IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 January 2023, accepted 31 January 2023, date of publication 3 February 2023, date of current version 8 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3241961

RESEARCH ARTICLE

Dynamic Object-Aware Visual Odometry (VO) Estimation Based on Optical Flow Matching

HAE MIN CHO AND EUNTAI KIM
School of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, South Korea

Corresponding author: Euntai Kim (etkim@yonsei.ac.kr)

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2022-0-01025, Development of core technology for mobile manipulator for 5G edge-based transportation and manipulation).

ABSTRACT In this work, we propose a new visual odometry (VO) system that exploits the dynamic parts of an image. The key idea of our method is to identify the dynamic parts by combining semantic segmentation and optical flow and to suppress the dynamic parts in the process of VO estimation. First, movable objects are detected using the semantic segmentation. If an object contains many pixels of inconsistent optical flow, the object is considered as dynamic and merged with other dynamic objects to create a dynamic mask. Next, the ego-motion of the camera is estimated by using only the remaining static parts of the image and suppressing its dynamic parts. Unlike the other popular deep learning-based VO approaches, our method uses geometric approach based on optimization to achieve high performance. That is, the ego-motion of the camera is obtained by optimizing the correspondences obtained using the dynamic mask and optical flow consistency. Finally, the proposed method is applied to the KITTI Odometry benchmark dataset, and its performance is compared with that of the previous VO methods. Our method achieves an average of 7.67(%) translation error and 2.186 rotation error(°/100m), which imply the performance improvement by 21% and 11% from the state-of-the-art baseline, DF-VO, respectively. In addition, our method yields the RPE (Relative Pose Error) of 0.186m, which is the performance improvement by 52% against ORB-SLAM2, a popular geometry-based method. The experimental results show that the proposed VO method reliably predicts excellent visual odometry in the presence of dynamic objects.

INDEX TERMS Visual odometry, dynamic environment, correspondences.

I. INTRODUCTION

Multi-view geometry-based methods have attracted considerable worldwide attention in monocular visual odometry over the past few decades because of their superior performance. In the past, the ego-motion of the camera was estimated through the correspondences made using hand-crafted features and descriptors [1], [2], [3]. These methods perform relatively well, but they also have the disadvantages of scale-drift issues and performance degradation when applied to dynamic environments. The reason for it is that many existing SLAM and visual odometry (VO) approaches have been developed under the assumption of a static environment, and their ability to extract reliable static visual features is significantly degraded. In other words, if dynamic objects are observed and some features are extracted from the objects, the features extracted from the dynamic objects will be used in the VO estimation while ignoring its relative motion, resulting in a degraded estimation of the camera movement.

On the other hand, we have observed that deep learning has advanced tremendously over the past decade, and it has been applied to many robotics applications including VO. Most of the deep learning-based VO methods use self-supervised learning to predict scale and ego-motion of a monocular camera. Specifically, the deep learning-based VO methods combine a network that extracts depth and another network that predicts pose [4], [5], [6]. End-to-end learning using only image sequences without ground truth is performed by predicting and fusing pose and depth estimations from two consecutive frames. These methods have also obtained

The associate editor coordinating the review of this manuscript and approving it for publication was Chuan Li.

relatively good results, but it is admitted that the performance of the deep learning-based methods is not as good as that of the geometric-based methods.

That is, the disadvantage of the geometry-based method is that the performance might be degraded in the presence of a moving object. Whereas the disadvantage of the deep learning-based method is that the camera movement predicted by the neural network is significantly less accurate than the geometry-based method. Recently, some studies combined both classic geometric methods and deep learning methods to produce good performance by [7] and [8]. Motivated by these works, we propose a novel visual odometry method that combines geometric-based method and deep learning-based method *while being aware of dynamic objects using semantic segmentation*. The candidates for the dynamic parts in the image are selected (1) by using the consistency of optical flow and (2) by checking whether the parts belong to the classes of ''movable objects''. The first condition is realized by using the optical flow network [9], and the second condition is checked by using the semantic segmentation network [10]. Using the two conditions, we can prevent our network from removing too much information by excluding all movable objects. Our network performs optimization using only the static parts with stable optical flow excluding the dynamic parts so that the ego-motion of the camera can be obtained robustly even in the presence of dynamic objects.

In summary, the contribution of the proposed method is as follows:

- A method of creating a dynamic mask by detecting a moving object using semantic segmentation and optical flow difference was proposed. The dynamic mask is applied to the correspondence selection, and optimization is performed using only static parts of the image.
- Three networks are combined to make use of the advantages of the deep learning-based method and the geometry-based method, and perform robust visual odometry in the presence of dynamic objects.

The rest of the paper is organized as follows. Section II reviews the relevant studies, and Section III describes the algorithm proposed in this paper in detail. The evaluation results and a comprehensive ablation study show the effectiveness of the proposed method in Section IV. Finally, Section V concludes the paper.

II. RELATED WORK

A. VISUAL ODOMETRY

Camera tracking, which is the pose estimation between two images, has been a long-studied problem in the field of computer vision. Most follow the Structure-from-Motion(SfM) method [11], a multi-view geometry method that predicts the 3D structure and poses of each image by optimizing two or more 2D images. Among them, many methods find the exact correspondences between two images using features and descriptors and perform bundle adjustment(BA) to obtain ego-motion between the images [1], [12], [13]. However, these methods cannot be used in a low texture environment where it is difficult to extract features. Not only that, it is impossible to predict exact scale with a monocular camera, and in a dynamic environment, it is difficult to predict ego-motion because the extracted features might be on moving objects.

