Accurate Dynamic SLAM using CRF-based Long-term Consistency

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

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 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 3D position 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 3D position tracking and environment reconstruction; it has received intense research interest from the computer graphics, computer vision and mixed/augmented/virtual reality communities. Since daily life usually contains dynamic items such as moving people and objects, an accurate visual SLAM method which works efficiently in dynamic environment is even 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 which could easily fail to estimate camera poses 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 detection and tracking of moving objects (DATMO) scheme [6]. However, these DATMO-based methods suffer from drawbacks arising from assumptions made about the moving objects e.g. a predetermined number of objects, or slow speed. 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 determining points or regions as static or dynamic relies 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

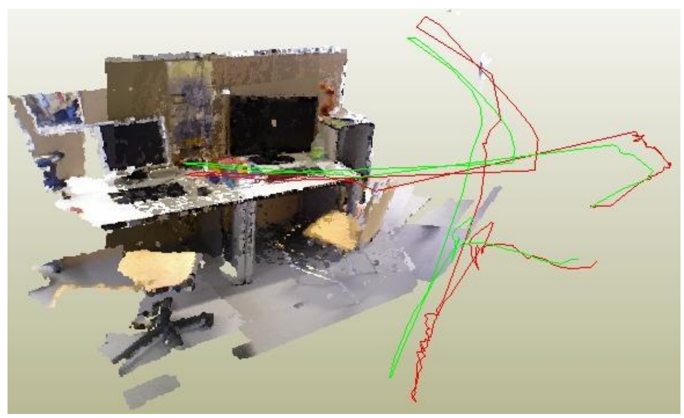


Fig. 1. Reconstructed scene for fr3/walking-halfsphere from the TUM RBG-D dynamic dataset. As an accurate 3D position tracking technique for dynamic environments, our efficient approach utilizing observationally consistent CRFs can calculate a high precision camera trajectory (red) close to the ground truth (green).

- *Manuscript received XX XX 2020; revised XX XX, 2020.*
- **Corresponding author.*
- *Zheng-Jun Du is with Department of Computer Technology and Application, Qinghai University and BNRist, Department of Computer Science and Technology, Tsinghua University, Beijing 100084. Email: duzj19@mails.tsinghua.edu.cn.*
- *Shi-Sheng Huang, Tai-Jiang Mu and Kun Xu are with BNRist, Department of Computer Science and Technology, Tsinghua University, Beijing 100084. Email: shishenghuang.net@gmail.com, {taijiang, xukun}@tsinghua.edu.cn.*
- *Qunhe Zhao is with DeepBlue Technology(Shanghai) Co.,Ltd, Shanghai 200336. Email: zhaoqh@deepblueai.com.*
- *Ralph R. Martin is with School of Computer Science and Informatics, Cardiff University, Cardiff CF243AA, U.K. Email: MartinRR@cardiff.ac.uk.*

registration [10], [12], [13] or 2D CNNs [11] is expensive both in computaional effort and memory, preventing them from providing 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 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 short observations. Based on this key observation, we first estimate the camera pose based on an initial static/dynamic labelling by inlier/outlier determination using 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 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 from the remaining static 3D landmarks.

Our LC-CRF based dynamic 3D landmark detection 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 having 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 achieve 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 an 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 results particularly relevant to our work, and refer readers to [2] 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 [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 perform camera position 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 those 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 perform the camera position tracking by detecting moving objects with the aid of dense scene flow [4] or object clustering [30]. Kerl et al. [31] 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. [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 [31] 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

