LOCALIZATION AND MAPPING IN DYNAMIC ENVIRONMENT USING MOVING OBJECTS SEGMENTATION FOR AUTONOMOUS DRIVING

Bo WANG[1], Hong BAO[1]*, Cheng XU[1]

Simultaneous localization and mapping (SLAM) is a key technology for localization in autonomous vehicles. This method only applies to the static environment; it limits the application of dynamic urban environments that contain many vehicles and pedestrians. In this paper we propose a visual SLAM system for dynamic urban environments. It has the capabilities of dynamic objects segmentation and removal. Localization and mapping are implemented by ORB-SLAM2. The difference is that we have a special pre-processing of the input images. We use instance-aware semantic segmentation to detect the objects, such as cars. Then use sparse optical flow to classify moving objects and potentially moving objects, such as driving cars and parked cars. The moving objects are removed directly, and potentially moving objects are utilized in the pose estimation section, but their corresponding landmarks are deleted in the mapping section, which is useful for loop-closure detection and relocalization. We evaluate our system on the public KITTI and TUM dataset. The results demonstrate that our system can work in highly dynamic urban environments and outperforms the accuracy of the state-of-the-art visual SLAM system.

Keywords: SLAM; dynamic urban environments; objects segmentation; sparse optical flow

1. Introduction

Simultaneous localization and mapping (SLAM) have been a popular research area in computer vision and mobile robotics since the 1980s [1]. In an unknown environment, robots carrying sensors can utilize the SLAM technology to estimate their position and orientation and build the environmental map. Recently, with the rise of autonomous driving, unmanned logistics, as well as virtual and augmented reality, SLAM technology has been received more attention. There are many sensors that can be used for localization, such as: Light Detection and Ranging (LIDAR), Global Positioning System (GPS) and Inertial Measurement Unit (IMU). Among them, LIDAR is very expensive and difficult to popularize in a short time. GPS has the disadvantage of large error (around 10

[1] Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing, 100101, China, e-mail: wangbo161819@gmail.com; baohong@buu.edu.cn; xc-f4@163.com
* Corresponding author: Hong Bao

meters) and cannot meet the requirements of lane level localization. There is a problem of the accumulated error in the IMU, and long-term use will bring a large error. Camera has the advantage of low cost, portability and rich visual information, so visual SLAM has become popular. In particular, in this work we focus on monocular and stereo camera.

In order to achieve practical application, visual SLAM has many key issues to be solved. SLAM requires that the environment in which the robot is located is static, but in reality, there are many dynamic objects in the environment, such as moving vehicles and pedestrians. These dynamic objects reduce the localization accuracy and the quality of the map. Although the Random Sample Consensus (RANSAC) algorithm can eliminate mismatched feature points caused by dynamic objects, it fails when there are many dynamic objects and occupying a large scene.

Visual SLAM has many system pipelines, including feature-based methods and direct methods. Feature-based methods [2, 3, 4, 5] minimize the reprojection error, and can only generate a sparse map, which cannot be used for further obstacle avoidance and navigation. Direct methods [6, 7, 8] minimize the photometric error and build a semi-dense map for a better interactive experience. In recent years, SLAM systems incorporating cameras and IMUs [9, 10, 11] have achieved better accuracy robustness, it can solve the problem of scale uncertainty caused by monocular camera, weak texture scene localization and camera motion blur.

None of the above classic and mainstream methods address the dynamic objects like cars, pedestrians and bicycles. More strictly speaking, dynamic objects include moving objects and objects that are now stationary and will move in the future. Bescós et al. [12] remove these objects, regardless of moving or static, which reduces the localization accuracy of scenes with more static objects such as parked cars. Bârsan et al. [13] is the same as the above method, but the static object is removed from the map to improve the reusability of the map.

