributions transform: A new approach to laser scan matching. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2003.

[4] R.J. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[5] D. Casado Herraez, M. Zeller, L. Chang, I. Vizzo, M. Heidingsfeld, and C. Stachniss. Radar-only odometry and mapping for autonomous vehicles. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024.

[6] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Journal on Image and Vision Computing (IVC)*, 10(3):145–155, 1992.

[7] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek. The trimmed iterative closest point algorithm. In *2002 International Conference on Pattern Recognition*, volume 3, pages 545–548. IEEE, 2002.

[8] F. Ding, A. Palffy, D.M. Gavrila, and C.X. Lu. Hidden gems: 4d radar scene flow learning using cross-modal supervision. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[9] X. Gu, Y. Wang, C. Wu, Y.J. Lee, and P. Wang. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[10] D.C. Herraez, M. Zeller, D. Wang, J. Behley, M. Heidingsfeld, and C. Stachniss. Rai-slam: Radar-inertial slam for autonomous vehicles. *IEEE Robotics and Automation Letters (RA-L)*, 2025.

[11] B. Hexsel, H. Vhavle, and Y. Chen. Dicp: Doppler iterative closest point algorithm. *arXiv preprint arXiv:2201.11944*, 2022.

[12] M. Jung, W. Yang, D. Lee, H. Gil, G. Kim, and A. Kim. Helipr: Heterogeneous lidar dataset for inter-lidar place recognition under spatiotemporal variations. *Intl. Journal of Robotics Research (IJRR)*, 43(12):1867–1883, 2024.

[13] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer. Instantaneous ego-motion estimation using doppler radar. In *Proc. of the IEEE Intl. Conf. on Intelligent Transportation Systems (ITSC)*, 2013.

[14] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer. Instantaneous ego-motion estimation using multiple doppler radars. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2014.

[15] A. Khoche, Q. Zhang, Y. Cai, S.S. Mansouri, and P. Jensfelt. Dogflow: Self-supervised lidar scene flow via cross-modal doppler guidance. *arXiv preprint arXiv:2508.18506*, 2025.

[16] H. Kim, M. Jung, C. Noh, S. Jung, H. Song, W. Yang, H. Jang, and A. Kim. Hercules: Heterogeneous radar dataset in complex urban environment for multi-session radar slam. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2025.

[17] J. Kim, J. Woo, U. Shin, J. Oh, and S. Im. Flow4d: Leveraging 4d voxel network for lidar scene flow estimation. *IEEE Robotics and Automation Letters (RA-L)*, 2025.

[18] Y. Kittenplon, Y.C. Eldar, and D. Raviv. Flowstep3d: Model unrolling for self-supervised scene flow estimation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[19] K. Koide, M. Yokozuka, S. Oishi, and A. Banno. Voxelized gicp for fast and accurate 3d point cloud registration. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[20] X. Li, H. Zhang, and W. Chen. 4d radar-based pose graph slam with ego-velocity pre-integration factor. *IEEE Robotics and Automation Letters (RA-L)*, 8(8):5124–5131, 2023.

[21] Y. Lin and H. Caesar. Icp-flow: Lidar scene flow estimation with icp. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[22] X. Liu, C.R. Qi, and L.J. Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[23] M. Magnusson. *The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection*. PhD thesis, Örebro universitet, 2009.

[24] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[25] G.N. Narasimhan, H. Vhavle, K.B. Vishvanatha, and J. Reuther. Aevascenes: A dataset and benchmark for fmcw lidar perception, 2025.

[26] A. Nuchter, K. Lingemann, and J. Hertzberg. Cached kd tree search for icp algorithms. In *Proc. of international conference on 3-D Digital Imaging and Modeling*, 2007.

[27] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proc. of international conference on 3-D Digital Imaging and Modeling*, 2001.

[28] A. Segal, D. Haehnel, and S. Thrun. Generalized-icp. In *Proc. of Robotics: Science and Systems (RSS)*, 2009.

[29] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 722–729. IEEE, 1999.

[30] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss. Kiss-icp: In defense of point-to-point icp–simple, accurate, and robust registration if done the right way. *IEEE Robotics and Automation Letters (RA-L)*, 8(2):1029–1036, 2023.

[31] D. Wang, H. Haag, D.C. Herraez, S. May, C. Stachniss, and A. Nuechter. Doppler-slam: Doppler-aided radar-inertial and lidar-inertial simultaneous localization and mapping. *IEEE Robotics and Automation Letters (RA-L)*, 2025.

[32] D. Wang, M. Masannek, S. May, and A. Nüchter. Infradar-localization: single-chip infrared-and radar-based monte carlo localization. In *Proc. of the Intl. Conf. on Automation Science and Engineering (CASE)*, 2023.

[33] Y. Wu, D.J. Yoon, K. Burnett, S. Kammel, Y. Chen, H. Vhavle, and T.D. Barfoot. Picking up speed: Continuous-time lidar-only odometry using doppler velocity measurements. *IEEE Robotics and Automation Letters (RA-L)*, 2022.

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

[35] D.J. Yoon, K. Burnett, J. Laconte, Y. Chen, H. Vhavle, S. Kammel, J. Reuther, and T.D. Barfoot. Need for speed: Fast correspondence-free lidar-inertial odometry using doppler velocity. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2023.

[36] J. Zhang, S. Singh, et al. Loam: Lidar odometry and mapping in real-time. In *Proc. of Robotics: Science and Systems (RSS)*, 2014.

[37] Q.Y. Zhou, J. Park, and V. Koltun. Fast global registration. In *European conference on computer vision*, pages 766–782. Springer, 2016.

[38] Q.Y. Zhou, J. Park, and V. Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.