Simple But Effective Scale Estimation for Monocular Visual Odometry in Road Driving Scenarios

MING FAN, SEUNG-WOOK KIM, (Member, IEEE), SUNG-TAE KIM, JEE-YOUNG SUN, (Member, IEEE), AND SUNG-JEA KO, (Fellow, IEEE)
Department of Electrical Engineering, Korea University, Seoul 02481, South Korea
Corresponding author: Sung-Jea Ko (sjko@korea.ac.kr)

This work was supported by the Institute for Information and communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (Intelligent Defense Boundary Surveillance Technology Using Collaborative Reinforced Learning of Embedded Edge Camera and Image Analysis), under Grant 2017-0-00250.

ABSTRACT In large-scale environments, scale drift is a crucial problem of monocular visual simultaneous localization and mapping (SLAM). A common solution is to utilize the camera height, which can be obtained using the reconstructed 3D ground points (3DGPs) from two successive frames, as prior knowledge. Increasing the number of 3DGPs by using more proceeding frames can be a natural extension of this solution to estimate a more precise camera height. However, merely employing multiple frames based on conventional methods is hard to be directly applicable in a real-world scenario because the vehicle motion and inaccurate feature matching inevitably cause large uncertainty and noisy 3DGPs. In this study, we propose an elaborate method to collect confident 3DGPs from multiple frames for robust scale estimation. First, we gather 3DGP candidates that can be seen in more than a predefined number of frames. To verify the 3DGP candidates, we filter out the 3D points at the exterior of the road region obtained by the deep-learning-based road segmentation model. In addition, we formulate an optimization problem constrained by a simple but effective geometric assumption that the normal vector of the ground plane lies in the null space of a movement vector of the camera center, and provide a closed-form solution. ORB-SLAM with the proposed scale estimation method achieves the average translation error with 1.19% on the KITTI dataset, which outperforms the state-of-the-art conventional monocular visual SLAM methods in road driving scenarios.

INDEX TERMS Monocular SLAM, scale estimation, 3D plane fitting.

I. INTRODUCTION
In autonomous driving, simultaneous localization and mapping (SLAM) algorithms play an important role. The SLAM algorithm estimates the motion of a vehicle using a mounted sensor such as a depth sensor, calibrated stereo cameras, or a single RGB camera. Especially, the monocular visual SLAM (vSLAM) algorithm is attracting the automobile industry because a single RGB camera is cheaper and requires less calibration effort than other sensors. However, since there is no depth information available in the monocular vSLAM system, a pose of the current camera is usually obtained based on the previously estimated camera poses. This consequently causes error propagation, referred to as the scale drift problem [1]. To solve this problem, the loop closure method [1], [2] or the prior knowledge of the environment [3]–[12] are widely adopted. In the loop closure method [1], once the closed-loop trajectory of a vehicle is detected, all camera poses are corrected through a pose-graph optimization [13] constrained with a similarity transformation. However, vehicles do not always move around a loop in real autonomous driving scenarios. Recently, the prior knowledge of the environment, such as the size of a detected object [4], [5], [8], [9] and the known height of a camera mounted on the vehicle [3], [6]–[12], has been used as a reference to modify the camera pose. The relative camera pose is scaled to the real-world coordinate data using the ratio between the size of a reference object and that in the real world. This ratio is commonly denoted as a scale factor. As the size of the reference objects, e.g., cars, may vary...
significantly, the information on the camera height, which can be easily measured, is generally employed to obtain the scale factor. In [7]–[12], the camera height is computed using homography decomposition, or 3D ground points (3DGPs) visible in the current frame, or both of them. Earlier works estimate the camera height using homography induced by the ground plane between two successive frames. However, the homography decomposition process is sensitive to noise and is challenging in terms of providing numerical stability [7], [8]. In [7], to improve the stability of the homography decomposition, a divide-and-conquer technique is proposed, in which the rotation and translation are obtained by using the entire set of feature points matched between the two successive frames; then, these camera motion parameters are substituted into the decomposition of homography. To compute the camera height, Grater et al. [12] fit the ground plane to the 3D points triangulated between the two frames using the random sample consensus (RANSAC) method [15]. In addition, based on the property that the direction of the vertical vanishing points of an image has the same direction as the normal vector of the ground plane, the normal vector is computed and then used to select the 3DGPs for the ground-plane-fitting. Compared with [4], [5], [7], multiple cues-based methods [8]–[10] have been also proposed, in which two or more predictions of the camera height are made using the homography decomposition, 3DGPs, and the size of an object. Then, the resulting camera height is obtained by using the Kalman filter [8], [9] or pre-trained classifier [10]. Currently, the SLAM methods employing the information from deep learning such as the estimated depth and segmented region or 3D bounding box of an object have also been developed to lessen the scale drift [16]–[19].

The hand-crafted methods mentioned above estimate the scale factor based on 3DGPs, planar homography, or both of them. Methods [11], [12] based on the 3DGPs are simple but their performance is limited since these methods only use the 3DGPs that are visible in the current frame and conduct inaccurate 3DGP validation. Multiple-cue-based methods [8]–[10] can produce reliable results. However, they require the auxiliary data-fusion process followed by the parameter training process. Considering that the scale estimation is a sub-system of the monocular vSLAM system, we propose a light-weight algorithm which estimates the scale factor only according to the available information from the conventional vSLAM system, i.e., we do not need additional processes such as optimization-based image patch registration.

Our motivation starts from a very natural extension that is enlarging a candidate 3DGP pool based on multiple frames. For ground plane estimation, robust estimation methods such as RANSAC or MLESAC [20] are widely applied. In these methods, the largest consensus set is found and used to estimate the plane parameters via least-square estimation. If we simply fit a plane with the likely 3DGPs collected from multiple keyframes, however, the estimation is often unstable since the 3DGPs collected from multiple keyframes have large uncertainty. For example, Fig. 1 illustrates the 3DGPs (black points) and the ground planes (differently colored) obtained by conducting three times the RANSAC-based 3D plane-fitting algorithm with these 3DGPs. Similar results can be obtained when the MLESAC was employed. In scale estimation, such non-unique solutions may yield a large error.

