Accurate RGB-D SLAM in Dynamic Environment using Observationally Consistent Conditional Random Fields

Zheng-Jun Du
Qinghai University, Tsinghua University
Xining,China Beijing,China
duzjqhu@aliyun.com

Shi-Sheng Huang
Tsinghua University
Beijing,China
shenghuang.net@gmail.com

Tai-Jiang Mu
Tsinghua University
Beijing,China
taijiang@tsinghua.edu.cn

Qunhe Zhao
DeepBlue Technology Co.,Ltd
Shanghai, China
zhaoqh@deepblueai.com

Ralph R. Martin
Cardiff University
Cardiff,U.K.
MartinRR@cardiff.ac.uk

Kun Xu
Tsinghua University
Beijng,China
xukun@tsinghua.edu.cn

Abstract

Accurate camera pose estimation is essential and challenging for real world dynamic 3D reconstruction and AR applications. In this paper, we present a novel RGB-D SLAM approach for accurate 3D position tracking in dynamic environments. Previous methods detect dynamic components only across a short time-span of consecutive frames. Instead, we provide a more accurate dynamic 3D landmark detection method, followed by the use of observationally consistent conditional random fields, which leverages long-term observations from multiple frames. We further introduce an efficient initial camera pose estimation method based on distinguishing dynamic from static points using graph-cut RANSAC. These static/dynamic labels are used as priors for the unary potential in the conditional random fields, which further improves the accuracy of dynamic 3D landmark detection. Evaluation using the public TUM RGB-D dynamic dataset shows that our approach significantly outperforms state-of-the-art methods, providing much more accurate camera trajectory estimation in a variety of highly dynamic environments. We also show that the dynamic 3D reconstruction can benefit from the camera poses estimated by our proposed RGB-D SLAM.

1. Introduction

Accurate 3D position tracking in an unknown environment is a fundamental technique towards 3D scene perception and understanding [20]. Visual Simultaneous Localization and Mapping (SLAM) is a basic technique for 3D position tracking and environment reconstruction, which receives intense research interest from the computer graphics, computer vision and mixed/augmented/virtual reality communities. Since our daily life often contains dynamic items such as moving people and objects, an accurate visual SLAM which works efficiently in dynamic environment is even urgently needed as basis for various applications in augmented/virtual reality, robotics, autonomous vehicles *etc*.

Although visual SLAM technology has made significant progress in the past few decades [7, 9], most works focus on static environments which could easily fail to track camera poses when facing with dynamics. The critical challenge for dynamic visual SLAM is that the presence of dynamic components violates the data relationships assumed in static SLAM, leading to poor pose estimation. Previous dynamic visual SLAM approaches [1, 27]often utilize a RGB-D depth camera and tackle the dynamic tracking problem following the detection and tracking of moving objects (DATMO) scheme [38]. However, these DATMO-based methods suffer from drawbacks arising from assumptions made about the moving objects with pre-defined number of objects or limited moving speed. The dynamic detection methods using foreground/background segmentation [19], dense scene flow [26] or static/dynamic edge point weighting [24] proposed to track the camera pose solely on the static parts by detecting and eliminating the dynamic region. However, the way in which all of these methods deter-

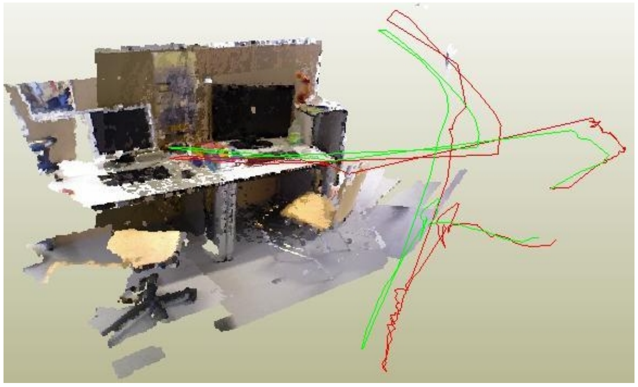


Figure 1. Reconstructed scene for fr3/walking-halfsphere from TUM RBG-D dynamic dataset. As an accurate 3D position tracking technique for dynamic environment, our approach utilizing observationality consistent CRFs can calculate high precision camera trajectory (red) closing to the ground truth (green) efficiently.

mine which regions are static and dynamic relies only on an analysis of a short time-span of consecutive frames, which precludes improving the accuracy of moving object detection over time, with a consequent impact on the accuracy of camera pose estimation. Recently, some fusion-based reconstruction method [42, 34, 35, 31] also provided camera tracking methods in dynamic environment.Though they aimed at reconstructing 3D dynamic scenes and achieved nice camera tracking results, however, these reconstruction-targeted camera pose tracking frameworks are too complicated to be a light-weight one for instant applications in mixed and augmented reality, etc.

In this paper, we provide a more accurate and light-weight dynamic visual SLAM method with an RGB-D sensor, by analysing frames over long-term timescales instead of only short-term ones. The key component of our RGB-D SLAM system is a dynamic camera tracking module based on accurate dynamic 3D landmark detection. Our key observation is that moving objects can be determined more reliably by using long-term observations rather than only brief observations.Based on this key observation, we first estimate the initial camera pose based on the temporally labeled static/dynamic identification by solving a inlier/outlier determination using graph-cut (GC) RANSAC [3]. Then we build an observationally consistent conditional random field (OC-CRF) model to assist in 3D dynamic landmark detection, by analyzing observations of static and dynamic landmarks over a long-term series of consecutive frames. Solving the labeling problem with the aid of the CRF provides highly accurate dynamic detection results. By using the results to eliminate the dynamic 3D landmarks, we can estimate the camera pose with much higher precision using only static 3D landmarks.

Our OC-CRF based dynamic 3D landmark detection is simple for calculation, using which we build an efficient RGB-D visual SLAM system for accurate 3D position tracking in dynamic environments with high precision. We have evaluated our approach on the public TUM RGB-D dynamic dataset [36], which contains several dynamic scenes, ranging from 720 frames to 4200 frames, with two persons walking through an office. The results show that our approach typically outperforms state-of-the-art approaches, such as BaMVO [19] and SPW [24]. Besides, we also propose a dynamic 3D scene reconstruction using our OC-CRF based dynamic SLAM, which can achieve more accurate camera position tracking results than other fusion-based dynamic reconstruction methods, e.g. MaskFusion [34], with good scene reconstruction quality. In summary, this paper makes the following contributions:

1. A reliable dynamic 3D landmark detection method based on an observationally consistent conditional random field, which constitutes the main component of the dynamic camera tracking method, and

2. An efficient method for obtaining an initial estimate of the camera pose for each frame, based on GC-RANSAC filtering, which also provides strong static versus dynamic priors for dynamic 3D landmark detection.

2. Related work

Simultaneous localization and mapping has been studied for more than four decades, with sub-topics of lidar SLAM, visual SLAM, and sensor fusion SLAM according to the different sensors used. In this paper, we focus on visual SLAM, which utilizes cameras (monocular, stereo, or RGB-D) as the primary sensors for localization. In this section, we discuss results particularly relevant to our work, and refer readers to [7] for a more detailed overview of visual SLAM progress in the past few decades.

2.1. Static Visual SLAM

There has been much progress in visual SLAM techniques since the pioneering work of MonoSLAM [8] in 2003. Current visual SLAM approaches can be divided into two categories: *feature-based* visual SLAM methods, which use sparse feature points as landmarks for camera tracking, e.g. PTAM [21] and ORB-SLAM2 [28], and *direct* visual SLAM, which directly uses image intensity for camera tracking without feature points or landmarks extraction, e.g. DTAM [30], SVO [12], LSD-SLAM [11], InfiniTAM [17], PSM-SLAM [39] and DSO [10]. Direct visual SLAM techniques have the advantage of allowing efficient camera tracking without the time-consuming requirement for 2D feature detection needed by feature-based visual SLAM techniques, but they often suffer from lack of robustness in changing light conditions. Besides, there are also approaches to perform camera position tracking by fusing multiple sensors, such as multiple cameras [2], inertial-cameras [32] and laser-inertial-camera [40], or with the aid of deep learning [13, 43].

Currently, most visual SLAM techniques assume a static environment and do not work well in dynamic environments which include human beings or other moving objects. Since feature-based visual SLAM methods such as ORB-SLAM2 [28] work well for robust camera tracking, like ORB-SLAM2, our approach is also a feature-based SLAM system containing three components: camera tracking, local mapping and loop closing. The novelty of our SLAM system lies in the camera tracking subsystem, which in our case handles scenes with dynamic objects. We integrate our dynamic 3D landmark detection and elimination method into the camera tracking component, allowing it to work more accurately in dynamic environments.

