OmniVO: Toward Robust Omni Directional Visual Odometry With Multicamera Collaboration for Challenging Conditions

ZEESHAN JAVED AND GON-WOO KIM, (Member, IEEE)
Department of Intelligent Systems and Robotics, Chungbuk National University, Cheongju, Chungbuk 28644, South Korea

Corresponding author: Gon-Woo Kim (gwkim@cbnu.ac.kr)

This work was supported in part by the Ministry of Science and ICT (MSIT), South Korea, under the Grand Information Technology Research Center Support Program Supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP), under Grant IITP-2021-2020-0-01462; and in part by the Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry (IPET) Grant through the Korea Government [Ministry of Agriculture, Food, and Rural Affairs (MAFRA)] under Grant 322039-3.

ABSTRACT With the recent developments in computer vision, vision-based odometry plays an increasingly important role in the field of autonomous systems. However, using traditional visual odometry or simultaneous localization and mapping (vSLAM) only performs better in simple environments with obvious structural features. Visual odometry may easily fail in a complex environment due to the sparsity of stable features, sensor failure, and extreme weather conditions or sunlight problems. Monocular camera-based traditional algorithms are highly affected by these issues, leading to stability and reliability problems. Therefore, to deal with such issues, a vision-based Omni-directional odometry based on multi-camera collaboration is proposed. The development of feature-based omnidirectional odometry and feature prioritization to limit the computational complexity offered by multiple cameras are major contributions. Firstly, the multi-camera state perception model is developed based on the spherical camera, which guarantees an accurate transformation from the camera to the spherical coordinate system. The feature detection and tracking are performed in the images of the individual cameras in a parallel thread. The features tracking across the different cameras prevent failures, however, it also adds extra computational complexity for the pose estimation and optimization module. To budget the feature distribution, a feature prioritization algorithm is proposed to limit the number of features. Furthermore, the multi-view pose refinement module further reduces the complexity of the system. The feature prioritization helps to maintain the smaller set of tracked features with similar accuracy and less computational complexity. Finally, the pose is estimated in a spherical coordinate system by projecting all the successfully tracked key points to the sphere with the help of the omnidirectional perception model. To validate the proposed method, the data set is collected from the outdoor environment with ground truth provided by high-accuracy GPS. Detailed qualitative and quantitative evaluations are performed, which show that the proposed algorithm improves the position accuracy by about 40%-60% as compared with state-of-the-art methods with limited computation time.

INDEX TERMS Panoramic camera, multi-camera, visual odometry, vSLAM, GNSS/INS, monocular camera.

I. INTRODUCTION
Simultaneous Localization and Mapping (SLAM) is an algorithm to solve the problem of simultaneously constructing maps of the environment while localizing the robot in an unknown environment [1], [2]. Over the past decade, the vision sensor has gained great attention in developing vision-based SLAM algorithms because of its rich visual information and low price. Many popular algorithms have been published in the literature and recognized widely, such
as ORB-SLAM2 [3], LSD-SLAM [4], PTAM [5], DSO [6], [7] based on either a monocular camera, stereo camera or RGB-D sensor. Therefore, vSLAM can be divided into mono-
SLAM, stereo-SLAM, and RGB-D-SLAM.

SLAM performance is highly affected by the environmental conditions, such as dynamic objects or structure-less complex environments. In the case of a dynamic environment, the dynamic object can be detected and tracked [8], [9] to classify dynamic objects to develop a robust dynamic SLAM [10]. Moreover, most of the visual SLAM systems discussed above are based on the single-camera system. The single-camera-based visual SLAM algorithms suffer in texture-less complex regions such as parking lots or off-road environments. The situation worsens in off-road environments, where rough roads, direct sunlight, and vegetation occlusion complicate stable tracking [11]. These problems can be solved by incorporating large field of view vision sensors, such as panoramic/omnidirectional cameras. These sensors provide 360° scenes in one shot and are used in many state-of-the-art applications such as map building for street view virtual reality and autonomous system navigation [12]. To address the above challenges, there are useful systems in the literature utilizing fisheye cameras for odometry or SLAM. [13], [14], [15], and [16] and some feature-based SLAM systems [17], [18]. Other than multi-camera methods, the sensor fusions algorithm can provide better results in a complex environment. The visual-lidar fusion [19], and wheel encoder methods [20], [21] for odometry provides better results in feature-less region.

The use of a wide field of view provides more structural information, but at the same time, the image is largely distorted, which complicates the reliable feature matching process for descriptor-based matching algorithms. Furthermore, adding more visual information in the case of a multi-camera setup adds a computational complexity to the algorithm. Therefore, to address the discussed challenges, an efficient optimization-based visual odometry algorithm is proposed. The proposed method successfully solves the problems offered by a limited field of view in challenging conditions. Furthermore, the problems offered by the fisheye images in multi-camera setup algorithms are handled effectively to provide a reliable pose estimation. The proposed method uses a spherical camera model for multi-camera collaboration. The features are extracted from each camera in a parallel thread. The KLT-based tracking algorithm is used to track features independently for each camera in a parallel thread. Once the feature detection and tracking is finished, the feature prioritization algorithm selects features from each view, and outliers are removed at the same time. The reliable features are projected into a spherical coordinate system and the pose is estimated by using spherical geometry. The 8-point method with RANSAC is used to estimate the pose, and optimization is performed based on triangulated feature points. Importantly, optimization with too many triangulated points increases the computational complexity. Therefore, one view is selected to have more inlier points for optimization.

The main contribution of this work is as follows:
- The complete feature-based visual odometry algorithm is developed for a multi-camera panoramic system with collaboration of multiple cameras by using a spherical state perception model for challenging situations.
- The feature selection and outlier removal scheme is proposed to budget the features in order to limit the computational performance. The multi-view optimization scheme is used based on the probability of features for each of camera to limit the 3D landmark for reducing optimization computational performance.
- Importantly, the extensive evaluation is performed with different camera settings and comparison with other methods. The data set is collected from the outdoor environment that covers many challenging situations with ground truth provided by RTK GPS.

The remainder of this paper is structured as follows: Related work is reviewed in Section II. A multi-camera panoramic perception including fisheye camera calibration and the multi-camera setup model is discussed in Section III. The detailed explanation of the proposed visual odometry is explained in section IV. The experimental evaluation is presented in section V. Sections VI and VII present a discussion and conclusions, respectively.

