IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

ROVINS: Robust Omnidirectional Visual Inertial Navigation System

Hochang Seok[1] and Jongwoo Lim*[1,2]

***Abstract*— Visual odometry is an essential component in robot navigation and autonomous driving; however, visual sensors are vulnerable in fast motion or sudden illumination changes. This weakness can be compensated with inertial measurement units (IMUs), which maintain the short-term motion when visual sensing becomes unstable and enhance the quality of estimated motion with inertial information. An omnidirectional multi-view visual odometry (ROVO) has recently demonstrated superior performance and stability with the unceasing feature observation of the omnidirectional setup; however, the shortcomings of visual odometry remain. This paper introduced an omnidirectional visual-inertial odometry system (ROVINS) that could seamlessly integrate the inertial information into the omnidirectional visual odometer algorithm: (a) The soft relative pose constraints from the inertial measurement are first added to the pose optimization formulation, which enables blind motion estimation when all visual features are lost; (b) Using the prediction results from the estimated velocity, the visual features in tracking are initialized, resulting in feature tracking that is more robust to visual disturbances. The experimental results showed that the proposed ROVINS algorithm outperforms the vision-only algorithm by a significant margin.**

I. INTRODUCTION

Estimating ego-motion is a critical task for robots and autonomous agents. This can be approached with several existing techniques, one of the most popular being visual odometry (VO), mainly because of the cameras, which, aside from being much cheaper than special sensors, makes a fairly accurate and robust motion estimation. Visual sensors provide rich information of environment structures and objects in high resolution and relatively high speed, however, they are sensitive to illumination conditions and motions. For instance, even with auto-exposure functionality in cameras, sudden illumination changes can make the subsequent images look very different and the feature tracking to fail. Another challenge is the fast motion of the shutter speed where motion blur originates to create blurry visual features and unstable feature tracking. Such drawbacks in the use of visual odometry have been compensated by the installation of inertial measurement units (IMUs) that can sense motion differently. IMUs measure the linear acceleration and angular rotation velocity, which are then integrated to provide the displacement and rotation. However, IMU-based motion estimation is normally affected by a drift: the (double) integration process allows a very small error in measurements to cause a large error at the positional or rotational estimate. Drift errors accompanying IMU measurements cannot be corrected as the units only measure the relative motion information. Accordingly, they can be detected or corrected using visual inputs, because they allow the absolute pose (position and rotation) with respect to the environment to be measured from multiple images.

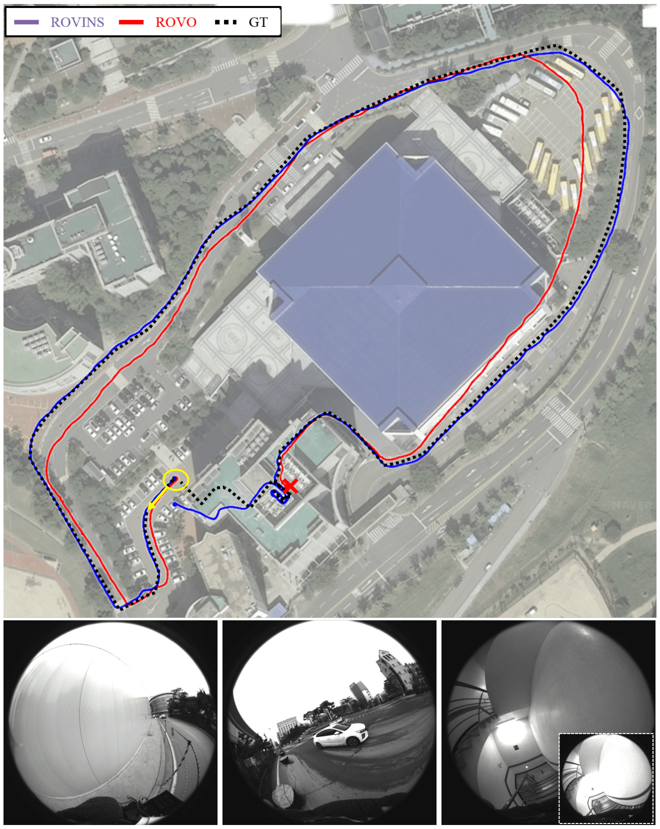


Fig. 1: Qualitative results of the proposed system in the Large-scale In-Out dataset, including various challenging scenes (bottom: textureless, large-scale, auto-gain). After being aligned to the gravity direction, the estimated trajectories of ROVINS(blue), ROVO(red), and GT(black) are overlaid on the satellite image. For comparison, the ground-truth trajectory is drawn by hand. Note that ROVO fails at point X marked in red.

In various challenging situations, visual-inertial odometry (VIO) can be used to provide an accurate and robust estimation of the camera-IMU rig motion. With this technique, visual and inertial motion information are complementary; thus, when the visual observation is missing due to illumi-

This work was supported in part by the Next-Generation Information Computing Development Program through National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT(NRF-2017M3C4A7069369), in part by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF-2019R1A4A1029800).

[1] Department of Computer Science, Hanyang University, Seoul, Korea `hochangseok@hanyang.ac.kr, jlim@hanyang.ac.kr`

[2] MultiplEYE Co., Ltd., Seoul, Korea.

*Corresponding author.

nation conditions or fast motion, the inertial information can keep the motion momentarily, while the visual information helps correct the bias and noise in inertial measurements. The most common setup of VIO consists of a monocular camera and an IMU that are commonly found in smartphones.

The main context of this paper is the proposal of a VIO system for an omnidirectional multi-view visual odometry (ROVO) camera rig. In ROVO [1], four ultra-wide fisheye cameras are utilized for motion estimation. These cameras are equipped with 220° field-of-view (FOV) lenses (Fig. 2) to maximize the overlapping regions for stereo matching of the tracked features. Compared to conventional monocular or stereo VO systems, ROVO demonstrates a superior performance as feature points remain in view until they are occluded by other objects or become too far away. However, it still possess the limitations of conventional VO algorithms as it purely relies on visual feature points. Integrating the inertial motion constraints into ROVO's optimization framework, the proposed robust omnidirectional visual inertial navigation system (ROVINS) further improves the motion estimation performance of the conventional VIO or ROVO systems. To improve the feature tracking, individual feature motions are predicted from the relative motion measurements of the IMUs, and then used as the initial feature locations for feature tracking, e.g., Kanade-Lucas-Tomasi (KLT), in the next frame. In this paper, the performance of the proposed ROVINS algorithm is evaluated by capturing very challenging test sequences with fast motions, severe illumination variations, and crowded situations. The ground-truth rig pose data are collected by a motion capture system, and both qualitative and quantitative comparisons are conducted to demonstrate the improved performance of the algorithm.