With the development of deep learning, studies have been conducted to predict the pose between two images by training deep neural networks [14], [15], [16]. UnDeepVO [4] proposed an unsupervised method using stereo images to train the network so that the network can learn exact scale, and the network is separated into rotation and translation so that it can predict poses well without setting weights. Some methods have attempted to obtain ego-motion between two images using optical flow [17], [18]. DeepAVO [19] divides the optical flow resulting image into quadrants and uses a CNN network to get the features of each piece. The features are then concatenated altogether and the pose between two images is estimated. Hwang et al. [20] proposed loss functions to reduce the local drift, increasing consistency under dynamic driving scenes to estimate frame-to-frame visual odometry. However, all of these deep learning-based methods failed to accurately predict the odometry. This is because estimation through a network cannot be more accurate than the multi-view geometry method. DF-VO [7], [21] attempted to solve the problem by combining the strengths of the deep learning-based method and the geometry-based method. This succeeded in ignoring many unnecessary parts by using the difference in the optical flow of consecutive frames, but still was not able to completely exclude the dynamic objects. Therefore, we propose a novel visual odometry method that recognizes dynamic objects of a scene and robustly estimates the ego-motion of the camera in a dynamic environment by combining deep learning-based and geometry-based methods.

B. VISUAL ODOMETRY IN DYNAMIC ENVIRONMENT

Moving objects in a dynamic environment have always been a problem in the field of visual odometry. Before deep learning was developed, many studies attempted to use human intervention to perform robust odometry estimation or SLAM in dynamic environments [22], [23]. DM-SLAM [24] proposed distribution and local-based RANSAC to extract static features to make SLAM stable in a dynamic environment. Xu et al. [25] proposed a method that combines the spatial geometric information of the image and removes the moving objects to perform robust visual odometry in dynamic environment. These traditional methods performed quite well in a dynamic environment. However, since they use hand-crafted features, it is necessary to extract the features and compute the descriptors. With the development of deep learning, a number of deep learning-based monocular VO methods have been proposed to solve the problem. DynaSLAM [26] proposed a method to improve SLAM performance by applying semantic segmentation to ORB-SLAM2 [1] to detect dynamic

objects and perform background impainting. Its extension DynaSLAM II [27] solves object pose estimation, object tracking, and SLAM through joint optimization. Kuo et al. citekuo2020dynamic proposed a system that predicts the pose between two images by creating a mask that corrects weights for each class. By training a network to predict the attention of each class and reflecting the attention into the input image and optical flow, the ego-motion of the camera can be estimated by focusing more on static objects. However, these methods have the disadvantage that the results are still not accurate compared to the optimization methods because the pose obtained through the network is quite unstable.

In this study, we propose a VO system that uses semantic segmentation and optical flow to identify dynamic parts of an image and uses classic geometry-based methods to obtain precise ego-motion of a monocular camera in a dynamic environment. A detailed explanation about proposed method is given in the following sections.

III. METHOD

In this section, we will introduce the proposed method in detail. First, the overview of the framework will be given and how its components interact with each other will be explained.

A. OVERVIEW

Fig. 1 illustrates an outline of the proposed method. The proposed method consists of three networks: optical flow estimation network, depth estimation network, and semantic segmentation network. When two consecutive images at times $t-1$ and t are presented as an image pair, forward and backward optical flows ($t \rightarrow t-1$, $t-1 \rightarrow t$), depth and semantic segmentation images at time t are estimated. The key feature of our method is to identify the dynamic parts in the given image pair and to improve the VO estimation by suppressing the dynamic parts. Here, we identify the dynamic parts using the flow consistency computed with the optical flows developed in [21]. Unfortunately, however, the flow consistency alone cannot segment dynamic objects *at the pixel level* and more needs to be done. In this paper, we also apply a semantic segmentation network to predict the classes in the image and segment dynamic objects at the pixel level. Specifically, for all the objects that are segmented and classified as "`movable`" class by a semantic segmentation network, we compute the average of flow consistency within each movable object. The state (=whether it is dynamic or static) of a movable object is predicted using the average flow consistency value within the object. If the object is considered as dynamic, the region of the object in the image is selected as an object mask and it is merged to create a single dynamic mask. The static parts of the image can be obtained by excluding the regions covered by the dynamic mask. Among the static parts in two consecutive images, we select the only pixel correspondences with high flow consistency because the correspondences with similar forward and backward flows are reliable. The ego-motion between the two images is calculated with the correspondences using multi-view geometry and scale is restored using the estimated depth map. In the following part, we first describe calculating flow consistency and finding dynamic objects. Then, how to select correspondences, estimate pose, and restore scale will be explained.

B. DEEP NETWORKS

Our system consists of three deep networks, as shown in Fig. 1. The three networks are all trained separately, and they are combined together to build our system. The three networks are not developed but used simply as components in this paper. So, they will be covered briefly in this section.

1) OPTICAL FLOW NETWORK

Dense optical flow, simply optical flow, is a pattern of motion between two images. Specifically, when a pair of images (I_i, I_j) is presented, the optical flow gives the 2D to 2D pixel association between a pixel in one image I_i and its corresponding pixel in the other image I_j. Because it is not easy to label the ground truth for actual image pairs, the network is trained using a synthetic dataset image. For a backbone network for optical flow, a fast, lightweight, and accurate LiteFlowNet [9] trained on a synthetic dataset Scene Flow [29] is employed in this paper. However, the optical flow networks have the drawback that their performance is significantly degraded when the given images have few patterns, or they include some dynamic objects. To solve the problem, we use consistency of the optical flow and semantic segmentation in this paper.

2) DEPTH NETWORK

Monocular depth estimation network applies CNN to a single image and estimates a depth map of the image. A depth network allows us to estimate dense depth maps even in non-textured environments and helps to solve the problem of scale ambiguity. First, the depth network is trained with a pose network simultaneously in an unsupervised way [15]. The unsupervised training is conducted without the ground truth using the relationship between a dense depth map and ego-motion. Then, only the depth network is taken from the combined system consisting of depth and the pose networks, and it is used as a monocular depth estimation network in our system. In this paper, Monodepth2 [30] which is high-performance and robust to occlusion is used as the depth network.