2.2. Dynamic Visual SLAM

The detection and tracking of moving objects (DATMO) proposed by Wang et al. [38] in 2006 inspired many dynamic visual SLAM approaches to perform the camera position tracking by detecting moving objects with the aid of dense scene flow [1] or object clustering [16]. Kerl et al. [18] gave the Dense Visual Odometry (DVO) algorithm, which uses a robust error function to reduce the influence of moving objects on camera pose estimation. However, since the error function is only computed across two consecutive frames, the DVO algorithm can only work well for slowly moving environments; rapidly changing ones cause incorrect data associations. Recently, Kim et al. [19] introduced a background-model-based dense-visual-odometry (BaMVO) algorithm to estimate the background of each frame and to perform camera pose estimation by eliminating foreground moving objects. Li et al. [24] provided a dynamic RGB-D SLAM method which uses foreground edge points to estimate the camera's ego-motion. In this method, every edge point is assigned with a static weight which is used in an intensity-assisted iterative closest point (IAICP) algorithm for ego-motion estimation; this reduces the influence of dynamic components. Bujanca et al. [6] presented a framework, FullFusion, for dense semantic reconstruction in dynamic scenes, which enables incremental reconstruction of semantically-annotated non-rigidly deforming objects; the RGB-D data is divided into static and dynamic frames via a segmentation module and only static ones are used for camera pose estimation. Most of these methods detect dynamic components by analysis of only a few consecutive frames, two frames in DVO [18] and just the current frame in BaMVO [19] and Li et al. [24].

However, short-term analysis is not sufficiently informative for moving object detection, since many dynamic components may remain static for short periods, which may mislead short-term determination of static/dynamic status. If not properly detected and eliminated, such dynamic components may be used as landmarks for later camera tracking, misleading downstream 3D to 2D data association, thus lowering the accuracy of camera pose estimation.

In this paper, instead, we provide a dynamic component detection method that uses long-term analysis. Distinguishing static from dynamic components can be done more reliably using long-term observations. Based on this insight, we build an observationally consistent conditional random field using feature vectors derived from multiple visual observation errors over a long period of consecutive frames.

Other works [41, 44] use deep networks such as Faster-RCNN [33] to detect moving objects. Although such methods perform well, the problem of misclassification still exists. Furthermore, the computational cost is much higher due to the use of deep networks. We believe that a geometric approach to dynamic component detection is still not well explored and show that accuracy can be significantly improved without the need for a deep network.

3. Method

3.1. System Overview

An overview of our approach is given in Fig. 2. Following ORB-SLAM2 [28] (RGB-D version), our system also has three components, i.e. dynamic camera tracking, local mapping and loop closing. Local mapping and loop closing are performed as in ORB-SLAM2. The dynamic camera tracking component aims to efficiently estimate the ego-motion for the incoming frames by accurately detecting and eliminating dynamic 3D landmarks, which contains two main subcomponents.

The first subcomponent performs *initial camera pose estimation* using GC-RANSAC (see Sec. 3.2). The initial camera pose estimate is important for our downstream dynamic 3D landmark detection process, since the 2D observations used in the OC-CRF are influenced by the initial camera pose. In this subcomponent, we make an initial identification of static and dynamic points using 2D to 2D matching with GC-RANSAC, which is both efficient and accurate. Then the points determined as static are used for initial camera pose estimation. This initial static/dynamic identification is also used in the dynamic 3D landmark detection step later.

The second subcomponent performs *dynamic 3D landmark detection* using an observationally consistent CRF (see Sec. 3.3). Given the initial camera pose estimate from the previous step, we build an observationally consistent conditional random field (OC-CRF) and use it to *accurately* identify static and dynamic feature points,by solving a labeling problem on the OC-CRF. This allows us to eliminate the dynamic points, and just use the static points to refine

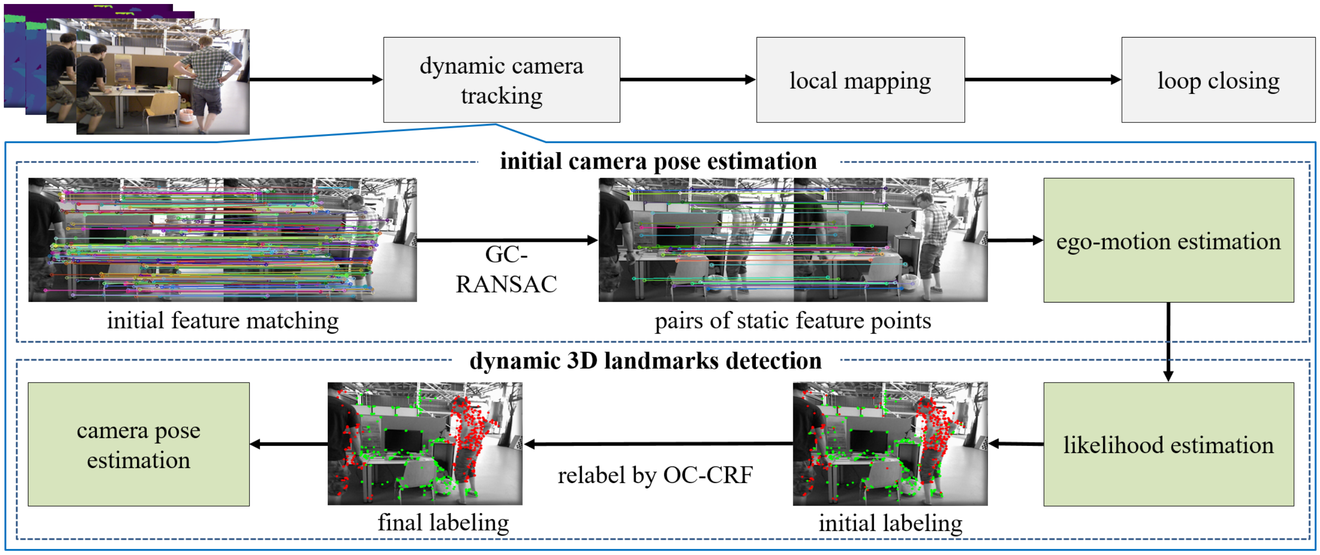


Figure 2. Overview of our approach. To achieve accurate pose estimation in dynamic scenes, camera tracking is performed in two stages, coarse (initial camera pose estimation) and fine (dynamic 3D landmarks detection). We first use GC-RANSAC to filter out dynamic feature points and estimate camera pose on the remaining static feature points, then, we apply OC-CRF to relabel all landmarks, and refine the camera pose using landmarks determined to be static.

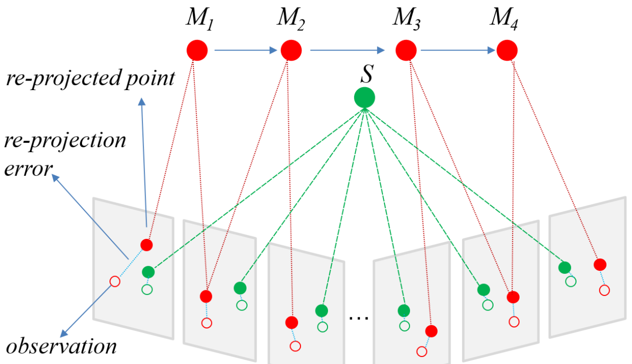


Figure 3. A static landmark has more consistent observations than a dynamic one. M is a landmark which moves from M_1 to M_4 quickly; just a few frames observe the same location. Static landmark S stays at the same location and is seen at the same position in more frames. Re-projected points from static landmarks triangulate to a consistent landmark, while re-projected points from dynamic landmarks triangulate to different landmarks.

the camera pose of the current frame.

Using our dynamic camera tracking method, we can accurately estimate the ego-motion between the current frame and the previous frame, and robustly detect and eliminate dynamic feature points. Then key-frames are selected as in ORB-SLAM2, and are sent to the local mapping components. Finally, the graph-based bundle adjustment further improves the camera pose estimate.

Consistent Observation. The observation of a 3D feature point to a given camera is the projected 2D feature points seen from that camera. Here we say the observation of static objects is *consistent*, since the 2D observations in the views of different frames can all be back-reprojected to one single object. In contrast, the 2D observations of dynamic objects will not be consistent and will back-reproject into multiple objects due to the objects' motion, as shown in Fig. 3.

3.2. Initial Camera Pose Estimation

For every incoming frame, we need to determine a reasonable initial estimation of its camera pose. A general way to do this is to estimate the ego-motion between two consecutive frames by solving a perspective-n-point (PnP) problem [25] with 3D to 2D data association, as ORB-SLAM2 does. However, in dynamic scenarios, the 3D to 2D data association will contain incorrect matches due to the existence of moving objects. To overcome this problem, feature points on moving objects must be detected and eliminated, leaving static feature points to provide an accurate estimate of the ego-motion.

