Vision-based Localization and Robot-centric Mapping in Riverine Environments

Junho Yang
Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801
e-mail: yang125@illinois.edu
Ashwin Dani
Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801
e-mail: adani@illinois.edu
Soon-Jo Chung
Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801
e-mail: sjchung@illinois.edu
Seth Hutchinson
Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801
e-mail: seth@illinois.edu

Received 2 May 2014; accepted 28 April 2015

This paper presents a vision-based localization and mapping algorithm developed for an unmanned aerial vehicle (UAV) that can operate in a riverine environment. Our algorithm estimates the three-dimensional positions of point features along a river and the pose of the UAV. By detecting features surrounding a river and the corresponding reflections on the water's surface, we can exploit multiple-view geometry to enhance the observability of the estimation system. We use a robot-centric mapping framework to further improve the observability of the estimation system while reducing the computational burden. We analyze the performance of the proposed algorithm with numerical simulations and demonstrate its effectiveness through experiments with data from Crystal Lake Park in Urbana, Illinois. We also draw a comparison to existing approaches. Our experimental platform is equipped with a lightweight monocular camera, an inertial measurement unit, a magnetometer, an altimeter, and an onboard computer. To our knowledge, this is the first result that exploits the reflections of features in a riverine environment for localization and mapping. © 2015 Wiley Periodicals, Inc.

1. INTRODUCTION

Recent advances in navigation technologies using onboard local sensing modalities are enabling intelligence, surveillance, and reconnaissance (ISR) missions by unmanned aerial vehicles (UAVs) in a range of diverse environments (Bachrach et al., 2012; Bryson & Sukkarieh, 2007; Chowdhary, Johnson, Magree, Wu, & Shein, 2013; Doitsidis et al., 2012; Weiss et al., 2013) where a GPS signal is intermittent or unavailable. Our goal in this research is to further expand the scope of future ISR missions by developing a localization and mapping algorithm particularly for a riverine environment. Navigating a UAV in a riverine environment is challenging due to the limited power and payload of a UAV. We present a localization and mapping algorithm that uses a lightweight monocular camera, an inertial measurement unit (IMU) integrated with a magnetometer, and an altimeter, which are typically available onboard for the autopilot control of a UAV.

A primary contribution of this work is to exploit a multiple-view geometry formulation with initial and current view projection of point features from real objects surrounding the river and their reflections. The correspondences of the features are used along with the attitude and altitude information of the UAV to improve the observability of the estimation system. If the features observed from the camera are in distant range while the UAV is navigating along the river, neither monocular simultaneous localization and mapping (SLAM) methods that do not use reflections nor stereovision SLAM with limited baseline distance would provide sufficient information to estimate the translation of the UAV. Light detection and ranging (LIDAR) sensors that are capable of ranging distant features can weigh too much for a small UAV. Recently, visual SLAM with an RGB-D camera (Kerl, Sturm, & Cremers, 2013) has become

Ashwin Dani is currently with Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06042 (ashwin.dani@engr.uconn.edu)
Direct correspondence to Seth Hutchinson: seth@illinois.edu

popular due to the dense depth information the sensor can provide, but the sensor is only functional in indoor environments where there is not much interference of sunlight.

Another contribution of our work is to use a robot-centric mapping framework concurrently with world-centric localization of the UAV. We exploit the differential equation of motion of the normalized pixel coordinates of each point feature in the UAV body frame in contrast with prior work using robot-centric SLAM, which estimates both the world frame and the features with respect to the robot's current pose indirectly through a composition stage. In this paper, we demonstrate that the observability of the estimation system is improved by applying the proposed robot-centric mapping strategy.

The rest of this paper is organized as follows. In Section 2, we provide an overview of related work. In Section 3, we describe our experimental platform and explain our motion models of both the UAV and the robot-centric estimates of point features. We also present our measurement model, which includes reflection measurements. In Section 4, we formulate an extended Kalman filter (EKF) estimator for UAV localization and point feature mapping. In Section 5, we analyze the observability of our estimation system under various conditions and show the advantage of our method. In Section 6, we validate the performance of our algorithm with numerical simulation results. In Section 7, we show experimental results of our monocular vision-based localization and mapping algorithm at Crystal Lake Park in Urbana, Illinois. In Section 8, we summarize our work with concluding remarks.