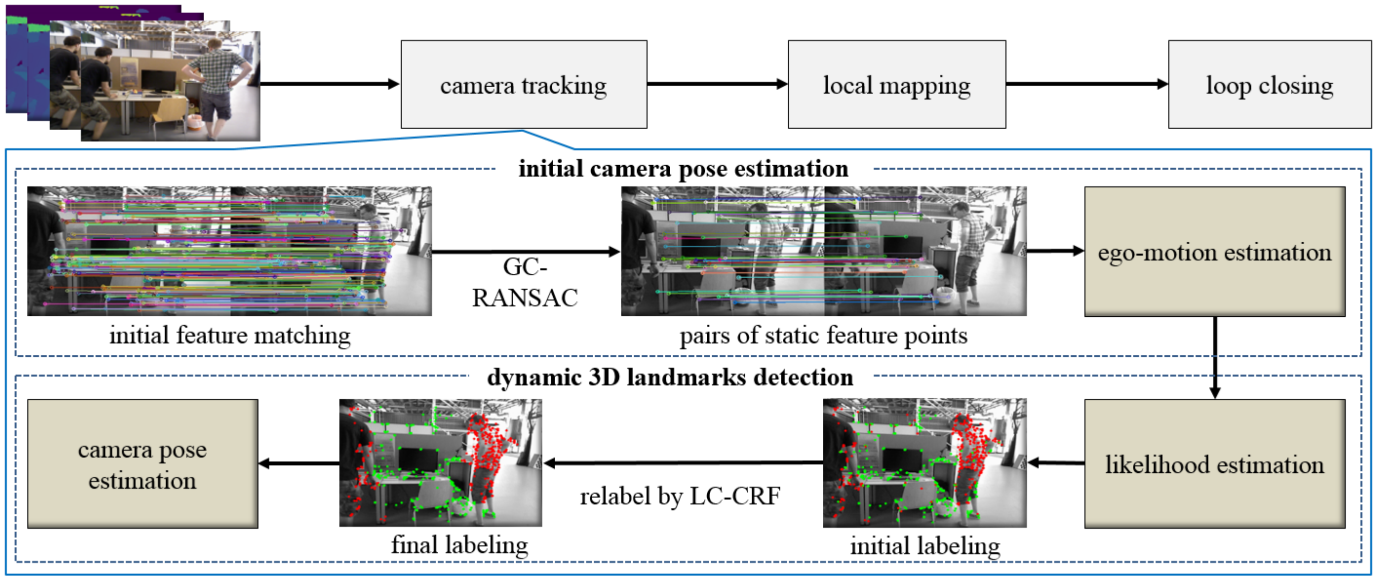


Fig. 2. Overview of our approach. To achieve accurate pose estimation in dynamic scenes, camera tracking is performed in two stages, coarse (initial camera pose estimation) and fine (dynamic 3D landmarks detection). We first use GC-RANSAC to filter out dynamic feature points and estimate camera pose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

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 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 [32] for the state-of-the-art in 3D reconstruction approaches. Human movement is an important source of dynamism in indoor scenes, and DynamicFusion [33] proposed the first dense SLAM system to reconstruct dynamic scenes with humans by accurately solving a volumetric flow field that transforms the current scene into a canonical frame. Subsequent works such as KillingFusion [34], SobolevFusion [35] extended the flow field with more accurate nonrigid motion estimation without templates or shape priors. To make the non-rigid registration robust with fast motion, Fusion4D [36] performs spatio-temporal coherent non-rigid registration across multiple views. Guo et al. [37] utilized a shading-based scheme for more accurate non-rigid registration, allowing the simultaneous reconstruction of a casual 3D scene with both a detailed geometry model and surface albedo. However, such non-rigid registration methods often have huge memory requirement to store such non-rigid transformation flows, preventing their use for large indoor 3D scenes. SurfelWarp [38] estimates a deformation field based on surfels but not TSDF voxels, which avoids the dense 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 to segment moving objects with the combination of 2D semantic detection and geometry priors. StaticFusion (SF) [12] presented a method for background reconstruction in dynamic environments, by a joint estimation of camera motion and scene segmentation. Refusion [13] introduced a direct tracking on the truncated signed distance function (TSDF) to estimate the camera pose in dynamic scenes. Bujanca et al. [39] 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.

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 or segmenting moving objects with time consuming complex camera pose estimation or with the aid of 2D CNN. With efficient and accurate static/dynamic identification, our light-weight SLAM system can accurately track camera positions in large indoor dynamic scenes.

Other works [40], [41] use deep networks such as Faster-RCNN [42] to detect moving objects or segment scenes with semantic parts with multiple camera views [43]. 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

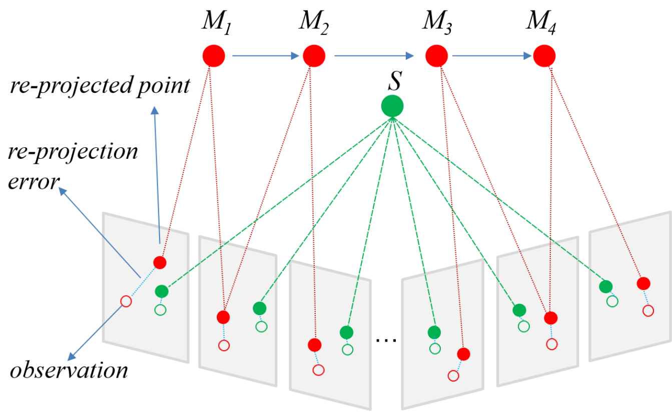


Fig. 3. A static landmark has more consistent observations than a dynamic one. M is a landmark which moves from M_1 to M_4 quickly; just a few frames observe the same location. Static landmark S stays at the same location and is seen at the same position in more frames. Re-projected points from static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

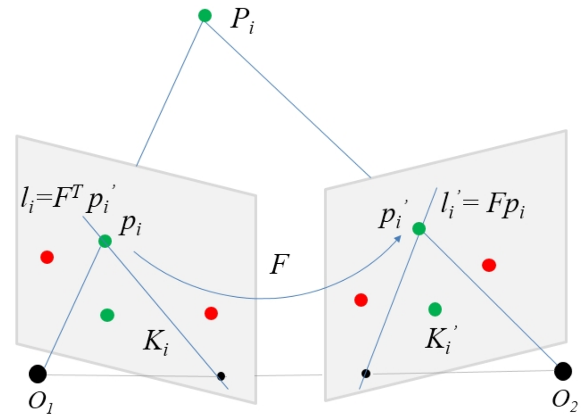


Fig. 4. Fundamental matrix and epipolar constraint. For a matched pair (p_i, p_i'), where p_i and p_i' are related to the same 3D point P_i, the epipolar constraint can be expressed as: $p_i'^{\top} F p_i = 0$, i.e. p_i' lies in the epipolar line $l_i' = Fp_i$ or p_i lies in the epipolar line $l_i =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

a 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, i.e. 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 Sec. 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 Sec. 3.3). Based on the initial camera pose estimate, we build a conditional random field with long-term consistency of observations (Fig. 3) and use it to *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 [44] 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 enable an accurate estimate of ego-motion to be made.