In this study, we propose an elaborate 3DGP validation method to estimate robustly the ground plane with multiple keyframes for scale estimation. To confirm the 3DGP candidates, we adopt a road segmentation network model and employ the segmentation mask to reduce the outliers. Also, we use a simple geometric assumption that the normal vector of the ground plane lies in the null space of a movement vector of the camera center. Under this assumption, we formulate a constrained optimization problem and provide a closed-form solution. We evaluated the performance of ORB-SLAM with the proposed scale estimation method on the KITTI [14] dataset. The experimental results demonstrate that the proposed method outperforms state-of-the-art conventional methods in road driving scenarios. The main contribution of this paper is summarized as follows:

- To the best of our knowledge, the proposed method is a first attempt to employ multiple keyframes to improve the performance of scale estimation. The proposed scale estimation method can be a baseline for a multi-keyframe-based approach.
- We adopt a supervised road-segmentation network and collect the 3DGPs among the 3D points employing the semantic information from the road segmentation.
- In the view of an optimization framework, we formulate the geometric assumption that, in the co-visibility graph of the current keyframe, a vehicle moves on a local plane, and provide a closed-form solution. As compared to the conventional objectives, the proposed closed-form solution can effectively improve the performance without a significant increase in computation.
ORB-SLAM using our scale estimation method w/ and w/o the proposed geometrically-constrained optimization achieves 1.19% and 1.23% in terms of the average translation error on the KITTI dataset [14], which is a superior result to state-of-the-art conventional monocular vSLAM methods in road driving scenarios.

II. BACKGROUNDS

The proposed method employs the monocular feature-based vSLAM system to obtain the camera pose. In this section, we briefly introduce the vSLAM algorithms and the 3D plane geometric model used in the proposed method.

A. FEATURE BASED vSLAM

The popular feature-based vSLAM algorithms include MonoSLAM [21], PTAM [22], and ORB-SLAM. MonoSLAM is the firstly proposed vSLAM algorithm. In this method, the camera poses and 3D points are estimated simultaneously by using the extended Kalman filter (EKF). However, MonoSLAM is computationally expensive and is not suitable for large scale environments due to the large size of state vectors of the EKF. PTAM is a ground-breaking vSLAM algorithm that is developed for augmented reality applications in small workspaces. To achieve real-time performance, they first propose to apply the tracking and local mapping in two separate threads. ORB-SLAM, as an extension of PTAM, has become one of the most popular and complete feature-based monocular vSLAM algorithms [23]. As compared with other feature-based vSLAM algorithms, ORB-SLAM achieves fully automatic initialization without requiring human intervention and obtains the outstanding performance [24]. The keyframes are adaptively selected to achieve robust tracking in various environments [25], [26].

The ORB-SLAM algorithm mainly consists of three steps: tracking, local mapping, and loop closure. To achieve acceptable real-time performance, these three steps are performed in parallel by using multi-threading. In the tracking step, the camera pose of the upcoming (current) frame is estimated based on the detected and matched ORB feature points [31].

In the local mapping step, the 3D point corresponding to each 2D feature is reconstructed and refined by the local bundle adjustment (BA) [32]. The local BA efficiently improves the accuracy of both the reconstructed 3D points and the camera pose by minimizing the re-projection error of the 3D points with respect to their corresponding 2D features. In practice, as running BA with all feature points and all frames is not feasible [22], [26], the local BA is executed on a part of frames. Such selected frames are referred to as keyframes. In general, the successive frames containing redundant information, such as frequently repeated features, are not included in the keyframes. In the loop-closure step, all the computed camera poses are optimized to eliminate the scale drift problem based on the similarity transformation if loop closure is detected.

B. PLANAR GEOMETRY

As mentioned before, some of the conventional methods adopt a plane equation to compute the camera height [7], [10], [12] based on the RANSAC-based plane-fitting method [33]. A plane in the 3D space can be represented by its unit normal vector \( \mathbf{n} \in \mathbb{R}^3 \) and the distance \( d \in \mathbb{R} \) from the plane to the camera center. Let \( \mathbf{g}_n \in \mathbb{R}^{3 \times 1} \) be the \( n \)-th point among \( N \) 3D points on the plane, referred to as 3DGPs; then, the plane equation can be defined as \( \mathbf{n}^T \mathbf{g}_n - d = 0 \). The normal vector \( \mathbf{n} \) is obtained by minimizing the constrained optimization problem defined as

\[
\min_{\mathbf{n}} \| \mathbf{G}^T \mathbf{n} \|_2^2 \quad \text{s.t.} \quad \| \mathbf{n} \|_2 = 1, (1)
\]

where \( \mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_N] - \bar{\mathbf{g}} \bar{\mathbf{1}}^T, \bar{\mathbf{g}} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{g}_n, \) and \( \mathbf{1} \) is the vector of which the entries are all ones with an appropriate length. The solution of (1) can be obtained as stated in the following basic theorem [34].

Theorem 1 (Rayleigh-Ritz): Let \( \mathbf{A} \in \mathbb{R}^{n \times n} \) be real and symmetric, and let the eigenvalues of \( \mathbf{A} \), \( \lambda_1, \ldots, \lambda_n \), be arranged in descending order. Then,

\[
\lambda_n = \min_{\mathbf{x} \neq \mathbf{0}} \frac{\mathbf{x}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \min_{\| \mathbf{x} \|_2 = 1} \mathbf{x}^T \mathbf{A} \mathbf{x},
\]

and the minimum is attained at \( \mathbf{x}_n \), which is the unit eigenvector corresponding to the smallest eigenvalue of \( \lambda_n \).

Let \( \mathbf{G} = \mathbf{U} \Sigma \mathbf{V}^T \) be the singular-value decomposition (SVD) of \( \mathbf{G} \). Based on Theorem 1, the normal vector is
a column vector of U corresponding to the smallest singular value. Then, the distance from the camera center to the ground plane, denoted as $h$, can be computed by

$$h = n^T \hat{g}. \quad (2)$$

![FIGURE 3. Illustration of the proposed 3DGP collection method.](image)

III. PROPOSED METHOD

Fig. 2 represents the flowchart of the proposed scale-correction framework. According to the results obtained in the local mapping step, we first collect 3D points $s_m$ for $m = 1, \ldots, M$ in the world coordinate system and the camera pose $T_k$ for $k = 1, \ldots, K$ representing a rigid transformation of each keyframe, where $M$ and $K$ are the numbers of 3D points and keyframes, respectively. With the camera pose, we can transform the coordinate of 3D points from the world coordinate system to that of each keyframe. The camera position $c^w$ without scale correction is also collected. Among the collected 3D points $s_m$, the 3DGP candidates $g_m$ are selected using the road segmentation of mask (Section III-A). If the numbers of 3DGPs, $N$, is larger than a threshold, $\delta$, a motion vector $m^w$ representing the movement direction of the camera is computed. Then, a 3D plane orthogonal to $m^w$ is fitted to the 3DGP. Finally, using the normal vector $n$ corresponding to the ground plane, and the mean point of the collected 3DGPs, $\hat{g}$, the scale factor is computed and used to correct the camera poses. If $N \leq \delta$, we assume that the result of the proposed method is unavailable, so the previously computed scale factor is used to correct the camera pose.

