IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 21, 2019, accepted December 4, 2019, date of publication December 9, 2019,
date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2958374

DSOD: DSO in Dynamic Environments

PING MA[1,2], YUE BAI[1], JUNAN ZHU[1,2], CHUNJUN WANG[3], AND CHENG PENG[1]
[1]Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
[3]CRSC Research and Design Institute Group Company, Ltd., Beijing 100049, China

Corresponding author: Yue Bai (baiy@ciomp.ac.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 11372309 and Grant 61304017, in part by the Jilin Province Science and Technology Development Program under Grant 20150204074GX and Grant 20160204010NY, in part by the Provincial and Academic Cooperation Science and Technology Special Fund under Grant 2017SYHZ0024, in part by the Innovation Fund Project of Chinese Academy of Sciences under Grant GQRC-19-13, and in part by the Chinese Academy of Sciences Youth Promotion Program under Grant 2014192.

ABSTRACT Recently, visual simultaneous localization and mapping (SLAM) has been widely used in robotics and autonomous vehicles. It performs well in static environments. However, real-world environments are often dynamic scenarios. Because it is difficult for SLAM to deal with moving objects such as pedestrians and moving cars, SLAM does not meet the actual needs of robots and autonomous vehicles in real-world scenarios. Visual odometry (VO) is a key component of SLAM systems. In this paper, to extend SLAM to dynamic scenarios, we propose a monocular VO based on direct sparse odometry (DSO) to solve the problems arising in a dynamic environment. The proposed method, called DSO-Dynamic (DSOD), combines a semantic segmentation network with a depth prediction network to provide prior depth and semantic information. Experiments were conducted on the KITTI and Cityscapes datasets, and the results show our method achieves good performance compared with the baseline algorithm, DSO.

INDEX TERMS DSO, dynamic environments, segmentation network, depth prediction network.

I. INTRODUCTION

In recent years, visual simultaneous localization and mapping (SLAM) has made many gains in accuracy and effectiveness because of the continuous efforts of many outstanding researchers. Visual odometry (VO) is the frontend of a SLAM system. The aim of SLAM and VO is to construct a map of an unknown environment with simultaneous localization in the updated map. SLAM frontends are mainly divided into two categories, indirect and direct methods. ORB-SLAM2 [1] is a representative algorithm of the indirect methods that employs the ORB descriptor [2] to match features extracted from adjacent frames so as to minimize the reprojected coordinate errors. Because the indirect method needs to extract features and compute their descriptors, these processes are computationally complex, which affects the real-time performance of the algorithm. Furthermore, an indirect method cannot extract enough features from low-texture environments. Therefore, Jakob *et al.* were the first to propose a direct method called DSO [3]. DSO utilizes a projection of the pixel intensities from adjacent frames to minimize the photometric error. This avoids consuming a lot of resources to compute the descriptor. DSO, which selects pixels according to their photometric values in the given grid, increases robustness in low-texture environments. In general, the direct method has some advantages over the indirect method.

However, these methods can only accurately estimate pose from static landmarks, and they are inaccurate in dynamic environments. There are still some problems to be solved, the most prominent of which is how to deal with moving objects in dynamic environments. RDSLAM [4] uses the RANSAC approach to filter out a part of the mismatched points. An M-estimator (such as the Huber norm) enhances the robustness to outliers of the squared-error loss function; it can filter out a small number of dynamic points, which are considered to be outliers because their depth cannot be converged. This problem still does not have a good solution, so we propose a novel monocular VO system called DSO-Dynamic (DSOD) to solve it. DSOD is based on DSO. We identify potential dynamic points in the scene and use prior information from the semantic segmentation of the current observed scene to determine the state of these points. Then, the remaining points are used to calculate the pose of the camera.

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

The main contributions of this paper are as follows:

1. A depth prediction network is introduced into our method to predict the initial depth and hence improve the scale problem of a monocular SLAM system.
2. Because the moving objects in an environment effect the accuracy of a SLAM system, a semantic segmentation network similar to Mask R-CNN [5] is introduced to provide a pixel-wise segmentation to discriminate potential moving objects from scenarios.
3. Experiments were carried out on the KITTI [6] and Cityscapes [7] datasets, and the results show our method performs better than the baseline DSO algorithm.

The remainder of this article is organized as follows. Closely related work is reviewed in Section II. Our proposed method DSOD is described in detail in Section III. In Section IV, the results of experiments on the KITTI and Cityscapes datasets are presented. Finally, we conclude this paper in Section V.

II. RELATED WORK

A. DEPTH PREDICTION IN SLAM

Geometry-based visual SLAM has no prior information about the current observed environment. Stereo DSO [8] can provide the depth map by direct stereo matching, and LDSO [9] is a DSO improved by the addition of a loop closure; it can correct the pose and scale by recognizing the same scene via a bag-of-words approach [10], [11]. Convolutional neural networks (CNNs) have achieved exceptional performance in computer vision. They have been introduced to depth prediction and obtained great performance improvements. Yevhen *et al.* proposed an improved DSO that employs semi-supervised learning [12] to predict the depth map [13]. CNN-SVO [14], which is based on SVO [15], employs depth prediction using unsupervised learning [16]. These algorithms, which use depth prediction to initialize the depth of the currently observed frame, are effective and can improve the performance of a SLAM system. DSO, which is a pure monocular system, is a state-of-the-art monocular algorithm, but is inaccurate when estimating the scale of a map reconstruction. Hence, we introduce a depth prediction network into DSO to provide prior knowledge of depth.

B. SEMANTIC SEGMENTATION IN SLAM

In recent years, most CNN-based segmentation networks such as SegNet [17], U-Net [18], Fully Convolutional DenseNet [19], and Mask R-CNN have mainly focused on detecting objects and building the semantic map in SLAM. Sünderhauf *et al.* [20] associate the semantic label with an object based on the nearest-neighbor method from a three-dimensional point segmentation and then update the point-cloud information of the target object. Abhijit *et al.* [21] remove points whose semantic label varies between observed frames. Vineet *et al.* [22] use the semantic label during fusion in a three-dimensional model to weight the measurements. Bao *et al.* [23] estimate camera pose by exploiting geometric and semantic information about the frame simultaneously to improve the accuracy of object recognition. DS-SLAM [24], which is based on the ORB-SLAM2 framework, utilizes a SegNet to filter out any people, which are considered to be dynamic objects in the scene. Brasch *et al.* [25] proposed Semantic Monocular SLAM for Highly Dynamic Environments. It models the dynamic points with a joint probabilistic model based on prior semantic information about the observed scene. To obtain prior information about objects in the observed dynamic environment for a monocular SLAM system, it is helpful to apply semantic segmentation to mark objects in the current frame and then classify the points into static and dynamic sets.