In this paper, we propose a method to deal with dynamic objects in visual SLAM. We have a special pre-processing of the input image captured by camera. We use a Convolutional Neural Network (CNN) to segment the dynamic objects in the images and then unify their pixel values to achieve the purpose of not extracting features on them. To distinguish moving and static objects, we track them and then use the sparse optical flow to classify them. For moving objects, remove them directly. After removing the moving objects, the images are input to the ORB-SLAM2 system for localization and mapping. In particular, static objects are used in the pose tracking phase to improve localization accuracy, but the corresponding map points are deleted when the map is built to improve the accuracy of the relocation. In the rest of the paper, we discuss related work in Section II, we describe our system in Section III, then present the evaluation

results in Section IV and end with conclusions in Section V.

2. Related work

Robust visual SLAM in dynamic environment generally performs motion segmentation prior to localization and 3D reconstruction. Using the motion segmentation method to detect moving objects in images, because localization and 3D reconstruction only utilize the image other than moving objects.

A. Background-Foreground Initialization.

Background-Foreground Initialization assumes that the system has prior knowledge about the environment and leverages that information to segment static and dynamic features [14]. Zhang et al. [15] employ 3-D motion segmentation method to segment the feature point trajectories into different motions. Lee et al. [16] detect humans from recorded video frames of a moving camera and tracks the humans in the V-SLAM-inferred 3-D space via a tracking-by-detection scheme. Babaee et al. [17] use CNN to perform the segmentation estimate background model from video.

B. Geometric Constrains.

Standard visual SLAM systems use outlier rejection algorithm by geometric models, such as by RANSAC [18]. Zou et al. [19] leverage the reprojection error, they project features from the previous frame into the current frame and measure the distance from the tracked features. If the distance exceeds a certain threshold, the feature point is treated as an outlier.

C. Optical Flow and Scene Flow

These literatures [20, 21] compute optical flow of dense stereo to detect moving objects. Alcantarilla et al. [22] segment moving objects by means of a dense scene flow (3D version of optical flow) representation.

D. Deep learning.

Based on deep learning, Lin and Wang [23] accomplish motion segmentation the images taken by a moving stereo camera. Fully Convolutional Network [24] is used to take input of arbitrary size and produce correspondingly-sized output with efficient inference and learning. Bescós et al. [12] uses Mask-RCNN and multi-view geometry to segment moving objects, and inpaint the frame background that has been occluded by such objects. Bârsan et al. [13] uses both instance-aware semantic segmentation and sparse scene flow to classify objects as either background, moving or potentially moving.

All of the above methods can only detect moving objects. For temporarily static objects, they cannot appear in the map, but can be used during pose tracking. Bescós et al. [12] treats all detected objects as moving objects, this results in a decrease in the accuracy of scene localization in which a plurality of temporarily static objects are located. Bârsan et al. [13] classifies dynamic objects

and temporarily static objects, and does not process the pose tracking, but reconstructs the dynamic objects and the temporarily static objects separately in the 3D reconstruction. The proposed method uses the instance-aware semantic segmentation and sparse optical flow to simultaneously detect moving objects and temporary static objects and uses the features of temporary static objects in the position tracking but deletes the corresponding map points when the 3D reconstruction.

3. System description

Fig. 1 show an overview of our proposed system. Input the raw image into the Mask R-CNN [25] to generate an image with a dynamic object mask. This step does not distinguish whether the dynamic object is moving or movable. The optical flow [26] of the multi-frame segmented image is calculated, and the moving object and the movable object are classified. The image after removing moving object is input to a state-of-the-art SLAM system to obtain a camera's motion trajectory and a sparse map with delete movable objects of the environment.

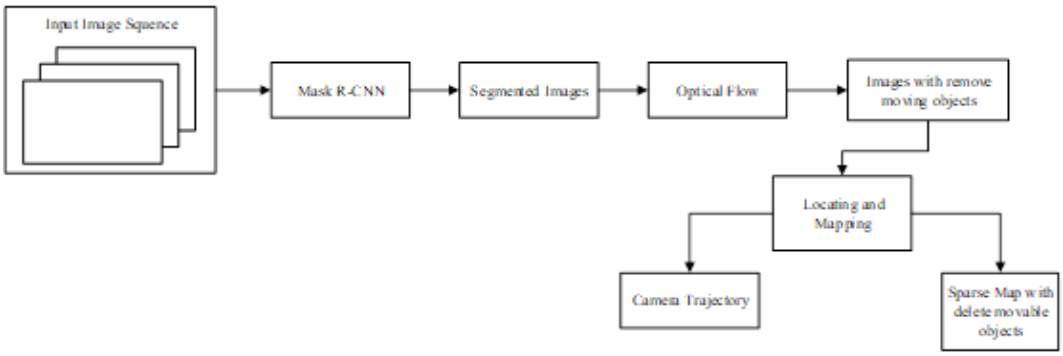


Fig. 1. An overview of system pipeline. The input image sequence is taken by a monocular or binocular camera.