II. RELATED WORKS

Various sensor configurations for pose estimation have been researched to ensure the robustness and performance. For vision-based pose estimation, several visual odometry methods [2]–[6] developed for monocular cameras have shown good performance, but often suffered from a scale drift; moreover, the estimated poses were up-to-scale (non-metric) unless additional information for metric upgrade has been provided. These limitations were overcome with stereo camera-based VO/VSLAM (simultaneous localization and mapping) systems. Mur-Artal and Tardós [7], and Wang *et al.* [8] developed a stereo camera-based visual SLAM system that runs robustly in small indoor or outdoor environments. Meanwhile, using wide-FOV cameras improves the robustness as apparent feature motions become more evident. For instance, Caruso *et al.* [9] proposed a fisheye camera-based visual SLAM system with a direct method that optimizes the photometric errors between images, whereas Liu *et al.* [10] and Matsuki *et al.* [11] developed stereo fisheye camera-based VO systems. Recently, VIO systems with a fusion of cameras and IMU sensors have become more popular, not only because IMU sensors provide noisy but camera-independent motion with high rates, but also as they help estimate the absolute metric scale of the trajectories. Such an advantage significantly improves the accuracy and robustness of conventional camera-based VO algorithms.

Many monocular VIO methods [12]–[16] have shown superior accuracy in real-world environments, while various multi-camera-based methods have been reported to improve the perception abilities. The visual odometry system presented in [17] shows a full surrounding view camera system, whose robustness has also been demonstrated with the visual SLAM of Heng *et al.* [18] with a self-calibration. More recently, Liu *et al.* [19] has presented a multi-camera system using a direct method with a plane sweeping stereo. Furthermore, the multi-camera based VIO methods [20]–[22] have been shown to deliver an outstanding performance. The VO algorithm of Seok and Lim [1] for a multi-view wide-FOV camera setup have also been reported to demonstrate a superior accuracy and robustness in large-scale motion estimation.

III. PRELIMINARIES

A. IMU Pre-Integration

Fig. 2 shows the IMU b, which measures the acceleration and angular velocity at regular time intervals Δt. Here, slowly varying biases $\mathbf{b}_a$ and $\mathbf{b}_g$ of the accelerometer and gyroscope, as well as sensor noise, affect the sensory information. Moreover, the gravity ${}^{\mathrm{w}}\mathbf{g}$ needs to be subtracted from the raw accelerometer output to remove its effect when computing the motion. The IMU motion between two consecutive keyframes can be defined in terms of the pre-integration $\Delta\mathbf{R}$, $\Delta\mathbf{v}$, and $\Delta\mathbf{p}$ from all measurements in-between, and the IMU orientations ${}^{\mathrm{b}}_{\mathrm{w}}\mathbf{R} \in SO(3)$, position ${}^{\mathrm{b}}_{\mathrm{w}}\mathbf{p}$, and velocity ${}^{\mathrm{b}}_{\mathrm{w}}\mathbf{v}$, as described in [15]:

$$\begin{aligned}
{}^{\mathrm{b}}_{\mathrm{w}}\mathbf{R}_{i+1} &= {}^{\mathrm{b}}_{\mathrm{w}}\mathbf{R}_i \Delta_{i,i+1} \operatorname{Exp}\left(\mathbf{J}^g_{\Delta R}\mathbf{b}_{gi}\right), \\
{}^{\mathrm{b}}_{\mathrm{w}}\mathbf{v}_{i+1} &= {}^{\mathrm{b}}_{\mathrm{w}}\mathbf{v}_i + {}^{\mathrm{w}}\mathbf{g}\Delta t_{i,i+1} \\
&\quad + {}^{\mathrm{b}}_{\mathrm{w}}\mathbf{R}_i \left(\Delta\mathbf{v}_{i,i+1} + \mathbf{J}^g_{\Delta v}\mathbf{b}_{gi} + \mathbf{J}^a_{\Delta v}\mathbf{b}_{ai}\right), \\
{}^{\mathrm{b}}_{\mathrm{w}}\mathbf{p}_{i+1} &= {}^{\mathrm{b}}_{\mathrm{w}}\mathbf{p}_i + {}^{\mathrm{b}}_{\mathrm{w}}\mathbf{v}_i\Delta t + \frac{1}{2}\,{}^{\mathrm{w}}\mathbf{g}\Delta {t_{i,i+1}}^2 \\
&\quad + {}^{\mathrm{b}}_{\mathrm{w}}\mathbf{R}_i \left(\Delta\mathbf{p}_{i,i+1} + \mathbf{J}^g_{\Delta p}\mathbf{b}_{gi} + \mathbf{J}^a_{\Delta p}\mathbf{b}_{ai}\right),
\end{aligned}$$

where the Jacobian ${\mathbf{J}^g}_{(\cdot)}$ and ${\mathbf{J}^a}_{(\cdot)}$ account for a first-order approximation of the effect of changing the biases without explicitly recomputing the pre-integrations. A more detailed, efficient computation of the pre-integrations and Jacobians can be found in [15].

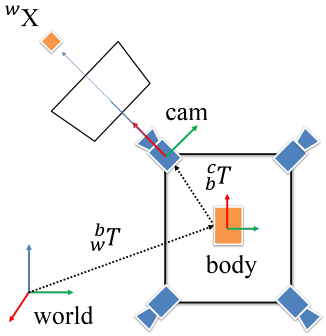


Fig. 2: The world, IMU(body), and camera coordinate systems of the proposed VIO system and their respective transformations.