A. 3DGP CANDIDATES COLLECTION USING ROAD SEGMENTATION

In conventional methods [8]–[10] that employs two successive frames, the accuracy of the reconstructed 3DGPs is low due to the feature-matching error and small triangulation angle. In the early stage of the proposed scale estimation, we first collect confident 3DGP candidates from each keyframe based on the semantic information from the road-segmentation network model. Fig. 3 depicts the proposed 3DGP candidates collection method using road segmentation. The marked points and camera-shaped icons represent 3D space points and the camera positions corresponding to keyframes in the world coordinates, respectively. Let an arbitrary 3D point $s_m$ be given in the world coordinate in the given keyframe. Then, we can determine whether this point is a 3DG or not by projecting $s_m$ onto the image plane as follows:

$$x_g = \pi(T(s_m)), \quad (3)$$

where $T(\cdot)$ is the rigid transformation of the given keyframe, which transforms a 3D point from the world coordinate system to that of the given keyframe, and $\pi(\cdot)$ is the transformation that projects the 3D point onto the image plane. If the projected point $x_g$ is inside the ground region of the keyframe, we assign $s_m$ as a 3DGP candidate. To obtain 3DGP candidates with high confidence, we obtain the ground region using our previous work [36], referred to as RBA-Net. We employ the VGGNet [45] as a backbone. The coarse road region is first predicted from the top-level feature maps. Then,

2 In this study, the 3D points are represented in the inhomogeneous coordinates. For a 3D point $x \in \mathbb{R}^3$, the transformation of the coordinate system is defined as $T(x) = Rx + t$, where $R$ and $t$ are the rotation matrix and translation vector, respectively.
the road region is gradually refined by learning the residual refinement module in a top-down scheme. Since most of the false predictions occur on the road boundary, the reverse and boundary attention units in the residual refinement module are employed, which guides the network to focus on the region near the road boundary. The detailed architecture and implementation details can be found in [45]. The road segmentation network model is trained on the URBAN road category of the KITTI benchmark [46]. Without the loss of generality, any road segmentation methods can be applied in the proposed scale estimation method. The scale estimation performance for different road segmentation methods will be presented in Section IV-B.

Fig. 4 shows an example of the road segmentation result and the points projected from the collected 3D points. The road region identified by the network is shown in green. The estimated 3D ground and space points are marked as red and blue points, respectively. Conventional methods employing the fixed region of interest (RoI) on the road [7]–[10], [12] often fail to estimate the camera height if the RoI is occluded by an unexpected object on the road. On the contrary, with the proposed segmentation-based 3DGP verification method, we can secure 3DGP candidates with high confidence.

B. 3DGP SELECTION FROM MULTIPLE KEYFRAMES

For a vehicle running on the road driving scenarios, generally, the movement in the up-down (y) direction is much smaller than that in the forward-back (z) and left-right (x) directions. Based on this fact, we assume that, if the captured images share enough common 3D points, the vehicle is running on a locally planar surface. To enlarge the candidate 3DGP pool for plane fitting, we propose to select 3DGPs from multiple keyframes.

To this end, we use the co-visibility graph adopted in ORB-SLAM [26]. In the co-visibility graph, any two keyframes are linked if they share observations of the same 3D points (at least 15 3D points). For the current keyframe, we first form a set of keyframes, denoted as \( K_1 \), in which each element contains shared 3D points with the current keyframe. For example, as shown in Fig. 3, keyframes linked with the current keyframe (see a red camera icon) in the co-visibility graph are denoted by the blue camera icons. The red dots and cross lines represent their 3DGP candidates and 3D space points, respectively. Then, we gather all 3DGP candidates of the keyframes in \( K_1 \), which are denoted by blue and red dots. To select 3DGPs with higher confidence, the 3DGP candidates with the large triangulation angle are going to be employed. For each 3DGP candidate, we count the number of keyframes in which that point can be also observed. To this end, we first enlarge the keyframe set \( K_1 \) by further collecting the keyframes that are linked with the keyframes in \( K_1 \). We denote the enlarged keyframe set as \( K_2 \), which is indicated by blue and green camera icons in the figure. Only the 3DGP candidates that can be observed in at least \( \lambda \) keyframes are used as 3DGPs for plane-fitting.

When a vehicle moves forward, the proposed method of 3DGP selection ensures that the triangulation angle is sufficiently large because the triangulation angle becomes larger as the distance between the camera centers gets bigger. Note that, since we collect 3DGPs from multiple keyframes, the number of the 3DGPs is sufficient for the robust plane-fitting algorithm. Even though the road region is largely occluded by unexpected objects or segmentation fails due to over-exposure for several frames, the proposed 3DGP collection strategy can work robustly. Fig. 5 illustrates an example in which the 3DGPs are robustly detected even a truck occludes a large part of the road region.

C. ESTIMATE MOVEMENT DIRECTION OF PREVIOUS KEYFRAMES

We compute the plane equation based on the movement direction of the camera. For a vehicle movement on a ground surface, the normal vector of the ground plane is expected to be orthogonal to the movement direction of the camera center. To obtain the movement direction of the camera, we first collect all 3D coordinates of camera positions for each keyframe in \( K_1 \). Then, we fit a line in a vector form to the 3D coordinates of the camera positions, as follows:

\[
\min_{\mathbf{m}^w} \sum_{k=1}^{K} \left\| \mathbf{m}^w \times (\mathbf{c}_k^w - \bar{\mathbf{c}}^w) \right\|
\]

where \( K = |K_1|, \mathbf{m}^w \in \mathbb{R}^{3 \times 1} \) is the motion vector of the camera to be estimated, \( \mathbf{c}_k^w \in \mathbb{R}^{3 \times 1} \) is the 3D coordinate of the \( k \)th camera center in the world coordinate system, and \( \bar{\mathbf{c}}^w = 1/\sum_{k=1}^{K} \mathbf{c}_k^w \). The solution of (4) can be obtained by the SVD of a matrix \( \mathbf{C} \in \mathbb{R}^{3 \times K} \) which is given by \( \left[ \mathbf{c}_1^w, \cdots, \mathbf{c}_K^w \right] = \tilde{\mathbf{C}} \mathbf{V}_c \mathbf{U}_c^T \). Let \( \mathbf{C} = \mathbf{U}_c \Sigma \mathbf{V}_c^T \) be the SVD of \( \mathbf{C} \). The movement direction of the camera in the world coordinate system \( \mathbf{m}^w \) is the column vector of \( \mathbf{U}_c \), corresponding to the largest eigenvalue. Computing the movement direction only

FIGURE 5. An example of 3DGP collection results when the road region is occluded by an object. The first row shows a keyframe in the co-visibility graph of the current frame. The second row shows the current frame. The space 3D points and 3DGPs are represented as blue and red points, respectively. One can note that since we collect 3DGPs from multiple keyframes, the 3DGPs can be successfully collected even though the road region is largely occluded by the truck.
with two camera centers tends to produce incorrect results since the movement direction may be frequently changed in pitch direction if the vehicle is moving on an uneven surface. Therefore, the proposed multi-keyframe-based camera-motion estimation method can provide a more stable result.