C. DYNAMIC ENVIRONMENTS IN SLAM

Most SLAM approaches are not robust to dynamic objects contained in real-world scenes. Because an inertial measurement unit can directly measure the change in the pose of a camera, some SLAM systems [26], [27], [28] add an inertial measurement unit to initialize the pose estimation and compensate for the effects of moving objects in a dynamic environment. Klappstein *et al.* [29] proposed a likelihood score for dynamic objects based on optical flow [30], which corresponds to the motion field in a scene. Alcantarilla *et al.* [31] exploit the scene flow based on residual motion likelihoods to segment dynamic objects. In this work, we employ a segmentation network to segment the dynamic object in an observed scene with monocular depth prediction based on the DSO framework.

III. PROPOSED METHOD

Our proposed method is based on DSO. Hence, we first describe DSO briefly in Section III-A. Then, the overall approach of our proposed algorithm is shown in Section III-B. Depth prediction and semantic segmentation are presented in Sections III-C and III-D, respectively. Lastly, our methods for checking for movement consistency and filtering out dynamic points are explained in Section III-E.

A. BASELINE DSO METHOD

Suppose a point set N_p from a reference frame I_i is observed in current frame I_j with respect to exposure times t_i and t_j. Then, the basic idea of DSO can be formulated as follows:

$$E_{pj} = \sum_{p \in N_p} w_p \left\| (I_j[p'] - b_j) - \frac{t_j e^{a_j}}{t_i e^{a_i}} (I_i[p] - b_i) \right\|_r, \quad (1)$$

where $||\dots||_r$ is the Huber norm and w_p is a weighting that down-weights high image gradients with some constant c as follows:

$$w_p = \frac{c^2}{c^2 + \|\nabla I_i(p)\|_2^2}, \quad (2)$$

where p' is the projection of p in current frame I_j, d_p is the inverse depth of p, and T_{ji} is the pose transformation from

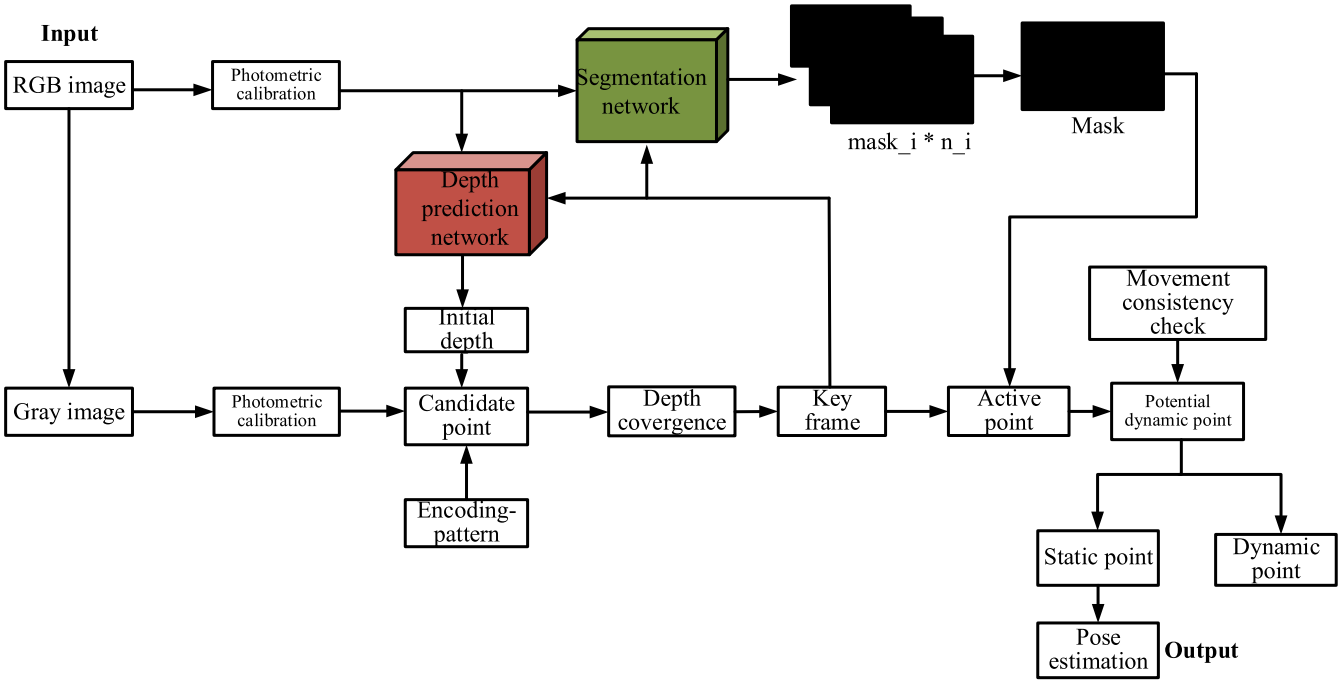


FIGURE 1. Overview of the DSOD framework. We incorporate a depth prediction network and semantic network to improve the SLAM system. We then check the consistency of the movement of potential dynamic points to update the status of each active point.

frame i to frame j. Projection p' can be calculated by

$$p' = \prod_K (T_{ji} \prod_K^{-1} (p, d_p)). \tag{3}$$

B. DSOD FRAMEWORK

The overall framework of our proposed method is shown in Fig. 1. The input of our method consists of RGB image sequences, which are decomposed into red, green, and blue channels. Photometric calibration is applied to the three channels and they are then merged. The corrected RGB images are the input of the depth prediction and segmentation networks. The raw RGB images are also converted to gray images, photometrically calibrated, and then used as the input for point selection.