B. Notation

A rigid transformation $\mathbf{T}$ is parameterized as a rotation vector $\mathbf{r}$ and a translation vector $\mathbf{t}$ in $\mathbb{R}^3$. Essentially, it transforms a 3D point $\mathbf{X}$ to $\mathbf{T}\odot\mathbf{X} = R(\mathbf{r})\mathbf{X}+\mathbf{t}$, where $R(\mathbf{r})$ is the 3×3 rotation matrix for the rotation $\mathbf{r}$. Accordingly, $\odot$ denotes the composition of transformations, whereas $^{-1}$ is the inverse transformation. As shown in Fig. 2, three coordinate systems are used, namely, world (w), body (b), and camera (c). When needed, the coordinate system is marked on the left side of a transformation, as in the case of ${}^{\mathrm{w}}_{\mathrm{b}}\mathbf{T}$, which implies a rigid transformation from the body-to-world coordinate system, or ${}^{\mathrm{w}}\mathbf{X}$, as a point in the world coordinate system. Note that in this paper, the body coordinate system is aligned with the IMU coordinate system. Time is specified using a right subscript, for example, the camera coordinate of a world point $\mathbf{X}$ at time t can be written as

$$ {}^{\mathrm{c}}\mathbf{X}_t = {}^{\mathrm{c}}_{\mathrm{b}}\mathbf{T} \odot {}^{\mathrm{b}}_{\mathrm{w}}\mathbf{T}_t \odot {}^{\mathrm{w}}\mathbf{X}. $$

The camera intrinsic parameters determine the mapping between a 3D point ${}^{\mathrm{c}}\mathbf{X}$ and a pixel coordinate $\mathbf{x}$ in the image; for example, the projection function $\mathbf{x} = \pi({}^{\mathrm{c}}\mathbf{X}; \boldsymbol{\phi})$ denotes the camera intrinsic parameter $\boldsymbol{\phi}$. Because of the ultra-wide FOV cameras, a unit sphere instead of a plane is used for computing rays for 3D points. Let $\pi_0(\cdot)$ represent the projection onto the unit sphere, thus, $\bar{\mathbf{x}} = \pi_0(\mathbf{X})$ is a unit-length ray pointing $\mathbf{X}$, which also serves as the normalized image coordinates of the projected point. The parameter ρ is the Cauchy robust norm used in the optimization. Accordingly, the state vector of the optimization $\boldsymbol{\theta}$ is defined as

$$ \boldsymbol{\theta} = \left\{ {}^{\mathrm{b}}_{\mathrm{w}}\mathbf{T}_i, {}^{\mathrm{b}}_{\mathrm{w}}\mathbf{v}_i, \mathbf{b}_{ai}, \mathbf{b}_{gi}, \{{}^{\mathrm{w}}\mathbf{X}\}_i \right\}_{i \in W} \quad , $$

where $\mathbf{T}$ and $\mathbf{v}$ represent the body poses and velocities in the world coordinate, $\mathbf{b}_a$ and $\mathbf{b}_g$ the accelerometer and gyro biases, and $\mathbf{X}$ the observed 3D landmark positions.

IV. ALGORITHM

Fig. 3 displays an overview of the system architecture of ROVINS with IMU-predicted feature tracking. Here, the camera intrinsic parameters and the IMU-to-camera extrinsic parameter were assumed to be calibrated and given, and that all cameras capture images synchronously with the IMU data time-synchronized with the cameras. First, the raw fisheye images were warped to the hybrid projection images [1], and the motion from IMU data was propagated using the mid-point pre-integration [12]. Next, feature detection and IMU-aided intra-view feature tracking were performed in the hybrid projection images. Propagated rotation from the IMU was input to the IMU-aided feature tracker for predict of the feature position in the current frame. This was followed by an inter-view stereo feature matching to find feature correspondences between the cameras. Once the data processing step was completed, the camera and the IMU were checked whether or not they were initialized. If not initialized, vision-only Structure-From-Motion(SFM) was carried out by ROVO [1] to process the visual-inertial alignment. Afterwards, the system was initialized to approach VIO using non-linear optimization. Each process is described fully in the following subsections.

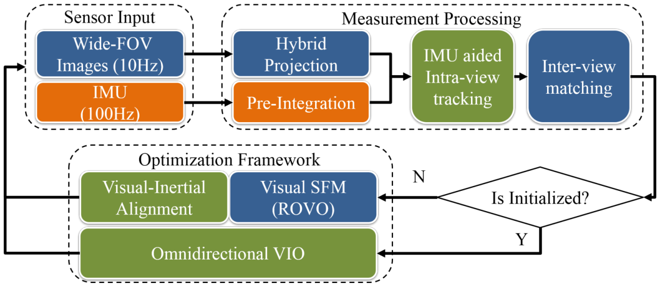


Fig. 3: An overview of the proposed system. Each color specifies the processing blocks relative to the camera(blue), the IMU(orange), and the camera+IMU(green), respectively. See Section IV for the detailed description.

A. Measurement Processing

The raw input images and IMU measurements were processed continuously throughout the algorithm. Initially, the raw images were projected to the hybrid-projected images [1] and used for feature extraction, tracking, and matching, as in the cases with ROVO [1]. Utilizing the hybrid-projection images should minimize the distortion and maximize the feature matching and tracking across views. These initial steps were necessary for the feature tracking to work from discovery to disappearance of visual features. Further, oriented FAST and rotated BRIEF (ORB) features were extracted from the hybrid-projected images as inputs to intra-view tracking and inter-view matching. Concurrently, IMU measurements were propagated using pre-integration methods [15]. The pre-integration computes the relative pose change from the previous image frames and its uncertainty in the pose covariance matrix. After both measurements have been processed, the pre-integrated IMU motion was utilized to improve the feature tracking performance, followed by stereo feature matching across views.

B. Feature Tracking with Prediction

The KLT-tracking [23] technique, given its accuracy and efficiency [1], [12], is generally popular for finding feature correspondences between consecutive images. By principle, a KLT-tracker starts at the previous feature location and searches the current feature location using photo-consistency. When handling large motions, KLT uses an image pyramid where a coarse position is computed first and then the finer motion is updated. Therefore, having good initial locations in a new frame is important for accurate and fast feature tracking. In this paper, the predicted feature location was calculated using the pre-integrated IMU pose, before the KLT tracker was initialized at the predicted position. Note that predicting precise feature locations requires highly accurate poses and depth of features. However, it is difficult to maintain such information at all times, especially at the process onset. Except very near features, the displacement due to the camera translation is much smaller than the camera rotation.