II. RELATED WORK

In recent years, the panoramic vision system is widely used by many researchers to obtain more visual information for...
use in various environments. In literature, three types of panoramic vision systems are used such as the catadioptric vision system [22], [23] is widely used for 2D motion estimation, fisheye cameras [24] are designed for multiple fisheye systems and multi-camera vision systems [17]. The most popular and widely used panoramic camera is the multi-camera vision, because of their relatively small distortion and rich environment information than pinhole or less field of view (FoV) camera models. The multi-camera rig system consists of separate fisheye lenses that can be stitched to a panoramic image using spherical projection. Panoramic or multi-camera system has been used in several studies of SLAM [25], [26], [27], [28] and visual odometry [12]. These methods has been used to perform feature-based visual odometry, visual-inertial odometry, direct odometry, and SLAM. Seok and Lim used a multi-camera system to design omnidirectional visual odometry for a wide baseline [16]. They proposed the projection model to make wrapped images for reliable feature tracking, the hybrid feature matching model is used by tracking intra-view with KLT tracking and the descriptor-based matching for inter-view. The pose is estimated with a proposed multi-view P3P RANSAC-based algorithm and the online camera calibration is proposed in optimization to overcome the deformation and motion of the camera. They further extended their work to omnidirectional visual-inertial navigation to overcome the shortcomings that affect visual sensors, such as fast motion and sudden illumination changes [29]. The soft relative pose constraints from the inertial sensor are added to pose optimization to deal with blind motion estimation, and the visual features in tracking are initialized based on estimated velocity from prediction results.

Similarly, the multi-camera vision system is also integrated to direct sparse odometry [30]. In the case of visual SLAM for panoramic or multi-fisheye camera systems, some algorithms have been published in the literature. The PanoSLAM [17] proposed a complete feature-based SLAM system with a multi-fisheye panoramic camera. They proposed a fisheye camera calibration method by combining an equidistant projection model and a trigonometric polynomial for high-accuracy transformation of the fisheye point to the corresponding panoramic point. Furthermore, the bundle adjustment with specific backpropagation error function has been developed for panoramic cameras. The experiment conducted on a self-collected and open-source dataset, demonstrated better performances in challenging scenes. The omni-SLAM [31] is also a feature-based SLAM system designed for multi-camera setup by adding deep learning for depth estimation of the panoramic image.

SLAM system based on monocular camera suffers from a small field of view that ultimately causes the failure of the system in challenging situations such as strong sunlight, abrupt motion changes, sharp turns, and feature-less regions in the surroundings. While multi-camera system methods track all the features points and perform optimization with each camera’s information, the camera information associated with all views helps to recover failure problems but adds the extra computational complexity of the algorithm. Therefore, we proposed an effective visual odometry algorithm for the omnidirectional camera with multi-camera collaboration. The feature prioritization and multi-view pose optimization are introduced in the traditional system that effectively improves the performance of the algorithm. The distortion in images affects the feature matching capabilities of the odometry algorithm. Therefore, the KLT tracking algorithm is used in a multi-threaded way for each camera. The complete hardware system is designed and data-set is collected from the outdoor environment, including parking, main road environment, and narrow road region across the building. The detailed experimentation suggests that the proposed method can considerably improve the performance and reliability in complex regions where single camera-based methods has difficult to track.

III. MULTICAMERA PANORAMIC PERCEPTION MODEL

To obtain a high-precision SLAM or odometry system, a suitable camera model is important for the projection relationship...
between 3D world points and 2D image points. Similarly, for omni-directional vision, the relationship between each camera and body frame needs to be developed. In this section, the multi-camera panoramic imaging model and calibration are described in detail.

A. CAMERA IMAGING MODEL

The multi-camera system used in this experiment is based on fish-eye lenses with a slightly fixed translation between each other and the panoramic rig center. The camera multi-camera calibration provides the relationship between each camera and its camera center for spherical projection. The transformation can be used to project each camera point to its spherical coordinates and to its respective panoramic coordinates. This paper utilized calibration information to develop the relationship between camera coordinates and their equivalent spherical coordinate system.

\[
I_d = f \left( \frac{\tan^{-1} \left( \frac{I_r}{R} \right)}{R} \right) I_r + r(\theta, \phi) u_r + \Delta r(\theta, \phi) u_r
+ \Delta_\theta(\theta, \phi) u_\phi + \begin{bmatrix} t_1 & t_2 & 0 & 0 \end{bmatrix} I_r
\]

where \(I_r = (x_r, y_r)\), are the rectified coordinates, \(f\) is the focal length, \(u_r = (1, 1)\) is unit vector in the radial distortion while \(t_1, t_2\) are tangential distortion. In the equation \(\Delta_r(\theta, \phi)\) are the radial tangential distortion terms and can be derived as explained in [17]. The equation 1 shows the equidistant projection model of the fish-eye camera lens instead of other different projection models such as stereographic, isometric, and orthographic. The equation 1 with an equidistant project model is widely used for fish-eye camera with a wide angle of view to minimize tracking loss in fast movement. Since this fish-eye camera was designed for a wide angle of view from the lens design, it is difficult to model it with a general perspective model. The equidistant projection model is the most used model in this fish-eye camera. In the perspective projection model, the angle of incoming and outgoing light is the same, whereas in the equidistant projection model, the angle of incoming light and the distance from the principal axis are linear.

B. MULTICAMERA SYSTEM PERCEPTION MODEL

Normally, a monocular camera is associated with a world coordinate system with an extrinsic transformation as shown in Figure 4. The Figure 4 depicts the relation between world coordinates to five camera of ladybug and with center of panoramic camera. In case of a multi-camera panoramic system, the world coordinate system is attached to its panoramic coordinate system based on a fixed rigid transformation from camera to panoramic sphere. Based on equations (2) and (3) the multi-camera panoramic space perception model can be derived, which shows the relationship between the 3D point \(p_w\) in space, 2D \(p_i\) in pixel coordinate and 3D \(p_s\) in sphere coordinate system similar to [17].

\[
P_{\text{cam}} = R_k[P_w] + T_k
\]

Equation 2, shows the general procedure for projecting 3D world point to corresponding image plane with known extrinsic calibration parameters. The \(R_k\) is the rotation and \(T_k\) is the translation parameters obtained from extrinsic calibration of the each lens of ladybug camera. The \(k\) represent number of cameras, \(k = 0, 1, 2, 3, 4\) total five cameras in the ladybug case. Similarly, the projection of 2D image point to its 3D spherical coordinates can be performed as:

\[
P_s = sR_iK_i(x)p_{\text{cam}} + T_i
\]

\[
||p_i|| = r^2
\]