3) SEMANTIC/INSTANCE SEGMENTATION NETWORK

The semantic/instance segmentation network (1) segments movable objects in the given images at the pixel level and (2) identifies every individual instance separately. The reason why we use not a simple semantic segmentation network but a semantic/instance segmentation network is that we cannot specify which movable objects are actually moving using only semantic information. For example, some cars

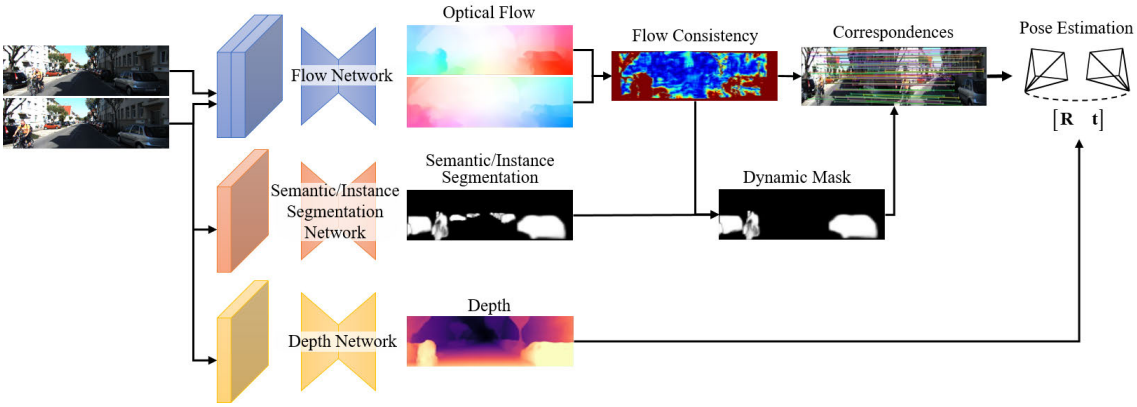

FIGURE 1. Overview of the proposed method.

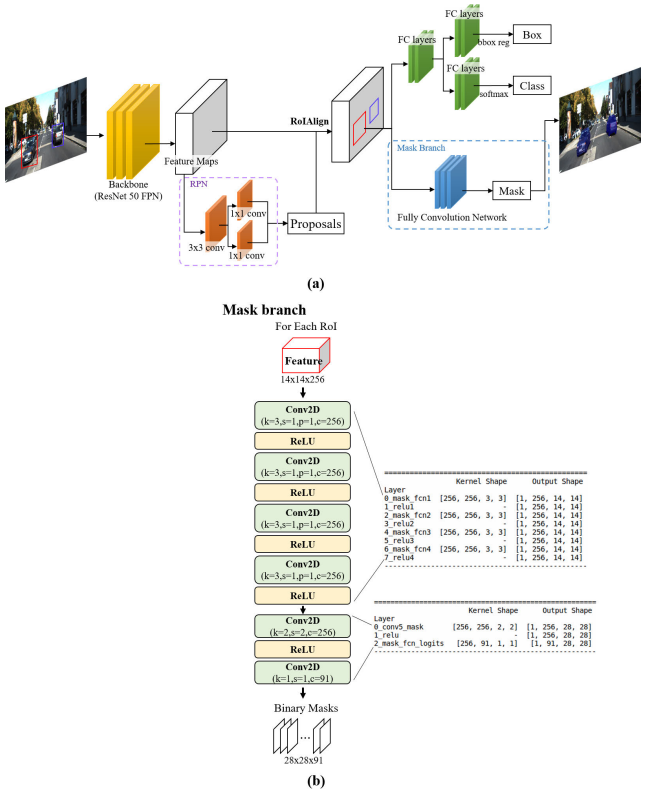

FIGURE 2. (a) Structure of Mask R-CNN. MASK R-CNN performs instance segmentation by predicting the RoI for an object from an input image and performing classification. (b) Structure of the mask branch.

are parked on the side of the road, not moving at all, while other cars move fast. Thus, we combine the output of the semantic/instance segmentation with the output of the flow network to determine which instance is moving.

In this paper, we use a Mask R-CNN [10] trained on MS COCO dataset [31] as the semantic/instance segmentation network. R-CNN is a network that performs object detection for a single image, and it consists of 2 stages. In the first stage, R-CNN generates a proposal as object candidates. In the second stage, R-CNN classifies the proposal and finds an object in the proposal. Mask R-CNN [10] is one of the R-CNN variations, and it detects all objects in an image and segments each instance at the pixel level. The architecture of Mask R-CNN is given in Fig. 2.

In Mask R-CNN, we used ResNet 50 FPN as a backbone, and fine-tune the network on the MS COCO dataset. Thus, the mask branches in the network outputs 91 object classes. The architecture of our mask R-CNN is the same as the original mask R-CNN with only one exception in the mask branch. In our implementation, the mask branch is modified to take the features of size $14 \times 14 \times 256$ from RoI (Region of Interest) and outputs binary mask of size $28 \times 28 \times 91$. Our mask consists of 4 blocks of 2D convolutional layer and ReLU, followed by one more block of conv2D layer that outputs binary masks. In the implementation, only 10 classes are considered for movable objects because they usually appear on roads. The 10 classes are described in detail in Section III-D.

C. FLOW CONSISTENCY

Classic visual odometry methods predict the correspondence between two images using hand-crafted feature points or gradient values of the images [32], [33]. Different from the classical VO methods which use hand-crafted features, we use optical flow for correspondence to analyze and use images in various environments in this paper. The classical feature-based methods perform well in relatively good environments, but their performance is greatly degraded with adversarial environments such as severe illumination change or insufficient texture. Meanwhile, deep learning-based methods extract robust features from the images and work better than the classical methods in diverse environments. Specifically, we use the optical flow from LiteFlowNet [9] to estimate the correspondence between two images. Using optical flow for correspondence is taken from [21].

