# Consistency Analysis and Improvement for Single-camera Localization

Joel A. Hesch and Stergios I. Roumeliotis*
Dept. of Computer Science and Engineering
University of Minnesota, Minneapolis, MN 55455
{joel|stergios}@cs.umn.edu

## Abstract

*In this paper, we study the problem of estimator inconsistency in single-camera simultaneous localization and mapping (MonoSLAM) from a standpoint of system observability. Specifically, we postulate that a leading cause of inconsistency is the gain of spurious information along unobservable directions, resulting in smaller uncertainties, larger estimation errors, and divergence. Moreover, we introduce an Observability-Constrained MonoSLAM (OC-MonoSLAM) approach, which explicitly enforces the unobservable directions of the system, hence preventing spurious information gain and reducing inconsistency. Our analysis, along with the proposed method for reducing inconsistency, are validated with simulation trials and real-world experimentation.*

## 1. Introduction

In egocentric vision tasks, it is often necessary to maintain an estimate of the camera's position and orientation (pose) over time as the person moves around. For example, a navigation aid for the visually impaired (e.g., [13]) must estimate its pose as the person walks, in order to provide them with turn-by-turn directions from point A to B. In a human-worn augmented reality system (e.g., [3]), maintaining the camera pose along with the environment structure, is necessary to annotate the scene with information.

Numerous vision-based localization approaches have been presented in the literature, including methods based on the Extended Kalman Filter (EKF) [5], the Unscented Kalman Filter (UKF) [15], and Batch-least Squares (BLS) [10, 14]. Non-parametric estimators, such as the Particle Filter (PF), have also been applied to visual odometry (e.g., [16]). While most existing works focus on vision navigation systems working in real-time [5] or providing dense-realistic maps [14], a key issue that has not yet been addressed in the literature is how estimator inconsistency impacts monocular localization. As defined in [1], a state estimator is consistent if the estimation errors are zero-mean and have covariance smaller than or equal to the one calculated by the filter. As we will demonstrate, a leading cause of inconsistency in monocular localization is due to spurious information gained about the scale of the scene, which is unobservable (i.e., scale cannot be determined using a monocular camera alone).

Until recently, little attention was paid to the effects that observability properties can have on nonlinear estimator consistency. The work by Huang et al. [7, 8, 9] was the first to identify this connection for several 2D localization problems (i.e., simultaneous localization and mapping, cooperative localization). The authors showed that, for these problems, a mismatch exists between the number of unobservable directions of the true nonlinear system and the linearized system used for estimation purposes. In particular, the estimated (linearized) system has one-fewer unobservable direction than the true system, allowing the estimator to surreptitiously gain spurious information along the direction corresponding to global orientation (yaw). This increases the estimation errors while reducing the estimator uncertainty, and leads to inconsistency.

In this paper, we analyze and improve consistency for monocular Simultaneous Localization and Mapping (MonoSLAM). The main contributions of this work are:

- We provide an overview of the MonoSLAM observability analysis using the system observability matrix, and show that seven d.o.f. are unobservable. These correspond to three-d.o.f. global translation, three-d.o.f. global rotation, and global scale.
- We report on MonoSLAM inconsistency, and demonstrate that a standard EKF-based MonoSLAM approach can gain spurious information about the scale of the system, leading to estimator inconsistency.
- We introduce an Observability-Constrained MonoSLAM (OC-MonoSLAM) algorithm which

*This work was supported by the University of Minnesota (DTC), the National Science Foundation (IIS-0643680, IIS-0811946, IIS-0835637), and AFOSR (FA9550-10-1-0567). J. A. Hesch was supported by the UMN Doctoral Dissertation Fellowship.

explicitly adheres to the system observability properties, and hence mitigates inconsistency. We validate our method with Monte-Carlo simulations and experimental results to show that it has increased consistency and lower errors compared to standard MonoSLAM.

The rest of this paper is organized as follows: In Sect. 2, we describe the system and measurement models, followed by our analysis of MonoSLAM inconsistency in Sect. 3. The proposed estimator modification is presented in Sect. 3.1, and subsequently validated both in simulations and experimentally (Sects. 4 and 5). Finally, we provide our concluding remarks and outline our future research directions in Sect. 6.

## 2. Estimator Description

We begin with an overview of the propagation and measurement models which govern the MonoSLAM system. We adopt the EKF as our framework for fusing the camera measurements across time, and we employ a tracking model to predict the camera's motion between images.[1] The sensing platform moves in a previously unknown environment, and localizes solely using Persistent Features (PFs) (e.g., SIFT keys [11]), which can be reliably tracked across images, and redetected when revisiting an area.

### 2.1. System State and Propagation Model

The EKF estimates the camera pose, as well as its linear and rotational velocities, and a map corresponding to the 3D coordinates of features in the environment. The filter state is the $(13+3N)\times 1$ vector:

$$\begin{aligned}\mathbf{x} &= \begin{bmatrix} {}^G\mathbf{p}_S^T & {}^S\bar{q}_G^T & {}^S\mathbf{v}^T & {}^S\boldsymbol{\omega}^T & | & {}^G\mathbf{f}_1^T \cdots {}^G\mathbf{f}_N^T \end{bmatrix}^T \\ &= \begin{bmatrix} \mathbf{x}_s^T & | & \mathbf{x}_m^T \end{bmatrix}^T, \end{aligned} \quad (1)$$

where $\mathbf{x}_s(t)$ is the $13\times 1$ sensor platform state, and $\mathbf{x}_m(t)$ is the $3N\times 1$ state of the map. The sensor platform state comprises ${}^S\bar{q}_G(t)$ which is the unit quaternion representing the orientation of the *global frame* $\{G\}$ in the sensor frame, $\{S\}$, at time $t$. The frame $\{S\}$ is attached to the camera, while $\{G\}$ is a reference frame whose origin coincides with the initial camera position. The linear and rotational velocities of the camera, ${}^S\mathbf{v}(t)$ and ${}^S\boldsymbol{\omega}(t)$, are expressed with respect to $\{S\}$, while the camera's position, ${}^G\mathbf{p}_S(t)$, is expressed in $\{G\}$.