The 2D images point are projected to the 3D sphere by equation 3 with known calibration parameters from each camera lens to the sphere. The \(R_i\) are the rotation parameters for each camera, \(T_i\) are the translation parameters for each camera and \(s\) is scale factor. The \(K_i(x)\) is the calibration function used to project 2D distorted points to their rectified coordinates. Similarly, the \(k\) shows number of cameras \(i = 0, 1, 2, 3, 4\), total five cameras in the ladybug camera system. Equation 4 is used to limit the panoramic point to fix sphere with radius \(r\). The camera model used in this work represents multi-camera rig with fixed slight offset between fisheye camera center \(C\) and the panoramic camera center \(S\) as shown in Figure 3. The camera model used in the paper is based on the multi-camera rig for omnidirectional cameras. The multi-camera is designed with separate five fish-eye lenses with a slight fixed offset between each camera center and the panoramic camera center. The above procedure from equation 3 and 4 express the col-linearity between camera center, pixel coordinate, and panoramic center. Figure 5 shows the camera images obtained from an omnidirectional camera. The results of image rectification and panoramic image generation from distorted camera images are shown in Figure 5. The top row shows the raw images covering 360 horizontal views, while the rectified camera images are shown in the middle row and projected into a panoramic image. The raw images have lens distortion that needs to be corrected before pose estimation. However, our algorithm extracts features from the raw image and then un-distorts andrectifies key points before sphere projection. The bottom row shows the rectified images obtained through the process of
FIGURE 4. The relationship between world coordinates and spherical center of omni-directional camera.

FIGURE 5. The sample images from Ladybug 3 camera. The (a-c) The top rows show the original distorted images from each of camera covering 360 horizontal view. The second rows shows the rectified individual camera images, while the last row shows the panoramic image.

rectification with the obtained calibration parameters. The panoramic image is generated by projecting each image to the sphere center of radius 20m (Ladybug sphere radius), the corresponding panoramic image is shown in the bottom row of Figure 5.

IV. PROPOSED VISUAL ODOMETRY SYSTEM

This paper proposed a robust omnidirectional visual odometry with a multi-camera panoramic system for a challenging environment. The calibration is performed based on the method explained in Section III, and then an omnidirectional perception model is developed based on the spherical camera system. The spherical perception model is used to perform pose estimation and optimization in the algorithm. The detailed overview of the proposed algorithm is shown in Figure 2. Instead of using a panoramic image, the individual fish-eye image of each camera is used for feature detection and tracking. The features are detected and tracked on the parallel thread for every single camera. To balance the computational complexity offered by a large number of features from all cameras, feature prioritization is performed.

The feature prioritization module’s job is to budget the number of features for tracking. A large amount of distortion present in images also complicates the KLT-based feature tracking and matching. Therefore, the outliers are identified and removed in the feature prioritization module. The limited number of selected features are projected onto the sphere based on the spherical camera geometry developed in section III. The pose is then estimated based on the monocular technique in a spherical coordinate system by using an 8-point algorithm. The pose is refined based on triangulated points obtained from the previous motion. The open-source library g2o [32] is used to carry out the optimization tasks. The triangulated points from all the cameras are still in large enough amounts so that they can increase the computational complexity of the algorithm. Therefore, multi-view optimization is adopted based on the highest triangulated camera features from all the cameras. The detailed procedure of the proposed algorithm is explained in the next sections.

A. FEATURE DETECTION AND TRACKING

The features are extracted from individual fisheye images for ladybug cameras. The Features from accelerated segment test (FAST) feature detector [33] is used to extract features from each image. The feature points are detected from high-resolution images for each camera view. FAST is a real-time approach for interest point detection to be used in mobile robot applications such as SLAM. Unlike SIFT, FAST has only one feature per keypoint. However, SIFT
has multiple feature points at one key point. The FAST can
differentiate the central pixel with pixels located within a
circle around it. The corner detection in the FAST algorithm
is based on the Harris corner method. To detect key points,
the threshold value is intentionally set low, and only top \( N \) key
points are extracted. The FAST feature is detected in the pyra-
mid method because it does not support multi-scale functions.
The detected features in each frame can be matched with the
previous frame either by descriptor-based matching or feature
tracking. The images in our setup contain distortion, therefore
the descriptor-based matching may have a wrong matching
result. Therefore, feature tracking is performed by KLT [34]
tracking algorithm for each image. The KLT algorithm works
in two steps; it locates the trackable features in the initial
frame and then tracks the detected feature in the next frame
using displacement that minimized the sum of differences.
The feature detection and tracking modules are implemented
in a multi-threaded application to balance the computation
offered by a multi-camera setup. The feature detection and
tracking results for frontal view camera only are shown in
Figure 6.

B. FEATURE SELECTION AND OUTLIER REMOVAL

Multiple cameras have advantages in the case of complex
regions. However, in the case of a structured environment, too
many feature points increase the computational complexity
of the system. It is important to budget the overall feature
distribution for tracking. To limit the initial feature detection
may not be a good solution because it may degrade the
overall pose estimation accuracy. Furthermore, fisheye image
distortion affects the quality of feature matching/tracking.
The wrongly tracked feature points affect the overall accuracy
of the algorithm. Therefore, the feature prioritization and
outlier detection methods are added to the conventional visual
odometry algorithm. The algorithm 1 shows the step by step
procedure of the feature prioritization module. In the case of
feature prioritization, for every tracked feature point in each
image, the 30 \( \times \) 30 window is selected. The sum of squared
differences (SSD) score for neighboring pixels is computed
for each feature point in each window for the current image
\( I_t \) and the next image \( I_{t+1} \).

\[
s = \sum_{(u,v) \in I} (I_{t+1}(u,v) - I_t(u,v))^2 \tag{5}
\]

where \( s \) is the similarity score, the features are selected or
removed as an outlier based on the similarity score. If the
SSD score is high that feature is treated as an outlier and
removed from further tracking. If the SSD score is minimum
that feature is selected for tracking. The remaining features
point are added to the queue bucket, they are added to the
tracking module if the distribution of the overall feature is not
enough for the pose estimation module. The algorithm takes
features from selected windows, computes the SSD score and
the feature is selected for pose estimation and optimization
based on the similarity score. The feature with a low SSD
score is selected and with a higher SSD score is removed as
an outlier, the other features with a low SSD score are
added into the queue and only included if an overall number
of features are less than the threshold. The methods help to
remove wrongly tracked feature influence considerably and
the features are budgeted for the next modules to reduce com-
putational complexities. While the features are distributed
evenly over the 360 regions of the environment to improve
overall pose estimation accuracy. Different from the existing
bucketing scheme used by many odometry algorithms, the
proposed method uses SSD score to accept or reject the
feature instead of selecting only one feature in the region.

