IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

The More The Better? Confidence-driven Residual Weighting and Depth Fusion for Multi-RGB-D Inertial Odometry

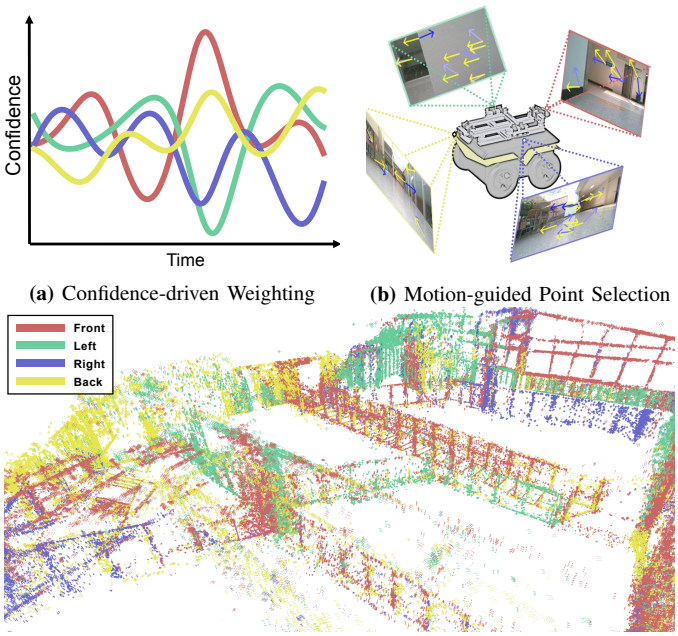
Seungsang Yun[1], Jaeho Shin[1], Jaekwang Cha[2] and Ayoung Kim[1*]

***Abstract*—Multi-camera systems hold considerable promise for enhancing visual odometry by expanding the field of view, yet simply adding more cameras does not guarantee higher accuracy. Because increasing the number of cameras also raises the likelihood of degraded or misaligned views, appropriate handling is essential to prevent severe outliers and corrupted global pose estimates. Previous methods discard points in back-end optimization based on residuals, which has been a bottleneck for real-time performance since erroneous measurements are inevitably incorporated into the main pipeline before removal. In response, we propose a direct *Multi-RGB-D Inertial Odometry* framework driven by *confidence-based weighting*, which adaptively down-weights unreliable cameras based on photometric quality and viewpoint alignment. To manage the heavy data load typical of multi-camera setups, we also incorporate a *motion-guided selection* strategy, filtering out non-informative points before costly alignment. This early pruning reduces computation yet retains critical constraints for odometry. By combining these techniques, our system achieves robust, scale-consistent pose estimation in real time, even with four cameras, as validated through challenging indoor-outdoor experiments involving saturation, occlusions, low-light conditions, and severe glare. We publicly release our multi-RGB-D-inertial dataset at https://github.com/seungsang07/multi-rgbd-inertial-dataset.**

***Index Terms*—RGB-D Inertial Odometry, Multi-camera SLAM**

I. Introduction

MULTI-camera systems offer a significantly wider field of view (FOV), potentially enhancing odometry by capturing richer environmental details than single-camera setups. However, fully exploiting this broader coverage remains challenging, as simply adding more cameras does not guarantee higher accuracy; indeed, a severely degraded camera can introduce outliers that corrupt the global solution. Existing methods often discard points with large re-projection errors to filter out outliers [1], but such retrospective filtering alone cannot explicitly address sensor degradation, resulting in a computational burden as outliers are incorporated throughout the entire pipeline. We instead propose a confidence-driven weighting mechanism, rooted in Entropy-Weighted Gradient (EWG) [2], which emphasizes well-exposed, texturally rich areas and also accounts for each camera's alignment with the motion. This approach dynamically assesses camera reliability, down-weighting degraded images to ensure robust odometry (see example in Fig. 1(a)). Despite confidence-driven weighting, multi-camera systems must handle large volumes of data, posing a risk to real-time performance. Some methods mitigate back-end load by selecting Hessian sub-blocks [3] or pruning high-error features, yet the large number of points often demands extra front-end operations (e.g., feature matching). Moreover, purely error-based rejection may fail to remove true outliers in the case of large motions. To address these challenges, we introduce a *motion-guided selection* step, as shown in Fig. 1(b), that discards non-informative or misaligned points before costly operations. This early filtering produces a smaller, odometry-effective subset that meets real-time demands without compromising performance in the multi-camera system. By combining these strategies with multiple RGB-D cameras and inertial data, our method delivers resilient, real-time motion tracking-even with four cameras. In diverse scenarios, confidence-driven weighting improves odometry reliability, while motion-guided selection keeps the four-camera pipeline real-time at 27 Hz from 30 Hz inputs. The setup further enables dense 3D mapping, as shown in Fig. 1(c).



(a) Confidence-driven Weighting **(b)** Motion-guided Point Selection

(c) Reconstructed Map from Multi-camera

Fig. 1: (a) Confidence-driven weighting scales each camera's residual contribution according to its reliability, reducing the impact of degraded views. (b) Motion-guided point selection removes non-informative pixels at an early stage. (c) A 3D map reconstructed from multiple cameras, with distinct colors representing individual camera IDs.

Manuscript received: March 23, 2025; Revised: June 17, 2025; Accepted: August 4, 2025.

This paper was recommended for publication by Editor Javier Civera upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported in part by Hyundai Motor Company and Kia and the Technology Innovation Program (1415187329, 20024355, Development of autonomous driving connectivity technology based on sensor-infrastructure cooperation) funded by MOTIE, Korea.

[1] S. Yun, J. Shin and A. Kim are with the Dept. of Mechanical Engineering, SNU, Seoul, S. Korea [seungsang, leah100, ayoungk]@snu.ac.kr

[2] J. Cha is with Hyundai Motor Company, S. Korea jaekwang@hyundai.com

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

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Our contributions are summarized as follows:

- **Confidence-driven Camera Weighting:** A scheme that assesses photometric reliability of multi-camera images and viewpoint alignment, robustly down-weighting sensors that are photometrically degraded or unfavorably aligned with the robot's motion.
- **Motion-guided Point Selection:** A front-end strategy that prunes non-informative points in multi-camera setups, reducing computation, preserving constraints, and maintaining real-time performance-its efficacy is also mathematically validated.
- **Extensive Experiments:** We verify our system across diverse environments. We analyze how multi-camera placements affect coverage and accuracy, offering insights into optimal camera configurations.
- **Dataset Release:** To the best of our knowledge, no publicly available multi-RGB-D odometry dataset exists; we therefore release our own to support research in multi-camera systems.