**D. 3D GROUND PLANE FITTING**

The RANSAC scheme is widely adopted in the conventional methods to estimate the plane parameter. The largest consensus set of inliers is found and used to estimate the plane parameters via least-square estimation. In the proposed method, based on the selected inliers, the 3D plane is obtained by solving the proposed constrained optimization problem as follows:

$$
\min_{n} \left\| G^T n \right\|_2^2 \quad \text{s.t.} \quad \left\| n \right\|_2 = 1 \text{ and } M^T n = 0,
$$

(5)

where $G \in \mathbb{R}^{3 \times N}$ is a matrix formulated as defined in (1), $M = [m^w, 0, 0]$, $0 \in \mathbb{R}^{3 \times 1}$ is a vector of which the entries are all zeros, and $m^w$ is the movement direction of the camera, obtained in (4). Unlike in (1), in (5), we add another constraint whereby the normal vector of a ground plane to be fitted should be orthogonal to the normal vector of the line direction (vehicle movement direction).

Note that, to insure that the normal vector of the fitted plane is perpendicular to the motion vector $m^w$, instead of solving (5) by least square minimization, we first find the left null space of the motion vector, denoted as null($M^T$), i.e., any vector in the left null space should be orthogonal to the motion vector $m^w$. Since $M$ is a rank-one matrix, the two vectors spanning null($M^T$) can be easily obtained from the SVD of $M$, i.e., $M = PSQ^T$. The null space is shown as a gray-colored plane in Fig. 3. Let $B = [p_2 \ p_3] \in \mathbb{R}^{3 \times 2}$, where $p_2$ and $p_3$ are the two column vectors of matrix $P$ corresponding to the singular values of 0. Then, the normal vector of the plane can be obtained by a linear combination of the basis vectors of $B$, as follows:

$$
n = By,
$$

(6)

where $y \in \mathbb{R}^2$. By substituting (6) into (5), the optimization problem used to fit the 3D plane can be rewritten as

$$
\min_{y} \left\| G^T By \right\|_2^2 \quad \text{s.t.} \quad \left\| y \right\|_2 = 1.
$$

(7)

It should be noted that $\left\| y \right\|_2 = 1$ as $B$ is formed by two orthonormal vectors and $\left\| n \right\|_2 = 1$. The minimum of (7) can be achieved by using Theorem 1. Consequently, the height of the camera is obtained as

$$
h = (By)^T \bar{g}.
$$

(8)

We collect 3DGPs from $|K_1|$ keyframes in the co-visibility graph of the current keyframe and fit a 3D line to the collected keyframes. If $|K_1|$ is larger than the number of keyframes in the co-visibility graph, only the available keyframes are collected. The normal vector of the ground plane is found in the null-space of the line direction (vehicle movement direction). In the case of vehicle rotation, since the view of the scene changes dynamically, only the limited number of previous keyframes share observations of the same, at least 15, 3D points with the current keyframe. As a result, the number of keyframes in the co-visibility graph is small and the trajectory of these keyframes exhibit negligible curvature. Therefore, the line direction of these keyframes can also be used to reduce the pitch error of the fitted ground surface. To illustrate this, we used the subsequence of sequence 06 of the KITTI dataset in which the road curvatures over 180 degrees are included. Fig. 6 shows the collected keyframes in the case of rotation. The trajectory of the vehicle is denoted by colored asterisks. The keyframes in $|K_1|$ are shown as filled red circles. The blue arrow indicates the fitted line direction. As can be seen, the number of collected keyframes decreases as the road curvature increases. The line direction obtained by (4) is almost parallel to the trajectory traveled by the collected keyframes.

**FIGURE 6.** The trajectory traveled by the vehicle and the selected keyframes are denoted as colored asterisks and red filled circles, respectively. The blue arrow indicates the line direction fitted by (4). (a) 10 keyframes are collected when the road curvature is small; (b) 7 keyframes are included in the co-visibility graph when the road curvature increases; (c) Only 5 keyframes are available. (d) The number of keyframes in the co-visibility graph increases when the curvature decreases.

**E. SCALE COMPUTATION AND APPLICATION**

After obtaining the camera height using (8), we compute the scale factor by $s = h_{gt}/h$, where $h_{gt}$ is the ground-truth camera height. For the KITTI dataset [14], $h_{gt} = 1.65m$. The camera center position of the current keyframe is updated as

$$
\bar{c}_w^p = s(c_1^w - c_2^w) + c_p^w,
$$

(9)

where $c_1^w$ and $c_2^w$ are the camera-center positions of the current and previous keyframes, respectively, in the world coordinate system, as obtained in the local mapping step. Since all camera poses are optimized by the local BA at the local mapping step, for the keyframes in $K_1$, we assume that the translation vector is accurate up to a scale factor [47]. Consequently, if the number of 3DGPs is smaller than $\delta$, the scale factor employed for the previous keyframe is adopted. Unlike conventional methods, the proposed method uses a scale factor directly, without applying any post-processing methods such as Kalman filter [8], [9], [12] or median filter [11]. In addition, the scaled camera pose is used as the final output, without performing any further BA optimization.

**IV. EXPERIMENTAL RESULTS**

The performance of the proposed method was compared with that of the following hand-crafted methods: Lee’s...
method [3], Song's method [9], Wang's method [11], and Fanani's method [10]. All these methods estimate the scale factor based on the prior knowledge of fixed camera height. In addition, we also compare the performance of the proposed method with that of three CNN-based methods: Yin’s method [16], Sucar’s method [17], Yang’s method [18], and Loo’s method [19]. ORB-SLAM using the proposed scale estimation method with and without the guidance of the motion vector are referred to as GCSEM (geometrically constrained scale estimation from multiple keyframes) and SEMK (scale estimation from multiple keyframes), respectively.

A. IMPLEMENTATION DETAILS

The proposed method includes three steps that run in parallel: tracking, local mapping, and scale correction. The tracking and local mapping steps are exactly the same as in ORB-SLAM. We considered that for the scale estimation methods [3]–[12] that use the prior knowledge, such as the camera height and the size of objects in the scene, the performance of scale drift elimination can be better evaluated without applying the loop closure. Since the proposed method also estimates the scale factor using the camera height, for a fair comparison, we did not apply the loop closure. The scale factor was obtained directly by only using a fitted 3D plane without post-processing or additional processes. We evaluated the performance of the proposed method on the KITTI visual odometry dataset [14]. The KITTI visual odometry dataset covers urban, residential, country, and highway roads. The images were captured at 10fps over a driving distance of 39.2 km. The vehicle speed varies from 0 to 90km/h. Since highly accurate camera poses are available, the KITTI dataset is wildly used for evaluating the scale estimation algorithm. The training sequences (00–10) were used as the ground truth pose is publicly available. The performance of the proposed method was quantified by using the KITTI visual odometry evaluation metric [14]. We compared the difference between the predicted and ground-truth camera poses, and computed the average of the rotation and translation errors for all possible sub-sequences of lengths of 100, 200, …, 800 meters [14]. To further test the capacity of the proposed scale estimation method, we also used the RobotCar dataset [48]. Comparing with the KITTI dataset, the RobotCar dataset is difficult for the vSLAM algorithms because of significant illumination changes. For some sequences in the RobotCar dataset, the state-of-the-art vSLAM algorithms such as SVO [27], DSO [29], and ORB-SLAM cannot estimate camera pose due to feature tracking failure [19]. Since ORB-SLAM was used as our baseline algorithm, we tested the proposed method on the sequence for which the ORB-SLAM system can produce camera poses and 3D points.