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 [45]. 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 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:

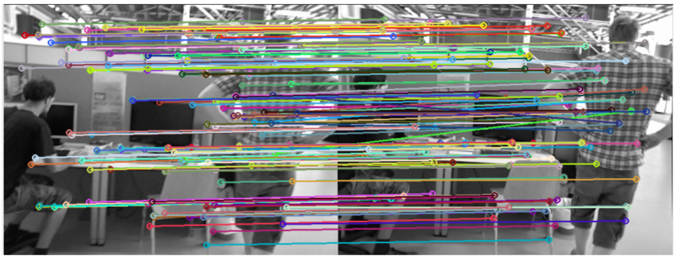
$$B(L_i) = \begin{cases} K\left(\phi\left(p_i, p_i', \theta\right), \epsilon\right) & \text{if } L_i = 0 \\ 1 - K\left(\phi\left(p_i, p_i', \theta\right), \epsilon\right) & \text{if } L_i = 1 \end{cases}, \quad (1)$$

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.

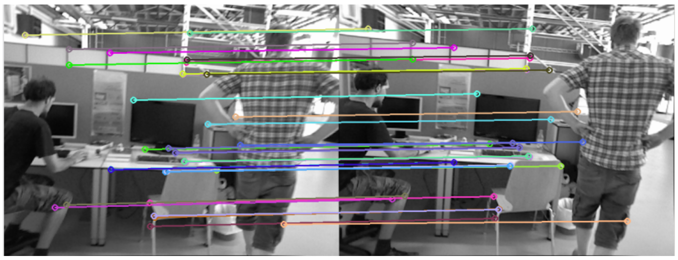
The pairwise energy term is defined as:

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

The total energy can be efficiently optimised by the graph cut algorithm [46]. Fig. 5 shows an example result of se-



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

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: $l_i = F^\top p_i'$ 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

lecting static feature points using our 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 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 $d_i = |l_i \cdot p_i|/\sqrt{A_i^2 + B_i^2}$ 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.

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 [47] linking each pair of landmarks. Every 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 optimal labels for the landmarks by minimizing the Gibbs energy E defined on the LC-CRF:

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

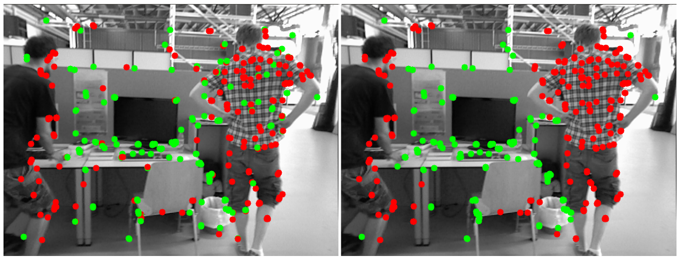
Unary potential $\psi_u(x_i)$. During SLAM processing, each landmark can be seen in several key-frames. We record the corresponding 2D observations $o_j^i \in R^2$, i.e. the 2D position in key-frame j for each 3D landmark P_i. The photometric re-projection error e_j^i between P_i and o_j^i is calculated. 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 $\mu_.$ 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 static confidence $1-c$. In our implementation, we set $\lambda_1 = \lambda_2 = \lambda_3 = 1/3$. Following [48], the unary potential is then defined as:

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

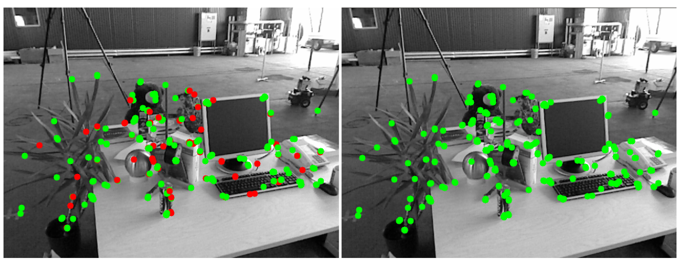
where $I(\cdot)$ is the indicator function.

Pairwise potential $\psi_p(x_i, x_j)$. We design the pairwise potential to encourage consistent labeling potential between



(a) Dynamic scene 1

(b) Dynamic scene 2



(c) Static scene

Fig. 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 LC-CRF optimization.

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 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 in many more key-frames over the longer-term. 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:

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

The *location kernel* is based on the idea that nearby 3D landmarks (P) 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 Fig. 6(a,b), some static feature points (p) 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:

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

The static/dynamic labeling problem represented by our LC-CRF can be solved efficiently by a mean field approximation method [47]. 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 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.