II. Related Work

A. Visual Odometry

Visual odometry can be categorized into two groups. **Feature-based approaches** such as *ORB-SLAM3* [4] and *PTAM* [5] detect and match features across frames, then minimize re-projection errors to estimate camera motion. They often excel in highly textured regions, but as the number of cameras increases, the growing volume of feature correspondences raises front-end computation. Moreover, in low-texture or dark environments, insufficient feature extraction limits performance. On the other hand, **direct methods**, such as *DSO* [6] and *LSD-SLAM* [7], bypass explicit feature matching by minimizing photometric errors, often achieving better performance in low-texture settings. Nonetheless, these methods can be sensitive to illumination changes and still suffer from scale ambiguity when used in a monocular configuration, necessitating additional sensor fusion to overcome this limitation.

B. Visual-inertial and RGB-D Integration

Monocular odometry inherently suffers from scale ambiguity, as trajectories can only be estimated up to an unknown scale. Fusing visual data with an inertial measurement unit (IMU) provides metric information to resolve this ambiguity while suppressing scale drift, as demonstrated by DM-VIO [8], VINS-Mono [9], and ORB-SLAM3 [4]. To address scale ambiguity, RGB-D cameras have also been integrated into various visual state-estimation frameworks [10, 11, 12, 13], leveraging synchronized color and depth data for robust pose tracking and mapping. However, these approaches typically rely on a single RGB-D camera, which limits the FOV and increases vulnerability to sensor degradation. In contrast, our multi-RGB-D inertial system uses non-overlapping viewpoints and selectively incorporates data from each camera to mitigate such degradation.

C. Multi-camera Approaches

Multi-camera systems can employ either overlapping fields of view for triangulation [14, 15] or non-overlapping configurations to maximize coverage [16, 1]. Although overlapping cameras simplify scale recovery by matching points in shared regions, they also increase front-end matching costs. In contrast, non-overlapping setups [17] reduce matching overhead but rely solely on the IMU for scale and are more susceptible to sensor degradation; in a multi-RGB-D system, however, independent depth estimation for each camera makes this configuration efficient for maximizing FOV with metric scale.

To reduce computational overhead in multi-camera VIO, some works perform *Hessian sub-selection* or feature budgeting to keep only informative Jacobian sub-blocks [3], while others apply *reprojection-error-based filtering* to discard high-error points during back-end optimization [1]. However, such retrospective filtering still incurs front-end overhead.

In this paper, we address these issues by (i) adaptively weighting each camera's residuals based on *photometric confidence* and *motion-aware weighting*, and (ii) pruning non-informative points at an early stage. This combination sustains real-time performance and ensures robust pose estimation, even with four non-overlapping RGB-D cameras in challenging scenarios.

III. Method

Our Multi-RGB-D Inertial Odometry pipeline (Fig. 2) unifies depth-based initialization, motion-guided point selection, confidence-driven multi-camera weighting, and IMU fusion, achieving robust real-time performance under challenging conditions. First, we initialize 3D points from synchronized RGB-D streams. Next, a motion-guided strategy discards non-informative pixels at an early stage. Each camera is then adaptively weighted by photometric reliability and optical-axis alignment. Finally, the refined data is fused with pre-integrated IMU measurements via photometric bundle adjustment.

A. Initialization with Depth Measurements

During the initialization stage, we incorporate both photometric and depth measurements to define the error function $E_{i,j}$ between keyframes i and j. Specifically, we combine the intensity residual $\left(r_{\text{photo}}\right)$ and the depth residual $\left(r_{\text{depth}}\right)$ into a single cost, adapting a delayed marginalization strategy similar to [8] for more stable initialization:

$$E_{i,j} = \sum_{k=1}^{n_c} \sum_{\substack{(u,v)\in P_i^k \\ D_i(u,v)\text{ is valid}}} \Big[\underbrace{\lambda_k \left\| I_i(u,v) - a_k I_j\left(u',v'\right) - b_k \right\|_\gamma}_{r_{\text{photo}}} + \underbrace{\left\| D_i(u,v) - D_j\left(u',v'\right) \right\|_\rho}_{r_{\text{depth}}} \Big], \tag{1}$$

where n_c is the number of cameras, and $(u,v) \in P_i^k$ indicates valid pixels from camera k in keyframe i. $I_i(u,v)$ and $I_j(u',v')$ are pixel intensities, with (u',v') the projection of (u,v) under the relative pose. $D(\cdot)$ denotes depth. Parameters a_k and b_k compensate for affine brightness changes, and $\|\cdot\|_{\gamma,\rho}$ is a robust norm for both photometric and depth errors.

Here, λ_k is a per-camera factor indicating how strongly camera k's photometric residual contributes relative to others (see Section III-C). Minimizing $E_{i,j}$ yields initial camera poses and 3D point depths, leveraging RGB-D range data while discarding invalid depths.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

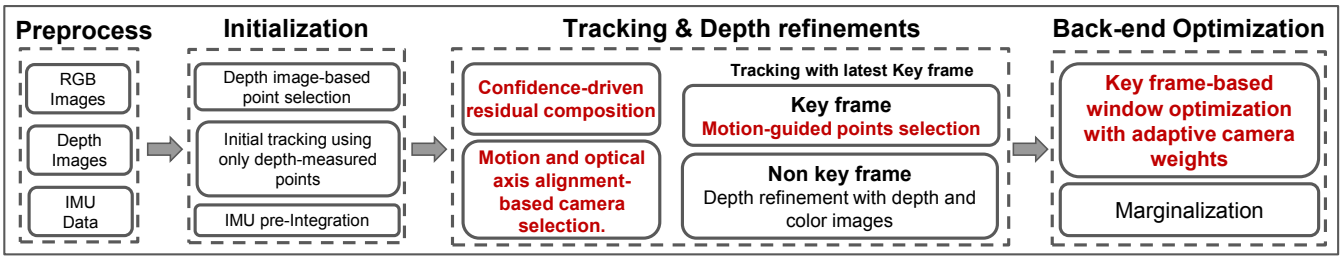


Fig. 2: Algorithm Flowchart of Our Multi-RGB-D Inertial Odometry. Multi-camera color and depth inputs first pass through an *initialization stage*, where valid depth points are selected for a coarse pose estimate based on *confidence-driven residuals*. Next, the *tracking and depth refinement* stage applies *motion-guided point selection* and per-camera *confidence weighting* to build a multi-camera photometric residual, which is then fused with IMU pre-integrations in the back end to perform photometric bundle adjustment. *Throughout both stages, points outside the depth validity range or with low confidence are filtered out early, improving efficiency and robustness.* The red modules highlight our main contributions: early point selection for efficiency and confidence-driven weighting for robust multi-camera odometry.