In contrast to baseline DSO, a depth prediction network is introduced in DSOD to provide the initial depth and encode the point patterns to match the initial location of the projection. The aim is to accelerate the convergence of depth. The network compensates for scale drift to some extent in the monocular SLAM system. In addition, the semantic segmentation network is used to check moving consistency to reduce errors in the dynamic environment. Finally, the output of DSOD is the estimated pose.

C. DEPTH PREDICTION NETWORK

The depth information plays a vital role in a SLAM system and has a substantial influence on depth estimation. However, DSO only selects pixels from the current key frame. Because it uses these pixels, which are initialized with uncertain depth information, as the candidate points, false projection pairs may be generated because of the search along the epipolar line over a large range. Hence, we introduce a depth prediction network into our method. We use an unsupervised monocular depth estimation to predict the initial depth of the candidate point. The estimation provides prior depth information for the candidate point initialization process through a single-image depth prediction network. Each initialized candidate point has a depth with an interval in which the corresponding projection point lies. The network reduces the depth uncertainty and narrows the search interval of an initial candidate point. Furthermore, the single-image depth prediction network accelerates the depth convergence of the candidate point.

In the depth map, *depth* stands for the initial depth value of candidate point p, and it has a confidence value χ. The maximum inverse depth $idepth_{max}$ and the minimum inverse depth $idepth_{min}$ can be calculated by

$$\begin{cases} idepth_{\max} = \dfrac{1}{depth \cdot \chi} \\ idepth_{\min} = \dfrac{1}{depth \cdot (2-\chi)} \end{cases}, 0 < \chi \leq 1. \tag{4}$$

The candidate point p is projected onto the current observed frame. The maximum projection point p'_{max} and minimum projection point p'_{min} can be formulated as follows:

$$p'_{\min} = K \cdot R \cdot K^{-1} \cdot p + K \cdot t \cdot idepth_{\min}, \tag{5}$$

$$p'_{\max} = K \cdot R \cdot K^{-1} \cdot p + K \cdot t \cdot idepth_{\max}. \tag{6}$$

Then, we perform a discrete search in the interval from $[u_{min}\, v_{min}]$ to $[u_{max}\, v_{max}]$ to find the most similar point

pattern. Matrix K is the camera's intrinsic matrix and (R, t) is the initial estimation of the transformation from the reference frame to the current observed frame. We can express $[u_{min}\ v_{min}]$ and $[u_{max}\ v_{max}]$ as

$$\begin{bmatrix} u_{\max} \\ v_{\max} \end{bmatrix} = \begin{bmatrix} p'_{\max}[0]/p'_{\max}[2] \\ p'_{\max}[1]/p'_{\max}[2] \end{bmatrix}, \tag{7}$$

$$\begin{bmatrix} u_{\min} \\ v_{\min} \end{bmatrix} = \begin{bmatrix} p'_{\min}[0]/p'_{\min}[2] \\ p'_{\min}[1]/p'_{\min}[2] \end{bmatrix}. \tag{8}$$

The goal of the matching process is to find reliable projection pairs whose photometric residuals are the smallest in a special pattern in DSOD. Just like census transformation [32] in stereo matching, the proposed initial matching method creates an encoding pattern depending on the photometric values of the points that surround the central point. As shown in Fig. 2, the pattern set is $\{S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$, the surrounding-point set is $\{g_0, g_1, g_2, g_3, g_4, g_5, g_6\}$, the center point is g, and the code sequence is $\{C_0, C_1, C_2, C_3, C_4, C_5, C_6\}$. We define the corresponding code C_i to be 1 if the photometric value of a surrounding point g_i is stronger than central point g; otherwise, C_i is 0. Point g_i and code C_i can be determined by

$$\begin{cases} g_0 = S_0 \\ g_1 = S_1 \\ g_2 = S_2 \\ g_3 = S_3 \\ g_4 = S_5 \\ g_5 = S_6 \\ g_6 = S_7 \\ g = S_4 \end{cases}, \tag{9}$$

$$C_i = \begin{cases} 1 & if\ \mathrm{g}_i - g \geq 0 \\ 0 & if\ \mathrm{g}_i - g < 0 \end{cases} i \in [0, 6]. \tag{10}$$

The candidate point pattern from an optical flow pyramid of the reference frame with the initial depth values is projected onto the current observed frame. The first step is to perform a discrete search along the epipolar line to find a similar point pattern and obtain its position in the current frame. Code C is the code of the candidate point pattern and C' is the code of projection point pattern. Codes C and C' can be expressed as follows:

$$C = [C_0, C_1, C_2, C_3, C_4, C_5, C_6], \tag{11}$$

$$C' = [C'_0, C'_1, C'_2, C'_3, C'_4, C'_5, C'_6]. \tag{12}$$

Variable Q is the result of an XOR operation between C and C'. It can be formulated as follows:

$$Q = C \oplus C'. \tag{13}$$

After we perform an XOR operation between the codes of the candidate point pattern and those of the search point pattern along the epipolar line, the most similar projection is our target pattern. The second step is to perform Gauss–Newton iteration to optimize the projection pairs and update the depth of the candidate point. Finally, we determine whether the depth of the candidate point (with respect to the reference frame) has converged after l iterations. If it has not, this candidate point should not be used for pose estimation.

