Robotica: (2020) volume 38, pp. 256–270. © Cambridge University Press 2019
doi:10.1017/S0263574719000584

Robust Semantic Mapping in Challenging Environments

Jiyu Cheng†, Yuxiang Sun‡ and Max Q.-H. Meng†*

†*Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong, China. E-mails: jycheng@ee.cuhk.edu.hk, max.meng@cuhk.edu.hk*
‡*Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, China. E-mails: yxsun@link.cuhk.edu.hk, sun.yuxiang@outlook.com*

(Accepted March 30, 2019. First published online: May 21, 2019)

SUMMARY
Visual simultaneous localization and mapping (visual SLAM) has been well developed in recent decades. To facilitate tasks such as path planning and exploration, traditional visual SLAM systems usually provide mobile robots with the geometric map, which overlooks the semantic information. To address this problem, inspired by the recent success of the deep neural network, we combine it with the visual SLAM system to conduct semantic mapping. Both the geometric and semantic information will be projected into the 3D space for generating a 3D semantic map. We also use an optical-flow-based method to deal with the moving objects such that our method is capable of working robustly in dynamic environments. We have performed our experiments in the public TUM dataset and our recorded office dataset. Experimental results demonstrate the feasibility and impressive performance of the proposed method.

KEYWORDS: Semantic Mapping; Dynamic Environments; CRF-RNN

1. Introduction

Visual simultaneous localization and mapping (visual SLAM) has been well applied in many robotic tasks in recent decades, and many state-of-the-art algorithms[1–5] have been proposed with rather satisfactory performance. One major capability of the visual SLAM system is to generate a map representing the surrounding environments. With the knowledge of the environments, the robots can implement some other tasks such as path planning[6] and exploration.[7] While the existing visual SLAM systems are able to generate a geometric map accurately, the semantic information[8] is usually overlooked. The lack of semantic information poses great challenges to some robotic applications such as exploration[9] and autonomous navigation.[10]

To address this problem, we propose a semantic mapping method to help the mobile robot generate a semantic map which contains both geometric and semantic information. Specifically, we start semantic mapping from a localization module. With accurate localization, the map can be reconstructed accurately based on the camera poses. To achieve good localization result, one big challenge is to deal with the moving objects in the dynamic environment. Although some algorithms like RANdom SAmple Consensus (RANSAC)[11] can filter out some dynamic factors, they cannot perform very well when the moving parts are nontrivial. In our method, we propose to use an optical-flow-based model to address the dynamic factors. In particular, inspired by the success of deep neural network recently,[12,13] we make use of it for the semantic generation. Finally, the semantic map is generated with the accurate camera pose estimation and semantics.

Our method enjoys several desirable properties which make it suitable for the robotic application. First, we take into account the importance of semantic information. With semantics, the robot is

* Corresponding author. E-mail: max.meng@cuhk.edu.hk

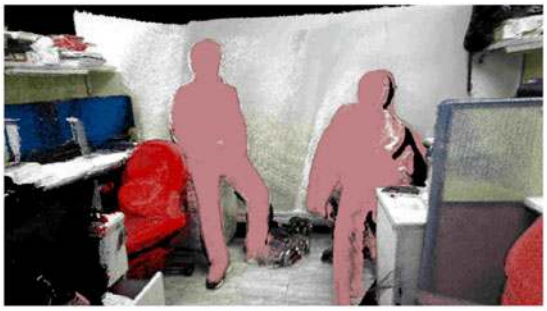

Fig. 1. An illustrative result of our proposed method. Intuitively, the regions of chairs, the persons and the monitors in the image are highlighted in red, pink and blue, respectively.

able to get a better knowledge of the surrounding environment which benefits implementing the task. For example, in ref.,[9] the robot needs to find a cup in an unknown environment. The cup is more likely to be located in a living room than a washroom. Based on this prior knowledge, the robot intends to find the cup in a living room, which improves the searching efficiency. Secondly, we consider the effect of moving objects. Although most of the visual SLAM systems assume that the environment is static, moving objects always exist in real-world scenes. Thirdly, the deep neural network ensures the robustness of semantic generation. The good generalization performance is able to deal with the deviation between different environments. Figure 1 shows an illustrative experimental result of our system. We adopt ORB-SLAM[14] as the basic mapping scheme. We integrate our optical-flow-based model to deal with dynamic factors, while we use CRF-RNN[15] to generate a pixel-wise labeled image. This paper is an extension of the method introduced in ref.[16] Different from ref.,[16] we incorporate the uncertainty of the localization due to the motion of the objects.

The contributions of our work are summarized as follows:

1. We have proposed a novel approach to combine the visual SLAM with the semantic segmentation to generate a semantic 3D map.
2. We have proposed an optical-flow-based method to deal with the dynamic factors, which ensures the localization accuracy.
3. We have tested the proposed method in the public TUM dataset and our recorded office dataset. Experimental results have demonstrated the feasibility of our method.

The rest of this paper is structured as follows. Section 2 describes the related work. Section 3 presents the details of the proposed approach. Section 4 analyzes some experimental results. In Section 5, the final conclusions are drawn while the future work is outlined.

2. Related Work

In this section, we will briefly review the works of addressing dynamic factors in visual SLAM and semantic mapping.

2.1. Addressing dynamic factors

To address the dynamic factors in visual SLAM, the existing algorithms can be divided into three categories.

Information fusion-based. Bloesch et al.[17] combined complementary information from vision and inertial sensors to enable robust performance for high-dynamic scenarios. Usenko et al.[18] used the inertial measurement unit (IMU) as an additional sensor. They used an energy function to combine photometric and inertial information. By minimizing the energy function, camera pose, velocity and IMU bias are jointly estimated. Kim et al.[19] used a Kinect-style RGB-D sensor[20] and IMU to accurately estimate camera trajectory in highly dynamic environments. The information fusion-based method usually relies on additional sensors like the IMU. The information from these sensors can compensate for the error caused by a single sensor and improve the localization accuracy. However, the fusion of multisource information is time-consuming. What's more, how to efficiently combine different kinds of information is still a challenge.

https://doi.org/10.1017/S0263574719000584 Published online by Cambridge University Press

Depth information-based. Kim et al.[21] proposed a robust background model-based dense-visual odometry algorithm that can deal with dynamic factors. Sun et al.[22] used a differencing frame to denote the moving regions based on two consecutive frames. Vector quantization-based segmentation is adopted to segment dynamic objects out. Li et al.[23] used weighted edge points to conduct visual odometry based on frame-to-keyframe registration. The point with a high static weight is adopted for further visual odometry. In this way, dynamic factors are eliminated effectively and the method can work in real time. Sun et al.[24] extended ref.[22] with an incremental learning capability, allowing updating the foreground model incrementally. The depth information can be used to segment the moving objects or detect the edge points. Researchers usually use it to choose the reliable region of the image for further visual localization. However, one limitation for these kinds of methods locates in that the depth sensors cannot be used in an outdoor or a large-scale environment. What's more, there are nonnegligible errors for these kinds of sensors.