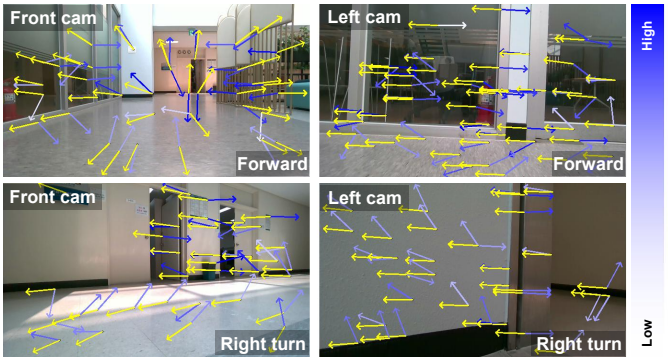


Fig. 3: Motion-Guided Point Selection. Two scenarios are shown: (*top*) forward motion with the front and left cameras, (*bottom*) a right turn with the front and left cameras. Yellow arrows represent estimated motion vectors, and the white-blue color scale indicates each point's importance score, with blue denoting stronger gradient-motion alignment. Even textureless surfaces may yield valid points if properly aligned. A final grid-based filter ensures an efficient, well-distributed set.

B. Motion-guided Point Selection

Given the data from multi-camera, distilling odometry-informative points is crucial for computational efficiency, particularly in direct odometry, where inter-frame intensity minimization depends on each local gradient's alignment with the robot's motion (e.g., yaw rotations favor horizontal edges). Hence, we propose *Motion-Guided Point Selection*, which combines motion direction with local gradients to identify flow-aligned points, yielding fewer but more informative features that improve convergence.

1) Influence of Motion Vector and Gradient on Residual Convexity: In this section, we first present a mathematical justification for selecting points based on the alignment of local gradients and motion-induced pixel velocity. By incorporating only informative points for camera pose estimation, we improve the convergence of optimization for the direct method and regulate the number of feature points, making our algorithm computationally efficient for multi-camera setups. We simplify our proof by defining the intensity difference minimization problem for a single point as follows:

$$\mathscr{E}(\boldsymbol{\xi}) = I_2(\mathbf{p}(\boldsymbol{\xi})) - I_1(\mathbf{p}_0), \tag{2}$$

where $\boldsymbol{\xi}$ parameterizes the Lie algebra of camera motion, and $\mathbf{p}$ denotes the tracked pixel point in the second image, represented by the camera motion. Iterative optimization algorithms for pose estimation require finding a small increment $\delta\boldsymbol{\xi}$ which minimizes a total photometric error $\frac{1}{2}(\mathscr{E}(\boldsymbol{\xi}_0 + \delta\boldsymbol{\xi}))^2$, where $\boldsymbol{\xi}_0$ denotes the current linearization point. Differentiating the error with respect to $\delta\boldsymbol{\xi}$ yields the following linear equation:

$$(\frac{\partial \mathscr{E}}{\partial \mathbf{p}}\frac{\partial \mathbf{p}}{\partial \delta\boldsymbol{\xi}})^\top(\frac{\partial \mathscr{E}}{\partial \mathbf{p}}\frac{\partial \mathbf{p}}{\partial \delta\boldsymbol{\xi}})\delta\boldsymbol{\xi} = -(\frac{\partial \mathscr{E}}{\partial \mathbf{p}}\frac{\partial \mathbf{p}}{\partial \delta\boldsymbol{\xi}})^\top\mathscr{E}(\boldsymbol{\xi}_0), \tag{3}$$

where $\frac{\partial \mathscr{E}}{\partial \mathbf{p}} = \nabla I_2(\mathbf{p}(\boldsymbol{\xi}_0))$ from Eq. 2.

When the estimated solution $\hat{\delta\boldsymbol{\xi}}$ from the constant velocity assumption is close to the null space of $\nabla I_2(\mathbf{p}(\boldsymbol{\xi}_0))\frac{\partial \mathbf{p}}{\partial \delta\boldsymbol{\xi}}$, which occurs when the estimated pixel velocity $\Delta\hat{\mathbf{p}} = \frac{\partial \mathbf{p}}{\partial \delta\boldsymbol{\xi}}\hat{\delta\boldsymbol{\xi}}$ is nearly orthogonal to the $\nabla I_2(\mathbf{p}(\boldsymbol{\xi}_0))$, the system (Eq. 2) becomes ill-conditioned, and the actual solution may have drastic perturbations during iterative optimization due to small changes in the Jacobian. Therefore, we expect measurements with a small $\nabla I_2(\mathbf{p}(\boldsymbol{\xi}_0))\Delta\hat{\mathbf{p}}$ to be less informative and filter out such measurements from the entire system. An example of this strategy is illustrated in Fig. 3, where points located at the boundary of the white and black wall are filtered out when the robot turns right.

2) Point Selection With Motion Vectors: The following describes how motion vectors and image gradients are used to score candidate points for motion-aware selection. To ensure spatial diversity, each input image is divided into a regular $M \times N$ grid ($M = N = 32$). Within each grid cell, we retain only those pixels that (i) have a valid depth measurement and (ii) exhibit a gradient magnitude at least 20% higher than the mean gradient over the entire image. The surviving pixels form the candidate set $\mathscr{C}$. For each candidate point $D_t(u_t, v_t)$ in frame t, we back-project it to 3D using its depth. Assuming constant velocity, we estimate its 3D position in frame $t+1$, re-project it onto the image plane, and define the resulting 2D displacement as the motion vector $(\Delta u, \Delta v)$. To assess the point's informativeness for direct odometry, we evaluate the alignment between this motion vector and the point's intensity gradient. We then compute a score M_i for each candidate $i \in \mathscr{C}$:

$$M_i \;=\; \|\nabla I_i\| \cos(\theta_i)\, G(d_i),$$

where $\|\nabla I_i\|$ is the local gradient magnitude, $\cos(\theta_i)$ measures alignment between the motion vector and the gradient direction, and $G(d_i)$ softly penalizes deviations from the constant-velocity model. Finally, the candidates are ranked by M_i, and the top n=100 points from each grid cell are retained