2. RELATED WORK

The problem of navigating a vehicle in an unknown environment with local measurements can be addressed by progressively constructing a map of the surroundings while localizing the vehicle within the map, a process known as SLAM (Bailey & Durrant-Whyte, 2006; Choset et al., 2005; Durrant-Whyte & Bailey, 2006). In Scherer et al. (2012), a navigation algorithm particularly suited for riverine environments is presented. A graph-based state estimation framework (Rehder, Gupta, Nuske, & Singh, 2012) is used to estimate the vehicle's state with vision and limited GPS. The river is mapped with a self-supervised river detection algorithm, and the obstacles are mapped with a LIDAR sensor. Autonomous flight in a riverine environment has also been demonstrated recently (Jain et al., 2013). Our prior work (Yang, Rao, Chung, & Hutchinson, 2011) presents a monocular vision-based SLAM algorithm that uses a planar ground assumption for riverine environments. In Leedekerken, Fallon, & Leonard (2010), a submapping approach is adopted to address the SLAM problem with an autonomous surface-craft that builds a map above and below the water's surface. Sonar is used for subsurface mapping while a LIDAR sensor, a camera, and a radar system are used for terrestrial mapping to account for degradation of GPS measurements. In Fallon, Papadopoulos, Leonard, & Patrikalakis (2010), a surface-craft equipped with an acoustic modem is used to support localization of autonomous underwater vehicles. A departure from point-feature based SLAM is reported using higher-level features represented by Bézier curves as stereo vision primitives (Rao, Chung, & Hutchinson, 2012) or tracking the image moments of region features (Dani, Panahandeh, Chung, & Hutchinson, 2013) with a new stochastic nonlinear estimator (Dani, Chung, & Hutchinson, 2015).

We employ a monocular camera as our primary sensor to solve six-degree-of-freedom (DOF) SLAM, but monocular vision presents a challenge because the distance to a feature cannot be directly estimated from a single image. In Davison, Reid, Molton, & Stasse (2007), a monocular vision-based SLAM problem is solved by sequentially updating measurements from different locations. The map is updated with new features after the camera moves sufficiently. Instead of estimating the Cartesian coordinates of features in the world reference frame, some recent work (Ceriani, Marzorati, Matteucci, Migliore, & Sorrenti, 2011; Civera, Davison, & Montiel, 2008; Sola, Vidal-Calleja, Civera, & Montiel, 2012) defines the locations of the moving camera (anchor locations) where the point features are first observed. The point features are parametrized using these anchor locations, the direction of each feature with respect to the world reference frame, and the distance between the feature and the anchor. Such methods reduce accumulation of the linearization errors by representing the uncertainty of the features with respect to a close-by anchor location. The inverse-depth parametrization (IDP) is used to alleviate the nonlinearity of the measurement model and introduce new features to the map immediately. The inverse-depth method ameliorates the known problem of the EKF-based monovision SLAM (Sola et al., 2012), which often appears when the features are estimated in Cartesian coordinates. We shall compare our localization and mapping approach with the anchored IDP method in this work.

The computational issues of EKF-based SLAM are addressed by keyframe-based optimization (Forster, Pizzoli, & Scaramuzza, 2014; Kaess et al., 2012; Kim, & Eustice, 2013; Klein & Murray, 2007; Leutenegger et al., 2013; Strasdat, Montiel, & Davison, 2010) and submapping (Clemente, Davison, Reid, Neira, & Tardós, 2007; Leedekerken et al., 2010). Keyframe-based methods select a small number of frames and solve the bundle adjustment problem (Strasdat et al., 2010) with multiple views. In this work, we consider the dynamics of the vehicle with an estimation filter and apply multiple-view measurements that include the initial view, current view, and reflection view of features in order to improve the observability of the estimation system. In keyframe-based optimization research, parallel tracking and mapping (PTAM) (Klein & Murray, 2007) achieves real-time processing by separating the tracking of the camera

and mapping of the environment into two parallel tasks. The UAV navigation (Weiss, Scaramuzza, & Siegwart, 2011; Weiss et al., 2013) and surveillance (Doitsidis et al., 2012) problems are addressed based on the PTAM method. In Mourikis & Roumeliotis (2007), the multistate constraint Kalman filter (MSC-KF) adds the estimate of the camera pose to the state vector every time a new image is received and gains optimality up to linearization errors without including the feature estimates in the state vector. The history of the camera pose is truncated from the estimation state vector after it reaches the maximum allowable number.