The map, $\mathbf{x}_m$, comprises $N$ PFs, ${}^G\mathbf{f}_i$, $i=1,\ldots,N$, and grows as new PFs are observed. With the state of the system now defined, we turn our attention to the continuous-time model we utilize to track the system state.

[1] While we focus on the case of the EKF, our observability analysis and proposed algorithm for improving consistency are extensible to any linearized estimation architecture (e.g., UKF and sliding window filter).

#### 2.1.1 Continuous-time model

We employ a constant-velocity tracking model, in which both linear and rotational velocities are expressed in the *sensor* frame. This has the advantage of being more flexible than the "constant-global-velocity" model originally proposed for MonoSLAM [5], while at the same time enabling a simpler estimator framework than the Interacting Multiple Model (IMM) approach of Civera et al. [4].

$$ {}^S\dot{\bar{q}}_G(t) = \frac{1}{2}\boldsymbol{\Omega}({}^S\boldsymbol{\omega}(t)) {}^S\bar{q}_G(t), \quad {}^S\dot{\boldsymbol{\omega}}(t) = \boldsymbol{\eta}_\omega \quad (2)$$

$$ {}^G\dot{\mathbf{p}}_S(t) = {}^G\mathbf{v}_S(t) = {}^S_G\mathbf{C}^T {}^S\mathbf{v}(t), \quad {}^S\dot{\mathbf{v}}(t) = \boldsymbol{\eta}_v \quad (3)$$

$$ {}^G\dot{\mathbf{f}}_i(t) = \mathbf{0}_{3\times 1}\ , \ i=1,\ldots,N. \quad (4)$$

where ${}^S_G\mathbf{C}$ is the rotational matrix corresponding to ${}^S\bar{q}_G(t)$, and $\boldsymbol{\Omega}(\boldsymbol{\omega})$ is the matrix governing the quaternion time derivative, i.e.,

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor \boldsymbol{\omega}\times \rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}, \quad \lfloor \boldsymbol{\omega}\times \rfloor \triangleq \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}.$$

The time derivatives of the rotational and linear velocities, ${}^S\boldsymbol{\omega}$ and ${}^S\mathbf{v}$, are modeled as zero-mean white Gaussian processes, $\boldsymbol{\eta}_\omega$ and $\boldsymbol{\eta}_v$, respectively. The PFs belong to the static scene, thus, their time derivatives are zero [see (4)].

Linearizing at the current estimates and applying the expectation operator on both sides of (2)-(4), we obtain the state estimate propagation model

$$ {}^S\dot{\hat{\bar{q}}}_G(t) = \frac{1}{2}\boldsymbol{\Omega}(\hat{\boldsymbol{\omega}}(t)) {}^S\hat{\bar{q}}_G(t), \quad {}^S\dot{\hat{\boldsymbol{\omega}}}(t) = \mathbf{0}_{3\times 1} \quad (5)$$

$$ {}^G\dot{\hat{\mathbf{p}}}_S(t) = {}^S_G\hat{\mathbf{C}}^T {}^S\hat{\mathbf{v}}_S(t), \quad {}^S\dot{\hat{\mathbf{v}}}(t) = \mathbf{0}_{3\times 1} \quad (6)$$

$$ {}^G\dot{\hat{\mathbf{f}}}_i(t) = \mathbf{0}_{3\times 1}\ , \ i=1,\ldots,N. \quad (7)$$

The $(12+3N)\times 1$ error-state vector is defined as

$$\begin{aligned}\tilde{\mathbf{x}} &= \begin{bmatrix} {}^G\tilde{\mathbf{p}}_S^T & {}^S\boldsymbol{\delta\theta}_G^T & {}^S\tilde{\mathbf{v}}^T & {}^S\tilde{\boldsymbol{\omega}}^T & | & {}^G\tilde{\mathbf{f}}_1^T \cdots {}^G\tilde{\mathbf{f}}_N^T \end{bmatrix}^T \\ &= \begin{bmatrix} \tilde{\mathbf{x}}_s^T & | & \tilde{\mathbf{x}}_m^T \end{bmatrix}^T, \end{aligned} \quad (8)$$

where $\tilde{\mathbf{x}}_s(t)$ is the $12\times 1$ error state corresponding to the sensing platform, and $\tilde{\mathbf{x}}_m(t)$ is the $3N\times 1$ error state of the map. For the position, linear and rotational velocities, and the map, an additive error model is utilized (i.e., $\tilde{x}=x-\hat{x}$ is the error in the estimate $\hat{x}$ of a quantity $x$). However, for the quaternion we employ a multiplicative error model [17]. Specifically, the error between the quaternion $\bar{q}$ and its estimate $\hat{\bar{q}}$ is the $3\times 1$ angle-error vector, $\boldsymbol{\delta\theta}$, implicitly defined by the error quaternion

$$\delta\bar{q} = \bar{q}\otimes\hat{\bar{q}}^{-1} \simeq \begin{bmatrix} \frac{1}{2}\boldsymbol{\delta\theta}^T & 1 \end{bmatrix}^T, \quad (9)$$

where $\delta\bar{q}$ describes the small rotation that causes the true and estimated attitude to coincide. This allows us to represent the attitude uncertainty by the $3\times 3$ covariance matrix $\mathbb{E}[\boldsymbol{\delta\theta}\boldsymbol{\delta\theta}^T]$, which is a minimal representation.