Purely vision-based. Zou et al.[25] proposed a multi-camera scheme. The cameras work cooperatively to sense the dynamic factors and conduct accurate localization. Wang et al.[26] grouped the neighboring feature points sharing the same scene flow. The feature points from the largest group are adopted as the static feature points and used for visual odometry. Terashima et al.[27] used the CG model to obtain the difference between two consecutive frames and distinguish the dynamic feature points. Cheng et al.[28] use the compensation to make two frames that share the same view. The dynamic parts are indicated by the difference between the two frames. Purely vision-based methods are applicable on most of the platforms and for most of the scenarios. They are cheap and easily calibrated. However, compared with the first two strategies, it provides less information which sometimes leads to ambiguities of the detection of dynamic factors.

There are some other methods such as the control strategy, shown in ref.[29] Applying the traditional visual SLAM in an active way can help the mobile robot to avoid entering the dynamic regions.

To make our system suitable for large-scale environments and general platforms, we choose the third kind of methods. Different from the existing works, we propose to directly compute the transformation between two consecutive frames. Based on the transformation, we convert the two frames to share with the same coordinates and detect the dynamic feature points based on the optical flow values of the corresponding feature points, which efficiently reduces the ambiguities.

2.2. Semantic Mapping

Semantic mapping has been developed in recent years and many the-state-of-the-art algorithms are proposed with very satisfactory results. Hermans et al.[30] used 2D semantic segmentations to generate 3D semantic representations of the environments. They also showed that not all the frames are needed for semantic segmentation. The work by Salas-Moreno[31] known as *SLAM++* introduced an "object-oriented" visual SLAM framework. The object information helps to predict camera poses generated from accurate IGP. The proposed framework can generate an object-level representation of the environment. Sunderhauf et al.[32] built a map of the environment containing both semantically meaningful object-level entities and point- or mesh-based geometrical representations. The object models are built on the fly and do not require prior known 3D models. Bowman et al.[33] connected the data association and recognition to formulate an optimization problem that integrates the metric information, the semantic information and the data association. Gan et al.[34] proposed a dense 3D semantic mapping algorithm using a sparse Bayesian model, the relevance vector machine. They formulate the problem as a high-dimensional multi-class classification. They solve the problem sequentially in a fully probabilistic framework. Different from the above-mentioned methods, the main idea of our method is to first ensure the localization accuracy of the camera that is capable of dealing with the dynamic factors. With the accurate localization, we reconstruct the environment using a point cloud map. Then the semantic information is registered into the 3D map. Finally, we can obtain a dense semantic map of the environment.

3. Method

An overview of our proposed approach is shown in Fig. 2. There are three modules in our approach. The first module called *ORB-SLAM-based mapping* is to generate a 3D point cloud map of the environment based on the localization result of the ORB-SLAM system. The second module called *CRF-RNN-based semantic segmentation* is to generate the pixel-wise labels for the corresponding

https://doi.org/10.1017/S0263574719000584 Published online by Cambridge University Press

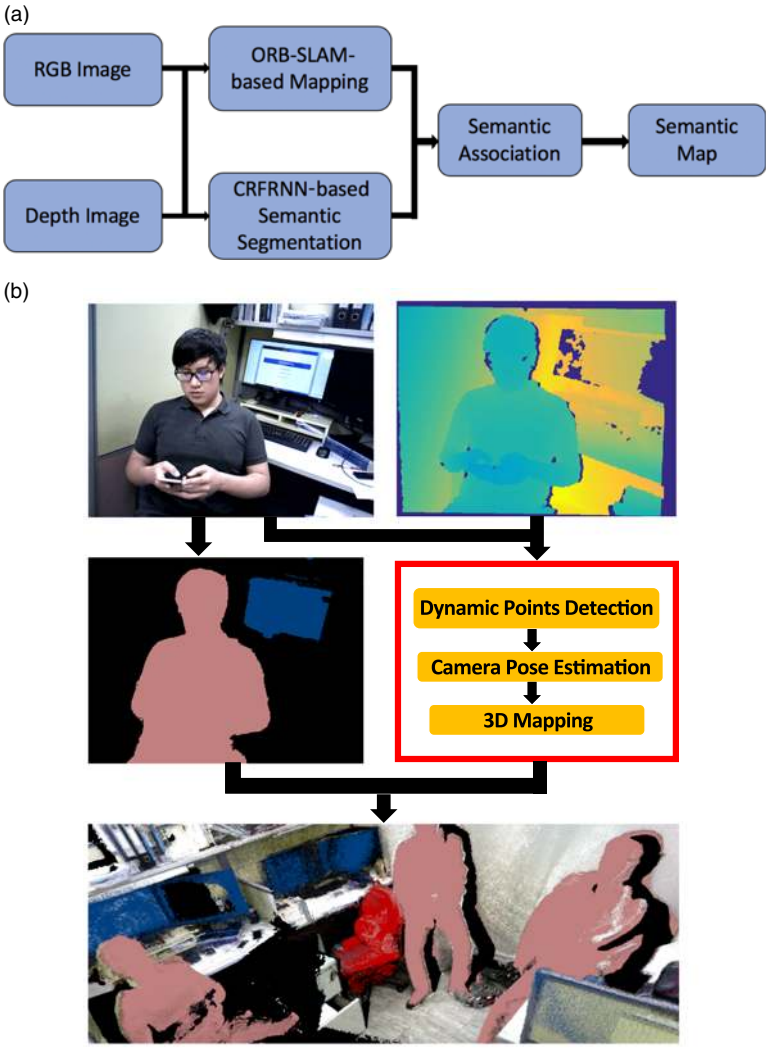

Fig. 2. The overview of our approach.

2D image. The third module called *Semantic Association* is to intelligently combine the results of the first two modules to generate a 3D semantic point cloud map. Finally, the 3D semantic point cloud map consists of both the geometric and the semantic information of the environments.

3.1. Robust Camera Localization

In our method, the accuracy of the camera pose estimation is closely correlated with the performance of the 3D mapping. In ORB-SLAM, the localization of the camera relies on a feature map. Camera poses are initialized first and then optimized by bundle adjustment (BA).[35] The optimization problem to be solved in BA can be formulated as follows:

$$\arg\min_{a_j,\, b_i} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} d(X_{ij} - Q(a_j, b_i))^2, \tag{1}$$

where we assume that n 3D points can be observed in m views and X_{ij} is projection of point i in image j. If point i is visible in image i, w_{ij} is 1 otherwise w_{ij} is 0. $Q(a_i, b_j)$ is predicted projection of point i in image j. $d(x, y)$ denotes the Euclidean distance between the image points represented by vectors x and y.