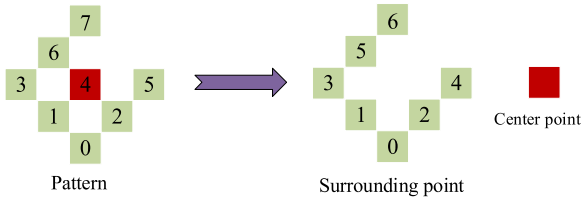


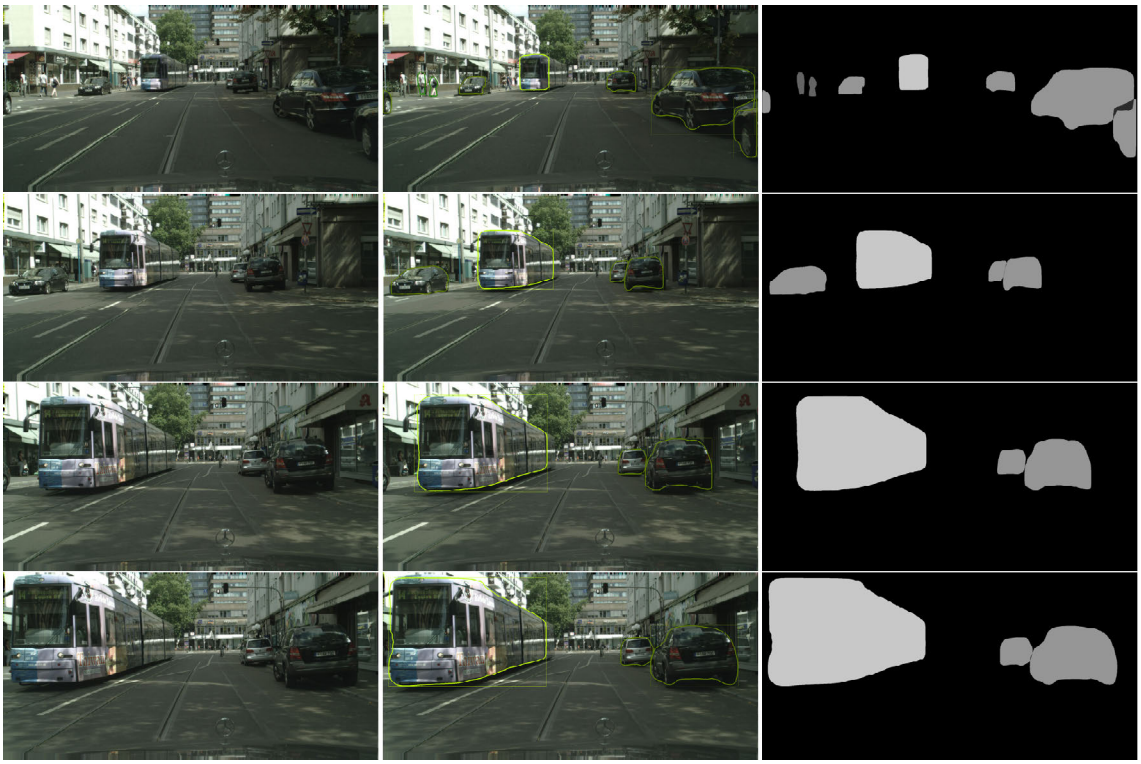
FIGURE 2. Patterns used in the proposed method: (left) original indexing of the pattern and (right) re-indexed surrounding points and center point.

D. SEMANTIC SEGMENTATION NETWORK

The category of an object in the current observed scene can help the processing of complex tasks in dynamic environments. However, purely traditional methods cannot provide this prior information. Hence, the combination of a semantic segmentation network with the traditional SLAM system has recently become a popular research topic. In many existing methods, these networks are trained to recognize the object and then attach the semantic information to the object. These methods focus on semantic mapping and object recognition. However, semantic information is not well used in other parts of the method. More recently, with the development of deep learning, the accuracy of semantic segmentation has improved greatly. In this paper, semantic information is utilized to mark potential dynamic points in a dynamic environment.

In particular, because monocular approaches have no further prior information from the current observed scene, it is difficult to discriminate the static and dynamic parts of the image directly. To increase the accuracy of segmentation, we employ a segmentation network that can provide a pixel-wise segmentation. The segmentation network is trained on the COCO dataset [33], and it can detect 30 classes of objects. Among them, people, cars, bicycles, buses, and motorcycles are defined as potential moving objects. These categories meet the requirements for complex and extremely dynamic scenes.

The segmentation of our method is similar to that of Mask R-CNN, which outputs a series of masks. However, the output of our segmentation is only one mask, and the pixel values of the mask are the corresponding labels. During the segmentation, each mask *mask_i*represents an object O_i, and the pixel value is 1 if there is an object O_i located at that pixel; otherwise, the pixel value is 0. We propose multiplying the mask by a fixed number *n_i* to mark the coordinates and label of the target object. Finally, we sum all the masks to create a final mask and transform it into a grayscale image. In this way, the label of object in the mask image matches the corresponding pixel in the current observed frame. Examples of semantic segmentation are shown in Fig. 3. The final *Mask* is



(a) Raw image (b) Segmentation image (c) Mask image

FIGURE 3. Examples of semantic segmentation: (a) Raw image, (b) Segmentation image, (c) Mask image.

calculated as follows:

$$Mask = \sum_{i=0}^{N} mask\_i * n\_i. \tag{14}$$

E. MOVEMENT CONSISTENCY CHECK

Because we can only obtain the class label of an object in the current observed scene through the segmentation network, the semantic label of a pixel provides the category and coordinates of the object in the current observed frame. Using the semantic label, which is a non-zero fixed value, the system can mark potential moving objects in the mask image, but it cannot determine whether the potential moving object is a real moving object or a fake moving object. Hence, we need to determine the real status of a potential moving object.

To distinguish between real and fake moving objects, we check the consistency of the movements of points of the potential object. All key points from the reference frames are matched to the target projection points in the current observed frame. We then use the pairs to compute the fundamental matrix between each reference frame and the current frame using RANSAC. If the label of a point in the current frame is determined to be a potential moving object, we need to compute the epipolar line in the current frame using the fundamental matrix and the coordinates of the point and projected point.

We assume that p_1 and p_2 respectively denote the point from the reference frame and the corresponding projection point in the current observed frame. Moreover, $(u_1,\ v_1)$ and $(u_2,\ v_2)$ denote the coordinates of a point in the reference frame and the projection point in the current observed frame, respectively. Then, the new homogeneous coordinates p_1 and p_2 are respectively defined as follows:

$$p_1 = \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix}, p_2 = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}. \tag{15}$$

The epipolar line $L=[A,B,Z]^T$ is computed by the fundamental matrix F and point p_1 as

$$L = F \cdot p_1. \tag{16}$$

To determine whether a potential dynamic point is a real or fake dynamic point, the distance between the coordinates of projection point p_2 in the current observed frame and the corresponding epipolar line L is computed by

$$D = \frac{\left| p_2^T \cdot L \right|}{\sqrt{\|A\|^2 + \|B\|^2}}. \tag{17}$$

In this system, real dynamic points are considered to be outliers and fake dynamic points are considered to be inliers. If the distance D exceeds a certain threshold, the projection point is considered to be a real moving point. That is,

$$status = \begin{cases} outlier, D > \varepsilon \\ inlier, D \leq \varepsilon \end{cases}, \tag{18}$$

TABLE 1. Results of DSOD and DSO on the KITTI dataset.