Here, the optical flow estimation is better than the classical hand-crafted feature-based correspondence methods but it should also be admitted that the optical flow estimation might

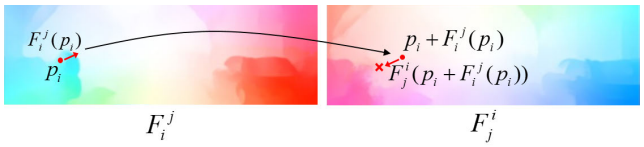

FIGURE 3. An example of the optical flow relationship between two frames. F_i^j is the optical flow of frame I_i to frame I_j, F_j^i is the optical flow of frame I_j to frame I_i.

not be enough: Using pixels from every flow will result in incorrect pixel correspondence selection and the subsequent performance can be degraded in the pose estimation. To solve the problem, flow consistency is used to evaluate the "confidence" of each optical flow. The concept of flow consistency is shown in Fig. 3. If the flow in one pixel p is consistent with the corresponding pixel $p + F_i^j(p)$ in the other image, or equivalently if the flow difference defined by

$$D_i(p) = \left| F_i^j(p) + F_j^i(p + F_i^j(p)) \right| \tag{1}$$

is quite low, the confidence of the correspondence can be regarded as high and we can use them in the pose estimation, where F_i^j is flow from frame I_i to frame I_j and vice versa. Meanwhile, if the flow in one pixel p is not consistent with the corresponding pixel $p+F_i^j(p)$ and it has the high difference $D_i(p)$, the pixels are likely to belong to dynamic object and they should not be used in the pose estimation. To realize our idea that we use only the pixels with high confidence based on the flow consistency, we can now create dynamic masks and pixel correspondences.

D. DYNAMIC MASK

Flow consistency in Section III-B alone can identify some pixels which are very likely to belong to dynamic objects as in [21]. Understandably, however, it is almost impossible to remove the dynamic objects at the pixel level, or equivalently to identify *all* the pixels which belong to the dynamic objects. Thus, when we identify dynamic pixels using only flow consistency, some of the dynamic pixels will not be identified and the remaining will be used in the pose estimation, resulting in incorrect correspondence selection and degradation of the pose estimation. To solve the problem, we apply a semantic segmentation network to our system to find dynamic parts that cannot be identified by the flow consistency alone. In this paper, we use the well-known Mask R-CNN [10] as a semantic segmentation network. Furthermore, we define `movable objects` such as pedestrians and vehicles among the classes of the semantic segmentation network's output. In this paper, a total of 10 classes {`'person'`, `'bike'`, `'car'`, `'motorcycle'`, `'airplane'`, `'bus'`, `'train'`, `'truck'`, `'boat'`, `'cat'`, `'dog'`} are defined as `movable objects`. The `movable objects` can either move or stay at the same location in the image. When the objects are moving, they are dynamic; but when the objects are not moving, they are static. To predict the state of each object, we combine the output of the semantic segmentation and flow consistency. Fig. 4 shows the dynamic mask for dynamic objects obtained using semantic segmentation and flow consistency. When an input image is given as in Fig. 4-(a), several binary object masks M_l are generated by Mask R-CNN as shown in Fig. 4-(b), where M_l denotes the l-th binary object mask. Then, the average of the flow difference within the object mask M_l is computed by

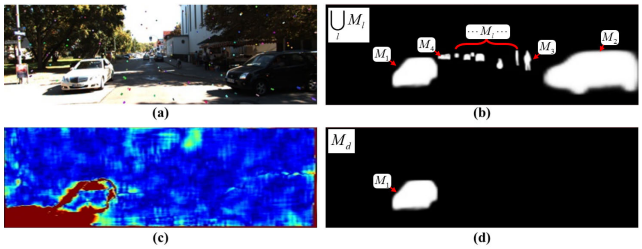

FIGURE 4. Dynamic mask obtained using flow consistency. (a) Current input frame. (b) Binary mask for all movable objects obtained by semantic segmentation network. (c) Flow consistency of the current frame with the previous frame. (d) Dynamic mask for the objects that are currently moving based on flow consistency.

$$D_l = \frac{1}{N_l} \sum_{p \in M_l} D(p), \tag{2}$$

where N_l denotes the number of pixels which belong to the l-th object mask M_l. In Fig. 4-(c), the flow difference is depicted. In the figure, high values are colored in red, whereas low values are colored in blue. The masks with low average consistency (=high D_l) can be considered as currently moving objects. That is, only the object masks M_l which has D_l value higher than a threshold $\theta_{flowdiff}$ are included in a dynamic mask M_d. In other words, the binary dynamic mask M_d is made by

$$M_d = \bigcup_l M_l \text{ where } \frac{1}{N_l} \sum_{p \in M_l} D(p) > \theta_{flowdiff}. \tag{3}$$

As can be seen in the figure, the dynamic mask (d) is created by removing non-moving objects by applying the flow consistency (c) to the movable objects detected by semantic segmentation (b). With our dynamic mask M_d, which shows the dynamic objects that are currently moving in the frame, dynamic objects-aware visual odometry estimation can be performed.