Thus, in this work, the feature location prediction was accomplished by re-projecting the 3D feature points to the current image plane using the IMU-propagated motion when the 3D feature point was available. Otherwise, if the feature has not been not registered yet, only the rotation of the propagated IMU was considered for the prediction. Although the propagated IMU motion becomes unreliable in the long-term, it should be accurate enough to initialize feature tracking points in the short propagation time period (of approximately 100 ms, the same as the image frequency).

C. Vision-Inertial Initialization

Vision-inertial initialization is the essential step to successfully fuse the two very different measurements, namely, the gravity direction and IMU biases. However, as these parameters are unknown at the beginning, they should be bootstrapped from the visual and inertial measurements. In this work, a similar approach of the loosely coupled sensor fusion method as Qin *et al.* [12] was employed.

1) Vision-Only SFM: A highly accurate vision-based SLAM system would work well on the initialization, which is dependent on good visual SFM results. Thus, fully utilizing the omnidirectional multi-view setup, ROVO [1] achieves high robustness even in the dynamic scenes and textureless indoor environments. Moreover, because it observes all direction, there is little chance for the initial motion to degenerate (e.g., pure rotation); such failure cases can be filtered by simply checking the number of pose inliers ($>$50). With this advantage as well, the initial SFM can be assumed to succeed and to return reasonably accurate poses, as long as the rig moves enough distance. When the system is turned on, it first monitors if enough motion (15 keyframes in the present setup) is generated, before the visual-inertial alignment is performed.

2) Visual-Inertial Alignment: In the present system, the metric scale is directly observable from the omnidirectional multi-view stereo setup, allowing the fusion of the IMU and camera measurements with no worries on the initial scale estimation and scale update. The methodology of Hong and Lim [16] was employed for the alignment. In particular, the implementation of the authors was modified so as not to estimate the scale. Positively, such modification worked well in the proposed multi-view system. However, only the gravity direction, initial velocity, and IMU biases were estimated in this paper using the modified code of [16] for the simplification.

D. Optimization-based Visual-Inertial Odometry

The proposed VIO algorithm continuously estimates the body pose, velocity, and IMU biases at a frame rate, which resulted in the reliable calculation of the rig system's trajectory. Once the initialization has been completed, the current frame pose was updated with the IMU pre-integrated pose, and then the outlier features were rejected based on the reprojection error, or the tangential error of unit ray in the study's ultra-wide FOV setup.

After the outlier rejection, the state vectors $\boldsymbol{\theta}$ of the current frame and the keyframes in the active local window W were optimized in both their visual and IMU constraints. The optimization problem was dealt with a Ceres solver [24], yielding the optimal state,

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{j,i_j} \sum_{t \in W} \left(\omega_{i_j} \mathbf{E}_{vis}(i,j,t) + \mathbf{E}_{imu}(t-1,t) \right),$$

where $\mathbf{E}_{vis}$ represents the visual constraints, $\mathbf{E}_{imu}$ the IMU constraints, and ω the weight parameters. Note that ω is proportional to the number of counts upon which the relative 3D landmark has been checked as an inlier.

1) Visual Constraint: With the given body poses, extrinsic and map points, the visual constraint was calculated based on the re-projection error of the feature points. A visual constraint is widely used as a geometric error term for many feature-based visual SLAM and VIO systems [1], [3], [12], [13], [25]. Accordingly, the re-projection error $\mathbf{E}_{vis}$ is given by

$$\mathbf{E}_{vis}(i,j,t) = \rho \left(||\bar{\mathbf{x}}_{i_j,t} - \pi_0({}^{j}_{\mathrm{b}}\mathbf{T} \odot {}^{\mathrm{b}}_{\mathrm{w}}\mathbf{T}_t \odot {}^{\mathrm{w}}\mathbf{X}_{i_j})||^2 \right),$$

with the extrinsic point from the body to the j'th camera ${}^{\mathrm{c}}_{\mathrm{b}}\mathbf{T}_t$, the body pose at time t ${}^{\mathrm{b}}_{\mathrm{w}}\mathbf{T}_t$, and the landmark positions observed by i and j'th camera $\{{}^{\mathrm{w}}\mathbf{X}_i\}_j$ in the world coordinate.

2) IMU Constraint: With the IMU readings between two consecutive keyframes, the measurements were pre-integrated [15] to obtain the relative IMU constraint. This restriction is a widely used pre-integration term for optimization-based VIO systems [12], [13], [25]. Here, the pre-integration error $\mathbf{E}_{imu}$ is defined by

$$\begin{aligned}
\mathbf{E}_{\text{imu}}(k,l) =& \rho\Big([\mathbf{e}_R^T \mathbf{e}_v^T \mathbf{e}_p^T] \boldsymbol{\Sigma}_I [\mathbf{e}_R^T \mathbf{e}_v^T \mathbf{e}_p^T]^T \Big) + \rho(\mathbf{e}_b^T \boldsymbol{\Sigma}_R \mathbf{e}_b), \\
\mathbf{e}_R =& \log\left((\Delta\mathbf{R}_{kl} \operatorname{Exp}(\mathbf{J}^g_{\Delta R} \mathbf{b}_{g_l}))^T \, {}^{\mathrm{w}}_{\mathrm{b}}\mathbf{R}_k \, {}^{\mathrm{b}}_{\mathrm{w}}\mathbf{R}_l \right), \\
\mathbf{e}_v =& {}^{\mathrm{w}}_{\mathrm{b}}\mathbf{R}_k \left({}^{\mathrm{b}}_{\mathrm{w}}\mathbf{p}_l - {}^{\mathrm{b}}_{\mathrm{w}}\mathbf{v}_k - {}^{\mathrm{w}}\mathbf{g}\Delta t_{kl} \right) \\
& - \left(\Delta\mathbf{v}_{kl} + \mathbf{J}^g_{\Delta v} \mathbf{b}_{g_k} + \mathbf{J}^a_{\Delta p} \mathbf{b}_{a_l} \right), \\
\mathbf{e}_p =& {}^{\mathrm{w}}_{\mathrm{b}}\mathbf{R}_l \left({}^{\mathrm{b}}_{\mathrm{w}}\mathbf{p}_l - {}^{\mathrm{b}}_{\mathrm{w}}\mathbf{p}_k - {}^{\mathrm{b}}_{\mathrm{w}}\mathbf{v}_k \Delta t_{kl} - \frac{1}{2} \, {}^{\mathrm{w}}\mathbf{g}\Delta t_{kl}^2 \right) \\
& - \left(\Delta\mathbf{p}_{kl} + \mathbf{J}^g_{\Delta p} \mathbf{b}_{g_l} + \mathbf{J}^a_{\Delta p} \mathbf{b}_{a_l} \right), \\
\mathbf{e}_b =& \mathbf{b}_l - \mathbf{b}_k,
\end{aligned}$$