To evaluate the performance of the scale drift elimination, we also computed the absolute trajectory error (ATE) [49] of the proposed method and compared it with those of ORB-SLAM, ORB-SLAM without loop closure (OS-NL), and ORB-SLAM with the ground-truth scale (OS-GTs). We followed the evaluation method used in Wang’s [11] and Loo’s method [19]. Since the absolute scale is unknown for the ORB-SLAM and OS-NL, for comparison, the ATE was computed after aligning the estimated trajectory with the ground truth by using a similarity transformation. All the experiments were performed on a PC with an AMD Ryzen 7 2700X CPU, 16GB RAM, and NVIDIA GeForce Titan XP. In the RANSAC-based ground-plane-fitting, the maximum number of iterations and the distance threshold were set to 500 and 0.01, respectively. For each iteration, three 3DGPs were used.

B. PARAMETER STUDY AND PROCESSING TIME

To learn the effect of the parameters, λ, |K|, and δ used in the proposed method, we evaluated the results obtained using SEMK and GCSEM with different parameter settings. For each parameter setting, we executed the proposed methods 5 times on the KITTI training sequences. The results of sequence 01 are not presented since the proposed methods failed at this sequence.

The proposed method was developed by using the open-source ORB-SLAM as a baseline algorithm. We selected 3DGP as a candidate point for plane fitting if only the 3GP can be observed in, at least, λ keyframes. In our experiment, we set λ = 3 based on the following reasons. First, in ORB-SLAM, the keyframe is selected adaptively to ensure that the feature can be robustly tracked in various environments [25], [26]. For instance, if the scene is changed largely in the case of rotation, or fast vehicle speed, the keyframe is selected as fast as possible for robust tracking. Second, since a larger baseline can produce higher precision of 3D reconstruction, the reconstructed 3DGP with a higher λ value is preferred. Third, we found that, when λ > 3, the number of collected 3DGPs is extremely decreased. This is due to the fact that as the camera moves forward, the feature points on the ground plane exit the field of view very quickly and fewer 3DGPs are simultaneously observable from 4 keyframes. In our experiment, when λ > 3, the scale factor cannot be obtained for several parts of sequence 06, 07, 08 and 09 since the number of collected 3DGPs is smaller than the minimum required number for plane fitting. Thus, under the ORB-SLAM keyframe selection method, we found that λ = 3 is an appropriate value to chose 3DGPs with high reconstruction precision. The reliable results for both KITTI and RobotCar dataset were also obtained by setting λ = 3.

To fully study the impact of the parameter |K|, we tested the proposed method 5 times for KITTI training sequences with different |K| values ranging from 1 to 20 as shown in Fig. 8. The average result for each sequence with different |K| was computed and shown as the dashed line with different colors. The overall performance was also computed for each |K|, and shown as a thick red line. Note that, when |K| > 10, the difference in overall performance is unnoticeable. This is because that, we only assembled the keyframes in the co-visibility graph of the current keyframe. The accuracy

3 An open-source python package [50] was used to compute ATE.
and processing time of the proposed method for different $|K_1|$ is shown in Fig. 8. The processing time per frame increased as $|K_1|$ increased since much more 3DGPs are collected. The processing time was measured without using any CPU and GPU acceleration techniques such as Intel streaming SIMD extensions and CUDA-accelerated libraries. Considering both processing time and accuracy, we set $|K_1| = 12$.

In the proposed method, the previously computed scale factor is used to correct the camera pose if the number of collected 3DGPs is smaller than $\delta$. After setting $|K_1| = 12$ and $\lambda = 3$, we run the proposed methods 5 times for KITTI training sequences and the results are shown in Fig. 3. One can note that the performance of the proposed method is almost unchanged when $\delta < 30$. When $\delta > 30$, the performance is degraded. This is because, for the textureless ground region, the number of collected 3DGPs is generally smaller than 30 although we collected 3DGPs from multiple keyframes. In summary, the default settings were determined as $|K_1| = 12$, $\lambda = 3$ and $\delta = 15$.

To study the impact of the road segmentation method, we tested the proposed methods on the KITTI dataset with SegNet [38], MultiNet [43], and RBANet. Table 1 lists the overall translation error of the proposed scale estimation method with different networks. For each network, the evaluation results of F1-measure are also listed in Table 1. For the URBAN road KITTI dataset, the SegNet, MultiNet, and RBANet produced F1-measure with 84.04%, 94.88%, and 96.30%, respectively. From the results, we found that similar results were obtained by the proposed methods with different road segmentation network. This indicates that the proposed scale estimation method is not very sensitive to the choice of road segmentation networks. To obtain the experimental results, we used RBANet since it showed slightly better performance than Segnet and MultiNet. For the RBANet, the segmentation process takes about 170ms for $1024 \times 768$ images.

### TABLE 1. Performance of the proposed methods with different segmentation methods.

| Method   | F1-measure (%) | T(%)/min/σ_T (%) |
|----------|----------------|------------------|
| SEMK     | 84.04/1.62/0.31 | 1.66/1.52/0.11  |
| GCSEMK   | 1.40/1.30/0.09  | 1.38/1.29/0.08  |

To obtain the experimental results, we used RBANet since it showed slightly better performance than Segnet and MultiNet. The example trajectories obtained by using GCSEMK are shown in Fig. 9. As can be seen, the estimated vehicle trajectories were closely aligned with the ground-truth vehicle trajectories. The average, minimum, and standard deviation of the translation error are denoted as T(%), min, and $\sigma_T$, respectively. The best is boldfaced.

### C. OVERALL PERFORMANCE OF KITTI DATASET

For the KITTI dataset, we used the image captured from the left camera of the KITTI binocular stereo setup. To evaluate the proposed scale estimation method as comprehensive as possible, we run SEMK and GCSEMK several times on all sequences. The ORB-SLAM occasionally loses tracking on sequence 02, 08, and 09. Since the translation error is accumulated over time, for each sequence, only the camera poses with full trajectory were used for evaluation. We obtained 30 full trajectories for each sequence and computed their average translation error. The performance of the conventional and proposed methods on the KITTI dataset is listed in Table 2. The example trajectories obtained by using GCSEMK are shown in Fig. 9. As can be seen, the estimated vehicle trajectories were closely aligned with the ground-truth vehicle trajectories. The average, minimum, and standard deviation of the translation error are denoted as T(%), min, and $\sigma_T$, respectively. The best is boldfaced.