In this step, we first efficiently and coarsely label landmarks as static or dynamic, and then estimate the ego-motion using only the static landmarks. As shown in Fig. 4, for a image pair $\{K_i, K_i'\}$ with fundamental matrix $F(K_i, K_i')$, a 3D landmark P_i with its 2D observation matching pair (p_i, p_i') tend be to static if $p' \in K_i'$ lies on the epipolar line $l_i' = Fp_i$, otherwise be dynamic. So we can formulate the static/dynamic landmark identi-

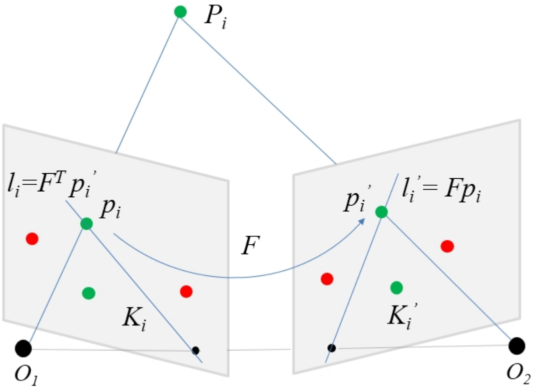


Figure 4. Fundamental matrix and epipolar constraint. For a matched pair (p_i, p_i'), where p_i and p_i' are related to the same 3D point P_i, the epipolar constraint can be expressed as: $p_i'^{\top} F p_i = 0$, i.e. p_i' lies in the epipolar line $l_i' = Fp_i$ or p_i lies in the epipolar line $l_i = F^{\top} p_i'$, where F is the fundamental matrix.

fication problem as inlier/outlier identification during fundamental matrix estimation using the GC RANSAC algorithm [4]. Specifically, during fundamental matrix F estimation, for a given 2D to 2D matching pair set $M = \{(p_i, p_i') | i = 1, \ldots, n\}$ with size n, on each iteration of RANSAC we label each matching pair as an inlier or an outlier for fundamental matrix F estimation. This is performed by optimizing the energy function $E(L) = \sum_i B(L_i) + \lambda \sum_{(i,j)\in G} R(L_i, L_j)$ with $L = \{L_i \in \{0,1\} | i = 1, \ldots, n\}$ being a label assignment for the matching pair set M, and G being a neighbor graph.

The unary term of the energy function is formulated as:

$$B(L_i) = \begin{cases} K\left(\phi\left(p_i, p_i', \theta\right), \epsilon\right) & \text{if} \quad L_i = 0 \\ 1 - K\left(\phi\left(p_i, p_i', \theta\right), \epsilon\right) & \text{if} \quad L_i = 1 \end{cases}, \quad (1)$$

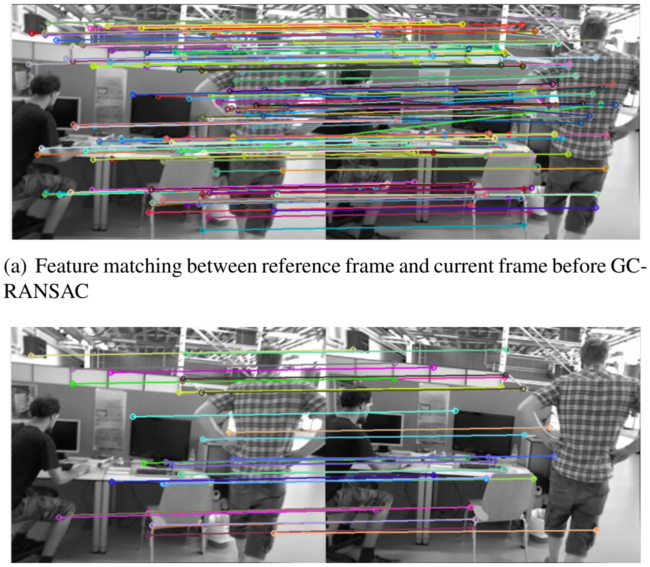
where θ is the angular parameter for fundamental matrix F, and $K(\sigma, \epsilon) = \exp(-\sigma^2/(2\epsilon^2))$. Label $L_i = 0$ indicates an inlier pair and 1 indicates an outlier pair. $\phi(p_i, p_i', \theta)$ is the distance from matching pair (p_i, p_i') to the fundamental matrix F, and ϵ is a threshold for inlier/outlier determination.

The pairwise energy is defined as follows:

$$R(L_i, L_j) = \begin{cases} 1 & \text{if } L_i \neq L_j \\ (B(L_i) + B(L_j))/2 & \text{if } L_i = L_j = 0 \\ 1 - (B(L_i) + B(L_j))/2 & \text{if } L_i = L_j = 1 \end{cases}, \quad (2)$$

The total energy can be efficiently optimised by the graph cut algorithm [5].

Given the purpose of the labelling, we should aggressively remove dynamic points, even at the expense of discarding some static ones. We achieve this by empirically setting $\epsilon = 0.1$, and $\lambda = 0.14$. Besides, since there may be many mismatches between adjacent frames arising due to dynamic objects, which are difficult to filter out by GC-RANSAC, we choose two frames that are far apart in time as input to GC-RANSAC, which helps to ensure that almost all inliers labeled as static are indeed static. Using GC-RANSAC, all landmarks are labeled as inlier (static) or outlier (dynamic). We then estimate the ego-motion using just the static landmarks, to obtain a more accurate pose estimation. We show a example result in Fig. 5 to select static feature points using our GC RANSAC based method, which is summarized in Algorithm 1.



(a) Feature matching between reference frame and current frame before GC-RANSAC

(b) Feature point pairs labeled as inliers after GC-RANSAC

Figure 5. Static feature points selection by the GC-RANSAC. Left: current frame. Right: reference frame. We choose the 10th frame before the current frame as the reference frame. After GC-RANSAC filtering, inliers are almost all static feature points, and are used for initial ego-motion estimation.

We later use the estimated fundamental matrix to derive static/dynamic priors for accurate dynamic point detection (see Sec. 3.3). Specifically, as shown in Fig. 4, for each 2D matching pair $(p_i, p_i'), p_i \in K_i, p_i' \in K_i'$, where K_i and K_i' are the current frame and the previous frame, respectively, assuming P_i is the corresponding 3D landmark, and $l_i \in K, l_i' \in K'$ are the corresponding epipolar lines $l_i = F^{\top} p_i' = (A_i, B_i, C_i)$, $l_i' = Fp_i = (A_i', B_i', C_i')$, we compute the distances between the 2D feature point and the epipolar line as $d_i = \frac{|l_i \cdot p_i|}{\sqrt{A_i^2 + B_i^2}}$ and $d_i' = \frac{|l_i' \cdot p_i'|}{\sqrt{A_i'^2 + B_i'^2}}$. In general, if landmark P_i is a static point, we expect the symmetric epipolar distance $\gamma_i = (d_i + d_i')/2$ to be small. We thus define a likelihood of being static for each landmark P_i as $P_i^{\gamma} = \exp(-(\gamma_i - \mu_{\gamma})^2/(2\sigma_{\gamma}^2))$, where μ_{γ} is the mean of γ_i. We then use P_i^{γ} as the static/dynamic identification prior for each landmark P_i for detecting dynamic points.

Algorithm 1 Initial Camera Pose Estimation

```
Input:
    current frame $f_c$, reference frame $f_r$, previous frame $f_l$
Output:
    camera pose of current frame $T_c$, static likelihood $P_i^\gamma$
    for each landmark $P_i$
Match features between frames $f_c$ and $f_r$
Suggest static feature points by GC-RANSAC
for each static feature point $p_i$ in $f_c$ do
    Find the corresponding 3D landmark $P_i$ in $f_r$
end for
Estimate ego-motion $T_c$ on static landmarks by PnP
Project all landmarks seen by $f_l$ to $f_c$
Estimate fundamental matrix $F$ by GC-RANSAC
for each pair of feature points $p_i$ and $p'_i$ do
    Compute the epipolar line: $l_i = F^\top p'_i$ and $l'_i = Fp_i$
    Compute the distances: $d_i$ and $d'_i$
    Compute the static/dynamic identification prior:
    $P_i^\gamma = \exp(-((d_i + d'_i)/2 - \mu_\gamma)^2/(2\sigma_\gamma^2))$
end for
return $T_c$
```

3.3. Dynamic Landmark Detection by CRF

After estimating the initial camera pose for the current frame, we now identify the 3D landmarks as static or dynamic. As shown in Fig. 3, the basis of our approach is that dynamic points tend to have more inconsistent observations than static points, especially over a long time. Here, by observation we mean the corresponding 2D feature point in the image plane as seen in a given frame. If a point's observations from multiple frames can be accurately triangulated to a single 3D landmark, we say that point's observations are consistent. Clearly, a dynamic point's observations will be less consistent than those of a static point. Furthermore, dynamic points often have larger photometric re-projection errors between the re-projected point and the corresponding 2D feature point. We also note that points in the neighborhood of a static or dynamic point also tend to be static or dynamic, respectively. This key observation motivates us to use an observationally consistent conditional random field (OC-CRF) for dynamic point detection.

