Supplementary Details
FS-DICP: A Fast and Simple Point-to-Point Iterative Closest Point algorithm with Doppler Velocity

Zhicong Xie, WeiXuan Zhu, Di Wang, Haozhi Huang*, Guanghua Yang*

School of Intelligent Systems Science and Engineering, Jinan University, Zhuhai, China.

Here we provide details on the **Explicit Assumptions in Odometry**, **Formal Derivation with Lemmas** and **Efficiency Analysis** of the proposed ICP algorithm in the main text. We further perform a rigorous efficiency comparison of our proposed approach against other SOTA methods to demonstrate that the accelerated computational performance of our method stems from a significantly higher convergence rate, for which we conduct an additional independent experiment to provide empirical verification of this key finding.

Explicit Assumption in Odometry

In the context of odometry, the observation scenario shifts from monitoring moving objects via a fixed sensor to perceiving a "moving world" from the reference frame of the sensor itself. For the case of going straight, utilizing the "spherical property" of background points with the "Inverse Rigid Body" assumption is the core principle behind Doppler-based Ego-Motion Estimation (often referred to as Instantaneous Odometry). The "Spherical Assumption" is an excellent approximation, where backgrounds are relatively stationary and "moving" straight act like perfect spheres in velocity space.

For the case of turning, Bruno Hexsel et al. [1] in their derivation (Eq. (11)–(12)) identifies a critical geometric simplification through **Orthogonality**. Here is the details of why the "Spherical Property" remains valid for ego-motion even when turning:

Firstly, the paper [1] rigorously proves that a Doppler sensor is "blind" to its own rotation on Page 4. Given equation 11 in Bruno Hexsel et al.[1] as follows, which defines the Doppler velocity ${}_L v_{LP} \in \mathbb{R}$ as the projection of the relative velocity vector along the look direction ${}_L\mathbf{d}_{LP}$:

$$ {}_L v_{LP} = -{}_L\mathbf{d}_{LP} \cdot ({}_L\mathbf{v}_L +_L \omega_{IL} \times_L \mathbf{t}_{LP}), $$

where ${}_L\mathbf{d}_{LP}$ is the look direction (unit vector), and ${}_L\mathbf{t}_{LP}$ is the position vector of the point. The look direction ${}_L\mathbf{d}_{LP}$ is perfectly parallel to the position vector ${}_L\mathbf{t}_{LP}$.

The rotational term ${}_L\mathbf{d}_{LP} \cdot ({}_L\omega_{IL} \times_L \mathbf{t}_{LP})$ corresponds to a scalar triple product, which vanishes since ${}_L\mathbf{d}_{LP}$ is parallel to ${}_L\mathbf{t}_{LP}$—a geometric condition that forces their cross product to be zero. Physically, this means the sensor's rotation creates purely tangential velocity relative to the background, i.e., the Doppler sensor measures radial velocity only, and the rotation of its own creates tangential velocity, which results in the case that the Doppler sensor measures 0 m/s from its own rotation.

Therefore, when the sensor rotates with angular velocity ${}_L\omega_{IL}$, the relative velocity of a static background point P at position ${}_L\mathbf{t}_{LP}$ is still:

$$ {}_L v_{LP} = -{}_L\mathbf{d}_{LP} \cdot_L \mathbf{v}_L. $$

This equation mathematically characterizes the "Spherical Property". It states that the measured Doppler velocity depends only on the sensor's instantaneous linear velocity. Therefore, even if the LiDAR is turning, the data points do form a perfect sphere, and the diameter of that sphere corresponds to the sensor's linear velocity.

Formal Derivation with Lemmas

Lemma 1 (**The Spherical Property**). *With the hypotheses of constant velocity, the vectorized Doppler velocities all lie on a single sphere passing through the origin.*

Proof. Constant velocity is also known as a Homogeneous Velocity Field, where objects are in a rigid-body state and exhibit no rotational motion. There are two methods to prove the claim: the first is using Thales' Theorem, and the second is using the definition of a sphere. We first prove the claim through Thales' Theorem in vector form.