The linearized continuous-time error-state equation is

$$\dot{\tilde{\mathbf{x}}} = \begin{bmatrix} \mathbf{F}_s & \mathbf{0}_{12\times 3N} \\ \mathbf{0}_{3N\times 12} & \mathbf{0}_{3N} \end{bmatrix} \tilde{\mathbf{x}} + \begin{bmatrix} \mathbf{G}_s \\ \mathbf{0}_{3N\times 6} \end{bmatrix} \mathbf{n} = \mathbf{F}_c\,\tilde{\mathbf{x}} + \mathbf{G}_c\,\mathbf{n}\,, \tag{10}$$

where $\mathbf{0}_{3N}$ denotes the $3N \times 3N$ matrix of zeros, $\mathbf{n} = \begin{bmatrix} \boldsymbol{\eta}_\omega^T & \boldsymbol{\eta}_v^T \end{bmatrix}^T$ is the system noise, which is modeled as a zero-mean white Gaussian process with autocorrelation $\mathbb{E}[\mathbf{n}(t)\mathbf{n}^T(\tau)] = \mathbf{Q}_c\delta(t-\tau)$. The matrix $\mathbf{F}_s$ is the continuous-time error-state transition matrix corresponding to the camera state, and $\mathbf{G}_s$ is the continuous-time input noise matrix, i.e.,

$$\mathbf{F}_s = \begin{bmatrix} \mathbf{0}_3 & -{}^S_G\mathbf{C}^T\lfloor {}^S\mathbf{v}\times\rfloor & {}^S_G\mathbf{C}^T & \mathbf{0}_3 \\ \mathbf{0}_3 & -\lfloor {}^S\boldsymbol{\omega}\times\rfloor & \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix},\ \mathbf{G}_s = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}.$$

#### 2.1.2 Discrete-time implementation

In order to propagate the state forward in time, we employ Euler integration of (5)–(7), with a specified step size, $\delta t$, selected to be significantly smaller than the camera frame-rate. Moreover, to derive the covariance propagation equation, we evaluate the discrete-time state transition matrix, $\boldsymbol{\Phi}_k$, and the discrete-time system noise covariance matrix, $\mathbf{Q}_{d,k}$, as

$$\boldsymbol{\Phi}_k = \boldsymbol{\Phi}(t_{k+1}, t_k) = \exp\left(\int_{t_k}^{t_{k+1}} \mathbf{F}_c(\tau)\mathrm{d}\tau\right) \tag{11}$$

$$\mathbf{Q}_{d,k} = \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1},\tau)\mathbf{G}_c\mathbf{Q}_c\mathbf{G}_c^T\boldsymbol{\Phi}^T(t_{k+1},\tau)\mathrm{d}\tau.$$

The propagated covariance is then computed as

$$\mathbf{P}_{k+1|k} = \boldsymbol{\Phi}_k\mathbf{P}_{k|k}\boldsymbol{\Phi}_k^T + \mathbf{Q}_{d,k}. \tag{12}$$

### 2.2. Measurement Update Model

As the camera moves it observes visual features. These measurements are exploited to concurrently estimate the motion of the sensing platform and the map of PFs. To simplify the discussion, we consider the observation of a single PF $\mathbf{f}_i$. The camera measures $\mathbf{z}_i$, which is the perspective projection of the 3D point, ${}^S\mathbf{f}_i = \begin{bmatrix} x & y & z \end{bmatrix}^T$, expressed in the current camera frame $\{S\}$, onto the image plane[2], i.e.,

$$\mathbf{z}_i = \frac{1}{z}\begin{bmatrix} x \\ y \end{bmatrix} + \boldsymbol{\eta}_i,\quad {}^S\mathbf{f}_i = {}^S_G\mathbf{C}\left({}^G\mathbf{f}_i - {}^G\mathbf{p}_S\right). \tag{13}$$

The measurement noise, $\boldsymbol{\eta}_i$, is modeled as zero mean white Gaussian with covariance $\mathbf{R}_i$. The linearized error model is $\tilde{\mathbf{z}}_i = \mathbf{z}_i - \hat{\mathbf{z}}_i \simeq \mathbf{H}_i\tilde{\mathbf{x}} + \eta_i$, where $\hat{\mathbf{z}}$ is the expected measurement computed by evaluating (13) at the current state estimate, and the measurement Jacobian, $\mathbf{H}_i$, is

$$\mathbf{H}_i = \mathbf{H}_{cam}\begin{bmatrix} \mathbf{H}_\mathbf{p} & \mathbf{0}_3 & \mathbf{H}_{\bar{q}} & \mathbf{0}_3 \,|\, \mathbf{0}_3 & \cdots & \mathbf{H}_{\mathbf{f}_i} & \cdots & \mathbf{0}_3 \end{bmatrix} \tag{14}$$

$$\mathbf{H}_{cam} = \frac{1}{z^2}\begin{bmatrix} z & 0 & -x \\ 0 & z & -y \end{bmatrix},\quad \mathbf{H}_\mathbf{p} = -{}^S_G\mathbf{C}$$

$$\mathbf{H}_{\bar{q}} = \lfloor {}^S_G\mathbf{C}\left({}^G\mathbf{f}_i - {}^G\mathbf{p}_S\right)\times\rfloor,\quad \mathbf{H}_{\mathbf{f}_i} = {}^S_G\mathbf{C}.$$

Here, $\mathbf{H}_{cam}$, is the Jacobian of the perspective projection with respect to ${}^S\mathbf{f}_i$, while $\mathbf{H}_{\bar{q}}$, $\mathbf{H}_\mathbf{p}$, and $\mathbf{H}_{\mathbf{f}_i}$, are the Jacobians of ${}^S\mathbf{f}_i$ with respect to ${}^S\bar{q}_G$, ${}^G\mathbf{p}_S$, and ${}^G\mathbf{f}_i$, respectively.

For PFs that are already in the map, we directly apply the measurement model (13)-(14) to update the filter. We compute the measurement residual, the covariance of the residual, and the Kalman gain

$$\mathbf{r}_i = \mathbf{z}_i - \hat{\mathbf{z}}_i,\quad \mathbf{S}_i = \mathbf{H}_i\mathbf{P}_{k+1|k}{\mathbf{H}_i}^T + \mathbf{R}_i \tag{15}$$

$$\mathbf{K} = \mathbf{P}_{k+1|k}{\mathbf{H}_i}^T\mathbf{S}_i^{-1}. \tag{16}$$

Employing these quantities, we compute the EKF state and covariance update as

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}\mathbf{r}_i \tag{17}$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{P}_{k+1|k}{\mathbf{H}_i}^T\mathbf{S}_i^{-1}\mathbf{H}_i\mathbf{P}_{k+1|k}. \tag{18}$$

For previously unseen PFs, we compute an initial estimate, along with covariance and cross-correlations by solving a bundle-adjustment over a short time window [18].