Conditional random fields (CRFs) are undirected graph models used for multi-class data segmentation and labeling [23], with unary potentials on individual nodes and pairwise potentials on adjacent nodes. In this paper, we construct a fully connected graph [22] linking each pair of 3D landmarks. For each node P_i, we assign a label $x_i = L_i \in \{0, 1\}$ where 0 represents a static point and 1 a dynamic point. By minimizing the Gibbs energy E, we obtain the optimum label for every 3D landmark:

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

We design the unary potential $\psi_u(x_i)$ and pairwise potential $\psi_p(x_i, x_j)$ to incorporate static/dynamic information from the long-term observations, which is why we call our CRF *observationally consistent* (OC-CRF).

The unary potential is defined as follows. During SLAM processing, each landmark can be seen in several key-frames. We record the corresponding 2D observations $o_j^i \in R^2$, i.e. the 2D position in key-frame j for each 3D landmark P_i. Specifically, the photometric re-projection error e_j^i between P_i and o_j^i is calculated. By averaging the re-projection errors we obtain $\alpha_i = (\sum_j e_j^i)/\beta_i$ where β_i is the total number of observations of P_i. As for the static likelihood prior P_i^γ for the landmark P_i, we define a second static likelihood from all the observations: $P_i^\beta = \exp(-(\beta_i - \mu_\beta)^2/(2\sigma_\beta^2))$, and a third one from the average re-projection error: $P_i^\alpha = \exp(-(\alpha_i - \mu_\alpha)^2/(2\sigma_\alpha^2))$, where $\mu_.$ and $\sigma_.$ and represent mean and standard deviation of respective quantities.

For each landmark, we thus have three different estimates of the likelihood that the landmark P_i is static: P_i^α, P_i^β and P_i^γ. We compute a weighted average of these estimates to give an overall likelihood that P_i is static: $P_i^s = \lambda_1 P_i^\alpha + \lambda_2 P_i^\beta + \lambda_3 P_i^\gamma$, where $\lambda_1 + \lambda_2 + \lambda_3 = 1$. If P_i^s exceeds a given threshold t, then P_i is initially labeled as static, and associated with a static confidence c; otherwise, it is labeled as dynamic, with the static confidence as $1 - c$. In our implementation, we set $\lambda_1 = \lambda_2 = \lambda_3 = 1/3$, $t = 0.3$ and $c = 0.7$. Following [29], the unary potential is then defined as:

$$\psi_u(x_i) = \left\{ \begin{array}{ll} -\log(c)I(P_i > t) & \text{if } \; x_i = 0 \\ -\log(1-c)I(P_i > t) & \text{if } \; x_i = 1 \end{array} \right. , \tag{4}$$

where $I(\cdot)$ is the indicator function.

The pairwise potential aims to encourage similar kinds of landmarks to have similar labels. The pairwise potential is the sum of two Gaussian kernels, as follows:

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_m \omega^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j), \tag{5}$$

where $\mu(x_i, x_j) = 1_{[x_i \neq x_j]}$ is a simple Potts model, $\mathbf{f}_i$ and $\mathbf{f}_j$ are feature vectors for nodes i and j, and each $k^{(m)}(\mathbf{f}_i, \mathbf{f}_j)$ is a Gaussian kernel.

The two Gaussian kernels used are an *observation kernel* and a *location kernel*. They are defined in terms of average re-projection error α_i, total number of observations β_i, the 3D location of landmark P_i, and the 2D location of p_i. For landmarks P_i and P_j with different labels, we expect there to be significant differences in the attributes mentioned above, so the pairwise potential of $\psi_p(x_i, x_j)$ should be assigned a low value, leaving the labels x_i and x_j more likely to be unchanged. However, if P_i and P_j have similar attributes, $\psi_p(x_i, x_j)$ should be assigned a high value, as P_i and P_j are more likely to belong to the same class.

The *observation kernel* is based on the idea that landmarks with a similar number of observations and average re-projection errors are likely to be in the same class. A dynamic landmark can be seen in the same position only for a few key-frames, while a static landmark can be seen in many more key-frames over a longer-term. Similarly, static landmarks have lower average re-projection errors than dynamic landmarks. Thus, landmarks with different labels should have apparent differences in the number of observations and average re-projection error, so the observation kernel is defined as:

$$k^{(1)}(\mathbf{f}_i, \mathbf{f}_j) = \exp(-\frac{|\alpha_i - \alpha_j|^2}{2\sigma_\alpha^2} - \frac{|\beta_i - \beta_j|^2}{2\sigma_\beta^2}). \quad (6)$$

The *location kernel* is based on the idea that nearby landmarks are likely to be in the same class, belonging to a compact object which is either static (e.g. a table) or dynamic (e.g. a person). Thus the location kernel penalizes pairs of landmarks with different labels but close to each other. This particularly helps to remove isolated landmarks surrounded by landmarks with the opposite label. As shown in Fig. 6(a,b), some static feature points in the person are surrounded by dynamic ones (left image), and these are re-labeled as dynamic by OC-CRF inference (right image). The location kernel function is defined as:

$$k^{(2)}(\mathbf{f}_i, \mathbf{f}_j) = \exp(-\frac{|P_i - P_j|^2}{2\sigma_P^2} - \frac{|p_i - p_j|^2}{2\sigma_p^2}). \quad (7)$$

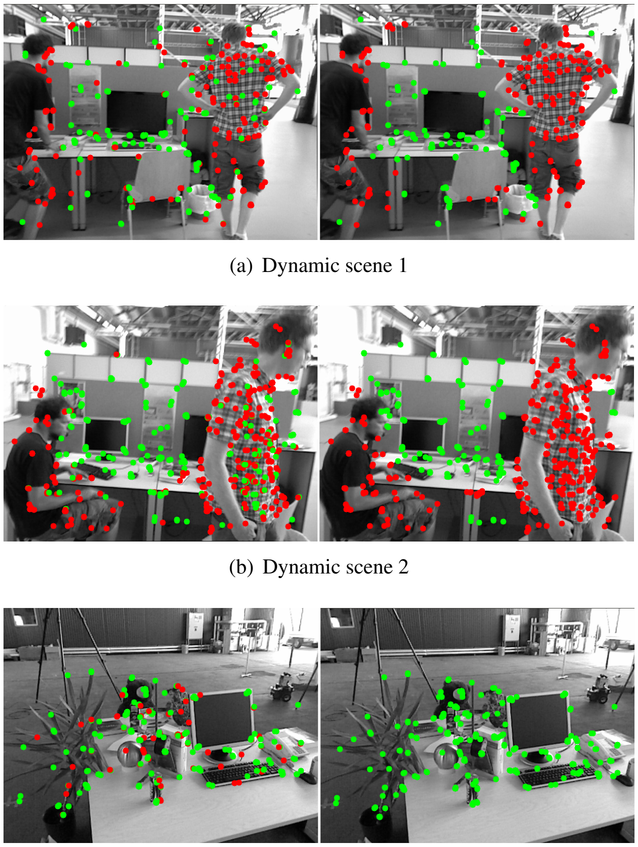
The static/dynamic labeling problem represented by our OC-CRF can be solved efficiently using a mean field approximation method [22]. We show several examples illustrating our landmark labeling results for sequences from the TUM RGB-D benchmark in Fig. 6. As can be seen, our method significantly improves the results for static/dynamic point labeling. Dynamic landmarks are segmented accurately even for highly dynamic scenes. See the supplementary video for further results.

After dynamic landmark detection, we discard dynamic landmarks and use the remaining static ones to redetermine the camera pose of the current frame more accurately. These steps are summarized in Algorithm 2.

4. Experiments

4.1. Preliminaries

To evaluate the accuracy of estimated camera pose, we tested our method on the public TUM RGB-D dynamic dataset [37] , where we selected 6 different indoor dynamic



(a) Dynamic scene 1

(b) Dynamic scene 2

(c) Static scene

Figure 6. Dynamic landmark detection in (a,b) dynamic scenes and (c) a static scene. Left: initial static/dynamic labeling. Green: static points ($p_i^s \geq t$). Red: dynamic points ($p_i^s < t$. Right: final dynamic 3D landmark detection results after OC-CRF optimization.

Algorithm 2 Dynamic Point Detection and Accurate Pose Estimation

```
Input:
    landmarks seen by the current frame $f_c$
Output:
    accurate camera pose of the current frame $T_c^*$
Initialize CRF graph
for Each landmark do
    Compute the likelihood: $P_i^s = (P_i^\alpha + P_i^\beta + P_i^\gamma)/3$
    Compute unary potentials from Eq. (4)
end for
for Each pair of landmarks do
    Compute pairwise potentials from Eqs. (6, 7)
end for
Determine dynamic landmarks by CRF inference
Estimate pose $T_c^*$ from static landmarks
Return $T_c^*$
```

Table 1. Absolute trajectory error for dynamic datasets for the ORB-SLAM method and our OC-CRF SLAM method, measured in metres.