where $\boldsymbol{\Sigma}_I$ is the information matrix from the pre-integration and $\boldsymbol{\Sigma}_R$ of the bias random walk. The pre-integration process is fully explained in [15]. The optimization structures of ROVO and ROVINS are shown in Fig. 4 in a comparative manner.

V. EXPERIMENTS

A. Experimental Setup

All datasets were captured using a small square-shaped rig (0.3 × 0.3 m) with four 220° wide-FOV cameras as in [1], [26], [27], and one Xsens MTi-10 IMU sensor. The capture system recorded synchronized $4\times(1600\times1532)$ gray

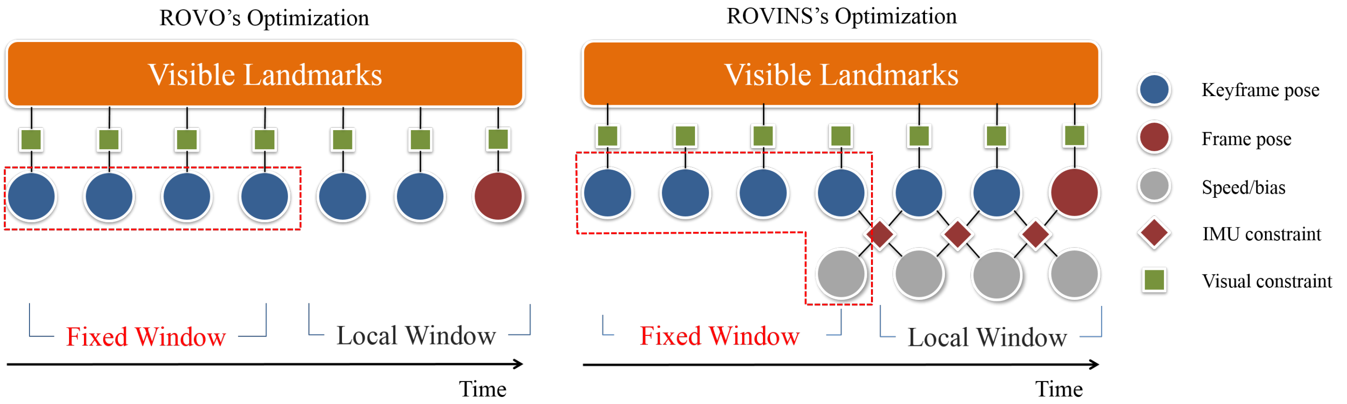


Fig. 4: Comparison of optimization methods between ROVO(left) and the proposed ROVINS(right). The local windows in both methods are retrieved by temporal order of keyframes. ROVO optimizes the poses and landmarks using visual constraints only, whereas ROVINS's optimization allows the poses, speed, biases, and landmarks to be simultaneously optimized by the visual and IMU constraints.

images at up to 10 Hz, as well as IMU measurements at up to 200 Hz. The intrinsic and extrinsic parameters between the cameras were calibrated using a checkerboard [26], [28], [29]. In particular, the extrinsic parameters between the camera and the IMU were calibrated using Kalibr [30], a popular open-source toolbox. The hardware configuration is shown in Fig. 5.

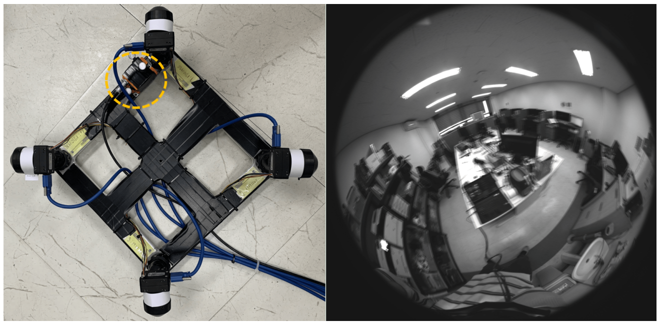


Fig. 5: The capture system used for this experiment. Four wide-FOV fisheye cameras and one IMU sensor are attached to the rig (left). The highlighted yellow dotted circle is the location of the IMU. The cameras capture an extremely challenging dataset with severe motion blurry images (right).

B. Dataset description

A total of 9 challenging real-world datasets were collected from the experiments, which were subsequently used for quantitative and qualitative evaluations. The datasets were labelled as follows: Normal Hand-held, Spinning Hand-held, Shaking Hand-held, Dynamic Objects, Spinning In-Out, Large-scale In-Out, and Mocap All (Mocap0, 1, and 2 are parts of Mocap All with different motions). The Normal, Spinning, and Shaking Hand-helds were captured with three different challenging motions; the dynamic Objects included an abruptly changing motion and multiple dynamic objects and; the Spinning In-Out and Large-scale In-Out were especially challenging, as both have been captured by a person walking around indoors and outdoors, includes the opening of doors, illumination changes, and moving objects. Specifically, the camera motion of the first one was highly dynamic, but the length of its trajectory was relatively shorter than the second one; second one was captured in moderate motion around the large area of a university campus with its illumination drastically changing by the auto-gain. Finally, the Mocap series were captured in a motion capture room with various motions, including spinning and shaking. These datasets were recorded with the ground-truth poses, captured by the highly accurate motion capture device, which enabled the precise evaluation of the estimated poses in the challenging environment.

