DPF-SLAM
Liu, Xin; Wen, Shuhuan; Yuan, Mingxing; Li, Pengjiang; Zhao, Yongjie; Manfredi, Luigi

Published in:
2022 IEEE International Conference on Real-Time Computing and Robotics, RCAR 2022

DOI:
10.1109/RCAR54675.2022.9872255

Publication date:
2022

Citation for published version (APA):
DPF-SLAM: Dense semantic SLAM based on dynamic probability fusion in dynamic environments

Xin Liu¹, Shuhuan Wen¹, Mingxing Yuan², Pengjiang Li¹, Yongjie Zhao¹ and Luigi Manfredi³

Abstract—With the development of simultaneous localization and mapping (SLAM) technology, Dynamic SLAM has been a challenging research topic. This paper presents a dynamic probability fusion SLAM (DPF-SLAM) algorithm, which adds a semantic segmentation thread and a dense reconstruction thread to ORB-SLAM2. We integrate dynamic prior probability obtained by semantic segmentation with dynamic probability obtained by dynamic point detection, which can decrease the effect of dynamic objects on the localization accuracy and meanwhile achieve the dense reconstruction of a static background. We evaluate DPF-SLAM system on the public TUM RGB-D dataset. The experimental results show that the proposed algorithm has better localization accuracy than DS-SLAM and ORB-SLAM2, which are two relatively new algorithms in this area, and can obtain a good dense reconstruction effect. Moreover, through the performance comparison with our previous work, it is found that the algorithm speed and positioning accuracy are improved.

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) have been widely used in various fields of applications [2]. SLAM techniques can build maps of unknown environments and estimate the pose of robots by visual sensor [3], [4] and Lidar [5], [6]. In recent decades, visual SLAM (VSLAM) has been well studied using visual sensors including monocular camera [7], stereo camera [4] and RGB-D camera [3], [8] to estimate ego-motion of camera and simultaneously update a map incrementally.

However, most of the current VSLAM approaches assume a static environment or classify some dynamic objects as outliers to a static model [9], [10], [11]. Although the static assumption can obtain poses and maps for some robotic applications, it restricts practical applications for highly dynamic environments due to moving people, bicycles, or cars. If these dynamic objects cannot be eliminated as outliers, it may cause wrong correspondences, which lead to false pose estimation and inconsistent maps.

Recently, some work has been explored to cope with moving objects and reconstruct maps in dynamic environments. In [12], [13], [14], [15], the accuracy of camera tracking is improved by detecting and removing possible moving objects and creating a static map. However, most of the moving targets only account for a small part of the image, and the use of sparse matching cannot guarantee robustness and long-term feature tracking. In [16], a dense optical flow method is used to maximize the number of points tracked on moving objects, but the computational complexity increases.

To address these problems, we judge the motion states of all objects by fusing the dynamic probability of objects and the calculation results of sparse feature points and complete the pose estimation according to the features in the static region to build a dense map.

Based on ORB-SLAM2 [9], we propose DPF-SLAM, a novel and effective Dynamic SLAM algorithm. To achieve accurate localization and dense map reconstruction. On the basis of our previous work [17], the main contributions of this paper are summarized as follows:

1) We design a framework that integrates an off-the-shelf semantic segmentation network (i.e., SegNet) and a TSDF dense reconstruction thread to ORB-SLAM2. The proposed method can perform dense reconstruction of static background in dynamic scene.
2) We propose a Bayesian framework that considers both the prior knowledge of the semantic information (obtained from SegNet) and the dynamic probability (obtained from the dynamic point detection method proposed in [14]) to handle dynamic objects, which can eliminate dynamic points more accurately.
3) We evaluate the performance of our algorithm on the TUM RGB-D dataset. The results show that DPF-SLAM can detect moving objects efficiently and improve the positioning accuracy in high-dynamic environments.

II. RELATED WORK

In recent years, some solutions have been proposed for dynamic SLAM. Li et al. [3] propose a real-time depth edge-based frame-to-keyframe registration to reduce the influence of the odometry algorithm for dynamic targets. However, the algorithm is not suitable for a textureless scene because it only derives the information from edges. Tan et al. [19] detect the changing features by comparing the appearance and structure of features projected from the keyframes to the current frame. But this algorithm is invalid for fast-moving objects. In [20], the authors detect dynamic targets using a dense scene flow algorithm, but the system falsely detects static points as dynamic points in a weak texture scene. Sun et al. [21] use the motion removal approach integrated into
the frontend of RGB-D SLAM to filter out data on moving objects, but this algorithm has low real-time performance.