### FIGURE 8. The performance of the proposed method with different $\delta$. The average result for each sequence with different $\delta$ is shown as the dashed line with different colors. The overall performance is also computed and shown as a thick red line. Similar results are obtained by using SEMK and can be found on our project page. For the sake of visualization, only the results obtained by GCSEMK is presented.
TABLE 2. Comparison of the proposed method with other state-of-the-art monocular visual odometry algorithms in terms of the translation error on the KITTI dataset. The best result is boldfaced.

| Seq | Lee’s [3] | Song’s [9] | Fannani’s [10] | Wang’s [11] | Yin’s [16] | Sugan’s [17] | Yang’s [18] | SEMK | GCSEMK |
|-----|-----------|------------|---------------|------------|-----------|-------------|-----------|------|--------|
|     | T(%)/σT  | T(%)/σT   | T(%)/σT       | T(%)/σT   | T(%)/σT | T(%)/min | T(%)/max | T(%)/min | T(%)/max |
| 00  | 4.42/na  | 2.04/0.33 | 1.65/0.3 a   | 1.01/0.3 a | n/a      | n/a        | 1.97/0 a  | 2.05/1.02/1.02 | 1.09/0.96/0.9 |
| 01  | 6.91/na  | 1.50/0.12 | 0.99/0.1 a   | n/a        | n/a      | 6.18/0 a  | 2.48/na  | n/a        | n/a |
| 03  | 8.49/na  | 3.37/0.54 | 1.98/0.5 a   | n/a        | n/a      | 3.39/0 a  | 1.62/0 a  | n/a        | n/a |
| 04  | 6.21/na  | 1.43/0.61 | 1.16/0.6 a   | n/a        | n/a      | 32.9/0.6 a | 1.12/0 a  | 0.62/0.51/0.07 | 0.72/0.49/0.13 |
| 05  | 5.44/na  | 2.19/0.45 | 1.50/0.4 a   | n/a        | n/a      | 4.47/0 a  | 1.64/0 a  | 0.77/0.73/0.03 | 0.75/0.71/0.03 |
| 06  | 6.51/na  | 2.09/0.78 | 2.93/0.8 a   | n/a        | n/a      | 12.5/0.8 a | 2.26/0 a  | 1.23/0.94/0.14 | 1.06/0.88/0.12 |
| 07  | 6.23/na  | 2.37/0.23 | 1.84/0.2 a   | n/a        | n/a      | 2.22/0 a  | 4.11/0 a  | 0.85/0.68/0.09 | 0.95/0.71/0.11 |
| 08  | 8.23/na  | 1.76/0.18 | 1.56/0.1 a   | n/a        | n/a      | 4.14/0 a  | 11.2/0 a  | 1.66/0 a  | 2.09/0.24/0.08 | 1.08/0.94/0.27 |
| 10  | 9.11/na  | 2.12/0.14 | 1.83/0.1 a   | 0.93/0 a   | 1.70/0 a | 16.8/0 a  | 1.46/0 a  | 1.55/0.96/0.31 | 1.37/0.99/0.21 |
| avg | 6.86/na  | 2.03/0.49 | 1.54/0.4 a   | n/a        | n/a      | 2.68/0 a  | 9.75/0 a  | 1.78/0 a  | 1.54/1.23/0.21 | 1.31/1.19/0.07 |

FIGURE 9. The estimated camera trajectories using GCSEMK for (a) sequence 00 and (b)-(j) sequences 02-10.

followed by the Fannani’s method and SEMK, with a translation error of 1.54%. For sequence 00, GCSEMK and wang’s methods provided the best result, with a translation error of 1.01%. For sequence 01, Only Lee’s and Fannani’s methods obtained results for the full-trajectory, as listed in Table 2. This sequence was a highway scenario with frequently repeated features. The images were captured only at 10fps in spite of the vehicle speed of 90km/h. The proposed method failed at sequence 01 because the 3DGPs collected across multiple keyframes were still insufficient in several parts of the sequence. For sequence 02, the best results were obtained by Fanani’s method, with a translation error of 0.99%, followed by Song’s method and the proposed method. For the sequences 03-08, the proposed methods, SEMK and GCSEMK, considerably improved the accuracy of translation compared with conventional methods. Particularly, for the sequence 07 in which the road region exhibited less texture than in other sequences and, for a short time, over 70% of the frames are occluded by a large truck, SEMK and GCSEMK obtained superior results to the other methods. As the proposed method used multi-keyframes, SEMK and GCSEMK can reliably collect the 3DGPs from the partially occluded keyframes. Comparing the results of SEMK and GCSEMK, we can conclude that identifying the normal vector of the ground plane in the null space of the motion vector is an effective and efficient strategy. In this case, the average translation error was improved from 1.54% to 1.31%. In addition, σT of GCSEMK is smaller than that of SEMK, indicating that more stable results can be obtained with the guidance of the movement vector of a camera. For sequence 10, Wang’s method provided the best result, with a translation error of 0.93%, followed by GCSEMK, with an error of 1.37%. Video results can be found from our project webpage.4

D. IMPACT OF SCALE DRIFT ELIMINATION

Table 3 provides the ATE results obtained using the standard ORB-SLAM, OS-GTS, OS-NL, Loo’s method, Wang’s method, SEMK, and GCSEMK. As listed in Table 3, there

https://scaleestimation.github.io/MonoScaleEstimation/
TABLE 3. The ATE results of OS-GTS, ORB-SLAM, OS-NL, Loo’s method [19], Wang’s method [11], SEMK, and GCSEMK.

| Seq | length (m) | OS-GTS | ORB-SLAM | OS-NL | Loo’s [19] | Wang’s [11] | SEMK | GCSEMK |
|-----|------------|--------|----------|--------|------------|-------------|-------|--------|
| 00  | 3735m      | 4.44   | 6.68     | 69.71  | 17.53      | 5.36        | 10.04 | 4.25   |
| 02  | 4965m      | 4.44   | 21.75    | 41.00  | 36.23      | n.a         | 8.94  | 8.30   |
| 03  | 561m       | 0.25   | 1.59     | 1.02   | 0.90       | 1.36        | 0.68  | 0.79   |
| 04  | 393m       | 0.14   | 1.79     | 0.93   | 2.44       | 1.60        | 0.38  | 0.38   |
| 05  | 2212m      | 1.36   | 8.23     | 40.35  | 29.55      | 8.03        | 2.58  | 2.45   |
| 06  | 1227m      | 1.26   | 14.68    | 52.23  | 47.60      | 18.36       | 2.04  | 1.88   |
| 07  | 690m       | 0.70   | 3.36     | 16.55  | 15.83      | 5.16        | 3.21  | 1.02   |
| 08  | 3213m      | 2.85   | 46.58    | 51.52  | 55.46      | n.a         | 5.84  | 5.68   |
| 09  | 1278m      | 0.73   | 7.62     | 58.14  | 19.29      | n.a         | 2.65  | 2.28   |
| 10  | 897m       | 1.96   | 8.68     | 8.83   | 4.84       | 2.33        | 2.06  | 2.2    |
|     | Average    | 1.96   | 12.10    | 35.09  | 12.66      | 6.17        | 4.30  | 2.75   |

FIGURE 10. The ATE results obtained by SEMK and GCSEMK of each keyframe for (a) sequence 05, (b) sequence 06, (c) sequence 07, and (d) sequence 08.