\begin{algorithm}
\caption{Feature Prioritization}
\begin{algorithmic}
\Require Current Features \( f_t \)
\Require Previous Features \( f_{t+1} \)
\Ensure Inlier \( f_t \), Outlier \( f_{t+1} \)
\For {Each Images}
\For {Each Camera}
\State \( P_t = \text{GetPatches}(I_t) \)
\State \( P_{t+1} = \text{GetPatches}(I_{t+1}) \)
\For {Each Feature in \( P_t, P_{t+1} \)}
\State \( N_t = \text{SearchNeighbors}(f_t) \)
\State \( N_{t+1} = \text{SearchNeighbors}(f_{t+1}) \)
\State \( score = \text{computesSSD}(N_t, N_{t+1}) \)
\EndFor
\If {\( score \geq \text{Threshold} \)}
\State Inlier \( f_t = f_t, f_{t+1} \)
\Else
\State Outlier \( f_t = f_t, f_{t+1} \)
\EndIf
\EndFor
\EndFor
\Return Inliers, Outliers
\end{algorithmic}
\end{algorithm}

C. POSE ESTIMATION

The successfully tracked features are then used for pose
estimation. The algorithm uses the spherical camera model
to estimate the pose between adjacent frames. The detailed
procedures of the pose estimation module is presented in
algorithm 2. The successfully tracked feature points are
projected onto the sphere as explained in section III. Each
camera is linked with panoramic camera center with fixed
transformation. These spherical projected points for consec-
utive frames are used to estimate the ladybug pose. The
spherical points \( p_{i+1}, p_{i+2} \) in two adjacent panoramic images
coordinate system satisfy the epipolar geometry [17] and
can be easily solved by 8 or more points. For every two
frame combinations, the fundamental matrix is recovered
using an eight-point algorithm inside a RANSAC loop. The
eight-point algorithm consists of three steps. Firstly, homoge-
neous linear equations are formulated, the equation is solved
by taking an account that they may have not an exact solution.
The constrained for fundamental/essential matrix are shown by equation 6.

\[ x^TEx = 0 \]  

(6)

The fundamental matrix is decomposed by singular value decomposition (SVD). The SVD can be described as follows:

\[ E = U \sum V \]  

(7)

where \( U \) and \( V \) are orthogonal matrices and \( \sum \), a is singular value matrix. The essential matrix decomposition will lead to the following solution of rotation and translation vector.

\[ R_1 = UW^T V^T, R_2 = U WV^T \]  

(8)

\[ t_1 = UW \sum V^T, t_2 = UW \sum U^T \]  

(9)

where \( W \) are the rotation matrix, and \( R \) and \( t \) are the rotation and translation vector respectively. Now, this decomposition leads us to a four solutions. The three solution points are removed and one is taken based on the triangulated 3D points lying in front of the camera observation. The equations 6, 7, 8, and 9 are use to obtained the pose of vehicle. The initial pose obtained is then further optimized based on the optimization algorithm.

D. MULTIVIEW POSE REFINEMENT

The estimated pose is then refined based on pose-only optimization of triangulated 3D landmark points. The initial pose between two adjacent spherical frames is used to triangulate the 3D landmarks for the sphere. The goal of the optimization is only to refine a pose between frames. The optimization algorithm is computationally demanding, a large number of landmarks triangulated from 2D feature points obtained from different camera views increases the computational complexity of the algorithm. Therefore, an effective multi-view optimization algorithm is proposed to budget landmarks for optimization. Instead of using all landmarks from all cameras for optimization, the highest landmark associated with any single camera is selected for pose refinement. Firstly, all feature points are triangulated separately for each camera with the initial pose value. Then the camera with more triangulated feature points is selected and their corresponding spherical points are used for optimization. If landmarks from one view are not enough for optimization, landmarks triangulated with other cameras are added for optimization.

The g2o is utilized to solve the optimization problem based on the panoramic camera model similar to [12] and [17], and only pose is optimized. The optimization problem is formulated as:

\[ \min \frac{1}{2} \sum_{k=1}^{n} \left\| X_k - \frac{r}{\| (\exp(\epsilon)X_w) \|} (\exp(\epsilon)X_w) \right\| \]  

(10)

where \( X_w \), and \( X_k \) are the 3D world and spherical coordinate of feature point respectively. The \( \epsilon \) represent camera pose donated by lie Algebra. The final goal is to minimize the cost function to refine the pose. Equation 10 represent the bearing vector error metric for panoramic camera as it performed best from other error metrics such as image error, tangent error, and unit plane error [12].

Algorithm 2 Pose Estimation and Multi-View Optimization

Data: Input Feature point \( f_i, f_{i+1} \)

Results: Optimized Pose

for Each Features Set do

\[ S_t = \text{ProjectkeypointToSphere}(f_i) \]

\[ S_{t+1} = \text{ProjectkeypointToSphere}(f_{i+1}) \]

end for

for Sphere Points do

\[ P_t = \text{GetPointsRANSAC}(S_t) \]

\[ P_{t+1} = \text{GetPointsRANSAC}(S_{t+1}) \]

\[ F = \text{ComputeFundamentalMatrix}(P_t, P_{t+1}) \]

if EnoughInliers then

 Break;

else

 Continue;

end if

end for

\[ E = \text{GetEssentialMatrix}(F) \]

\[ R, t = \text{DecomposeSVD}(E) \]

Start Multi-view Optimization

for Each Camera do

\[ X_{3d} = \text{Triangulate}(S_t) \]

end for

\[ l = \text{LargestSetForCam}(X_w) \]

\[ X_s = \text{GetSpherePoints}(S_t(l)) \]

\[ X_w = \text{GetLandmarks}(X_{3d}(l)) \]

\[ R_f, t_f = \text{Optimize}(X_w, X_s, R, t) \]

Return: Refined Pose

V. EXPERIMENTS AND ANALYSIS

A detailed experiment has been performed with self-collected data in order to validate the proposed system. The data set is
collected from an outdoor environment with GPS provided ground truth pose. The Average Trajectory Error (ATE translation) is computed to compare the result with state-of-the-art visual methods. The detailed explanation is explained in subsequent sub-sections.

A. HARDWARE
To evaluate the proposed system, we conduct experiments with real-world datasets. The proposed framework is implemented and tested on Hyundai i30 (Hyundai Motor Company, Seoul, South Korea), shown in Figure 1. The platform is equipped with a Ladybug omnidirectional camera, mounted on the center top of the platform. The RTK Novatel GPS is on the left side of the platform with dual antennas setup. The Robot Operating System (ROS) based software is developed to record the dataset with a ROS-provided time-stamp. The GPS data is synchronized with camera data by using a ROS-provided time-stamp. The manual transformation has been calculated between GPS and Ladybug camera. The Ladybug SDK provides the calibration parameters for each lens of Ladybug.