With the development of deep learning, Convolutional Neural Networks (CNNs) have been used to detect moving objects in dynamic environments. Yu et al. [14] adopt Polar geometric constraints to detect dynamic target points. But this algorithm only segments people in the scene and detects dynamic objects falsely. Zhang et al. [25] use Mask R-CNN [26] to perform semantic segmentation, which cannot segment the book taken in one’s hand. Xu et al. [27] can achieve dense reconstruction for static background and the proposed system is robust in a dynamic scene, but the moving cup is also reconstructed and included in the map.

Our previous work [17] proposes a method using mask R-CNN to detect dynamic objects and build a semantic map. To improve the efficiency in a dynamic environment, we propose a new SLAM framework, which uses SegNet [28] algorithm and probabilistic fusion method to estimate camera pose in dynamic environments and construct consistent dense maps.

III. THE PROPOSED SLAM SYSTEM

Fig. 1 shows the overall pipeline of our DPF-SLAM. There are five threads in DPF-SLAM, tracking, segmentation, local mapping, loop detection and TSDF reconstruction. We use SegNet [28] to perform semantic segmentation on input RGB images and assign a dynamic prior probability for every segmented object in the semantic segmentation thread. In the tracking thread, we adopt LK optical flow algorithm [29] to track the ORB features of the previous frame and detect dynamic feature points using the Epipolar constraint. Dynamic probability for every class can be obtained according to the amounts of features in different classes. We use the Bayesian model to update the dynamic probability and obtain dynamic classes in the scene. The dynamic feature points on the detected dynamic classes are used as seed points, with which we use the region growing algorithm to dynamic segment objects. Finally, we remove the feature points from dynamic objects and estimate the camera poses using the static feature points in the tracking thread. The estimated camera pose, RGB-D image and the segmentation results of dynamic objects are sent to the dense reconstruction thread. Dense reconstruction for the scene is achieved by using the TSDF algorithm.

A. Semantic segmentation in DPF-SLAM

The input of SegNet is 3-channels RGB image. The output is the semantic label. Different classes are visualized with different colors. We use PASCAL VOC [30] dataset to train SegNet. The detected objects in an image are assigned prior probabilities being dynamic objects according to the semantic segmentation results. Fig. 2 Shows the four-layer probability pyramid used in this work. Each layer includes different classes. The dynamic probability decreases from the top layer to the bottom layer.

The top layer includes 4 classes, in which people, cats, and dogs are assigned the probability of 0.9 since they are often in moving states. The birding class is assigned 0.8 because our target environment is indoor and birds barely appear in indoor environments. The second layer includes 12 classes, among which bottles, chairs and bicycles are movable objects. They are assigned the probability of 0.7. For the other classes, we assign the probability of 0.5 as they are not likely to appear in indoor environments. The third layer includes objects that are unlikely to be moved during the SLAM process, such as the dining table, potted plant, sofa and tv/monitor. They are assigned the probability of 0.1. The bottom layer only includes the background, for which the probability is set to 0.

B. Dynamic points detection

With the prior probability, we adopt the Epipolar constraint to detect dynamic points in our DPF-SLAM. First, we use the Pyramid LK optical flow algorithm to track ORB feature points between adjacent frames. Second, the fundamental matrix \( F \) between adjacent frames is found by using the feature points with relative low prior probabilities. Specifically, we only use the feature points from the classes of human, dog and cat, while other classes are removed. Finally, the Epipolar constraint is used to filter out the dynamic feature points.

The mathematics model of Epipolar constraint is described as follows:

\[
E_{cc}(i) = q_i^T l = q_i^T F p_i = 0 \quad (1)
\]

where \( p_i \) is the \( i \)th feature point in the previous frame, \( q_i \) is the matched feature in the current frame, \( l \) is the epipolar line corresponding to \( p_i \) in the current frame. Then is the epipolar line \( l \) corresponding to feature points \( p_i \) in the current frame. The projection of \( p \) on the second image \( I_2 \) must lie on the epipolar line \( l \), which is described in Eq.(1). Due to disturbances from dynamic objects, Eq.(1) is difficult to be established, but the distance between \( q_i \) and epipolar line \( l \) should be very small for static points. The model is shown in Fig. 3.