C. Parameter Settings

All key parameters were shared by both ROVO and ROVINS for fair comparison. First, the size of the hybrid projected images were set to 640 × 480. The maximum number of tracked features in each image was set to 150 in all sequences. To spatially and uniformly distribute the features in an image, the margins between the detected features were set to 11 pixel. The patch size of the KLT tracker was 15×15 with four pyramid levels. Note that increasing the resolution of hybrid images or the number of features improves the overall performance of the feature-based SLAM systems, like ORB-SLAM2, as it provides more detailed information about the visual observations [31]. However, because of the trade-off between performance and running-time, the above parameters were selected based on both the performance and time-complexity of the system. Two keyframe selection parameters were used: the relative translation and rotation thresholds from the last keyframe were set to 0.15 m and 5° respectively in all experiments, except in Large-scale In-Out, where the parameters were set to 2 m and 5°. Additionally, these parameters were chosen heuristically considering the characteristics of the environment. The respective sizes of the local-bundle optimization window and the fixed window were 10 and 15. With the above parameters as inputs, both ROVO and ROVINS were ran at nearly 10 Hz with multi-threaded implementation.

D. Evaluation of Estimated Poses

The captured datasets were utilized for extensive experimental comparisons evaluating the accuracy and robustness of the proposed ROVINS system. Two error metric types, the absolute trajectory error of translation (ATE$_{trans}$) and the start-to-end error, were employed for the quantitative evaluation. On one hand, ATE$_{trans}$ is widely used for measuring the accuracy of VO/VIO systems; it is calculated by the root mean-squared error (RMSE) of all rig positions after rigidly aligning the estimated trajectory to the ground-truth trajectory. On the other hand, the start-to-end error is calculated by the difference between the start and end positions for the sequences whose start and end positions should be the same. Although not as sophisticated as the ATE, it is still the start-to-end error remains useful for measuring the overall performance of the algorithms, especially for large or complex environments where the ground-truth poses are difficult to obtain. Accordingly, the proposed VIO method was compared to ROVO [1] to reveal the improvement of the system against the previous work, by utilizing the ATE for Mocap datasets in Table II and the start-to-end error for the other datasets in Table I. From the results, the proposed system made a more accurate and robust estimation of the trajectories with no failure in very challenging situations.

Dataset	ROVO [1]	ROVINS	Total Length
Typical Hand-held	0.27	**0.17**	27 m
Spinning Hand-held	X	**0.11**	30 m
Shaking Hand-held	1.57	**0.37**	52 m
Dynamic Objects	1.93	**1.64**	42 m
Spinning In-Out	2.82	**1.00**	170 m
Large-scale In-Out	X	**11.26**	770 m

TABLE I: The start-to-end error evaluation results. In all datasets, ROVINS shows better performance than ROVO [1].

1) Start-to-end Error Evaluation: In the Hand-held datasets, the camera rig intermittently and quickly rotated, creating motion blur for short periods of time. Motion blurs make the feature tracking difficult, one of the main reasons degrading the performance of vision-based pose estimation algorithms. Conversely, VIO can utilize inertial information when the feature tracking is unstable (i.e., when the number of successfully tracked features decrease), resulting in a more accurate and robust performance, as validated in the experimental results of ROVO and ROVINS in Table I. Taking a close look at the results, this advantage was more evident in the other two Hand-held sequences, where the cameras have been shaken more violently or with longer spin. Although such extreme sequences lead to significant motion blur degradation and feature tracking failure, ROVINS was able to maintain a good pose estimation performance.

The Dynamic Objects sequence was captured where many moving people continuously occluded the background scene and wiped out the observed landmarks, as well as when the camera was moved drastically, generating a motion blur. Such an environment also induced the difficulty for long feature tracking, which is a critical element in ROVO as it solely depends on the visual information. ROVINS is not exempted from this problem, however, it showed a better performance as the outliers have been rejected by the predicted poses from IMU measurements. The most complex and challenging datasets for both ROVO and ROVINS were the Spinning In-Out and Large-scale In-Out. ROVO particularly suffered from the drift problem in the Spinning In-Out dataset, when there was an insufficient number of stereo-matched inliers to estimate the metric scale of poses. Also, when the camera was outside the building, there was a drastic change in illumination that dipped the number of successfully tracked and stereo-matched inliers for a moment, and drifted the scale thereby degrading the overall accuracy. Given the same scenario, ROVINS robustly maintained the correct trajectory because the IMU constraints can correct the metric scale during optimization. In Large-scale In-Out datasets, another drastic illumination change occurred when entering the building from the outside. At this time, as shown in Fig. 1, ROVO failed before reaching the end point with a significant error for the large illumination changes between the indoor and outdoor environments, whereas ROVINS correctly arrived near the start point(yellow) along the ground-truth trajectory.

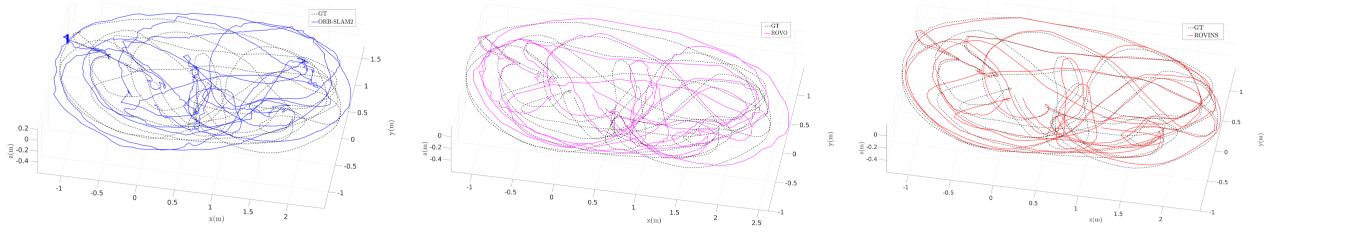
Finally, ROVINS stopped near the start point of full trajectory, showing about 98.5% start-to-end accuracy. These results demonstrated the superior accuracy and robustness of the proposed system.

