A Right Invariant Extended Kalman Filter for Object based SLAM

Yang Song[1], Zhuqing Zhang[2], Jun Wu[2], Yue Wang[2], Liang Zhao[1], Shoudong Huang[1]

***Abstract*—With the recent advance of deep learning based object recognition and estimation, it is possible to consider object level SLAM where the pose of each object is estimated in the SLAM process. In this paper, based on a novel Lie group structure, a right invariant extended Kalman filter (RI-EKF) for object based SLAM is proposed. The observability analysis shows that the proposed algorithm automatically maintains the correct unobservable subspace, while standard EKF (Std-EKF) based SLAM algorithm does not. This results in a better consistency for the proposed algorithm comparing to Std-EKF. Finally, simulations and real world experiments validate not only the consistency and accuracy of the proposed algorithm, but also the practicability of the proposed RI-EKF for object based SLAM problem. The MATLAB code of the algorithm is made publicly available.**

***Index Terms*—SLAM, Localization, Mapping**

I. INTRODUCTION

DURING the past decade, visual sensors are very popular due to the properties of low cost and rich information, and various kinds of frameworks are designed for visual SLAM [1]-[3]. However, most of these works only utilize low-level features such as points [1], lines [4] or planes [5] and neglect the high-level features such as objects which contain strong geometric constraints [6]. High-level features have many advantages over low-level features, including broader perspective loop closures, longer feature tracking and considering the intrinsic constraints among low-level features [6][7].

Many existing object level SLAM systems have shown the benefits of using object level features. For examples, Salas-Moreno et al. in [8] propose a paradigm for 3D object SLAM system, and present its advantages in a vast compression of map storage, large scale loop closure, and relocalisation. Gálvez-López et al. [9] show that their object SLAM system can yield more accurate maps than RGB-D SLAM. Yang et al. in [10] combine 2-D and 3-D object detection with SLAM pose estimation together, for both static and dynamic environments. Their proposed system achieves better pose estimation on many datasets comparing to the point feature based SLAM system. The back-end techniques used in almost all the existing object level SLAM systems are optimization based methods, such as pose graph optimization [8][9] and bundle adjustment [10]. This is mainly because optimization based methods not only achieve the best accuracy, but also use many ready-made tools (for example, G2o [11] and Ceres [12]). However, optimization based methods require much more computations than filter based methods when the trajectory of robot is very long. Thus they are not very suitable for deployment on lightweight platforms. Therefore, developing filter based methods are important. Nevertheless, to the best of our knowledge, works focused on the filter based SLAM frameworks that consider the object features are still blank. One of the reasons is that the conventional filter based SLAM (like standard EKF) usually suffers from the problem of inconsistency. And a consistent filter based estimator requires elaborate modelling and algorithmic design.

An inconsistent filter based SLAM system will underestimate the uncertainty of the estimated state, which gradually leads to poor results and even makes the algorithm diverge. The discovery of inconsistency of point feature based EKF SLAM can date back to 2001 [13]. In the last decades, many works focus on analyzing the cause of inconsistency and proposing improvement algorithms to alleviate the inconsistency of the system [13]-[17]. According to [16], the explanation of such phenomena is that EKF linearizes the system model at the latest estimated values so that Jacobians of the process and observation functions are not evaluated at the same state value. Furthermore, Huang et al. in [17] analyze the observability of the system and argues that the ever-changing linearization points break the unobservable subspace of the system, therefore spurious information along the unobservable direction is introduced to the system, which makes the estimate inconsistent. To solve this problem, first Jacobian estimate (FEJ) [18] and observability constraint (OC) [19] methods are proposed.

On the contrary, instead of adding artificial constraint into the estimator, algorithms designed by Lie group theory are found to have the potential to naturally handle invariance including observability constraints in point feature based SLAM algorithms [20]-[22]. Exploiting the properties of invariant tangent vector field, Lie group theory generates the invariant-EKF methodology [23]-[25], which is firstly applied to EKF-SLAM in [25]. Recently, EKF designed on Lie groups has become popular in filter based SLAM [22][26]-[28], as many such algorithms perform well in terms of consistency and convergence. Based on a specific Lie group representation, a right invariant EKF (RI-EKF) algorithm is proposed for 2D point feature based SLAM to alleviate the inconsistency and make the state estimates obtained more accurate [21]. Furthermore, the convergence and consistency for RI-EKF point feature based SLAM in 3D environment are analyzed in [20], showing the advantages of the invariant algorithm for the point based SLAM problem.

However, all filter based SLAM methods mentioned above

This work was supported in part by the National Nature Science Foundation of China under Grant 61903332, 62173293, and UTS-CSC International Research Scholarship.

[1]Yang Song, Liang Zhao and Shoudong Huang are with Robotics Institute, University of Technology Sydney, Australia. Email:Yang.Song-4@student.uts.edu.au

[2]Zhuqing Zhang, Jun Wu, and Yue Wang are with the State key Laboratory of Industrial Control and Technology, Zhejiang University, P.R. China.

The MATLAB code is available at
https://github.com/YangSONG-SLAM/RIEKF_objectSLAM

Digital Object Identifier (DOI): 10.1109/LRA.2021.3139370.

only consider the point features, and how to handle the object features (poses) consistently is still an untouched problem. In this paper, we propose a consistent EKF algorithm for object based SLAM. To be specific, the contributions of this paper are shown as follows:

- An invariant EKF is designed on a new Lie group for SLAM with object features.
- The observability analysis shows that our proposed algorithm naturally maintains the correct unobservable subspace.
- The effectiveness of our proposed algorithm is validated via simulations and real-data experiments.

Notations: In this paper, bold lower-case and upper-case letters are reserved for vectors and matrices/elements in the Lie group, respectively. The notation $\mathbb{SO}(3)$ represents 3D special rotation group, consisting of all rotation transformations in $\mathbb{R}^3$. $\mathfrak{so}(3)$ is the Lie algebra of $\mathbb{SO}(3)$, containing all 3×3 skew symmetric matrices. $(\cdot)^\wedge$ represents the skew symmetric operator that transforms a 3-dimensional vector into a skew symmetric matrix. $\exp^G$ represents the exponential map on a Lie group G. $\log^G$ is the inverse of exponential map on a Lie group G. $N(\mathbf{0},\mathbf{P})$ represents a zero mean Gaussian distribution with covariance $\mathbf{P}$.

II. Object based SLAM Problem

An object feature considered in this work is represented as a 3D pose of the object in the environment. A robot moves in an unknown 3D environment and observes some object features. The object based SLAM focuses on estimating the current robot pose and the poses of all the object features using the process model and observation model.

1) State Space: An object feature is defined as

$$(\mathbf{R}^f, \mathbf{p}^f), \tag{1}$$

where $\mathbf{R}^f \in \mathbb{SO}(3)$ and $\mathbf{p}^f \in \mathbb{R}^3$ are the rotation and position of feature, respectively.

The set consists of all states combining with the robot pose and K observed features is denoted as $\mathcal{G}_K$, where

$$\begin{aligned}\mathcal{G}_K = \big\{&(\mathbf{R}^r, \mathbf{R}^{f_1}, \cdots, \mathbf{R}^{f_K}, \mathbf{p}^r, \mathbf{p}^{f_1}, \cdots, \mathbf{p}^{f_K})\\ &|\mathbf{R}^r, \mathbf{R}^{f_j} \in \mathbb{SO}(3),\ \mathbf{p}^r, \mathbf{p}^{f_j} \in \mathbb{R}^3\big\},\end{aligned} \tag{2}$$

$\mathbf{R}^r, \mathbf{R}^{f_j} \in \mathbb{SO}(3)$ represent the rotations of robot and the j-th feature respectively, and $\mathbf{p}^r, \mathbf{p}^{f_j} \in \mathbb{R}^3$ represent the positions of robot and the j-th feature respectively, all described in the global coordinate system[1].