Definitions. Let us define the following variables in a 3D cartesian coordinate system: $\vec{v}_{true}$ denotes the true, constant 3D velocity vector of the target; $\hat{n}$ denotes the unit direction vector (line of sight) from the LiDAR sensor to the target point; v_r denotes the scalar Doppler velocity measured by the FMCW LiDAR. This is the projection of the true velocity onto the line of sight.

The Doppler Measurement. The scalar radial velocity is given by the dot product:

$$v_r = \vec{v}_{true} \cdot \hat{n} \tag{1}$$

Vectorize the Doppler Velocity. The vectorized Doppler velocity $\vec{v}_r$ is obtained by multiplying the scalar measurement v_r with the direction vector $\hat{n}$:

$$\vec{v}_r = v_r \hat{n} = (\vec{v}_{true} \cdot \hat{n})\hat{n} \tag{2}$$

The Spherical Property (with Thales' Theorem). Consider a sphere S defined by the diameter vector $\vec{v}_{true}$ extending from the origin O. A point $\vec{P}$ lies on this sphere if and only if the vector from the origin to $\vec{P}$ (which is $\vec{v}_r$) is orthogonal to the vector from $\vec{P}$ to the tip of the diameter ($\vec{v}_{true} - \vec{v}_r$).

Mathematically, we must prove that the dot product is zero:

$$Q = \vec{v}_r \cdot (\vec{v}_{true} - \vec{v}_r)$$

Substitute the definition of $\vec{v}_r$ from the vectorized velocity (Eq.(2))

$$Q = [(\vec{v}_{true} \cdot \hat{n})\hat{n}] \cdot [\vec{v}_{true} - (\vec{v}_{true} \cdot \hat{n})\hat{n}]$$

Distribute the dot product:

$$Q = [(\vec{v}_{true} \cdot \hat{n})\hat{n} \cdot \vec{v}_{true}] - [(\vec{v}_{true} \cdot \hat{n})\hat{n} \cdot (\vec{v}_{true} \cdot \hat{n})\hat{n}]$$

Analyze the first term:

$$(\vec{v}_{true} \cdot \hat{n})(\hat{n} \cdot \vec{v}_{true}) = (\vec{v}_{true} \cdot \hat{n})^2$$

Analyze the second term:

$$(\vec{v}_{true} \cdot \hat{n})^2(\hat{n} \cdot \hat{n})$$

Since $\hat{n}$ is a unit vector, $\hat{n} \cdot \hat{n} = 1$. Thus, the second term is also $(\vec{v}_{true} \cdot \hat{n})^2$. And thus

$$Q = (\vec{v}_{true} \cdot \hat{n})^2 - (\vec{v}_{true} \cdot \hat{n})^2 = 0$$

Since the dot product is zero, the vector $\vec{v}_r$ is always orthogonal to the vector $(\vec{v}_{true} - \vec{v}_r)$. We obtain the claim.

Another way to prove the claim is to show all static points in the scene have a fixed distance between their true velocity and the vectorized velocity.

The Spherical Property (with Definition).

Based on the definitions of Doppler measurement (Eq.(1)) and vectorized velocity (Eq.(2)), the spherical constraint is established. , we need to prove that the distance between each vectorized velocity $\vec{v}_r$ and half of the true velocity $\vec{v}_{true}$ is uniform across all vectors, and that this uniform distance is equivalent to the norm of half the true velocity $\vec{v}_{true}$.