Ladybug is a high-resolution spherical digital camera system with 360-degree coverage at a high-speed interface. The Ladybug3 has six 2-Megapixel cameras with five cameras in a circular rig and one camera is positioned at the top. This helps the system to cover more than 80 percent area of the full sphere, and all the cameras are pre-calibrated to enable high-quality spherical image stitching. The Ladybug3 allows for capturing data at multiple resolutions and as well as different frame rates. Moreover, it also provides the hardware jpeg compression to support a high frame rate. Novatel RTK GPS is a compact, robust, high-precision fully integrated global positioning system. It has multi-frequency, dual-antenna input that allows the receiver to harness the power of Novatel CORRECT® with RTK and ALIGN functionality which makes them very suitable for ground and marine-based systems. The maximum data rate of GNSS is up to 100hz. In the proposed platform, GPS measurements are recorded at 100hz to provide the ground truth trajectories of a dataset that are used to calculate average trajectory error (ATE) for the evaluation of visual odometry and SLAM algorithms.

The three sequences are recorded from the outdoor environment including outdoor parking, outdoor in front of the main building (narrow road), and campus main road as shown in Figure 7. The RTK GPS is used as ground truth for testing and evaluation of the proposed method. In the experiments, the images are captured at full resolution (1616 × 1232). The parking sequence contains more than 400 images captured at 6 fps with a total length of nearly 300 meters and contains the featureless region, especially at turns. Similarly, the other two sequences have lengths of nearly 200 meters and 0.7 km respectively. The sequences are recorded on GPS-supported regions as the environment is urban and the GPS is not able to track satellites in such an environment, while the test sequences are not considered for GPS blocked areas because the evaluation may not be possible without GPS-provided ground truth. The proposed algorithm is implemented on Ubuntu 16.04 with the c/c++ programming language. The evaluation is carried out on a desktop system with Core (TM) i7 CPU @ 3.20GHz specification.

B. VISUAL ODOMETRY EVALUATION
The proposed method uses an omnidirectional camera with five individual cameras placed in a horizontal row with fixed baselines from each other and from the sphere center. The 1000 fast features are extracted from each of five images in a parallel thread. In order to evaluate the proposed system, extensive experiments have been conducted on a real dataset. The RTK GPS is used to quantitatively measure the accuracy by computing the translation Root Mean Squared Error (RMSE) between the estimated poses and the ground truth. The trajectories are aligned using Horn Method [35] and the two error metrics are computed similarly to [36]. The RMSE of Absolute Translation Error and the Rotation Error (RE) with yaw-only orientation. The RMSE of ATE can be calculated as shown in equation 11.

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{n}(t_{i,e} - t_{g,i})^2}{n}}
\]

where \(t_{i,e}\), \(t_{g,i}\) and \(i\) are the estimated and ground-truth translation vector respectively. The total number of observation are denoted by \(n\), while \(i\) denotes the \(i^{th}\) number of observation. Besides RMSE, the minimum, maximum, and standard deviation of the error matrix is also calculated.

C. ABLATIONS STUDY
To evaluate the overall performance of the proposed method, we conducted several experiments with different camera settings. The study is conducted to show the advantage of adding more cameras to pose estimation accuracy and the ability of the system to perform in a challenging situation. Three different combinations are used, namely cam2 (2 cameras), Trifocular (3 Cameras), and the 5 camera setting named OmniVO. For each experiment, 1000 features per frame are extracted. For each camera combination, only the related features are tracked and used for pose estimation and optimization. The resultant output trajectories with different combinations of cameras are shown in Figures 8 and 9. The output trajectory plot of all different combinations with the reference ground truth trajectory is shown in Figure 8, where Figures 6(a), 6(b), and 6(c) show the resulting trajectories for three different data sets, it can be seen from the figure that adding features from different viewing direction makes the system reliable for tracking and increases pose estimation accuracy in complex situations.

The sequence 1 of the experiments is recorded from outdoor parking environment, which normally contains a sufficient number of features due to the large buildings and trees surrounding it, but in the case of sharp turns, a smaller number of features are successfully tracked due to less structural
TABLE 1. Position and rotation accuracy of proposed algorithm with different camera setting. Three different camera combinations Cam2, Cam3 (Trinocular) and Cam5 (Omni-VO) are used for ablation study.

| Sequence /Environment | Camera Setting | F.Recover/T.Frame | ATE Translation (Meters) | ARE Yaw (Degree) |
|-----------------------|----------------|-------------------|--------------------------|------------------|
|                       |                |                   | RMSE | Min | Median | Max | RMSE | Min | S.D |
| Sequence 1 / Parking  | Cam2           | 340/340           | 1.4471 | 0.0076 | 1.2670 | 3.1084 | 0.2457 | 0.0060 | 0.2405 |
|                       | Cam3 (Trinocular) | 340/340         | 1.7480 | 0.0089 | 1.4894 | 3.7145 | 0.4271 | 0.0099 | 0.3710 |
|                       | Cam5 (Omni-VO) | 340/340           | 1.3125 | 0.0042 | 0.7471 | 2.6280 | 0.2380 | 0.0047 | 0.1648 |
| Sequence 2 / Building Front | Cam2 | 252/252 | 2.2704 | 0.0339 | 1.7278 | 6.0566 | 0.7379 | 0.0155 | 0.4906 |
|                       | Cam3 (Trinocular) | 252/252         | 1.3812 | 0.0216 | 0.9878 | 3.8433 | 1.1266 | 0.0235 | 0.7387 |
|                       | Cam5 (Omni-VO) | 252/252           | 0.9881 | 0.0177 | 0.6953 | 2.4442 | 0.5061 | 0.0103 | 0.3008 |
| Sequence 3 / Main Road | Cam2          | 600/600           | 6.7366 | 0.0388 | 5.1521 | 14.1245 | 6.0572 | 0.0113 | 3.0732 |
|                       | Cam3 (Trinocular) | 600/600         | 6.5672 | 0.0173 | 5.4671 | 13.1245 | 6.1663 | 0.0152 | 3.1125 |
|                       | Cam5 (Omni-VO) | 600/600           | 4.8722 | 0.0327 | 4.3717 | 8.1084 | 2.6096 | 0.0071 | 1.4743 |

FIGURE 8. The output trajectory for all sequence with different camera setting (Cam2, Trifocal (3 cameras), OmniVO). (a) The output trajectory plot for sequence 1 against ground truth trajectory. (b) The output trajectory plot of results for sequence 2 against ground truth trajectory. (c) The plot of results in sequence 3 against ground truth trajectory.

