Accurate Dynamic SLAM Using CRF-Based Long-Term Consistency

Zheng-Jun Du, Shi-Sheng Huang, Tai-Jiang Mu, Qunhe Zhao, Ralph R. Martin, and Kun Xu, *Member, IEEE*

Abstract—Accurate camera pose estimation is essential and challenging for real world dynamic 3D reconstruction and augmented reality applications. In this article, we present a novel RGB-D SLAM approach for accurate camera pose 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 long-term consistency via conditional random fields, which leverages long-term observations from multiple frames. Specifically, we first 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 TUM and Bonn RGB-D dynamic datasets 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 dynamic 3D reconstruction can benefit from the camera poses estimated by our RGB-D SLAM approach.

Index Terms—RGB-D SLAM, dynamic SLAM, long-term consistency, conditional random fields,graph-cut RANSAC

1 Introduction

Accurate pose tracking in an unknown environment is a fundamental issue in 3D scene perception and understanding [1]. Visual simultaneous localization and mapping (SLAM) is a basic technique for pose tracking and 3D reconstruction; it has received intense research interest from the computer graphics, computer vision and mixed/augmented/virtual reality communities. Observed scenes often contain dynamic items such as moving people and objects, so an accurate visual SLAM method which is efficient and effective in such dynamic environments is urgently needed as a basis for various applications in augmented/virtual reality, robotics etc.

Although visual SLAM technology has made significant progress in the past few decades [2], [3], most works focus on static environments, and cannot estimate camera pose when faced with dynamic situations. 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 [4], [5] often utilize an RGB-D depth camera and tackle the dynamic tracking problem following the DATMO—detection and tracking of moving objects—scheme [6]. However, these DATMO-based methods suffer from drawbacks arising from assumptions made about the moving objects e.g., the number of objects is predetermined, or the objects move slowly. Dynamic detection methods using foreground/background segmentation [7], dense scene flow [8] or static/dynamic edge point weighting [9] estimate the camera pose solely from static entities by detecting and eliminating the dynamic region. However, since the determination of points or regions as static or dynamic is based on only a few consecutive frames, moving object detection in these methods is not robust, with a consequent impact on the accuracy of camera pose estimation. Recent online 3D reconstruction methods [10], [11], [12], [13] aim to reconstruct dynamic 3D scenes. However, performing static/dynamic determination with ICP-style registration [10], [12], [13] or 2D CNNs [11] is expensive both in computation and memory, so they are unsuitable for use in a light-weight system to track camera positions for online applications in mixed and augmented reality, etc.

In this paper, we provide a more accurate and light-weight dynamic visual SLAM method using an RGB-D sensor, by analyzing frames over long-term timescales rather than 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 using long-term observations rather than short-term observations. Based on this key observation, we first estimate the camera pose using an initial static/dynamic labelling from inlier/outlier determination with graph-cut (GC) RANSAC [14]. Then we build a long-term consistent conditional random field (LC-CRF) model to assist in 3D dynamic landmark detection, by analyzing observations of static and

- *Zheng-Jun Du is with the Department of Computer Technology and Application, Qinghai University, Xining 810016, China, and also with the BNRist, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: duzj19@mails.tsinghua.edu.cn.*
- *Shi-Sheng Huang, Tai-Jiang Mu and Kun Xu are with the BNRist, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: shishenghuang.net@gmail.com, {taijiang, xukun}@tsinghua.edu.cn.*
- *Qunhe Zhao is with the DeepBlue Technology, Shanghai, Co.,Ltd, Shanghai 200336, China. E-mail: zhaoqh@deepblueai.com.*
- *Ralph R. Martin is with the School of Computer Science and Informatics, Cardiff University, Cardiff CF243AA, U.K. E-mail: MartinRR@cardiff.ac.uk.*

Manuscript received 14 Feb. 2020; revised 26 Sept. 2020; accepted 28 Sept. 2020. Date of publication 1 Oct. 2020; date of current version 28 Feb. 2022. (Corresponding author: Tai-Jiang Mu.) Recommended for acceptance by P. Sander. Digital Object Identifier no. 10.1109/TVCG.2020.3028218

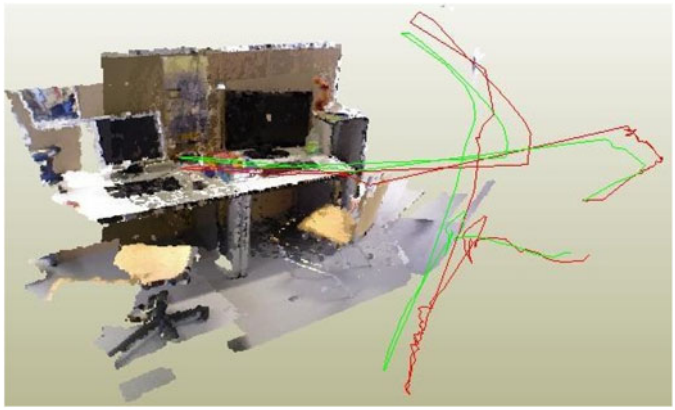


Fig. 1. The reconstructed scene for *fr3/walking-halfsphere* from the TUM RBG-D dynamic dataset. As an accurate pose tracking technique for dynamic environments, our efficient approach utilizing CRF-based long-term consistency can estimate a camera trajectory (red) close to the ground truth (green).

dynamic landmarks over a long-term series of consecutive frames. Solving the static-dynamic labeling problem with the aid of the CRF provides highly accurate dynamic detection results. Using the results to eliminate the dynamic 3D landmarks, we can estimate the camera pose with much higher precision from the remaining static 3D landmarks.

Our LC-CRF based dynamic 3D landmark detection method is efficient, leading to a light-weight SLAM system for accurate 3D position tracking in dynamic environments. We have evaluated our approach on two public datasets: (i) the TUM RGB-D dynamic dataset [15] and (ii) the Bonn RGB-D dynamic dataset [13]; the latter has highly dynamic sequences. The results show that our approach typically outperforms state-of-the-art approaches, such as BaMVO [7] and SPW [9]. We also propose a dynamic 3D scene reconstruction method using our approach, which can provide good scene reconstruction quality, and more accurate camera position tracking results than other fusion-based dynamic reconstruction methods, e.g., MaskFusion [11]. In summary, this paper makes the following contributions:

1) A reliable dynamic 3D landmark detection method based on a long-term consistent conditional random field, which constitutes the main component of our 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 works particularly relevant to ours, and refer readers to [2] for a more detailed overview of the progress of the visual SLAM 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 [16] 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 [17] and ORB-SLAM2 [18], and *direct* visual SLAM methods, which directly use image intensity for camera tracking, e.g., DTAM [19], SVO [20], LSD-SLAM [21], InfiniTAM [22], PSM-SLAM [23] and DSO [24]. 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 performing camera pose tracking by fusing multiple sensors, such as multiple cameras [25], inertial-cameras [26] and laser-inertial-camera [27], or with the aid of deep learning [28], [29].

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. Unlike these methods, our approach aims to provide robust camera tracking in dynamic scenarios. Like ORB-SLAM2 [18], it contains three components. 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.* [6] in 2006 inspired many dynamic visual SLAM approaches to performing the camera pose tracking by detecting moving objects with the aid of dense scene flow [4] or object clustering [30], [31]. Kerl *et al.* [32] presented 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.* [7] 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.* [9] 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. Most of these methods detect dynamic components by analysis of only a few consecutive frames, two frames in DVO [32] and just the current frame in BaMVO [7] and Li *et al.* [9].

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 performed more reliably using long-term observations. Based on this insight, we build a long-term consistent conditional random field using feature vectors derived from multiple visual observation errors over a long period of consecutive frames.