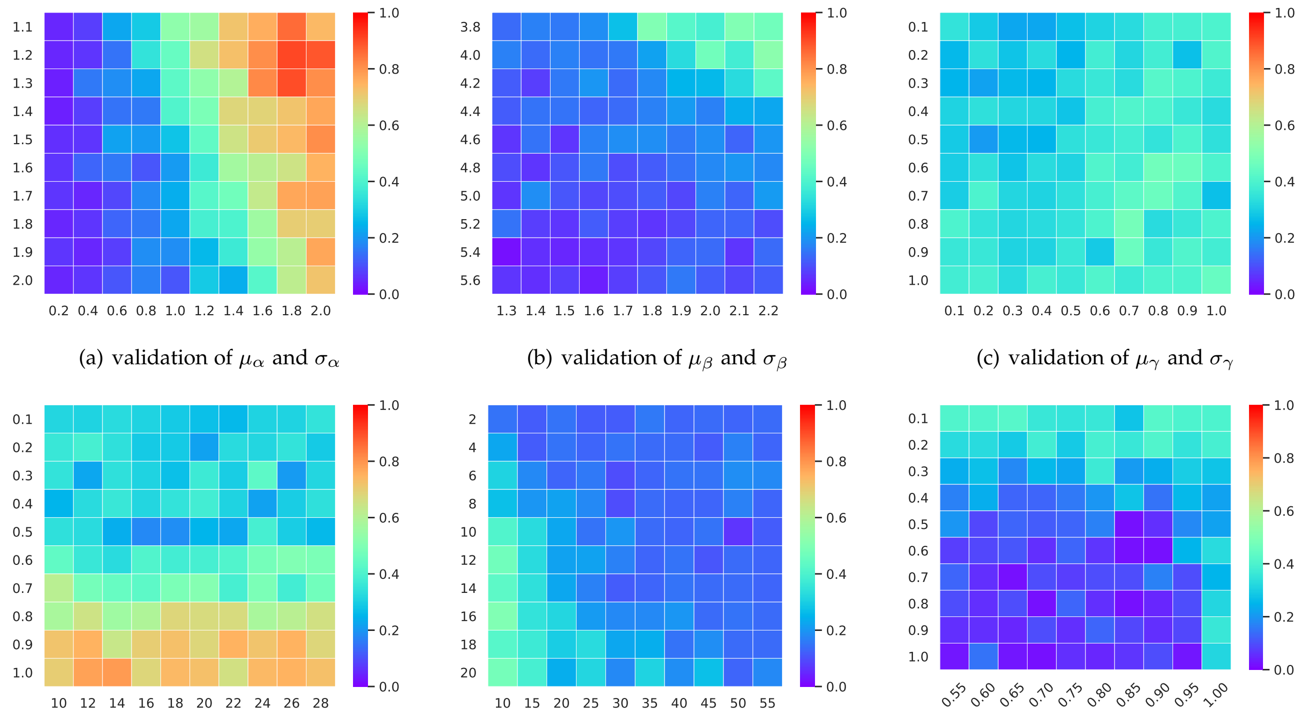
Algorithm 2 Dynamic Point Detection and Accurate Pose Estimation

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^*

4 EXPERIMENTS

4.1 Preliminaries

To evaluate the accuracy of estimated camera pose, we tested our method on the TUM [49] and Bonn [13] RGB-D dynamic datasets. The former contains 6 different indoor dynamic sequences with moving people and violent camera shaking. The latter includes 20 dynamic sequences of more complex dynamic movements 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 relative pose error (RPE, meausred in meters per second), as defined in [49]. 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.



(a) validation of μ_α and σ_α (b) validation of μ_β and σ_β (c) validation of μ_γ and σ_γ

(d) validation of σ_P and σ_p (e) validation of w^1 and w^2 (f) validation of t and c

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.

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.

Parameter Ranges. We first group these parameters into 6 pairs of parameters, and set the range for these parameter pairs as: $\{\mu_\alpha, \sigma_\alpha\} \in [1.1, 2.0] \times [0.2, 2.0]$, $\{\mu_\beta, \sigma_\beta\} \in [3.8, 5.6] \times [1.3, 2.2]$ 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 the observation measurements, such as α is for the re-projection errors, β is for observation number etc, we made statistics for these observation measurements on the test sequences beforehand and empirically set the varying ranges of the first four parameter pairs around the statistical values. 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 landmark points was set to be over 0.5, i.e in $[0.55, 1.0]$. The ranges of weight parameter pair $\{w^1, w^2\}$ were set to penalize pairs of neighbor landmarks with different labels.

Parameter Configuration. Since the enumeration of all the parameter configurations is too large, to make a practical study, we choose to select the parameter configuration of the 6 parameter pairs one by one with parameter cross validation. Specifically, for a parameter pair to be studied, we evenly sample n candidate values of the parameters in this parameter pair and randomly select m candidate values of the parameters from the other 5 parameter pairs. For each parameter configuration, we perform our LC-CRF SLAM on the six TUM RGB-D dynamic sequences and calculate the average ATE as the accuracy measurement of this parameter configuration. Then in total we get $6 \times n \times m$ parameter configuration in the parameter study. Typically, we set $n = 10 \times 10$, $m = 10$.

For each candidate value in each parameter pair, we further compute the average of the averaged ATE of it s m corresponding parameter configurations. Totally, we get $6 \times n$ averaged ATEs, as shown in Fig. 7. In general, each pair of parameters tends to have an optimal choice across its m corresponding parameter configurations, independent of the choices of other variables. According to the parameter study, we choose the 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 $\{t = 0.8, c = 0.7\}$ as our practical parameter settings for all experiments in this paper, which achieves relatively small ATE errors across all parameter configurations. Please refer to the supplementary materials for the detail ATE errors for all the $6 \times n \times m$ parameter configuration.

4.3 Comparison with unmodified ORB-SLAM

We first evaluate the performance of our dynamic camera tracking compared with the original ORB-SLAM to evaluate 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.

TABLE 1
ATE (m) and RPE (m/s, translational RPE-RMSE) of different methods on TUM RGB-D dynamic datasets.