The observability problems of vision-aided inertial navigation systems (VINSs) (Hesch, Kottas, Bowman, & Roumeliotis, 2014; Kelly & Sukhatme, 2011; Martinelli, 2014; Weiss et al., 2013) and SLAM (Bryson & Sukkarieh, 2008; Huang, Mourikis, & Roumeliotis, 2010; Lee, Wijesoma, & Guzman, 2006) have been studied in the literature. In general, *a priori* knowledge of the position of a set of features in the map is required for the system to be observable. The three-dimensional (3D) location of the robot and its orientation with respect to the gravity vector in the world frame (e.g., heading angle) are the unobservable modes of a world-centric 6-DOF localization and 3D mapping system that uses a monocular camera and inertial sensors (Hesch et al., 2014). The MSC-KF (Mourikis & Roumeliotis, 2007) and the observability constrained EKF (Huang et al., 2010), which finds a linearization point that can preserve the unobservable subspace while minimizing the linearization error, are applied to a visual-inertial navigation system in Hesch et al. (2014) to improve the consistency of the estimation.

As stated in Section 1, we present a world-centric localization and robot-centric mapping system and exploit multiple views from real point features and their reflections from the surface of the river to improve the observability of the estimation system. Observations of known points through a mirror are used to estimate the 6-DOF pose of the camera with a maximum-likelihood estimator in Hesch, Mourikis, & Roumeliotis (2009). In Panahandeh & Jansson (2011), an approach for estimating the camera intrinsic parameters as well as the 6-DOF transformation between an IMU and a camera by using a mirror is proposed. In Mariottini, Scheggi, Morbidi, & Prattichizzo (2012), epipolar geometry with multiple planar mirrors is used to compute the location of a camera and reconstruct a 3D scene in an indoor experimental setup. In contrast, we exploit geometrical constraints from reflection measurements in a natural environment for localization and mapping.

Robot-centric estimation, as opposed to world-centric SLAM, is used with different meanings and purposes. Robot-centric SLAM for both localization and mapping is introduced in Castellanos, Martinez-cantin, Tardos, & Neira (2007) and applied to monocular visual odometry in Civera, Grasa, Davison, & Montiel (2010) and Williams, & Reid (2010). The method defines the origin on the current robot frame and estimates both the previous pose of the robot and the location of the features with respect to the current robot frame. This scheme reduces the uncertainty in the estimate and alleviates the linearization error in the EKF. Another category of robot-centric work (Boberg, Bishop, & Jensfelt, 2009; Haner & Heyden, 2010) estimates the features with respect to the robot by using a dynamic model with velocity and angular velocity information without estimating the pose of the robot and circumvents the observability issue in SLAM. Nonlinear observers are derived (Dani, Fischer, & Dixon, 2012; Dixon, Fang, Dawson, & Flynn, 2003; Jankovic & Ghosh, 1995; Yang, Chung, Hutchinson, Johnson, & Kise, 2013a) for feature tracking and depth estimation, which can also be viewed as robot-centric mapping with a monocular camera.

In this paper, we employ a robot-centric feature mapping framework for localization and mapping in riverine environments primarily with a monocular camera. We exploit the differential equation of motion of the normalized pixel coordinates of each point feature in the UAV body frame in contrast with prior work using robot-centric SLAM, which estimates both the world frame and the features with respect to the current pose of the robot indirectly through a composition stage. We estimate the state of the robot along with the features by composing a measurement model with multiple views of point features surrounding the river and the reflections on the water's surface. The observability of the estimation system is substantially improved as a result. Our previous work (Yang, Dani, Chung, & Hutchinson, 2013b) showed preliminary results of the localization and robot-centric mapping by using reflections. In this work, we analyze the observability of our estimation system under various conditions and validate the effectiveness of our algorithm through numerical simulations and experiments in a larger environment. To our knowledge, we report the first result of exploiting the reflections of features for localization and mapping in a riverine environment.