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

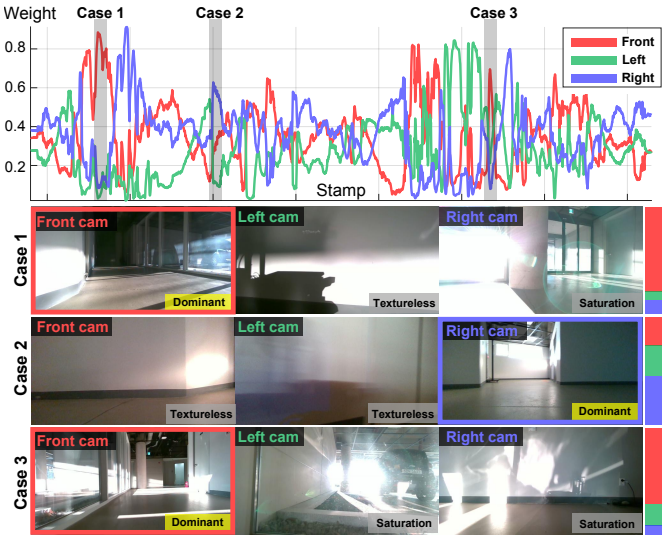


Fig. 4: Confidence-Driven Camera Weighting. Weights for front (red), left (green), and right (blue) cameras over time in the *txl.* sequence. *Case 1:* The front camera retakes top weight as the left sees blank regions and the right suffers saturation. *Case 2:* The right dominates when it observes richer textures. *Case 3:* Both left and right degrade, letting the front camera regain the highest weight.

for the odometry pipeline. Fig. 3 shows an example of the selected points, with the colorbar on the right visualizing the corresponding M_i scores. This process effectively selects odometry-relevant points from multi-camera images.

C. Confidence-Driven Weight-Based Multi-Camera Tracking

We extend the error (Eq. 1) across multiple cameras in a confidence-driven manner. Building on EWG [2], we assign each camera a *photometric confidence* that penalizes saturated or uniform regions yet preserves well-exposed areas, quantifying how *odometry-informative* each view is. Specifically, EWG is used to compute a scalar weight for each camera by aggregating pixel-wise entropy and gradient statistics, enabling camera-wise confidence assessment in the multi-camera setting. We also add a motion-aware factor, aligning each camera's optical axis with the robot's velocity to up-weight closer views.

*1) **Entropy-based Score for Multi-camera Confidence**:* In multi-camera setups, sensors often face distinct lighting or texture conditions, making it necessary to down-weight degraded cameras for reliable odometry. We build upon the method in [2] to compute camera-level photometric weights. For each pixel (u,v), we measure the gradient magnitude $\|\nabla I(u,v)\|^2$, local entropy $H(u,v)$, and a binary saturation mask $M(u,v) \in \{0,1\}$, and define an average gradient G_{avg} over the entire image. We then compute a local reliability score $u(u,v)$ as follows:

$$u(u,v) = W(u,v)\,[\|\nabla I(u,v)\|^2 + \pi(H(u,v))\,M(u,v)\,G_{\text{avg}}]. \quad (4)$$

Here, $W(u,v)$ is an entropy-based weight, and $\pi\big(H(u,v)\big)$ penalizes pixels with very low local entropy. Next, we sum $u(u,v)$ over all valid pixels in camera i to obtain a photometric score C^i_{photo}. Normalizing it among all cameras yields a per-camera weight:

$$C^i_{\text{photo}} = \sum_{(u,v)} u(u,v), \quad w^i_{\text{photo}} = \frac{C^i_{\text{photo}}}{\sum_{j=1}^{n_c} C^j_{\text{photo}}}. \quad (5)$$

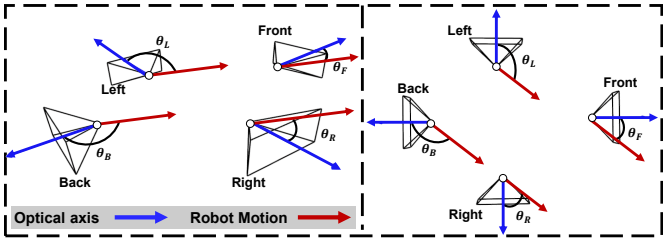


Fig. 5: Optical-Axis Alignment with Robot Motion. The robot's motion direction is shown in red and each camera's optical axis in blue. Angles θ_F, θ_L, θ_R, and θ_B measure the offset from the velocity vector, with smaller angles indicating stronger alignment and thus a higher weight.

Here, n_c denotes the total number of cameras in the rig, and w^i_{photo} is the normalized photometric confidence assigned to camera i. Figure 4 depicts how these weights evolve over time, with sharp drops coinciding with intervals of saturation or texture-less. In this way, our framework adaptively down-weights degraded images, ensuring that more reliable viewpoints dominate in the multi-camera odometry solution.

*2) **Motion-aware Weighting**:* When a camera's optical axis aligns with the robot's forward motion, the front view's parallax accumulates largely with track length, giving it inherently stronger depth constraints than side or backward views. As detailed in the supplementary material, although side-view cameras often provide larger per-frame parallax, the longer temporal persistence of points in the front view can yield greater benefits for odometry.

Hence, along with exposure and gradient quality, we also consider each camera's viewing direction (Fig. 5). Let $\mathbf{v}_r \in \mathbb{R}^3$ be the robot's velocity in the camera frame and $\mathbf{n}_i \in \mathbb{R}^3$ the optical axis of camera C_i. We compute their normalized dot product:

$$\delta_i = \max(-1, \min(1, \hat{\mathbf{v}}_r \cdot \hat{\mathbf{n}}_i)), \quad \hat{\mathbf{v}}_r = \frac{\mathbf{v}_r}{\|\mathbf{v}_r\|}, \quad \hat{\mathbf{n}}_i = \frac{\mathbf{n}_i}{\|\mathbf{n}_i\|}. \quad (6)$$

Cameras with higher δ_i yield more reliable measurements. To map δ_i to $[0,1]$ we apply

$$w^i_{\text{dir}} = \alpha_{\min} + \big(1 - \alpha_{\min}\big)\left[\tfrac{1+\delta_i}{2}\right]^{\beta}, \quad (7)$$

where $\alpha_{\min} > 0$ ensures that no camera is completely dropped, even if others fail, and β penalizes large angular offsets to reduce the impact of poorly aligned views. The 5 % floor $\alpha_{\min} = 0.05$ avoids rank deficiency when all well-aligned cameras degrade and preserves side-view constraints, keeping tracking alive even for nearly blind cameras. This ensures that even backward or side-facing views, despite contributing less in direct odometry, still provide meaningful geometric constraints. In backward-facing views, scene points move farther from the camera, reducing the effectiveness of depth refinement in direct methods. We set $\beta = 0.7$ to halve the weight of side views while keeping about 60% for moderately aligned ones, which yielded the best performance and ensured symmetry across all viewing directions.