2) Process Model and Observation Model: Since $\mathbb{R}^3 \cong \mathfrak{so}(3)$, for simplification, we can define the exponential map of $\mathbb{SO}(3)$ as follows: For $\boldsymbol{\xi} \in \mathbb{R}^3$,

$$\begin{aligned}\exp^{\mathbb{SO}(3)}: \quad \mathbb{R}^3 &\rightarrow \mathbb{SO}(3)\\ \boldsymbol{\xi} &\rightarrow \textstyle\sum_{k=0}^{\infty} \frac{(\boldsymbol{\xi}^\wedge)^k}{k!}.\end{aligned} \tag{3}$$

[1]Sometimes we use the notation $\mathcal{G}$ instead of $\mathcal{G}_K$ for brevity. Also, in the remaining of this paper, without losing generality, we assume that there is only one object feature, i.e. $K=1$, to simplify the equations.

The following first-order integration scheme of discrete noisy process model is widely used [18][21]:

$$\begin{aligned}&\mathbf{X}_{n+1} = f(\mathbf{X}_n, \mathbf{U}_n, \mathbf{w}_n)\\ &= (\mathbf{R}^r_n \exp^{\mathbb{SO}(3)}(\mathbf{w}^R_n)\mathbf{R}^u_n, \mathbf{R}^f_n, \mathbf{p}^r_n + \mathbf{R}^r_n(\mathbf{p}^u + \mathbf{w}^p_n), \mathbf{p}^f_n),\end{aligned} \tag{4}$$

where $\mathbf{X}_i = (\mathbf{R}^r_i, \mathbf{R}^f_i, \mathbf{p}^r_i, \mathbf{p}^f_i)$ is the state at time step i, $i = n, n+1$, $\mathbf{U}_n = (\mathbf{R}^u_n, \mathbf{p}^u_n)$ is the odometry, $\mathbf{w}_n = ((\mathbf{w}^R_n)^\top, (\mathbf{w}^p_n)^\top)^\top (\in \mathbb{R}^6) \sim N(\mathbf{0}, \boldsymbol{\Sigma}_n)$ is the odometry noise.

After the object detection and matching from the SLAM front-end, the observation can be regarded as relative poses of object features in the current robot frame. Then the observation model for the object feature, $(\mathbf{R}^f, \mathbf{p}^f)$, in the $(n+1)$-th robot frame, $(\mathbf{R}^r_{n+1}, \mathbf{p}^r_{n+1})$, can be described as

$$\mathbf{Z} = h(\mathbf{X}_{n+1}, \mathbf{v}_{n+1}) = (\mathbf{R}^z, \mathbf{p}^z), \tag{5}$$

where

$$\begin{aligned}\mathbf{R}^z &= \exp^{\mathbb{SO}(3)}(\mathbf{v}^R_{n+1})(\mathbf{R}^r_{n+1})^\top \mathbf{R}^f_{n+1},\\ \mathbf{p}^z &= (\mathbf{R}^r_{n+1})^\top(\mathbf{p}^f_{n+1} - \mathbf{p}^r_{n+1}) + \mathbf{v}^p_{n+1},\end{aligned}$$

and $\mathbf{v}_{n+1} = ((\mathbf{v}^R_{n+1})^\top, (\mathbf{v}^p_{n+1})^\top)^\top (\in \mathbb{R}^6) \sim N(\mathbf{0}, \boldsymbol{\Omega_{n+1}})$ is the observation noise.

III. RI-EKF for Object based SLAM

A. RI-EKF framework for object based SLAM

1) A novel Lie group structure on state space: For all $(\mathbf{R}_i, \mathbf{R}^f_i, \mathbf{p}^r_i, \mathbf{p}^f_i) \in \mathcal{G}$, $i = 1, 2$, an operator $\oplus$ is defined by

$$\begin{aligned}&(\mathbf{R}^r_1, \mathbf{R}^f_1, \mathbf{p}^r_1, \mathbf{p}^f_1) \oplus (\mathbf{R}^r_2, \mathbf{R}^f_2, \mathbf{p}^r_2, \mathbf{p}^f_2) =\\ &(\mathbf{R}^r_1\mathbf{R}^r_2, \mathbf{R}^f_1\mathbf{R}^f_2, \mathbf{R}^r_1\mathbf{p}^r_2 + \mathbf{p}^r_1, \mathbf{R}^r_1\mathbf{p}^f_2 + \mathbf{p}^f_1).\end{aligned} \tag{6}$$

Then we can check that equiped with $\oplus$ defined in (6), the state space $\mathcal{G}$ becomes a Lie group and is isomotric to $\mathbb{SE}_{K+1}(3) \times (\mathbb{SO}(3))^K$, where K (set as 1 for simplicity) is the number of observed features. The Lie group $\mathbb{SE}_{K+1}(3)$, defined in [20][21], plays a significant role in RI-EKF for point feature based SLAM. The notation $\ominus$, the minus of $\oplus$, is defined by $\mathbf{X}_a \ominus \mathbf{X}_b = \mathbf{X}_a \oplus \mathbf{X}_b^{-1}$ and $\mathbf{X}_b \oplus \mathbf{X}_b^{-1} = \mathbf{X}_b^{-1} \oplus \mathbf{X}_b = (\mathbf{I}, \mathbf{I}, \mathbf{0}, \mathbf{0})$. Denote $\mathfrak{g}$ as the Lie algebra of $\mathcal{G}$. And we have $\mathfrak{g} \cong \mathfrak{se}_{K+1}(3) \times (\mathfrak{so}(3))^K \cong \mathbb{R}^{6+6K}$. Therefore, the form of an element $\boldsymbol{\xi}$ in Lie algebra $\mathfrak{g}$ can be constructed as

$$\boldsymbol{\xi}^\top = ((\boldsymbol{\xi}^{R^r})^\top, (\boldsymbol{\xi}^{R^f})^\top, (\boldsymbol{\xi}^{p^r})^\top, (\boldsymbol{\xi}^{p^f})^\top), \tag{7}$$

where $\boldsymbol{\xi}^{R^r}, \boldsymbol{\xi}^{R^f}, \boldsymbol{\xi}^{p^r}, \boldsymbol{\xi}^{p^f} \in \mathbb{R}^3$. The exponential map $\exp^{\mathcal{G}}$ on this Lie group can be defined by

$$\begin{aligned}\exp^{\mathcal{G}}(\boldsymbol{\xi}) =&(\exp^{\mathbb{SO}(3)}(\boldsymbol{\xi}^{R^r}), \exp^{\mathbb{SO}(3)}(\boldsymbol{\xi}^{R^f})\\ &J_l(\boldsymbol{\xi}^{R^r})\boldsymbol{\xi}^{p^r}, J_l(\boldsymbol{\xi}^{R^r})\boldsymbol{\xi}^{f}),\end{aligned} \tag{8}$$

where

$$J_l(\boldsymbol{\xi}^{R^r}) = \sum_{k=0}^{\infty} \frac{((\boldsymbol{\xi}^{R^r})^\wedge)^k}{(k+1)!}. \tag{9}$$

Then an error state $\boldsymbol{\xi}$ for an estimated state $\hat{\mathbf{X}}$ satisfies

$$\mathbf{X} = \exp^{\mathcal{G}}(\boldsymbol{\xi}) \oplus \hat{\mathbf{X}}, \tag{10}$$

where $\mathbf{X}$ represents the true state.