The existing visual SLAM systems assume that the environment is static. However, in many cases, the real-world environments usually contain dynamic objects, particularly humans. Dynamic objects will corrupt the localization accuracy and thus degrade the performance of visual SLAM. To solve

https://doi.org/10.1017/S0263574719000584 Published online by Cambridge University Press

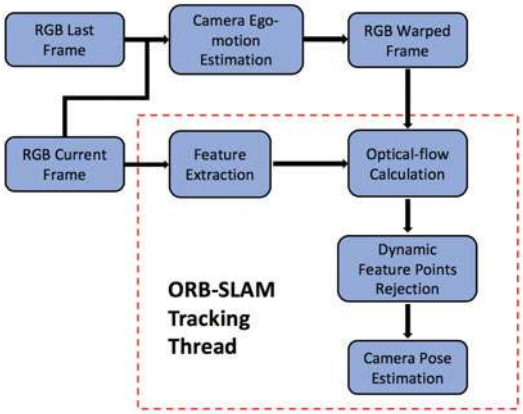

Fig. 3. The pipeline of the dynamic feature points elimination method.

this problem, we propose a novel method to eliminate the effects of dynamic objects. As shown in Fig. 3, we firstly use Five-Point Algorithm[36] to estimate the camera motion between two consecutive frames by computing the essential matrix T. Then we project the last frame onto the current frame to obtain the warped frame. The current frame and the warped frame are both delivered into the ORB-SLAM system. Once the current frame is derived, the system extracts the feature points that may include dynamic and static feature points. We use Lucas–Kanade[37] algorithm to calculate the optical flow of the matched feature points between the warped image and the current image while detecting the moving feature points for the current frame based on the optical flow values. A predefined tolerance τ is used to determine whether each point is dynamic or static using the following inequalities:

$$\begin{cases} d > \tau, \text{ if } f_i \in F_{\text{dynamic}}, \\ d < \tau, \text{ if } f_i \in F_{\text{static}}, \end{cases} \tag{2}$$

where $d = \sqrt{d_x^2 + d_y^2}$ is the unit length of the flow vector for feature point f_i, while F_{dynamic} and F_{static} are dynamic and static feature points sets, respectively.

After producing the detection of the dynamic feature points, we eliminate them and maintain the static feature points as the feature points for the current frame. Subsequently, the static feature points will be used for camera pose estimation in the ORB-SLAM tracking thread. Finally, we collect the optimized keyframes as one part of the input of the 3D semantic mapping.

3.2. CRF-RNN based Semantic Segmentation

In the fully connected pairwise CRF model, the energy of a label assignment x is given by

$$E(x) = \sum_i \psi_u(x_i) + \sum_{i \neq j} \psi_p(x_i, x_j), \tag{3}$$

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{M} \omega^{(m)} k_G^{(m)}(f_i, f_j), \tag{4}$$

where the unary energy component $\psi_u(x_i)$ measures the inverse likelihood of the pixel i taking the label x_i , and the pairwise energy component $\psi_p(x_i, x_j)$ measures the cost of assigning labels x_i, x_j according to pixels i,j. Each $k_G^{(m)}$ for $m = 1, ..., M$ is a Gaussian kernel applied to feature vectors. Feature vector f_i of pixel i is derived from image features. The function $\mu(.,.)$ captures the compatibility between different pairs of labels as the name implies.

By minimizing the energy $E(x)$, we obtain the most probable label assignment x for the given image. The resulting images of semantic segmentation will be one part of the inputs for the semantic mapping system. Note that we only conduct the semantic segmentation for keyframes of the ORB-SLAM system.

https://doi.org/10.1017/S0263574719000584 Published online by Cambridge University Press

3.3. Semantic Mapping

From the above two modules, we generate the optimized keyframes and 2D labeled images for the corresponding keyframes. In this section, we will show how to produce a 3D semantic map from these two parts.

Once a keyframe is determined, the semantic segmentation module gives each pixel a label and the pixel-wise labeled image is used for generating a 3D semantic map. In this process, due to the existence of the moving objects, we will remove the dynamic objects when building the 3D semantic map. Here we take two steps in practice. At the first step, we will check the number of the dynamic feature points on the segmented objects. If the number is bigger than Num, we regard the object as a dynamic one and we will not incorporate it into the semantic map. At the second step, we deal with the rest of the dynamic feature points. We first compute the depth difference between the pixel and the dynamic feature points. If the difference is under a threshold, we then compute the image distance between them. Only if the distance is small enough, we regard the point as a dynamic one and will not include it into our semantic map. For each pixel in a keyframe, it contains semantic and geometric information, and whether it can be used to generate a 3D point is determined based on Algorithm 1.

Algorithm 1: Generation of dynamic and static regions of the image

Input: RGB image I of size $640 * 480$, depth image D of size $640 * 480$, dynamic feature points set S for I, threshold σ_1, threshold σ_2, threshold Num

Output: Dynamic points set M and static points set N

```
Input an RGB image I;
Use CRF-RNN to generate the pixel-wise labeled image;
Check the number of the dynamic feature points on the segmented objects;
for each segmented object do
    if num>Num then
        Add all the points of the object into M
for Each point p of the rest of the points in S do
    if depth distance between point P in the image and p is less than σ1 then
        if image distance between P and p is less than σ2 then
            Add P into M
Add the rest of the points in the image into N;
final ;
return P;
```

Finally, we obtain the dynamic and static regions of the keyframe. We generate a 3D map by reprojection of the static regions of the keyframe. The problem can be formulated as follows:

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{5}$$

where f_x, f_y, u_0 and v_0 are camera intrinsics. R and t are camera extrinsics. (u, v) represents pixel coordinates, and (X, Y, Z) represents world coordinates. Z_c is the scale factor.

Thus, the resulting 3D map will contain both geometric and semantic information.

3.4. Model Comparison

To address the dynamic factors for visual SLAM, different strategies are proposed based on the intrinsic characteristics of the visual SLAM system. For direct visual SLAM systems, motion segmentation[24] is always utilized to eliminate the negative information. Since the depth information is needed in this process, a depth sensor like Kinect needs to be mounted on the mobile platform. For semi-direct visual SLAM systems, robust batches or edges are extracted[23] and used for visual odometry. The extraction process is always based on the depth information. As a result, a depth sensor is also needed. For indirect visual SLAM systems, like our method, the feature points are extracted and

https://doi.org/10.1017/S0263574719000584 Published online by Cambridge University Press

Table I. Descriptions of the selected sequences used for our experiments.