*3) **Final Weighted Residual**:* Each camera compose a photometric error E^i_{photo} and a depth error E^i_{depth}. We combine the photometric weight w^i_{photo} and directional weight w^i_{dir} in the multi-camera residual:

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

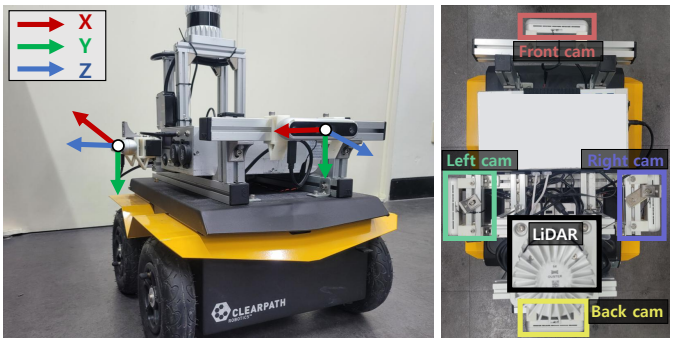


Fig. 6: The sensor system mounted on the Jackal mobile robot platform consists of four RealSense D455 RGB-D cameras, a Microstrain 3DM-GX5-25 IMU, and a Light Detection and Ranging (LiDAR) for ground truth acquisition. The four cameras are arranged with no overlapping FOV, and the LiDAR is used for making ground truth.

$$E_{\text{multiCam}} = \sum_{i=1}^{n_c} \left[\left(w^i_{\text{photo}} E^i_{\text{photo}} + E^i_{\text{depth}} \right) \times w^i_{\text{dir}} \right]. \quad (8)$$

Hence, each camera's residual is first scaled by photometric confidence, then adjusted by depth error and motion alignment, reducing the influence of misaligned or inaccurate views. In addition to maintaining consistent results for any heading, the system dynamically weights cameras according to their viewing direction. When the robot turns, the camera with the most favorable baseline is emphasized, preserving accuracy under diverse motion trajectories.

D. Back-end Optimization with IMU Integration

We refine pose and depth via keyframe-based optimization, fusing visual and pre-integrated IMU residuals [18]. The total error function combines visual and inertial terms:

$$E_{\text{total}} = E_{\text{multiCam}} + \lambda_{\text{IMU}} E_{\text{IMU}}, \quad (9)$$

where E_{multiCam} is the visual residual, E_{IMU} the inertial one, and λ_{IMU} is a regularization term that balances the contribution of the IMU residual relative to the visual residual. By integrating IMU data, our system maintains robust pose estimation even under challenging conditions. We empirically fix this weight to $\lambda_{\text{IMU}} = 0.12$, which yields a good trade-off between visual and inertial information.

IV. Experiment

A. Experimental Setup

We evaluate our Multi-RGB-D Inertial Odometry system on a mobile robot (Fig. 6) equipped with four Intel RealSense D455 RGB-D cameras mounted in rigid, non-overlapping directions (front, left, right, back), a Microstrain 3DM-GX5-25 IMU, and a LiDAR for ground-truth via LiDAR-Inertial SLAM. The RGB-D cameras are hardware-synchronized to ensure consistent multi-view capture. In addition, to obtain accurate extrinsic calibration in the non-overlapping setup, we used a large ArUco marker board and performed optimization to minimize the re-projection error of points observed within each camera's partial field of view. All experiments were performed on a computer equipped with an Intel® Core™ i9 CPU at 2.50 GHz and 64 GB RAM.

B. Datasets

The diverse sequences, which span a wide range of lighting and occlusion conditions, are summarized in Table. I; Fig. 7

TABLE I: Example table for visual conditions in different sequences. The abbreviations used are: Dur (Duration), Sat (Saturation), Ref (Reflection), Glr (Glare), Drk (Dark), Txl (Textureless), Occ (Occlusion).

Sequence	Length	Dur.	Sat	Ref	Glr	Drk	Txl	Occ
`Indoor`	100.0m	104s		O				
`Indoor2`	69.107m	159s		O	O			
`Glare`	44.81m	155s			O			
`Txl.`	67.61m	150s	O	O		O	O	
`Occ.`	67.05m	146s						O
`Dark`	72.68m	169s				O	O	

complements this overview with front-view snapshots of the four sequences, providing a general visual impression of their environments. Furthermore, as far as we know, no publicly available dataset offers multi-RGB-D inertial data, so we recorded our own using the sensor setup shown in Fig. 6. We release all raw sensor data and LiDAR-based ground truth to support research in multi-RGB-D odometry. The primary attributes of our dataset are as follows:

- **Indoor (Indr. / Indr.2):** `Indr.` is a corridor-like environment with moderate textures, while `Indr.2` provides an open indoor space with widely distributed features.
- **Glare:** Strong sunlight reflections cause severe saturation, especially on side cameras than the front camera.
- **Txl.:** A long corridor with minimal textures (mostly white walls), illuminated by bright sunlight from the windows.
- **Occ.:** Front camera occlusion in part of the sequence tests odometry robustness under temporary visual loss.
- **Dark:** A predominantly dark sequence featuring large rotations in a textureless environment.

C. Baseline Methods

We compare our approach (**Ours**) against three representative odometry and SLAM pipelines. First, **ORB-SLAM3** [4] is a feature-based visual-inertial simultaneous localization and mapping (SLAM) system that we execute in its official *RGB-D IMU* mode. Second, **VINS-RGBD** [12] integrates RGB, depth, and IMU, extending the VINS-Mono [9]. Lastly, **MCVIO** [17] is, as far as we know, the only publicly available method designed for multi-camera VIO with non-overlapping FOV. For a fair comparison with our Multi-RGB-D approach, we apply Sim(3) alignment to the MCVIO results.

D. Quantitative Results

Table II reports the absolute trajectory errors (ATE_r, ATE_t) for each sequence. We evaluate our method and baselines under one- to four-camera settings, where `F`, `L`, `R`, and `B` denote front, left, right, and back cameras, respectively. Fig. 8 illustrates the trajectories for each sequence along with major events occurring during the runs.

a) Indr.: This indoor sequence poses moderate complexity, where a single front camera (`F`) yields lower translation accuracy but better rotation accuracy compared to ORB-SLAM3 and VINS-RGBD. Nevertheless, adding more cameras yields nearly linear improvements: moving from front–left (`FL`, 0.315 m) to front–left–right (`FLR`, 0.197 m) reduces the translational error by 37%, largely because `Indr.` includes frequent right turns with many features inside the turning radius, which are well captured by the right camera. Incorporating the

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE II: Quantitative Comparison across Six Multi-Camera Sequences. We evaluate ORB-SLAM3 (RGB-D), VINS-RGBD, MCVIO, and our method on each of the six sequences, which cover challenging scenarios such as indoor settings, glare, textureless areas, occlusions, and dark environments. Because MCVIO does not use depth and therefore estimates scale only up to an unknown factor, we first align its trajectory to the ground truth with a full Sim(3). For each approach, we report the Absolute Trajectory Error (ATE_r, ATE_t). "TF" stands for *tracking failure*, while "IF" indicates *initialization failure*. ORB-SLAM3 (RGB-D version) and VINS-RGBD use only the front camera, whereas MCVIO and our method leverage additional cameras to improve accuracy in difficult conditions. The lowest (best) errors per row are marked in **bold**, and the second lowest are underlined.