2) Propagation: Based on the Lie group structure introduced above, the process model (4) becomes

$$\mathbf{X}_{n+1} = \mathbf{X}_n \oplus (\exp^{\mathbb{SO}(3)}(\mathbf{w}_n^R)\mathbf{R}_n^u, \mathbf{I}_3, \mathbf{p}_n^u + \mathbf{w}_n^p, \mathbf{0}_{3\times 1}). \quad (11)$$

The predicted state, $\mathbf{X}_{n+1|n}$, by propagation is computed by

$$\mathbf{X}_{n+1|n} = (\mathbf{R}_{n|n}^r\mathbf{R}_n^u, \mathbf{R}_{n|n}^f, \mathbf{R}_{n|n}^r\mathbf{p}_n^u + \mathbf{p}_{n|n}^r, \mathbf{p}_{n|n}^f). \quad (12)$$

where $\mathbf{X}_{n|n} = (\mathbf{R}_{n|n}^r, \mathbf{R}_{n|n}^f, \mathbf{p}_{n|n}^r, \mathbf{p}_{n|n}^f)$ is the updated state at time n. And the estimated error $\boldsymbol{\xi}_{n+1|n}$ by propagation is

$$\begin{aligned}\boldsymbol{\xi}_{n+1|n} &\doteq \log(\mathbf{X}_{n+1} \ominus \mathbf{X}_{n+1|n}) \\ &\approx \mathbf{F}_n\boldsymbol{\xi}_{n|n} + \mathbf{G}_n\mathbf{w}_n,\end{aligned} \quad (13)$$

where $\boldsymbol{\xi}_{n|n} \sim N(\mathbf{0}, \boldsymbol{P}_n)$ is the estimation error for $\mathbf{X}_{n|n}$, $\mathbf{w}_n = ((\mathbf{w}_n^R)^\top, (\mathbf{w}_n^p)^\top)^\top$ is the odometry noise, the coefficient matrices $\mathbf{F}_n$ and $\mathbf{G}_n$ in RI-EKF are

$$\begin{aligned}&\mathbf{F}_n = \mathbf{I}_{6+6K}, \\ &\mathbf{G}_n = \begin{bmatrix} \mathbf{R}_{n|n}^r & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ (\mathbf{p}_{n|n}^r + \mathbf{R}_{n|n}^r\mathbf{p}_n^u)^\wedge\mathbf{R}_{n|n}^r & \mathbf{R}_{n|n}^r \\ (\mathbf{p}_{n|n}^f)^\wedge\mathbf{R}_{n|n}^r & \mathbf{0}_{3\times 3} \end{bmatrix}.\end{aligned} \quad (14)$$

Then, the covariance matrix of the state $\mathbf{X}_{n+1|n}$ by propagation is

$$\mathbf{P}_{n+1|n} = \mathbf{F}_n\mathbf{P}_n\mathbf{F}_n^\top + \mathbf{G}_n\boldsymbol{\Sigma}_n\mathbf{G}_n^\top. \quad (15)$$

3) Update: Suppose $\mathbf{Z}$ is an observation in (5). By introducing a new minus operator $\boxminus$ for the observation, innovation $\mathbf{y}$ is defined as

$$\begin{aligned}\mathbf{y} &= \begin{bmatrix} \mathbf{y}^R \\ \mathbf{y}^p \end{bmatrix} = \mathbf{Z} \boxminus [(\hat{\mathbf{R}}^r)^\top\hat{\mathbf{R}}^f, (\hat{\mathbf{R}}^r)^\top(\hat{\mathbf{p}}^f - \hat{\mathbf{p}}^r)] \\ &\doteq \begin{bmatrix} \log^{\mathbb{SO}(3)}(\exp^{\mathbb{SO}(3)}((\mathbf{v}^R)^\wedge)(\mathbf{R}^r)^\top\mathbf{R}^f(\hat{\mathbf{R}}^f)^\top\hat{\mathbf{R}}^r) \\ (\mathbf{R}^r)^\top(\mathbf{p}^f - \mathbf{p}^r) + \mathbf{v}^p - [(\hat{\mathbf{R}}^r)^\top(\hat{\mathbf{p}}^f - \hat{\mathbf{p}}^r)] \end{bmatrix},\end{aligned} \quad (16)$$

where $\mathbf{y}^R, \mathbf{y}^p \in \mathbb{R}^3$. The linearization of $\mathbf{y}^R$ is obtained by

$$\begin{aligned}\mathbf{I}_3 + (\mathbf{y}^R)^\wedge &\approx \exp^{\mathbb{SO}(3)}((\mathbf{y}^R)^\wedge) \\ &= \exp^{\mathbb{SO}(3)}((\mathbf{v}^R)^\wedge)(\mathbf{R}^r)^\top\mathbf{R}^f(\hat{\mathbf{R}}^f)^\top\hat{\mathbf{R}}^r \\ &\approx \mathbf{I}_3 - ((\hat{\mathbf{R}}^r)^\top\boldsymbol{\xi}^{R^r})^\wedge \\ &\quad +((\hat{\mathbf{R}}^r)^\top\boldsymbol{\xi}^{R^f})^\wedge + (\mathbf{v}^R)^\wedge.\end{aligned} \quad (17)$$

Then, by omitting the second-order small quantities, we have

$$\mathbf{y}^R = -(\hat{\mathbf{R}}^r)^\top\boldsymbol{\xi}^{R^r} + (\hat{\mathbf{R}}^r)^\top\boldsymbol{\xi}^{R^f} + \mathbf{v}^R. \quad (18)$$

The linearization of $\mathbf{y}^p$ can be derived directly from point feature RI-EKF SLAM in [20][21]. For the innovation at time step $n+1$, we have

$$\mathbf{y}_{n+1} = \mathbf{H}_{n+1}\boldsymbol{\xi}_{n+1|n} + \mathbf{v}_{n+1}, \quad (19)$$

where

$$\begin{aligned}\mathbf{H}_{n+1} &= \begin{bmatrix} \mathbf{H}_{n+1}^{R,R^r} & \mathbf{H}_{n+1}^{R,R^f} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{H}_{n+1}^{p,p^r} & \mathbf{H}_{n+1}^{p,p^f} \end{bmatrix}, \\ &\mathbf{H}_{n+1}^{R,R^r} = \mathbf{H}_{n+1}^{p,p^r} = -(\mathbf{R}_{n+1|n}^r)^\top, \\ &\mathbf{H}_{n+1}^{R,R^f} = \mathbf{H}_{n+1}^{p,p^f} = (\mathbf{R}_{n+1|n}^r)^\top,\end{aligned} \quad (20)$$

and $\mathbf{v}_{n+1} \sim N(\mathbf{0}, \boldsymbol{\Omega}_{n+1})$ is the observation noise. Then the state is updated by

$$\mathbf{X}_{n+1|n+1} = \exp^{\mathcal{G}}(\boldsymbol{\xi}_{n+1|n+1}) \oplus \mathbf{X}_{n+1|n}, \quad (21)$$

where $\boldsymbol{\xi}_{n+1|n+1} = \mathbf{K}_{n+1}\mathbf{y}_{n+1}$ is the update state error vector, and

$$\mathbf{K}_{n+1} = \mathbf{P}_{n+1|n}\mathbf{H}_{n+1}^\top(\mathbf{H}_{n+1}\mathbf{P}_{n+1|n}\mathbf{H}_{n+1}^\top + \boldsymbol{\Omega}_{n+1})^{-1}.$$

Its covariance is updated as

$$\mathbf{P}_{n+1} = (\mathbf{I} - \mathbf{K}_{n+1}\mathbf{H}_{n+1})\mathbf{P}_{n+1|n}. \quad (22)$$

The whole process of RI-EKF SLAM with object features can be summarized in **Algorithm 1**.