## 3. Observability-constrained MonoSLAM

Using the system and measurement models presented above, we hereafter describe how the system observability properties influence estimator consistency. In particular, we show that MonoSLAM has seven unobservable directions, corresponding to global translation, global rotation, and global scale. However, when using a linearized estimator, such as the EKF, errors in linearization while evaluating the system and measurement Jacobians change the directions in which information is acquired by the estimator. Over time, these directions can span the whole state space, including directions which should be unobservable. In particular, for MonoSLAM we observe that the estimator gains *scale* information, which can lead to scale drift over time. When spurious information is gained along unobservable directions, it leads to larger errors, smaller uncertainties, and inconsistency. In what follows, we first analyze the system observability properties and show why the standard MonoSLAM violates them. Subsequently, we present an Observability-Constrained MonoSLAM (OC-MonoSLAM) estimation algorithm that explicitly adheres to the observability properties of the system.

[2] Without loss of generality, we express the image measurement in normalized pixel coordinates [2].

The observability matrix [12] is defined as a function of the linearized measurement model, $\mathbf{H}$, and the discrete-time state transition matrix, $\mathbf{\Phi}$, which are in turn functions of the linearization point, $\mathbf{x}$, i.e.,

$$\mathbf{M}\left(\mathbf{x}\right) = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2\mathbf{\Phi}_{2,1} \\ \vdots \\ \mathbf{H}_k\mathbf{\Phi}_{k,1} \end{bmatrix} \tag{19}$$

where $\mathbf{\Phi}_{k,1} = \mathbf{\Phi}_{k-1}\cdots\mathbf{\Phi}_1$ is the state transition matrix from time step $t_1$ to $t_k$. We compute the discrete-time state transition matrix, $\mathbf{\Phi}_{k,1}$ as the solution to the following matrix differential equation,

$$\dot{\mathbf{\Phi}}_{t,t_1} = \mathbf{F}_c(t)\mathbf{\Phi}_{t,t_1} \quad i.c.\ \mathbf{\Phi}_{t_1,t_1} = \mathbf{I}. \tag{20}$$

To simplify the discussion, we consider only a single landmark in the state vector. Using the initial condition and the structure of $\mathbf{F}_c$ [see (10)], we obtain $\mathbf{\Phi}_{t,t_1}$ as

$$\mathbf{\Phi}_{t,t_1} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{\Phi}_{[1,2]} & \mathbf{\Phi}_{[1,3]} & \mathbf{\Phi}_{[1,4]} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{\Phi}_{[2,2]} & \mathbf{0} & \mathbf{\Phi}_{[2,4]} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \tag{21}$$

where

$$\mathbf{\Phi}_{[1,2]} = -\lfloor {}^G\mathbf{p}_{S(t)} - {}^G\mathbf{p}_{S(t1)} \times \rfloor \, {}^G_{S(t1)}\mathbf{C} \tag{22}$$

$$\mathbf{\Phi}_{[1,3]} = \int_{t1}^{t} {}^G_{S(\tau)}\mathbf{C}\, d\tau \tag{23}$$

$$\mathbf{\Phi}_{[1,4]} = -\int_{t1}^{t} \lfloor {}^G\mathbf{v}_{S(r)} \times \rfloor \int_{t1}^{r} {}^G_{S(\tau)}\mathbf{C}\, d\tau\, d\kappa \tag{24}$$

$$\mathbf{\Phi}_{[2,2]} = {}^{S(t)}_{S(t1)}\mathbf{C} \tag{25}$$

$$\mathbf{\Phi}_{[2,4]} = \int_{t1}^{t} {}^{S(t)}_{S(\tau)}\mathbf{C}\, d\tau \tag{26}$$

where $S(t)$ denotes the frame $\{S\}$ at time $t$. Employing (14) and (21), the $k-th$ block row of the observability matrix [see (19)] is

$$\mathbf{H}_k\mathbf{\Phi}_{k,1} = \mathbf{A}_1 \begin{bmatrix} -\mathbf{I}_3 & \mathbf{A}_2 & \mathbf{A}_3 & \mathbf{A}_4 & \mathbf{I}_3 \end{bmatrix} \tag{27}$$

where

$$\mathbf{A}_1 = \mathbf{H}_{cam,k} \cdot {}^{S(k)}_{G}\mathbf{C} \tag{28}$$

$$\mathbf{A}_2 = \lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{S(k)} \times \rfloor \, {}^G_{S(1)}\mathbf{C} \tag{29}$$

$$\mathbf{A}_3 = -\int_{t1}^{tk} {}^G_{S(\tau)}\mathbf{C}\, d\tau \tag{30}$$

$$\mathbf{A}_4 = \lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{S(k)} \times \rfloor \int_{t1}^{tk} {}^G_{S(\tau)}\mathbf{C}\, d\tau - \mathbf{\Phi}_{[1,4]}. \tag{31}$$

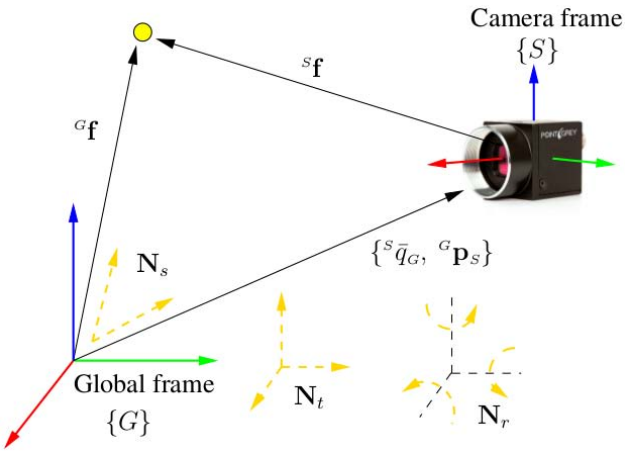


Figure 1: The unobservable directions are depicted in gold. $\mathbf{N}_s$ corresponds to global scale (i.e., translating the whole scene and the camera towards or away from the origin). $\mathbf{N}_t$ corresponds to global translations of the scene and camera along any of the cardinal axes. $\mathbf{N}_r$ corresponds to rotating the whole scene and the camera about the cardinal axes.