E. CORRESPONDENCE SELECTION

In this section, pixel correspondences are created using the flow consistency and the dynamic mask to estimate the pose between two consecutive frames. To accurately calculate the ego-motion of the camera, correspondences must be spread uniformly across the image domain, rather than focused in one place. So we divide the image into grids, and for each grid, a pair of points with high flow consistency (=low flow difference) is selected so the correspondences are evenly distributed across the image. To prevent numerous correspondences from being made in areas that shouldn't be

TABLE 1. Quantitative result on KITTI Odometry Seq. 00-10. The best result is in bold and second best is underlined.

	method	00		01		02		03		04		05		06		07		08		09		10	
		t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}
SfMLearner	deep learning-based	21.32	6.19	**22.41**	2.79	24.1	4.18	12.56	4.52	4.32	3.28	12.99	4.66	15.55	5.58	12.61	6.31	10.66	3.75	11.32	4.07	15.25	4.06
Depth-VO-Feat		6.23	2.44	23.78	1.75	6.59	2.26	15.76	10.62	3.14	2.02	4.94	2.34	5.80	2.06	3.56	2.39	5.45	2.39	11.89	3.70	12.82	3.41
SC-SfMLearner		11.01	3.39	27.09	1.31	6.74	1.96	9.22	4.93	4.22	2.01	6.70	2.38	5.36	1.65	8.29	4.53	8.11	2.61	7.64	2.19	10.74	4.58
ORB-SLAM2 (w/o LC)	geometry-based	11.43	0.58	107.57	0.89	10.34	**0.26**	0.97	**0.19**	1.30	**0.27**	9.04	0.26	14.56	0.26	9.77	0.36	11.46	**0.28**	9.31	0.26	**2.66**	0.39
ORB-SLAM2 (w LC)		2.35	**0.35**	109.10	**0.45**	3.32	0.31	**0.91**	**0.19**	1.56	**0.27**	1.84	**0.20**	4.99	**0.23**	1.91	**0.28**	9.41	0.30	2.84	**0.25**	2.67	**0.38**
DF-VO	deep learning-based + geometry-based	2.01	0.79	61.17	18.96	**2.46**	0.79	3.27	0.89	0.79	0.56	1.5	0.74	1.95	0.76	2.28	1.16	**2.11**	0.74	3.21	0.59	2.89	0.97
Ours (Full mask)		**1.76**	0.80	102.89	23.95	2.71	0.79	3.31	0.95	**0.58**	0.49	1.48	0.78	1.71	0.61	1.40	0.80	2.33	0.78	2.78	0.58	3.05	1.02
Ours (Dynamic mask)		1.77	0.79	64.38	16.87	2.62	0.74	3.06	0.89	0.65	0.55	**1.31**	0.74	**1.6**	0.56	**1.06**	0.67	2.28	0.76	**2.66**	0.53	2.95	0.95

t_{err}: Average translational RMSE drift (%) on length from 100, 200 to 800 m.
r_{err}: Average rotational RMSE drift ($^\circ/100m$) on length from 100, 200 to 800 m

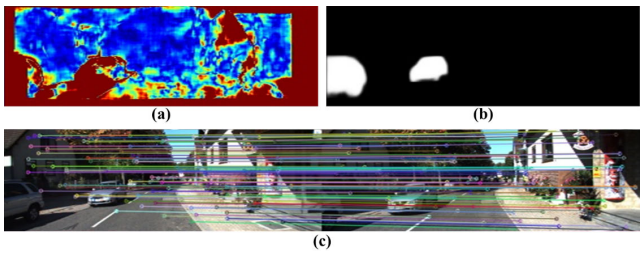

FIGURE 5. Pixel correspondences between two consecutive frames, corresponding flow consistency, and dynamic mask. (a) Magnitude of the flow consistency. (b) Dynamic mask of the current frame. (c) 2D-2D correspondences.

(vehicles, etc.), the dynamic mask created earlier is applied. With the dynamic mask M_d, we can consider only the static part of the frame when selecting pixel correspondences. That is, the ego-motion between the two frames can be obtained robustly using only the stable part by excluding the part where the flow consistency is low and the part with dynamic objects. The example of the pixel correspondences is shown in Fig. 5. Each pixel correspondence in correspondences (c) is represented by a small circle and connected with a line with the same color. As can be seen in the figure, the correspondence does not exist in the dynamic mask part. That is, dynamic objects, which are in a moving state, are excluded due to the dynamic mask, so more correspondences are selected in the static part with high flow consistency.

F. POSE ESTIMATION

Finally, the pose between the two consecutive frames is predicted based on the 2D-2D correspondences obtained in Section III-E. In the prediction, we use epipolar geometry [34], [35]. First, we define the geometric relationship between two consecutive images viewed by an epipolar constraint given in terms of the essential matrix $\mathbf{E}$.

$$\mathbf{p}_j^T \mathbf{K}^{-T} \mathbf{E} \mathbf{K}^{-1} \mathbf{p}_i = 0, \tag{4}$$

where $(\mathbf{p}_i, \mathbf{p}_j)$ is a pair of pixel correspondences of the two frames. The camera motion $[\mathbf{R}, \mathbf{t}]$ can be solved by decomposing the essential matrix $\mathbf{E}$. That is,

$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}, \tag{5}$$

where $[\mathbf{t}]_\times$ means the matrix representation of the cross product with $\mathbf{t}$.

However, the essential matrix alone cannot recover the scale of the pose, which is a chronic problem of monocular visual odometry. To recover the scale, a scale recovery process is required. The method of restoring the scale is the same as [21], which restores the scale using the depth map estimated from the depth network. If there is a sufficient number of correspondences, the scale is restored according to the ratio with the depth map, and if there is a limited number of correspondences, ego-motion is calculated through 2D-3D geometry calculation. That is, when the size of the optical flow is small due to small camera movement, the perspective-n-point (PnP) method for the predicted depth map is used.

IV. EXPERIMENT

In this section, the experimental results of the proposed method are shown qualitatively and quantitatively.

A. IMPLEMENTATION DETAIL

The proposed framework is implemented using Pytorch [36] framework and is trained on an Intel Core i5-7600 desktop computer with 32 GB RAM and NVIDIA GeForce 1070 Ti GPU. The depth network is trained with the famous KITTI Odometry benchmark dataset [37], which contains 11 driving sequences (00-10) with ground truth camera poses. The sequences consist of images with size 640 × 192. Following the protocol of [21], we train our depth network on sequences 00-08.

Each network is trained separately and used in the proposed method. For optimization, the Adam optimizer [38] is used, and the optical flow network used the weights of [21]. That

TABLE 2. Network performance with different hyper-parameters.

