Tightly Coupled Multi-RGBD-Inertial Odometry: Leveraging Front and Rear Camera for Trajectory Estimation in Construction Site

Seungsang Yun[1] and Ayoung Kim[1*]

Abstract— **This paper presents a novel odometry framework that utilizes the tight integration of forward and backward RGB-D sensors to enhance environmental mapping and navigation accuracy. We exploits the complementary capabilities of binocular vision, tailored to maintain robust performance in challenging environments such as indoor construction sites. We introduce a new point selection algorithm that aligns with the robot's rotational direction, employing a motion-guided point selection method to tightly integrate two cameras facing different directions. This approach enhances the convexity of the optimized residual, thereby bolstering the tracking robustness. Additionally, we devise keypoints managing strategy to effectively manage key points that exceed the detection capabilities of RGB-D sensors, enhancing the thoroughness of spatial mapping. Validation tests was performed on our mobile robot platform.**

I. Introduction and Related Works

Estimating the robot's pose and mapping is essential in construction site areas. Primarily, Light Detection and Ranging (LiDAR) and cameras are employed for this purpose. While LiDAR provides accurate depth information, its high cost represents a substantial hindrance. Conversely, cameras are cost-effective and widely utilized. However, scale drift is a common issue in Odometry tasks when using monocular cameras. This challenge can be mitigated by implementing stereo camera systems that calculate depth through triangulation or by integrating inertial measurement unit (IMU) to minimize scale drift. Additionally, employing RGB-D cameras can effectively eliminate scale drift in Odometry tasks. RGB-D cameras, which utilize stereo matching with infrared pattern light, provide precise depth measurements and are cost-effective, enhancing their versatility. Consequently, RGB-D cameras are frequently employed in Odometry and simultaneous localization and mapping (SLAM) research. Huang et al. equipped a drone with an RGB-D camera for indoor mapping and navigation tasks. Furthermore, Whelan et al. built on [3] and introduced a real-time visual Odometry algorithm incorporating a GPU-based approach within the RGB-D camera Odometry framework for comprehensive mapping.

Yuan et al. enhanced the [5] by incorporating measured depth images for initial depth assignment and subsequent depth refinement. Nonetheless, these research exclusively utilize monocular RGB-D cameras, which can suffer from degraded tracking performance in the absence of valid points. This is particularly problematic in featureless environments such as construction sites with ongoing internal construction, where tracking failures may occur frequently (e.g., when a robot faces a gray wall). To address this issue, we expanded [4] to include a Multi-RGB-D camera system. Unlike stereo cameras, RGB-D cameras can deliver accurate depth from a single unit, rendering configurations with overlapping field of view (FOV) unnecessary.

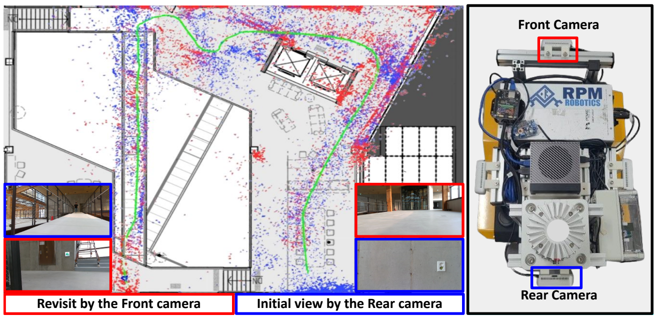

Fig. 1: **Odometry and Mapping Results of Our Approach:** The visualization features a point cloud delineated by a red boundary, emanating from the frontal camera, juxtaposed with a point cloud encapsulated by a blue boundary, originating from the rear camera. This representation underscores the initial point generation by the rear camera and its subsequent alignment upon the robotic unit's return.

Consequently, we developed a system that optimizes the utilization of data in Odometry by equipping RGB-D cameras on both the front and rear of the robot, as depicted in Fig. 1.