It is straightforward to verify that the right nullspace of $\mathbf{M}\left(\mathbf{x}\right)$ spans seven directions, i.e., $\mathbf{M}\left(\mathbf{x}\right)\mathbf{N}_1 = \mathbf{0}$, where

$$\begin{aligned} \mathbf{N}_1 &= \begin{bmatrix} \mathbf{I}_3 & -\lfloor {}^G\mathbf{p}_{S(1)} \times \rfloor & {}^G\mathbf{p}_{S(1)} \\ \mathbf{0}_3 & {}^{S(1)}_{G}\mathbf{C} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_3 & \mathbf{0}_3 & {}^{S(1)}\mathbf{v} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times 1} \\ \mathbf{I}_3 & -\lfloor {}^G\mathbf{f} \times \rfloor & {}^G\mathbf{f} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{N}_{t,1} & | & \mathbf{N}_{r,1} & | & \mathbf{N}_{s,1} \end{bmatrix} \end{aligned} \tag{32}$$

where $\mathbf{N}_{t,1}$ corresponds to global translations of the camera and landmark together, $\mathbf{N}_{r,1}$ corresponds to global rotations of both together, and $\mathbf{N}_{s,1}$ is the direction corresponding to global scale (see Fig. 1).

Ideally, any estimator we employ should correspond to a system with an unobservable subspace that matches these directions, both in number and structure. However, when linearizing about the estimated state $\hat{\mathbf{x}}$, $\mathbf{M}\left(\hat{\mathbf{x}}\right)$ gains rank due to errors in the state estimates across time. This can be easily verified by numerically evaluating (19) during any experiment. To address this problem and ensure that (32) is orthogonal to every block row of $\mathbf{M}$ when the state estimates are used for computing $\mathbf{H}_\ell$, and $\mathbf{\Phi}_{\ell,1}$, $\ell = 1,\ldots,k$, we must ensure that $\mathbf{H}_\ell\mathbf{\Phi}_{\ell,1}\mathbf{N}_1 = \mathbf{0}$, $\ell = 1,\ldots,k$.

One way to enforce this is by requiring that at each time step

$$\mathbf{N}_{\ell+1} = \mathbf{\Phi}_\ell\mathbf{N}_\ell \tag{33}$$

$$\mathbf{H}_\ell\mathbf{N}_\ell = \mathbf{0}, \quad \ell = 1,\ldots,k \tag{34}$$

both hold. This can be accomplished by propagating the

nullspace in time and appropriately modifying $\mathbf{H}_\ell$ following the process described in the next section.

### 3.1. OC-MonoSLAM: Algorithm Description

Hereafter, we present our OC-MonoSLAM algorithm which enforces the observability constraints dictated by the MonoSLAM system structure. Rather than changing the linearization points explicitly (e.g., as in [7]), we maintain the nullspace, $\mathbf{N}_k$, at each time step, and use it to enforce the unobservable directions.

#### 3.1.1 Nullspace initialization for the camera

The initial nullspace corresponding to the camera state elements is analytically defined as

$$\mathbf{N}_1 = \begin{bmatrix} \mathbf{I}_3 & -\lfloor {}^G\hat{\mathbf{p}}_{S,0|0} \times \rfloor & {}^G\hat{\mathbf{p}}_{S,0|0} \\ \mathbf{0}_3 & \mathbf{C}\left({}^S\hat{\bar{q}}_{G,0|0}\right) & \mathbf{0}_{3\times 1} \\ \mathbf{0}_3 & \mathbf{0} & {}^S\hat{\mathbf{v}}_{0|0} \\ \mathbf{0}_3 & \mathbf{0} & \mathbf{0}_{3\times 1} \end{bmatrix} \tag{35}$$

where the $\hat{\mathbf{x}}_{i|j}$ denotes the estimate of $\mathbf{x}$ at time step $i$ based on all measurements up to time step $j$. We note that in SLAM it is common to (arbitrarily) assign the global frame to coincide with the initial camera frame, while the initial velocity can be set to be unity along the estimated direction of translation between the first image pair, to set the scale. However, any other preferred method for initializing the MonoSLAM state can also be employed to initialize the nullspace.

#### 3.1.2 Nullspace initialization for new landmarks

Each time a new landmark is initialized into the state vector, we must augment the nullspace, $\mathbf{N}_k$, so as to account for the new feature, and fulfill (33) and (34) at subsequent time steps. To accomplish this, we form the $3 \times 7$ block row

$$\mathbf{N}_{fi} = \begin{bmatrix} \mathbf{I}_3 & -\lfloor {}^G\hat{\mathbf{f}}_{k|k} \times \rfloor & {}^G\hat{\mathbf{f}}_{k|k} \end{bmatrix} \tag{36}$$

which we concatenate with the current nullspace $\mathbf{N}_k$.

#### 3.1.3 Nullspace propagation

During the propagation step, we need to compute the new nullspace at time $k+1$, $\mathbf{N}_{k+1}$. Based on the observability constraint (33), this entails propagating the nullspace from time step $k$ to $k+1$ using the computed state transition matrix $\mathbf{\Phi}_k$.

#### 3.1.4 Modification of H

During each update step, we must ensure that $\mathbf{H}_k\mathbf{N}_k = \mathbf{0}$ is satisfied. Hence, we seek a modified $\mathbf{H}_k$ that fulfills (34), while maintaining its structure. Based on (14), we can write this relationship *per feature* as

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} = \mathbf{H}_{cam} \begin{bmatrix} \mathbf{H}_\mathbf{p} & \mathbf{H}_{\bar{q}} & \mathbf{0}_3 & \mathbf{0}_3 & | & \mathbf{H}_\mathbf{f} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_3 & -\lfloor {}^G\hat{\mathbf{p}}_{S,k|k-1} \times \rfloor & {}^G\hat{\mathbf{p}}_{S,k|k-1} \\ \mathbf{0}_3 & \mathbf{C}\left({}^S\hat{\bar{q}}_{G,k|k-1}\right) & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & {}^G\hat{\mathbf{v}}_{S,k|k-1} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{I}_3 & -\lfloor {}^G\hat{\mathbf{f}}_{k|k-1} \times \rfloor & {}^G\hat{\mathbf{f}}_{k|k-1} \end{bmatrix} \tag{37}$$