Parameter		Depth Network	Segmentation Network
		Relative Square Error	AP@[.5:.95]
Learning rate	10^{-3}	0.1003	0.334
	10^{-4}	0.0946	**0.318**
	10^{-5}	**0.0911**	0.328
Weight decay	0	0.0911	0.318
	10^{-4}	0.0927	**0.297**
	10^{-5}	**0.0871**	0.314

FIGURE 6. Examples of input images and augmented images.

is, the flow network is trained with a learning rate of 10^{-4} for the first 15 epochs and 10^{-5} for the rest. The depth network is fine-tuned based on the weights of [21] using the stereo images of the KITTI dataset, and the semantic/instance segmentation network is fine-tuned based on the weights of the pre-trained network with the COCO dataset [31]. For both networks, we conducted a training experiment while changing the initial learning rate η from 10^{-3} to 10^{-5} by increasing one order of magnitude. Also, we conducted the same experiment while varying weight decay λ used in L2 regularization. The results are given in Table 2.

In Table 2, the best value for each network is shown in bold. Since each network is trained and used separately, the weight of the bolded results are used as the weight of each network. That is, the learning rate of 10^{-5} and weight decay of 10^{-5} are used to train the depth network learning and while the learning rate of 10^{-4} and weight decay of 10^{-4} are used to train the semantic/instance segmentation network.

During the training process, augmentation methods are applied to prevent the underfitting/overfitting. Figure 6 is an example of data augmentation methods used when training the depth network. As can be seen in the figure, the color jittering is applied to the input image while changing brightness, hue, etc. Also, horizontal flip operation is applied randomly to improve the performance of the neural network.

B. COMPARISON ON THE KITTI DATASET

We evaluated the proposed method and other methods and compared the trajectory accuracy for all 11 sequences in the KITTI Odometry dataset, as in [21]. In the experiments, our VO method is compared with three kinds of VO methods. The first kind is deep learning-based method such as SfMLearner [15], Depth-VO-Feat [39], SC-SFMLearner [16], the second kind is classical geometry-based method, which is ORB-SLAM2 [1]; and the third one is hybrid method which includes DF-VO [21]. Understandably, our method also belongs to the hybrid method. The quantitative results are shown in Table 1. In the table, it should be noted that two different results are listed for ORB-SLAM2 because its performance highly depends on whether the loop is closed or not. All competing methods are compared in terms of rotation error $r_{err}(^\circ/100m)$ and translation error $t_{err}(\%)$, which are averaged over 100m to 800m intervals. The best result among all methods is indicated in bold, and the second-best result is underlined. Fig. 7 shows the qualitative results of eight sequences in the KITTI Odometry dataset. As shown in the table and figure, our method demonstrates good performance compared to other methods by using the dynamic mask. Specifically, most of our results are much better than those of the learning-based methods. Compared to geometry-based method (=ORB-SLAM2), our method is generally superior in terms of translation error, but might be inferior in terms of some rotational errors. This is because the geometry-based method also performs well using the optimization over multiple frames. To measure the accuracy of visual odometry between two images, we compared the RPE (Relative Pose Error) of ORB-SLAM2 and the RPE of the proposed method, which is shown in Table 3. RPE is a frame-to-frame error that indicates how accurately the odometry is predicted between consecutive frames. As shown in Table 3, the proposed method shows lower (=better) or similar RPE results than ORB-SLAM2 in most sequences except one sequence. Therefore, our method outperforms the geometry-based method for predicting ego-motion between two consecutive frames. By applying semantic segmentation and optical flow to effectively remove dynamic objects of the image, our method can obtain better and more reliable performance. This shows that the proposed method has the potential to outperform the geometry-based method even for rotational error.

In most sequences that contain dynamic objects, defined in Section III-D, our method achieves good performance by suppressing the region of dynamic objects in the image. Among the hybrid methods that use deep learning-based method and geometry-based method together, our method works best in sequences with many dynamic objects. That is, in most sequences, the proposed method performs better than [21]. Further, it is interesting to see that using a full mask also works well for some sequences. In sequence 04, it is better to use the full mask rather than the dynamic mask. This is because when the ego-motion of the camera (vehicle) is too large and most objects are moving, it is not easy to judge the movement by only the optical flow. In this case, it is better to just suppressing the influence of all movable objects by applying the full mask. However, when static objects are dominant in the image, the performance of the full mask might be inferior to even not using any mask because too much information is lost with the mask. In this case, using a dynamic mask rather than a full mask is better in performance. This demonstrates the dangers of blindly removing all movable objects and the necessity of dynamic detection like our method. In general, the proposed method using a