Algorithm 1 RI-EKF for Object based SLAM

$\mathbf{F}_n$, $\mathbf{G}_n$, $\mathbf{H}_{n+1}$ are given in (14) and (20).
Input: $\mathbf{X}_{n|n}$, $\mathbf{P}_n$, $\mathbf{U}_n$, $\mathbf{Z}_{n+1}$
Output: $\mathbf{X}_{n+1|n+1}$, $\mathbf{P}_{n+1}$
Propagation:
$\mathbf{X}_{n+1|n} \leftarrow f(\mathbf{X}_{n|n}, \mathbf{U}_n, \mathbf{0})$
$\mathbf{P}_{n+1|n} \leftarrow \mathbf{F}_n\mathbf{P}_n\mathbf{F}_n^\top + \mathbf{G}_n\boldsymbol{\Sigma}_n\mathbf{G}_n^\top$
Update:
$\mathbf{K}_{n+1} \leftarrow \mathbf{P}_{n+1|n}\mathbf{H}_{n+1}^\top(\mathbf{H}_{n+1}\mathbf{P}_{n+1|n}\mathbf{H}_{n+1}^\top + \boldsymbol{\Omega}_{n+1})^{-1}$
$\mathbf{y}_{n+1} \leftarrow \mathbf{Z}_{n+1} \boxminus h_{n+1}(\mathbf{X}_{n+1|n}, \mathbf{0})$
$\mathbf{X}_{n+1|n+1} \leftarrow \exp^{\mathcal{G}}(\mathbf{K}_{n+1}\mathbf{y}_{n+1}) \oplus \mathbf{X}_{n+1|n}$
$\mathbf{P}_{n+1} \leftarrow (\mathbf{I} - \mathbf{K}_{n+1}\mathbf{H}_{n+1})\mathbf{P}_{n+1|n}$

Algorithm 2 New Feature Initialization

Input:
The state and its covariance before augmentation:
$\hat{\mathbf{X}} = \begin{bmatrix} \hat{\mathbf{R}}^r & \hat{\mathbf{R}}^f & \hat{\mathbf{p}}^r & \hat{\mathbf{p}}^f \end{bmatrix}$
$\mathbf{P} = \begin{bmatrix} \mathbf{P}^{R,R} & \mathbf{P}^{R,p} \\ \mathbf{P}^{p,R} & \mathbf{P}^{p,p} \end{bmatrix}$
The observation of new feature: $\mathbf{Z} = (\mathbf{R}^z, \mathbf{p}^z) \in \mathbb{SO}(3) \times \mathbb{R}^3$
The covariance of observation noise: $\boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{\Omega}^{R,R} & \boldsymbol{\Omega}^{R,p} \\ \boldsymbol{\Omega}^{p,R} & \boldsymbol{\Omega}^{p,p} \end{bmatrix}$

Output: The augmented state and its covariance:
$\hat{\mathbf{X}}_{\text{aug}} = \begin{bmatrix} \hat{\mathbf{R}}^r & \hat{\mathbf{R}}^f & \hat{\mathbf{R}}^r\mathbf{R}^z & \hat{\mathbf{p}}^r & \hat{\mathbf{p}}^f & \hat{\mathbf{p}}^r + \hat{\mathbf{R}}^r\mathbf{p}^z \end{bmatrix}$,

$$\mathbf{P}_{\text{aug}} = \begin{bmatrix} \mathbf{P}^{R,R} & \mathbf{P}^{R,R}\mathbf{M}_1^\top & \mathbf{P}^{R,p} & \mathbf{P}^{R,p}\mathbf{M}_2^\top \\ \mathbf{M}_1\mathbf{P}^{R,R} & \mathbf{P}_f^{R,R} & \mathbf{M}_1\mathbf{P}^{R,p} & \mathbf{P}_f^{R,p} \\ \mathbf{P}^{p,R} & \mathbf{P}^{p,R}\mathbf{M}_1^\top & \mathbf{P}^{p,p} & \mathbf{P}^{p,p}\mathbf{M}_2^\top \\ \mathbf{M}_2\mathbf{P}^{p,R} & (\mathbf{P}_f^{R,p})^\top & \mathbf{M}_2\mathbf{P}^{p,p} & \mathbf{P}_f^{p,p} \end{bmatrix}.$$

B. New Feature Initialization

Besides propagation and updating, another indispensable procedure is object feature initialization. For brevity, the mathematical derivation is ignored here (details are in the full version [32]). The whole process to augment the state is summarized in **Algorithm 2**, where

$$\begin{aligned}\mathbf{M}_1 &= [\mathbf{I}_3\ \mathbf{0}_{3,3K}], \\ \mathbf{M}_2 &= [\mathbf{I}_3\ \mathbf{0}_{3,3K}], \\ \mathbf{P}_f^{R,R} &= \mathbf{M}_1\mathbf{P}^{R,R}\mathbf{M}_1^\top + \hat{\mathbf{R}}^r\boldsymbol{\Omega}^{R,R}(\hat{\mathbf{R}}^r)^\top, \\ \mathbf{P}_f^{R,p} &= \mathbf{M}_1\mathbf{P}^{R,p}\mathbf{M}_2^\top + \hat{\mathbf{R}}^r\boldsymbol{\Omega}^{R,p}(\hat{\mathbf{R}}^r)^\top, \\ \mathbf{P}_f^{p,p} &= \mathbf{M}_2\mathbf{P}^{p,p}\mathbf{M}_2^\top + \hat{\mathbf{R}}^r\boldsymbol{\Omega}^{p,p}(\hat{\mathbf{R}}^r)^\top.\end{aligned} \quad (23)$$

IV. Observability Analysis

Based on the previous researches about inconsistency [15]-[21], the inconsistency of EKF-SLAM is mainly caused by the violation of the observability constraints. A consistent EKF-SLAM estimator should satisfy the following observability constraints: the unobservable subspace for the system model of the estimator is the same as that of the real system (the ideal case where the Jacobians are evaluated at the true state). In this section, we prove that our RI-EKF for object based SLAM can automatically maintain the observability constraints. On the contrary, standard EKF for object based SLAM (briefly introduced in Sec. IV-B) is unable to maintain the observability constraints. These explain the better performance of our algorithm in terms of consistency in the following experiments.

Definition 1: The unobservable subspace $\hat{\mathcal{N}}$ based on the state estimates is the null space of the corresponding observability matrix $\hat{\mathbf{O}}$, where

$$\hat{\mathbf{O}} = \begin{bmatrix} \hat{\mathbf{H}}_0 \\ \hat{\mathbf{H}}_1\hat{\mathbf{F}}_{0,0} \\ \vdots \\ \hat{\mathbf{H}}_{n+1}\hat{\mathbf{F}}_{n,0} \end{bmatrix}, \tag{24}$$

$\hat{\mathbf{H}}_i$ is the Jacobian matrix for the i-th step observation model evaluated at the state estimate $\hat{\mathbf{X}}_i$, and $\hat{\mathbf{F}}_{i,0} = \hat{\mathbf{F}}_i\hat{\mathbf{F}}_{i-1}\cdots\hat{\mathbf{F}}_0$, $\hat{\mathbf{F}}_j$ is the Jacobian matrix for the j-th step propagation model of the estimator evaluated at the state $\hat{\mathbf{X}}_j$, $j = 0, \cdots, i$. If the models are linearized at the ground truth, the unobservable subspace based on the true states is denoted by $\breve{\mathcal{N}}$, and the corresponding observability matrix is denoted by $\breve{\mathbf{O}}$.