2.3 Dynamic Reconstruction

3D reconstruction with RGB-D cameras has made much progress in past decades. Here, we only focus on 3D reconstruction methods for dynamic scenes, and refer readers to the survey paper [33] for the state-of-the-art in 3D reconstruction approaches. Human movement is an important source of dynamism in indoor scenes, and DynamicFusion [34] proposed the first dense SLAM system to reconstruct dynamic scenes with humans by accurately determining a volumetric flow field that transforms the current scene into a canonical frame. Subsequent works such as KillingFusion [35], SobolevFusion [36] extended the flow field with more accurate non-rigid motion estimation without templates or shape priors. To make the non-rigid registration robust for fast motion, Fusion4D [37] performs spatio-temporal coherent non-rigid registration across multiple views. Guo *et al.* [38] utilized a shading-based scheme for more accurate non-rigid registration, allowing the simultaneous reconstruction of a casual 3D scene with both a detailed geometric model and surface albedo. However, such non-rigid registration methods often require huge memory to store the non-rigid transformation flows, preventing their use for large indoor 3D scenes. SurfelWarp [39] estimates a deformation field based on surfels but not truncated signed distance function (TSDF) voxels, which avoids the high memory consumption. However, such methods still need to solve the non-rigid registration problem, with real-time performance provided by GPU acceleration.

Recently, MaskFusion [11] proposed segmenting moving objects using a combination of 2D semantic detection and geometric priors. StaticFusion (SF) [12] presented a method for background reconstruction in dynamic environments, by joint estimation of camera motion and scene segmentation. Refusion [13] introduced direct tracking on the TSDF to estimate the camera pose in dynamic scenes. Bujanca *et al.* [40] presented a framework, FullFusion, for dense semantic reconstruction in dynamic scenes, which enables the incremental reconstruction of semantically-annotated non-rigidly deforming objects; the RGB-D data is divided into static and dynamic frames using a segmentation module, and only static frames are used for camera pose estimation.

Unlike these works, our approach detects dynamic landmarks and estimates the camera motion from static parts, and thus avoids solving the time-consuming non-rigid registration problem, or detecting/segmenting moving objects with time-consuming camera pose estimation or the aid of a 2D CNN. With efficient and accurate static/dynamic identification, our lightweight SLAM system can accurately track the camera pose in dynamic scenes.

Other works [41], [42] use deep networks such as Faster-RCNN [43] to detect moving objects or segment scenes with semantic parts from multiple camera views [44]. 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. Our system has three components: camera tracking with dynamics, local mapping and loop closing. Local mapping and loop closing are performed as in ORB-SLAM2 [18]. Camera tracking with dynamics aims to efficiently estimate the ego-motion between frames by accurately detecting and eliminating dynamic 3D landmarks. It contains two main subcomponents.

The first subcomponent performs *initial camera pose estimation* using GC-RANSAC (see Section 3.2). 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. The points determined as static are then 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 a long-term consistent CRF (see Section 3.3). Based on the initial camera pose estimation, we build a conditional random field with long-term consistency of observations (Fig. 3) and use it to *more accurately* identify static and dynamic feature points. This allows us to eliminate the dynamic points, and refine the camera pose estimation using the static points.

3.2 Initial Camera Pose Estimation

For each 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 [45] 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 issue, feature points on moving objects must be detected and eliminated, leaving static feature points to enable an accurate estimation of ego-motion.

In this step, we first roughly label landmarks as static or dynamic, and then estimate the ego-motion using only the static landmarks. As shown in Fig. 4, for an image pair $\{K_i, K_i'\}$ with fundamental matrix $F(K_i, K_i')$, a 3D landmark P_i with its matching pair of 2D observations (p_i, p_i') is likely to be static if $p' \in K_i'$ lies on the epipolar line $l_i' = Fp_i$, and otherwise be dynamic. Thus, we can formulate the static/dynamic landmark identification problem as inlier/outlier identification during fundamental matrix estimation using the GC RANSAC algorithm [46]. Specifically, for a given set $M = \{(p_i, p_i') | i = 1, \ldots, n\}$ of n 2D to 2D matching pairs, on each iteration of RANSAC we label each matching pair as

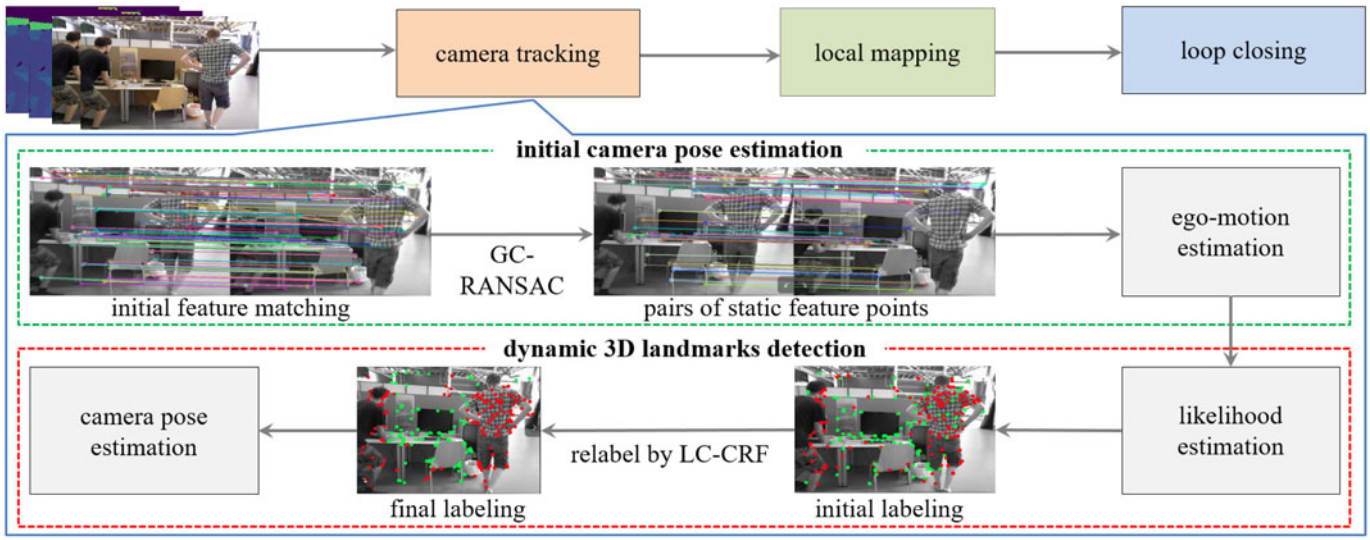


Fig. 2. Overview of our approach. To achieve accurate pose estimation in dynamic scenes, camera tracking is performed in two stages, from coarse (initial camera pose estimation) to fine (dynamic 3D landmark detection). We first use GC-RANSAC to detect and remove dynamic feature points and estimate an initial camera pose using the remaining static feature points. Then, we apply LC-CRF to relabel all landmarks, and refine the camera pose using landmarks determined to be static.

an inlier or an outlier for the estimated fundamental matrix F. 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:

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

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. $\phi\big(p_i, p_i', \theta\big)$ 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 term is defined as:

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

We empirically set $\lambda = 0.14$ and $\epsilon = 0.1$. The total energy can be efficiently optimized by the graph cut algorithm [47]. Fig. 5 shows an example of selecting static feature points using this GC-RANSAC-based method, which is summarized in Algorithm 1.

