2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

A Switching-coupled Backend for Simultaneous Localization and Dynamic Object Tracking

Yuzhen Liu, Jiacheng Liu, Yun Hao, Bowen Deng and Ziyang Meng

Abstract—**Simultaneous localization and object tracking (SLOT) is essentially important for autonomous systems. Tightly-coupled and loosely-coupled methods are two commonly used back-end frameworks for the state-of-the-art solutions of SLOT problem. However, some inherent limitations exist in these two frameworks. In particular, the tightly-coupled method is usually disturbed by the poor observations of some dynamic objects, and the performance of a loosely-coupled one completely depends on that of classical static simultaneous localization and mapping (SLAM) process. Motivated by these observations, we propose a novel switching-coupled back-end solution and theoretically derive its concrete form using probability representation. Based on the switching strategy and the proposed objects classification criteria where the object uncertainty, observation quality and prior information are jointly considered, the dynamic objects' states are flexibly coupled with camera's state and static landmarks' states. For implementation, the measurement constraints of "good" dynamic objects and static landmarks are simultaneously leveraged to perform SLAM and good object tracking (SLAMGOT) process, and those of "bad" objects are used for bad object tracking (BOT) process based on the obtained camera state. Extensive evaluations on synthetic scenes, KITTI datasets and real-world experiments demonstrate the performance of the proposed method.**

Index Terms—**Visual tracking, localization, SLAM, probability and statistical methods.**

I. Introduction

Simultaneous localization and object tracking is an essential task in many applications for robotics and augmented reality [1]-[17]. For example, in autonomous driving, vehicles need to achieve self localization and perceive their surrounding moving cars, so as to avoid obstacles and remain safe [13]. On the other hand, for practical applications of augmented reality, dynamic targets need to be explicitly tracked in 3D space to enable interactions of virtual instances with real moving objects [11].

From technical perspective, SLOT algorithms are roughly classified into tightly-coupled and loosely-coupled solutions. In particular, in the tightly-coupled solutions, all the states of camera, dynamic objects and static landmarks are tightly coupled, and all the measurement equations are established at one time for the state estimation of the system. The concept of "simultaneity" is emphasized since the two problems of SLAM [18] [19] and dynamic object tracking (DOT) [3] are jointly solved in a same posterior estimation framework. In contrast, loosely-coupled methods usually decouple the considered SLOT problem into two separate processes, one classical SLAM process and the other DOT process [12]. For implementation, SLAM process is first performed to estimate the states of camera and static landmarks, and then DOT process is performed based on the obtained camera poses. Compared with the tightly-coupled methods, the loosely-coupled ones are computationally more efficient since the estimation processes are two-step and each step is relatively independent [13]. However, such a solution is actually not optimal since all the measurements from dynamic objects are directly ignored in the first process, and it is essentially a classical SLAM solution for the problem of camera state estimation. Therefore, loosely-coupled methods cannot work properly in the environments where the traditional SLAM process is prone to failure, e.g., automatic driving scene where the front vehicle obstructs the view. In contrast, the tightly-coupled methods achieve better estimation accuracy in theory, while significant computing resources are required especially when lots of tracking objects exist. More importantly, since all the states of camera and dynamic objects are tightly interrelated, the estimations of camera ego-motion and dynamic objects' motion may interfere with each other in the environments where some dynamic objects exist with poor observation (e.g., moving fast or being far away from the camera) and without proper prior information (e.g., scale and motion model) [5].

Motivated by the observation that loosely-coupled and tightly-coupled solutions have their respective limitations, we propose a *switching-coupled* back-end method in this paper in order to achieve a robust and accurate simultaneous localization and object tracking. The contributions of this paper are threefold. First, a novel switching-coupled back-end solution for SLOT is proposed. Compared with the tightly-coupled solutions [6]-[10], the proposed algorithm reduces the influence of the measurements from "bad" dynamic objects thanks to the proposed objects classification criteria and the switching strategy. On the other hand, compared with the loosely-coupled solutions [11]-[16], the proposed algorithm allows leveraging the measurement constraints of "good" dynamic objects to improve the accuracy and robustness of the system. Second, we propose a reasonable and effective objects classification strategy, where the object uncertainty, observations quality and prior information are jointly considered. Third, extensive evaluations on synthetic scenes,

Y. Liu, J. Liu, Y. Hao, B. Deng and Z. Meng are with Department of Precision Instrument, Tsinghua University, Beijing 100084, China. Corresponding author: Ziyang Meng. Emails: lyz17@mails.tsinghua.edu.cn (Y. Liu), liu-jc18@mails.tsinghua.edu.cn (J. Liu), haoy17@mails.tsinghua.edu.cn (Y. Hao), dengbowen@mail.tsinghua.edu.cn (B. Deng), and ziyang-meng@tsinghua.edu.cn (Z. Meng).

978-1-7281-9077-8/21/$31.00 ©2021 IEEE

KITTI datasets and real-world experiments demonstrate that the proposed method achieves better accuracy and robustness even compared with the state-of-the-art approaches.

The rest of this paper is organized as follows. In Section II, the related works are discussed. The notations and the problem formulation are given in Section III. Section IV introduces the proposed switch-coupled backend and objects classification criteria. In Section V, implementation details and experimental results are presented and the conclusions are finally summarized in Section VI.

II. Related works

The authors of [11]-[16] focus on loosely-coupled solutions for SLOT problem. In particular, C. Wang et al. [12] first decouple the solution of SLOT problem into two different estimators in the filter-based framework, one for static SLAM problem and the other for object tracking problem. P. Li et al. [13] use stereo vision to realize simultaneous localization and moving vehicles tracking in autonomous driving scene. The classical SLAM process is first performed to estimate the camera ego-motion by leveraging the observations of static landmarks. Then, combining with the prior scale information of vehicles, the pose and motion parameters of the moving vehicles are estimated via a separate batch optimization. In [14], K. Qiu et al. propose a loosely-coupled method for 3-D motion tracking of dynamic objects. In particular, an existing visual-inertial odometry (VIO) algorithm (VINS-mono [20]) is first performed to estimate the robot states. Then, relative pose without scale factor between the object and robot is calculated via a region-based bundle adjustment (BA). The unknown scale factor is finally recovered by a signal correlation-based estimation method. While this method displays impressive ability to track motion over a sliding window of images, the key assumption that the robot self-motion is completely unrelated to the object motion is actually not always guaranteed in practical tracking applications. In [16], H. Lim et al. employ a loosely-coupled method to achieve simultaneous localization and pedestrian tracking. In particular, a monocular visual odometry (VO) is proposed to estimate the up-to-scale camera pose, and then the measurement constraints of pedestrian with prior height information are leveraged to recover the scale factor.