	Sequence Indr.										Sequence Glare									
	ORB-SLAM3	VINS-RGBD	MCVIO				Ours				ORB-SLAM3	VINS-RGBD	MCVIO				Ours			
Cam	F	F	F	FL	FLR	FLRB	F	FL	FLR	FLRB	F	F	F	FL	FLR	FLRB	F	FL	FLR	FLRB
ATE_r	2.544	3.048	5.194	7.034	6.088	5.099	1.432	1.384	1.314	**1.277**	2.614	3.742	13.587	11.674	14.50	8.953	2.482	2.097	**2.022**	2.187
ATE_t	0.289	0.210	1.594	1.043	0.570	1.130	0.399	0.315	0.197	**0.174**	**0.340**	0.344	2.180	1.003	1.585	1.784	0.519	0.433	0.497	0.495

	Sequence Txl.										Sequence Occ									
	ORB-SLAM3	VINS-RGBD	MCVIO				Ours				ORB-SLAM3	VINS-RGBD	MCVIO				Ours			
Cam	F	F	F	FL	FLR	FLRB	F	FL	FLR	FLRB	F	F	F	FL	FLR	FLRB	F	FL	FLR	FLRB
ATE_r	TF	3.410	10.873	27.234	7.887	7.450	4.668	2.477	**2.279**	2.992	TF	TF	TF	11.085	7.244	7.052	TF	5.220	**5.064**	5.098
ATE_t		0.545	1.141	2.937	0.967	5.612	1.300	1.067	**0.533**	0.698				1.185	3.131	1.697		0.571	0.397	**0.378**

	Sequence Dark										Sequence Indr.2									
	ORB-SLAM3	VINS-RGBD	MCVIO				Ours				ORB-SLAM3	VINS-RGBD	MCVIO				Ours			
Cam	F	F	F	FL	FLR	FLRB	F	FL	FLR	FLRB	F	F	F	FL	FLR	FLRB	F	FL	FLR	FLRB
ATE_r	TF	IF	IF	IF	IF	IF	IF	6.593	4.153	**1.596**	2.179	4.896	9.250	10.984	8.468	13.850	4.107	1.490	**1.320**	1.385
ATE_t								1.824	0.554	**0.460**	0.549	0.421	6.176	4.796	4.484	6.654	0.634	0.427	0.424	**0.407**

back camera (FLRB, 0.174 m) provides a further improvement-particularly during rotations in textureless corridors where the back camera observes valid features. Notably, when MCVIO adds a back camera in this scenario, its translational error nearly doubles, indicating that simply increasing the number of cameras does not necessarily guarantee better performance without proper weighting strategies.

b) Glare: Strong sunlight causes glare, especially on side cameras, often leading to sudden darkening. Feature-based methods such as ORB-SLAM3 remain robust to this effect, whereas direct methods struggle even with affine brightness optimization due to their residual sensitivity to abrupt illumination changes. In our dataset, glare events occurred 4, 32, 8, and 12 times in the front, left, right, and back views, respectively. Nonetheless, adding the side cameras (FL, FLR) improves performance over a single front camera. Specifically, ATE_r reaches 2.02° with FLR, and remains close at 2.19° for FLRB. This indicates that, even under partial degradation, our confidence-driven framework dynamically down-weights affected cameras to maintain residual consistency across views. In contrast, MCVIO's ATE_t deteriorates with the inclusion of the right camera, suggesting that it lacks the ability to suppress degraded views and thus suffers from inconsistent residuals. These results show that, unlike MCVIO, our method remains robust under adverse illumination, so even when glare-degraded images come from side or rear views, adding those cameras does not deteriorate performance.

c) Txl.: This corridor-like environment has minimal texture. During rotation in these featureless sections, ORB-SLAM3 fails to maintain tracking. Among the front-only configurations (F), VINS-RGBD achieves the lowest error; however, our approach remains stable and benefits from additional cameras. Adding a left camera (FL) greatly reduces both rotation and translation errors, while incorporating a right camera (FLR) further halves the translation error, despite occasional saturations. When the back camera is added (FLRB), MCVIO's translation error rises, revealing vulnerability to rear-view saturation. By contrast, our confidence-driven weighting suppresses saturated measurements, preventing back-camera degradation and mitigating the effects of extra cameras.

d) Occ.: This sequence includes an occlusion in the front camera (Fig. 8), causing single-camera methods to fail. Although multi-camera setups can fall back on side or rear views, MCVIO's translational error remains relatively high and fluctuates as additional cameras are introduced, whereas our method maintains consistently lower and more stable errors across all camera combinations and shows minimal change in ATE_t. The largest improvement occurs when the right camera is added, while adding a back camera offers only marginal gains, suggesting that three cameras suffice in this scenario.

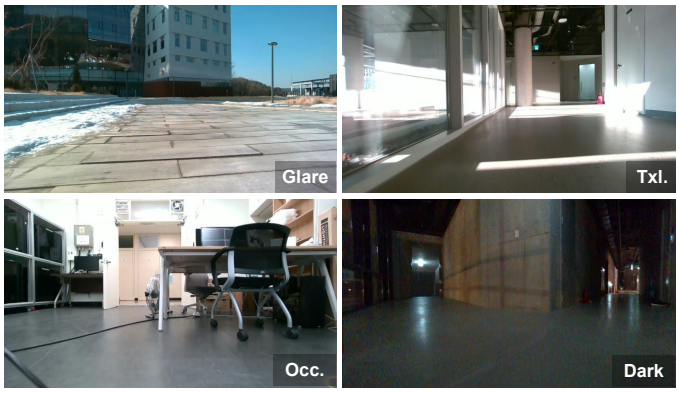


Fig. 7: Experimental Sequences. An overview of the four test sequences used in our experiments. *Glare* captures an outdoor environment under strong sunlight. *Txl.* contains white walls with minimal texture. *Occ.* experiences a complete blockage of the forward camera at some point. *Dark* begins in a dark environment and remains under low-light conditions overall.