A. Observability Analysis for RI-EKF

Theorem 1: For RI-EKF, the unobservable subspace $\hat{\mathcal{N}}^{RI}$ is the same as $\breve{\mathcal{N}}^{RI}$, where

$$\hat{\mathcal{N}}^{RI} = \breve{\mathcal{N}}^{RI} = \underset{col.}{\text{span}} \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix}, \tag{25}$$

and $\dim(\hat{\mathcal{N}}^{RI}) = \dim(\breve{\mathcal{N}}^{RI}) = 6$.

Proof 1: See Appendix in the full version of this paper [32].

Therefore, Theorem 1 shows that RI-EKF automatically maintains the correct unobservable subspace, which will significantly improve the consistency.

B. Standard EKF for Object based SLAM

The standard EKF (Std-EKF) for object based SLAM is $\mathbb{SO}(3)$-EKF, whose state space is isomorphic to $(\mathbb{SO}(3))^{K+1}\times(\mathbb{R}^3)^{K+1}$. Suppose there is only one feature in the state vector, an error state $\boldsymbol{\eta} \in (\mathfrak{so}(3))^{K+1}\times(\mathbb{R}^3)^{K+1}$ ($K = 1$ for simplicity) in standard EKF is obtained by

$$\begin{aligned} \boldsymbol{\eta} &= (\boldsymbol{\eta}^{R^r}, \boldsymbol{\eta}^{R^f}, \boldsymbol{\eta}^{p^r}, \boldsymbol{\eta}^{p^f}) \\ &= (\log^{\mathbb{SO}(3)}(\mathbf{R}^r(\hat{\mathbf{R}}^r)^\top), \log^{\mathbb{SO}(3)}(\mathbf{R}^f(\hat{\mathbf{R}}^f)^\top), \\ &\quad \mathbf{p}^r - \hat{\mathbf{p}}^r, \mathbf{p}^f - \hat{\mathbf{p}}^f), \end{aligned} \tag{26}$$

where $(\mathbf{R}^r, \mathbf{p}^r)$ and $(\hat{\mathbf{R}}^r, \hat{\mathbf{p}}^r)$ are the true and the estimated robot poses, $(\mathbf{R}^f, \mathbf{p}^f)$ and $(\hat{\mathbf{R}}^f, \hat{\mathbf{p}}^f)$ are the true and the estimated object features, respectively. Based on this linearization method, the Jacobians of considered system are

$$\begin{aligned} \mathbf{F}_n^{Std} &= \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -(\mathbf{R}^r_{n|n}\mathbf{p}^u)^\wedge & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix}, \\ \mathbf{G}_n^{Std} &= \begin{bmatrix} \mathbf{R}^r_{n|n} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{R}^r_{n|n} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}, \\ \mathbf{H}_{n+1}^{Std} &= \begin{bmatrix} \mathbf{H}_{n+1}^{R,R^r} & \mathbf{H}_{n+1}^{R,R^f} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{H}_{n+1}^{p,R^r} & \mathbf{0}_{3\times3} & \mathbf{H}_{n+1}^{p,p^r} & \mathbf{H}_{n+1}^{p,p^f} \end{bmatrix}, \end{aligned} \tag{27}$$

where

$$\begin{aligned} \mathbf{H}_{n+1}^{R,R^r} &= \mathbf{H}_{n+1}^{p,p^r} = -(\mathbf{R}^r_{n+1|n})^\top, \\ \mathbf{H}_{n+1}^{R,R^f} &= \mathbf{H}_{n+1}^{p,p^f} = (\mathbf{R}^r_{n+1|n})^\top, \\ \mathbf{H}_{n+1}^{p,R^r} &= (\mathbf{R}^r_{n+1|n})^\top(\mathbf{p}^f_{n+1|n} - \mathbf{p}^r_{n+1|n})^\wedge, \end{aligned}$$

$\mathbf{F}_n$ and $\mathbf{G}_n$ are the Jacobians of process model for the state error $\boldsymbol{\eta}_{n|n}$ and the odometry noise $\mathbf{w}_n$, respectively. And $\mathbf{H}_{n+1}^{Std}$ evaluated at $\mathbf{X}_{n+1|n}$ is the Jacobian of innovation $\mathbf{y}$ defined in (16).

C. Observability Analysis for Standard EKF

Theorem 2: For Std-EKF, the unobservable subspace $\hat{\mathcal{N}}^{Std}$, is a proper subspace of $\breve{\mathcal{N}}^{Std}$, where

$$\hat{\mathcal{N}}^{Std} = \underset{col.}{\text{span}}[\mathbf{0}_{3\times3}, \mathbf{0}_{3\times3}, \mathbf{I}_3, \mathbf{I}_3]^\top, \tag{28}$$

and

$$\breve{\mathcal{N}}^{Std} = \underset{col.}{\text{span}} \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 \\ \mathbf{I}_3 & (\mathbf{p}^r_0)^\wedge \\ \mathbf{I}_3 & (\mathbf{p}^f)^\wedge \end{bmatrix}, \tag{29}$$

$\mathbf{p}^r_0$ and $\mathbf{p}^f$ are respectively the true positions of robot starting point and object feature. And the dimension of $\hat{\mathcal{N}}^{Std}$ is 3, while the dimension of $\breve{\mathcal{N}}^{Std}$ is 6.

Proof 2: See Appendix in the full version of this paper [32].

According to Theorem 2, due to this improper linearization for object based SLAM, standard EKF does not maintain the correct observability constraints. Consequently, standard EKF mistakenly takes spurious information into estimation, leading to overconfident estimate (inconsistency) [19].

V. Simulations

In this section, we compare our proposed RI-EKF with standard EKF (Std-EKF) and Ideal-EKF (a variant of the Std-EKF where Jacobians are evaluated at the ground truth). It should be noted that Ideal-EKF is impossible to be applied in the real scenario, since the ground truth is not available. It is just used to explain the influence of observability constraints on inconsistency. We use Normalized Estimation Error Squared

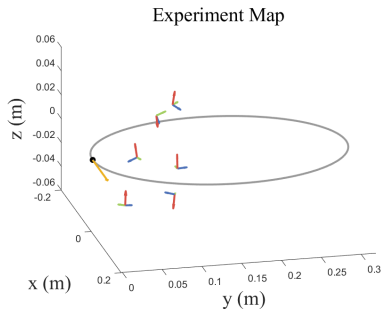

Fig. 1. Simulation environment: 6 object features (the poses are shown as red-green-blue arrows) in a 3D environment, robot moves on the circle (the yellow arrow shows the initial heading of the robot).

TABLE I
SIMULATION RESULTS OF RI-EKF, STD-EKF AND IDEAL-EKF

	Std-EKF	RI-EKF	Ideal-EKF
	RMSE		
Robot Rotation (rad)	.0239	.0230	.0199
Robot Position (m)	.0039	.0038	.0035
Feature Rotation (rad)	.0072	.0066	.0046
Feature Position (m)	.0007	.0007	.0006
	NEES		
Robot Rotation	1.215	0.973	0.903
Robot Position	1.155	1.090	0.959
Robot Pose	1.216	1.016	0.973
Feature Rotation	1.531	0.953	0.735
Feature Position	1.135	1.113	1.058
Feature Pose	1.306	1.007	0.909

(NEES) indicator to evaluate the consistency of an estimation method

$$\text{NEES} = \frac{1}{m \times d} \sum_{i=1}^{m} \mathbf{e}_i^\top \mathbf{P}_i^{-1} \mathbf{e}_i, \tag{30}$$

where m is the number of samples, and $\mathbf{e}_i$ is a d dimensional error sample vector, which is estimated to be a zero mean Gaussian with a $d \times d$ covariance matrix $\mathbf{P}_i$. NEES should approximately equal to 1 for large m, if the estimator is consistent. In addition, root mean squared error (RMSE) is used to evaluate the accuracy of each estimator.