Sequence	**ORB-SLAM ATE (m)**				**OC-CRF SLAM ATE (m)**			
	RMSE	**Std**	**Mean**	**Median**	**RMSE**	**Std**	**Mean**	**Median**
fr3/walking-xyz	0.366287	0.255601	0.262363	0.162223	**0.018335**	**0.008711**	**0.015775**	**0.013733**
fr3/walking-halfsphere	0.382438	0.187725	0.333194	0.318089	**0.029800**	**0.014495**	**0.022781**	**0.022489**
fr3/walking-static	0.214124	0.083655	0.197106	0.175119	**0.010446**	**0.006624**	**0.015890**	**0.010950**
fr3/walking-rpy	0.744576	0.401184	0.627252	0.603298	**0.114289**	**0.084301**	**0.077172**	**0.051715**
fr3/sitting-xyz	0.010889	0.005032	0.009656	0.008829	**0.009333**	**0.004939**	**0.007922**	**0.007235**
fr2/desk-with-person	0.074397	**0.016205**	0.072610	0.073081	**0.071795**	0.016229	**0.069937**	**0.070072**

Table 2. Relative pose error for dynamic datasets for the original ORB-SLAM and our OC-CRF SLAM, in m/s or °/s as appropriate.

Sequence	**ORB-SLAM RPE**				**OC-CRF SLAM RPE**			
	t.RMSE	**t.Std**	**r.RMSE**	**r.Std**	**t.RMSE**	**t.Std**	**r.RMSE**	**r.Std**
fr3/walking-xyz	0.517820	0.387784	8.958071	7.274051	**0.025704**	**0.011565**	**0.645109**	**0.375732**
fr3/walking-halfsphere	0.570142	0.316124	9.254973	7.292348	**0.039497**	**0.020618**	**0.815216**	**0.352377**
fr3/walking-static	0.311129	0.211853	5.509479	3.687236	**0.016719**	**0.009602**	**0.402316**	**0.158952**
fr3/walking-rpy	1.093185	0.632268	9.554797	9.790038	**0.164724**	**0.119515**	**3.050945**	**2.183991**
fr3/sitting-xyz	0.015945	0.007168	0.598171	**0.300411**	**0.013663**	**0.006734**	**0.579074**	0.309012
fr2/desk-with-person	0.104156	0.052091	0.722745	0.328341	**0.100364**	**0.049486**	**0.714301**	**0.315173**

sequences with moving people and violent camera shaking for evaluation. We calculated the absolute trajectory error (ATE) and relative pose error (RPE), as defined in [37], between the camera poses estimated by our method and the ground truth.

We also compared our method with the original ORB-SLAM2, which does not have dynamic point detection, to evaluate the improvement that our dynamic point detection module makes in a dynamic environment. We further compared our proposed OC-CRF approach with other related dynamic SLAM methods: DVO [18], BaMVO [19] and static point weighting (SPW) [24]. Besides, we propose a dynamic reconstruction using the pose estimated by our proposed approach. All experiments were performed on a desktop computer with a 3.6 GHz Intel Core i9-9900K CPU and 16 GB RAM, without GPU acceleration.

Thereafter, we give an extensive study to justify the parameters used in our OC-CRF SLAM system. We further quantitatively evaluate the influence of proportion of dynamic objects in a scene on the accuracy of pose estimation provided by our OC-CRF SLAM system. Finally, we discuss the benefit of our approach of using long-term consistent observations and consider limitations and feasible future solutions.

4.2. Comparison with unmodified ORB-SLAM

We first evaluate the performance for our dynamic camera tracking compared with the original ORB-SLAM. We tested our method on the six dynamic sequences from the TMU RGB-D dataset, and compared the resulting ATE and RPE with those of the original ORB-SLAM.

The comparison of ATE is shown in Table 1, where 'RMSE' means the root mean squared error of ATE and 'Std' means the standard deviation of ATE. For highly dynamic sequences (those whose names begin with 'walking', i.e. fast moving persons or camera), our proposed method achieves significantly lower RMSE, Std, Mean and Median than unmodified ORB-SLAM. In the last two scenarios 'sitting-xyz' and 'desk-with-person' with less dynamic environments, our algorithm also achieves slightly better results.

The ATE between estimated trajectories and ground-truth is visualized in Fig 7. As can be seen clearly, the trajectories estimated by our OC-CRF SLAM (middle row) are much closer to the real trajectories than those of the unmodified ORB-SLAM (top row).

RPE is compared in Table 2; 't' and 'r' denote translational and rotational error. For all highly dynamic RGB-D sequences, our method has significantly lower RMSE and Std. However, for less dynamic environments, some static landmarks may be detected as dynamic landmarks, affecting subsequent pose estimation. Nevertheless, our method is still better than or very close to ORB-SLAM in such cases.

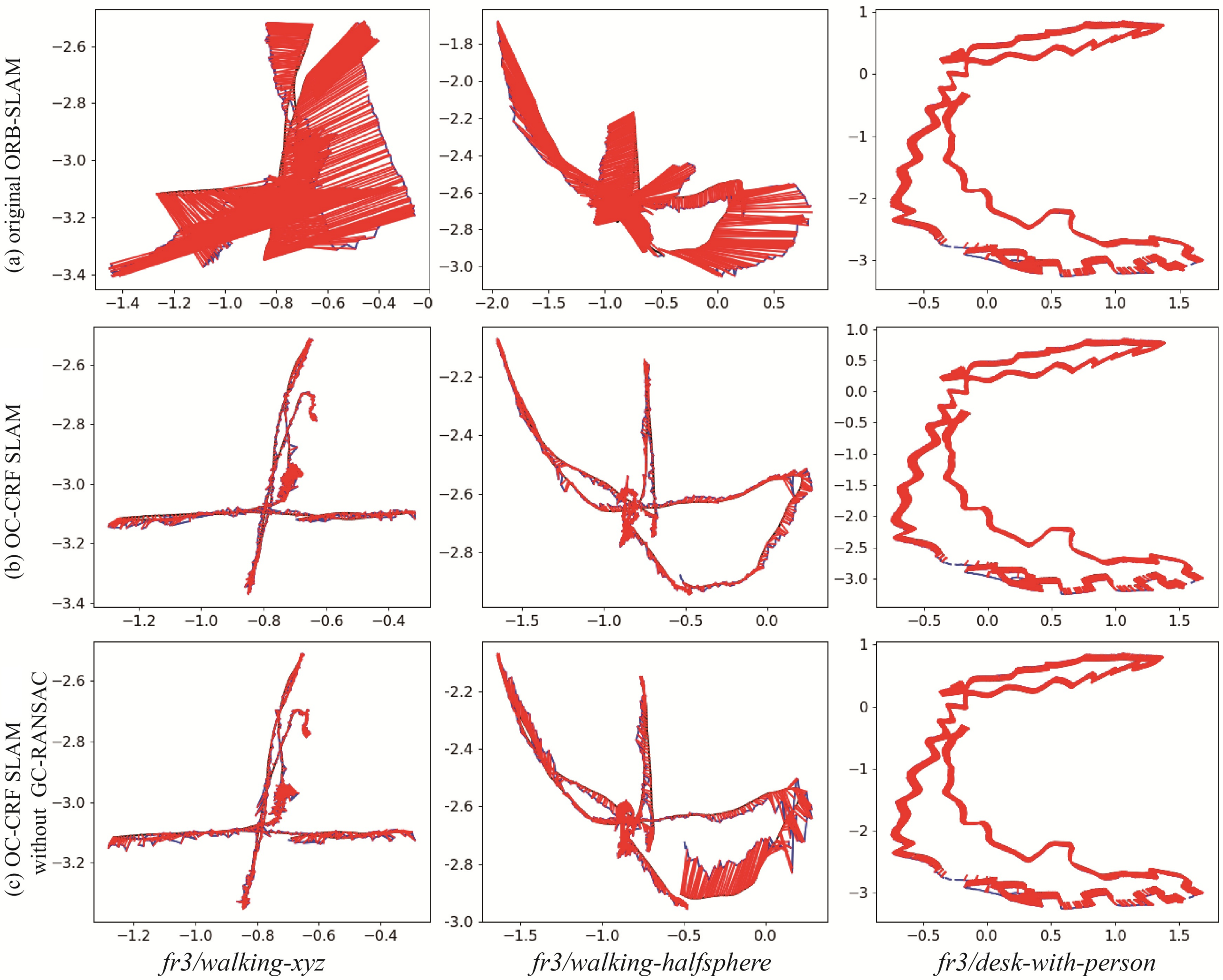


Figure 7. Visualization of ATE on *fr3/walking* (left), *fr3/desktop-person* (center) and *fr3/desktop-person* (right). Blue: estimated trajectories. Black: ground-truth trajectories. Red lines connect corresponding points in these two trajectories: their length indicates the estimation error. Top: trajectories from unmodified ORB-SLAM. Center: trajectories from OC-CRF SLAM. Bottom: trajectories from OC-CRF SLAM without GC-RANSAC. Clearly, OC-CRF SLAM generates more accurate camera trajectories than unmodified ORB-SLAM. OC-CRF using CRF alone has somewhat lower accuracy than OC-CRF SLAM with GC-RANSAC.