KITTI	DSOD		DSO	
Sequence	t_{rel} (%)	r_{rel} (°/rad)	t_{rel} (%)	r_{rel} (°/rad)
01	4.2767	0.0010887	93.7506	0.0013752
02	15.0481	0.0021774	67.0263	0.0021201
03	16.3702	0.0015471	91.3737	0.0017190
04	3.2991	0.0005157	98.4965	0.0013179
05	14.4391	0.0018909	52.2164	0.0020055
06	14.6982	0.0019482	59.8516	0.0020628
07	15.5274	0.0029796	56.2375	0.016617
08	15.1056	0.0024639	50.7294	0.0025212
09	13.8505	0.0018336	72.2738	0.0020055
10	13.5269	0.0021201	80.8137	0.0021201

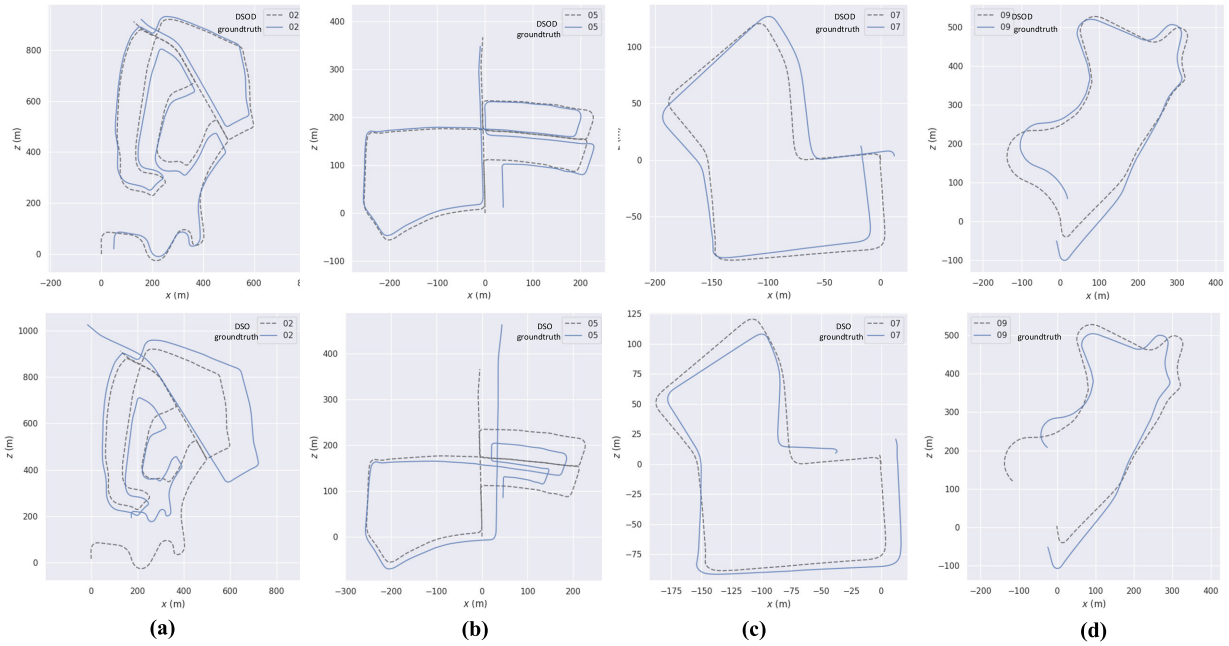


FIGURE 4. Trajectories of DSOD (top row) and DSO (bottom row) on (a) sequence 02, (b) sequence 05, (c) sequence 07, and (d) sequence 09 of the KITTI dataset.

where ϵ is the threshold. The corresponding point is then removed from the reference frame to keep it from participating in pose estimation.

IV. EXPERIMENTS

In this section, our algorithm is compared with the baseline DSO method. Experiments were conducted on the KITTI and Cityscapes datasets. To fairly evaluate our algorithm, our method and DSO were both evaluated on an Intel i7-7700K CPU with a GTX 1060 GPU card. Our algorithm can run at an average speed of 8 frames per second (fps). We describe the experiments on the KITTI and Cityscapes datasets in detail in Sections IV-A and IV-B, respectively. An ablative study on the Cityscapes dataset and further discussion about the depth prediction and semantic segmentation are described in Section IV-C.

A. EXPERIMENTS ON THE KITTI DATASET

The KITTI dataset is wildly used to evaluate the SLAM system. In the KITTI dataset, the image sequences contain a variety of moving objects, such as people, cars, and buses. The main evaluation metric is the rotational root mean squared error (r_{rel}) and translational root mean squared error (t_{rel}). Both r_{rel} and t_{rel} were averaged over 100 m to 800 m intervals. The evaluation tool is publicly available for download.[1]

The results are shown in Table 1, which presents the translational errors and rotational errors for the challenging sequences (01–10) from the KITTI dataset. The results show that r_{rel} and t_{rel} have been substantially reduced by the use of DSOD in all sequences.