Sequence	Duration	Frames	Description
fr3/w/half	35.81 s	1067	Two persons walk through an office scene. The camera has been moved on a small half sphere of approximately one meter diameter.
fr3/w/rpy	30.61 s	910	Two persons walk through an office scene. The camera has been rotated along the principal axes (roll-pitch-yaw) at the same position.
fr3/w/xyz	28.83 s	862	Two persons walk through an office scene. The camera has manually been moved along three directions (xyz) while keeping the same orientation.
cuhk_office	32.15 s	985	Four persons sit at a desk, talk and gesticulate a little bit. The camera has randomly been moved while finishing a loop.

divided into dynamic and static feature points. The former ones are eliminated, while the latter ones are used for further visual odometry. By comparing the aforementioned strategies, the incorporation of the depth information can improve the localization accuracy significantly. One limitation for them is that the depth sensors generally do not perform well in the outdoor environments. Different from the above-mentioned strategies, our scheme can still work well when the depth information is not available. However, our approach suffers the degraded localization accuracy due to the limited availability of the surrounding information. What's more, we do not compare with the state-of-the-art algorithms, since the datasets or codes are not publicly available. As a result, it is not involved in our comparative study, which is a major limitation of our work. In our future work, we will design a more accurate model to achieve the performance comparable to those methods incorporating depth information. Furthermore, in terms of the semantic mapping task, with the available depth information, we will take the depth information into account to improve the performance of our system.

4. Experiments

4.1. Dataset

We use the public TUM dataset[38] and an office dataset to evaluate the proposed method. For the TUM dataset, each sequence comprises both RGB and depth images of 640×480 size. The dataset provides the information including the camera pose ground truth and the frequency. Once an estimated camera trajectory is generated, evaluations can be conducted based on the ground truth. For brevity, we use the words *fr*, *half*, *w*, *s*, *d*, *v* to denote *freiburg*, *halfsphere*, *walking*, *sitting*, *desk*, *validation* corresponding to different sequences. We list the details of some representative sequences in Table I.

4.2. Experiment setup

In our experiments, we use a computer with an i7 CPU, 32 GB memories, a GTX 1070 GPU. For each sequence used in our experiments, we preprocess it using CRF-RNN and obtain the pixel-wise labeled images. We empirically set the tolerance τ as 7 pixels, which implies that the corresponding feature point is regarded as a dynamic feature point if the norm of the optical flow vector is larger than 7 pixels. We empirically set the parameters in Algorithm1 Num as 20, σ_1 as $0.1m$ and σ_2 as 5 pixels.

4.3. Results of dynamic elimination

Figure 4 shows an illustrative experimental result of dynamic feature points detection. Figure 4(a) represents the current RGB image. Figure 4(b) is the result of dynamic feature points detection. Red dots represent the static feature points, while green "+" symbols represent the dynamic feature points. It can be shown that the proposed method can effectively distinguish the dynamic feature points, whereas several misclassification results are also produced resulting from the optical flow calculated based on two matched feature points. In addition, the blur in the image generated from the camera motion and moving objects tends to result in incorrect matches, and thus causes misclassifications in our method. Without loss of generality, the misclassifications can be divided into two cases. On one hand, some static feature points are misclassified as dynamic ones. In our scenario, the dynamic feature points will be removed and we only adopt the static feature points in our system. As a result, this kind of misclassification only slightly reduces the number of the feature points we use for the

https://doi.org/10.1017/S0263574719000584 Published online by Cambridge University Press

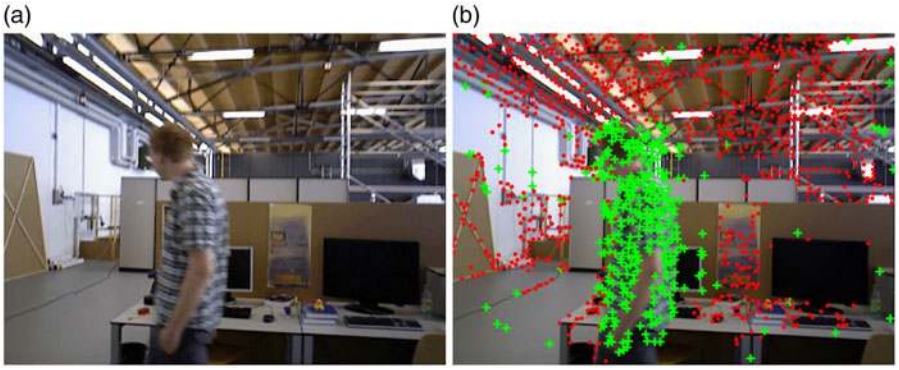

Fig. 4. An experimental result of dynamic feature points detection: (a) the current RGB image and (b) the result of dynamic feature points detection. Red dots represent the static feature points, while green "+" symbols represent the dynamic feature points. The figure is best viewed in color.

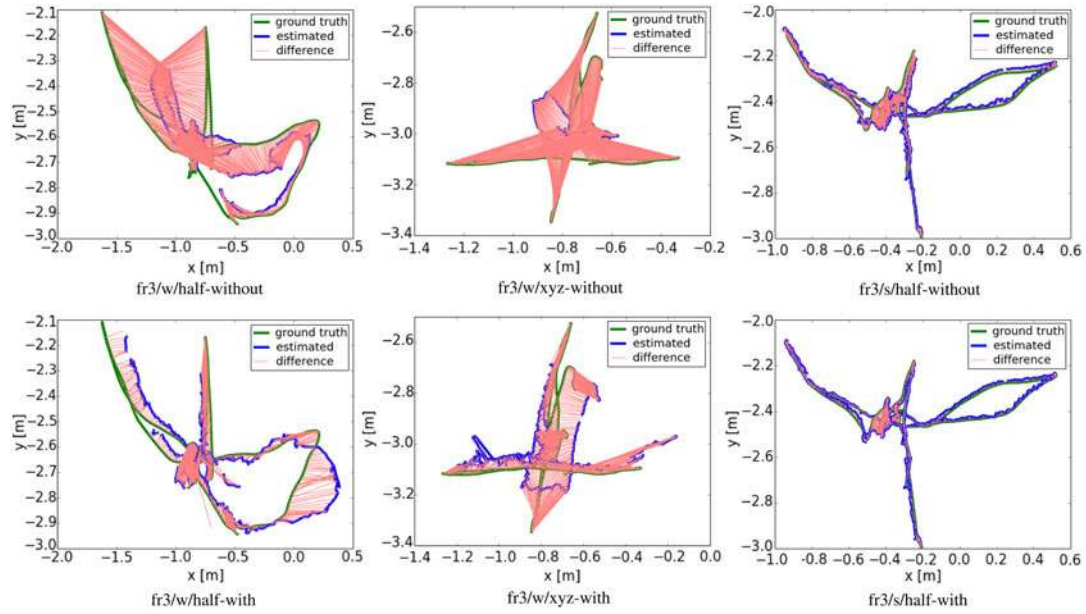

Fig. 5. Plots of ATE for the sequences fr3/w/half, fr3/w/xyz/ and fr3/s/half. The words *with* and *without* represent the experiments performed with and without our method. The ground-truth, the estimated trajectory and the differences for each subfigure are, respectively, represented as the green, blue and red lines.