FIGURE 11. The estimated camera trajectories using GCSEMK for Oxford Robotcar sequence (a) 2014/05/06/12/54/54 and (b) 2014/06/25/16/22/15.

TABLE 4. Absolute keyframe trajectory RMSE (in meters) on Oxford Robotcar dataset.

| Seq     | length (m) | ORB-SLAM | Loo’s [19] | SEMK | GCSEMK |
|---------|------------|----------|------------|------|--------|
| 2014/05/06/12/54/54 | 1030m | 17.75 | 8.66 | 3.80 | 3.83 |
| 2014/06/25/16/22/15 | 716m  | 7.65  | 3.70 | 5.13 | 5.15 |

are great gaps between the ATE results obtained using the GT scale and the scale estimated via the loop closure. This indicates that, if the scale factor is estimated accurately, we can significantly improve the performance of monocular visual odometry/SLAM in a large-scale environment just applying the scale correction. Among these methods, GCSEMK obtained the best result with an average root mean square error (RMSE) of 2.75m. For the short-distance sequences 03 and 04, all methods obtained appropriate results. The ATE results obtained for each keyframe are shown in Fig. 10. The results of the other sequences are available on the project webpage. The results of SEMK and GCSEMK are marked using green dashed lines and blue solid lines, respectively. For each sequence, the blue solid lines are generally below the green dashed lines, indicating that, in road driving scenarios, the normal vector of the ground surface traveled by the vehicle can be effectively found from the nullspace of the movement direction of the vehicle.

E. EXPERIMENT RESULT ON RobotCar DATASET

For the Robotcar dataset, we used the image captured from the center camera of the Bumblebee XB3 trinocular stereo setup. The real-world camera height was set to $h_{gt} = 1.58m$ according to Robotcar dataset. For a fair comparison with Loo’s method [19], the first 200 frames for each RobotCar sequence were skipped because the images were highly over-exposed at the beginning. In addition, the images were cropped to $1248 \times 376$. The results of the proposed methods were obtained by using the same parameters as used for the KITTI dataset. The ATE results and estimated trajectories are shown in Table 4 and Fig. 11, respectively. For sequence 2014/05/06/12/54/54, the SEMK and GCSEMK obtained better results than Loo’s method [19] and ORB-SLAM. The best results for sequence 2014/06/25/16/22/15 was obtained by Loo’s method since their method is more robust to the illumination changes than the ORB-SLAM, SEMK, and GCSEMK. We found that for the short trajectory, such as sequence 03, 04 of the KITTI dataset and the sequences of RobotCar dataset, the performance of SEMK and GCSEMK are similar. The experimental results indicate that the proposed method can accurately estimate the scale factor if the camera pose is reliably provided by ORB-SLAM.

F. DISCUSSION OF LIMITATIONS OF THE PROPOSED METHOD

The proposed 3DGP collection method has been made based on the assumption that the multiple keyframes share similar fields of view. When the view of the scene changes...
FIGURE 12. The process on the non-planar surface. Due to insufficient 3DGPs, the previously computed scale factor is used to correct the camera pose of keyframes included by the solid black box: (a) The estimated (blue asterisks) and ground truth (green asterisks) trajectories obtained for subsequence(470 – 700) of sequence 10 shown in y – z plane. The collected 3DGPs are shown in red points; (b) the number of collected 3DGPs via z; (c) the used scale factor via z; (d) the RMSE (in meters) between ground truth and estimated trajectories; (e)-(g) the corresponding images of the keyframes marked as blue, green, and yellow circles in (a). The matched feature points between frames are denoted as blue and red points. Red circles represent the feature points correctly matched on the ground region. Between the dashed line, the number of the collected 3DGPs is smaller than 15.

V. CONCLUSION

In this paper, aiming to estimate the scale factor accurately, we first collected 3DGPs across multiple keyframes. Then, we collected 3DGPs for each keyframe in the co-visibility graph. The proposed 3DGP collection method provides a sufficiently large number of 3DGPs for robust plane-fitting. As a result, we can only use the 3D plane-fitting algorithm to accurately compute the scale factor without additional process. To further reduce the scale estimation error caused by the inevitable uncertainty of reconstructed 3DGPs, we found the normal vector of the 3D ground plane in the null space of a movement direction of the camera center. We modeled the 3D plane-fitting process as a constrained optimization problem and provided a closed-form solution. The experimental results demonstrated that the proposed geometric constraint can effectively reduce the solution space of the normal vector. The performance on the challenging KITTI dataset showed that ORB-SLAM with the proposed scale estimation method outperforms the state-of-the-art conventional methods.

ACKNOWLEDGMENT

(Ming Fan and Seung-Wook Kim contributed equally to this work.)

REFERENCES

[1] H. Strasdat, J. Montiel, and A. J. Davison, “Scale drift-aware large scale monocular SLAM,” in Robotics: Science and Systems, vol. 2, no. 3. Spain: Zaragoza, 2010, p. 7.
[2] L. A. Clemente, A. J. Davison, I. D. Reid, J. Neira, and J. D. Tardós, “Mapping large loops with a single hand-held camera,” in Robotics: Science and Systems, vol. 2, no. 2. Cambridge, MA, USA: MIT Press, 2007.
[3] B. Lee, K. Daniilidis, and D. D. Lee, “Online self-supervised monocular visual odometry for ground vehicles,” in Proc. IEEE Int. Conf. Robot. Automat. (ICRA), May 2015, pp. 5232–5238.
[4] D. P. Frost, O. Kähler, and D. W. Murray, “Object-aware bundle adjustment for correcting monocular scale drift.” in Proc. IEEE Int. Conf. Robot. Automat. (ICRA), May 2016, pp. 4770–4776.
[5] T. Botterill, S. Mills, and R. Green, “Correcting scale drift by object recognition in single-camera SLAM,” IEEE Trans. Cybern., vol. 43, no. 6, pp. 1767–1780, Dec. 2013.
[6] A. Geiger, J. Ziegler, and C. Stiller, “Stereoscan: Dense 3D reconstruction in real-time,” in Proc. IEEE Intell. Vehicles Symp., Jun. 2011, pp. 963–968.
[7] D. Zhou, Y. Dai, and H. Li, “Reliable scale estimation and correction for monocular visual odometry,” in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2016, pp. 490–495.
S. Song and M. Chandraker, “Robust scale estimation in real-time monocular SFM for autonomous driving,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 1566–1573.