On the other hand, tightly-coupled methods have been also proposed in [6]-[10]. In particular, K. Lin et al. [9] realize a simultaneous localization, mapping and moving object tracking within the extented Kalman filter (EKF) framework based on a stereo vision. In [7], M. Chojnacki propose a tightly-coupled method based on light bundle adjustment (LBA), where the case of monocular vision is considered and the moving object is treated as a particle with constant global velocity. In [6], K. Eckenhoff et al. propose a tightly-coupled method based on multi-state constraint Kalman filter (MSCK-F) [18], where three motion models for the dynamic objects are proposed and the system observability is correspondingly analyzed under these three different motion assumptions. In [10], S. Yang propose a dynamic SLAM algorithm using monocular vision in a tightly-coupled optimization-based framework, denoted by CubeSLAM, where a single image 3-D cuboid detection approach is proposed based on the estimation of vanishing points.

III. Notations and problem formulation

A. Notations

We consider four reference frames in this paper, including the world frame $\{w\}$, the camera frame $\{c\}$, the robot body frame $\{b\}$, and the object frame $\{q\}$. Without loss of generality, the robot body frame $\{b\}$ is assumed to be coincident with the camera frame $\{c\}$. For camera state, $\mathbf{x}_i = \left\{{}^{w}\mathbf{T}_c^i\right\}$ denotes the camera pose in the world frame at time i, where ${}^{w}\mathbf{T}_c^i \in \mathrm{SE}(3)$. For the state of dynamic object, we use $\mathbf{o}_i^q = \left\{{}^{w}\mathbf{T}_q^i, {}^{w}\mathbf{v}_q^i\right\}$ to represent the state of the q-th object at time i containing the 6-DoF pose ${}^{w}\mathbf{T}_q^i \in \mathrm{SE}(3)$ and the linear speed ${}^{w}\mathbf{v}_q^i \in \mathbb{R}^3$ both in the world frame. Also, we use $q = 0$ to denote the static background. For the landmark state, ${}^{j}\mathbf{l}^q = \{{}^{q}\mathbf{p}_j\}$ denotes the position of j-th landmark belonging to the q-th object in the object frame, where ${}^{q}\mathbf{p}_j \in \mathbb{R}^3$. For observations, ${}^{j}\mathbf{z}_i^q$ denotes the location of the j-th feature belonging to the q-th object on the image plane at time i. In addition, we define the following sets: $\mathbf{X}_k = \{\mathbf{x}_i\}_{i=0:k}$, $\mathbf{O}_k = \{\mathbf{O}_k^q\}_{q=1:N}$, $\mathbf{O}_k^q = \{\mathbf{o}_i^q\}_{i=0:k}$, $\mathbf{l} = \{\mathbf{l}^q\}_{q=0:N}$, $\mathbf{l}^q = \left\{{}^{j}\mathbf{l}^q\right\}_{j=1:N_q}$, $\mathbf{Z}_k = \{\mathbf{Z}_k^q\}_{q=0:N}$, $\mathbf{Z}_k^q = \{\mathbf{z}_i^q\}_{i=0:k}$, $\mathbf{z}_i^q = \left\{{}^{j}\mathbf{z}_i^q\right\}_{j=1:N_q}$, where time k denotes the current time instant, $\mathbf{X}_k$ denotes the camera states at all times, $\mathbf{O}_k$ and $\mathbf{O}_k^q$ respectively represent the states of all objects and those of the q-th object at all times, N is number of the dynamic objects, $\mathbf{l}$ and $\mathbf{l}^q$ respectively represent the states of all landmarks and those of the landmarks belonging to the q-th object, $N_q \in \mathbb{N}^+$ represents the number of landmarks belonging to the q-th object, $\mathbf{Z}_k$ and $\mathbf{Z}_k^q$ respectively represent the feature observations of all objects and those of the q-th object at all times, and $\mathbf{z}_i^q$ denotes the feature observations of the q-th object at time i.

B. Problem Formulation

Consider a robot equipped with a camera moving in an unknown dynamic environment. The inputs of the SLOT process are the measurements from camera, and the outputs are both camera pose and the environment map, together with the poses and velocities of dynamic objects. Using probabilistic representation, the objective for the considered SLOT problem is to maximize the following posterior,

$$p(\mathbf{X}_k, \mathbf{O}_k, \mathbf{l} | \mathbf{Z}_k). \tag{1}$$

IV. Switching-coupled solution for SLOT

In this section, we detail the proposed switching-coupled solution for the considered SLOT problem. In particular, we first derive the concrete probability form of the proposed switching-coupled backend, and then a core objects classification criteria used in our algorithm is given.

A. Probability Derivation for Switching-coupled Backend

Before introducing the proposed method, we briefly review two existing popular back-end frameworks for the state-of-the-art solutions of SLOT problem, i.e., tightly-coupled method and loosely-coupled one. In the tightly-coupled method, the camera state is related to the measurements of all dynamic objects and static landmarks. Therefore, the performance of the system will be greatly affected if some dynamic objects exist with poor observation quality and without enough prior information (see Section I). On the other hand, in the loosely-coupled method, a key artificial assumption is imposed,

Assumption 1. *The posteriors of the camera state and the static landmarks' states are independent of the states of dynamic objects and landmarks, and also independent of their measurements, i.e.,* $p(\mathbf{X}_k, \mathbf{l}^0 | \mathbf{O}_k^{1:N}, \mathbf{l}^{1:N}, \mathbf{Z}_k) = p(\mathbf{X}_k, \mathbf{l}^0 | \mathbf{Z}_k^0)$.

Under Assumption 1, the SLOT problem is decomposed into two separate posteriors, one for the classical SLAM process and the other for the dynamic object tracking process [12]. However, the loosely-coupled method may not obtain the optimal result for the camera state estimation since all the measurements from dynamic objects are directly ignored according to Assumption 1. Moreover, the performance is unsatisfactory in the environments where the observation quality of static background is inferior, e.g. a vehicle on a bridge or inside a tunnel occluded by other vehicles driving alongside and failing to track static structure [8].

Motivated by these observations, in this section, we propose a novel switching-coupled back-end method, where dynamic objects are divided into "good" ones and "bad" ones. By imposing a more practical assumption that the camera state are independent with the measurements of "bad" dynamic objects, the considered SLOT problem can be also decomposed into two separate posteriors for practical applications, and the camera state is more likely to maintain an accurate and robust estimation compared with the pure tightly-coupled or loosely-coupled back-end solutions.