To compute NEES, it is worth noting that for our proposed RI-EKF, the estimated covariance is corresponding to the nonlinear error defined in (7) instead of the standard error in the vector space. However, for a fair comparison, we still use the standard error to compute the RMSE of RI-EKF.

A. Settings

The simulation environment and the robot trajectory are shown in Fig. 1. There are 6 object features in the environment and their poses are represented by the red-green-blue arrows. The robot moves along a circle 2 times (trajectory length: 2m) with a constant linear velocity 5×10^{-4} m per frame and a constant angular velocity $\pi/1000$ rad per frame. The robot is able to measure the relative poses of all 6 object features in the environment. The covariance matrices of odometry noise and observation noise in (4) and (5) are set to be $\mathbf{\Sigma}_n = \text{diag}(0.01^2, 0.01^2, 0.01^2, 0.02^2, 0.02^2, 0.02^2)$ and $\mathbf{\Omega}_n = \text{diag}(0.04^2, 0.04^2, 0.04^2, 0.002^2 0.002^2 0.002^2)$. The settings in this simulation are similar to the real data experiments shown in the next section.

B. Results and Analysis

We conducted 50 Monte-Carlo simulations, i.e. $m = 50$. The NEES and RMSE results are shown in Fig. 2 and Table I. Fig. 2 shows the RMSE and NEES results for robot pose and feature pose every 50 steps. Table I lists the average RMSE and NEES for rotation and position error (in rad and m respectively) in the last time step. The results show that in this experiment, RI-EKF and Ideal-EKF perform better than Std-EKF in terms of both accuracy and consistency. Based on the comparison of Std-EKF and Ideal-EKF, we can see that the inconsistency of Std-EKF mainly comes from the inaccuracy of linearization points. And according to the analysis in Section IV, these linearization points in Std-EKF break the observability constraints, leading Std-EKF to obtain spurious information from unobservable subspace. As a result, its estimation will be more inaccurate and its estimated covariance is smaller than the actual uncertainty, and become more and more inconsistent over time. In contrast, RI-EKF remains consistent (NEES ≈ 1) in a longer duration, behaving like Ideal-EKF. The analysis for RI-EKF in Section IV indicates that RI-EKF naturally maintains the observability constraints, as in Ideal-EKF, while the Jacobians are evaluated at the latest estimate. These make the results of RI-EKF more reliable than those of Std-EKF.

VI. REAL DATA EXPERIMENTS

In this section, we test our algorithm on a real dataset YCB-Video [29] and compare it with Std-EKF, DVO [2], ORB-SLAM3 [3] and pose graph optimization (PGO) to show its effectiveness. All of these algorithms are fed by the RGB-D images from YCB-Video. Four sequences (0019, 0036, 0041, and 0049), which have relatively long trajectory, are selected to be used in the experiments (Fig. 3). Different from simulation, the data collected from real world may have many outliers. The matches of point clouds in Sequence 0019 are very accurate, but in other sequences, some object observations are very inaccurate, as shown in the last three images in the lower row of Fig. 3.

In order to make the algorithms robust, we need to detect and remove outliers. In addition, there is no information about odometry in these data sequences. To apply our algorithms, we make a simple constant velocity assumption for these data sequences which are obtained at the low camera motion speeds.

A. Dataset and Object Detection

YCB-Video dataset contains 21 objects with various textures from YCB objects. There are 92 RGB-D videos used for training and testing object detection, in which 80 videos are used for training and 2949 keyframes from the rest 12 videos are used for testing. Besides, 80000 synthetic images are released for training. There are many scenes of stacking objects with partial occlusion, as shown in Fig. 3.

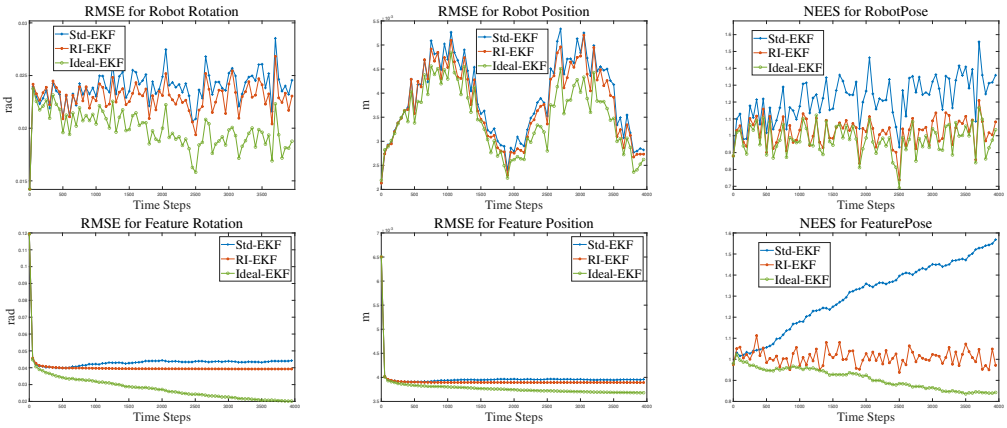

Fig. 2. Accuracy (RMSE) and consistency (NEES) of Std-EKF, RI-EKF, and Ideal-EKF in simulations.

B. Observation of Object Features

To get the observation of the object features, in the front-end, we utilize the algorithm called REDE from [30], which is an end-to-end object pose estimator using RGB-D data as inputs. In YCB dataset, the results of the pose estimator can realize 98.9% recall under the metric of average ADD [31]. The outputs of REDE are directly fed into (5) as observations.

C. Constant Velocity Assumption

Since the data are collected from a low speed camera, we assume that the camera is moving at constant velocity. Here we just take a very simple method to obtain the odometry. We assume the angular velocity is zero with noise. And the expected linear velocity is the average of the linear velocities calculated using the previous 6 estimations. The variances are computed based on these 6 step estimations.

D. Outlier Removal

Suppose there is an object feature observed in the $n+1$ step, and $\mathbf{Z} \in \mathbb{SO}(3) \times \mathbb{R}^3$ is its observation. Before updating the state vector, we will first compute $\mathbf{y} = \mathbf{Z} \boxminus h(\mathbf{X}_{n+1|n}, \mathbf{0})$ in (16). According to EKF framework,

$$\mathbf{y} \approx \mathbf{H}_{n+1} \boldsymbol{\xi}_{n+1|n} + \mathbf{v}_{n+1} \tag{31}$$

where $\mathbf{H}_{n+1}$ is the Jacobian of observation function evaluated at $\mathbf{X}_{n+1|n}$, and $\mathbf{v}_{n+1} \sim N(\mathbf{0}, \boldsymbol{\Omega}_{n+1})$ is the noise. If the estimation is accurate, then $\mathbf{y}$ should form a zero mean Gaussian distribution with covariance

$$\mathbf{P}_y = \mathbf{H}_{n+1} \mathbf{P}_{n+1|n} (\mathbf{H}_{n+1})^\top + \boldsymbol{\Omega}_{n+1}. \tag{32}$$

Therefore, if all the elements of $\mathbf{y}$, i.e. $\mathbf{y}(k),\ k = 1, \cdots, 6$, are in their 3σ bounds, i.e.

$$|\mathbf{y}(k)| < 3\sqrt{\mathbf{P}_y(k,k)},\ k = 1, \cdots 6,$$

then we will use $\mathbf{y}$ to update the state. Otherwise, the observation will be considered as an outlier. This outlier removal is similar to that using Mahalanobis distance. The EKF methods with this outlier removal are called Robust EKF methods in the following.