Sequence	ORB-SLAM	BaMVO	DVO	SPW	RF	SF	MF	FF	ours w/o GC	ours
ATE										
fr3/walking-xyz	0.3662	-	0.0932	0.0601	0.0880	0.0937	0.1040	0.0410	0.0276	**0.0158**
fr3/walking-halfsphere	0.3824	-	0.0471	0.0432	0.0627	0.4341	0.1060	0.0290	0.0576	**0.0277**
fr3/walking-static	0.2141	-	0.0656	0.0261	0.0157	0.0149	0.0350	0.0140	0.0160	**0.0112**
fr3/walking-rpy	0.7445	-	0.1333	0.1791	0.2570	1.1828	0.8265	-	0.1004	**0.0455**
fr3/sitting-xyz	0.0108	-	0.0482	0.0397	0.0352	0.0392	0.0310	0.0430	0.0242	**0.0090**
fr2/desk-with-person	0.0743	-	0.0596	**0.0484**	0.0568	0.0549	0.3083	-	0.0653	0.0687
Mean	0.2987	-	0.0745	0.0661	0.0859	0.3033	0.2351	0.0317*	0.0485	**0.0296**
Std	0.2417	-	0.0333	0.0565	0.0873	0.4581	0.2801	**0.0115***	0.0293	0.0213
t.RPE										
fr3/walking-xyz	0.5178	0.2326	0.4360	0.0651	0.1223	0.1329	0.1592	0.0600	0.0402	**0.0210**
fr3/walking-halfsphere	0.5701	0.1738	0.2628	0.0527	0.0921	0.5788	0.0847	0.0710	0.0852	**0.0352**
fr3/walking-static	0.3111	0.1339	0.3818	0.0327	0.0246	0.0226	0.0885	0.0360	0.0242	**0.0141**
fr3/walking-rpy	1.0931	0.3584	0.4038	0.2252	0.3558	1.7507	0.5343	-	0.1869	**0.0503**
fr3/sitting-xyz	0.0159	0.0482	0.0453	0.0219	0.0503	0.0555	0.0853	0.0360	0.0341	**0.0116**
fr2/desk-with-person	0.1041	0.0352	0.0354	**0.0173**	0.0915	0.1261	0.0907	-	0.0905	0.0857
Mean	0.4353	0.1637	0.2609	0.0692	0.1228	0.4444	0.1737	0.0507*	0.0768	**0.0363**
Std	0.3556	0.1212	0.1806	0.0786	0.1193	0.6711	0.1634	**0.0152***	0.0552	0.0257

As we can see, 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 'sitting-xyz' and 'desk-with-person' with less dynamic environments, our algorithm also achieves slightly better results.

4.4 Comparison using TUM dataset

To evaluate the effectiveness of our approach in camera position tracking of dynamic scenes, we compared our proposed LC-CRF SLAM method with other state-of-the-art dynamic SLAM systems, i.e. dense visual odometry (DVO) [31], background-model-based dense-visual-odometry (BaMVO) [7], and static point weighting (SPW) [9], and dynamic fusion methods: ReFusion(RF) [13], StaticFusion(SF) [12], MaskFusion(MF) [11] and FullFusion(FF) [39], using the standard TUM RGB-D dynamic dataset. To make a fair comparison, we used the results produced by the publicly released correct code, or the ones 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 have results for FullFusion with 'fr3/walking-rpy' and 'fr2/desk-with-person' sequences and only reported the RPE for BaMVO from the original paper. Averages and standard deviations of accuracy results over all sequences are also calculated (only 4 sequences are considered for FullFusion). As shown in Table 1, our full LC-CRF SLAM achieves an average ATE error of 0.0296 m (with standard deviation 0.0213 m) and average RPE error of 0.0363 m/s (with standard deviation 0.0257 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 achieve 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 estimate.

4.5 Comparison using Bonn dataset

To further evaluate the accuracy of our method in camera position tracking, we compare our approach with three start-of-the-art dense reconstruction methods: ReFusion (RF) [13], StaticFusion (SF) [12] and MaskFusion (MF) [11] on twenty sequences from the Bonn RGB-D dynamic dataset. Results were obtained by running available open source implementations for each method. Table 2 shows the ATE error and RPE error for each sequence. Our methods 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 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 of RF, SF and MF. This confirms again that our 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 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 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 effective role for camera pose estimation especially in highly dynamic scenarios.