$$\begin{aligned}\left\|\vec{v}_r - \frac{1}{2}\vec{v}_{true}\right\|_2^2 &= (\vec{v}_r)^2 - \vec{v}_r \cdot \vec{v}_{true} + \left\|\frac{1}{2}\vec{v}_{true}\right\|_2^2 \\ &= (\vec{v}_{true} \cdot \hat{n})^2 - (\vec{v}_{true} \cdot \hat{n})^2 + \left\|\frac{1}{2}\vec{v}_{true}\right\|_2^2 \\ &= \left\|\frac{1}{2}\vec{v}_{true}\right\|_2^2,\end{aligned}$$

with the assumption of constant velocity, we obtain the claim. □

Corollary 1 (Geometric Separability of Static and Dynamic Points): *Let $\mathcal{P}$ be the set of observed points. The subset of static points $\mathcal{P}_{static} \subset \mathcal{P}$ (where $\mathbf{v}_p = \mathbf{0}$) corresponds to measurements lying on a sphere $\mathcal{S}_{static}$ centered at $\mathbf{c}_{static} = -\frac{1}{2}\mathbf{v}_{ego}$ with radius $r = \frac{1}{2}\|\mathbf{v}_{ego}\|$. Conversely, points belonging to a rigid body with translational velocity $\mathbf{v}_{obj} \neq \mathbf{0}$ lie on a distinct sphere $\mathcal{S}_{obj}$ centered at $\mathbf{c}_{obj} = \frac{1}{2}(\mathbf{v}_{obj} - \mathbf{v}_{ego})$. Thus, provided $\mathbf{v}_{obj} \neq \mathbf{0}$, the static and dynamic structures are geometrically distinct in the velocity domain. Furthermore, The translational velocity of a dynamic rigid body remains the translational velocity in the LiDAR coordinate system.*

Remark 1 (Practical Robustness): In real-world driving scenarios, static points (roads, buildings) constitute the dominant spherical structure. Although sensor noise and vibrations introduce bounded perturbations, this dominance allows robust estimation algorithms like RANSAC to identify the static sphere parameters $\mathbf{v}_{ego}$ by treating dynamic objects (the minor spherical clusters) as outliers.

Efficiency Analysis

We first present a detailed exposition of the **Theoretical Foundation underlying the 6-DoF to 3-DoF Reduction**. To rigorously verify that the superior computational performance of our proposed method originates not from lowering the per-iteration computational cost, but from drastically decreasing the total number of iterations necessary for convergence via dimensionality reduction, we not only provide an analysis of **Computational Complexity and Convergence Properties**, but also perform an additional validation **Experiment**. In this experiment, we compare FS-DICP with three state-of-the-art counterparts (DICP, Steam-DICP, and KISS-ICP) to substantiate the theoretical assertion.

Theoretical Foundation: 6-DoF to 3-DoF Reduction

Lemma 2 (Dimensionality Reduction via Kinematic Constraints): *Let $\mathcal{P}_{s-1}$ and $\mathcal{P}_s$ be consecutive point clouds acquired at timestamps $s-1$ and s, separated by the interval $\Delta s \in \mathbb{R}^+$. Let the sensor pose at time s be defined by $\mathbf{T}_s \in SE(3)$, comprising a rotation $\mathbf{R}_s \in SO(3)$ and translation $\mathbf{t}_s \in \mathbb{R}^3$. Given the instantaneous ego-velocity $\mathbf{V}_s \in \mathbb{R}^3$ measured in the sensor frame, and assuming constant velocity over Δs, the translation $\mathbf{t}_s$ is constrained by the kinematic update:*

$$\mathbf{t}_s = \mathbf{t}_{s-1} + \mathbf{R}_s(\mathbf{V}_s \Delta s)$$

This constraint establishes a functional dependency $\mathbf{t}_s(\mathbf{R}_s)$, thereby reducing the pose estimation problem from a joint optimization over the 6D manifold $SE(3)$ to a 3D optimization over $SO(3)$.