We divide the states of dynamic objects into "good" ones and "bad" ones, i.e., $\mathbf{O}_k = \{\mathbf{O}_k', \mathbf{O}_k''\}$, and the measurements and landmarks' states are correspondingly divided, i.e., $\mathbf{l} = \{\mathbf{l}', \mathbf{l}'', \mathbf{l}^0\}$, $\mathbf{Z}_k = \{\mathbf{Z}_k', \mathbf{Z}_k'', \mathbf{Z}_k^0\}$, and $\mathbf{z}_i = \{\mathbf{z}_i', \mathbf{z}_i'', \mathbf{z}_i^0\}$, where the superscripts "′" and "″" denote "good" and "bad", respectively. Before preceding on, the following assumption is first imposed.

Assumption 2. *The posteriors of the camera state, the states of static landmarks, and the states of good dynamic objects and landmarks, are independent of the states of bad dynamic objects and landmarks, and also independent of their measurements, i.e.,* $p(\mathbf{X}_k, \mathbf{O}_k', \mathbf{l}', \mathbf{l}^0 | \mathbf{O}_k'', \mathbf{l}'', \mathbf{Z}_k) = p(\mathbf{x}_k, \mathbf{O}_k', \mathbf{l}', \mathbf{l}^0 | \mathbf{Z}_k', \mathbf{Z}_k^0)$,

Remark 1. *Assumption 2 is a practical assumption due to the following reasons: (i) compared with the tightly-coupled solution, the states and the measurements of bad dynamic objects will not influence the state estimations of camera, static landmarks and good dynamic objects, and the robustness and accuracy of the whole system can be therefore improved. (ii) Compared with the loosely-coupled solution based on Assumption 1, Assumption 2 allows the estimation of camera state leveraging the measurements of good dynamic objects, instead of directly ignoring all the measurements from dynamic ones. Such a property is quite helpful in the situations that the classical SLAM are prone to failure, e.g., texture less or automatic driving scene where the front dynamic vehicle obstructs the view.*

Using the conditional probability theorem, the posterior (1) is derived as $p(\mathbf{X}_k, \mathbf{O}_k, \mathbf{l} | \mathbf{Z}_k) = p(\mathbf{X}_k, \mathbf{O}_k', \mathbf{l}', \mathbf{l}^0 | \mathbf{O}_k'', \mathbf{l}'', \mathbf{Z}_k) \cdot p(\mathbf{O}_k'', \mathbf{l}'' | \mathbf{Z}_k)$. It then follows from Assumption 2 that

$$p(\mathbf{X}_k, \mathbf{O}_k, \mathbf{l} | \mathbf{Z}_k) = p(\mathbf{X}_k, \mathbf{O}_k', \mathbf{l}', \mathbf{l}^0 | \mathbf{Z}_k', \mathbf{Z}_k^0) \cdot p(\mathbf{O}_k'', \mathbf{l}'' | \mathbf{Z}_k). \tag{2}$$

According to (2), the posterior (1) is decomposed into two separate posteriors, one simultaneous localization, mapping and good objects tracking (SLAMGOT) process, and the other bad objects tracking (BOT) process. In particular, for the SLAMGOT part, we can obtain

$$\begin{aligned} p(\mathbf{X}_k, \mathbf{O}_k', \mathbf{l}', \mathbf{l}^0 | \mathbf{Z}_k) &= \int\int p(\mathbf{X}_k, \mathbf{O}_k, \mathbf{l} | \mathbf{Z}_k) d\mathbf{l}'' d\mathbf{O}_k'' \\ &= \int\int p(\mathbf{X}_k, \mathbf{O}_k', \mathbf{l}', \mathbf{l}^0 | \mathbf{Z}_k', \mathbf{Z}_k^0) p(\mathbf{O}_k'', \mathbf{l}'' | \mathbf{Z}_k) d\mathbf{l}'' d\mathbf{O}_k'' \\ &= p(\mathbf{X}_k, \mathbf{O}_k', \mathbf{l}', \mathbf{l}^0 | \mathbf{Z}_k', \mathbf{Z}_k^0). \end{aligned} \tag{3}$$

For the BOT part, we can obtain

$$\begin{aligned} p(\mathbf{O}_k'', \mathbf{l}'' | \mathbf{Z}_k) &= \int\int\int\int p(\mathbf{X}_k, \mathbf{O}_k, \mathbf{l} | \mathbf{Z}_k) d\mathbf{l}^0 d\mathbf{l}' d\mathbf{O}_k' d\mathbf{X}_k \\ &= \int\int\int\int p(\mathbf{X}_k, \mathbf{O}_k', \mathbf{l}', \mathbf{l}^0 | \mathbf{Z}_k', \mathbf{Z}_k^0) \\ &\quad \cdot p(\mathbf{O}_k'', \mathbf{l}'' | \mathbf{Z}_k) d\mathbf{l}^0 d\mathbf{l}' d\mathbf{O}_k' d\mathbf{X}_k \\ &= \int p(\mathbf{O}_k'', \mathbf{l}'' | \mathbf{Z}_k) \cdot \overbrace{p(\mathbf{X}_k | \mathbf{Z}_k', \mathbf{Z}_k^0)}^{\textbf{Camera state estimation}} d\mathbf{X}_k, \end{aligned} \tag{4}$$

where the camera state estimation term is the marginal distribution of the joint distribution of (3), i.e., $p(\mathbf{X}_k | \mathbf{Z}_k', \mathbf{Z}_k^0) = \int\int\int p(\mathbf{X}_k, \mathbf{O}_k', \mathbf{l}', \mathbf{l}^0 | \mathbf{Z}_k', \mathbf{Z}_k^0) d\mathbf{l}^0 d\mathbf{l}' d\mathbf{O}_k'$.