[1] http://www.cvlibs.net/datasets/kitti/eval_odometry.php

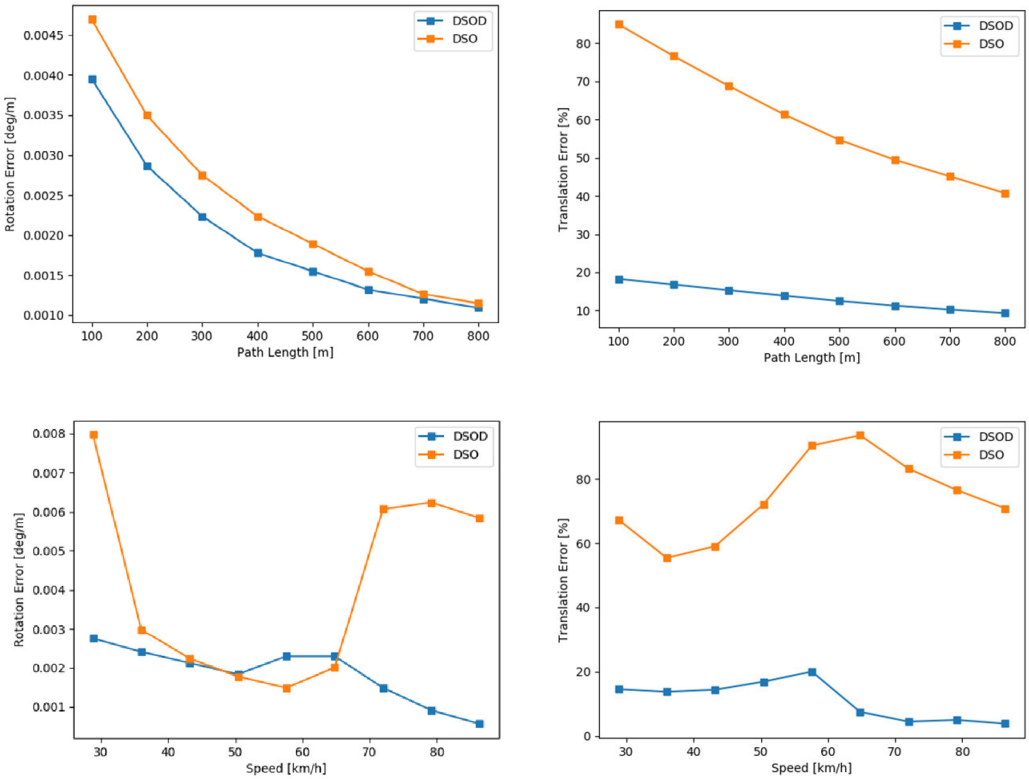


FIGURE 5. Results for (left) rotation errors and (right) translation errors for (top row) path length and (bottom row) speed on the KITTI dataset.

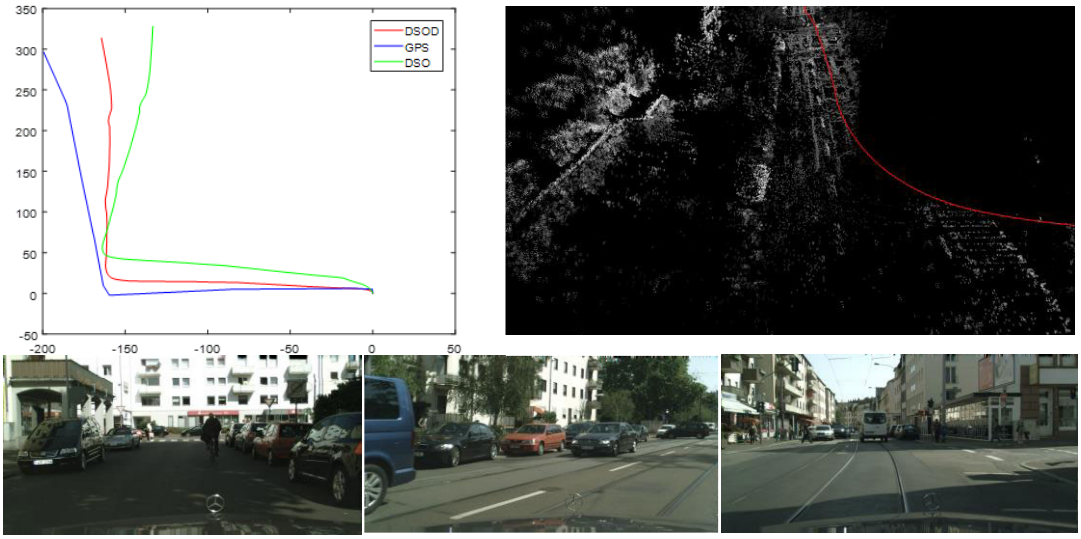


FIGURE 6. To simulate a real-world dynamic scene, 2,215 frames were selected from the Frankfurt sequence. (Top left) Trajectory. The ground truth (blue) has some errors caused by the low accuracy of the GPS location. The performance of DSOD (red) is better than that of DSO (green) when the scale is corrected. (Top right) Reconstruction result. (Bottom row) Actual road conditions.

To better demonstrate the effectiveness of our algorithm, we selected some representative sequences. The trajectories of several sequences[2] are shown in Fig. 4. Because our algorithm and DSO only deal with key frames, the lengths of the pose value sequences are not the same. However, the evaluation tool requires the sequences to be of the same length, so we extract the corresponding pose value of the key frames from the ground truth separately to match the results of DSOD and DSO and plot their trajectories.

In Fig. 5, we plot the translational errors and rotation errors for path length and speed, respectively. These errors

[2]The trajectory drawing tool can be download from https://github.com/MichaelGrupp/evo.

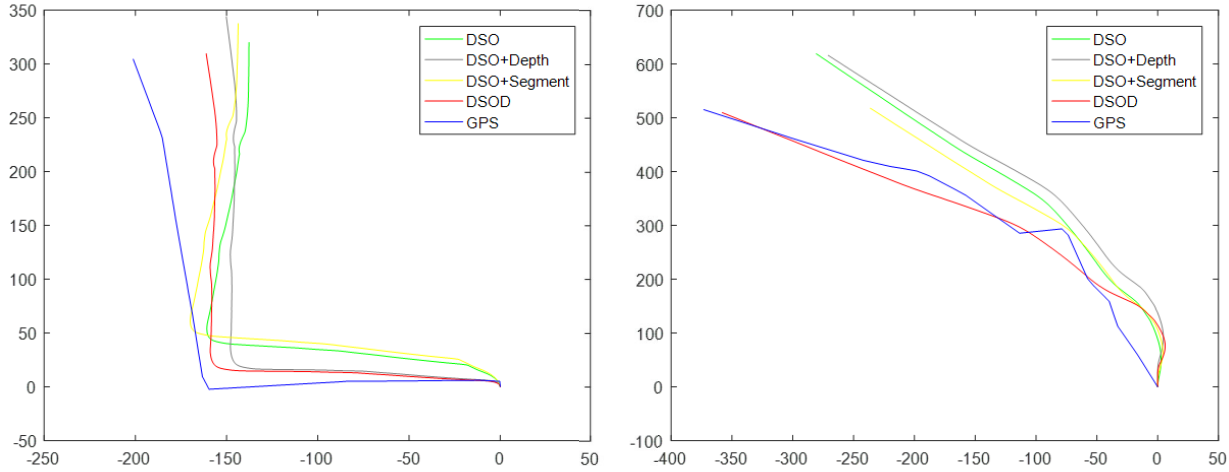


FIGURE 7. Trajectory of two image sequences from the Cityscapes dataset. (Left) Frames 374–2588. (Right) Frames 3544–4860. The DSO and DSO+Segment results are enlarged so that they are at the correct scale.