Analogously, Meng et al. introduced a trajectory estimation pipeline that computes estimations from each of three RGB-D cameras using a loosely coupled approach, updating the pose for the camera with minimal error. Optimization involves three separate estimators in this arrangement, potentially increasing computational demands. Moreover, should tracking failure arise in any single camera, re-initialization is necessary.

Alternatively, our approach integrates the front and rear RGB-D cameras through a tightly coupled strategy, creating a unified convex residual for optimization. To enhance tracking accuracy and facilitate extensive mapping across a broad FOV, we employed an adaptive point selection metric considering the robot's motion dynamics for both cameras. The efficacy of the proposed method was confirmed via the sensor system illustrated in Fig. 1, demonstrating that robust trajectory estimation is feasible even without valid points in indoor construction environments.

[†] This paper was supported by Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government(MOTIE) (P0020536, HRD Program for Industrial Innovation).

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

In summary, our contributions are as follows: We propose a tightly coupled RGB-D Odometry system, illustrated in Fig. 1, which incorporates both front and rear cameras without an overlapping FOV. We also demonstrated improvements in tracking performance, even under conditions of feature scarcity in the images from front or rear cameras.

- We developed a new point selection algorithm that employs a motion-oriented gradient direction for achieving a consistent convex direct-based optimized residual for the tightly integrated RGB-D Odometry.
- We formulated a strategy for effectively handling key points that fall outside the detection scope of RGB-D sensors.
- We validated our proposed methodology using our mobile robot platform, confirming the system's practical effectiveness.

II. Methodology

The framework of our system is depicted in Fig. 2. The system comprises three primary modules: Initialization, Tracking and Depth Refinement, and Back-end Optimization. The initialization stage entails defining a world coordinate system oriented along the direction of gravity, as measured by an IMU. This stage also includes activating initial points and estimating the initial pose (see Section 3-A). The Tracking and Depth Refinement module is designed to configure the residual and compute a coarse pose relative to the latest keyframe by optimizing this residual. Then, it detects point correspondences between consecutive frames through direct-based matching and adjusts depth values(see Section 3-B). Finally, the pose and point depth values are refined using a keyframe-based sliding window approach within the back-end optimization module (see Section 3-C).

A. *Initialization*

Initially, the direction of the gravity vector is established by averaging the measurements from the IMU. Subsequently, the world coordinates and the initial pose are established by aligning the z-axis with the gravity vector. Subsequently, the adaptive grid-based point selection metric is applied to both the front and rear cameras following the methodology outlined in [5]. Consequently, candidate points are identified throughout the image based on the magnitude of their gradients. If the depth measurements for these selected points lie within an acceptable range (notably, between 0.105m and 20m for the Realsense D435i), such measurements are set to their initial values during this selection phase. If the depth value is invalid, it is assigned an infinite value. Next, the error using solely the points with valid depth values is determined when the next frame is coming in, as described below.

$$E_{i,j} = \sum_{k \in \{F,R\}} \sum_{(u,v) \in P_i} \|\mathbf{I}_i(u,v) - a_{i,j}\mathbf{I}_j(u',v') - b_{i,j}\|_\gamma \tag{1}$$

Where the error function $E_{i,j}$ calculates the intensity difference for each point between frames i and j, applying by the affine parameters $a_{i,j}$ and $b_{i,j}$. The term $\|\cdot\|_\gamma$ represents the Huber loss, which is utilized for robust error estimation.

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \frac{d_{u,v}}{z'_{u,v}} \mathbf{K}_j \mathbf{T}^k_{i,j} \mathbf{K}_i^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{2}$$

The coordinate transformation $\mathbf{T}^k_{i,j}$ from frame i to frame j involves multiplication with the inverse intrinsic matrix $\mathbf{K}_i$ and projection using $\mathbf{K}_j$. The normalization factor $\frac{d_{u,v}}{z'_{u,v}}$ adjusts the depth to ensure that the transformed coordinates u' and v' are correctly scaled to the image plane of camera j.