Next, we will describe the details of each of the main modules.

A. Object segmentation and tracking

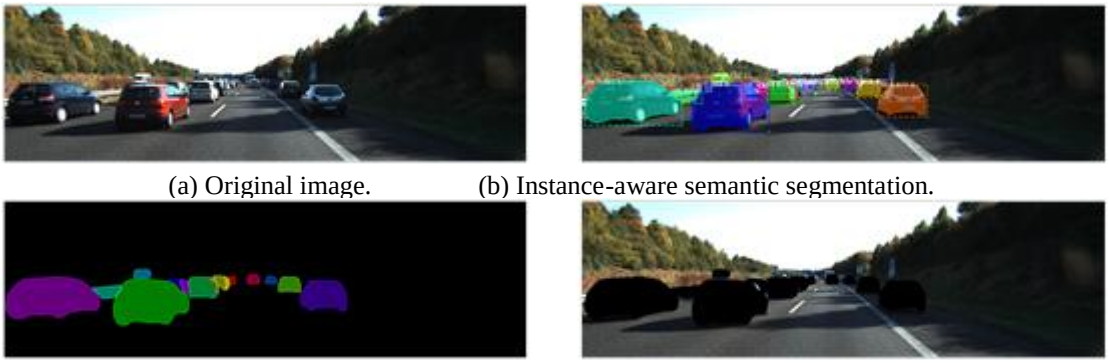
In order to distinguish the dynamic part and the static part of the image more finely, we use pixel-wise semantic segmentation to segment the dynamic objects from the image. In our experiments we use Mask R-CNN, a state-of-the-art deep neural network architecture for pixel-wise semantic segmentation. The input of Mask R-CNN is the original RGB image, and the output is segmentation masks for each instance of an object in the image (Fig. 2). We use pre-trained weight for MS COCO to segment objects, mainly cars, pedestrians and buses.

In order to determine whether dynamic objects are moving or temporarily static, we need to track them across multiple frames. Instance labels are useful for

tracking different objects and inspired by [13]. We associate new detection with existing tracks by ranking them based on the Intersection-over-Union (IoU) score. The same or similar IoU scores are detected as the same object, otherwise it is a new object. IoU is simply an evaluation metric. It is defined as

$$IoU = \frac{A_o}{A_u} \tag{1}$$

A_o is the area of overlap between the inter-frame bounding box. A_u is the area of union between the inter-frame bounding box.



(a) Original image. (b) Instance-aware semantic segmentation.

(c) Masks. (d) Removed masks.

Fig. 2. Picture (a) is an input RGB image. It can be seen that there are many cars driving on the road. Picture (b) is the output of the instance-aware semantic segmentation. Picture (c) is the masks of the segmented instances. Picture (d) is the result of removing the masks from the original image.

B. Sparse optical flow

Optical flow describes the motion of objects between frames. The optical flow method infers the moving speed and direction of the object by detecting the change of the intensity of the pixel of the image with time. In order to reduce the computational complexity, we use the sparse optical flow to calculate only the optical flow of the representative pixel points in the segment and the static part of the image. The car is rigid, and the speed and direction of the pixels are the same, so we can select a small number of pixels with obvious features on the segmented objects. By comparing the velocity and direction of the segmented objects and the static portion of the pixels, we can determine whether the segmented objects are motion or static.