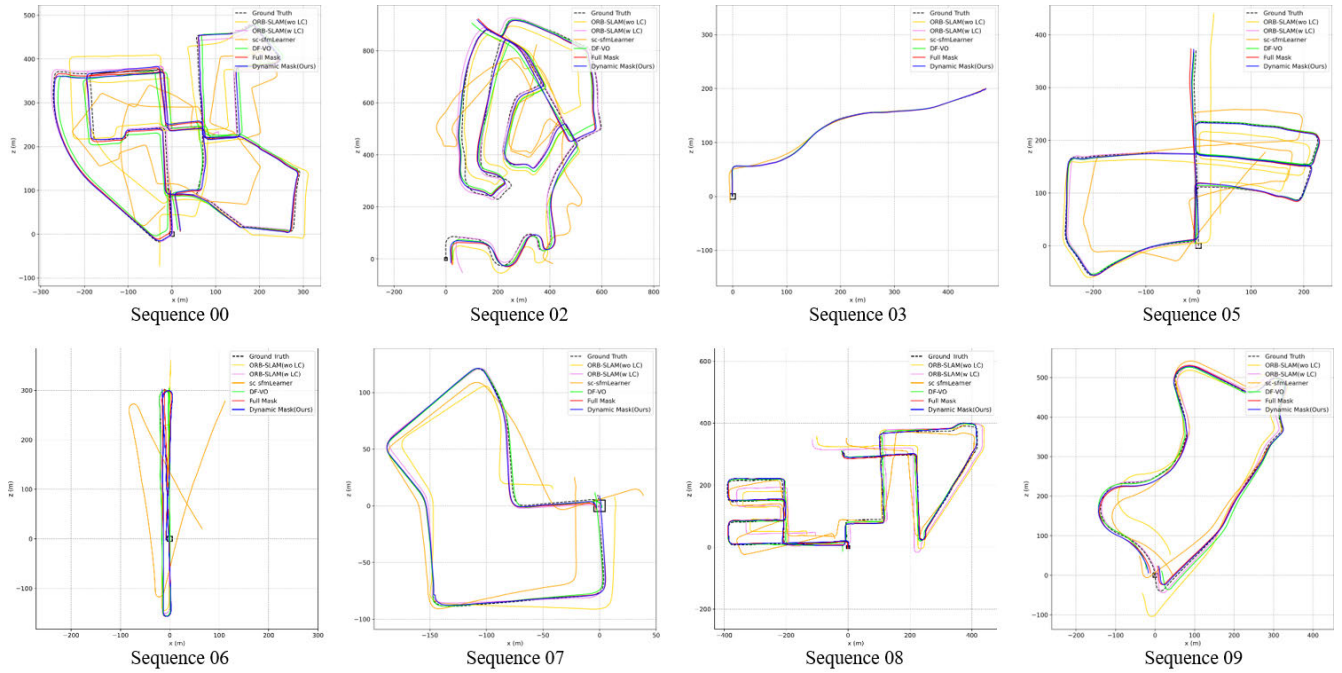

FIGURE 7. Qualitative VO results on KITTI benchmark dataset.

TABLE 3. Comparison between the proposed method and the state-of-the-art geometry-based method ORB-SLAM2 in Relative pose error. The best result is in bold. The proposed method performed better or equal in most of the sequences.

	00		01		02		03		04		05		06		07		08		09		10	
	RPE (m)	RPE (°)	RPE (m)	RPE (°)	RPE (m)	RPE (°)	RPE (m)	RPE (°)	RPE (m)	RPE (°)	RPE (m)	RPE (°)	RPE (m)	RPE (°)	RPE (m)	RPE (°)	RPE (m)	RPE (°)	RPE (m)	RPE (°)	RPE (m)	RPE (°)
ORB-SLAM2 (w/o LC)	0.17	0.08	2.97	0.10	0.17	0.07	0.03	0.06	0.08	0.08	0.14	0.06	0.24	0.06	0.11	0.05	0.19	0.06	0.13	0.06	**0.05**	0.07
ORB-SLAM2 (w LC)	0.21	0.09	3.04	**0.09**	0.22	0.08	0.04	0.06	0.08	0.08	0.29	0.06	0.73	**0.05**	0.51	0.05	0.16	0.07	0.34	0.06	**0.05**	0.07
Proposed method	**0.03**	**0.06**	**1.71**	0.67	**0.04**	**0.06**	**0.02**	**0.04**	**0.04**	**0.04**	**0.02**	**0.05**	**0.03**	**0.05**	**0.02**	**0.04**	**0.03**	**0.05**	**0.06**	**0.05**	**0.05**	**0.06**

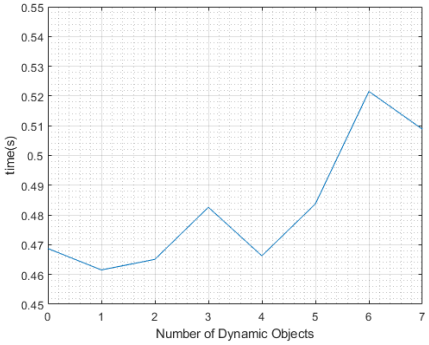

FIGURE 8. Runtime vs. the number of moving objects.

dynamic mask demonstrates excellent VO results in dynamic environments.

C. RUNTIME ANALYSIS

In this section, we conduct some additional experiments to investigate the relation between the runtime of our algorithm and the number of moving objects. In the experiment, we measure the runtime for various numbers of moving objects. Here, we used the 00 sequence of KITTI dataset as a test set because it includes a variety of situations. The results are shown in Fig. 8. Figure 8 is the runtime vs. the number of moving objects. As can be seen in the figure, the runtime of the proposed method increases roughly in proportion to the number of moving objects. That is, the more moving objects appear, the longer computation is required, thereby decreasing the processing speed.

V. CONCLUSION

This paper has proposed a novel visual odometry method that works well in dynamic environments by detecting parts with dynamic objects. Our VO is based on the idea of detecting dynamic objects from movable objects by combining optical flow and semantic segmentation. By creating a dynamic mask using dynamic objects, it was able to remove the unnecessary information in the image. Based on the dynamic mask and difference in optical flow between the two images, pixel correspondences are selected in the static part. The selected correspondences are applied to a

geometry-based method to obtain the ego-motion of the camera through optimization. Combining the learning-based method and the geometry-based method allowed us to perform dynamic object-aware visual odometry using the advantages of both methods. We evaluate the proposed method on the well-known KITTI Odometry benchmark dataset. Experimental results show that our method predicts the ego-motion of the camera well in a dynamic environment by excluding the information of the dynamic objects. Regarding the applicability of the proposed method, we believe that our algorithm can be applied to various situations in which a number of nearly moving objects prevent the accurate estimation of VO. For example, when a robot navigates in a crowded museum or department store, we believe that our VO will outperform the classical VO's because our method estimates the odometry while removing the effects of moving objects.