e) Dark: Under extreme low light, front-only methods-VINS-RGBD, MCVIO (F), and our own-fail to initialize, and ORB-SLAM3 initializes successfully but eventually loses tracking during dark turns. MCVIO likewise fails even when additional cameras are added. Our method, however, tracks the entire sequence once a left camera is added (FL), although scale drift appears early on. Adding the right camera (FLR) dramatically mitigates this drift, cutting the translation error to roughly one-third of that in the FL case and significantly lowering rotation error. Finally, introducing the back camera (FLRB) reduces the rotation error by a further 62 %, because the directional weight w_{dir} retains a non-zero contribution for

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

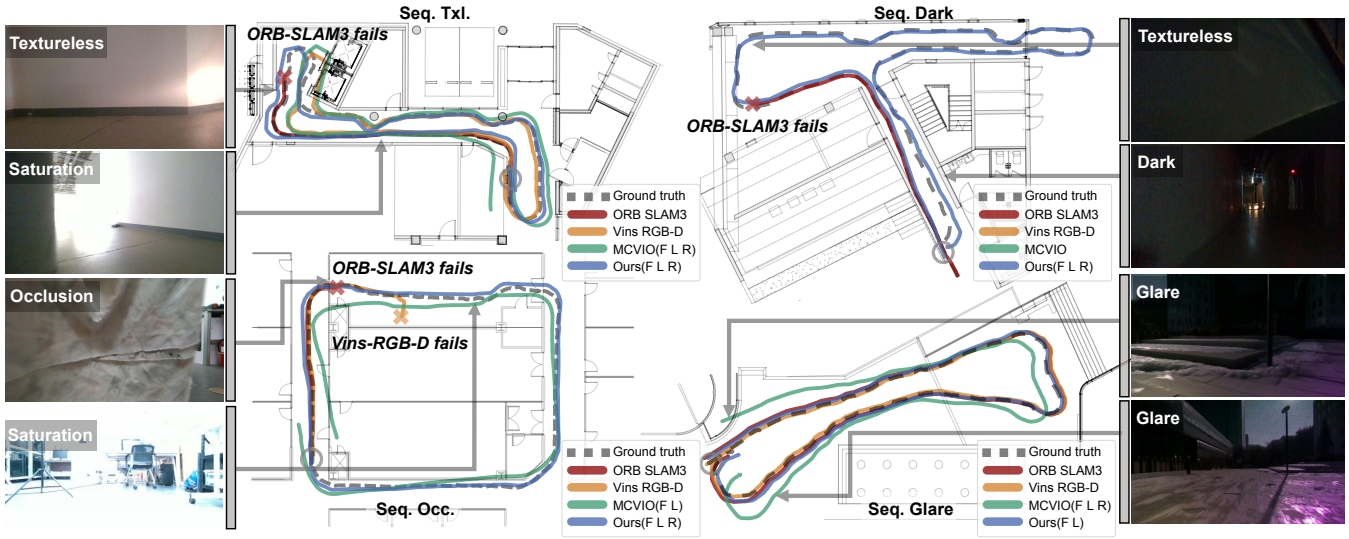


Fig. 8: Qualitative results on Txl., Dark, Occ., and Glare sequences, illustrating key conditions such as textureless areas, saturation, occlusion, darkness, and glare. The images highlight representative scenes of these challenging conditions in each sequence. The trajectory plot shows the results, which are aligned to the actual blueprint. For comparison, ORB-SLAM3 (RGB-D) and VINS-RGBD utilize the front camera, while for multi-camera methods (MCVIO, Ours), we plot the results using up to three cameras that achieve the lowest ATE_t. Note that in the Dark sequence, VINS-RGBD and MCVIO both fail to initialize.

the backward view; its residuals bolster tracking when the `FLR` configuration lacks valid features.

f) Indr.2: This open indoor environment exposes MCVIO's difficulty with extra cameras, as its rotation errors fluctuate from 9.250° to 13.850°. For the single front-camera configuration (`F`), ORB-SLAM3 achieves better rotation and translation accuracy. By contrast, our method leverages a wider FOV, reducing rotation error from 4.107° (`F`) to 1.490° (`FL`), and refining slightly with `FLR` (1.320°) and `FLRB` (1.385°). Since `FL`, `FLR`, and `FLRB` yield nearly the same accuracy, the right and back cameras add little benefit in this well-textured scene. Our motion-guided point selection and confidence-driven weighting deliver robust, precise odometry across diverse conditions, ensuring consistent accuracy as the sensor suite grows.

E. Selecting the Number and Combination of Cameras

Our pipeline can scale to n cameras, although we showcase results with up to four RGB-D sensors. The optimal configuration can depend on both the environment and sensor arrangement. We consider three main factors: *(i) coverage and robustness to degradation*, *(ii) achievable accuracy*, and *(iii) real-time performance (Table IV)*. Additionally, we investigate how specific camera combinations influence these factors.

a) Single Camera (1EA): Using only one RGB-D camera (`F`) achieves a stable processing rate of 30 Hz. However, as seen in Table II, single-camera methods often fail in challenging sequences such as occlusion and poor illumination.

b) Two Cameras (2EA): Adding a second camera (e.g., `FL` or `FR`) greatly expands coverage and reduces the risk of critical failure. Table III indicates a 15–30% drop in rotation error for two-camera setups compared to a single camera in moderately textured environments. Our system also maintains 30 Hz with two cameras, striking a good balance between sensor cost and robustness. Notably, in sequences like `Occ`, where the front camera is obstructed, a side camera preserves tracking; in `Dark`, a second viewpoint captures faint textures unavailable to the front camera. Hence, two-camera configurations often strike a practical balance for real-world use, offering notable gains in coverage and accuracy without compromising real-time performance.