Proof. The objective function of the standard point-to-point ICP is to minimize the error metric between the source point cloud $\mathcal{P}_{s-1}$ and the target point cloud $\mathcal{P}_s$. Therefore, the optimization finds the optimal

Rotation $\mathbf{R}_s$ and Translation $\mathbf{t}_s$ that minimizes the residual:

$$\mathbf{R}_s^*, \mathbf{t}_s^* = \underset{\mathbf{R},\mathbf{t}}{\arg\min} \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}} \left\|\mathbf{R}\mathbf{p} + \mathbf{t} - \mathbf{q}\right\|_2$$

where $\mathbf{p} \in \mathcal{P}_{s-1}$ and $\mathbf{q} \in \mathcal{P}_s$ are corresponding points. In order to reduce the 6-DoF ICP optimization to a 3-DoF formulation under the velocity constraint, we prove that the objective function depends only on the rotation matrix $\mathbf{R}$ through the following 5 steps:

Step 1: Express Relative Pose in Sensor Frame. Let the world-to-vehicle transformation matrices be $\mathbf{T}_{s-1} = \begin{bmatrix}\mathbf{R}_{s-1} & \mathbf{t}_{s-1} \\ \mathbf{0}^\top & 1\end{bmatrix}$ and $\mathbf{T}_s = \begin{bmatrix}\mathbf{R}_s & \mathbf{t}_s \\ \mathbf{0}^\top & 1\end{bmatrix}$. The relative pose of frame $s-1$ expressed in the coordinate system of frame s (the superscript (s)) is given by:

$$\mathbf{T}_s^{(s)} = \mathbf{T}_s^{-1}\mathbf{T}_{s-1} = \begin{bmatrix}\mathbf{R}_s^\top \mathbf{R}_{s-1} & \mathbf{R}_s^\top(\mathbf{t}_{s-1} - \mathbf{t}_s) \\ \mathbf{0}^\top & 1\end{bmatrix}$$

Step 2: Relate Velocity to Displacement. The translational component of the relative pose, $\mathbf{R}_s^\top(\mathbf{t}_{s-1} - \mathbf{t}_s)$, represents the physical displacement from time s to $s-1$ as viewed in the sensor frame s (the superscript (s)). By definition, the ego-linear velocity $\mathbf{V}_s^{(s)}$ in the sensor frame is the negative time derivative of this displacement over the interval Δs. Assuming constant velocity:

$$\mathbf{V}_s^{(s)} = -\frac{\mathbf{R}_s^\top(\mathbf{t}_{s-1} - \mathbf{t}_s)}{\Delta s}$$

Step 3: Isolate the Translation Variable. We rearrange the velocity equation to solve for the current translation $\mathbf{t}_s$:

$$\begin{aligned}\mathbf{R}_s^\top(\mathbf{t}_{s-1} - \mathbf{t}_s) &= -\Delta s \cdot \mathbf{V}_s^{(s)} \\ \mathbf{t}_{s-1} - \mathbf{t}_s &= -\Delta s \cdot \mathbf{R}_s \mathbf{V}_s^{(s)} \\ \mathbf{t}_s &= \mathbf{t}_{s-1} + \Delta s \cdot \mathbf{R}_s \mathbf{V}_s^{(s)}.\end{aligned}$$

This equation establishes that $\mathbf{t}_s$ is no longer an independent variable; it is deterministically defined by the rotation $\mathbf{R}_s$ and the known previous state.

Step 4: Substitute into the Objective Function. We substitute the expression for $\mathbf{t}_s$ directly into the original 6-DoF ICP objective function. The term $\mathbf{t}$ in the residual is replaced by $\mathbf{t}_{s-1} + \Delta s \cdot \mathbf{R}_{s,i-1}^* \mathbf{V}_s^{(s)}$ from its previous iteration according to $\mathbf{R}_{s,i-1}^*$:

$$\mathbf{R}_{s,i}^* = \underset{\mathbf{R}\in SO(3)}{\arg\min} \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}} \left\|\mathbf{R}\mathbf{p} + \underbrace{(\mathbf{t}_{s-1} + \Delta s \cdot \mathbf{R}_{s,i-1}\mathbf{V}_s^{(s)})}_{\mathbf{t}_{s,i-1}} - \mathbf{q}\right\|_2,$$