For a 2D+t dimensional case a pixel at location (x, y, t) with intensity $I(x, y, t)$ will have moved by $\Delta x \Delta x$, Δy, and Δt Δt between the two image frames, and the following brightness constancy constraint can be given:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{2}$$

Assuming the movement to be small, the image constraint at $I(x, y, t)$ with Taylor series can be developed to get:

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x,y,t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t \quad (3)$$

From these equations it follows that:

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} = 0 \quad (4)$$

where V_x, V_y V_x, V_y are the x and y components of the velocity or optical flow of $I(x,y,t)$ and $\frac{\partial I}{\partial x}\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}\frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}\frac{\partial I}{\partial t}$ are the derivatives of the image at (x,y,t) in the corresponding directions.

Fig. 3 shows the optical flow of an image. First, extract some key points on the image and then track them in the next sequence, which represents the motion trajectory of the key points. The static object moves in the opposite direction to the camera, and the key points on the car have no trajectory, indicating that the car is moving in the same direction as the camera. The core of the optical flow method used here is to judge whether the object moves by comparing the moving direction and speed of the key points.

Fig. 3. Sparse optical flow. The dot represents a key point, and the line segment represents a motion trajectory of a key point.

C. Handling moving and movable objects

By the object segmentation and optical flow method, we determine whether the segmented objects are moving or static and mark them. For moving objects, we remove them directly from the image. For objects that are now static and likely to move in the future, we use them in pose estimation, but delete their corresponding map points when building the map. Deleting objects (such as cars parked on the side of the road) that should not exist on the map can help improve the accuracy of relocation when reusing the map. In the long-term SLAM, when we return to a certain position, the previously stationary object has now left, and we can use the marker information of the stationary object to improve the success rate of place recognition.

With the optical flow method, we judge that the instances in Fig. 2(c) are moving, so they are removed from the original input image. On the contrary, the instances in Fig. 4(c) are static. We don't do the removal process, just save their semantic information.

(a) Original image. (b) Instance-aware semantic segmentation.

(c) Masks. (d) Do not remove the masks.

Fig. 4: Picture (a) is an input RGB image, it can be seen that there are many cars parked on the road. Picture (b) is the output of the instance-aware semantic segmentation. Picture (c) is the masks of the segmented instances. Picture (d) is the original image without remove the masks.

D. Localization and mapping

Visual SLAM is mainly divided into feature-based methods and direct methods. Since the direct methods are greatly influenced by illumination variations and speed, a more robust feature-based method is used in our experiments. In some scenes with more dynamic objects, after removing the moving objects, fewer feature points are available, and the pose tracking fails. In order to improve the robustness of the system in low-textured environment, in addition to extracting the point features in the image, we extract the line features at the same time (Fig. 5).

(a) (b)

Fig. 5. Extract point features and line features of the image. Picture (a) improves the accuracy and robustness of the system by extracting more features. Picture (b) compensates for the adverse effects caused by the reduction of point features by extracting line features.

We input the images of the removed dynamic objects and with the stationary object markers into the state of the art ORB-SLAM2 [18] and PL-SLAM [27], and then verify the effectiveness of our system by comparing the localization accuracy with the original system on the KITTI and TUM dataset.

4. Experiments

In order to evaluate the performance of the system, we have tested our system on the well-known autonomous driving dataset KITTI Vison Benchmark Suite [28] and compared to other state-of-the-art SLAM systems.

All the experiments have been run on an Intel Core I7-6800K CPU @ 3.40GHz and 16GB RAM with an NVIDIA GeForce 1080Ti GPU. This hardware configuration is similar to that used in state-of-the-art papers, so we use the results published in the paper directly for comparison. Due to the randomness brought by the system, we have run each sequence over 10 times and show always median results.

The sparse map we generated using feature-based method has no obvious structural features so that cannot be used to qualitatively and quantitatively evaluate the effect of removing dynamic objects. Therefore, the localization accuracy is mainly discussed in this paper.

A. KITTI dataset

The KITTI odometry benchmark consists of 11 sequences (00-10) with ground truth trajectories collected by the camera-loaded car in urban and highway environments. Sequence 10 has a lot of cars on the road. Sequence 00 has a lot of cars parked on the roadside. After processing these cars, we can verify the effectiveness and advantages of our system. Table I shows a comparison between our results and state-of-the-art ORB-SLAM2 in the eleven sequences.