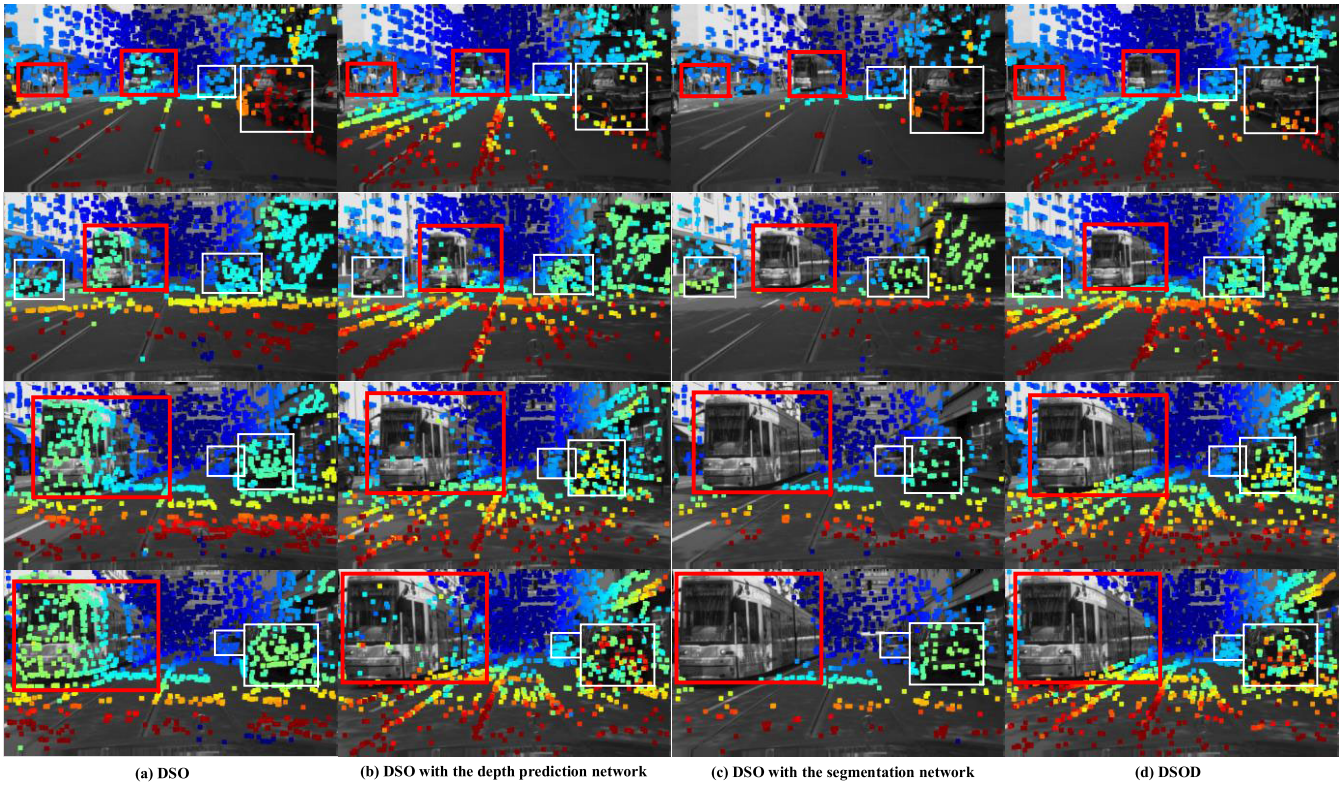


FIGURE 8. Intermediate results for pose estimation on several frames: (a) intermediate results of DSO, (b) intermediate results of DSO+Depth, (c) intermediate result of DSO+Segment, and (d) intermediate result of DSOD. The red boxes indicate the real moving objects and the white boxes represent fake moving objects.

indicate the errors between our results and the ground truth. Smaller error values indicate that the algorithm works better. As shown in Fig. 5, all errors are reduced when DSOD is used. Consequently, our algorithm performs better than DSO.

B. EXPERIMENTS ON THE CITYSCAPES DATASET

Cityscapes is a large-scale dataset that focuses on the semantic understanding of urban street scenes. It can also be used for evaluating SLAM systems. This dataset contains a variety of dynamic scenes, such as cars and pedestrians, so it is suitable for evaluating our DSOD, which focuses on dealing with dynamic environments.

An image sequence was selected from the Frankfurt sequence in Cityscapes to simulate an extremely dynamic real-world environment. We then compared the results of our DSOD method with those of the baseline DSO method. The trajectory and reconstruction result are shown in Fig. 6 (top

left and top right panels, respectively). The three images in the bottom row of Fig. 6 are actual road conditions, showing that this extremely dynamic scene consists of several potential moving objects (such as pedestrians, cars, bikes, and buses) and static buildings. The ground truth only gives GPS information, so the Mercator projection is used to calculate the relative coordinates, and the trajectory has some error due to the characteristics of GPS. It can be seen from the trajectory plot that DSOD performs better than DSO.

C. ABLATION STUDY

In order to verify the effectiveness of our proposed DSOD, we implemented two other algorithms based on DSO. We employed DSO as our baseline and compared these algorithms to evaluate the impact of the depth prediction network and semantic segmentation network on two image sequences from the Cityscapes dataset. Their trajectory results are presented in Fig. 7. Among them, the DSO+Depth method is DSO with the depth prediction network, the DSO+Segment method is DSO with the segmentation network. As shown in Fig. 7, DSO+Depth and DSO+Segment are better than DSO. Our DSOD method, which contains both the depth prediction network and segmentation network, provides the best results.