Step 5: Dimensionality Reduction. The objective function now depends only on the rotation matrix $\mathbf{R}$. $\mathbf{t}_{s-1}$ and $\mathbf{R}_{s,i-1}^*$ are fixed constants from the previous iteration and the previous timestamp. Δs is a known time constant. $\mathbf{V}_s^{(s)}$ is estimated from the measurements provided by the Doppler sensor. Since $\mathbf{t}_s$ has been eliminated as an independent optimization variable, the search space is reduced from the 6-dimensional Special Euclidean group $SE(3)$ to the 3D Special Orthogonal group $SO(3)$. Therefore, the optimization reduces to a 3-DoF iterative scheme where only the rotation is optimized. Once $\mathbf{R}_s^*$ is found, the optimal translation $\mathbf{t}_s^*$ is recovered deterministically via the velocity constraint equation, and we then obtain the claim.

Computational Complexity and Convergence Properties

Following the 3-DoF reduction derived in Lemma 2, we analyze the specific computational advantages in terms of linear system solving and optimization stability. Regarding linear system complexity, in the Gauss-Newton optimization step, solving the linearized system for the increment $\Delta\mathbf{x}$ requires inversion of the approximate Hessian matrix $\mathbf{H}$. The update equation is given by:

$$(\mathbf{J}^\top\mathbf{J})\Delta\mathbf{x} = -\mathbf{J}^\top\mathbf{r}$$

where $\mathbf{H} = \mathbf{J}^\top \mathbf{J}$. In standard 6-DoF ICP, $\Delta\mathbf{x} \in \mathbb{R}^6$ (comprising 3 rotational and 3 translational components), resulting in a 6×6 Hessian matrix. In our proposed FS-DICP formulation, the state vector is reduced to $\Delta\mathbf{x} \in \mathbb{R}^3$ (rotation only), reducing $\mathbf{H}$ to a 3×3 matrix. While the asymptotic complexity of ICP is typically dominated by the nearest-neighbor search ($O(N \log N)$), the reduction in the dimensionality of the linear solve from $\mathbb{R}^6$ to $\mathbb{R}^3$ provides a constant-factor reduction in computational cost per iteration. More significantly, it reduces the memory footprint and numerical instability associated with inverting larger matrices. With respect to convergence rate, as the primary efficiency gain arises from the conditioning of the underlying optimization problem, standard ICP jointly optimizes rotation and translation, which represent inherently physically coupled state variables in ego-motion estimation. This coupling often leads to "zig-zag" convergence behavior, particularly in environments where geometric constraints on translation are weak (e.g., corridors or open roads). By enforcing the kinematic constraint $\mathbf{t}(\mathbf{R})$, we effectively eliminate the cross-coupling terms between rotation and translation in the Hessian. This theoretically implies that the optimization trajectory is more direct, requiring significantly fewer iterations to reach a minimum. This theoretical assertion that efficiency stems from iteration reduction rather than just per-iteration velocity is empirically verified in the following experiment.

Experiments and Analysis

To comprehensively analyze the computational efficiency of FS-DICP relative to state-of-the-art baselines (KISS-ICP, DICP, and Steam-DICP), we dissect the registration pipeline into three constituent components: preprocessing overhead, per-iteration computational cost, and total iteration count. This tripartite analysis elucidates how FS-DICP achieves superior end-to-end runtime through synergistic advantages across all stages. Additionally, to investigate the relationship between iteration count, optimization dimensionality, and velocity noise, we introduce a dedicated ablation experiment.

Table 1 (reproduced from Table II of the revised manuscript) summarizes the computational efficiency of all compared methods. First, FS-DICP achieves the lowest total runtime despite not having the minimal per-iteration cost; this advantage stems from a dramatic reduction in iteration count driven by dimensionality reduction to 3-DoF rotation-only optimization. Second, KISS-ICP exhibits the lowest per-iteration time due to its simplest residual formulation, but requires substantially more iterations in the unconstrained 6-DoF search space. Third, both DICP and Steam-DICP incur high per-iteration costs from point-to-plane residuals and Doppler consistency evaluation. Steam-DICP further requires per-point continuous-time trajectory interpolation during every iteration, making its per-iteration cost the highest among all methods and resulting in the longest total runtime.