For implementation, the SLAMGOT process is first performed according to (3), and we have the following maximum-a-posteriori (MAP) estimation,

$$\begin{aligned} (\mathbf{X}_k, \mathbf{O}_k', \mathbf{l}', \mathbf{l}^0)^*_{\mathbf{MAP}} &= \underset{\mathbf{X}_k, \mathbf{O}_k', \mathbf{l}', \mathbf{l}^0}{\arg\max}\ p(\mathbf{X}_k, \mathbf{O}_k', \mathbf{l}', \mathbf{l}^0 | \mathbf{Z}_k', \mathbf{Z}_k^0) \\ &= \underset{\mathbf{X}_k, \mathbf{O}_k', \mathbf{l}', \mathbf{l}^0}{\arg\max} \prod_{i=0}^{k} p(\mathbf{z}_i^0 | \mathbf{x}_i, \mathbf{l}^0) \prod_{i=0}^{k} p(\mathbf{z}_i' | \mathbf{x}_i, \mathbf{o}_i', \mathbf{l}') \\ &\quad \cdot \prod_{i=1}^{k} p(\mathbf{o}_i' | \mathbf{o}_{i-1}') \cdot p(\mathbf{x}_0, \mathbf{o}_0', \mathbf{l}'). \end{aligned} \tag{5}$$

Algorithm 1 Threshold−based evaluation strategy

```
Input: $\psi_{\mathbf{\Sigma}}$ , $\psi_O$, $\psi_A$ of the object.
Output: $label$ of the object: $Good$ or $Bad$.
  if $\psi_{\mathbf{\Sigma}} \in [0, th_1)$ or $\psi_O \in [0, th_2)$ or $\psi_A \in [0, th_3)$ then
    return $Bad$;
  else
    $\Omega = \beta_{\mathbf{\Sigma}} \cdot \psi_{\mathbf{\Sigma}} + \beta_O \cdot \psi_O + \beta_A \cdot \psi_A$, where $\beta_{\mathbf{\Sigma}}$, $\beta_O$ and
    $\beta_A$ are three balance factors;
    if $\Omega < th_4$ then
      return $Bad$;
    else
      return $Good$;
    end if
  end if
```

Then, the BOT process is performed based on the the camera states $\hat{\mathbf{X}}_k$ calculated in the SLAMGOT process,

$$(\mathbf{O}''_k, \mathbf{l}'')^*_{\mathbf{MAP}} = \underset{\mathbf{O}''_k, \mathbf{l}''}{\arg\max}\, p(\mathbf{O}''_k, \mathbf{l}'' | \mathbf{Z}_k, \hat{\mathbf{X}}_k)$$

$$= \underset{\mathbf{O}''_k, \mathbf{l}''}{\arg\max} \prod_{i=0}^{k} p(\mathbf{z}''_i | \hat{\mathbf{x}}_i, \mathbf{o}''_i, \mathbf{l}'') \prod_{i=1}^{k} p(\mathbf{o}''_i | \mathbf{o}''_{i-1}) \cdot p(\mathbf{o}''_0, \mathbf{l}''). \tag{6}$$

Finally, under Gaussian distribution assumption, the maximum-a-posteriors of (5) and (6) can be converted into the nonlinear least square problems and further solved by Gauss-Newton method or Levenberg-Marquardt method. The details can be found in [7][13].

B. Objects Classification Criteria

We note that in order to effectively perform the proposed switching-coupled backend, it is crucial to classify the observed dynamic objects into good ones and bad ones. We therefore introduce an effective classification criteria in which three practical evaluation factors are jointly considered, i.e., object uncertainty $\psi_{\mathbf{\Sigma}}$, observation quality ψ_O and prior information ψ_A, and a fast threshold-based evaluation strategy $\mathbf{E}(\psi_{\mathbf{\Sigma}}, \psi_O, \psi_A)$ described in Algorithm 1 is employed.

Object uncertainty $\psi_{\mathbf{\Sigma}}$. In fact, re-projection constraints of the features belonging to an object are crucial for state estimation of the object. For the observation of the j-th landmark belonging to the q-th object at time i, we have

$${}^j\mathbf{z}^q_i = \pi({}^c\mathbf{T}^i_w\, {}^w\mathbf{T}^i_q\, {}^q\mathbf{p}_j) = \pi({}^c\mathbf{T}^i_w\, {}^w\mathbf{p}^i_{qj}), \tag{7}$$

where π denotes camera projection function and ${}^w\mathbf{p}^i_{qj}$ represents the position of the landmark in the world frame at time i. Here, we omit the transformation between homogeneous and non-homogeneous coordinates. Therefore, under Gaussian distribution assumption, mean trace of covariance matrices of the states of all the landmarks belonging to the object is leveraged to approximately characterize the object uncertainty. In particular, for the q-th object at time i,

$$\psi_{\mathbf{\Sigma}} = \frac{\sum_{j \in M_{q_i}} 1/\mathbf{tr}({}^j_w\mathbf{\Sigma}^q_i)}{N_{M_{q_i}}}, \tag{8}$$

where $\mathbf{tr}(\cdot)$ denotes the trace of a matrix, and ${}^j_w\mathbf{\Sigma}^q_i$ represents the covariance matrix of ${}^w\mathbf{p}^i_{qj}$, M_{q_i} denotes the set of landmarks belonging to the object observed at time i, and $N_{M_{q_i}}$ represents the number of landmarks in M_{q_i}. According to (7), ${}^j_w\mathbf{\Sigma}^q_i$ actually reflects both the uncertainty of the landmark's position in the object frame and the uncertainty of the object's pose in the world frame. In the filter-based frameworks, the covariances of landmarks' positions and objects' poses can be directly obtained, and therefore the calculation of $\psi_{\mathbf{\Sigma}}$ is direct and simple. On the other hand, in the optimization-based frameworks, ${}^j_w\mathbf{\Sigma}^q_i$ should be extracted from the system information matrix from the last state estimation step while the computation burden is heavy. In order to simplify calculation, we use optimization times to approximately reflect the object uncertainty. In particular, $\psi_{\mathbf{\Sigma}}$ is calculated through the following formula,

$$\psi_{\mathbf{\Sigma}} = \frac{\sum_{j \in M_{q_i}} \mathbf{1}({}^jB^q_i > B_1)}{N_{M_{q_i}}} \cdot \mathbf{1}(B^q_i > B_2), \tag{9}$$

where ${}^jB^q_i$ and B^q_i represent the number of times that the landmark state and the object state have been optimized before time i, respectively, B_1 and B_2 denote two thresholds of the optimization times, and the function $\mathbf{1}(P) = \begin{cases} 1, & P \text{ is true} \\ 0, & \text{otherwise} \end{cases}$.

Observation quality ψ_O. For the q-th object at time i, its observation quality ψ_O is defined as

$$\psi_O = \begin{cases} s_{q_i}, & N_{M_{q_i}} \in [N_{th1}, +\infty) \\ s_{q_i} \cdot \dfrac{N_{M_{q_i}} - N_{th2}}{N_{th1} - N_{th2}}, & N_{M_{q_i}} \in [N_{th2}, N_{th1}) \\ 0, & N_{M_{q_i}} \in [0, N_{th2}), \end{cases} \tag{10}$$

where N_{th1} and N_{th2} denote two thresholds of the number of observed landmarks for an object, respectively. In addition, s_{q_i} represents matching similarity, i.e., $s_{q_i} = \frac{\sum_{j \in M_{q_i}} \mathbf{1}(\theta^j < \gamma)}{N_{M_{q_i}}} \cdot P_{q_i}$, where θ^j reflects matching degree between the j-th landmark and its corresponding observation (e.g., descriptor distance for descriptor-based feature matching methods and end-point error (EPE) for optical flow tracking methods), γ is a matching threshold, P_{q_i} represents tracking confidence of the object, and α_l and α_q are balance factors.