4.3. Effectiveness of GC-RANSAC Filter

We also evaluated the performance of the initial camera pose estimation using the GC-RANSAC filter from Sec. 3.2. We built a SLAM system without the initial camera pose estimation component by just assigning an initial camera pose using velocity prediction as ORB-SLAM does. Consequently, the unary and pairwise potentials also do not contain the initial static/dynamic priors for the OC-CRF for the dynamic landmark detection. We compared such a system (without the GC-RANSAC filter) with our full OC-CRF SLAM system by evaluating the ATE and RPE of the six dynamic sequences of the TUM RGB-D dataset.

Table 3 shows ATE results on our OC-CRF SLAM with and without the GC-RANSAC filter. Without the GC-RANSAC filter, the ATEs are significantly greater for highly dynamic sequences such as *fr3/walking-xyz* and *fr3/walking-halfsphere*. For less dynamic sequences, the ATEs are slightly increased. In Fig. 7, the bottom row shows the ATE generated by OC-CRF SLAM without GC-RANSAC, which has larger errors than the middle row with GC-RANSAC, verifying the usefulness of GC-RANSAC.

4.4. Comparison with existing methods

We compared our proposed OC-CRF SLAM method with other state-of-the-art RGB-D SLAM systems: BaMVO, dense visual odometry (DVO), and static point weighting (SPW). Table 4 gives the corresponding ATE

Table 3. Absolute trajectory error (m) of OC-CRF SLAM with and without GC-RANSAC.

Sequence	GC(w)	GC(w/o)
fr3/walking-xyz	**0.018335**	0.027677
fr3/walking-halfsphere	**0.029800**	0.057692
fr3/walking-static	**0.010446**	0.016060
fr3/walking-rpy	**0.114289**	0.100444
fr3/sitting-xyz	**0.009333**	0.024205
fr2/desk-with-person	0.071795	**0.065325**

Table 4. Absolute trajectory error (m) for dynamic datasets, for DVO, SPW and our OC-CRF SLAM methods.

Sequence	DVO	SPW	OC-CRF
fr3/walking-xyz	0.0932	0.0601	**0.0183**
fr3/walking-halfsphere	0.0470	0.0432	**0.0298**
fr3/walking-static	0.0656	0.0261	**0.0104**
fr3/walking-rpy	0.1333	0.1791	**0.1142**
fr3/sitting-xyz	0.0482	0.0397	**0.0093**
fr2/desk-with-person	0.0596	**0.0484**	0.0717

results (BaMVO's results are missing since they were not provided in the corresponding paper). We can see that for all of the highly dynamic datasets, our system outperforms the others, often by a considerable margin. The only case in which our method performs poorly is the sequence *fr2/desk-with-person*. This is an almost static scene, and a few static landmarks are labeled as dynamic by the GC-RANSAC filter with its standard parameter settings, which degrades the accuracy of the initial pose estimate.

Table 5 shows relative pose error results for these methods. For RMSE of rotational drift, our method performs better than all other methods. For RMSE of translational drift, our method also achieves more accurate results for almost all datasets. Specifically, in the best cases, our method has RPE errors which are less than 1/3 of those of the state-of-the-art method, while our method performs worse on the *fr2/desk-with-person* sequence in terms of translation drift, again for the reason mentioned above. This verifies that OC-CRF SLAM effectively reduces the influence of dynamic objects, especially for highly dynamic scenes.

4.5. Dynamic Dense Reconstruction

Since our dynamic RGB-D SLAM not only calculate the camera position but also identify the static/dynamic 3D landmarks information. To further evaluate the accuracy of our method in camera position tracking and the benefit of static/dynamic 3D landmark detection, we propose a simple dense reconstruction based on our dynamic RGB-D SLAM. Specifically, similar to MaskFusion [34], we recognise the dynamic regions, i.e., people, using the mask predicted by Mask R-CNN [15] as well as the dynamic points determined by the registering errors between the current frame and the previous one; finally we fuse the remaining static points together using the camera poses tracked by our proposed RGB-D SLAM.

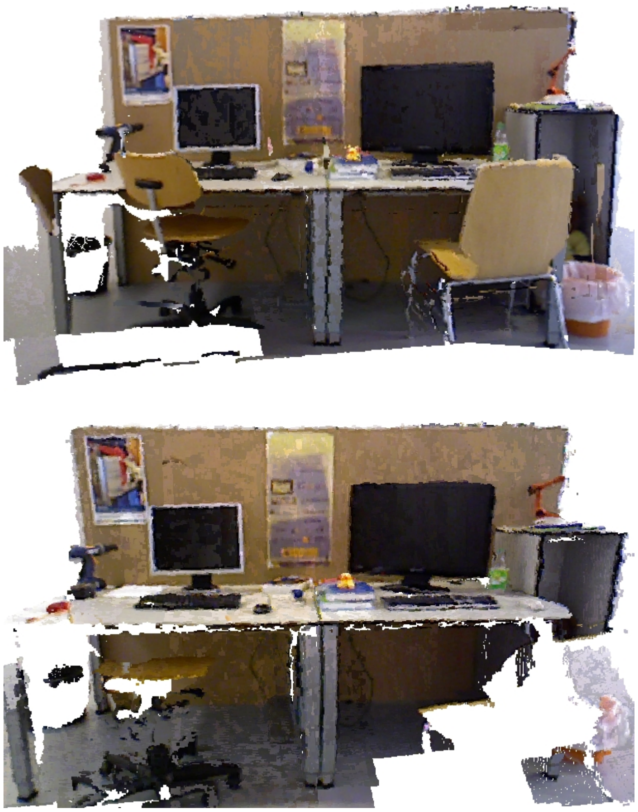


Figure 8. Reconstructed point clouds for two scenes (top: fr3/walking-xyz, bottom: fr3/walking-static) from TUM RGB-D dataset.

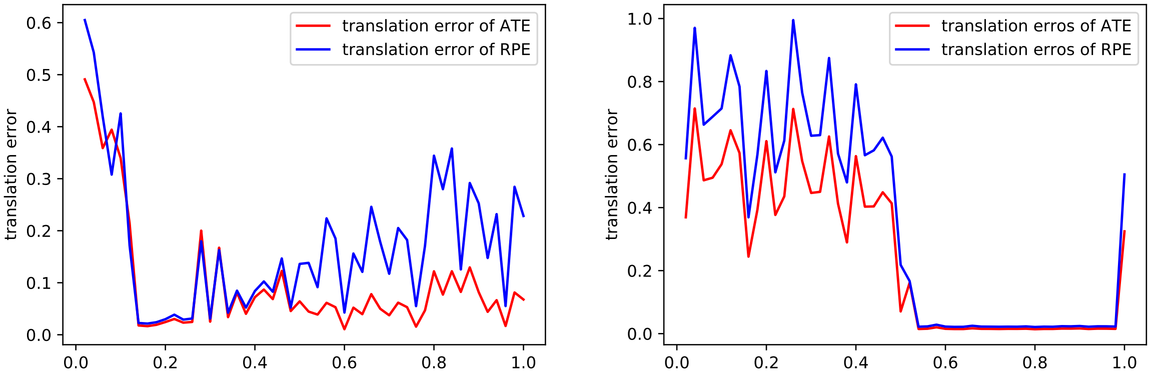
We compare our approach with StaticFusion(SF) [35], MaskFusion(MF) [34], and ReFusion(RF) [31] on four sequences of TUM RGB-D dynamic dataset. Results are obtained by running the available open source implementations for different methods or from their original published papers. Table 6 shows the ATE error for each sequence. We also show example dense reconstruction results in Fig. 1 and 8. As can be clearly seen, dynamic regions, i.e. moving people, are effectively removed from the reconstruction scenes. These results demonstrate that our method achieves more accurate camera pose with good reconstruction quality for dynamic scenes.

4.6. Parameter Study

The main parameters in our OC-CRF SLAM are those in the unary and pairwise potential computations in dynamic landmark detection. We performed an extensive study of these parameters to justify the chosen settings.

Table 5. Relative pose error for DVO, BaMVO, SPW and our OC-CRF SLAM methods.

Sequence	**RMSE of translational drift (m/s)**				**RMSE of rotational drift (°/s)**			
	DVO	**BaMVO**	**SPW**	**our OC-CRF**	**DVO**	**BaMVO**	**SPW**	**our OC-CRF**
fr3/walking-xyz	0.4360	0.2326	0.0651	**0.0257**	7.6669	4.3911	1.6442	**0.6451**
fr3/walking-halfsphere	0.2628	0.1738	0.0527	**0.0394**	5.2179	4.2863	2.4048	**0.8152**
fr3/walking-static	0.3818	0.1339	0.0327	**0.0167**	6.3502	2.0833	0.8085	**0.4023**
fr3/walking-rpy	0.4038	0.3584	0.2252	**0.1647**	7.0662	6.3398	5.6902	**3.0509**
fr3/sitting-xyz	0.0453	0.0482	0.0219	**0.0136**	1.4980	1.3885	0.8446	**0.5790**
fr2/desk-with-person	0.0296	0.0299	**0.0173**	0.1003	1.3920	1.1167	0.7266	**0.3151**



(a) evaluation of threshold t

(b) evaluation of confidence c

Figure 9. Variation in translation error with changing thresholds t and confidence c. x-axis: threshold in (a) and confidence in (b), y-axis: translation error. Red: ATE translation error, Blue: RPE translation error.