An approximate pose between the initial two frames is estimated by optimizing the specified error. Subsequently, utilizing the result of the estimated pose, points are projected onto the jth image for depth refinement. Points that are not matched correctly are projected onto the depth image, and if the point is within the depth sensor's measurement range, the corresponding depth value is set.

Concurrently, the IMU measurements between successive frames i and j are pre-integrated utilizing the approach described in [7]. Subsequently, the discrepancy between the pre-integrated pose and the tracked pose is defined as the IMU Error. A sliding window optimization is then conducted to refine the 6 degree of freedom (DOF) pose, the depth of points, and the IMU biases.

B. *Motion-guided points selction and Tracking*

1) Motion-guided points selection: The convexity of residuals, employed in visual tracking and back-end optimization processes, is a fundamental component of deducing the robot's pose. In the direct-based approach, the residual's convexity is determined by the orientation of gradients at points extracted from the image and the steepness of the residual is influenced by the magnitude of the gradients associated with the points utilized.

In the current methodology in [5], point selection involves choosing candidates according to the magnitude of their gradients. Points exhibiting arbitrary gradient directions are selected by aligning them with the directional vector specified in the left graph of Fig. 3. However, this approach does not incorporate the robot's movement, and substantial rotational movements may lead to diminished tracking performance. Consequently, we endeavored to enhance tracking performance by choosing points whose gradient direction considers the robot's motion. Additionally, points were adaptively selected from the images from both the front and rear cameras.

Upon receiving a new frame, the angle of the motion vector is calculated based on matched point pairs from the most recent keyframe, referred to as the motion vector. Subsequently, the interval of the direction vector previously employed was narrowed. Additionally, as depicted on the right side of Fig. 3, the reference vector was rotated by the angle corresponding to the parallax vector, enabling the selection of points with gradients perpendicular to the

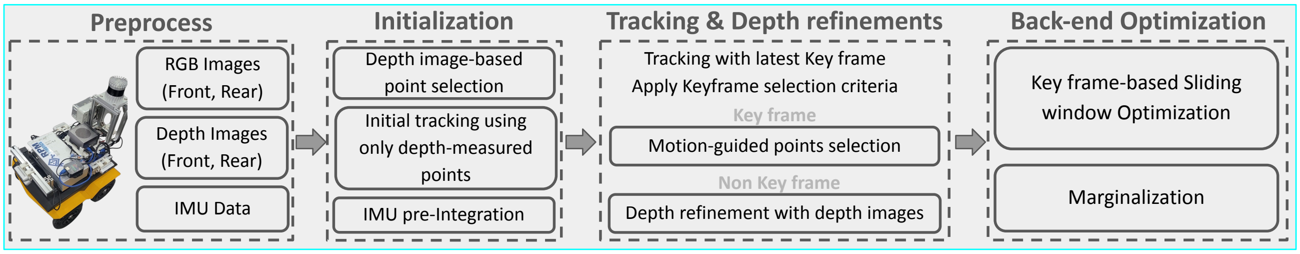

Fig. 2: **Overview of the Proposed Methodology:** This represents the framework of the Multi-RGB-D Inertial Odometry approach. Upon the reception of color and depth images from both the frontal and rear cameras, points deemed relevant for robot motion are discerned employing a motion-guided points selection metric. Subsequently, the tracking module endeavors to approximate the robot's initial pose. Following this, the refinement of point depth utilizing this estimated pose is executed, paving the way for bundle adjustment within the back-end optimization module. This operation serves to jointly optimize both the pose parameters and depth values associated with the identified points.