We later use the estimated fundamental matrix to derive static/dynamic priors for accurate dynamic point detection (see Section 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 $d_i =$

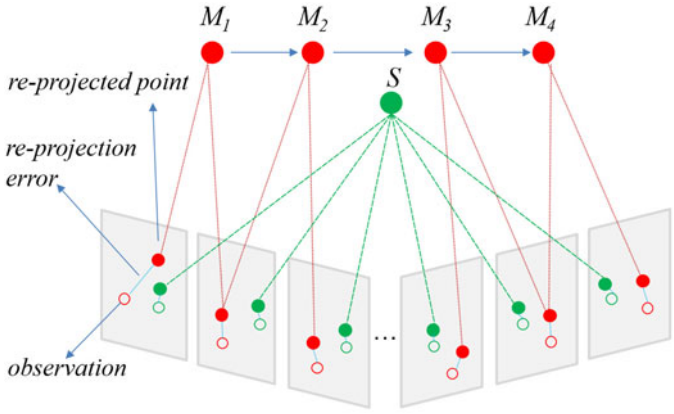


Fig. 3. A static landmark has more consistent observations than a dynamic one. M is a dynamic landmark which moves from M_1 to M_4 quickly; just a few frames observe the same location. Static landmark S stays at the same location and is seen at the same position in more frames. Re-projected points from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

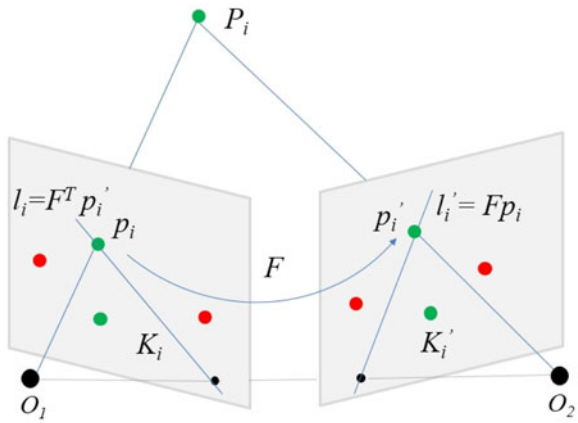


Fig. 4. Fundamental matrix and epipolar constraint. For a matched pair (p_i, p_i'), where p_i and p_i' are related to the same 3D point P_i, the epipolar constraint can be expressed as: $p_i'^\top F p_i = 0$, i.e. p_i' lies in the epipolar line $l_i' = Fp_i$ or p_i lies in the epipolar line $l_i = F^\top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

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

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

Fig. 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 static feature points, and are used for initial ego-motion estimation.

$|l_i \cdot p_i|/\sqrt{A_i^2 + B_i^2}$ and $d_i' = |l_i' \cdot p_i'|/\sqrt{A_i'^2 + B_i'^2}$. 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^γ 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^T p_i' and l_i' = F p_i
    Compute distances d_i and d_i'
    Compute the static/dynamic identification prior:
    P_i^γ = exp(-((d_i + d_i')/2 - μ_γ)^2/(2σ_γ^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 an extended time. Furthermore, dynamic points often have larger photometric re-projection errors between the re-projected point and the corresponding 2D feature point. Finally, we also note that points in the neighborhood of a static or dynamic point also tend to be static or dynamic, respectively. This key set of observations motivates us to use a long-term consistent conditional random field (LC-CRF) for dynamic point detection.

Specifically, we build the LC-CRF on the current detected landmarks, with a fully connected graph [48] linking each pair of landmarks. Each landmark P_i is assigned a label $x_i = L_i \in \{0, 1\}$ (0 for static and 1 for dynamic). Our goal is to find the optimal label assignment for all landmarks by minimizing the Gibbs energy E defined on the LC-CRF:

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

Unary Potential $\psi_u(x_i)$. 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. 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 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 $\mu_\cdot$ and $\sigma_\cdot$ 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 static confidence $1 - c$. In our implementation, we set $\lambda_1 = \lambda_2 = \lambda_3 = 1/3$. Following [49], the unary potential is then defined as:

$$\psi_u(x_i) = \begin{cases} -\log(c) I(P_i^s > t) & \text{if } x_i = 0 \\ -\log(1-c) I(P_i^s > t) & \text{if } x_i = 1 \end{cases}, \tag{4}$$

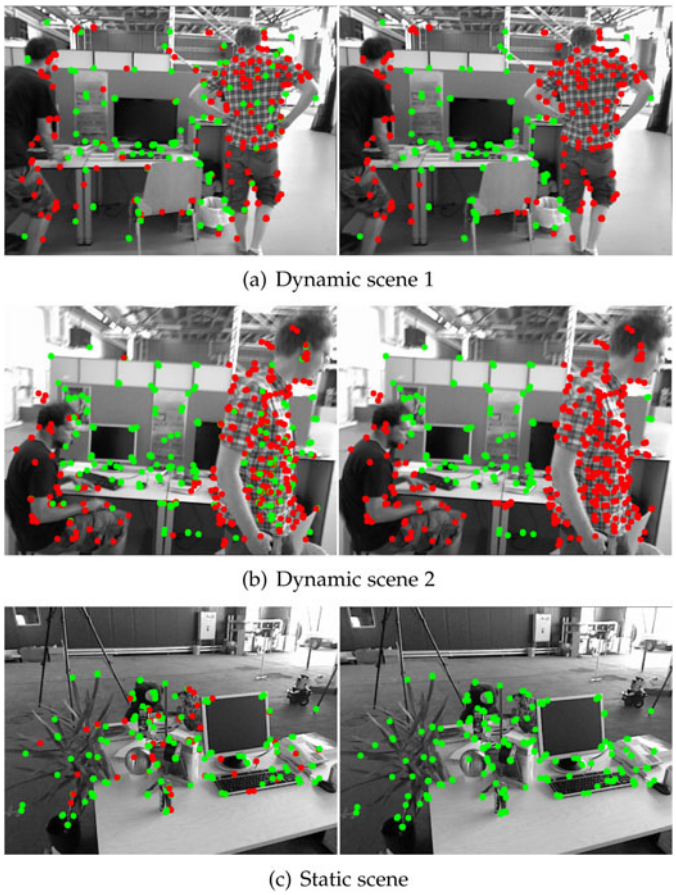
where $I(\cdot)$ is the indicator function.

Pairwise Potential $\psi_p(x_i, x_j)$. We design the pairwise potential to encourage consistent labeling between a landmark and its neighbors 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. Here we use two Gaussian kernels, an *observation kernel* and a *location kernel*.

The *observation kernel* is based on the idea that landmarks with similar average re-projection errors (α) and the number of observations (β) 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 across



(a) Dynamic scene 1

(b) Dynamic scene 2

(c) Static scene

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

many more key-frames over a long period. Similarly, static landmarks have lower average re-projection errors than dynamic landmarks. Thus, landmarks with different labels should have apparent differences both in the number of observations, and average re-projection error, so the observation kernel is defined as:

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

The *location kernel* is based on the idea that nearby 3D landmarks are likely to belong to the same compact object which is either static (e.g., a table) or dynamic (e.g., a person), and hence be in the same class. 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 Figs. 6a, 6b, some static feature points in the person are surrounded by dynamic ones (left image), and these are re-labeled as dynamic by LC-CRF inference (right image). The location kernel function is defined as:

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

The static/dynamic labeling problem represented by our LC-CRF can be solved efficiently using a mean field approximation method [48]. We show several examples illustrating 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 accurately segmented even for highly dynamic scenes. We refer the reader to the supplementary video, for more results, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2020.3028218..

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

Algorithm 2. Dynamic Landmark Detection and Accurate Pose Estimation

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

4 Experiments

4.1 Preliminaries

To evaluate the accuracy of the estimated camera pose, we tested our method on the TUM [50] and Bonn [13] RGB-D dynamic datasets. For the former, we selected 6 different indoor dynamic sequences with moving people and violent camera shaking; for the latter, we selected 20 sequences of more complex dynamic motion in indoor scenes. The evaluation uses two metrics to measure the accuracy between the estimated camera poses and the ground truth: the absolute trajectory error (ATE, measured in meters) and the relative pose error (RPE, measured in meters per second), as defined in [50]. 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.