Prior information ψ_A. Last but not least, the prior information of an object is important for state estimation, including scale information s, motion model m, and prior pose information p measured by other approaches (e.g., GPS, laser range finder, and deep-learning-based 3D object detection methods). For example, if the scale information of the objects is known, their 2D measurement equations can be accurately established. In the second place, for the case when GPS measurements are available, corresponding GPS measurement constraints for these active objects can be leveraged to aid the state estimation of the whole system. In particular, ψ_A is defined as $\psi_A = \alpha_s C_s + \alpha_m C_m + \alpha_p C_p$, where α_s, α_m and α_p are weighting factors, and C_s, C_m and C_p reflect the accuracies of these three kinds of prior information.

Remark 2. *If all the dynamic objects are labeled as "good", the switching-coupled method is equivalent to the tightly-coupled one. On the other hand, if all dynamic objects are labeled as "bad", the switching-coupled method becomes the loosely-coupled one. In this sense, the tightly-coupled and loosely-coupled solutions are two special cases of the proposed switching-coupled back-end method.*

Remark 3. *The label of a dynamic object is switchable at different time instant. For example, a dynamic object is labeled as "bad" at the current time, but it can also be labeled as "good" in the future due to the improved observation quality, reduced uncertainty or new added prior information, and then their measurement constraints are used for SLAMGOT process.*

V. Implementation and Results

In this section, we first give the implementation details, where a complete SLOT system based on the proposed switching-coupled backend is proposed. Then, we detail the evaluation results of the proposed method on synthesized scenes, KITTI datasets [24] and real-world experiments.

A. Implementation

In order to evaluate the proposed switching-coupled back-end method, we construct a monocular SLOT system based on ORB-SLAM2 [19].The pipeline is illustrated in Fig.1. For each incoming frame, semantic bounding boxes of objects are detected using YOLOv3 [22], and ORB features [19] are extracted. According to semantic information, we distinguish the regions of dynamic and static in the image. Then, dynamic observation composed by 2D bounding boxes of objects and their features is associated with dynamic map composed by 3D objects and their landmarks through discriminative scale space tracker (DSST) [23] and feature matching. After that, dynamic objects composed by dynamic observation, dynamic map and prior information, are divided into "good" ones and "bad" ones according to the proposed objects classification criteria (Section IV-B). Next, SLAMGOT solver is first performed to optimize camera state, static landmarks' states and good objects' states, and then, BOT solver is performed to optimize the states of bad objects based on the obtained camera state calculated by SLAMGOT solver. In particular, we utilize the g2o [25] framework to implement the SLAMGOT and BOT solvers according to Section V-B. All the experiments have been run on a laptop with an i7-8750H CPU and a GTX 1060 Max-q GPU, where the GPU is only used for implementing YOLOv3.

B. Synthesized Scenes

A simulation environment is built for analysis, where an active tracking robot and two passive dynamic objects are included. They all move with 3D motions. In particular, the tracking robot is equipped with a monocular camera with an image resolution of 640 × 480. Each dynamic object is treated as a cuboid (its size is set to a common car size, i.e., length = 3.6 m, width = 1.6 m, height = 1.5 m) and a motion model of constant global linear velocity [6] is used. The features of the objects lie on the surface of the boundary and the static features are simulated around the motion space of the active robot. We simulate four different practical tracking cases: (i) ideal-case: more than 400 static landmarks and 60 dynamic landmarks belonging to an object are observed in each frame, and the feature observation is interfered by Gaussian white noise with mean value of 1 pixel; (ii) static-bad-case: within a certain period of time, the number of matched static features is very few (less than 15 in our implementation). In practice, this case often occurs in the automatic driving scene where the front vehicle obstructs the view. (iii) dynamic-bad-case: within a certain period of time, the observations of 50 features belonging to object 1 are interfered by Gaussian white noise with mean value of 5 pixel, and the prior information including scale and motion model of object 1 is unknown. (iv) both-bad-case: within different time periods, the aforementioned cases (ii) and (iii) occur successively, and the specific parameter configuration is similar with the ones of cases (ii) and (iii).

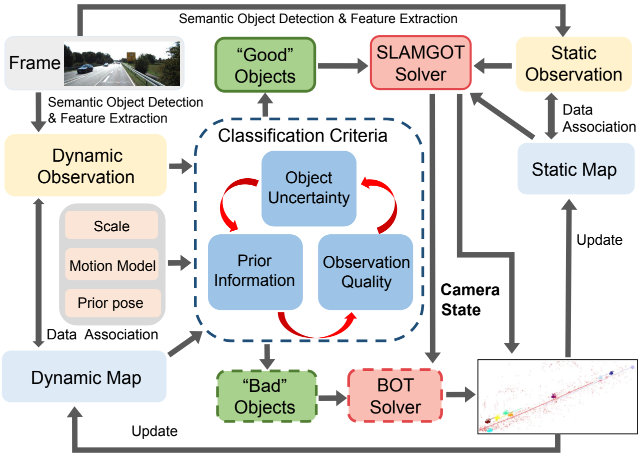


Fig. 1. Overview of the proposed SLOT system based on switching-coupled backend.

The proposed switching-coupled method is compared against the tightly-coupled method and the loosely-coupled method based on Assumption 1 [12]. The performance metrics used are the root mean squared errors (RMSE) of the absolute trajectory error (ATE) and the translational relative pose error (T.RPE), where the results of camera (robot) ego-motion, object 0 motion and object 1 motion are recorded, respectively. In addition, we also evaluate the running time consumption of each method, where each optimizer runs for 15 iterations. All the quantitative results are given in Table 1, and some qualitative results are illustrated in Fig. 2. In summery, we can see that for most of cases the proposed method achieves the best performance in terms of ATE

TABLE I
QUANTITATIVE COMPARISON ON SYNTHETIC SCENES.