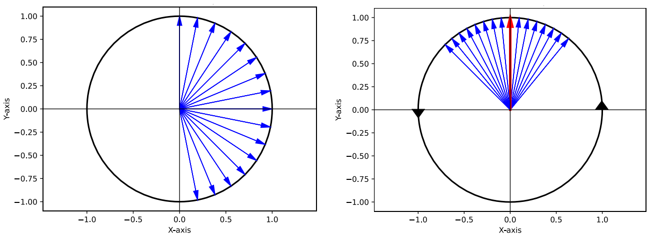

Fig. 3: This represents a directional vector utilized for the selection of pertinent points in direct-based odometry. The depiction on the left illustrates a directional vector encompassing the entire 360-degree field, while the depiction on the right showcases a directional vector with a gradient direction perpendicular to the motion of the robot, as determined by the central red arrow.

direction of rotation. This process is executed adaptively for both the front and rear camera images. Thus, if the front camera images are deficient in points valid for the direction of motion, additional points that consider motion are selected from the images from the rear camera. Tracking performance was enhanced by implementing a point selection metric incorporating the robot's motion across the front and rear cameras.

2) Visual Tracking: The Visual Tracking module utilizes an identical tracking methodology to that outlined in the initialization module (1), where discrepancies are delineated using activated points from the most recent keyframe of the current frame. Nevertheless, in calculating the Jacobian concerning the 6DOF pose, points activated by the rear camera produce a distinct Jacobian compared to those recorded by the front camera.

$$\mathbf{r}_{i,j} = \left\| \mathbf{I}_i(u,v) - a_{i,j}\mathbf{I}_j(u',v') - b_{i,j} \right\|_\gamma \tag{3}$$

Where $\mathbf{r}_{i,j}$ represents the residual between the latest keyframe and the current frame, the Jacobian associated with this residual, derived from points by the rear camera, was modified in accordance with the pose of the front camera using the following equation.

$$\frac{\partial \mathbf{r}_{i,j}}{\partial \delta \xi_F} = \mathrm{Adj}^T_{\mathbf{T}_{RF}} \frac{\partial \mathbf{r}_{i,j}}{\partial \delta \xi_R} \tag{4}$$

$$\mathrm{Adj}_{\mathbf{T}_{FR}} = \begin{bmatrix} \mathbf{R}_{FR} & \mathbf{0} \\ \mathbf{t}^{\wedge}_{FR}\mathbf{R}_{FR} & \mathbf{R}_{FR} \end{bmatrix} \in \mathbb{R}^{6\times 6} \tag{5}$$

Where, $\mathrm{Adj}_{\mathbf{T}_{FR}}$ denotes the adjoint matrix of extrinsic parameter between front and rear camera. ξ_F denotes the twist transformation associated with the front camera, while ξ_R represents the twist transformation corresponding to the rear camera. Among the points selected from both the front and rear cameras, solely those points that exhibited valid measurement values derived from the depth image were utilized to formulate a residual. Subsequently, the coarse pose corresponding to the front camera was estimated by optimizing (3). Based on the coarse pose, the depth value was refined by searching the epipolar line to establish correspondence. Concurrently, points without depth refinement—specifically, low-quality—were assigned the corresponding value from the depth image. Subsequently, analogous to the methodology described in [8], the measurement residual about point depth and the residual derived from tracking are propagated to the back-end optimization module. Finally, these residuals are added to the pre-integrated IMU residual, and the keyframe-based optimization is performed.

III. Experiment

A. *Experimental Setup*

To assess the effectiveness of the proposed method, tightly coupled Multi-RGB-D odometry, a dataset was generated utilizing the system depicted in Fig. 1. This setup comprises front and rear RGB-D cameras, precisely two Intel-Realsense D435i cameras, and a Microstrain 3DM-GX5-25 IMU. LiDAR SLAM results were used as the ground truth to provide a reliable baseline for trajectory estimations and data was collected using a Jetson Orin from Nvidia.