4.2 Parameter Choice

The main parameters in our LC-CRF SLAM are those in the unary and pairwise potentials in dynamic landmark detection. We performed an extensive study of these parameters to determine appropriate settings.

4.2.1 Parameter Ranges

We first grouped these parameters into 6 pairs of parameters, and set the range for these parameter pairs as: $\{\mu_\alpha, \sigma_\alpha\} \in$

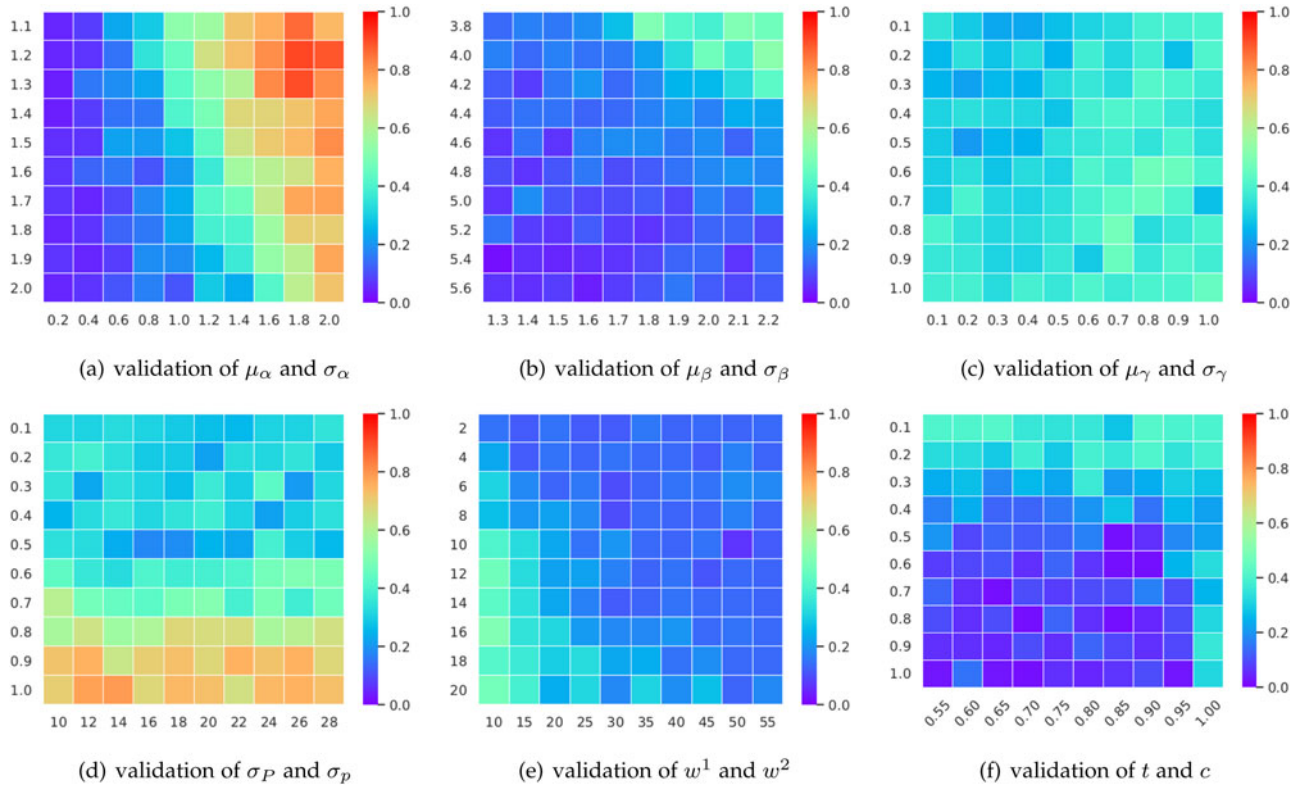


Fig. 7. Averaged ATE on TUM RGB-D dynamic sequences for parameter validation. y- and x-axes represent (a) μ_α and σ_α, respectively; (b) μ_β and σ_β, respectively; (c) μ_γ and σ_γ, respectively; (d) σ_P and σ_p, respectively; (e) w^1 and w^2, respectively; (f) t and c, respectively. Red presents higher error, blue lower.

$[1.1, 2.0] \times [0.2, 2.0]$, $\{\mu_\beta, \sigma_\beta\} \in [3.8, 5.6] \times [1.3, 2.2]$ and $\{\mu_\gamma, \sigma_\gamma\} \in [0.1, 1.0] \times [0.1, 1.0]$, $\{\sigma_P, \sigma_p\} \in [0.1, 1.0] \times [10, 28]$, $\{w^1, w^2\} \in [2, 20] \times [10, 55]$, threshold and confidence $\{t, c\} \in [0.1, 1.0] \times [0.55, 1.00]$. Since the first four parameter pairs come from observations, e.g., α concerns re-projection errors, we computed statistics for these observations on the test sequences beforehand and empirically set the ranges for the first four parameter pairs based on these statistics. The static likelihood threshold t was set to be less than 1, i.e., in $[0.1, 1]$, and the static confidence c for static landmarks was set to be over 0.5, i.e in $[0.55, 1.0]$. The ranges for the weight parameter pair $\{w^1, w^2\}$ were set to penalize pairs of neighboring landmarks with different labels.

4.2.2 Parameter Configuration

Since considering all parameter combinations is infeasible, we choose to select the parameter configuration of the 6 parameter pairs sequentially with parameter cross validation. Specifically, for each parameter pair, we evenly sampled n candidate values for the parameters and randomly selected m pairs of values of the parameters for the other 5 parameter pairs. For each parameter configuration, we performed LC-CRF SLAM on the six TUM RGB-D dynamic sequences and calculated the average ATE to assess the accuracy for this parameter configuration. In total, we thus considered $6nm$ parameter configurations in the parameter study. Typically, we set $n = 10 \times 10$, $m = 10$.

For each candidate value in each parameter pair, we further computed the average of the averaged ATE for its m corresponding parameter configurations. In total, this led to $6n$ averaged ATEs, as shown in Fig. 7. Typically, each pair of parameters tends to have an optimal choice within its m corresponding parameter configurations, independently of the choice for the other parameters. The parameter study led to the following parameter configuration: $\{\mu_\alpha = 1.7, \sigma_\alpha = 0.6\}$, $\{\mu_\beta = 5.4, \sigma_\beta = 1.5\}$, $\{\mu_\gamma = 0.3, \sigma_\gamma = 0.2\}$, $\{\sigma_P = 0.5, \sigma_p = 18\}$, $\{w^1 = 8, w^2 = 30\}$ and $\{t = 0.8, c = 0.7\}$. These settings achieved relatively small ATE errors across all parameter configurations, and were used for all experiments in this paper. The supplementary materials available online detail the ATE errors for all $6nm$ parameter configurations.

4.3 Comparison With Unmodified ORB-SLAM

We first evaluate the performance of our dynamic camera tracking compared with the original ORB-SLAM to demonstrate the effectiveness of our dynamic point detection module as a dynamic SLAM method. 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 ORB-SLAM in Table 1.

As can be seen, for highly dynamic sequences (those whose names begin with 'walking', i.e. fast moving persons or camera), our proposed method achieves significantly lower ATEs and RPEs than ORB-SLAM. In the last two scenarios with less dynamic contents, i.e., 'sitting-xyz' and 'desk-with-person', our algorithm also achieves slightly better results.