TABLE III: Ablation Study of Adaptive Weighting (A.W.) We evaluate our adaptive weighting across multiple camera combinations on the `Indr.` sequence, a representative indoor case. For each setup, we compare ATE_r and ATE_t with (**w A.W.**) and without (**w/o A.W.**). "IF" indicates initialization failure, and bold numbers highlight best results.

seq. Indoor		Multi-Camera Configurations									
Cam		FL	FR	FB	LR	LB	RB	FLB	FRB	FLR	FLRB
w/o A.W.	ATE_r	1.591	**1.256**	2.893	2.505	2.792	2.643	3.484	2.798	1.847	2.687
	ATE_t	0.265	0.255	0.404	**0.229**	0.290	0.314	0.755	0.267	0.407	0.258
w A.W.	ATE_r	1.384	1.486	1.288	1.932	1.496	2.626	1.360	1.414	1.314	**1.277**
	ATE_t	0.315	0.238	0.480	0.293	0.652	0.505	0.282	0.263	0.197	**0.174**

c) Three or Four Cameras (3–4EA): Placing three or four RGB-D sensors (e.g., `FLR`, `FLRB`) ensures near-complete coverage, boosting resilience against glare, occlusions, or textureless areas. As Table II shows, three-camera setups usually outperform two-camera ones, and adding a fourth camera can further reduce errors under our adaptive weighting. For example, in the `Indr.` sequence, going from front–left (`FL`, 1.384°) to front–left–right (`FLR`, 1.314°) and then adding the back camera (`FLRB`, 1.277°) steadily refines the rotation error. Without such weighting as in Table. III, a backward camera can degrade accuracy in forward motion, but once weighting is active, misaligned views are automatically down-weighted. Meanwhile, the `Txl.` sequence (`FLR` shows how adding a third camera cuts the translation error nearly in half (from 1.067 m to 0.533 m), highlighting the benefit of a broader FOV in textureless corridors. Some scenes (*e.g.*, `Indr.2`) already saturate available features with three cameras, making a fourth one largely redundant. Nevertheless, our motion-guided point sel ection helps maintain real-time operation by pruning non-informative points early.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE IV: Average odometry output frequency on the `Indr.` sequence, measured for different numbers of cameras. Input camera images are synchronized at 30 Hz.

	1EA	2EA	3EA	4EA
MCVIO [17]	30.0 Hz	20.0 Hz	15.0 Hz	13.0 Hz
Ours	30.0 Hz	30.0 Hz	29.5 Hz	27.6 Hz

d) Specific Camera Combinations: The performance depends not only on the number of cameras but also on their spatial arrangement. Table III shows that in forward motion scenarios, a front-side configuration (`FL`, `FR`) usually outperforms a front-back configuration (`FB`), unless adaptive weighting is employed. With weighting enabled, even rear-facing views (e.g., `FLRB`) do not degrade overall accuracy because misaligned axes are down-weighted. Hence, our system can effectively handle glare or occlusions without excessive overhead or residual instability.

e) Recommendation: A single camera (`F`) may suffice in simpler environments (e.g., `Indr.`), but is vulnerable to occlusion (`Occ`) or glare (`Glare`). Two cameras (`FL`, `FR`) strike a good compromise between cost and robustness at 30 Hz (e.g., `Indr.2`), while three or four cameras (`FLR`, `FLRB`) handle harsher conditions (e.g., `Dark`, `Txl.`) with broad coverage. However, once coverage saturates (as seen in `Indr.2`), an extra camera may be unnecessary. Nevertheless, a rear-facing camera remains useful for backward motion, offering symmetrical coverage via our motion-aware alignment. Overall, our adaptive weighting and motion-guided selection efficiently manage multi-camera data in real-time, making three- or four-camera setups especially robust in high-risk scenarios.

F. Qualitative Evaluation and Real-Time Performance

We present representative trajectories from four challenging scenarios in Figure 8: `Txl.`, `Dark`, `Occ`, and `Glare`. Even when confronted with occlusions (`Occ`), poor illumination (`Dark`), or glare (`Glare`), our system retains stable tracking by adaptively down-weighting degraded cameras. For instance, in `Txl.`, the front camera mostly observes featureless white walls, so side cameras receive higher weight to preserve accurate poses. Table IV compares real-time performance across different camera configurations. While MCVIO [17] drops from 30 Hz to 13 Hz when scaling from one to four cameras, our pipeline remains at about 27.6 Hz by discarding non-informative points *before* alignment. This early filtering reduces computational overhead while preserving accuracy. As a result, our approach maintains accuracy and remains real-time even with up to four cameras in challenging scenarios.

V. Conclusion

We presented a Multi-RGB-D Inertial Odometry framework designed for non-overlapping camera views. By employing *motion-guided point selection* to discard uninformative or misaligned features, our approach maintains efficiency even as the number of cameras grows, enabling real-time operation at 27 Hz with four RGB-D sensors. In addition, our *adaptive weighting* mechanism down-weights degraded measurements, preventing low-quality cameras from compromising the odometry. As indicated by our title, "*The More, The Better?*," simply increasing the number of cameras does not guarantee improved performance without careful handling. In future work, we aim to explore multi-spectral sensor fusion to further enhance odometry robustness in challenging conditions.

References

[1] T. Zhang, J. Xu, H. Shen, R. Yang, and T. Yang, "Rmsc-vio: Robust multi-stereoscopic visual-inertial odometry for local visually challenging scenarios," *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4130–4137, 2024.

[2] J. Kim, Y. Cho, and A. Kim, "Proactive camera attribute control using bayesian optimization for illumination-resilient visual navigation," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1256–1271, 2020.

[3] L. Zhang, D. Wisth, M. Camurri, and M. Fallon, "Balancing the budget: Feature selection and tracking for multi-camera visual-inertial odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1182–1189, 2022.

[4] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[5] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007, pp. 225–234.

[6] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[7] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 834–849.

[8] L. v. Stumberg and D. Cremers, "Dm-vio: Delayed marginalization visual-inertial odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1408–1415, 2022.

[9] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[10] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 2100–2106.

[11] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, 2011, pp. 127–136.

[12] Z. Shan, R. Li, and S. Schwertfeger, "Rgbd-inertial trajectory estimation and mapping for ground robots," *Sensors*, vol. 19, no. 10, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/10/2251

[13] Z. Yuan, K. Cheng, J. Tang, and X. Yang, "Rgb-d dso: Direct sparse odometry with rgb-d cameras for indoor scenes," *IEEE Transactions on Multimedia*, vol. 24, pp. 4092–4101, 2021.

[14] L. Heng, G. H. Lee, and M. Pollefeys, "Self-calibration and visual SLAM with a multi-camera system on a micro aerial vehicle," *Auton. Robots*, vol. 39, no. 3, pp. 259–277, Oct. 2015.

[15] P. Liu, M. Geppert, L. Heng, T. Sattler, A. Geiger, and M. Pollefeys, "Towards robust visual odometry with a multi-camera system," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1154–1161.

[16] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for rgb-d cameras," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 3748–3754.

[17] Y. He, H. Yu, W. Yang, and S. Scherer, "Towards robust visual-inertial odometry with multiple non-overlapping monocular cameras," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 9452–9458.

[18] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration theory for fast and accurate visual-inertial navigation," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2015.