Condition	Tightly-coupled			Loosely-coupled			Proposed		
	ATE(m)	T.RPE(m)	Time(s)	ATE(m)	T.RPE(m)	Time(s)	ATE(m)	T.RPE(m)	Time(s)
Case(i)	0.45/0.45/0.47	**0.03/0.02/0.02**	0.111	0.40/0.41/**0.41**	0.05/0.06/0.06	**0.071**	**0.38/0.39**/0.42	0.04/0.03/0.03	0.082
Case(ii)	0.50/0.50/0.51	0.15/0.15/0.15	0.105	27.3/27.2/26.7	0.80/0.81/0.82	**0.072**	**0.47/0.47/0.50**	**0.09/0.09/0.10**	0.075
Case(iii)	1.10/1.90/3.85	0.14/0.11/**0.28**	0.109	**0.38/0.55/3.26**	0.06/**0.06**/0.29	**0.074**	0.40/**0.55**/3.28	**0.05**/0.07/**0.28**	0.078
Case(iv)	0.92/0.98/3.74	0.31/0.32/0.40	0.106	26.6/26.2/27.5	0.21/0.81/0.88	**0.072**	**0.40/0.32/3.27**	**0.09/0.10/0.37**	0.079

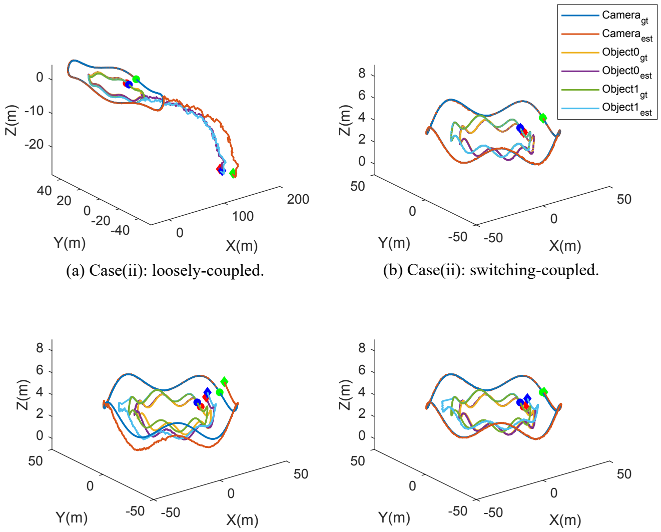
"(·)/(·)/(·)/" represent the corresponding results of camera, object 0 and object 1, respectively.

TABLE II
EGO MOTION COMPARISON ON KITTI SEQUENCES.

Sequence	ORB-SLAM2[19]		DynaSLAM[1]		DynSLAM[2]		SLAMMOT[12]		CubeSLAM[10]		Proposed	
	ATE(m)	T.RPE(m)	ATE(m)	T.RPE(m)	ATE(m)	T.RPE(m)	ATE(m)	T.RPE(m)	ATE(m)	T.RPE(m)	ATE(m)	T.RPE(m)
0926-0011	0.15	1.00	0.64	1.08	-	-	0.13	1.01	**0.12**	**0.99**	0.16	1.03
0926-0013	7.95	1.68	*	*	1.97	**1.41**	0.63	1.72	0.49	1.71	**0.37**	1.51
0926-0014	4.90	1.84	**0.73**	1.99	5.98	2.73	3.65	1.86	5.42	1.84	1.64	**1.61**
0926-0015	0.66	1.80	0.31	1.69	-	-	**0.26**	**1.65**	1.99	1.86	0.47	1.70
0926-0018	0.25	0.54	0.30	1.00	-	-	0.22	0.54	0.24	0.54	**0.19**	**0.51**
0926-0032	6.86	2.08	1.70	2.11	-	-	2.98	2.13	9.07	2.47	**1.52**	**1.75**
0926-0036	**1.36**	1.36	7.71	1.36	-	-	3.51	1.36	7.60	1.36	**1.36**	**1.33**
0926-0056	1.10	2.02	1.35	2.02	-	-	0.98	**0.61**	5.39	2.02	**0.57**	1.02
0926-0059	0.91	0.93	0.31	0.93	-	-	0.49	0.90	0.93	0.93	**0.25**	**0.85**
0929-0004	1.25	1.26	**0.54**	**1.11**	2.59	2.03	0.72	1.21	1.00	1.25	0.88	1.24

"*" stands for algorithm failure, and "-" denotes that the algorithm is not evaluated due to the missing preprocessing data.

and T.RPE. Especially in cases (ii), (iii) and (iv), tightly-coupled and loosely-coupled methods almost fail due to the different kinds of challenges, while the proposed method can still achieve accurate pose estimations for the camera and dynamic objects. In addition, we can see that the mean time consumption of the proposed method is close to that of the loosely-coupled one, which indicates that the proposed method has a comparable computational efficiency and better robustness and accuracy.



(a) Case(ii): loosely-coupled.

(b) Case(ii): switching-coupled.

(c) Case(iii): tightly-coupled.

(d) Case(iii): switching-coupled.

Fig. 2. The estimated trajectory comparisons between the proposed switching-coupled backend with the tightly-coupled and loosely-coupled ones in case (ii) and case (iii). The total path lengths of the robot, object 0 and object 1 are 255.6 m, 142.4 m and 151.6 m, respectively. The circle and diamond denote the start and end of a trajectory, respectively. The subscripts "gt" and "est" denote the ground truth and the estimation result, respectively.

C. KITTI Datasets

The proposed method is compared with state-of-the-art systems including ORB-SLAM2 [19], DynaSLAM [1], DynSLAM [2], SLAMMOT[12], CubeSLAM [10] on KITTI datasets. In particular, we demonstrate the strength of our method in 10 representative raw sequences, in which multiple dynamic cars can be observed over a long time and the ground truth of camera poses are provided by GPS/INS. The prior scale of vehicles and a nonholonomic wheel motion model [13] are used in SLAMMOT, CubeSLAM and the proposed system. Table. II shows the quantitative results of camera ego-motion after scale adjustment and Fig. 3 shows a visualized sample of the proposed method. We can see that for most of sequences, the proposed method has the minimum translation error compared with the other methods. In particular, ORB-SLAM2 is based on a static world assumption and therefore the performance is degraded in dynamic environments. DynaSLAM is a dynamic SLAM system which removes dynamic features by using mask R-CNN [21] to generate object masks, while it cannot estimate the trajectories of dynamic objects. Note that DynSLAM is a stereo SLOT system which maintains a dense dynamic map. The underlying sparse scene flow estimation is based on a frame-to-frame visual odometry *libviso*, which leads