Assume that the equation for the epipolar line is \( l : A x + B y + C = 0 \) and its vector is \([A, B, C]^T\), feature point \( p_i = [u_i^q, v_i^q, 1]^T \) and its matching point \( q_i = [u_i^q, v_i^q, 1]^T \), then the distance is expressed as:

\[
d_i = \frac{|A u_i^q + B v_i^q + C|}{\sqrt{A^2 + B^2}} = \frac{|q_i^T F p_i|}{\sqrt{A^2 + B^2}} \quad (2)
\]

\[
\sqrt{A^2 + B^2} = \left\| [F p_i]_{x,y} \right\|_2 \quad (3)
\]

where \( l = F p_i \), \([\cdot]_{x,y} \) represents the first two dimensions of vector, \( \left\| \cdot \right\|_2 \) is L-2 norm. The error of epipolar constraint is obtained by Eq.(2) and compared with threshold \( \theta_d \). If the error is larger than \( \theta_d \), then the point is dynamic point. In this paper, \( \theta_d = 0.1 \).

We remove the dynamic points on the edge of an object using depth images in order to better segment the dynamic object. Depth variance for dynamic points-centered and in neighborhood \( \Omega \) with the size of the window \( l_x \times l_y \) is solved by traversing the detected dynamic points and compared with
depth variance threshold $\sigma_d$. If the variance is smaller than $\sigma_d$, the point is not an edge point and is added to the dynamic point set. Instead, the point is on the edge of the object and should be discarded. Algorithm steps are shown in Algorithm 1.

C. Dynamic probabilistic fusion and Dense reconstruction

Assume that N dynamic points \( \{ p_1, \cdots, p_i, \cdots, p_n \} \) are detected in B part, and the current image includes M categories \( \{ B_1, \cdots, B_j, \cdots, B_M \} \) detected by SegNet. Thus, the number of dynamic points in each category \( \{ n_1, \cdots, n_j, \cdots, n_M \} \) can be obtained by traversing N dynamic points. The dynamic probability of the \( j \) category based on dynamic points distribution is:

$$P(B_j) = \frac{n_j}{N}$$  \hspace{1cm} (4)

After segmentation using SegNet, each category is distributed by a prior probability. The segmented dynamic object using prior probability is not accurate because the prior probability is obtained by experience and exists falsely segmentation. A few incorrect detection points could appear because of noise in the process of dynamic points detection. The dynamic probability solved only by dynamic points is not accurate. In order to improve the accuracy of dynamic probability, we fuse dynamic probability using the Bayesian model. The probability of the \( j \) category as a dynamic object is defined as follows:

$$P(S_j|V) = \frac{P(V|S_j)P(S_j)}{P(V)}$$  \hspace{1cm} (5)

where \( S_j \) represents the \( j \) category state (dynamic or static), \( V \) represents observation information, \( P(V) \) is constant, \( P(S_j|V) \) is the \( j \) category as dynamic object and is also called posterior probability. \( P(V|S_j) \) is likelihood probability. In this paper, \( P(B_j) \) in Eq.(4) is regarded as likelihood probability. \( P(S_j) \) is prior probability that is the distributed dynamic probability according to semantic segmentation results. So the probability of the \( j \) category as dynamic object is redefined as follows:

$$P(S_j|V) = \eta P(B_j)P(S_j)$$  \hspace{1cm} (6)

where \( \eta = 1/P(V) \) and it is a constant.

After updating the dynamic probability of every category using the Bayesian model, all kinds of sports in the scene are determined by threshold \( \varepsilon \). If the dynamic probability is less than \( \varepsilon \), then this kind of object is static. Otherwise, this kind of object is dynamic. We define \( \varepsilon = 0.2 \) in our experiments.
If there exist many same category objects in the scene, all the same category objects will be removed as dynamic objects because the semantic segmentation does not belong to case segmentation. For example, there exist two people at different positions in the scene, one is static and the other is dynamic. The static people are removed as dynamic objects, which will reduce the tracked static feature points and influence the accuracy of the building map. In order to decrease the misdetection, the screened dynamic points in the dynamic category are used as seed points and the dynamic objects are completely detected using the region growing algorithm. After the object is segmented, the feature points on the dynamic object are removed. Then we use a uniform speed model or reference frame model or relocation model in the tracking thread to estimate the pose of the camera $T$.

We use TSDF to obtain good dense reconstruction. Dense reconstruction thread obtains RGB-D image and pose of every frame from Tracking thread, and uses pose and depth image to recover 3D information of every frame and overlap.

**Algorithm 1 Dynamic feature points detection**

**Require:** previous frame $F_{ref}$, previous frame ORB feature $p_{ref}$, dynamic probability of previous frame category $P_{pos}$, current frame $F_{cur}$, depth image of current frame $D_{cur}$, prior probability of current frame category $P_{prior}$, neighborhood window with the size of $l_x$ by $l_y$.