Table I

COMPARISON OF ACCURACY IN THE KITTI DATASET

Sequence	ORB-SLAM2			Our method		
	t_{rel} (%)	R_{rel} (deg/100m)	t_{abs} (m)	t_{rel} (%)	R_{rel} (deg/100m)	t_{abs} (m)
00	0.70	0.25	1.3	**0.65**	**0.24**	1.3
01	1.39	0.21	10.4	**1.35**	**0.20**	**9.8**
02	**0.76**	**0.23**	**5.7**	0.81	0.26	6.1
03	**0.71**	0.18	0.6	0.75	0.18	0.6
04	0.48	0.13	0.2	**0.45**	**0.11**	0.2
05	0.40	0.16	0.8	**0.39**	0.16	0.8
06	**0.51**	**0.15**	0.8	0.52	0.20	0.8
07	0.50	0.28	0.5	**0.48**	0.28	0.5
08	1.05	0.32	3.6	1.05	**0.29**	**3.4**
09	**0.87**	**0.27**	3.2	0.91	0.28	2.5
10	**0.60**	**0.27**	**1.0**	0.65	0.29	1.1

We use two different metrics, the absolute translation RMSE t_{abs} proposed in [29], and the average relative translation t_{rel} and rotation r_{rel} errors proposed in [30].

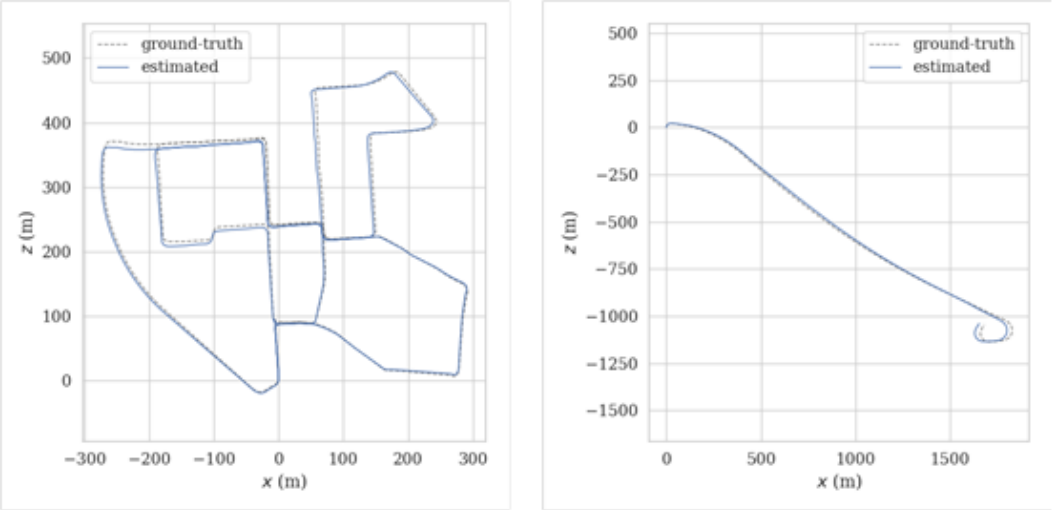


Fig. 6. Two examples of estimated trajectories with our system (in solid line) and ground-truth in dotted line.

Our system outperforms ORB-SLAM2 in most sequences. In particular, for the sequences KITTI 01 and KITTI 04 with more dynamic objects, our system accuracy is significantly improved. The key is that we remove the dynamic object and no longer extract the feature points on it. These wrong feature matching will reduce the accuracy of the tracking. For sequences with fewer dynamic objects, our system accuracy is not stable enough, and the accuracy of some sequences (KITTI 09, KITTI 10) is reduced. This is because we use the optical flow method to judge whether the object is moving, and the optical flow method has the assumption of small motion. However, the speed of the vehicle using the KITTI data set is faster, which may reduce the accuracy of the optical flow method. If a large number of static objects are removed as judged as dynamic objects, the available feature points are reduced, and the accuracy of the system is reduced. However, we obtain the semantic information of the image through segmentation, which can provide effective constraints in loop detection and long-term application.

Fig. 6 shows three examples of estimated trajectories in KITTI dataset.