camera localization producing little effect on the localization accuracy. On the other hand, some dynamic feature points are misclassified as static ones. Since these points only account for a small portion, they will be recognized as outliers by the RANSAC algorithm in the visual SLAM system and filtered out. Thus, the localization accuracy is guaranteed.

Figure 5 shows the qualitative results of ORB-SLAM which is combined with our approach. We use absolute trajectory error (ATE) to evaluate the localization accuracy. The ATE directly measures the difference between points of the true and the estimated trajectory. In each subfigure, the ground truth, the estimated trajectory and the differences are, respectively, represented as the green, blue and red lines. Longer red lines indicate larger estimation errors and worse localization accuracy. We use *-with* and *-without* to represent that the experiments are performed with and without our method. It is shown that the localization is significantly improved after integrating our method into ORB-SLAM framework.

Table II shows the global consistency performance. We observe that our method shows significant improvements for all the sequences in terms of RMSE and S.D. In high-dynamic scenarios, further performance gains can be observed, by reporting the highest 95.86%. These experimental results demonstrate that our method can perform well in high-dynamic scenarios. As for the low-dynamic

https://doi.org/10.1017/S0263574719000584 Published online by Cambridge University Press

Table II. ATE in meters for the experiments without and with our proposed method.

	Without our approach				**With our approach**				**Improvements**			
Sequences	**RMSE**	**Mean**	**Median**	**S.D.**	**RMSE**	**Mean**	**Median**	**S.D.**	**RMSE**	**Mean**	**Median**	**S.D.**
fr3/w/half	**0.4579**	0.3987	0.3774	**0.2252**	**0.1612**	0.1091	0.0637	**0.1187**	**64.80%**	72.64%	83.12%	**52.71%**
fr3/w/rpy	**0.9046**	0.7685	0.7092	**0.4772**	**0.1533**	0.1048	0.0635	**0.1119**	**83.05%**	86.36%	91.05%	**76.55%**
fr3/w/xyz	**0.4808**	0.4367	0.4276	**0.2011**	**0.1899**	0.1537	0.1148	**0.1115**	**60.50%**	64.80%	73.15%	**44.55%**
fr3/w/half/v	**0.5591**	0.4567	0.2934	**0.3226**	**0.0671**	0.0435	0.0283	**0.0506**	**88.00%**	90.48%	90.35%	**84.31%**
fr3/w/rpy/v	**0.5799**	0.3534	0.0556	**0.4599**	**0.0299**	0.0240	0.0186	**0.0178**	**95.86%**	93.21%	66.55%	**96.13%**
fr3/w/xyz/v	**1.4212**	1.2811	1.1664	**0.6153**	**0.1415**	0.0561	0.0305	**0.1299**	**90.04%**	95.62%	97.39%	**78.89%**
fr3/s/half*	**0.0198**	0.0158	0.0135	**0.0120**	**0.0179**	0.0147	0.0131	**0.0102**	**9.60%**	6.96%	2.96%	**15.00%**
fr3/s/xyz*	**0.0097**	0.0088	0.0083	**0.0042**	**0.0092**	0.0081	0.0075	**0.0043**	**5.15%**	7.95%	9.64%	**−2.38%**
fr2/d/person*	**0.0090**	0.0083	0.0082	**0.0036**	**0.0067**	0.0061	0.0057	**0.0029**	**25.56%**	26.51%	30.49%	**19.44%**

Notes: Low-dynamic sequences are denoted with a superscript star. Others are high-dynamic sequences. Our method effectively improves the ORB-SLAM performance in all scenarios in terms of ATE.

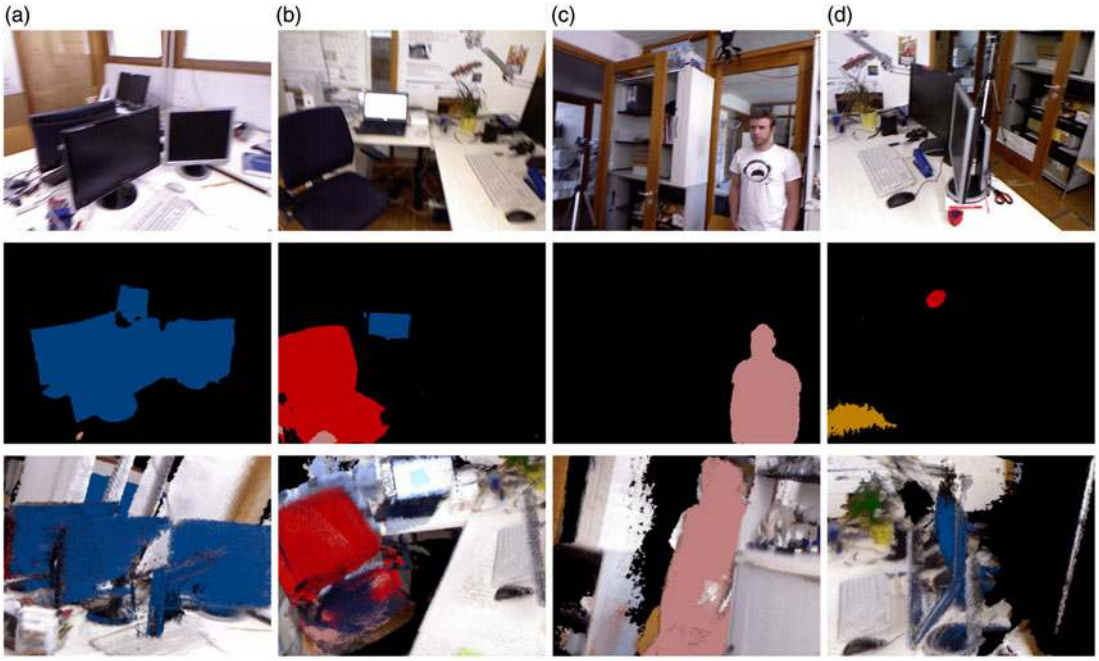

Fig. 6. Some illustrative results of our method using the TUM dataset. With the input RGB images displayed in the top row, the results of semantic segmentation and mapping are shown in the middle and bottom rows, respectively. Intuitively, the regions of chairs, the persons and the monitors in the images are highlighted in red, pink and blue, respectively.

scenarios, our method provides less improvements ranging from 5.15% to 25.56%, which implies that relatively less dynamic feature points can be easily distinguished; thus, the original ORB-SLAM can perform quite well in such a situation.