All methods evaluated in this work process voxel-downsampled point clouds containing 2 000–10 000 points per frame. The total runtime comprises two distinct phases: (i) one-time preprocessing (e.g., spatial indexing, normal estimation, velocity computation) and (ii) iterative refinement, where each Gauss–Newton iteration evaluates residuals and Jacobian across all point correspondences. This per-iteration cost constitutes the dominant computational burden, scaling linearly with point count.

Table 1: Average registration time (ms) and iterations. The best results in **bold** and the second best in <u>underscored</u>.

	DICP	Steam-DICP	KISS-ICP	FS-DICP (Ours)
iterations	10.35	<u>10.00</u>	33.27	**6.65**
time (ms)	64.10	162.50	<u>26.33</u>	**14.25**
time/iter (ms)	6.19	16.25	**0.79**	<u>2.14</u>

Preprocessing Stage

All methods apply voxel-downsampled to reduce point clouds to several thousand points per frame. Point-to-plane formulations (Steam-DICP, DICP) require surface normal estimation for residual computation, necessitating k-d tree construction to support efficient local neighborhood queries for plane fitting. Point-to-point methods (KISS-ICP, FS-DICP) avoid both normal estimation and tree construction by utilizing hash-based voxel grids that enable constant-time neighborhood queries. FS-DICP additionally estimates ego-velocity from Doppler measurements and removes dynamic points based on velocity inconsistency prior to registration.

Per-Iteration Cost

Residual evaluation For point-to-point methods (KISS-ICP, FS-DICP), the residual is computed as the Euclidean distance $\mathbf{r} = \mathbf{R}\mathbf{p} + \mathbf{t} - \mathbf{q}$, requiring only vector subtraction after rigid transformation. For point-to-plane methods (Steam-DICP, DICP), the residual is the signed distance $\mathbf{r} = (\mathbf{R}\mathbf{p} + \mathbf{t} - \mathbf{q})^\top \mathbf{n}$, requiring an additional dot product with the precomputed normal vector per point. Both Steam-DICP and DICP evaluate Doppler consistency during each iteration by comparing the Doppler velocity derived from the current motion estimate against the measured value; points with small discrepancies are retained as inliers.

Jacobian computation KISS-ICP achieves minimal computational overhead through direct assignment of Jacobian entries from point coordinates without arithmetic operations. Steam-DICP and DICP exhibit substantially higher costs due to additional terms involving surface normals and Doppler velocity residuals. FS-DICP computes Jacobian with respect to only three rotational degrees of freedom; however, the embedding of displacement computation within the rotational parameterization introduces moderate complexity beyond KISS-ICP's direct assignment.

Additional processing Steam-DICP further incorporates continuous-time trajectory modeling, necessitating per-point timestamp interpolation to estimate instantaneous poses during every iteration, thereby introducing additional computational overhead beyond standard frame-to-frame registration.

Table 2: Ablation study on convergence efficiency: FS-DICP vs. 6-DoF ICP under varying velocity noise levels.