FIGURE 9. The final output trajectory against ground truth. (a) The output trajectory of sequence 1. (c) Output of proposed algorithm for sequence 2. (d) The sequence 3 output trajectory plot against ground truth.

information. Therefore, the combination with the lower or single camera either fails to track or the pose estimation accuracy is degraded in such situations. Similarly, sequence 2 region has the narrow road around the building with sharp turns, the road region, and space that cover the majority of the area in the image. Therefore, adding more structural information distributed across all the camera views considerably improves the position accuracy. Sequence 3 is recorded
Z. Javed, G.-W. Kim: OmniVO: Toward Robust Omni Directional Visual Odometry With Multicamera Collaboration

FIGURE 10. The feature detection and tracking results for 5 cameras (a-e). Top row images shows the feature detection results from original images. Middle row shows the feature tracking result with KLT tracking algorithm. Bottom row shows the results after feature selection and outlier removal.

FIGURE 11. The output trajectory of the proposed method against state of the art visual odometry methods and GPS based ground truth for sequence 1 of the data-set.

from the main road of the campus, and the environment contains enough structural information even for front-facing cameras, so the output trajectories have almost similar or small differences for all combinations. However, in the case of bumps and other lightning problems, multiple camera setups improve the positioning accuracy.

FIGURE 12. The output trajectory of the proposed method against state of the art visual odometry methods and GPS based ground truth for sequence 2 of the data-set.
The output trajectory of OmniVO with reference to ground truth is shown in Figure 9 for all three sequences. The figure shows that the output of the proposed algorithm is close to the ground truth for all sequences. The quantitative results for different combinations of cameras are presented in Table 1. The average trajectory error (translation) and average rotation error (yaw only) are computed for comparison. The results are compared by computing root mean square error (RMSE), minimum (min), median, maximum (max), and standard deviation (S.D) for each sequence. The RMSE error along with other parameters suggest that adding more structural information improves the result as compared to fewer camera settings. The pose estimation accuracy is better with multiple camera settings because in some situations single camera cannot have obvious feature information for successful tracking. In conclusion, the large field of view benefits the visual odometry or SLAM algorithm to perform better in complex environment situations with less or even few obvious structural information. This might help to perform tracking for long-term navigation. Similarly, the even distribution of feature points across the view significantly improves the position accuracy.

D. FEATURE PRIORITIZATION EVALUATION

The structural information with 360-degree coverage improves the pose estimation accuracy as well as prevents tracking failure in a complex environment. However, the computational complexity of the algorithm increases with too many extracted features from different views. A feature prioritization algorithm is presented in order to budget features for multi-camera setups. The detailed comparison results of feature prioritization with different cameras configuration is shown in Table 2. The results indicate that feature budgeting and removing outliers significantly reduce the time consumption of the pose estimation module of the algorithm. The table shows the amount of time consumed by the visual odometry algorithm with respect to several features with and without the feature prioritization module. A significant improvement can be seen from the table by adding a feature prioritization module. Moreover, the output of feature detection, tracking, feature prioritization, and outlier removal is shown in Figure 10. The top row images show the feature detection results from each camera, while the center row presents the tracked feature using the KLT tracking algorithm for all cameras. It can be shown from results that there are a large number of detected features as well as wrongly tracked features for each camera due to lens distortion. The bottom row shows the results after feature selection with outlier removal. The results in Figure indicate that features are distributed evenly in the images as well as the wrongly tracked features are successfully removed.

E. COMPARISON WITH STATE OF THE ART METHODS

To further confirm the results of the proposed method, a detailed comparison is performed with state-of-the-art visual odometry algorithms from the literature. The comparison is performed quantitatively by computing the RMSE of a translation error. Similarly, the qualitative results are presented with the ground truth trajectories provided by the GNSS system. The ROVO [16] and ROVINS [29] are multi-camera based state of the art visual odometry algorithms, while ORB-SLAM-monocular [37] and viso2-mono [38] are widely used monocular camera-based methods. ROVINS is a visual-inertial navigation algorithm, both ROVO and ROVINS are discussed in the related work section. The ROVO and ROVINS are not open source for comparison, therefore, widely used and open source ORBSLAM-monocular [37] and viso2-mono [38] algorithms are used for comparison. The comparison is performed with ORBSLAM-monocular and viso2-mono by scale alignment. The front view camera is selected to run ORBSLAM-mono and Viso2-mono. The images are rectified and distortions are corrected for both algorithms. The qualitative results between the proposed method (OmniVO), ORBSLAM-mono, and viso2-mono against the GNSS ground truth trajectories are shown in Figure 11, 12, and 13 for all three sequences respectively.

As shown from the qualitative results, the proposed algorithm with an omnidirectional camera setup performs better in each of the selected scenarios. In the case of sequence 1, the proposed methods have slightly better results than ORBSLAM-mono and viso2. ORBSLAM-mono failed in the case of a very sharp turn due to a lack of reliable feature tracking. Similarly, viso2-mono can be able to track some features, but due to fewer matches, pose estimation accuracy is not very accurate, so viso2-mono slightly drifts away from the actual path as shown in Figure 11. The proposed method...
TABLE 3. Position and rotation accuracy of proposed algorithm with comparison with state of the art methods.