4.4. Results of semantic mapping

Figures 6 and 7 qualitatively demonstrate the results of our method. With the input RGB images displayed in the top row, the results of semantic segmentation and mapping are shown in the middle and bottom rows, respectively. Intuitively, the regions of chairs, the persons and the monitors in the images are highlighted in red, pink and blue, respectively. As can be shown in the figures, our approach shows highly accurate semantic mapping results, which sufficiently suggests that the point-wise labeled point cloud allows well reconstructing the environment. However, the semantic segmentation process often misclassified some objects due to the insufficient and ambiguous information. On one hand, insufficient information cannot provide enough data to recognize an object. For example, in Fig. 6, the sizes of the monitors are too small to provide enough information. Consequently, inaccurate semantic segmentation results are produced leading to the failure case when some objects are not correctly recognized. On the other hand, ambiguity may lead to the biased information. For example, in Fig. 7, most of the chair region is occluded by a person leading to the ambiguity that the scene contains only one person or one person and one chair. Thus, the chair is incorrectly recognized as one part of a person leading to the degraded segmentation performance.

In terms of the mapping result, our approach is capable of recovering the environments accurately as shown in Figs. 6 and 7. However, there are some drifts in each sequence. For example, in Fig. 7(a), there are two monitors in the semantic mapping result. The sequence finishes a loop here, and the accumulative error is detrimental to the camera pose estimation.

Table III qualitatively illustrates the experimental results on the TUM dataset and the office dataset. In particular, we use monitors, chairs and persons as our segmentation samples. Firstly, the following statistics are calculated. Let's take the chair for example.

- TP (true positive): The number of frames where there is a chair in the frame and the chair has been correctly segmented.

https://doi.org/10.1017/S0263574719000584 Published online by Cambridge University Press

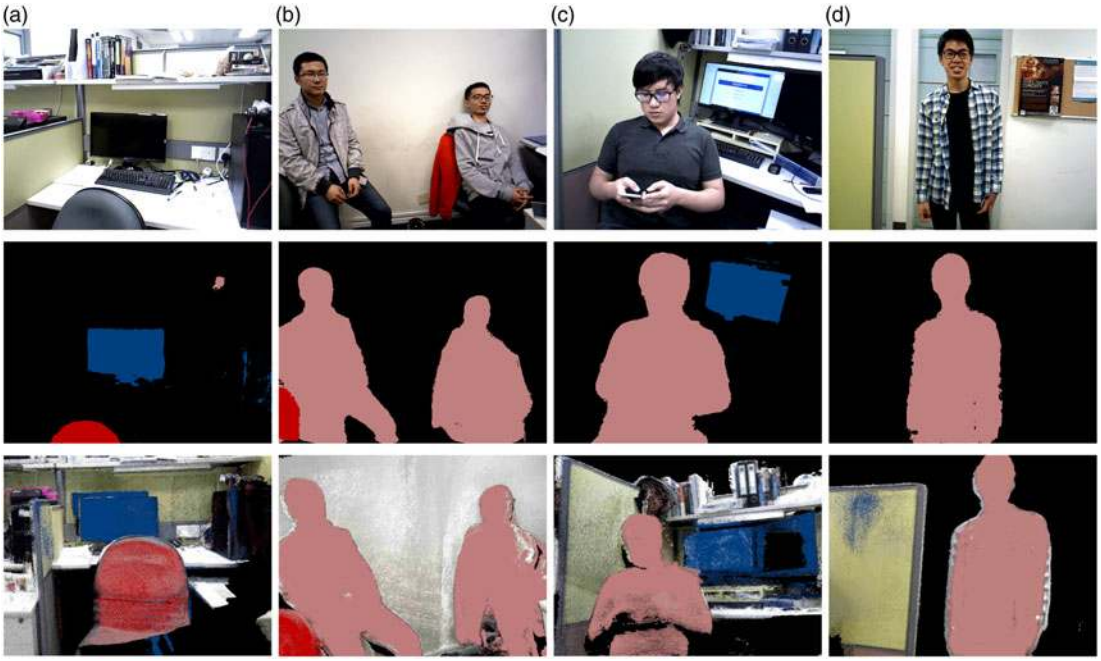

Fig. 7. Some illustrative results of our method using our office dataset. With the input RGB images displayed in the top row, the results of semantic segmentation and mapping are shown in the middle and bottom rows, respectively. Intuitively, the regions of chairs, the persons and the monitors in the images are highlighted in red, pink and blue, respectively.

- FP (false positive): The number of frames where there is no chair in the frame but some object has been incorrectly segmented as a chair.
- TN (true negative): The number of frames where there is no chair in the frame and no object has been segmented as a chair.
- FN (false negative): The number of frames where there is a chair in the frame but no object has been correctly segmented as chair.

We use the following metrics for the quantitative evaluations: False Positive Rate (FPR), False Negative Rate (FNR), Recall (Re), Precision (Pr) and Percentage of Wrong Classifications (PWC). These are calculated as follows:

$$FPR = \frac{FP}{FP + TN} \tag{6}$$

$$FNR = \frac{FN}{TP + FN} \tag{7}$$

$$Re = \frac{TP}{TP + FN} \tag{8}$$

$$Pr = \frac{TP}{TP + FP} \tag{9}$$

$$PWC = \frac{FN + FP}{TP + FN + FP + TN} \tag{10}$$

For *Re* and *Pr*, high values indicate high precision of the semantic segmentation. For *FPR*, *FNR* and *PWC*, high values indicate low precision of the semantic segmentation.