4.4 Comparison Using TUM Dataset

To evaluate the effectiveness of our approach for camera pose tracking of dynamic scenes, we compared our LC-CRF

TABLE 1
ATE (m) and RPE (m/s, Translational RPE-RMSE) of Different Methods on TUM RGB-D Dynamic Datasets

Sequence	BaMVO	DVO	SPW	FF	ORB-SLAM	RF	SF	MF	ours w/o GC	ours
	ATE									
	Mean	Mean	Mean	Mean	Mean (Std)	Mean (Std)	Mean (Std)	Mean (Std)	Mean (Std)	Mean (Std)
fr3/walking-xyz	-	0.093	0.060	0.041	0.366 (0.256)	0.088 (0.065)	0.094 (0.064)	0.104 (0.330)	0.028 (0.016)	**0.016** (**0.011**)
fr3/walking-halfsphere	-	0.047	0.043	0.029	0.382 (0.188)	0.063 (0.034)	0.434 (0.333)	0.106 (0.335)	0.058 (0.032)	**0.028** (**0.015**)
fr3/walking-static	-	0.066	0.026	0.014	0.214 (0.084)	0.016 (0.008)	0.015 (0.008)	0.035 (0.129)	0.016 (**0.007**)	**0.011** (0.008)
fr3/walking-rpy	-	0.133	0.179	-	0.745 (0.401)	0.257 (0.157)	1.183 (0.667)	0.827 (0.374)	0.100 (0.062)	**0.046** (**0.034**)
fr3/sitting-xyz	-	0.048	0.040	0.043	0.011 (**0.005**)	0.035 (0.020)	0.039 (0.022)	0.031 (0.049)	0.024 (0.012)	**0.009** (**0.005**)
fr2/desk-with-person	-	0.060	**0.048**	-	0.074 (0.016)	0.057 (0.018)	0.055 (0.027)	0.308 (0.121)	0.065 (**0.014**)	0.069 (0.015)
Mean (all sequences)	-	0.075	0.066	0.032*	0.299	0.086	0.303	0.235	0.049	**0.030**
Std (all sequences)	-	0.030	0.052	**0.012***	0.242	0.080	0.418	0.280	0.029	0.021
	t.RPE									
fr3/walking-xyz	0.233	0.436	0.065	0.060	0.518 (0.388)	0.122 (0.080)	0.133 (0.083)	0.159 (0.736)	0.040 (0.021)	**0.021** (**0.015**)
fr3/walking-halfsphere	0.174	0.263	0.053	0.071	0.570 (0.316)	0.092 (0.047)	0.579 (0.499)	0.085 (0.533)	0.085 (0.048)	**0.035** (**0.024**)
fr3/walking-static	0.134	0.382	0.033	0.036	0.311 (0.212)	0.025 (0.013)	0.023 (**0.011**)	0.089 (0.384)	0.024 (0.012)	**0.014** (**0.011**)
fr3/walking-rpy	0.358	0.404	0.225	-	1.093 (0.632)	0.356 (0.219)	1.751 (1.116)	0.534 (0.673)	0.187 (0.088)	**0.050** (**0.046**)
fr3/sitting-xyz	0.048	0.045	0.022	0.036	0.016 (**0.007**)	0.050 (0.028)	0.056 (0.031)	0.085 (0.088)	0.034 (0.017)	**0.012** (**0.007**)
fr2/desk-with-person	0.035	0.035	**0.017**	-	0.104 (0.052)	0.092 (**0.043**)	0.126 (0.082)	0.091 (0.316)	0.091 (0.051)	0.086 (0.045)
Mean (all sequences)	0.164	0.261	0.069	0.051*	0.435	0.123	0.444	0.174	0.077	**0.036**
Std (all sequences)	0.111	0.165	0.072	**0.015***	0.356	0.109	0.613	0.163	0.055	0.026

* *only 4 sequences are considered for FullFusion (FF).*

SLAM method with other state-of-the-art dynamic SLAM systems, i.e. dense visual odometry (DVO) [32], background-model-based dense-visual-odometry (BaMVO) [7], and static point weighting (SPW) [9], as well as dynamic fusion methods: ReFusion(RF) [13], StaticFusion(SF) [12], MaskFusion(MF) [11] and FullFusion(FF) [40], using the standard TUM RGB-D dynamic dataset. To make a fair comparison, we used the results produced by publicly released code or reported in the original paper. Table 1 gives the corresponding ATE and translational RPE accuracy results for the various dynamic SLAM systems. Due to a lack of public source code or correct code, we do not show results for FullFusion with 'fr3/walking-rpy' and 'fr2/desk-with-person' sequences and only report the RPE for BaMVO from the original paper. Averages and standard deviations of accuracy results for every single sequence and for all sequences are also calculated (BaMVO, DVO, SPW and Fullfusion do not report the standard deviation in their original papers). As shown in Table 1, our full LC-CRF SLAM achieves an average ATE error of 0.030 m (with standard deviation 0.021 m) and an average RPE error of 0.036 m/s (with standard deviation 0.026 m/s), which is significantly lower than methods like DVO, RF, SF, MF and FF, and better than the SPW method. For all sequences, our LC-CRF SLAM achieves almost the lowest ATE and RPE errors, except for the 'fr2/desk-with-person' sequence. In this almost static scene, a few static landmarks are labeled as dynamic by the GC-RANSAC filter with its standard parameter settings, degrading the accuracy of the initial pose estimation.

4.5 Comparison Using Bonn Dataset

To further evaluate the accuracy of camera pose tracking, we compared our approach with three start-of-the-art dense reconstruction methods: ReFusion (RF) [13], StaticFusion (SF) [12] and MaskFusion (MF) [11], on the Bonn RGB-D dynamic dataset. Results were obtained by running available open source implementations for each method. Table 2 shows the statistics of ATE and RPE errors for both single sequence and all sequences. Our method outperforms the others in most sequences (17 of 20) in terms of ATE, and achieves the lowest RPE for half of the sequences.

The ATE between estimated trajectories and ground-truth is further visualized in Fig. 8. As can be seen clearly, the trajectories estimated by our LC-CRF SLAM are much closer to the real trajectories than those provided by RF, SF and MF. This confirms again that long-term consistency is effective for dynamic landmark detection in such highly dynamic scenes using only sparse feature points.

4.6 Effectiveness of GC-RANSAC Filter

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

Table 1 includes ATE results for our LC-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. This shows that GC-RANSAC plays an effective role for camera pose estimation, especially in highly dynamic scenarios.

4.7 Dynamic Dense Reconstruction

To further evaluate the benefit of static/dynamic 3D landmark detection and intuitively show the accuracy of camera pose determined by our method, we considered a simple dense reconstruction method based on our dynamic RGB-D SLAM. Specifically, like MaskFusion [11], we recognise the dynamic regions, e.g., people, using the mask predicted by Mask R-CNN [51] as well as the dynamic points determined

TABLE 2
ATE (m) and RPE (m/s, Translational RPE-RSME) for Different Methods on Bonn RGB-D Dynamic Datasets