B. *Comparison*

The efficacy of the proposed algorithm was assessed against two state-of-the-art algorithms, [9] (RGB + IMU) and [10] (RGB-D + IMU), using [11]. The evaluation metrics included root mean square error (RMSE) across four dimensions: Absolute Translation Error (ATE), Absolute Rotation Error (ARE), Relative Translation Error (RTE), and

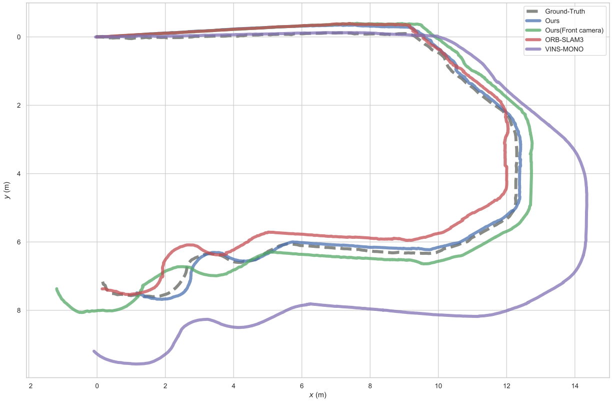

Fig. 4: **Qualitative Evaluation of Trajectory:** The presented illustration depicts the plotted trajectory of the proposed methodology alongside that of the comparative group. Ground truth trajectory was derived via LiDAR-Inertial SLAM and exhibits superior convergence to the ground truth compared to the methodologies referenced as [9] and [10].

TABLE I: **Quantitative comparative evaluation of the proposed methodology:** Assessed via Root Mean Square Error RMSE measures, comprising Absolute Translation Error ATE, Absolute Rotation Error ARE, Relative Translation Error (RTE), and Relative Rotation Error RRE.

Sequence 1	Vins Mono[9]	ORB SLAM3[10]	Ours(Front only)	Ours
(ATE)Rotation	2.79	3.42	2.66	**2.13**
(ATE)Translation	1.27	0.36	0.37	**0.33**
(RTE)Rotation	3.96	3.94	3.55	**3.45**
(RTE)Translation	2.47	0.53	0.63	**0.52**

Relative Rotation Error (RRE). All evaluation tests were conducted within a Ubuntu ROS environment, utilizing an Intel i9-11900 CPU and 64GB RAM. Table. I details the error metrics for [9], [10], Ours using only the front camera, and Ours integrating both front and rear cameras. The experiments covered approximately 28 meters in an indoor setting. As shown in Table. I, our methodology outperformed both [9] and [10] in all evaluated metrics. The integrated method of front and rear camera data resulted in enhanced performance compared to using only the front camera. This improvement is linked to the situations illustrated in Fig. 1, where the lack of suitable 6DOF residual-forming points in the front image required the inclusion of points from the rear camera to enhance the convexity of the residual during the optimization process. Fig. 1 visually represents the estimated trajectory used in the study, with our results showing closer alignment with the ground truth.

IV. Conclusion

This paper presents a Multi-RGB-D Camera Direct Inertial Odometry system. It showcases its robust tracking capabilities in environments with sparse features through the integrated fusion of front and rear RGB-D cameras. The research introduces a novel point selection metric that facilitates a convex of the optimization error by employing a motion-guided point selection metric. Adaptively selecting corresponding points across the front and rear cameras using the points selection method, we can empirically confirm that the proposed system produces a convex residual in environments lacking distinct features, such as construction site environments. Our proposed framework exhibits superior performance metrics compared to leading-edge methodologies, particularly VINS-Mono [9] and ORB SLAM3[10].

References

[1] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, “Visual odometry and mapping for autonomous flight using an rgb-d camera.”

[2] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, “Robust real-time visual odometry for dense rgb-d mapping,” in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 5724–5731.

[3] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, “Kintinuous: Spatially extended kinectfusion,” 2012.

[4] 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.

[5] 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.

[6] X. Meng, W. Gao, and Z. Hu, “Dense rgb-d slam with multiple cameras,” *Sensors*, vol. 18, no. 7, p. 2118, 2018.

[7] 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.

[8] L. Von Stumberg, V. Usenko, and D. Cremers, “Direct sparse visual-inertial odometry using dynamic marginalization,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2510–2517.

[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. 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.

[11] Z. Zhang and D. Scaramuzza, “A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7244–7251.