The first block column of (37) requires that $\mathbf{H}_\mathbf{f} = -\mathbf{H}_\mathbf{p}$. Hence, we rewrite the second and third block columns of (37) as

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} \end{bmatrix} = \mathbf{H}_{cam} \begin{bmatrix} \mathbf{H}_\mathbf{p} & \mathbf{H}_{\bar{q}} \end{bmatrix} \cdot \begin{bmatrix} \lfloor {}^G\hat{\mathbf{f}}_{k|k-1} - {}^G\hat{\mathbf{p}}_{S,k|k-1} \times \rfloor & {}^G\hat{\mathbf{p}}_{S,k|k-1} - {}^G\hat{\mathbf{f}}_{k|k-1} \\ \mathbf{C}\left({}^S\hat{\bar{q}}_{G,k|k-1}\right) & \mathbf{0}_{3\times 1} \end{bmatrix} \tag{38}$$

This is a constraint of the form $\mathbf{0} = \mathbf{A}\mathbf{U}$, where $\mathbf{U}$ is a fixed quantity determined by elements in the nullspace, and $\mathbf{A}$ comprises elements of the measurement Jacobian, which we seek to modify. To compute the minimum perturbation, $\mathbf{A}^*$, of $\mathbf{A}$, we formulate the following minimization problem

$$\min_{\mathbf{A}^*} ||\mathbf{A}^* - \mathbf{A}||_\mathcal{F}^2 \ , \quad \text{s.t. } \mathbf{A}^*\mathbf{U} = \mathbf{0} \tag{39}$$

where $||\cdot||_\mathcal{F}$ denotes the Frobenius matrix norm. After employing the method of Lagrange multipliers, and solving the corresponding KKT optimality conditions, the optimal $\mathbf{A}^*$ that fulfills (39) is $\mathbf{A}^* = \mathbf{A} - \mathbf{A}\mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T$. Finally, the elements of the measurement Jacobian are computed as

$$\mathbf{H}_{cam}\mathbf{H}_\mathbf{p} = \mathbf{A}^*_{1:2,1:3} \tag{40}$$

$$\mathbf{H}_{cam}\mathbf{H}_\mathbf{f} = -\mathbf{A}^*_{1:2,1:3} \tag{41}$$

$$\mathbf{H}_{cam}\mathbf{H}_{\bar{q}} = \mathbf{A}^*_{1:2,4:6} \tag{42}$$

where the subscripts (i:j, m:n) denote the submatrix spanning rows i to j, and columns m to n. After computing the modified measurement Jacobian, we proceed with the filter update as described in Sect. 2.2.

## 4. Simulations

We conducted Monte-Carlo simulations to evaluate the impact of the proposed Observability-Constrained MonoSLAM (OC-MonoSLAM) method on estimator consistency. We compared its performance to standard MonoSLAM (Std-MonoSLAM), as well as an ideal MonoSLAM method that linearizes the Jacobians at the true state. We note that the ideal MonoSLAM is not realizable in practice, but is utilized as a benchmark for performance comparison.

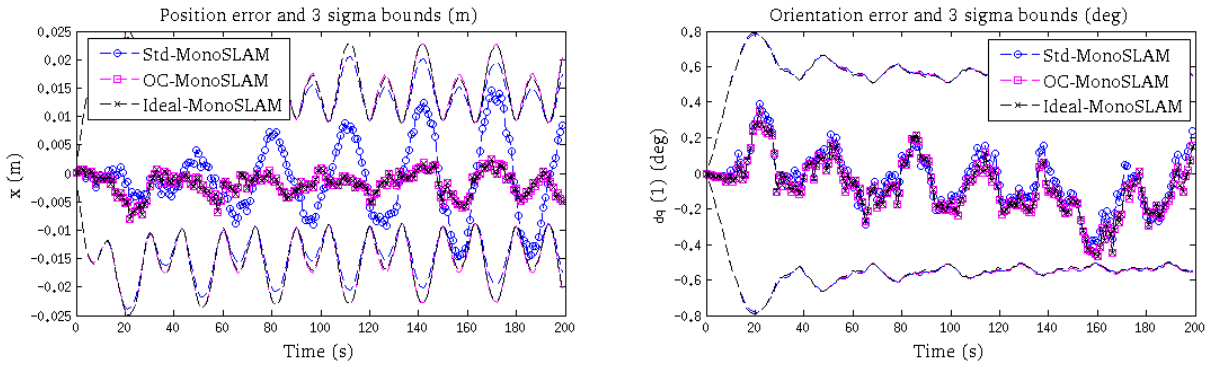


Figure 2: Errors and $3\sigma$ bounds plotted for the x-axis position (left) and $\delta\theta_1$ orientation (right) for the first 200 seconds of a representative run.

To evaluate the accuracy and consistency of the proposed approach, we computed the Root Mean Squared Error (RMSE) and Normalized Estimation Error Squared (NEES) [1] over 50 trials in which a simulated camera traversed a circular trajectory for 500 sec at an average speed of 11 cm/s.[3] The environment contained 72 visual features distributed in a planar grid pattern, which the camera observed while moving.

The effect of inconsistency during a single run is demonstrated in Fig. 2 where we depict the error and corresponding $3\sigma$ bounds for the x-axis position and $\delta\theta_1$ orientation. All three filters attain comparable accuracy and uncertainty for orientation, which is not surprising since there are sufficient points in the scene to precisely track the camera's rotations. However, from the position error plot, it is clear that the $3\sigma$ bounds for the Std-MonoSLAM are smaller than for either the OC-MonoSLAM, or the Ideal-MonoSLAM. This indicates that the Std-MonoSLAM gains spurious information. Furthermore, the x-axis position error for Std-MonoSLAM starts to increase over time, eventually causing inconsistency.