Meanwhile, to show the effect of our algorithm intuitively, the intermediate results for pose estimation are presented in Fig. 8, which shows the consequence of using DSOD to filter out dynamic points on the Frankfurt sequence. Figure 8 shows an example of a complex and extremely dynamic scene from the Cityscapes dataset, in which there are both moving and static objects. Here, the bus is moving and other cars are parked by the curb. In contrast to DSO, DSO+Depth can filter out the portion of points whose depth cannot be converged and provide a depth map. Moreover, DSO+Segment can filter out moving objects but provides no depth values, and the point density decreases. In summary, the monocular DSOD SLAM system can filter out the points of moving objects effectively and yields better performance in a dynamic environment than the baseline DSO method.

V. CONCLUSION

In this paper, we proposed a novel monocular VO algorithm called DSOD. In DSOD, we introduced a depth prediction network and a semantic segmentation network. The depth prediction network is used to obtain depth priors. The segmentation network is used to acquire the semantic priors of potential dynamic points. Here, the semantic label of a pixel is helpful for selecting static pixels to minimize the photometric errors and optimize the pose estimation. Then, the depth and semantic priors are combined with a movement consistency check to filter out real dynamic points from the dynamic scene. Finally, the remaining points are used to reconstruct a map of the unknown environment and estimate pose. Experiments on the KITTI and Cityscapes datasets demonstrate that DSOD significantly improves the scale, accuracy, and robustness of the SLAM results in dynamic environments.

REFERENCES

[1] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: 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] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.
[3] J. Engel, V. Koltun, and D. Cremers, "Direct Sparse Odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Apr. 2017.
[4] W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao, "Robust monocular SLAM in dynamic environments," in *Proc. Int. Symp. Mixed Augmented Reality*, 2013, pp. 209–218.
[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.
[6] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.
[7] M. Cordts, "The cityscapes dataset for semantic urban scene understanding," in *Proc. Comput. Vis. Pattern Recognit.*, 2016, pp. 3213–3223.
[8] R. Wang, M. Schworer, and D. Cremers, "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 3923–3931.
[9] X. Gao, R. Wang, N. Demmel, and D. Cremers, "LDSO: Direct sparse odometry with loop closure," in *Proc. Intell. Robots Syst.*, 2018, pp. 2198–2204.
[10] D. Filliat, "A visual bag of words method for interactive qualitative localization and mapping," in *Proc. Int. Conf. Robot. Autom.*, 2007, pp. 3921–3926.
[11] D. Galvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.
[12] Y. Kuznietsov, J. Stuckler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 2215–2223.
[13] N. Yang, R. Wang, J. Stuckler, and D. Cremers, "Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 835–852.
[14] 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.*, 2019, pp. 5218–5223.
[15] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. Int. Conf. Robot. Automat.*, 2014, pp. 15–22.
[16] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 6602–6611.
[17] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.
[18] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer Assisted Intervention*. 2015, pp. 234–241.
[19] S. Jegou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional DenseNets for semantic segmentation," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 1175–1183.
[20] N. Sunderhauf, T. Pham, Y. Latif, M. Milford, and I. Reid, "Meaningful maps with object-oriented semantic mapping," in *Proc. Intell. Robots Syst.*, 2017, pp. 5079–5085.
[21] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg, "Joint semantic segmentation and 3D reconstruction from monocular video," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 703–718.
[22] V. Vineet, "Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction," in *Proc. Int. Conf. Robot. Automat.*, 2015, pp. 75–82.
[23] S. Y. Bao, M. Bagra, Y. Chao, and S. Savarese, "Semantic structure from motion with points, regions, and objects," in *Proc. Comput. Vis. Pattern Recognit.*, 2012, pp. 2703–2710.
[24] C. Yu, "DS-SLAM: A semantic visual SLAM towards dynamic environments," in *Proc. Int. Robots Syst.*, 2018, pp. 1168–1174.

[25] N. Brasch, A. Bozic, J. Lallemand, and F. Tombari, "Semantic monocular SLAM for highly dynamic environments," in *Proc. Int. Robots Syst.*, 2018, pp. 393–400.
[26] E. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *Int. J. Robot. Res.*, vol. 30, no. 4, pp. 407–430, 2011.
[27] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.
[28] L. Von Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," in *Proc. Int. Conf. Robot. Automat.*, 2018, pp. 2510–2517.
[29] J. Klappstein, T. Vaudrey, C. Rabe, A. Wedel, and R. Klette, "Moving object segmentation using optical flow and depth information," in *Proc. Pacific-Rim Symp. Image Video Technol.*, 2009, pp. 611–623.
[30] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, nos. 1–3, pp. 185–203, Aug. 1981.
[31] P. F. Alcantarilla, J. J. Yebes, J. Almazan, and L. M. Bergasa, "On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments," in *Proc. Int. Conf. Robot. Automat.*, 2012, pp. 1290–1297.
[32] S. Park, T. Schops, and M. Pollefeys, "Illumination change robustness in direct visual SLAM," in *Proc. Int. Conf. Robot. Automat.*, 2017, pp. 4523–4530.
[33] T. Lin, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

PING MA received the B.E. degree from Northeastern University, Shenyang, China, in 2015. She is currently pursuing the Ph.D. degree with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, China. Her main research area is SLAM.

YUE BAI received the Ph.D. degree from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, China, in 2006, where he is currently a Research Fellow and a Supervisor of Ph.D. candidates. His main research areas are UAV technology and agricultural aviation remote sensing.

JUNAN ZHU received the B.E. degree from Northeastern University, Shenyang, China, in 2015. He is currently pursuing the Ph.D. degree with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, China. His research interest includes object detection, tracking, and recognition.

CHUNJUN WANG received the M.S. degree from Beijing Jiaotong University, Beijing, China, in 2008. His main research area is 3D reconstruction.

CHENG PENG received the Ph.D. degree from Jilin University, Changchun, China, in 2015, where she is currently an Assistant Researcher. Her main research areas are UAV control technology and target tracking.