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
balloon	0.205237	0.264313	0.165218	**0.027025**	0.576177	0.585218	**0.509159**	0.612309
balloon2	0.195335	0.250055	0.114207	**0.023702**	0.539682	0.534409	**0.499210**	0.541184
balloon-tracking	0.444809	0.201526	0.194377	**0.025375**	1.031259	**0.899960**	0.991192	0.964838
balloon-tracking2	0.276541	0.286383	0.237533	**0.045088**	1.058717	0.948770	0.936857	**0.935175**
crowd	0.114327	0.132148	0.473352	**0.019194**	**0.198360**	0.211246	0.632631	0.238465
crowd2	0.192484	0.192562	0.652845	**0.030546**	0.314578	0.313426	0.854333	**0.199361**
crowd3	0.115207	0.146438	0.341056	**0.023024**	0.223178	0.266283	0.502591	**0.193993**
kidnapping-box	0.169041	0.251910	0.200313	**0.023343**	0.886491	0.852810	**0.839666**	1.001061
kidnapping-box2	0.132354	0.185596	0.181658	**0.019768**	1.077405	1.029510	**1.027437**	1.184211
moving-no-box	0.079135	0.087211	0.120255	**0.018111**	0.938572	0.938927	0.947417	**0.935672**
moving-no-box2	0.185903	0.223555	0.193382	**0.038269**	1.287263	1.267181	**1.252494**	1.398832
moving-o-box	0.319171	0.376174	**0.215559**	0.252694	1.274084	0.893685	**0.846522**	1.157946
moving-o-box2	0.608118	**0.242198**	0.298111	0.341286	1.522578	0.648539	**0.576068**	1.143141
person-tracking	0.354255	0.390163	0.301143	**0.034658**	1.209339	1.197149	1.312280	**1.192553**
person-tracking2	0.493568	0.496624	0.220311	**0.040429**	**1.164570**	1.191935	1.266656	1.296885
placing-no-box	0.108778	0.133295	0.325125	**0.014395**	0.355211	0.360681	0.598094	**0.332851**
placing-no-box2	0.120715	0.209213	0.152902	**0.015975**	0.281757	0.361393	0.330426	**0.270980**
placing-no-box3	0.181348	0.218882	0.156333	**0.035837**	0.511173	0.534316	0.491054	**0.481747**
placing-o-box	0.604833	**0.261445**	0.424042	0.319946	1.179726	0.528098	0.790644	**0.505206**
removing-no-box	0.049739	0.061347	0.058481	**0.013284**	0.261537	0.274229	0.263182	**0.239712**
Mean	0.247545	0.230552	0.251310	**0.068097**	0.794583	**0.691888**	0.773396	0.741306
Max	0.608118	0.496624	0.652845	**0.341286**	1.522578	**1.267181**	1.312280	1.398832
Min	0.049739	0.061347	0.058481	**0.013284**	0.198360	0.211246	0.263182	**0.193993**
Std	0.169786	0.103619	0.140356	**0.103425**	0.431667	0.342259	**0.307184**	0.418909

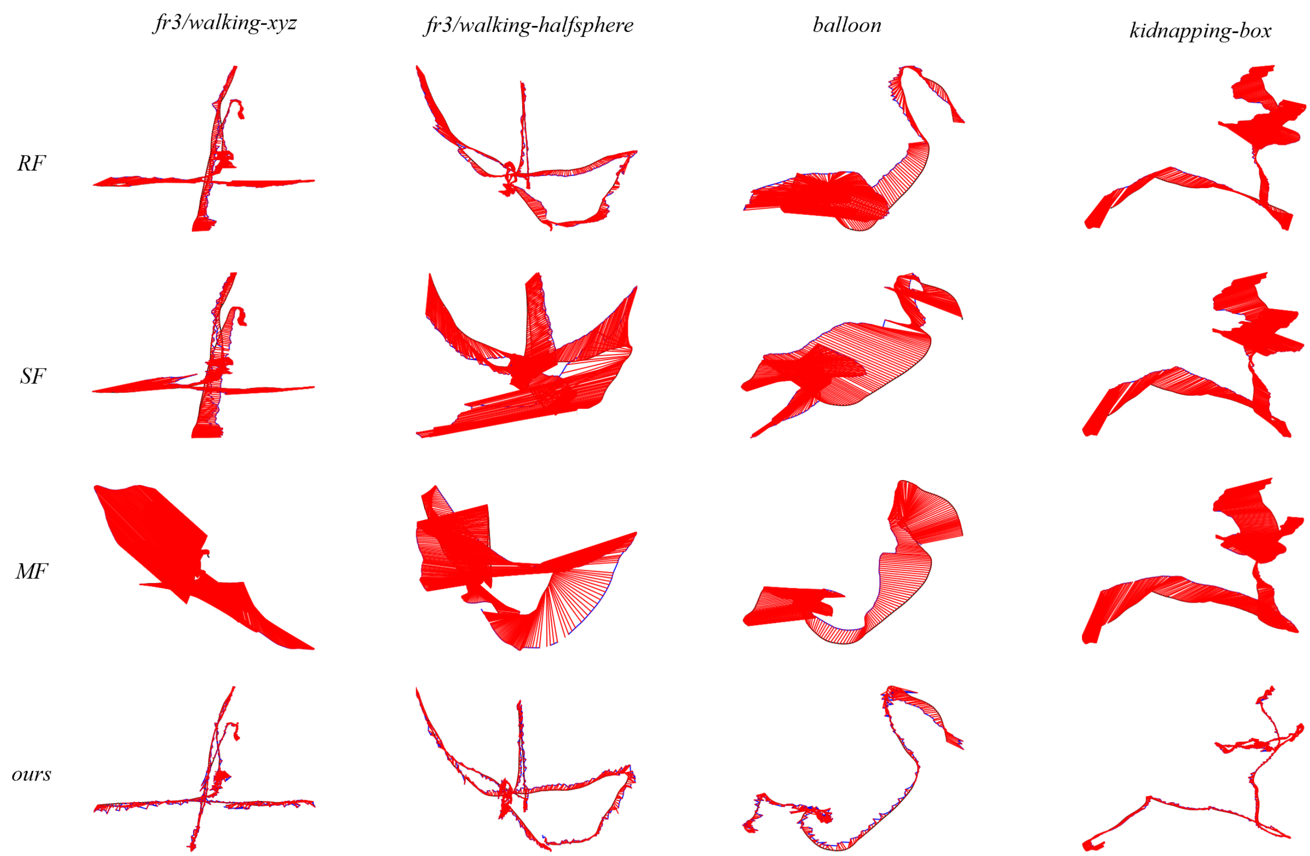


Fig. 8. Demonstration of the camera trajectories (blue) estimated by approaches ReFusion, StaticFusion, MaskFusion and our LC-CRF method (each column), along with the ground truth trajectories (red) of sequences (from left to right) *fr3/walking-xyz*, *fr3/walking-halfsphere* (from TUM dataset), *balloon*,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

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

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 consider 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 [50] as well as the dynamic points determined by the 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.