**Ensure:** dynamic points set \( \{DyPoints\} \)

1. Track the matched points set in the current frame $p_{cur}$: $calcOpticalFlowPyrLK(F_{ref}, F_{cur}, p_{ref}, p_{cur})$
2. for each $p_i$ in $p_{ref}$ do
3. \( q = p_i^{cur} \)
4. if $P_{prior}(q) < 0.8$ or $P_{pos}(p) = 0$ then
5. Save static matched point pairs:
6. Add $p_i$ to $p_{ref}^{static}$, add $q$ to $p_{cur}^{static}$
7. end if
8. end for
10. for each $p_i$ in $p_{ref}$ do
11. \( q = p_i^{cur} \)
12. Compute epipolar constraint error $d$: $\frac{|q^T F p_i|}{||F p_i||}$
13. if $d > \theta_d$ then
14. Add $q$ to $\{DyPoints\}$
15. end if
16. end for
17. for each $p_i$ in $\{DyPoints\}$ do
18. Compute depth mean value and variance in the center of $p_i$ with neighborhood window $\Omega$: meanStdDev
   $(D_{cur}(\Omega), \mu, \sigma)$
19. if $\sigma > \sigma_d$ then
20. remove $p_i$ from $\{DyPoints\}$
21. end if
22. end for
23. return $\{DyPoints\}$

It is not useful to construct a map for the detected foreground object or the object with a prior probability greater than 0.8. Some uncertain depth data or undetected depth points can also influence mapping. The dense reconstruction thread can obtain semantic segmentation image and dynamic object segmentation image from the semantic segmentation thread and tracking thread, separately. Then we remove the detected foreground object, the object with prior probability greater than 0.8, uncertain depth points and zero-depth points. These points do not take part in map construction, which can realize dense reconstruction of the background in a scene.

**IV. EXPERIMENTAL RESULTS**

We use the public TUM dataset [31] to evaluate the proposed algorithm. Absolute Pose Error (APE) and Relative Pose Error (RPE) is used to compare the performance of these algorithms. We design two sets of experiments, one on the detection results of dynamic objects and the removing results of dynamic features, and the other on the performance comparison of the proposed with ORB-SLAM2 [7], DS-SLAM [14] and our previous work [17] in different sequences.

**A. Dynamic Object Detection and Dynamic Feature Removing**

To demonstrate the effect of the proposed under dynamic environments, we choose f3/walking_halfsphere, f3/walking_static and f3/sitting_halfsphere to testify our algorithm. In f3/walking_halfsphere, two people constantly move before the camera. In f3/walking_static, two people are stationary, then one moves and the other is still stationary, and finally both people move. In f3/sitting_halfsphere, the bodies of the two people are mostly stationary while their hands are moving.

Fig. 4 to Fig. 6 are the results of objects detection using Segnet, DS-SALM and DPF-SLAM on f3/walking_halfsphere, f3/walking_static and f3/sitting_halfsphere. In Fig. 4, the detection result is not complete or incorrect detection because the segmentation accuracy of Segnet is not very high. The segmentation result using DS-SLAM is the same as Segnet because DS-SLAM relies on the detection result of Segnet. But the proposed DPF-SLAM algorithm can
Fig. 5: The segmentation results of the 85th frame and the 587th frame on f3/walking_static, the first column is original image, the second column is the detection result using Segnet, the third column is the detection result using DS-SALM, the last column is the detection result using DPF-SLAM.

Fig. 6: The segmentation results of the 9th frame and the 58th frame on f3/sitting_halfsphere, the first column is original image, the second column is the detection result using Segnet, the third column is the detection result using DS-SALM, the last column is the detection result using DPF-SLAM.

detect completely dynamic targets. Fig. 5 shows a static person and a moving person, and the proposed DPF-SLAM algorithm can detect the true moving person for the same category target. In Fig. 6, the person is static while his hands are moving slightly. The proposed DPF-SLAM algorithm can segment the moving part effectively, which improves the localization accuracy. The experiments demonstrate that DPF-SLAM can segment dynamic objects well and separate dynamic points and static points for extracted features.

B. System performance analysis

In order to further verify the effectiveness of the algorithm, we compare these algorithms quantitatively. The comparison of APE and RMSE is shown in Table I, and the comparison of RPE and RMSE is shown in Table IV. (S) represents target detection algorithm using Segnet, (S+PF) represents target detection algorithm using Segnet and dynamic probability fusion algorithm.