| Sequence /Environment | Camera Setting | F.Recover/T.Frame | ATE Translation (Meters) | ARE Yaw (Degree) |
|------------------------|----------------|-------------------|--------------------------|------------------|
|                        |                |                   | RMSE        | Min     | Median | Max     | RMSE        | Min     | S.D     |
| Sequence 1 / Parking   | ORBSLAM-mono [37] | 220/340          | 2.2734      | 0.0912  | 1.8584 | 4.5832  | 7.6301      | 0.0545  | 5.0784  |
|                        | Viso2-mono [38]  | 340/340          | 5.9658      | 0.0162  | 3.6926 | 16.1360 | 8.9691      | 0.0295  | 9.8167  |
|                        | Cam2 (Omnivo)   | 340/340          | 1.4771      | 0.0076  | 1.2670 | 3.2628  | 0.2457      | 0.0099  | 0.3710  |
|                        | Omnivo         | 340/340          | 1.3125      | 0.0042  | 0.7471 | 3.2628  | 0.2380      | 0.0047  | 0.1648  |
| Sequence 2 / Building Front | ORBSLAM-mono [37] | 150/252         | 6.0184      | 0.0829  | 4.5511 | 8.8717  | 3.8023      | 0.0087  | 2.8881  |
|                        | Viso2-mono [38]  | 252/252          | 2.3946      | 0.0265  | 1.9139 | 7.8059  | 4.5611      | 0.0145  | 3.0147  |
|                        | Cam2 (Omnivo)   | 252/252          | 2.2704      | 0.0339  | 1.7278 | 6.0566  | 0.7379      | 0.0155  | 0.4906  |
|                        | Omnivo         | 252/252          | 0.9881      | 0.0177  | 0.6953 | 2.4442  | 0.5061      | 0.0103  | 0.3008  |
| Sequence 3 / Main Road | ORBSLAM-mono [37] | 600/600         | 4.8923      | 0.0253  | 4.6968 | 8.1253  | 5.1235      | 0.0059  | 3.3245  |
|                        | Viso2-mono [38]  | 600/600          | 7.3317      | 0.0461  | 2.8060 | 14.2460 | 5.0123      | 0.0059  | 3.3245  |
|                        | Cam2 (Omnivo)   | 600/600          | 6.7366      | 0.0388  | 5.1521 | 14.1245 | 6.0572      | 0.0113  | 3.0732  |
|                        | Omnivo         | 600/600          | 4.8722      | 0.0327  | 4.3717 | 8.1084  | 2.6096      | 0.0071  | 1.4743  |

FIGURE 14. The Translation error plot of proposed method, ORBSLAM-mono and Viso2-mono for all three sequences (a-c).

TABLE 4. Time consumption (millisecond) of proposed method with comparison of the state-of-the-art methods.

| Methods                  | No. Features | Time(ms) |
|--------------------------|--------------|----------|
| Viso2-mono               | 1000         | 124      |
| ORBSLAM2-mono            | 1000         | 32       |
| OmniVO (single camera)   | 1000         | 40       |
| OmniVO                   | 1000         | 65       |

provides better results due to feature distribution across all the views. Sequence 2 still contains a very sharp turn and some motion blur in the images, so ORBSLAM-mono is still not able to track features at a very sharp turn and fails to track the complete sequence. Viso2-mono is able to find matches in this sequence and provide better results with low drift, as depicted in Figure 12. The proposed method is able to track the feature points and provide better pose estimation accuracy than viso2-mono. Sequence 3 is selected because it provides enough features and a structured main road environment. Due to the simplicity of the environment, all methods have almost similar accuracy.

Other than the qualitative comparison, Table 3 shows the quantitative evaluation of the proposed method with other open source methods. The results are compared based on the translation error and rotation error (RE) by computing the RMSE, minimum, and standard deviation. Moreover, successfully tracked frames are also shown to check the tracking failure of each algorithm. In the case of the recovered frame, the proposed method is able to track all frames over the entire sequence. While ORBSLAM-mono fails to recover all frames, similarly, the proposed method has high positioning and orientation accuracy in comparison with state-of-the-art single-camera methods for sequence 1 and sequence 2. Because the environment is slightly feature-less and contains very sharp turns, single camera-based methods are not able to provide reasonable performance. In the case of sequence 3, the viso2-mono and ORBSLAM-mono provide similar results in comparison with the proposed method, as this data is recorded from the main road with obvious feature information.

The translation error for each method is shown in Figure 14. The proposed algorithm with multi-camera setup provides better than ORBSLAM-mono and viso2-mono for the first two sequences. While the position error of proposed method and ORBSLAM-mono is almost similar for third sequence, However, the accuracy is better than viso2-mono.
for this sequence. The computational performance of the proposed method is presented in Table 4. A total number of 1000 features are detected. The overall time consumption of the proposed method is approximately 67 ms with 5 cameras. while a single camera setup has a time consumption of approximately 40 ms. The feature detection and tracking modules are implemented on parallel threads and features are prioritized before pose estimation, so the time consumption with all camera setups is reasonable. While ORBSLAM-mono approximately takes 32 ms mean-time and viso2-mono takes 124 ms. The ORBSLAM-mono has less time consumption than the proposed method due to a single camera, while the proposed method needs to process 5 cameras for visual odometry. In short, the proposed method has the ability to incorporate a multi-camera setup for robust motion estimation. From the experimentation, it can be seen that the large FOV has the advantage of preventing the failure of the odometry algorithm in challenging situations. In short, the proposed method has the ability to incorporate a multi-camera setup for robust motion estimation. From the experimentation, it can be seen that the large FOV has the advantage of preventing the failure of the odometry algorithm in challenging situations.

F. DISCUSSION, LIMITATION AND FUTURE WORK
This paper proposed the method to utilize the omnidirectional camera to perform visual odometry in a complex situation. Normally, the single camera-based method has difficulty performing in a complex situation, such as dynamic conditions or direct sunlight, an off-road environment, lack of texture in the image, parking situations, and sensor failure may complicate the reliable tracking procedures is shown in Figure 15. Figure 15(a) contains the turning scene when the front-facing camera is mostly covered by the sky region. Similarly, the featureless parking region and the scene that is affected by sunlight are shown in Figures 15(b), and 15(c). ides an efficient way of utilizing the multi-camera system with a spherical camera model to estimate camera pose. The presented result shows that the proposed method can alleviate the problem offered by a single camera. Furthermore, feature prioritization can significantly help to improve the computational performance of the system. The proposed method can efficiently estimate the motion for the multi-camera system but cannot provide accurate real scale information as the overlapped region between cameras is very small. Therefore, the 3D LiDAR can be used to provide real-scale depth for visual information, and a tightly coupled approach with another sensor could be a good choice for future implementation. Furthermore, without mapping, loop closures, and back-end optimization, odometry drifts from the actual path for a longer trajectory. Therefore, integration of the proposed method to design a visual-LiDAR SLAM algorithm for an omnidirectional camera system will be considered.

VI. CONCLUSION
In this paper, aiming to provide a robust visual odometry system for omnidirectional multi-camera setup, the method is proposed which can significantly improve the performance of visual odometry. Firstly, the features are detected from each view in a parallel thread and tracked with the help of the KLT algorithm. The outlier removal algorithm is proposed to remove the wrong tracked feature after KLT tracking. The features are prioritized to reduce the computational complexities of the pose estimation inside a RANSAC loop. The pose is estimated with an 8-point algorithm by using a camera model. At last, the inlier landmarks are obtained from all the cameras and then the pose is optimized using a camera having a large number of inlier sets. Most importantly, the hardware setup is developed and the dataset is recorded from the outdoor environment. The detailed experimental results show that the proposed method can effectively improve the result by approximately 40% in comparison with state-of-the-art methods. To address the real-scale issues, the 3D
Lidar could be used with the proposed method for future implementation.