It is shown that *FPR* is reported at less than 0.1, which demonstrates that our method enables segmenting out the specific objects in the sequence accurately. In the case when the chairs appear in the sequence, higher FPR scores regarding the chairs can be achieved compared with the counterparts of the other two objects, due to the observation that the features of a chair are not distinctive or some

https://doi.org/10.1017/S0263574719000584 Published online by Cambridge University Press

Table III. The quantitative evaluation results of our method using the TUM and our sequences.

	fr1_room			**fr1_360**			**fr1_xyz**			**cuhk_office**		
	Monitor	**Chair**	**Person**	**Monitor**	**Chair**	**Person**	**Monitor**	**Chair**	**Person**	**Monitor**	**Chair**	**Person**
FPR	1.31%	5.24 %	0.50 %	4.83 %	7.41 %	1.12 %	0.00 %	10.34 %	0.50 %	0.00 %	10.29 %	0.00 %
FNR	6.31 %	10.22 %	1.72 %	8.18 %	14.53 %	3.21 %	10.20 %	30.50%	44.32 %	14.22 %	30.53 %	5.51 %
Re	85.11%	70.00 %	91.17 %	76.69 %	67.84 %	81.27 %	89.68 %	76.59%	68.50 %	88.44%	71.98 %	94.32%
Pr	97.15 %	88.00 %	98.40 %	93.35 %	84.67 %	95.59 %	95.20 %	82.44%	60.73 %	94.58 %	78.90 %	98.26%
PWC	12.45%	20.40 %	5.53 %	15.41%	25.19 %	10.00 %	8.87 %	31.54 %	23.56 %	11.62 %	33.37%	3.19 %

parts of the chair are occluded by the other object like a sitting person. In addition, observed from different viewpoints, the shape of a chair changes more significantly than other objects. By contrast, higher *FNR* scores are reported than those of the *FPR*, which implies that insufficient and ambiguous information imposes a disturbance on our method. In addition, the system performs better in the cases when monitors are included. The scores of the *Re* and the *Pr* are impressively high, while those of *PWC* are low. This demonstrates that our model has high precision on the test dataset. Note that our approach shows good performance for persons, due to the availability of a large amount of training data.

4.5. Discussion

While it is efficient to leverage the CRF-RNN for semantic mapping, it still suffers from the following limitations. Firstly, in real-world scenarios, the motion of the camera can be drastically varying and objects are sometimes occluded deteriorating the performance of CRF-RNN. Secondly, the depth information is not used in the model. With the RGB-D dataset,[39–43] the model should be trained using both the RGB and the depth information. With the depth information incorporated into our system, further performance improvements can be well expected. Finally, there are close correlations between the consecutive frames that the CRF-RNN does not take into consideration. For example, one person may change his location in the sequence. Once the person is recognized, we should track him until he leaves out of the view. In this way, even if some parts of the person are occluded, accurate recognition can be still achieved via robust human tracking. As a result, semantic segmentation should be beneficial for visual SLAM, whilst visual SLAM should provide rewarding feedback to semantic segmentation.

In the implementation, we empirically set the tolerance τ as 7 pixels. The predefined parameter is suitable for most of the cases. However, in some cases when drastic camera motion or objects motion is present, the fixed parameter cannot generalize very well resulting from the noise generated from the motion. In our future work, we will attempt to find an adaptive way to adjust the value of the parameter based on the property for the certain sequence. Also, we will eliminate the noise in an efficient way.

5. Conclusions

This paper has proposed a novel approach to conduct semantic mapping using the convolutional neural network (CNN). The approach enables a robot to utilize both the geometric and semantic information in challenging environments. A dynamic elimination method is proposed and applied to ensure the accuracy of the camera pose estimation in dynamic environments. The advantage of this approach manifests itself to the ability to combine the semantic information with the traditional output of the visual SLAM system. With the dynamic elimination method, the proposed approach can be applied in dynamic scenarios. We conduct both qualitative and quantitative evaluations that demonstrate the feasibility of our proposed method.

In our future work, we will extend this work in several directions. We have successfully registered the semantic information into a 3D geometric map. Then we will use our approach in an exploration application and test it in a real-world scenario. While the proposed system works well for indoor scenes, it does not suit for outdoor environments like those described in KITTI.[44] In our future work, we will further develop the current system to generalize well for both indoor and outdoor scenarios. In addition, we will make full use of the relation between the consecutive frames to robustly and precisely segment objects out while comparing the performance with the one that deals with the individual frame. To further improve the performance of our system, we will take into account the depth information to achieve much better performance. Thus, further extension of our system can be achieved for handling different dynamic scenarios.

Acknowledgments

The authors thank Zhe Min for his very helpful discussions during the preparation of this manuscript. The authors also thank Ang Zhang, Po-Wen Lo and Danny Ho for acting as objects in our dataset.

This work was supported by the Hong Kong Research Grants Council General Research [grant number 14205914], Innovation and Technology Commission Innovation and Technology [grant number ITS/236/15], and Shenzhen Science and Technology Innovation projects [grant numberJCYJ20170413161616163 to Max Q.-H. Meng].