Figure 3 displays the RMSE and NEES, in which we observe that all three filters obtain similar accuracy and consistency performance for orientation. However, the OC-MonoSLAM attains significantly better positioning accuracy and consistency compared to Std-MonoSLAM, and is almost indistinguishable from the Ideal-MonoSLAM. Based on our analysis and these results, we postulate that the key source of position error and inconsistency in the Std-MonoSLAM is violation of the unobservable scale direction [i.e., $\mathbf{N}_s$, see (32)].

[3]The camera was simulated with a 45x45 deg fov, with $\sigma_{px} = 1px$.

## 5. Experimental Validation

Our experimental set-up comprised a monochrome Point Grey Chameleon camera which recorded images at 7.5 Hz. We moved the camera on a circular trajectory in front of a calibration board comprising 72 corner features, whose positions are accurately known Fig. 4.

Using the observations of the visual features over 25 seconds (approx. 4.5 rotations), we estimated the camera trajectory and corresponding map using both the Std-MonoSLAM and the OC-MonoSLAM methods. The filters were initialized using the PnP estimate of the camera pose at the first image, along with the linear and rotation velocities computed between the first two images. In order to obtain an "approximate" ground truth trajectory, we utilized DLS-PnP [6] to compute the camera pose independently for each image, given the known landmark locations.

The estimated 3D trajectories and maps are depicted in Fig. 4. The PnP trajectory is plotted in black and closely coincides with the one computed by OC-MonoSLAM, while the Std-MonoSLAM position estimates follow an estimated circular trajectory with a smaller radius (indicating inconsistent scale). The scale inconsistency is also visually apparent in Fig. 4 (right), which depicts a top view of the landmarks and trajectories. The true landmarks lie in the $y = 0$ cm plane, hence, the Std-MonoSLAM underestimates the depth to the scene.

In Fig. 5, we plot the estimated $3\sigma$ bounds and corresponding errors with respect to the PnP trajectory for two representative axes (i.e., x-axis position and $\delta\theta_1$ orientation). It is evident that the orientation performance of both filters is comparable, while the OC-MonoSLAM outperforms the Std-MonoSLAM in position accuracy. In addition, the OC-MonoSLAM is more conservative that the Std-MonoSLAM in terms of position uncertainty.

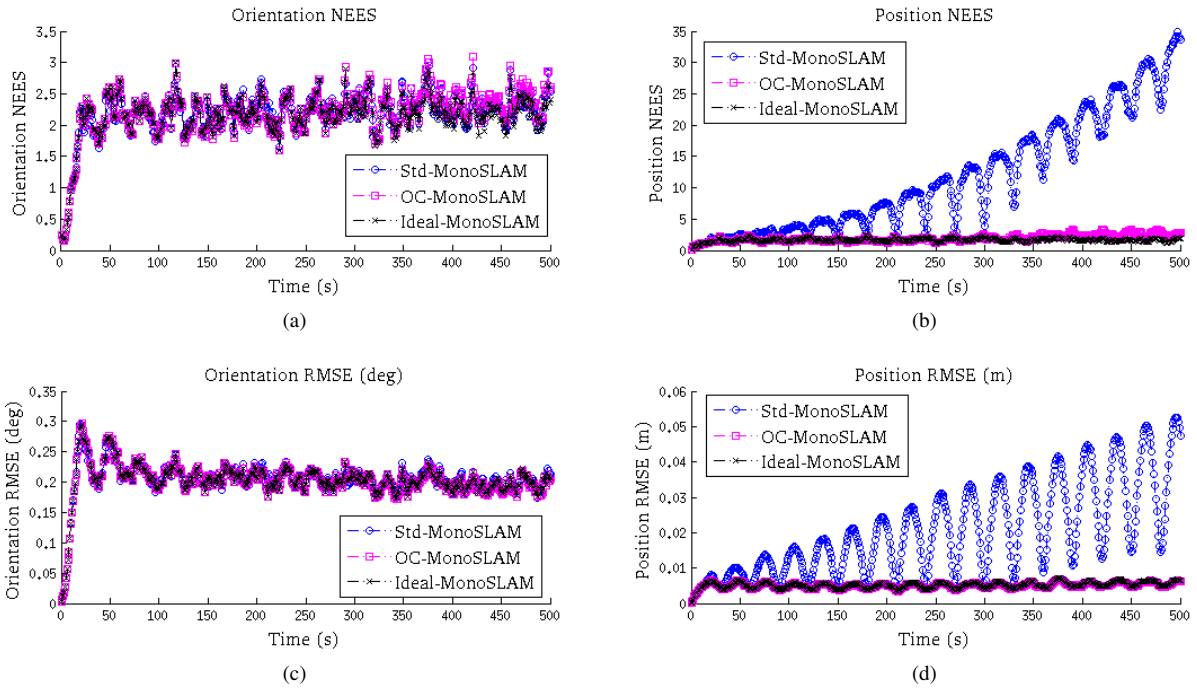


Figure 3: The NEES and RMSE for orientation (left) and position (right) plotted for all three filters, averaged per time step over 50 Monte-Carlo trials.