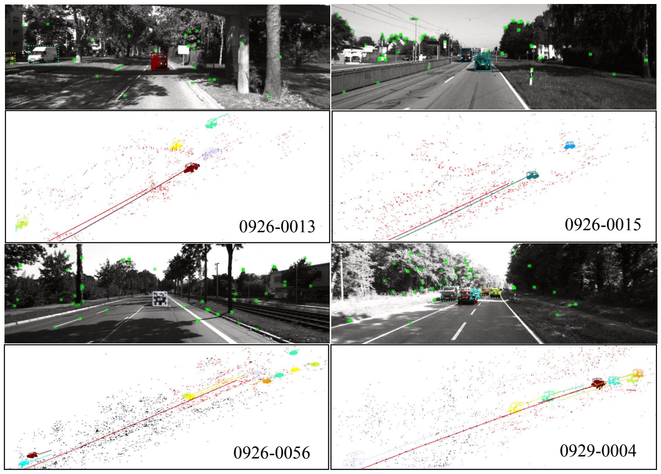


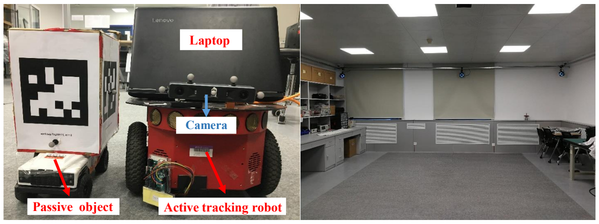
Fig. 3. A visualized sample of the proposed SLOT system on KITTI raw datasets. The solid and dotted line boxes in the image represent "good" objects and "bad" objects labeled by the proposed objects classification criterion, respectively.

to remarkable drift over long travel distances (e.g., 0926-0014) [11]. CubeSLAM is a tightly-coupled monocular SLOT system, and thus the poor measurements of some dynamic objects negatively affect state estimation of the system. SLAMMOT is a parallel SLAM and object tracking method [12]. We realize it by using the same frontend in the proposed SLOT system, and only the backend is replaced by a loosely-coupled one. Therefore, it depends on the quality of the returned static map and may perform poorly in environments with insufficient number of reliable static structure (e.g., 0926-0032 and 0926-0036). In contrast, the proposed method achieves comparable or even better results than all the previous methods thanks to the proposed switching-coupled strategy and objects classification criteria.

D. Real-world Experiments

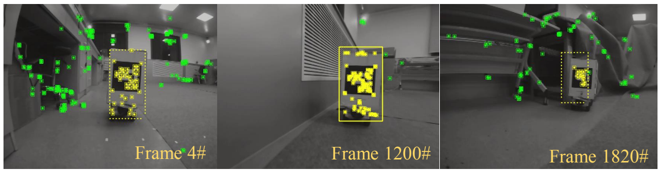
1) Indoor: The indoor experimental setup is displayed in Fig. 4 (a). Active tracking robot is a pioneer robot equipped with a Mynteye binocular camera D1000-IR-120[1], and the passive object is an off-road toy vehicle equipped with a cuboid shell. Meanwhile, an OptiTrack[2] motion capture system as shown in Fig. 4 (b) is used to record the three-dimensional positions as the groundtruth.

We compare the proposed switching-coupled backend against the tightly-coupled and loosely-coupled methods. To compare fairly, we replace the backend in the proposed SLOT system with the ones of loosely-coupled and tightly-coupled, while the frontend remains unchanged. Fig. 6 shows the trajectory comparisons among these three methods in 3D space. Snapshots taken during the experiments are shown in Fig. 5. We can see that the translation error of the proposed method is much smaller than the ones of the tightly-coupled and loosely-coupled algorithms. Note that the trajectory of



(a) Experimental setup. (b) OptiTrack.

Fig. 4. The indoor experimental setup and an OptiTrack motion capture system.



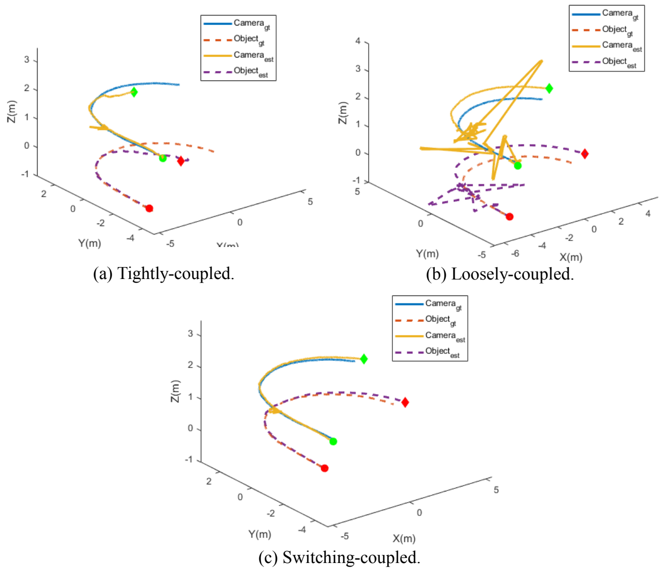
(a) (b) (c)

Fig. 5. Snapshots taken during the indoor experiments. (a) The object is initially labeled as "bad" due to the high uncertainty. (b) Lots of static features are lost because of the sudden rotation of the camera and the occupying the part field of vision by smooth walls, while the object is labeled as "good". (c) The object is labeled as "bad" due to the bad observation quality.

the loosely-coupled method is very messy within a period of time. The reason may be that the sudden rotation of the camera and the occupying the part field of vision by smooth walls lead to the matching failure of lots of static features (see Fig. 5 (b)). However, the observation quality of the passive object is good at the same time, and the other two methods can therefore work properly. This indicates that dynamic object tracking can also benefit the classical SLAM as long as a proper back-end method is selected.

2) Outdoor: The outdoor experimental setup is displayed in Fig. 7. The active tracking robot and the passive object are two tracer unmanned ground vehicles (UGVs), respectively. A RTK-GPS is attached to the passive object to allow for groundtruth comparison. Based on the setup, we collect an outdoor dataset with a total length of 364.9 m (object's GPS trajectory) on the campus of Tsinghua University. The proposed SLOT system is also compared with ORB-SLAM2 and CubeSLAM. In order to get an absolute scale for these three monocular systems, the camera height is provided to scale the map. The similar method is also used in [10]. The paths of the tracking robot, passive object, and RTK-GPS are overlaid onto satellite imagery as shown in Fig. 8. We can see that ORB-SLAM2 and CubeSLAM almost fail on this outdoor dataset. The reason for ORB-SLAM2 may be its static scene assumption. For CubeSLAM, the system performance decreases sharply once the object observation is poor due to the tightly-coupled backend. On the other hand, it only optimizes the object state in keyframes, which makes that lots of object information in normal frames are ignored and leads to over-dependent on the results of 3D object detection from the frontend. In contrast, the proposed

[1] http://www.myntai.com/cn/mynteye/
[2] http://www.optitrack.com/



(a) Tightly-coupled.

(b) Loosely-coupled.

(c) Switching-coupled.