Dataset	ORB2 (no loop)	ORB2 (loop)	ROVO	ROVINS	Total Length
Mocap0	0.61	0.55	0.15	**0.08**	63 m
Mocap1	0.50	0.51	0.17	**0.07**	61 m
Mocap2	0.46	0.44	0.19	**0.09**	62 m
Mocap All	0.55	0.52	0.34	**0.15**	183 m

TABLE II: Quantitative comparison between the algorithms using ATE$_{trans}$. ORB-SLAM2 with and without loop closing are specified as ORB2(loop) and ORB2(no loop), respectively.

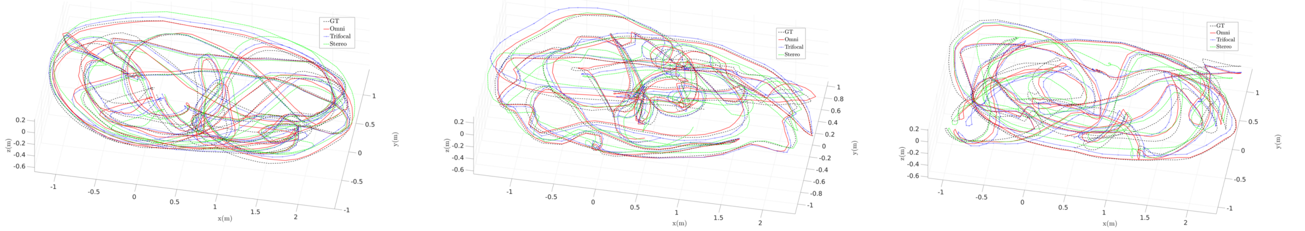
2) RMS Error Evaluation: This section discusses the evaluation of the proposed system and its comparison with the other systems in terms of specifically, the motion-captured Mocap datasets with ground-truth poses. These datasets were captured in a poorly textured room in three challenging motions scenarios. From Mocap0 to Mocap2, the camera motion was increased and changed more rapidly to induce motion blurs, which are a typical challenge for vision-based SLAM systems. To show the overall superiority of the proposed system, it was compared against other state-of-the art systems. However, as there was no public SLAM and VIO systems to support the omnidirectional camera system, ORB-SLAM2 [7] was chosen instead, by utilizing the rectified 120°stereo images generated from the datasets as inputs, as in [27]. The generated stereo images consisted of four pairs: (0,1), (1,3), (2,0), and (3,2). They were run and the best result was selected for comparison.

In addition, an attempt was made to compare VI-ORB [13] and VINS-FUSION [12], [25]; however, no public implementation was made for VI-ORB to run the datasets. In the case of VINS-FUSION, VO failed when the camera moved fast, causing the VIO divergence in the early stages of all datasets due to the poor quality of visual SFM, which is important for the VIO initialization. This continuously



(a) ORB-SLAM2 **(b)** ROVO **(c)** ROVINS

Fig. 6: Comparative results in the Mocap0 dataset. ORB-SLAM2, ROVO, ROVINS, and ground-truth are indicated in blue, magenta, red, and black line, respectively The estimated poses of ROVINS are smoothly aligned to the ground-truth.



(a) Mocap0 **(b)** Mocap1 **(c)** Mocap2

Fig. 7: Qualitative results in the Mocap0, 1, and 2 datasets. ROVINS, ROVINS$_{Tri}$, ROVINS$_{St}$, and ground-truth are indicated in red, blue, green, and black lines, respectively. The estimated poses of all three methods smoothly follow the ground-truth without any bumpy section.

Dataset	ROVINS	ROVINS$_{Tri}$				ROVINS$_{St}$				Total
		(0,1,2)	(1,3,0)	(2,0,3)	(3,2,1)	(0,1)	(1,3)	(2,0)	(3,2)	Length
Mocap0	**0.08**	0.17	0.18	0.09	0.12	0.13	0.19	0.18	0.18	63 m
Mocap1	**0.07**	0.10	0.11	0.09	0.09	0.12	0.13	0.11	0.16	61 m
Mocap2	**0.09**	0.09	0.13	0.12	0.15	0.13	0.20	0.14	0.15	62 m
Mocap All	**0.15**	0.17	0.19	0.23	0.20	0.26	0.26	0.25	0.28	183 m

TABLE III: Ablation study on the different numbers of camera settings. ROVINS$_{Tri}$ and ROVINS$_{St}$ are the results of trifocal and stereo settings. The values in the parenthesis are the camera IDs used in the experiments.

happened even after tuning the parameters (the number of features, keyframe parallax, etc), and thus, leading to a decision to compare only with the ORB-SLAM2, after the number of features has been set to 5000, the minimum settings that should not fail for the whole sequences. During the evaluation, the loop closing modules of ORB-SLAM2 were manually turned on and off for measurement of the accuracy of both VO and SLAM.

Table II displays a comparison of the ATE$_{trans}$ of both ROVO and ROVINS, and ORB-SLAM2 in the Mocap datasets. ROVINS was overall 2× more accurate than ROVO, and 7× more accurate than ORB-SLAM2 (even with the loop closing), in terms of the RMS error. With the absence of drastic illumination changes, all methods ran well without any catastrophic failures. However, for the vision trajectory, only systems (ROVO and ORB-SLAM2) became bumpy as the camera started to move very fast, mainly because such drastic camera changes would induce motion blurry images that degrade the overall performance of the vision-based system. In contrast, ROVINS demonstrated a smooth trajectory estimation regardless of the camera motion (Fig. 6). These results validated that ROVINS can operate in an accurate and robust manner even in challenging situations.

E. Simple Evaluation of Feature tracking

This section discusses the simple evaluation conducted for the proposed feature tracking with prediction. To evaluate the effect of the present algorithm, the average number of inliers and ATE$_{trans}$ was compared between the ROVINS with and without the IMU-aided feature tracking, labelled here as ROVINS(w/ pred) and ROVINS(w/o pred), respectively. Based on the results in Table IV, ROVINS(w/ pred) performed slightly better than ROVINS(w/o pred) in both measures, which confirms that the proposed algorithm improves the number of inliers to helps boost the overall performance of the system.