TABLE II
AVERAGE RMSE* RESULTS FROM FOUR RGB-D SEQUENCES OF YCB-VIDEO DATASET [29].

	0019	0036	0041	0049
DVO	.015/.029	.021/.055	.023/.038	.010/.046
ORB-SLAM3	.005/.025	.009/.051	.012/.026	.010/.044
Std-EKF	.007/.009	.012/.017	.008/.010	.201/.252
Robust Std-EKF	.007/.009	.015/.021	.008/.010	.011/.023
RI-EKF	.004/.006	.004/.007	.006/.007	.156/.205
Robust RI-EKF	.004/.006	.004/.006	.006/.007	.007/.022
PGO	.003/.004	.003/.005	.005/.006	.007/.022

*RMSE of Robot Position (m)/RMSE of Robot Rotation (rad)

E. Results and Analysis

The standard deviations of the measurement noises of sequence 0019, 0036, and 0041 are around 0.001 m/0.04 rad for position and rotation. The noise level of sequence 0049 is around 0.04 m/0.6 rad for position and rotation, which is much larger than the other three sequences. In the estimators, we tend to set the measurement uncertainties slightly lower than the actual noise levels, in order to remove the outliers. In these experiments, we set the standard deviations of the measurement noises in Std-EKF and RI-EKF to be around 0.001 m/0.03 rad for the four data sequences.

We compare our algorithm with DVO [2], ORB-SLAM3 [3] and Std-EKF using the four data sequences. In addition, PGO is a common back-end approach in the existing object SLAM systems [8][10]. We also test PGO using the same information as in Robust RI-EKF. The average RMSE for robot rotation and position are shown in Table II. In general, PGO performs the best on these four data sequences as listed in Table II as expected. The Robust RI-EKF performs the closest to this optimization based method in terms of accuracy for the tested sequences. However, assuming all the objects are observed at all the steps, the computational complexity of EKF methods is $\mathcal{O}(T \cdot N^3)$ for the whole trajectory, while the computational complexity of pose graph optimization (using Schur complement) is $\mathcal{O}(T^3 + T^2 \cdot N)$, where T is the number of time steps and N is the number of objects [33]. Therefore when $N << T$ (for examples, the presenting experiments), EKF methods are more efficient than this optimization method. Although both DVO and ORB-SLAM3 utilize all the features in the whole

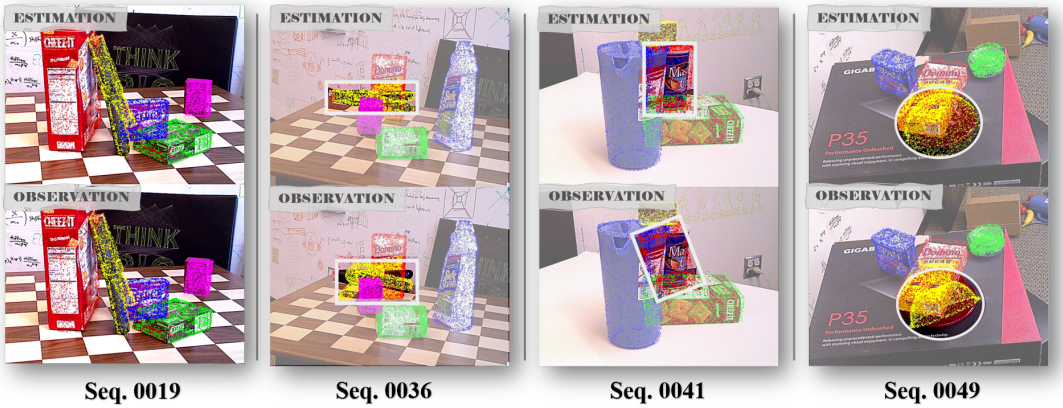

Fig. 3. Sample images from the four sequences in YCB-Video Dataset [29] used in the experiments. For each column, the lower row shows an image with feature observations while the upper row shows the final estimate from our proposed method.

trajectory, they only exploit low-level features (point features). In contrast, the object features have broader perspective loop closures, longer feature track, and more intrinsic constraints information between low-level features. Additionally, constant velocity model also provides extra information. Hence, DVO and ORB-SLAM3 are not fairly comparable. These could explain why the filter based Robust RI-EKF outperforms these two optimization based methods.

The comparison of Robust RI-EKF, Robust Std-EKF, RI-EKF, and Std-EKF on Sequence 0049 shows that robust methods are far better on such inaccurate data sequences. We also note that some RMSE of Robust Std-EKF are larger than that of Std-EKF, especially on Sequence 0036. This is mainly caused by the inconsistency of Std-EKF which results in mistaken deletion for correct data in Robust Std-EKF.

Fig. 4 shows the errors of robot pose estimates and the 3σ bounds of each component for Robust RI-EKF and Robust Std-EKF on Sequence 0036, respectively. Robust RI-EKF perfoms well in terms of consistency, while Robust Std-EKF underestimates the uncertainty of the state in the latter half of the sequence. This inconsistency of robust Std-EKF can finally result in larger errors. In contrast, the estimates by Robust RI-EKF are more reliable, making our outlier removal more effective.

Fig. 5 shows the RMSE of an object (Pitcher Base in Sequence 0041) in every 100 steps. The RMSE of each object in Sequence 0041 can be found in the full version of this paper [32]. They illustrate that Robust RI-EKF also generates more accurate estimates on the object poses than Robust Std-EKF.

In general, from the real data experiments, we can see Robust RI-EKF can generate good estimation results. In Fig. 3, the upper row images show the estimated objects from Robust RI-EKF, which significantly improve the corresponding observations shown in the lower row images for Sequences 0036, 0041 and 0049.

VII. Conclusion

In this work, we propose a right invariant EKF (RI-EKF) algorithm for object based SLAM, where object features are represented by 3D poses and are estimated together with the latest robot pose. From theoretical analysis, we prove that our RI-EKF generated by the proposed Lie group automatically maintains the correct observability properties. This is different from standard EKF with object features that does not have the correct observability property. Results from simulations and real data experiments confirm the good performance of the proposed RI-EKF algorithm.

As general EKF, RI-EKF framework also assumes the models are under Gaussian white noises and only considers the first order errors. Hence the proposed method could have some limitations in non-Gaussian noise or large noise level problems.

In this paper, we focus on the SLAM back-end and assume the objects observed are within a given database such that they can be detected and matched relatively easily from the SLAM front-end. In the future, we will investigate the more challenging object based SLAM problem where the objects in the environment are more general and may not belong to a known database.

References

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

[2] C. Kerl, J. Sturm and D. Cremers, "Dense visual SLAM for RGB-D cameras," 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 2100-2106.

[3] C. Campos, R. Elvira, J. J. G. Rodríguez, et al., "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM," IEEE Transactions on Robotics, 2021. doi: 10.1109/TRO.2021.3075644.

[4] R. Gomez-Ojeda, J. Briales and J. Gonzalez-Jimenez, "PL-SVO: Semi-direct Monocular Visual Odometry by combining points and line segments," 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016, pp. 4211-4216.

[5] A. J. B. Trevor, J. G. Rogers and H. I. Christensen, "Planar surface SLAM with 3D and 2D sensors," 2012 IEEE International Conference on Robotics and Automation, 2012, pp. 3041-3048.