Table 6. Absolute trajectory error (m) on TUM RGB-D dynamic datasets, for StaticFusion(SF), MaskFusion(MF), ReFusion(RF) and ours.

Sequence	**SF**	**MF**	**RF**	**ours**
fr3/sitting-xyz	0.039	0.031	0.040	**0.009**
fr3/walking-static	0.015	0.035	0.017	**0.010**
fr3/walking-xyz	0.093	0.104	0.099	**0.018**
fr3/walking-halfsphere	0.681	0.106	0.104	**0.030**

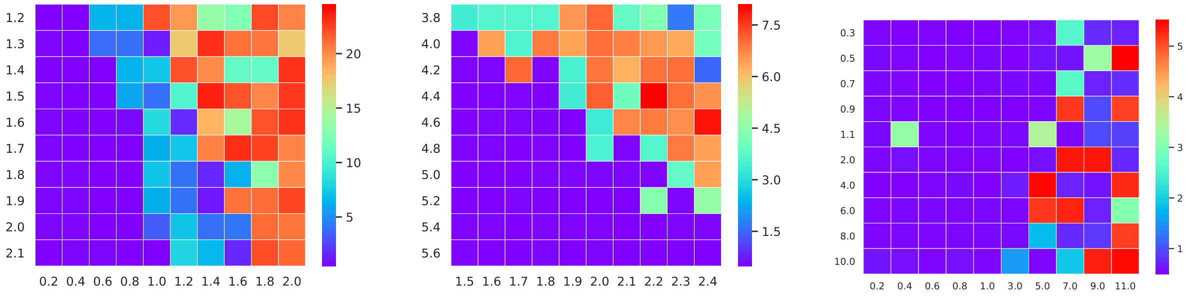
4.6.1 Unary Potential Parameters

The threshold t and confidence c parameters control the unary potential, as described in Equation 4. To justify their settings, we tested OC-CRF SLAM on the TUM dynamic dataset with varying values of threshold t and confidence c, and computed the ATE and RPE accuracy. See Fig. 9, which shows average ATE and RPE accuracy when varying threshold t and confidence c in $(0.0, 1.0)$. Using this information, we set $t = 0.3$ and $c = 0.7$ to ensure unary potential computation leads to relatively small errors in both ATE and RPE.

The Gaussian kernel parameters, $\{\mu_\alpha, \sigma_\alpha\}$, $\{\mu_\beta, \sigma_\beta\}$ and $\{\mu_\gamma, \sigma_\gamma\}$ for the three likelihoods $\{P^\alpha, P^\beta, P^\gamma\}$ in unary potential computation were also tuned. Fig. 10 shows RPE rotation errors for different values of $\{\mu_\alpha, \sigma_\alpha\}$, $\{\mu_\beta, \sigma_\beta\}$ and $\{\mu_\gamma, \sigma_\gamma\}$, respectively. Using this information, we set $\{\mu_\alpha = 1.51, \sigma_\alpha = 0.6\}$, $\{\mu_\beta = 4.81, \sigma_\beta = 1.86\}$ and $\{\mu_\gamma = 0.3, \sigma_\gamma = 0.2\}$ to give low-drift RPE accuracy.

4.6.2 Pairwise Potential Parameters

The observation kernel parameters $\{\sigma_\alpha, \sigma_\beta\}$, location kernel parameters $\{\sigma_P, \sigma_p\}$ and balance weight parameters $\{w^1, w^2\}$ control the pairwise potential computation. We also evaluated ATE accuracy of OC-CRF SLAM on the TUM dynamic dataset when varying these parameters. See Fig. 11 shows ATE translation error variation with (a) observation kernel parameters $\{\sigma_\alpha, \sigma_\beta\}$, (b) location kernel parameters $\{\sigma_P, \sigma_p\}$ and (c) weights $\{w^1, w^2\}$. Using this information, to ensure low-drift ATE accuracy in pairwise kernel computation, we set $\{\sigma_\alpha = 0.6, \sigma_\beta = 1.86\}$ and $\{\sigma_P = 0.5, \sigma_p = 20\}$ for observation and location kernels, and $\{w^1 = 10, w^2 = 30\}$ to balance these kernels.

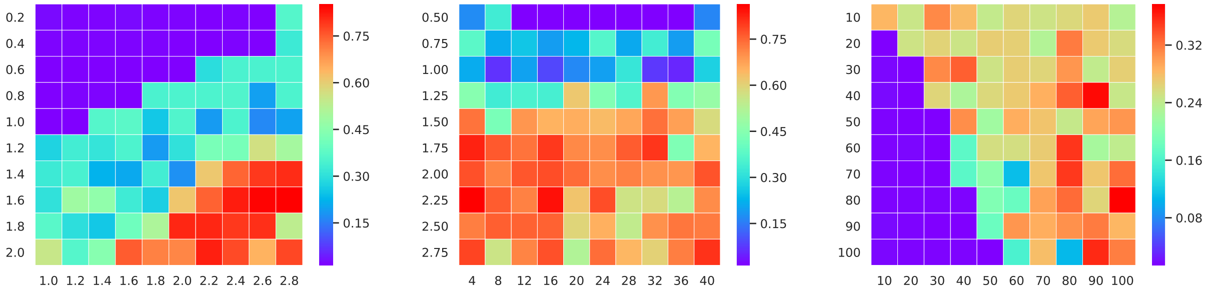


(a) evaluation of μ_α and σ_α

(b) evaluation of μ_β and σ_β

(c) evaluation of μ_γ and σ_γ

Figure 10. RPE rotation errors (in °/s) for different settings of P^α, P^β and P^γ. x-axis: standard deviation. y-axis: mean value. Red represents higher error, blue lower.



(a) evaluation of the observation kernel

(b) evaluation of the location kernel

(c) evaluation of the balance factors

Figure 11. Parameter evaluation for observation kernel, location kernel and weights respectively. (a) x- and y-axis show σ_α and σ_β, respectively; (b) x- and y-axis show σ_p and σ_P, respectively; (c) x- and y-axis show w^1 and w^2, respectively.

4.7. Impact of dynamic objects

Clearly, the accuracy of camera pose estimation for a dynamic scene will be affected by the presence of human beings and other moving objects. To quantitatively evaluate the impact of dynamic objects on the accuracy of pose estimation, we analyzed the relationship between the proportion of dynamic content in the scene and camera pose estimation error using OC-CRF SLAM, by computing the ATE and RPE for each frame with respect to the ratio of dynamic objects, using the TUM dynamic dataset. Here we define the ratio of dynamic objects to be $r(k) = n_d(k)/n(k)$, where $n_d(k)$ denotes the number of dynamic feature points in frame f_k and $n(k)$ is the total number of feature points in that frame.

Fig. 9 shows ATE and RPE translation error variation with dynamic ratio. These errors become larger with increasing dynamic ratio, so as expected, camera pose estimates provided by our approach become worse with increasingly dynamic scenes. Our approach can handle small to medium amounts of dynamic content (up to about $< 50\%$) if we wish to keep the ATE and RPE translation error to under about 0.2m and 0.2 m/s, respectively.

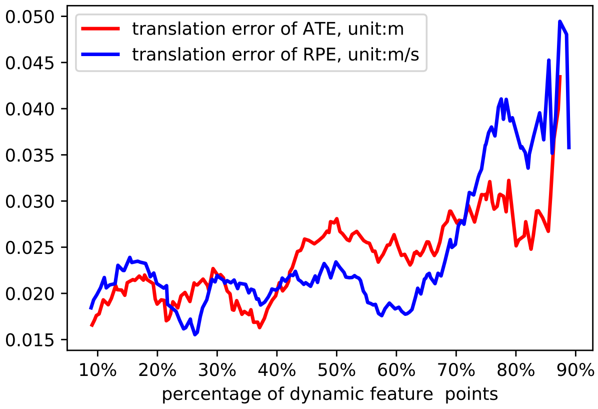


Figure 12. Variation in ATE and RPE translation error with differing proportions of dynamic content. x-axis: percentage of dynamic feature points compared to all feature points. y-axis: Red: ATE translation error. Blue: RPE translation error.