References

1. G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," *ISMAR 2007 Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality,* Nara, Japan (2007) pp. 225–234.
2. J. Engel, T. Schöps and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular Slam," *European Conference on Computer Vision*, Zurich, Switzerland (2014) pp. 834–849.
3. R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.* **33**(5), 1255–1262 (2017).
4. T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker and A. Davison, "Elasticfusion: Dense Slam Without a Pose Graph." *Robotics: Science and Systems: A Robotics Conferences*, Rome, Italy (2015).
5. C. Kerl, J. Sturm and D. Cremers, "Dense Visual Slam for RGB-D Cameras," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan (2013) pp. 2100–2106.
6. C. Wang, L. Meng, S. She, I. M. Mitchell, T. Li, F. Tung, W. Wan, M. Meng, C. W. de Silva and W. Clarence, "Autonomous Mobile Robot Navigation in Uneven and Unstructured Indoor Environments," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, Canada (2017) pp. 109–116.
7. D. Zhu, T. Li, D. Ho, C. Wang and M. Q.-H. Meng, "Deep Reinforcement Learning Supervised Autonomous Exploration in Office Environments," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia (2018) pp. 7548–7555.
8. Y. Sun, W. Zuo and M. Liu, "Rtfnet: RGB-thermal fusion network for semantic segmentation of urban scenes," *IEEE Robot. Autom. Lett.*, 2576–2583 (2019).
9. C. Wang, J. Cheng, J. Wang, X. Li, and M. Q.-H. Meng, "Efficient object search with belief road map using mobile robot," *IEEE Robot. Autom. Lett.* **3**(4), 3081–3088 (2018).
10. J. Cheng, H. Cheng, M. Q.-H. Meng and H. Zhang, "Autonomous Navigation by Mobile Robots in Human Environments: A Survey," *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Kuala Lumpur, Malaysia (2018) pp. 1981–1986.
11. R. Raguram, O. Chum, M. Pollefeys, J. Matas and J.-M. Frahm, "USAC: a universal framework for random sample consensus." *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(8), 2022–2038 (2013).
12. J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA (2016) pp. 779–788.
13. J. Long, E. Shelhamer and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, USA (2015) pp. 3431–3440.
14. R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: an open-source slam system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics* **33**(5), 1255-1262 (2017).
15. S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang and P. H. Torr, "Conditional Random Fields as Recurrent Neural Networks," *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile (2015) 1529–1537.
16. J. Cheng, Y. Sun and M. Q.-H. Meng, "A Dense Semantic Mapping System Based on CRF-RNN Network," *2017 18th International Conference on Advanced Robotics (ICAR),* Hong Kong, China (2017) pp. 589–594.
17. M. Bloesch, S. Omari, M. Hutter and R. Siegwart, "Robust Visual Inertial Odometry using a Direct EKF-Based Approach," *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Hamburg, Germany (2015) pp. 298–304.
18. V. Usenko, J. Engel, J. Stückler and D. Cremers, "Direct Visual-Inertial Odometry with Stereo Cameras," *2016 IEEE International Conference on Robotics and Automation (ICRA),* Stockholm, Sweden (2016) pp. 1885–1892.
19. D.-H. Kim, S.-B. Han and J.-H. Kim, "Visual odometry algorithm using an RGB-D sensor and IMU in a highly dynamic environment," **In:** *Robot Intelligence Technology and Applications 3*, (Springer, Cham, 2015) pp. 11–26.
20. Y. Sun, M. Liu and M. Q.-H. Meng, "Active perception for foreground segmentation: An RGB-D data-based background modeling method," *IEEE Trans. Autom. Sci. Eng.* (2019). Early Access.
21. D.-H. Kim and J.-H. Kim, "Effective background model-based RGB-D dense visual odometry in a dynamic environment," *IEEE Trans. Robot.* **32**(6), 1565–1573 (2016).
22. Y. Sun, M. Liu and M. Q.-H. Meng, "Improving RGB-D slam in dynamic environments: A motion removal approach," *Robot. Autonom. Syst.* **89**, 110–122 (2017).
23. S. Li and D. Lee, 'RGB-D slam in dynamic environments using static point weighting," *IEEE Robot. Autom. Lett*. **2**(4), 2263–2270 (2017).
24. Y. Sun, M. Liu and M. Q.-H. Meng, "Motion removal for reliable RGB-D slam in dynamic environments," *Robot. Autonom. Syst.* **108**, 115–128 (2018).
25. D. Zou and P. Tan, "Coslam: Collaborative visual slam in dynamic environments," *IEEE Trans. Pattern Anal. Machine Intell.* **35** (2), 354–366 (2013).
26. Y. Wang and S. Huang, "Motion Segmentation Based Robust RGB-D Slam," *2014 11th World Congress on Intelligent Control and Automation (WCICA),* Shenyang, China (2014) pp. 3122–3127.
27. T. Terashima and O. Hasegawa, "A Visual-Slam for First Person Vision and Mobile Robots," *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA),* Nagoya, Japan (2017) pp. 73–76.

https://doi.org/10.1017/S0263574719000584 Published online by Cambridge University Press

28. J. Cheng, Y. Sun, W. Chi, C. Wang, H. Cheng and M. Q.-H. Meng, "An Accurate Localization Scheme for Mobile Robots Using Optical Flow in Dynamic Environments," *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Kuala Lumpur, Malaysia (2018) pp. 723–728.
29. A. A. Panchpor, S. Shue and J. M. Conrad, "A Survey of Methods for Mobile Robot Localization and Mapping in Dynamic Indoor Environments," *2018 Conference on Signal Processing and Communication Engineering Systems (SPACES),* Vijayawada, India (2018) pp. 138–144.
30. A. Hermans, G. Floros and B. Leibe, "Dense 3d Semantic Mapping of Indoor Scenes from RGB-D Images," *2014 IEEE International Conference on Robotics and Automation (ICRA),* Hong Kong, China (2014) pp. 2631–2638.
31. R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly and A. J. Davison, "Slam++: Simultaneous Localisation and Mapping at the Level of Objects," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Portland, Oregon, USA (2013) 1352–1359.
32. N. Sünderhauf, T. T. Pham, Y. Latif, M. Milford and I. Reid, "Meaningful Maps – Object-Oriented Semantic Mapping," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, Canada (2017) pp. 5079–5085.
33. S. L. Bowman, N. Atanasov, K. Daniilidis and G. J. Pappas, "Probabilistic Data Association for Semantic Slam," *2017 IEEE International Conference on Robotics and Automation (ICRA),* Marina Bay Sands, Singapore (2017) pp. 1722–1729.
34. L. Gan, M. G. Jadidi, S. A. Parkison and R. M. Eustice, "Sparse Bayesian inference for dense semantic mapping," *arXiv preprint arXiv:1709.07973*, (2017).
35. B. Triggs, P. F. McLauchlan, R. I. Hartley and A. W. Fitzgibbon, "Bundle Adjustment – A Modern Synthesis," *International Workshop on Vision Algorithms*, Corfu, Greece (1999) pp. 298–372.
36. D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Machine Intell.* **26**(6), 756–770 (2004).
37. S. Baker and I. Matthews, "Lucas-kanade 20 years on: A Unifying Framework," *Int. J. Comp. Vision* **56**(3), 221–255 (2004).
38. J. Sturm, N. Engelhard, F. Endres, W. Burgard and D. Cremers, "A Benchmark for the Evaluation of RGB-D Slam Systems," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Algarve, Portugal (2012) pp. 573–580.
39. S. Song, S. P. Lichtenberg and J. Xiao, "Sun RGB-D: A RGB-D Scene Understanding Benchmark Suite," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, USA (2015) pp. 567–576.
40. N. Silberman and R. Fergus, "Indoor Scene Segmentation Using a Structured Light Sensor," *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops),* Barcelona, Spain (2011) pp. 601–608.
41. J. Xiao, A. Owens, and A. Torralba, "Sun3d: A Database of Big Spaces Reconstructed using SFM and Object Labels," *Proceedings of the IEEE International Conference on Computer Vision*, Sydney, Australia (2013) pp. 1625–1632.
42. S. Song and J. Xiao, "Deep Sliding Shapes for Amodal 3D Object Detection in RGB-D Images," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA (2016) pp. 808–816.
43. B.-S. Hua, Q.-H. Pham, D. T. Nguyen, M.-K. Tran, L.-F. Yu and S.-K. Yeung, "Scenenn: A Scene Meshes Dataset with Annotations," *2016 Fourth International Conference on 3D Vision (3DV),* Stanford, California, USA (2016) pp. 92–101.
44. A. Geiger, P. Lenz and R. Urtasun, "Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite," *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* Providence, Rhode Island, USA (2012) pp. 3354–3361.

https://doi.org/10.1017/S0263574719000584 Published online by Cambridge University Press