We think that our method has two limitations. First, our method consists of three networks, but the three are trained separately without considering the other two networks. Second, only frame-to-frame odometry (=odometry based on two consecutive frames) is calculated; and the odometry based on more than two consecutive frames is not considered at all, reducing the accuracy of VO. Thus, we believe that we can improve our VO method by resolving the limitations of the current method in the future work. Specifically, first, we can develop a way that the three networks are trained simultaneously, or our system is trained end-to-end. Second, we have to construct a graph to estimate the VO based on more than two consecutive frames. The two items are recommended as our future work.

REFERENCES

[1] R. Mur-Artal and J. D. Tardós, "ORB-SLAM$_2$: An open-source slam system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[2] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 1691–1696.

[3] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, "An overview to visual odometry and visual SLAM: Applications to mobile robotics," *Intell. Ind. Syst.*, vol. 1, no. 4, pp. 289–311, Nov. 2015.

[4] R. Li, S. Wang, Z. Long, and D. Gu, "UnDeepVO: Monocular visual odometry through unsupervised deep learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 7286–7291.

[5] N. Yang, L. von Stumberg, R. Wang, and D. Cremers, "D3 VO: Deep depth, deep pose and deep uncertainty for monocular visual odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1281–1292.

[6] Q. Jia, Y. Pu, J. Chen, J. Cheng, C. Liao, and X. Yang, "D^{2}VO: Monocular deep direct visual odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10158–10165.

[7] H. Zhan, C. S. Weerasekera, J.-W. Bian, R. Garg, and I. Reid, "DF-VO: What should be learnt for visual odometry?" 2021, *arXiv:2103.00933*.

[8] W. Zhao, S. Liu, Y. Shu, and Y.-J. Liu, "Towards better generalization: Joint depth-pose learning without PoseNet," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9151–9161.

[9] T.-W. Hui, X. Tang, and C. C. Loy, "LiteFlowNet: A lightweight convolutional neural network for optical flow estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8981–8989.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.

[11] S. Ullman, "The interpretation of structure from motion," *Proc. Roy. Soc. London B, Biol. Sci.*, vol. 203, no. 1153, pp. 405–426, Jan. 1979.

[12] R. Mur-Artal, J. M. 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.

[13] H. M. Cho, H. Jo, and E. Kim, "SP-SLAM: Surfel-point simultaneous localization and mapping," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 5, pp. 2568–2579, Oct. 2022.

[14] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 2043–2050.

[15] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1851–1858.

[16] J. Bian, "Unsupervised scale-consistent depth and ego-motion learning from monocular video," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 35–45.

[17] Z. Yin and J. Shi, "GeoNet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1983–1992.

[18] C. Zhao, L. Sun, P. Purkait, T. Duckett, and R. Stolkin, "Learning monocular visual odometry with dense 3D mapping from dense 3D flow," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 6864–6871.

[19] R. Zhu, M. Yang, W. Liu, R. Song, B. Yan, and Z. Xiao, "DeepAVO: Efficient pose refining with feature distilling for deep visual odometry," *Neurocomputing*, vol. 467, pp. 22–35, Jan. 2022.

[20] S. Hwang, M. Cho, Y. Ban, and K. Lee, "Frame-to-frame visual odometry estimation network with error relaxation method," *IEEE Access*, vol. 10, pp. 109994–110002, 2022.

[21] H. Zhan, C. S. Weerasekera, J.-W. Bian, and I. Reid, "Visual odometry revisited: What should be learnt?" in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 4203–4210.

[22] R. A. Newcombe, D. Fox, and S. M. Seitz, "DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 343–352.

[23] R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon, and D. Cremers, "StaticFusion: Background reconstruction for dense RGB-D SLAM in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 3849–3856.

[24] J. Cheng, Z. Wang, H. Zhou, L. Li, and J. Yao, "DM-SLAM: A feature-based SLAM system for rigid dynamic scenes," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 4, p. 202, Mar. 2020.

[25] G. Xu, Z. Yu, G. Xing, X. Zhang, and F. Pan, "Visual odometry algorithm based on geometric prior for dynamic environments," *Int. J. Adv. Manuf. Technol.*, vol. 122, no. 1, pp. 235–242, Sep. 2022.

[26] 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.

[27] B. Bescos, C. Campos, J. D. Tardos, and J. Neira, "DynaSLAM II: Tightly-coupled multi-object tracking and SLAM," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5191–5198, Jul. 2021.

[28] X.-Y. Kuo, C. Liu, K.-C. Lin, and C.-Y. Lee, "Dynamic attention-based visual odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 36–37.

[29] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[30] C. Godard, O. M. Aodha, M. Firman, and G. Brostow, "Digging into self-supervised monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3828–3838.

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft COCO: Common Objects in Context*. Cham, Switzerland: Springer, 2014, pp. 740–755.

[32] 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.

[33] 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.

[34] Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review," *Int. J. Comput. Vis.*, vol. 27, no. 2, pp. 161–195, 1998.
[35] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[36] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. NIPS-W*, 2017.
[37] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Rob. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.
[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[39] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. M. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 340–349.

HAE MIN CHO received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2015, where she is currently pursuing the Ph.D. degree in electrical and electronic engineering. Her research interests include visual simultaneous localization and mapping and computer vision.

EUNTAI KIM received the B.S., M.S., and Ph.D. degrees in electronic engineering from Yonsei University, South Korea, in 1992, 1994, and 1999, respectively. From 1999 to 2002, he was a Full-Time Lecturer at the Department of Control and Instrumentation Engineering, Hankyong National University, South Korea. Since 2002, he has been a Faculty Member at the School of Electrical and Electronic Engineering, Yonsei University, where he is currently a Professor. He was a Visiting Researcher at the Berkeley Initiative in Soft Computing, University of California, USA, in 2008. He was a Visiting Researcher at the Korea Institute of Science and Technology (KIST), South Korea, in 2018. His current research interests include computational intelligence, statistical machine learning and deep learning, and their application to intelligent robotics, autonomous vehicles, and robot vision.