B. TUM RGB-D dataset

The TUM RGB-D dataset [19] containing RGB-D data and ground-truth data and is an important dataset for evaluating camera localization accuracy. In particular, in some sequences of dynamic objects, some people move in small areas in front of the camera or walk in a wide range, which can effectively test the performance of our system. We use the absolute trajectory RMSE [19] as the error metric of our experiments. Table II shows the median results over 10 executions in each of the 5 sequences selected.

Table II

COMPARISON OF ACCURACY IN THE TUM RGB-D DATASET

	Absolute KeyFrame Trajectory RMSE	
Sequence	ORB-SLAM2	Our method
fr3_sit_xyz	0.79 cm	0.75 cm
fr3_sit_halfsph	1.34 cm	1.28 cm
fr2_desk_person	0.63 cm	0.60 cm
fr3_walk_xyz	1.24 cm	1.08 cm
fr3_walk_halfsph	1.74 cm	1.59 cm

Compared to the KITTI dataset, the accuracy of our system is generally higher than that of ORB-SLAM2 on the TUM dataset. This is because the TUM dataset is collected by a human hand-held camera, and the relative displacement between frames is small, making the optical flow method more reliable. Moreover, our system is more advantageous when there are highly dynamic objects in the scene.

5. Conclusions

We have presented a robust visual SLAM system for dynamic urban environments. Our system uses a deep convolutional neural network for instance-aware semantic segmentation to segment dynamic and potentially dynamic objects from the original image. Sparse optical flow is used to identify the dynamic and potentially dynamic objects. Localization and mapping is building on the ORB-SLAM2. The difference is that the input image removes the moving objects. Experiments on the KITTI and TUM dataset show that our system accuracy is better than ORB-SLAM2, especially in scenes with more dynamic objects, so that our system can be applied to large-scale urban dynamic environments. The original ORB-SLAM2 is running in real time, and because the semantic segmentation takes more computation time, the whole system cannot meet the real-time requirements of the automatic driving. Moreover, the optical flow method is prone to failure when the motion between frames is large. Therefore, future work is to speed up the calculation of the semantic segmentation module and use the scene flow [31].

Acknowledgements

This work was supported, the Supporting Plan for Cultivating High Level Teachers in Colleges and Universities in Beijing (IDHT20170511), the Talents Cultivation and Cooperation Oriented to Intelligent Vehicle Industrialization(Grant No. UK-CIAPP\324), Newton Fund Project supported by Royal Academy of Engineering of UK, Graduate Fund Project supported by Beijing Union University.

REFERENCES

[1] *C. Cadena, L. Carlone, H. Carrillo, et al,* Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age[J], IEEE Transactions on Robotics, **vol. 32**, no. 6, 2016, pp. 1309-1332.

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

[3] *R. Gomez-Ojeda, D. Zuñiga-Noël, F. A. Moreno, et al.* PL-SLAM: a Stereo SLAM System through the Combination of Points and Line Segments[J], 2017.

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

[5] *F. Endres, J. Hess, J. Sturm, et al.* 3-D mapping with an RGB-D camera[J]. IEEE Transactions on Robotics, **vol. 30**, no. 1, 2014, pp. 177-187.

[6] *J. Engel, T. Schöps, D. Cremers,* LSD-SLAM: Large-scale direct monocular SLAM[C]//European Conference on Computer Vision. Springer, Cham, 2014, pp. 834-849.

[7] *J. Engel, V. Koltun, D. Cremers,* Direct sparse odometry[J]. IEEE transactions on pattern analysis and machine intelligence, **vol. 40**, no. 3, 2018, pp. 611-625.

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

[9] *T. Qin, P. Li, S. Shen, Vins-mono:* A robust and versatile monocular visual-inertial state estimator[J]. IEEE Transactions on Robotics, **vol. 34**, no. 4, 2018, pp. 1004-1020.

[10] *T. Schneider, M. Dymczyk, M. Fehr, et al.* maplab: An open framework for research in visual-inertial mapping and localization[J]. IEEE Robotics and Automation Letters, **vol. 3**, no. 3, 2018, pp. 1418-1425.