Velocity Source	Noise Level (m/s)	Method	Init. Strategy	Runtime (s)	Iter.	RPE Trans (m)
GT-derived	0.0	FS-DICP	constant-velocity	0.0111	1.6	0.0041
		FS-DICP	Velocity-aided	0.0113	1.6	0.0041
		6-DoF ICP	constant-velocity	1.0480	23.6	0.1539
		6-DoF ICP	Velocity-aided	1.0356	23.4	0.1535
Noisy Estimate	0.3	FS-DICP	constant-velocity	0.0114	1.6	0.0304
		FS-DICP	Velocity-aided	0.0115	1.6	0.0289
		6-DoF ICP	constant-velocity	1.0497	23.6	0.1539
		6-DoF ICP	Velocity-aided	1.0531	23.6	0.1534
Noisy Estimate	0.5	FS-DICP	constant-velocity	0.0115	1.6	0.0462
		FS-DICP	Velocity-aided	0.0117	1.7	0.0476
		6-DoF ICP	constant-velocity	1.0428	23.6	0.1539
		6-DoF ICP	Velocity-aided	1.0731	24.1	0.1549
Noisy Estimate	2.0	FS-DICP	constant-velocity	0.0120	1.8	0.1933
		FS-DICP	Velocity-aided	0.0159	2.5	0.1927
		6-DoF ICP	constant-velocity	1.0499	23.6	0.1539
		6-DoF ICP	Velocity-aided	1.2332	27.3	0.1606
Noisy Estimate	3.0	FS-DICP	constant-velocity	0.0124	1.9	0.2807
		FS-DICP	Velocity-aided	0.0188	3.1	0.2863
		6-DoF ICP	constant-velocity	1.0455	23.6	0.1539
		6-DoF ICP	Velocity-aided	1.3706	29.5	0.1649

Convergence Rate Analysis

To quantitatively isolate the factors governing convergence speed, we conduct controlled ablation studies comparing FS-DICP against standard 6-DoF ICP. The experiment systematically investigates two key mechanisms: (i) the impact of initialization quality by evaluating both constant-velocity initialization and velocity-aided initialization under varying levels of velocity noise (0.0, 0.3, 0.5, 2.0, and 3.0 m/s), and (ii) the impact of optimization dimensionality by comparing 3-DoF rotation-only optimization (FS-DICP) versus full 6-DoF pose optimization (standard ICP). Specifically, the *constant-velocity* strategy computes an initial pose estimate from the motion between the previous two frames under a constant-velocity assumption, while the *velocity-aided* strategy further refines the translation component using Doppler-derived ego-velocity with controlled noise injection. Detailed results are presented in Table 2, where we evaluate on 10 test sequences, sampling 5 frame pairs from each sequence, with 20 independent trials conducted per pair under each experimental setting.

Experiment results in Table 2 demonstrate that FS-DICP requires substantially fewer iterations than 6-DoF ICP across all tested conditions. Under noise-free conditions, FS-DICP converges in approximately 1.6 iterations versus 23.6 iterations for 6-DoF ICP. Critically, this advantage persists under strong velocity noise (3.0 m/s), where FS-DICP still requires fewer than 2 iterations while 6-DoF ICP exceeds 23 iterations regardless of initialization strategy. This confirms that dimensionality reduction to 3-DoF rotation—enabled by embedding ego-velocity constraints—is the dominant factor driving iteration reduction. Within the same optimization dimensionality and test scenario, more accurate initialization (velocity-aided versus constant-velocity) yields modest further reductions in iteration count, particularly under moderate noise levels. In contrast, KISS-ICP operates in an unconstrained 6-DoF space without motion priors, resulting in the highest iteration count among all methods. The convergence acceleration of FS-DICP fundamentally stems from principled dimensionality reduction to 3-DoF rotation, with accurate velocity estimation providing supplementary benefits through high-quality translation initialization.

Summary

FS-DICP achieves the shortest end-to-end runtime by synergistically combining three advantages. First, 3-DoF rotation-only optimization—enabled by ego-velocity constraints—reduces iteration count by approximately 80% compared to 6-DoF ICP, fundamentally accelerating convergence. Second, the point-to-point residual formulation simplifies per-point computation to pure vector subtraction, avoiding surface normal estimation and extra dot products required by point-to-plane methods. Third, although FS-DICP's Jacobian computation incurs modest overhead compared to KISS-ICP's direct assignment due to rotation-translation coupling, its computational cost remains substantially lower than that of DICP and Steam-DICP. The one-time cost of ego-velocity estimation is fully compensated by the dramatic iteration reduction and simplified per-iteration operations, yielding superior overall efficiency.

References

[1] B. Hexsel, H. Vhavle, and Y. Chen, "DICP: Doppler Iterative Closest Point Algorithm," in *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022.