We also show example dense reconstruction results in Figs. 1 and 9. As can be clearly seen, dynamic regions, e.g. moving people, are effectively removed from the reconstructed scenes. These results demonstrate that our method provides accurate camera pose and is able to provide good reconstruction quality 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 frames, using the TUM dynamic dataset. Here we define

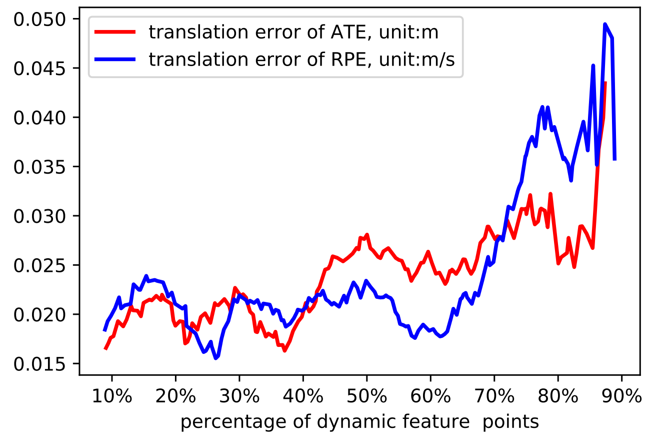


Fig. 10. Variation in ATE and RPE translation error with differing proportions of dynamic content. x-axis: percentage of dynamic feature points compared to all feature points. y-axis: Red: ATE translation error. Blue: RPE translation error.

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 ATE and RPE translation error variation with dynamic ratio. These errors become larger with increasing dynamic feature point ratio, so as expected, camera pose estimates provided by our approach worsen with increasingly dynamic scenes. Our approach can handle significant amounts of dynamic content (up to about 50%) while keeping ATE and RPE errors to under about 0.2m and 0.2 m/s, respectively.

4.9 Speed

Our approach has two main processes, initial pose estimation and dynamic landmark detection. The times taken by these processes were recorded for both TUM and Bonn RGB-D sequences, and are given in Table 3. While initial pose estimation in our approach is expensive due to 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.

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 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. Firstly, wrong static/dynamic landmark detection would

TABLE 3
Time taken for each step of the tracking thread (s). #Fr.: number of frames, IPE: initial pose estimation, DLD: dynamic landmark detection.

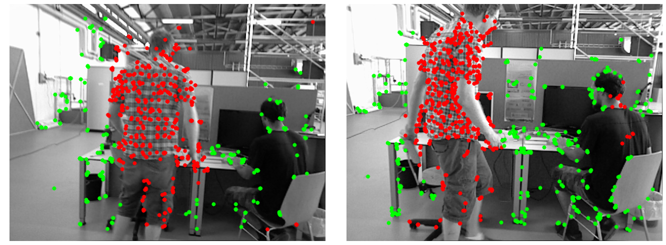
Sequence	#Fr.	IPE	DLD	Total	FPS
fr3/walking-xyz	859	26.50	3.18	47.68	18.02
fr3/walking-halfsphere	1067	34.75	3.92	65.68	16.24
fr3/walking-static	743	17.71	4.81	40.31	18.43
fr3/walking-rpy	910	28.05	3.13	53.23	17.10
fr3/sitting-xyz	1261	60.81	4.42	92.56	13.62
fr2/desk-with-person	4067	149.40	13.44	241.82	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

exist which will influence the final camera pose estimation. The wrong static/dynamic landmark detection occurs mainly in scenarios when the ratio of the dynamic parts is relatively too large to find enough reliable static landmarks for camera pose estimation. Another point might come from the epipolar line constraints used in the GC-RANSAC filter. There would be some dynamic objects moving along the direction of the epipolar line between consecutive two frames, which will lead to wrong determination for the initial static/dynamic labeling. One possible solution to this problem may be to introduce more structure prior hints such as planar constraint to overcome such challenge.

Secondly, 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 detection module. If it is turned off, the final pose estimate is mainly determined by the process of initial camera pose estimation.

Thirdly, as shown in Fig. 11, our approach does not perform very well for long term 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 [51].

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 robustly works in differing scenarios.



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

Fig. 11. Typical case causing failure: both in frame 699 (a) and frame 727 (b) of the sequence of *fr3/walking-xyz*, the person further back sits still without moving for a long time, and almost all landmarks for this person are labeled as static. In sharp contrast, the landmarks in the moving person on the left are labeled as dynamic accurately.

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 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 LC-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 dynamic 3D reconstruction. In 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

Thanks for the reviewers' detail 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] 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)*, vol. 24, no. 11, pp. 2947–2962, 2018.

[2] 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)*, vol. 32, no. 6, pp. 1309–1332, 2016.

[3] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE Robotics Automation Magazine (RAM)*, vol. 13, no. 2, pp. 99–110, 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 *2012 IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 1290–1297.