Fig. 6. The trajectory comparisons in indoor experiments. The proposed switching-coupled backend is compared with the ones of tightly-coupled and loosely-coupled. For clear observation, Z-coordinate of the camera position is manually shifted upward by 1.5m.

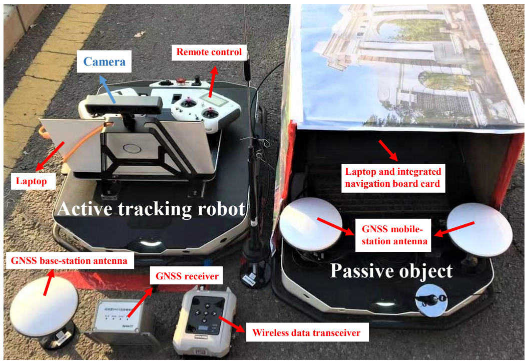

Camera
Remote control
Laptop
Active tracking robot
Laptop and integrated navigation board card
GNSS mobile-station antenna
Passive object
GNSS base-station antenna
GNSS receiver
Wireless data transceiver

Fig. 7. The outdoor experimental setup.

method can work properly and maintain correct camera and object trajectories, which indicates that the proposed method has better robustness and accuracy.

VI. Conclusions

In this paper, we propose a novel switching-coupled backend solution for simultaneous localization and object tracking problem. Based on the switching strategy and objects classification criteria, the proposed method reduces the influence of "bad" dynamic objects and leverages the measurement constraints of "good" dynamic objects to improve the performance of the overall system. Simulations and experiments demonstrate that the proposed method achieves better accuracy and robustness even compared with the existing state-of-the-art systems do.

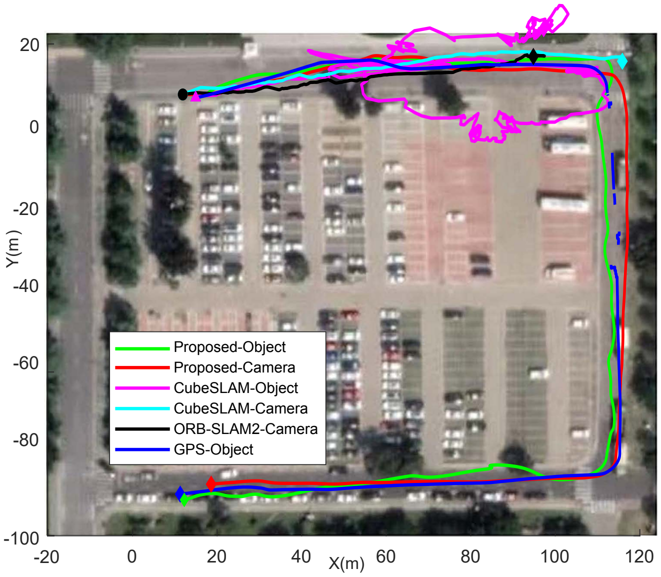


Fig. 8. Top-down view of the trajectories generated by the proposed method, ORB-SLAM2 and CubeSLAM. The circle and triangle denote the start of the trajectories of camera and object, respectively, and diamond denotes the end of a trajectory. There is a missing part of the GPS groundtruth due to the transmission interference by the occlusion of nearby big trucks.

References

[1] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 4076-4083, Oct. 2018.

[2] I. A. Barsan, P. Liu, M. Pollefeys, and A. Geiger, "Robust dense mapping for large-scale dynamic environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 7510-7517.

[3] W. Tian, M. Lauer, and L. Chen, "Online multi-object tracking using joint domain information in traffic scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 374-384, Jan. 2020.

[4] J. Luiten, T. Fischer, and B. Leibe, "Track to reconstruct and reconstruct to track," *IEEE Robot.Automat. Lett.*, vol. 5, no. 4, pp. 1803-1810, Apr. 2020.

[5] J. Huang, S. Yang, Z. Zhao, Y. Lai, and S. Hu, "ClusterSLAM: A SLAM backend for simultaneous rigid body clustering and motion estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5875-5884.

[6] K. Eckenhoff, Y. Yang, P. Geneva, and G. Huang, "Tightly-coupled visual-inertial localization and 3-D rigid-body target tracking," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1541-1548, Apr. 2019.

[7] M. Chojnacki and V. Indelman, "Vision-based dynamic target trajectory and ego-motion estimation using incremental light bundle adjustment," *Int. J. Micro Air Veh.*, vol. 10, no. 2, pp. 157-170, 2018.

[8] M. Henein, J. Zhang, R. Mahony, and V. Ila, "Dynamic SLAM: The need for speed," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 2123-2129.

[9] K. Lin and C. Wang, "Stereo-based simultaneous localization, mapping and moving object tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 3975-3980.

[10] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D object SLAM," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 925-938, Aug. 2019.

[11] J. Huang, S. Yang, T. Mu, and S. Hu, "ClusterVO: Clustering moving instances and estimating visual odometry for Self and surroundings," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2165-2174.

[12] C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *Int. J. Robot. Res.*, vol. 26, no. 9, pp. 3376-3385, 2007.

[13] P. Li and T. Qin, "Stereo vision-based semantic 3D object and ego-motion tracking for autonomous driving," in *Proc. Eur. Conf. Comput. Vis.*, Munich, 2018, pp. 646-661.

[14] K. Qiu, T. Qin, W. Gao, and S. Shen, “Tracking 3-D motion of dynamic objects using monocular visual-inertial sensing,” *IEEE Trans. Robot.*, vol.35, no. 4, pp. 799-816, Aug. 2019.

[15] J. Chen, T. Liu, and S. Shen, “Tracking a moving target in cluttered environments using a quadrotor,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 446-453.

[16] H. Lim and S. N. Sinha, “Monocular localization of a moving person onboard a quadrotor MAV,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 2182ÍC2189.

[17] J. Zhang, M. Henein, R. Mahony, and V. Ila, “Robust ego and object 6-DoF motion estimation and tracking,” 2020, *arXiv:2007.13993*.

[18] A. I. Mourikis and S. I. Roumeliotis, “A multi-state constraint Kalman filter for vision-aided inertial navigation,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2007, pp. 3565-3572.

[19] 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.

[20] T. Qin, P. Li, and S. Shen, “VINS-Mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004-1020, Aug. 2018.

[21] 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.

[22] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” 2018, *arXiv:1804.02767*.

[23] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, “Discriminative scale space tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1561-1575, Aug. 2017.

[24] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354ÍC3361.

[25] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “G2o: A general framework for graph optimization,” in *Proc. IEEE Int. Conf. Robot. Automat*, 2011, pp. 3607ÍC3613.