Sequence	ATE				t.RPE			
	RF	SF	MF	ours	RF	SF	MF	ours
	Mean (Std)	Mean (Std)	Mean (Std)	Mean (Std)	Mean (Std)	Mean (Std)	Mean (Std)	Mean (Std)
balloon	0.205 (0.126)	0.264 (0.095)	0.165 (0.073)	**0.027** (**0.014**)	0.576 (0.370)	0.585 (0.372)	**0.509** (**0.317**)	0.612 (0.373)
balloon2	0.195 (0.110)	0.250 (0.120)	0.114 (0.049)	**0.024** (**0.011**)	0.540 (0.301)	0.534 (0.287)	**0.499** (0.286)	0.541 (**0.275**)
balloon-tracking	0.445 (0.237)	0.202 (0.147)	0.194 (0.135)	**0.025** (**0.019**)	1.031 (0.587)	**0.900** (**0.565**)	0.991 (0.614)	0.965 (0.575)
balloon-tracking2	0.277 (0.137)	0.286 (0.098)	0.238 (0.097)	**0.045** (**0.023**)	1.059 (0.687)	0.949 (0.624)	0.937 (0.600)	**0.935** (**0.596**)
crowd	0.114 (0.062)	0.132 (0.081)	0.473 (0.161)	**0.019** (**0.012**)	**0.198** (**0.104**)	0.211 (0.128)	0.633 (0.400)	0.238 (0.170)
crowd2	0.192 (0.100)	0.193 (0.114)	0.653 (0.282)	**0.031** (**0.033**)	0.315 (0.178)	0.313 (0.184)	0.854 (0.618)	**0.199** (**0.103**)
crowd3	0.115 (0.067)	0.146 (0.094)	0.341 (0.108)	**0.023** (**0.015**)	0.223 (0.125)	0.266 (0.153)	0.503 (0.306)	**0.194** (**0.109**)
kidnapping-box	0.169 (0.078)	0.252 (0.097)	0.200 (0.106)	**0.023** (**0.015**)	0.886 (0.754)	0.853 (0.715)	**0.840** (**0.683**)	1.001 (0.801)
kidnapping-box2	0.132 (0.057)	0.186 (0.078)	0.182 (0.076)	**0.020** (**0.010**)	1.077 (0.752)	1.030 (0.729)	**1.027** (**0.724**)	1.184 (0.798)
moving-no-box	0.079 (0.035)	0.087 (0.046)	0.120 (0.050)	**0.018** (**0.009**)	0.939 (0.584)	0.939 (**0.583**)	0.947 (0.604)	**0.936** (0.586)
moving-no-box2	0.186 (0.092)	0.224 (0.110)	0.193 (0.084)	**0.038** (**0.019**)	1.287 (0.791)	1.267 (0.782)	**1.252** (**0.765**)	1.399 (0.850)
moving-o-box	0.319 (0.134)	0.376 (0.125)	**0.216** (0.066)	0.253 (**0.064**)	1.274 (0.839)	0.894 (0.546)	**0.847** (**0.520**)	1.158 (0.772)
moving-o-box2	0.608 (0.248)	**0.242** (**0.116**)	0.298 (0.133)	0.341 (0.259)	1.523 (1.145)	0.649 (0.412)	**0.576** (**0.378**)	1.143 (0.932)
person-tracking	0.354 (0.125)	0.390 (0.137)	0.301 (0.128)	**0.035** (**0.013**)	1.209 (0.684)	1.197 (0.678)	1.312 (0.857)	**1.193** (**0.676**)
person-tracking2	0.494 (0.235)	0.497 (0.214)	0.220 (0.069)	**0.040** (**0.014**)	**1.165** (**0.681**)	1.192 (0.701)	1.267 (0.837)	1.297 (0.892)
placing-no-box	0.109 (0.056)	0.133 (0.086)	0.325 (0.120)	**0.014** (**0.009**)	0.355 (0.240)	0.361 (0.240)	0.598 (0.332)	**0.333** (**0.192**)
placing-no-box2	0.121 (0.074)	0.209 (0.071)	0.153 (0.067)	**0.016** (**0.011**)	0.282 (**0.187**)	0.361 (0.226)	0.330 (0.192)	**0.271** (0.199)
placing-no-box3	0.181 (0.076)	0.219 (0.097)	0.156 (0.058)	**0.036** (**0.023**)	0.511 (0.342)	0.534 (0.365)	0.491 (0.358)	**0.482** (**0.339**)
placing-o-box	0.605 (0.365)	**0.261** (0.107)	0.424 (0.144)	0.320 (**0.095**)	1.180 (0.862)	0.528 (0.285)	0.791 (0.443)	**0.505** (**0.257**)
removing-no-box	0.050 (0.028)	0.061 (0.040)	0.058 (0.040)	**0.013** (**0.006**)	0.262 (0.150)	0.274 (0.163)	0.263 (0.157)	**0.240** (**0.126**)
Mean (**Std**)-all	0.248 (0.170)	0.231 (0.104)	0.251 (0.140)	**0.068** (**0.103**)	0.795 (0.432)	**0.692** (0.342)	0.773 (**0.307**)	0.741 (0.419)
Max (**Min**)-all	0.608 (0.050)	0.497 (0.061)	0.653 (0.058)	**0.341** (**0.013**)	1.523 (1.198)	**1.267** (0.211)	1.312 (0.263)	1.399 (**0.194**)

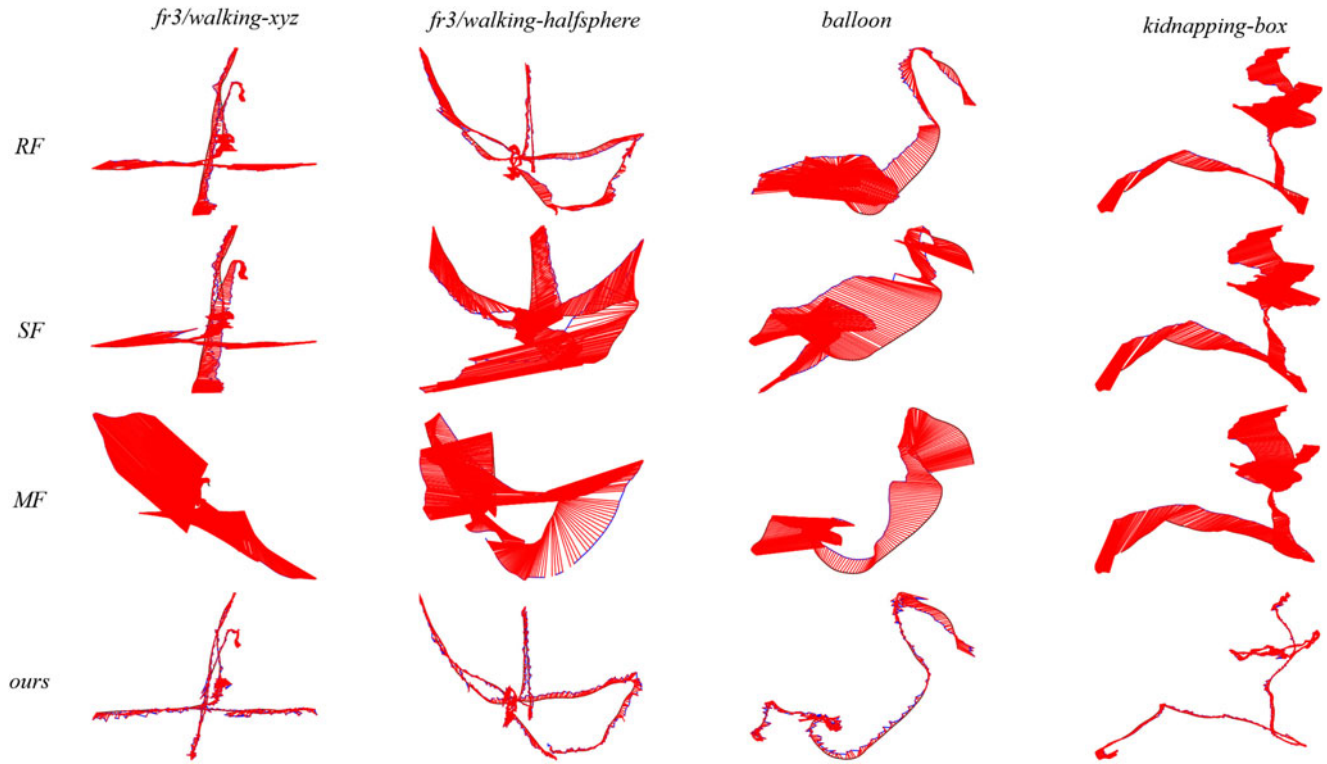


Fig. 8. Demonstration of the camera trajectories (blue) estimated by approaches ReFusion, StaticFusion, MaskFusion and our LC-CRF method (each column), along with the ground truth trajectories (black) of sequences (from left to right) *fr3/walking-xyz*, *fr3/walking-halfsphere* (from the TUM dataset), *balloon*, and *kidnapping-box* (from the Bonn dataset). The red segments connecting the corresponding positions between ground truth trajectories and estimated trajectories represent the ATE.

by registration errors between the current frame and the previous one; finally we fuse the remaining static points using the camera poses tracked by our RGB-D SLAM method.