[5] 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)*, 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," *International Journal of Robotics Research (IJRR)*, vol. 26, no. 9, pp. 889–916, 2007.

[7] D. Kim and J. Kim, "Effective background model-based rgb-d dense visual odometry in a dynamic environment," *IEEE Transactions on Robotics (TRO)*, vol. 32, no. 6, pp. 1565–1573, 2016.

[8] S. Meerits, D. Thomas, V. Nozick, and H. Saito, "Fusionmls: Highly dynamic 3d reconstruction with consumer-grade rgb-d cameras," *Computational Visual Media*, vol. 4, no. 4, pp. 287–303, 2018.

[9] S. Li and D. Lee, "Rgb-d slam in dynamic environments using static point weighting," *IEEE Robotics and Automation Letters (RA-L)*, vol. 2, no. 4, pp. 2263–2270, 2017.

[10] H. Zhang and F. Xu, "Mixedfusion: Real-time reconstruction of an indoor scene with dynamic objects," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 24, no. 12, pp. 3137–3146, 2018.

[11] 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)*, 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 *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[14] D. Barath and J. Matas, "Graph-cut ransac," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 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 *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 573–580.

[16] 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)*, vol. 29, no. 6, pp. 1052–1067, 2007.

[17] 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)*, 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 Transactions on Robotics (TRO)*, vol. 33, no. 5, pp. 1255–1262, 2017.

[19] 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)*, 2011, pp. 2320–2327.

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

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

[22] 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)*, vol. 21, no. 11, pp. 1241–1250, 2015.

[23] Z. Yan, M. Ye, and L. Ren, "Dense visual slam with probabilistic surfel map," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 23, no. 11, pp. 2389–2398, 2017.

[24] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 40, no. 3, pp. 611–625, 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 Transactions on Visualization and Computer Graphics (TVCG)*, vol. 22, no. 11, pp. 2358–2367, 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 *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 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 Transactions on Visualization Computer Graphics (TVCG)*, 2020 to appear.

[28] M. Garon and J. Lalonde, "Deep 6-dof tracking," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 23, no. 11, pp. 2410–2418, 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 Transactions on Visualization and Computer Graphics (TVCG)*, vol. 25, no. 11, pp. 3052–3062, 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 *The IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 5875–5884.

[31] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 3748–3754.

[32] M. Zollhöfer, P. Stotko, A. Görlitz, C. Theobalt, M. Nießner, R. Klein, and A. Kolb, "State of the art on 3d reconstruction with RGB-D cameras," *Comput. Graph. Forum*, vol. 37, no. 2, pp. 625–652, 2018.

[33] R. A. Newcombe, D. Fox, and S. M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 2015, pp. 343–352.

[34] M. Slavcheva, M. Baust, D. Cremers, and S. Ilic, "Killingfusion: Non-rigid 3d reconstruction without correspondences," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 5474–5483.

[35] M. Slavcheva, M. Baust, and S. Ilic, "Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 2018, pp. 2646–2655.

[36] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, P. Kohli, V. Tankovich, and S. Izadi, "Fusion4d: Real-time performance capture of challenging scenes," *ACM Trans. Graph.*, vol. 35, no. 4, Jul. 2016.

[37] 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, 2017.

[38] W. Gao and R. Tedrake, "Surfelwarp: Efficient non-volumetric single view dynamic reconstruction," in *Robotics: Science and System (RSS)*, 2018.

[39] 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.

[40] 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)*, 2017, pp. 2398–2402.

[41] 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)*, 2018, pp. 1001–1010.

[42] 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)*, vol. 39, no. 6, pp. 1137–1149, 2017.

[43] A. Dai and M. Nießner, "3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation," in *ECCV*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11214. Springer, 2018, pp. 458–474.

[44] Long Quan and Zhongdan Lan, "Linear n-point camera pose determination," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 21, no. 8, pp. 774–780, 1999.

[45] Y. Boykov and G. Funkalea, "Graph cuts and efficient n-d image segmentation," *International Journal of Computer Vision (IJCV)*, vol. 70, no. 2, pp. 109–131, 2006.

[46] 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)*, vol. 1, 2001, pp. 105–112.

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

[48] 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.
[49] 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)*, 2012, pp. 573–580.
[50] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 42, no. 2, pp. 386–397, 2020.
[51] 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)*, 2019, pp. 2541–2550.

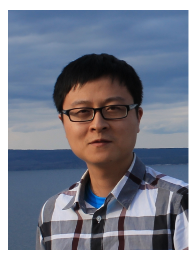
Zheng-Jun Du received his bachelor's degree from Qinghai University in 2012 and received his master's degree from Tsinghua University in 2015. He is currently a PhD candidate in the Department of Computer Science and Technology, Tsinghua University. His research interests include 3D reconstruction, dynamic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

Shi-Sheng Huang is currently a PostDoc researcher in Tsinghua University. He got his Ph.D. degree in computer science and technology from Tsinghua University, Beijing, in 2015. His primary research interests include fields of computer graphics, computer vision and visual SLAM.

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

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

Ralph R. Martin is an Emeritus Professor of 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.



Kun Xu is an associate professor in the Department of Computer Science and Technology, Tsinghua University. Before that, he received his bachelor's and doctor's degrees from the same university in 2005 and 2009, respectively. His research interests include computer graphics.