REFERENCES

[1] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, “Simultaneous localization and mapping: A survey of current trends in autonomous driving,” IEEE Trans. Intell. Veh., vol. IV-3, no. 3, pp. 15–64, Sep. 1964.

[2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” IEEE Trans. Robot., vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[3] R. Mur-Artal and J. D. Tardos, “ORB-SLAM: An open-source slam system for monocular, stereo, and RGB-D cameras,” IEEE Trans. Robot., vol. 32, no. 5, pp. 1255–1257, 2017.

[4] J. Engel, T. Schops, and D. Cremers, “LSD-SLAM: Large-scale direct monocular SLAM,” in Proc. Eur. Conf. Comput. Vis., 2014, pp. 834–849.

[5] G. Klein and D. Murray, “Parallel tracking and mapping for small AR workspaces,” in Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality, Nov. 2007, pp. 225–234.

[6] J. Engel, V. Koltun, and D. Cremers, “Direct sparse odometry,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 3, pp. 611–625, Mar. 2017.

[7] C. Forster, M. Pizzoli, and D. Scaramuzza, “SVO: Fast semi-direct monocular visual odometry,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2014, pp. 15–22.

[8] J. Luiten, T. Fischer, and B. Leibe, “Track to reconstruct and reconstruct to track,” IEEE Robot. Autom. Lett., vol. 5, no. 2, pp. 1803–1810, Apr. 2020.

[9] H. Karunasekera, H. Wang, and H. Zhang, “Multiple object tracking with attention to appearance, structure, motion and size,” IEEE Access, vol. 7, pp. 104423–104434, 2019.

[10] B. Bescos, J. M. Fácil, J. Civera, and J. L. Neira, “DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes,” IEEE Robot. Autom. Lett., vol. 3, no. 4, pp. 4076–4083, Oct. 2018.

[11] Y. Yang, D. Tang, D. Wang, W. Song, and J. Mu, “Multi-camera visual SLAM for off-road navigation,” Robot. Auto. Syst., vol. 128, Jun. 2020, Art. no. 103505.

[12] Z. Zhang, H. Rebca, C. Forster, and D. Scaramuzza, “Benefit of large field-of-view cameras for visual odometry,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2016, pp. 801–808.

[13] L. Peidong, H. Lionel, S. Torsten, G. Andreas, and P. Marc, “Direct visual odometry for a fisheye-stereo camera,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Sep. 2017, pp. 1746–1752.

[14] W. Yahui, C. Shaqian, L. Shi-Jie, L. Yun, G. Yangyan, and L. Tao, and C. Ming-Ming, “CubemanSLAM: A piecewise-pinhole monocular fisheye SLAM system,” in Proc. Asian Conf. Comput. Vis., 2018, pp. 34–49.

[15] U. Steffen and H. Stefan, “MultiCol-SLAM—A modular real-time multi-camera SLAM system,” 2016, arXiv:1610.07336.

[16] H. Seok and J. Lim, “ROVINS: Robust omnidirectional visual inertial navigation system,” IEEE Trans. Robot. Autom., vol. 33, no. 2, pp. 499–508, Apr. 2016.

[17] Y. Zhang and F. Huang, “Panoramic visual SLAM technology for spherical panoramas,” in Proc. IEEE 11th Int. Conf. Comput. Vis., Oct. 2007, pp. 1–8.

[18] Y. Shi, S. Ji, Z. Shi, Y. Duan, and R. Shibasaki, “GPS-supported visual SLAM with a rigorous sensor model for a panoramic camera in outdoor environments,” Sensors, vol. 13, no. 1, pp. 119–136, 2013.

[19] H. Seok and J. Lim, “ROVINS: Robust omnidirectional visual inertial navigation system,” IEEE Robot. Autom. Lett., vol. 5, no. 4, pp. 6225–6232, Oct. 2020.

[20] H. Matsu ki, L. von Stumberg, V. Usenko, J. Stückler, and D. Cremers, “Omnidirectional DSO: Direct sparse odometry with fisheye cameras,” IEEE Robot. Autom. Lett., vol. 3, no. 4, pp. 3693–3700, Oct. 2018.

[21] C. Won, H. Seok, Z. Cui, M. Pollefeys, and J. Lim, “OmnISLAM: Omnidirectional localization and dense mapping for wide-baseline multi-camera systems,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2020, pp. 559–566.

[22] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “G2o: A general framework for graph optimization,” in Proc. IEEE/RSJ Int. Conf. Robot. Autom., May 2011, pp. 3607–3613.

[23] A. Howard, “Real-time stereo visual odometry for autonomous ground vehicles,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Sep. 2008, pp. 3946–3952.

[24] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in Proc. IJCAI, Aug. 1891, pp. 674–679.

[25] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in Proc. IJCAI, Aug. 1891, pp. 674–679.

[26] B. K. Horn, H. M. Hilden, and S. Negahdaripour, “Closed-form solution of absolute orientation using orthonormal matrices,” J. Opt. Soc. Amer. A, Opt. Image Sci., vol. 5, no. 7, pp. 1127–1135, 1988.

[27] Z. Zhang and D. Scaramuzza, “A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 7244–7251.

[28] R. Mur-Artal, J. M. Montiel, and J. D. Tardós, “ORB-SLAM: A versatile and accurate monocular SLAM system,” IEEE Trans. Robot., vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[29] A. Geiger, J. Ziegler, and C. Stiller, “StereoScan: Dense 3D reconstruction in real-time,” in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2011, pp. 963–968.

[30] D. Caruso, J. Engel, and D. Cremers, “Large-scale direct SLAM for omnidirectional cameras,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Sep. 2015, pp. 141–148.

[31] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, “SVO: Semidirect visual odometry for monocular and multicamera systems,” IEEE Trans. Robot., vol. 33, no. 2, pp. 249–265, Apr. 2016.

[32] F. Kangi and R. Laganiere, “Orientation and pose recovery from spherical panoramas,” in Proc. IEEE 11th Int. Conf. Comput. Vis., Oct. 2007, pp. 1–8.

[33] Y. Shi, S. Ji, Z. Shi, Y. Duan, and R. Shibasaki, “GPS-supported visual SLAM with a rigorous sensor model for a panoramic camera in outdoor environments,” Sensors, vol. 13, no. 1, pp. 119–136, 2013.

[34] H. Seok and J. Lim, “ROVINS: Robust omnidirectional visual inertial navigation system,” IEEE Robot. Autom. Lett., vol. 5, no. 4, pp. 6225–6232, Oct. 2020.