[11] *S. Leutenegger, S. Lynen, M. Bosse, et al.* Keyframe-based visual–inertial odometry using nonlinear optimization[J]. The International Journal of Robotics Research, **vol. 34**, no. 3, 2015, pp. 314-334.

[12] *B. Bescós, J. M. Fácil, J. Civera, et al.* DynSLAM: Tracking, Mapping and Inpainting in Dynamic Scenes[J]. arXiv preprint arXiv:1806.05620, 2018.

[13] *I. A. Bârsan, P. Liu, M. Pollefeys, et al.* Robust Dense Mapping for Large-Scale Dynamic Environments[C]//2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 7510-7517.

[14] *M. R. U. Saputra, A. Markham, N. Trigoni*, Visual SLAM and Structure from Motion in Dynamic Environments: A Survey[J]. ACM Computing Surveys (CSUR), **vol. 51**, no. 2, 2018, pp. 37.

[15] *D. Zhang, P. Li,* Visual odometry in dynamical scenes[J]. Sensors & Transducers, **vol. 147**, no. 12, 2012, pp. 78.

[16] *K. H. Lee, J. N. Hwang, G. Okopal, et al.* Ground-Moving-Platform-Based Human Tracking Using Visual SLAM and Constrained Multiple Kernels[J]. IEEE Transactions on Intelligent Transportation Systems, **vol. 17**, no. 12, 2016, pp. 3602-3612.

[17] *M. Babaee, D. T. Dinh, G. Rigoll,* A Deep Convolutional Neural Network for Background Subtraction[J]. Pattern Recognition, 2017.

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

[19] *D. Zou, P. Tan,* CoSLAM: Collaborative Visual SLAM in Dynamic Environments[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, **vol. 35**, no. 2, 2013, pp. 354-366.

[20] *M. Derome, A. Plyer, M. Sanfourche, et al.* Real-time mobile object detection using stereo[C]// International Conference on Control Automation Robotics & Vision. IEEE,

2014, pp. 1021-1026.
[21] *M. Derome, A. Plyer, M. Sanfourche, et al.* Moving object detection in real-time using stereo from a mobile platform[J]. Unmanned Systems, **vol. 3**, no. 4, 2015, pp. 253-266.
[22] *P. F. Alcantarilla, J. J. Yebes, J. Almazán, et al.* On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments[C]// IEEE International Conference on Robotics and Automation. IEEE, 2012, pp. 1290-1297.
[23] *T. H. Lin, C. C.Wang,* Deep learning of spatio-temporal features with geometric-based moving point detection for motion segmentation[C]//Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE, 2014, pp. 3058-3065.
[24] *J. Long, E. Shelhamer, T. Darrell,* Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, pp. 3431-3440.
[25] *K. He, G. Gkioxari, P. Dollar, et al.* Mask R-CNN.[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, no.99, 2017, pp. 1-1.
[26] *G. Zhang, H. Chanson,* Application of local optical flow methods to high-velocity free-surface flows: Validation and application to stepped chutes[J]. Experimental Thermal and Fluid Science, **vol. 90**, 2018, pp. 186-199.
[27] *R. Gomez-Ojeda, D. Zuñiga-Noël, F. A. Moreno, et al.* PL-SLAM: a Stereo SLAM System through the Combination of Points and Line Segments[J]. 2017.
[28] *A. Geiger, P. Lenz, R. Urtasun,* Are we ready for autonomous driving? the kitti vision benchmark suite[C]//Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012, pp. 3354-3361.
[29] *J. Sturm, N. Engelhard, F. Endres, et al.* A benchmark for the evaluation of RGB-D SLAM systems[C]// Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 2012, pp. 573-580.
[30] *A. Geiger, P. Lenz, C. Stiller, et al.* Vision meets robotics: The KITTI dataset[J]. International Journal of Robotics Research, **vol. 32**, no. 11, 2013, pp. 1231-1237.
[31] *M. Menze, C. Heipke, A. Geiger,* Object scene flow[J]. ISPRS Journal of Photogrammetry and Remote Sensing, **vol. 140**, 2018, pp. 60-76.