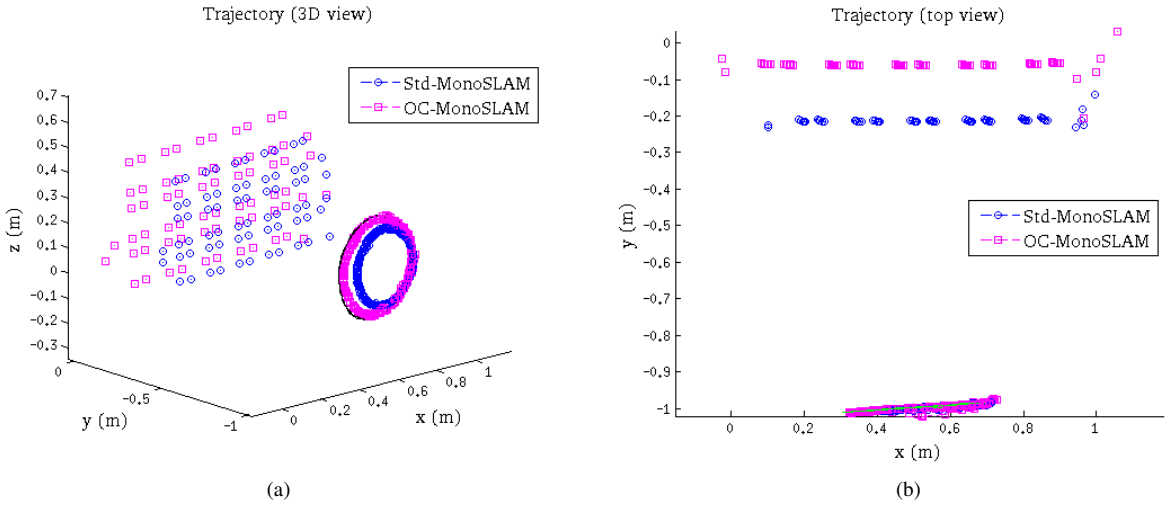


Figure 4: (left) The estimated 3D trajectory for the Std-MonoSLAM and the OC-MonoSLAM, along with the estimated map. The PnP estimated trajectory is plotted in black, and is overlapped by the OC-MonoSLAM estimate. (right) A top view of the trajectories and landmarks. The true landmarks lie on the y = 0 plane, hence the Std-MonoSLAM underestimates the depth to the scene, demonstrating scale drift.

## 6. Conclusion and Future Work

In this paper, we analyzed the inconsistency of MonoSLAM from the standpoint of observability. Specifically, we showed that using a standard EKF-based approach leads to spurious information gain, in particular for scale, since it does not adhere to the unobservable directions of the true system. Moreover, we introduced an observability-

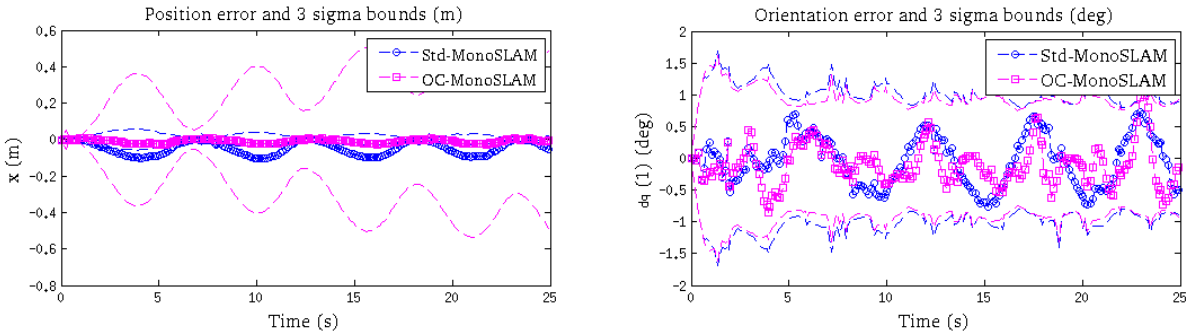


Figure 5: (left) The position error and corresponding $3\sigma$ bounds for the x-axis computed with respect to the PnP pose estimates. (right) The orientation error and $3\sigma$ bounds for $\delta\theta_1$.

constrained MonoSLAM method to mitigate estimator inconsistency by enforcing the nullspace explicitly. Finally, we presented simulation and experimental results to support our claims and validate the proposed estimator. In our future work, we are interested in analyzing additional potential sources of estimator inconsistency in MonoSLAM such as the existence of multiple local minima.

## References


[1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, New York, NY, 2001. 1, 6

[2] J.-Y. Bouguet. Camera calibration toolbox for matlab, 2006. 3

[3] R. Castle, G. Klein, and D. Murray. Combining MonoSLAM with object recognition for scene augmentation using a wearable camera. *Image and Vision Computing*, 28(11):1548–1556, Nov. 2010. 1

[4] J. Civera, A. J. Davison, and J. M. M. Montiel. Interacting multiple model monocular SLAM . In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 3704–3709, Pasadena, CA, May 19–23, 2008. 2

[5] A. J. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, June 2007. 1, 2

[6] J. A. Hesch and S. I. Roumeliotis. A direct least-squares (dls) solution for PnP. In *Proc. of the Int. Conf. on Computer Vision*, Barcelona, Spain, Nov. 6–13, 2011. 6

[7] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis. Analysis and improvement of the consistency of extended kalman filter based SLAM. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 373–382, Pasadena, CA, May 19–23, 2008. 1, 5

[8] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis. Observability-based rules for designing consistent EKF SLAM estimators. *Int. Journal of Robotics Research*, 29(5):502–528, Apr. 2010. 1

[9] G. P. Huang, N. Trawny, A. I. Mourikis, and S. I. Roumeliotis. Observability-based consistent EKF estimators for multi-robot cooperative localization. *Autonomous Robots*, 30(1):99–122, Jan. 2011. 1

[10] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. of the IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234, Nara, Japan, Nov. 13–16, 2007. 1

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, Nov. 2004. 2

[12] P. S. Maybeck. *Stochastic models, estimation, and control*, volume I. Academic Press, New York, NY, 1979. 4

[13] N. Molton, S. Se, J. M. Brady, D. Lee, and P. Probert. A stereo vision-based aid for the visually impaired. *Image and Vision Computing*, 16(4):251–263, Mar. 1998. 1

[14] R. A. Newcombe, S. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *International Conf. on Computer Vision*, pages 2320–2327, Barcelona, Spain, Nov. 6–13, 2011. 1

[15] N. S underhauf, S. Lange, and P. Protzel. Using the unscented Kalman filter in mono-SLAM with inverse depth parametrization for autonomous airship control. In *Proc. of the IEEE International Workshop on Safety, Security, and Rescue Robotics*, pages 1–6, Rome, Italy, Sept. 27–29, 2007. 1

[16] J. Teddy Yap, M. Li, A. I. Mourikis, and C. R. Shelton. A particle filter for monocular vision-aided odometry. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 5663–5669, Shanghai, China, May 9–13, 2011. 1

[17] N. Trawny and S. I. Roumeliotis. Indirect Kalman filter for 3D attitude estimation. Technical Report 2005-002, University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, Mar. 2005. 2

[18] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer-Verlag, 2000. 3