Table III shows the comparison between DPF-SLAM and the previous algorithm in APE and RPE. By analyzing the data in the table, it can be seen that in high dynamic sequences (i.e., f3/w_halfsphere, f3/w_rpy, f3/w_static and f3/w_xyz), the APE values of DPF-SLAM are slightly higher, but the RPE values are lower. The reason is that the target is constantly moving in the high dynamic scene, so it is feasible to take the potential dynamic target as the dynamic target. However, in the low dynamic scenario (f3/s_halfsphere, f3/s_xyz and f3/s_static), the performance of DPF-SLAM is significantly improved compared with the previous algorithm. The reason is that in low dynamic scenes, the target is basically in a static state or has a small displacement, so the characteristics of the target have little influence on pose estimation. However, the previous algorithm treats the potential dynamics in the scene as dynamic targets, which leads to the problem of false detection of dynamic targets, reduces the features that can be used for pose estimation, and further reduces the positioning accuracy. In conclusion, according to the above analysis, compared with our previous algorithm, DPF-SLAM not only improves the algorithm speed but also maintains a higher positioning accuracy.

Finally, dense map without dynamic objects is built using DPF-SLAM. Fig. 7 is the dense map using DPF-SLAM on f3/walking_static, which shows that people don’t take part in dense reconstruction and the proposed algorithm can realize dense reconstruction in dynamic environment.

V. CONCLUSIONS

In this paper, we designed a DPF-SLAM framework by adding a segmentation thread and TSDF dense reconstruction thread to the ORB-SLAM2 framework. Specifically, we presented a novel dynamic object segmentation algorithm based on dynamic probability fusion, which removes the features of dynamic objects to improve the robustness of the SLAM system in dynamic environments. Then, we used (f3/s_halfsphere, f3/s_xyz and f3/s_static). The performance of DPF-SLAM is significantly improved compared with the previous algorithm. The reason is that in low dynamic scenes, the target is basically in a static state or has a small displacement, so the characteristics of the target have little influence on pose estimation. However, the previous algorithm treats the potential dynamics in the scene as dynamic targets, which leads to the problem of false detection of dynamic targets, reduces the features that can be used for pose estimation, and further reduces the positioning accuracy. In conclusion, according to the above analysis, compared with our previous algorithm, DPF-SLAM not only improves the algorithm speed but also maintains a higher positioning accuracy.

Finally, dense map without dynamic objects is built using DPF-SLAM. Fig. 7 is the dense map using DPF-SLAM on f3/walking_static, which shows that people don’t take part in dense reconstruction and the proposed algorithm can realize dense reconstruction in dynamic environment.

V. CONCLUSIONS

In this paper, we designed a DPF-SLAM framework by adding a segmentation thread and TSDF dense reconstruction thread to the ORB-SLAM2 framework. Specifically, we presented a novel dynamic object segmentation algorithm based on dynamic probability fusion, which removes the features of dynamic objects to improve the robustness of the SLAM system in dynamic environments. Then, we used
TABLE III: The comparison of Detection Time.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Segmentation Cost (s)</th>
<th>Tracking Cost(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our Pervious Work [17]</td>
<td>19.6</td>
<td>0.77</td>
</tr>
<tr>
<td>The Proposed</td>
<td>0.30</td>
<td>0.18</td>
</tr>
</tbody>
</table>

TABLE IV: The comparison of RPE.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Comparison of APE</th>
<th>Comparison of RPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>f3/w_halfsphere</td>
<td>0.02679</td>
<td>0.02759</td>
</tr>
<tr>
<td>f3/w_rpy</td>
<td>0.031644</td>
<td>0.186598</td>
</tr>
<tr>
<td>f3/w_static</td>
<td>0.007131</td>
<td>0.010211</td>
</tr>
<tr>
<td>f3/w_xyz</td>
<td>0.019099</td>
<td>0.036129</td>
</tr>
<tr>
<td>f3/s_halfsphere</td>
<td>0.027866</td>
<td>0.019256</td>
</tr>
<tr>
<td>f3_sxyz</td>
<td>0.029580</td>
<td>0.019797</td>
</tr>
<tr>
<td>f3_sstatic</td>
<td>0.069697</td>
<td>0.007582</td>
</tr>
</tbody>
</table>

the TSDF algorithm to realize dense reconstruction for static background in dynamic environments. The feasibility and validity of the proposed algorithm were verified using the public TUM dataset. In our future work, we will improve the deep segmentation network to label more objects, making our algorithm applicable to more environments. We will refine our output culling strategy to make it applicable to a wide range of scenarios.

REFERENCES