Dataset	avg. # of inliers		ATE$_{trans}$(m)	
	w/o pred	w/ pred	w/o pred	w/ pred
Mocap0	275.45	**282.15**	0.10	**0.08**
Mocap1	228.25	**232.27**	0.09	**0.07**
Mocap2	250.00	**255.04**	0.10	**0.09**
Mocap All	248.56	**259.30**	0.17	**0.15**

TABLE IV: Ablation study for the IMU-aided feature tracking.

F. Effectiveness of the Omnidirectional Setup

To confirm the effectiveness of the omnidirectional setup, an experiment was conducted by varying the number of cameras. Here, four ultra-wide FOV cameras were employed to maximize the stereo overlapping regions. However, note that the proposed algorithm also works without full 360° coverage. However, the drawback with just two or three cameras for motion estimation are the missing parts in the scene and the features in such area that cannot be tracked. Thus, it would be logical to assume that the four-view system would perform better than a-fewer-camera setup. Table III and Fig. 7 show a comparison of the three different versions of the proposed algorithms namely, ROVINS, ROVINS$_{Tri}$, and ROVINS$_{St}$, using four(omnidirectional), three(trifocal), and two(stereo) cameras, respectively. For comparison, the performance of all possible configurations of ROVINS$_{Tri}$ and ROVINS$_{St}$ were given. For example, ROVINS$_{Tri}$ was aligned with four configurations: (0,1,2), (1,3,0), (2,0,3), and (3,2,1). ROVINS with an omnidirectional view is the most accurate in all tests, even after comparisons with the best candidates of ROVINS$_{Tri}$ or ROVINS$_{st}$. These results verified that the proposed omnidirectional setup is advantageous in the overall accuracy and robustness of the system.

VI. CONCLUSIONS

This study introduced an omnidirectional multi-view visual-inertial odometry algorithm called ROVINS. Unlike conventional VIO systems, ROVINS fully utilizes a 360°-FOV with stereo overlaps, which drastically improves the accuracy and stability of pose estimation drastically. Compared to ROVO, an omnidirectional multi-view VO algorithm, the inertial information in ROVINS are seamlessly integrated into the pose optimization framework via formulations of the relative motions from IMU as soft-pose constraints. The biases of the IMU are robustly estimated from rich visual information, thereby significantly improving the quality of inertial measurements. The feature tracking also benefits from this formulation, as the initial locations of the features in the subsequent frames are updated by the estimated IMU motion. For extensive experimental validation, many challenging sequences were selected covering fast and abrupt motions, severe illumination changes between the indoor and outdoor environments, and many dynamic objects near the camera. Based on the experimental results, the proposed VIO algorithm is very effective and significantly improves the motion estimation performance.

References

[1] H. Seok and J. Lim, "Rovo: Robust omnidirectional visual odometry for wide-baseline wide-fov camera systems," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6344–6350.

[2] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 15–22.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[4] D. Cremers, "Direct methods for 3d reconstruction and visual slam," in *Proceedings of the IEEE International Conference on Machine Vision Applications (MVA)*, 2017, pp. 34–38.

[5] M. Yokozuka, S. Oishi, S. Thompson, and A. Banno, "Vitamin-e: Visual tracking and mapping with extremely dense feature points," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9641–9650.

[6] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 40, no. 3, pp. 611–625, 2017.

[7] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[8] R. Wang, M. Schwörer, and D. Cremers, "Stereo dso: Large-scale direct sparse visual odometry with stereo cameras."

[9] D. Caruso, J. Engel, and D. Cremers, "Large-scale direct slam for omnidirectional cameras." in *IROS*, vol. 1, 2015, p. 2.

[10] P. Liu, L. Heng, T. Sattler, A. Geiger, and M. Pollefeys, "Direct visual odometry for a fisheye-stereo camera," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 1746–1752.

[11] H. Matsuki, L. von Stumberg, V. Usenko, J. Stückler, and D. Cremers, "Omnidirectional dso: Direct sparse odometry with fisheye cameras," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3693–3700, 2018.

[12] 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.

[13] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular slam with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[14] 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)*. IEEE, 2018, pp. 2510–2517.

[15] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2016.

[16] E. Hong and J. Lim, "Visual-inertial odometry with robust initialization and online scale estimation," *Sensors*, vol. 18, no. 12, p. 4287, 2018.

[17] G. Hee Lee, F. Faundorfer, and M. Pollefeys, "Motion estimation for self-driving cars with a generalized camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2746–2753.

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

[19] P. Liu, M. Geppert, L. Heng, T. Sattler, A. Geiger, and M. Pollefeys, "Towards robust visual odometry with a multi-camera system."

[20] K. Eckenhoff, P. Geneva, J. Bloecker, and G. Huang, "Multi-camera visual-inertial navigation with online intrinsic and extrinsic calibration," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3158–3164.

[21] L. Heng, B. Choi, Z. Cui, M. Geppert, S. Hu, B. Kuan, P. Liu, R. Nguyen, Y. C. Yeo, A. Geiger, *et al.*, "Project autovision: Localization and 3d scene perception for an autonomous vehicle with a multi-camera system," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4695–4702.

[22] S. Houben, J. Quenzel, N. Krombach, and S. Behnke, "Efficient multi-camera visual-inertial slam for micro aerial vehicles," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1616–1622.

[23] B. D. Lucas, T. Kanade, *et al.*, "An iterative image registration technique with an application to stereo vision," 1981.

[24] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[25] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," 2019.

[26] C. Won, J. Ryu, and J. Lim, "Sweepnet: Wide-baseline omnidirectional depth estimation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6073–6079.

[27] ——, "End-to-end learning for omnidirectional stereo matching with uncertainty prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[28] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A flexible technique for accurate omnidirectional camera calibration and structure from motion," in *Proceedings of the IEEE International Conference on Computer Vision Systems (ICVS)*, 2006, pp. 45–45.

[29] S. Urban, J. Leitloff, and S. Hinz, "Improved wide-angle, fisheye and omnidirectional camera calibration," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 108, pp. 72–79, 2015.

[30] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4304–4311.

[31] N. Yang, R. Wang, X. Gao, and D. Cremers, "Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2878–2885, 2018.