S. Song, M. Chandraker, and C. C. Guest, “High accuracy monocular SFM and scale correction for autonomous driving,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 4, pp. 730–743, Apr. 2016.

N. Fanani, A. Sturck, M. Barnad, and R. Mester, “Modular system for monocular scale estimation,” in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2017, pp. 1714–1721.

X. Wang, H. Zhang, X. Yin, M. Du, and Q. Chen, “Monocular visual odometry scale recovery using geometrical constraint,” in Proc. IEEE Int. Conf. Robot. Automat. (ICRA), May 2018, pp. 988–995.

J. Gräter, T. Schwarz, and M. Lauer, “Robust scale estimation for monocular visual odometry using structure from motion and vanishing points,” in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2015, pp. 475–480.

G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg, “Hierarchical optimization on manifolds for online 2D and 3D mapping,” in Proc. IEEE Int. Conf. Robot. Automat., May 2010, pp. 273–278.

A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp. 3354–3361.

M. A. Fischler and R. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” Commun. ACM, vol. 24, no. 6, pp. 381–395, 1981.

X. Yin, X. Wang, X. Du, and Q. Chen, “Scale recovery for monocular visual odometry using depth estimated with deep convolutional neural fields,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 5870–5878.

E. Sucar and J.-B. Hayet, “Bayesian scale estimation for monocular SLAM based on generic object detection for correcting scale drift,” in Proc. IEEE Int. Conf. Rob. Automat. (ICRA), May 2018, pp. 1–7.

S. Yang and S. Scherer, “CubeSLAM: Monocular 3-D object SLAM,” IEEE Trans. Robot., vol. 35, no. 4, pp. 925–938, Aug. 2019.

S. Y. Loo, A. J. Amiri, S. Mashohor, S. H. Tang, and H. Zhang, “CNN-SVO: Improving the mapping in semi-direct visual odometry using single-image depth prediction,” in Proc. Int. Conf. Robot. Automat. (ICRA), May 2019, pp. 5218–5223.

P. H. S. Torr and A. Zisserman, “MLESAC: A new robust estimator with application to estimating image geometry,” Comput. Vis. Image Understand., vol. 78, no. 1, pp. 138–156, Apr. 2000.

J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, “1-point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry,” J. Field Robot., vol. 27, no. 5, pp. 609–631, Sep. 2010.

G. Klein and D. Murray, “Parallel tracking and mapping for small AR worksspaces,” in Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality, Nov. 2007, pp. 1–10.

T. Taketomi, H. Uchiyama, and S. Ikeka, “Visual SLAM algorithms: A survey from 2010 to 2016,” IPSJ Trans. Comput. Vis. Appl., vol. 9, no. 1, p. 16, Dec. 2017.

A. Q. Li, A. Coskun, S. M. Doherty, S. Ghasemlou, A. S. Jagtap, M. Modasshir, S. Rahman, A. Singh, M. Xanthidis, J. M. O’Kane, and I. Rekleitis, “Experimental comparison of open source vision-based state estimation algorithms,” in Proc. Int. Symp. Exp. Robot. Cham, Switzerland: Springer, 2016, pp. 775–786.

X. Lin, F. Wang, L. Guo, and W. Zhang, “An automatic key-frame selection method for monocular visual odometry of ground vehicle,” IEEE Access, vol. 7, pp. 70742–70754, 2019.

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

C. Forster, Z. Zhang, M. Gassner, M. Welberger, and D. Scaramuzza, “SVO: Semidirect visual odometry for monocular and multicamera systems,” IEEE Trans. Robot., vol. 33, no. 2, pp. 249–265, Apr. 2017.

J. Engel, T. Schöps, and D. Cremers, “LSD-SLAM: Large-scale direct monocular SLAM,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2014, pp. 834–849.

J. Engel, V. Kolten, and D. Cremers, “Direct sparse odometry,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 3, pp. 611–625, Mar. 2016.

G. Younes, D. Asmar, E. Shammas, and J. Zelek, “Keylem-based monocular SLAM: Design, survey, and future directions,” Robot. Auton. Syst., vol. 98, pp. 67–88, Dec. 2017.
SEUNG-WOOK KIM (Member, IEEE) received the B.S. and Ph.D. degrees from the School of Electrical Engineering, Korea University, Seoul, South Korea, in 2012 and 2019, respectively. He is currently a Staff Researcher with the Samsung Advanced Institute of Technology, Suwon, South Korea. From 2018 to 2019, he was a Research Professor with the Semiconductor Research Center, Korea University. His research interests include computer vision, image processing, machine learning, and deep learning.

SUNG-TAE KIM received the B.S. degree in electrical engineering from Korea University, in 2013, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include human–computer interface, digital signal processing, and computer vision.

JEE-YOUNG SUN (Member, IEEE) received the B.S. degree in electrical engineering from Korea University, Seoul, South Korea, in 2014, where she is currently pursuing the Ph.D. degree. She joined the Computer Vision and Image Processing Laboratory, Department of Electrical Engineering, Korea University, in March 2014. Her current research interests include image processing, computer vision, autonomous driving, and artificial intelligence.

SUNG-JEA KO (Fellow, IEEE) received the B.S. degree in electronic engineering from Korea University, in 1980, and the M.S. and Ph.D. degrees in electrical and computer engineering from the State University of New York at Buffalo, in 1986 and 1988, respectively.

In 1992, he joined the Department of Electronic Engineering, Korea University, where he is currently a Professor. From 1988 to 1992, he was an Assistant Professor with the Department of Electrical and Computer Engineering, University of Michigan-Dearborn. He has published over 210 international journal articles. He also holds over 60 registered patents in fields such as video signal processing and computer vision.

Dr. Ko was a 1999 recipient of the LG Research Award. He received the Hae-Dong best paper award from the Institute of Electronics and Information Engineers (IEIE), in 1997, the best paper award from the IEEE Asia Pacific Conference on Circuits and Systems, in 1996, a research excellence award from Korea University, in 2004, and a technical achievement award from the IEEE Consumer Electronics (CE) Society, in 2012. He received a 15-year service award from the TPC of ICCE, in 2014, and the Chester Sall award (First Place Transaction Paper Award) from the IEEE CE Society, in 2017. He was honored with the Science and Technology Achievement Medal from the Korean government, in 2020. He has served as the General Chairman for ITC-CSCE 2012 and the General Chairman for IEICE 2013. He was the President of the IEIE, in 2013, the Vice President of the IEEE CE Society, from 2013 to 2016, and the Distinguished Lecturer of the IEEE, from 2015 to 2017. He is a member of the National Academy of Engineering of Korea. He is a member of the editorial board of the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS.

* * *