Example dense reconstruction results are shown in Figs. 1 and 9. As can be seen clearly, dynamic regions, e.g., moving people, are effectively removed from the reconstructed scenes. These results demonstrate that our method provides

Fig. 9. The reconstructed point clouds for two scenes (top: fr3/walking-xyz, bottom: fr3/walking-static) from TUM RGB-D dataset.

accurate camera poses and can produce good 3D reconstruction results for dynamic scenes.

4.8 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 by computing the ATE and RPE for each frame, 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. 10 shows the ATE and translational RPE with respect to the dynamic ratio. These errors increase with a greater proportion of dynamic feature points. As expected, the camera poses estimated by our approach get worse with increasing amounts of dynamic content. Our approach can still handle significant amounts of dynamic content (up to about 50 percent) while keeping the ATE and translational RPE under about 0.2 m and 0.2 m/s, respectively.

4.9 Timings

Our approach has two main processes, i.e., initial pose estimation and dynamic landmark detection. The time taken by these processes was recorded for both TUM and Bonn RGB-D sequences and listed in Table 3. While the initial pose estimation is expensive due to the time-consuming GC-RANSAC, our method still achieves a near-real-time processing rate: 16 and 13 fps for TUM and Bonn RGB-D sequences, respectively. These experiments were performed on a CPU without GPU acceleration.

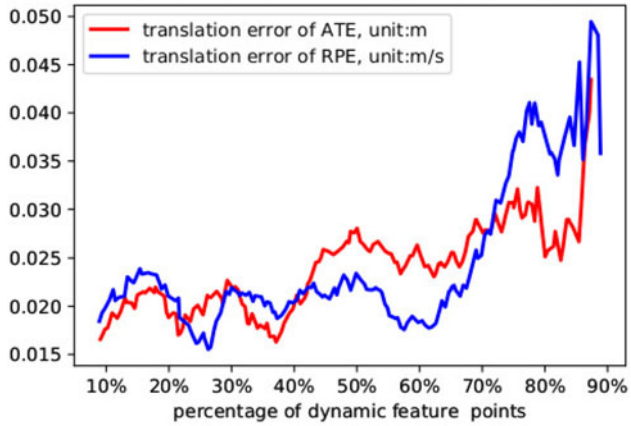


Fig. 10. Variation in ATE and RPE translation error with differing proportions of dynamic content. x-axis: percentage of dynamic feature points compared to all feature points. y-axis: Red: ATE translation error. Blue: RPE translation error.

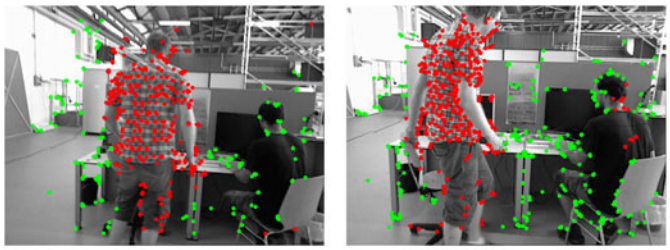
4.10 Discussion and Limitations

One of the main benefits of our approach comes from the unary and pairwise potentials used in dynamic landmark detection, 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) of the whole video sequence, which implicitly

TABLE 3
Time Taken for Each Step of the Tracking Thread (s)

Sequence	#Fr.	IPE	DLD	Total	FPS
fr3/walking-xyz	859	26.50	3.18	47.68	18.02
fr3/walking-halfsphere	1067	34.75	3.92	65.68	16.24
fr3/walking-static	743	17.71	4.81	40.31	18.43
fr3/walking-rpy	910	28.05	3.13	53.23	17.10
fr3/sitting-xyz	1261	60.81	4.42	92.56	13.62
fr2/desk-with-person	4067	149.40	13.44	241.82	16.82
balloon	439	12.74	1.91	25.40	17.28
balloon2	469	14.03	2.39	27.47	17.07
balloon-tracking	590	32.84	1.96	48.04	12.28
balloon-tracking2	451	17.81	1.74	26.39	17.09
crowd	928	32.61	3.91	58.51	15.86
crowd2	895	27.47	5.80	58.43	15.32
crowd3	854	32.43	5.38	59.41	14.38
kidnapping-box	1091	60.09	5.67	81.80	13.34
kidnapping-box2	1294	72.67	6.95	92.16	14.04
moving-no-box	778	39.46	3.52	61.33	12.69
moving-no-box2	937	61.83	4.81	80.58	11.63
moving-o-box	590	34.32	1.71	48.91	12.06
moving-o-box2	783	45.91	2.80	66.90	11.70
person-tracking	580	20.07	1.77	33.39	17.37
person-tracking2	567	26.25	1.62	41.27	13.74
placing-no-box	721	40.01	3.18	54.10	13.33
placing-no-box2	677	32.94	3.69	45.60	14.85
placing-no-box3	662	34.47	3.02	42.55	15.56
placing-o-box	998	54.48	2.95	68.36	14.60
removing-no-box	494	21.60	2.40	31.21	15.83

#Fr.: number of frames, IPE: initial pose estimation, DLD: dynamic landmark detection.



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

Fig. 11. Typical failure cases: in both frame 699 (a) and frame 727 (b) of the sequence of *fr3/walking-xyz*, the landmarks on the moving person are labeled as dynamic accurately; however, almost all landmarks for the person sitting stationary for a long time are labeled as static.

enforces long-term consistency in the unary potential computation. Also, the observation kernel used in the pairwise potential computation leverages the total number of observations, again providing a feature across long spans of frames.

Our approach still suffers from four main drawbacks. First, wrong static/dynamic landmark detection occurs, influencing the final camera pose estimation. This happens mainly in highly dynamic scenes with insufficient reliable static landmarks for camera pose estimation. Another cause arises from the epipolar line constraints used in the GC-RANSAC filter. Some dynamic objects may move along the direction of the epipolar line between consecutive frames, leading to incorrect initial static/dynamic labeling. One possible solution to this problem may be introducing more structural prior hints, such as planarity constraints.

Second, it is not as effective for almost static scenes, mainly because it may wrongly label static feature points as dynamic, thereby lowering camera pose estimation accuracy. One possible solution is to allow the user to choose whether to use the dynamic object detection module. If it is turned off, the final pose estimate is mainly determined by the process of initial camera pose estimation.

Third, as shown in Fig. 11, our approach does not perform very well for objects which are stationary for a long time before starting 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 [52].

Lastly, 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 ensure that the system works robustly in various scenarios.

5 Conclusion

This paper has presented our LC-CRF SLAM system for accurate pose estimation and effective dynamic point detection. To reduce the impact of dynamic points on pose estimation, we first 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 LC-CRF SLAM is significantly more accurate than existing methods for the highly dynamic examples in the public TUM RGB-D dataset and Bonn RGB-D dataset, and that it can be incorporated into the dynamic 3D reconstruction. In the future, we hope to explore potential AR/VR applications for dynamic scenarios, taking advantage of the static/dynamic information identified by our light-weight camera pose tracking.

Acknowledgements

The authors would like to thank the reviewers for their detailed and constructive comments on this article. This work was supported by the Natural Science Foundation of China (Grant No. 61863031, 61902210, 61521002) and the China Postdoctoral Science Foundation (Grant No. 2019M660646).

References

[1] 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 Trans. Vis. Comput. Graph.*, vol. 24, no. 11, pp. 2947–2962, Nov. 2018.

[2] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[3] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Magazine*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[4] 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 *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 1290–1297.

[5] D. Moratuwage, B. Vo, and D. Wang, "Collaborative multi-vehicle SLAM with moving object tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 5702–5708.