[6] J. Civera, D. Galvez-Lopez, L. Riazuelo, et al., "Towards semantic SLAM using a monocular camera," IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2011.

[7] M. Sualeh, G. W. Kim, "Simultaneous localization and mapping in the epoch of semantics: a survey," in International Journal of Control, Automation and Systems, vol. 17, no. 3, pp. 729-742, 2019.

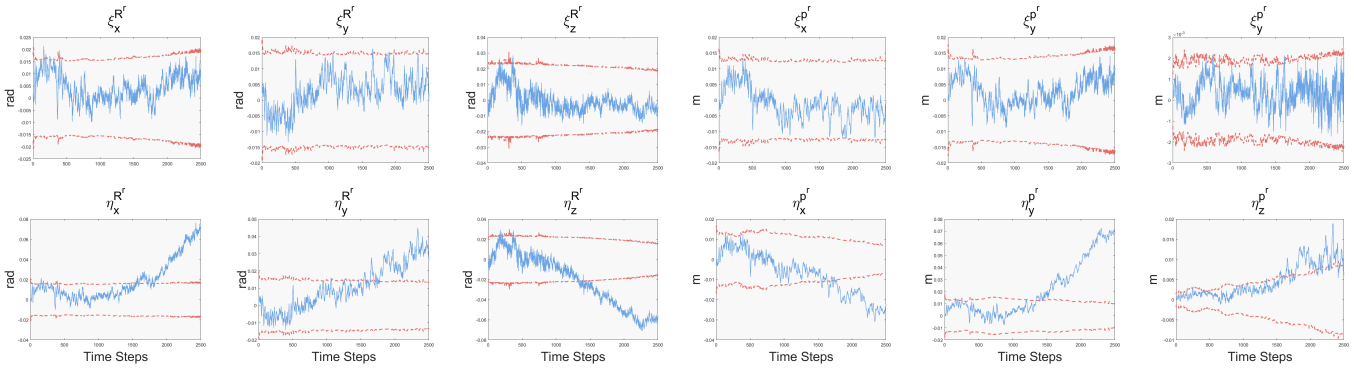

Fig. 4. Robot pose estimate errors and the corresponding 3σ bounds for Sequence 0036: the upper six figures are for Robust RI-EKF, the lower six figures are for Robust Std-EKF.

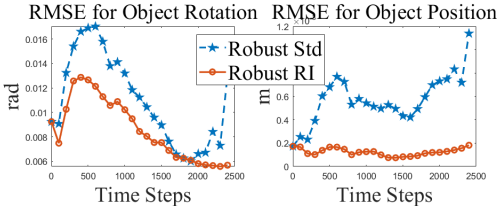

Fig. 5. RMSE of an object (Pitcher Base) in Sequence 0041 for Robust RI-EKF and Robust Std-EKF.

[8] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, et al., "SLAM++: Simultaneous localisation and mapping at the level of objects," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2013, pp. 1352–1359

[9] D. Gálvez-López, M. Salas, J. D. Tardós, and J. M. M. Montiel, "Real-time monocular object SLAM," Robotics and Autonomous Systems, 2016, 75: 435-449.

[10] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D Object SLAM," in IEEE Transactions on Robotics, vol. 35, no. 4, pp. 925-938, Aug. 2019.

[11] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige and W. Burgard, "G2o: A general framework for graph optimization," 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 3607-3613, doi: 10.1109/ICRA.2011.5979949.

[12] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[13] S. J. Julier, J. K. Uhlmann, "A counter example to the theory of simultaneous localization and map building," in Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on, vol. 4, 2001, pp. 4238–4243.

[14] J. A. Castellanos, J. Neira, and J. D. Tard´os, "Limits to the consistency of ekf-based slam," in 5th IFAC Symp, Intell. Autonom. Veh. IAV'04, 2004.

[15] T. Bailey, J. Nieto, J. Guivant, et al., "Consistency of the EKF-SLAM Algorithm," 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006, pp. 3562-3568.

[16] S. Huang and G. Dissanayake, "Convergence and consistency analysis for extended Kalman filter based SLAM," IEEE Transactions on Robotics, 23(5), 1036-1049. 2007.

[17] G. P. Huang, A. I. Mourikis and S. I. Roumeliotis, "Analysis and improvement of the consistency of extended Kalman filter based SLAM," 2008 IEEE International Conference on Robotics and Automation, 2008, pp. 473-479.

[18] G. P. Huang, A.I. Mourikis, and S.I. Roumeliotis. "A first-estimates Jacobian EKF for improving SLAM consistency," In 11th International Symposium on Experimental Robotics (ISER'08), Athens, Greece, July 2008.

[19] G. P. Huang, A. I. Mourikis, S. I. Roumeliotis, "Observability-based rules for designing consistent EKF SLAM estimators," The International Journal of Robotics Research, 2010, 29(5): 502-528.

[20] T. Zhang, K. Wu, J. Song, et al., "Convergence and consistency analysis for a 3-D invariant-EKF SLAM," IEEE Robot. Autom. Lett., vol. 2, no. 2, pp. 733–740, Apr. 2017.

[21] A. Barrau, S. Bonnabel, "An EKF-SLAM algorithm with consistency properties," Technical report, 2016. URL: https://arxiv.org/abs/1510.06263v3.

[22] A. Barrau, S. Bonnabel, "The Invariant Extended Kalman Filter as a Stable Observer," In: IEEE Transactions on Automatic Control 62.4 (2017), pp. 1797–1812.

[23] N. Aghannan, P. Rouchon, "On invariant asymptotic observers," Decision and Control, 2002, Proceedings of the 41st IEEE Conference on. IEEE, 2003.

[24] C. Forster, L. Carlone, F. Dellaert, et al., "On-Manifold Preintegration for Real-Time Visual-Inertial Odometry," IEEE Transactions on Robotics, 2015, 33(1):1-21.

[25] S. Bonnabel, "Symmetries in observer design: Review of some recent results and applications to ekf-based slam," In Robot Motion and Control, 2011:3-15.

[26] R. Mahony and T. Hamel, "A geometric nonlinear observer for simultaneous localisation and mapping," 2017 IEEE 56th Annual Conference on Decision and Control (CDC), 2017, pp. 2408-2415.

[27] M. Brossard, A. Barrau and S. Bonnabel, "Exploiting Symmetries to Design EKFs With Consistency Properties for Navigation and SLAM," in IEEE Sensors Journal, 2019, vol. 19, no. 4, pp. 1572-1579.

[28] A. Barrau and S. Bonnabel, "Stochastic observers on Lie groups: a tutorial," 2018 IEEE Conference on Decision and Control (CDC), 2018, pp. 1264-1269.

[29] Y. Xiang, T. Schmidt, V. Narayanan, et al., "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," in Robotics: Science and Systems, 2018.

[30] W. Hua, Z. Zhou, J. Wu, et al., "REDE: End-to-End Object 6D Pose Robust Estimation Using Differentiable Outliers Elimination," in IEEE Robotics and Automation Letters.

[31] S. Hinterstoisser, V. Lepetit, S. Ilic, et al., "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in Asian conference on computer vision. Springer, 2012, pp. 548–562.

[32] Y. Song, Z. Zhang, J. Wu, et al., "A Right Invariant Extended Kalman Filter for Object based SLAM," (Full version with Appendix,) 2021. URL: https://arxiv.org/abs/2109.05297.

[33] H. Strasdat, J.M.M. Montiel, A. J. Davison, "Visual SLAM: Why filter?," Image and Vision Computing, 2012, 30(2):65-77.