4.8. Discussion and Limitations

One of the main benefits of our approach comes from the unary and pairwise potentials used in dynamic landmark de-

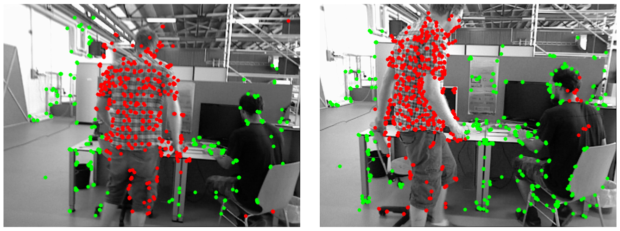

(a) Failure case 1 (b) Failure case 2

Figure 13. Typical failure case of our method, both in frame 699 (a) and frame 727 (b) of the sequence of *fr3/walking-xyz*, the person in back sits still without moving for a long time, almost all landmarks in this person are labeled as static. In sharp contrast, the landmarks in the moving person on the left are labeled as dynamic accurately.

tection, which leverages information from widely separated frames, not just consecutive frames. The static likelihood is estimated for every landmark for every frame (see Section 3.3) during the whole video sequence, which implicitly encodes long-term consistency information in the unary potential computation. Also, the observation kernel used in pairwise potential computation leverages the total number of observations, again providing a feature across long-term spans of frames.

Our approach suffers from three main drawbacks: Firstly, it is not as effective for almost static scenes, mainly because it may wrongly label static feature points as dynamic, decreasing camera pose estimation accuracy accordingly. One possible solution is to allow the user to choose whether to use the dynamic detection module. If it is turned off, the final pose estimate is mainly determined by the process of initial camera pose estimation.

Secondly, as shown in Fig. 13, our approach does not perform very well for long time stationary objects which then start to move, since our approach mainly relies on geometric rules to identify static/dynamic feature points without understanding the scene. This could be overcome by temporally matching object arrangements (including object locations and spatial relationships) for the whole scene to infer when previously static objects start to move [14].

Thirdly, initial ego-motion estimation depends on GC-RANSAC, a randomized algorithm. Thus the final result of dynamic landmark detection is inherently somewhat random. Nevertheless, our method is still typically superior to many existing methods. We hope to explore non-random initial ego-motion estimation methods to make ensure that the system robustly works on various scenarios.

5. Conclusion

In this paper, we have presented the OC-CRF SLAM system for accurate pose estimation and effective dynamic point detection. To reduce the impact of dynamic points on pose estimation, we firstly compute an initial pose using GC-RANSAC and assign each landmark a static/dynamic prior. Then, we use a CRF with appropriate unary and pairwise potentials to label each landmark as static or dynamic. We have shown that our proposed OC-CRF SLAM is significantly more accurate than existing methods for the highly dynamic examples in the public TUM RGB-D dataset and can be incorporated into the dynamic 3D reconstruction. In the future, we would explore potential AR/VR applications for dynamic scenarios, taking advantage of the static/dynamic information identified by our proposed lightweight camera pose tracking.

Acknowledgements

We thank the reviewers for their detailed comments on this paper. This work was supported by the Natural Science Foundation of China (Grant No.: 61521002, 61863031, 61902210) and the China Postdoctoral Science Foundation (Grant No.: 2019M660646).

References

[1] P. F. Alcantarilla, J. J. Yebes, J. Almazán, and L. M. Bergasa. On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1290–1297, 2012. 1, 3

[2] A. Bapat, E. Dunn, and J. Frahm. Towards kilo-hertz 6-dof visual tracking using an egocentric cluster of rolling shutter cameras. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 22(11):2358–2367, 2016. 3

[3] D. Barath and J. Matas. Graph-cut ransac. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6733–6741, 2018. 2

[4] Y. Boykov and G. Funkalea. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision (IJCV)*, 70(2):109–131, 2006. 5

[5] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 105–112, 2001. 5

[6] M. Bujanca, M. Luján, and B. Lennox. Fullfusion: A framework for semantic reconstruction of dynamic scenes. In *The IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2019. 3

[7] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics (TRO)*, 32(6):1309–1332, 2016. 1, 2

[8] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(6):1052–1067, 2007. 2

[9] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics Automation Magazine (RAM)*, 13(2):99–110, 2006. 1

[10] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 40(3):611–625, 2018. 2

[11] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: large-scale direct monocular SLAM. In *13th European Conference on Computer Vision (ECCV)*, pages 834–849, 2014. 2

[12] C. Forster, M. Pizzoli, and D. Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, 2014. 2

[13] M. Garon and J. Lalonde. Deep 6-dof tracking. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 23(11):2410–2418, 2017. 3

[14] M. Halber, Y. Shi, K. Xu, and T. Funkhouser. Rescan: Inductive instance segmentation for indoor rgbd scans. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 2541–2550, 2019. 13

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 42(2):386–397, 2020. 10

[16] J. Huang, S. Yang, Z. Zhao, Y.-K. Lai, and S.-M. Hu. Clusterslam: A slam backend for simultaneous rigid body clustering and motion estimation. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 5875–5884, 2019. 3

[17] O. Kähler, V. Adrian Prisacariu, C. Yuheng Ren, X. Sun, P. Torr, and D. Murray. Very high frame rate volumetric integration of depth images on mobile devices. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 21(11):1241–1250, 2015. 2

[18] C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for RGB-D cameras. In *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3748–3754, 2013. 3, 8

[19] D. Kim and J. Kim. Effective background model-based rgb-d dense visual odometry in a dynamic environment. *IEEE Transactions on Robotics (TRO)*, 32(6):1565–1573, 2016. 1, 2, 3, 8

[20] K. Kim, M. Billinghurst, G. Bruder, H. B. Duh, and G. F. Welch. Revisiting trends in augmented reality research: A review of the 2nd decade of ismar (2008–2017). *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 24(11):2947–2962, 2018. 1

[21] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 225–234, 2007. 2

[22] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *International Conference on Neural Information Processing Systems (NIPS)*, pages 109–117, 2011. 6, 7

[23] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *18th International Conference on Machine Learning (ICML)*, pages 282–289, 2001. 6

[24] S. Li and D. Lee. Rgb-d slam in dynamic environments using static point weighting. *IEEE Robotics and Automation Letters (RA-L)*, 2(4):2263–2270, 2017. 1, 2, 3, 8

[25] Long Quan and Zhongdan Lan. Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 21(8):774–780, 1999. 4

[26] S. Meerits, D. Thomas, V. Nozick, and H. Saito. Fusionmls: Highly dynamic 3d reconstruction with consumer-grade rgb-d cameras. *Computational Visual Media*, 4(4):287–303, 2018. 1

[27] D. Moratuwage, B. Vo, and D. Wang. Collaborative multi-vehicle slam with moving object tracking. In *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5702–5708, 2013. 1

[28] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics (TRO)*, 33(5):1255–1262, 2017. 2, 3

[29] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. In *International Conference on Intelligent Robots and Systems (IROS)*, 2019. 6

[30] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *2011 International Conference on Computer Vision (ICCV)*, pages 2320–2327, 2011. 2

[31] E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss. Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019. 2, 10

[32] J. R. Rambach, A. Tewari, A. Pagani, and D. Stricker. Learning to fuse: A deep learning approach to visual-inertial camera pose estimation. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 71–76, 2016. 3

[33] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 39(6):1137–1149, 2017. 3

[34] M. Runz, M. Buffier, and L. Agapito. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 10–20, 2018. 2, 10

[35] R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon, and D. Cremers. Staticfusion: Background reconstruction for dense rgb-d slam in dynamic environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3849–3856, 2018. 2, 10

[36] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 573–580, 2012. 2

[37] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 573–580, 2012. 7, 8

[38] C. C. Wang, C. Thorpe, M. Hebert, S. Thrun, and H. Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. *International Journal of Robotics Research (IJRR)*, 26(9):889–916, 2007. 1, 3

[39] Z. Yan, M. Ye, and L. Ren. Dense visual slam with probabilistic surfel map. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 23(11):2389–2398, 2017. 2

[40] S. Yang, B. Li, M. Liu, Y.-K. Lai, L. Kobbelt, and S.-M. Hu. Heterofusion: Dense scene reconstruction integrating multi-sensors. *IEEE Transactions on Visualization Computer Graphics (TVCG)*, 2020 to appear. 3

[41] S. Yang, J. Wang, G. Wang, X. Hu, M. Zhou, and Q. Liao. Robust rgb-d slam in dynamic environment using faster r-cnn. In *3rd IEEE International Conference on Computer and Communications (ICCC)*, pages 2398–2402, 2017. 3

[42] H. Zhang and F. Xu. Mixedfusion: Real-time reconstruction of an indoor scene with dynamic objects. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 24(12):3137–3146, 2018. 2

[43] J. Zhang, M. Gui, Q. Wang, R. Liu, J. Xu, and S. Chen. Hierarchical topic model based object association for semantic SLAM. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 25(11):3052–3062, 2019. 3

[44] F. Zhong, S. Wang, Z. Zhang, C. Chen, and Y. Wang. Detect-slam: Making object detection and slam mutually beneficial. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1001–1010, 2018. 3