[6] C. C. Wang, C. Thorpe, M. Hebert, S. Thrun, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *Int. J. Robot. Res.*, vol. 26, no. 9, pp. 889–916, Sep. 2007.

[7] D. Kim and J. Kim, "Effective background model-based RGB-D dense visual odometry in a dynamic environment," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1565–1573, Dec. 2016.

[8] S. Meerits, D. Thomas, V. Nozick, and H. Saito, "FusionMLS: Highly dynamic 3D reconstruction with consumer-grade RGB-D cameras," *Comput. Vis. Media*, vol. 4, no. 4, pp. 287–303, Dec. 2018.

[9] S. Li and D. Lee, "RGB-D SLAM in dynamic environments using static point weighting," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2263–2270, Oct. 2017.

[10] H. Zhang and F. Xu, "MixedFusion: Real-time reconstruction of an indoor scene with dynamic objects," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 12, pp. 3137–3146, Dec. 2018.

[11] M. Runz, M. Buffier, and L. Agapito, "MaskFusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2018, pp. 10–20.

[12] R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon, and D. Cremers, "Staticfusion: Background reconstruction for dense RGB-D slam in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 3849–3856.

[13] E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss, "Refusion: 3D reconstruction in dynamic environments for rgb-d cameras exploiting residuals," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 7855–7862.

[14] D. Barath and J. Matas, "Graph-cut RANSAC," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6733–6741.

[15] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 573–580.

[16] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[17] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, 2007, pp. 225–234.

[18] R. Mur-Artal and J. D. Tardós, "Orb-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
[19] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2320–2327.
[20] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 15–22.
[21] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: large-scale direct monocular SLAM," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 834–849.
[22] O. Kähler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. Torr, and D. Murray, "Very high frame rate volumetric integration of depth images on mobile devices," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 11, pp. 1241–1250, Nov. 2015.
[23] Z. Yan, M. Ye, and L. Ren, "Dense visual SLAM with probabilistic surfel map," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 11, pp. 2389–2398, Nov. 2017.
[24] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal.Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.
[25] A. Bapat, E. Dunn, and J. Frahm, "Towards kilo-hertz 6-DoF visual tracking using an egocentric cluster of rolling shutter cameras," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 11, pp. 2358–2367, Nov. 2016.
[26] J. R. Rambach, A. Tewari, A. Pagani, and D. Stricker, "Learning to fuse: A deep learning approach to visual-inertial camera pose estimation," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2016, pp. 71–76.
[27] S. Yang, B. Li, M. Liu, Y.-K. Lai, L. Kobbelt, and S.-M. Hu, "HeteroFusion: Dense scene reconstruction integrating multi-sensors," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 11, pp. 3217–3230, Nov. 2020.
[28] M. Garon and J. Lalonde, "Deep 6-DoF tracking," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 11, pp. 2410–2418, Nov. 2017.
[29] J. Zhang, M. Gui, Q. Wang, R. Liu, J. Xu, and S. Chen, "Hierarchical topic model based object association for semantic SLAM," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 11, pp. 3052–3062, Nov. 2019.
[30] 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 *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 5875–5884.
[31] J. Huang, S. Yang, T.-J. Mu, and S.-M. Hu, "ClusterVO: Clustering moving instances and estimating visual odometry for self and surroundings," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2168–2177.
[32] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 3748–3754.
[33] M. Zollhöfer *et al.*, "State of the art on 3D reconstruction with RGB-D cameras," *Comput. Graph. Forum*, vol. 37, no. 2, pp. 625–652, May 2018.
[34] R. A. Newcombe, D. Fox, and S. M. Seitz, "DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 343–352.
[35] M. Slavcheva, M. Baust, D. Cremers, and S. Ilic, "KillingFusion: Non-rigid 3D reconstruction without correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5474–5483.
[36] M. Slavcheva, M. Baust, and S. Ilic, "SobolevFusion: 3D reconstruction of scenes undergoing free non-rigid motion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2646–2655.
[37] M. Dou *et al.* "Fusion4D: Real-time performance capture of challenging scenes," *ACM Trans. Graph.*, vol. 35, no. 4, Jul. 2016, Art. no. 114.
[38] K. Guo, F. Xu, T. Yu, X. Liu, Q. Dai, and Y. Liu, "Real-time geometry, albedo, and motion reconstruction using a single RGB-D camera," *ACM Trans. Graph.*, vol. 36, no. 4, Jun. 2017, Art. no. 32.
[39] W. Gao and R. Tedrake, "Surfelwarp: Efficient non-volumetric single view dynamic reconstruction," in *Proc. Robot.: Sci. Syst.*, 2018, pp. 29:1–29:10.
[40] M. Bujanca, M. Luján, and B. Lennox, "FullFusion: A framework for semantic reconstruction of dynamic scenes," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2019, pp. 2168–2177.
[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 *Proc. 3rd IEEE Int. Conf. Comput. Commun.*, 2017, pp. 2398–2402.
[42] F. Zhong, S. Wang, Z. Zhang, C. Chen, and Y. Wang, "Detect-SLAM: Making object detection and SLAM mutually beneficial," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2018, pp. 1001–1010.
[43] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[44] A. Dai and M. Nießner, "3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 458–474.
[45] L. Quan and Z. Lan, "Linear N-point camera pose determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 8, pp. 774–780, Aug. 1999.
[46] Y. Boykov and G. Funkalea, "Graph cuts and efficient N-D image segmentation," *Int. J. Comput. Vis.*, vol. 70, no. 2, pp. 109–131, Nov. 2006.
[47] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2001, vol. 1, pp. 105–112.
[48] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with gaussian edge potentials," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 109–117.
[49] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji, "PanopticFusion: Online volumetric semantic mapping at the level of stuff and things," in *Proc. Int. Conf. Intell. Robots Syst.*, 2019, pp. 4205–4212.
[50] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. Int. Conf. Intell. Robots Syst.*, 2012, pp. 573–580.
[51] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020.
[52] M. Halber, Y. Shi, K. Xu, and T. Funkhouser, "Rescan: Inductive instance segmentation for indoor RGBD scans," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 2541–2550.

Zheng-Jun Du received the bachelor's degree from Qinghai University, in 2012, and the master's degree from Tsinghua University, in 2015. He is currently working toward the PhD dehgree with the Department of Computer Science and Technology, Tsinghua University. His research interests include 3D reconstruction, dynamic SLAM, geometric modeling, and image processing.

Shi-Sheng Huang received the PhD degree in computer science and technology from Tsinghua University, Beijing, in 2015. He is currently a Post-Doc researcher with Tsinghua University. His primary research interests include fields of computer graphics, computer vision, and visual SLAM.

Tai-Jiang Mu received the bachelor's and doctor's degrees from the Department of Computer Science and Technology, Tsinghua University, in 2011 and 2016, respectively, where he is currently an assistant researcher with the Department of Computer Science and Technology, Tsinghua University. His research interests include visual media learning, SLAM, and human robot interaction.

Qunhe Zhao received the PhD degree from the Chinese Academy of Sciences, in 2017. He is currently a researcher in artificial intelligence at DeepBlue Technology (Shanghai) Co., Ltd. He is mainly responsible for perception and mapping algorithms for autonomous driving.

Kun Xu (Member, IEEE) received the bachelor's and doctor's degrees from the Department of Computer Science and Technology, Tsinghua University, in 2005 and 2009, respectively, where he is currently an associate professor. His research interests include computer graphics.

Ralph R. Martin is an emeritus professor with Cardiff University. His past activities have included editorial board memberships of various journals and chairing various conferences. He was a fellow of the Learned Society of Wales. In 2014, Ralph was awarded the Friendship Award, China's highest Award for foreign nationals.