3. RIVERINE LOCALIZATION AND MAPPING SYSTEM

In this section, we describe the overall architecture of our riverine localization and mapping algorithm. We present the motion model for the localization of the UAV and the robot-centric mapping of point features. We also derive the measurement model with multiple views of each point feature and its reflection.

3.1. Overview of The Experimental Platform

Our system estimates features with respect to the UAV body frame while estimating the location of the UAV in the world frame. Figure 1 shows the block diagram of our riverine localization and mapping system. We define our world reference frame with the projection of the X- and Y-axes of the UAV body frame on the river surface when the estimation begins. The Z-axis points downward along the gravity

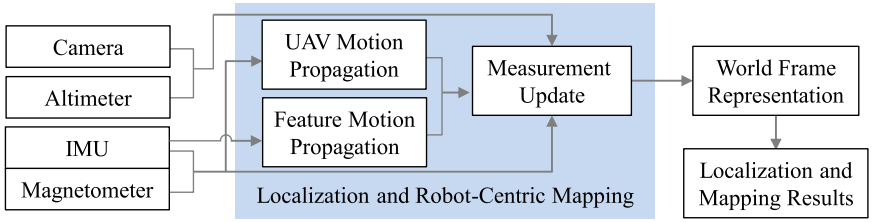


Figure 1. The block diagram of our riverine localization and mapping system.

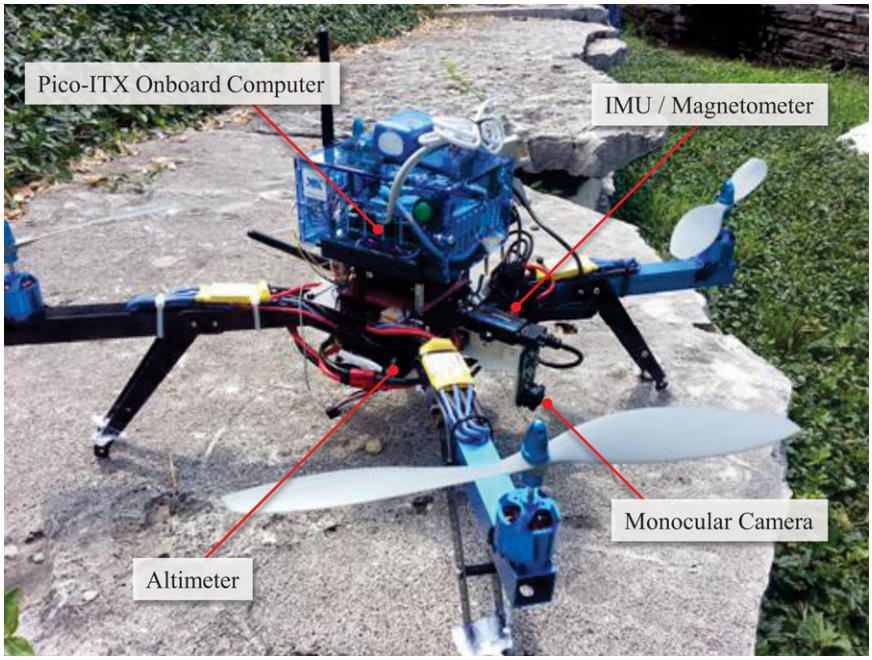


Figure 2. Our quadcopter is equipped with a lightweight monocular camera, an IMU, a magnetometer, an altimeter, and a compact Pico-ITX onboard computer.

vector (see Figure 3). We set the origin of the UAV body frame on the center of the IMU, which is mounted on the UAV, and we define the UAV body frame as the coordinate frame of the IMU. We use onboard sensor readings for the motion propagation and the measurement update stages of our EKF estimator in order to simplify the process and alleviate the nonlinearity of the system.

Figure 2 shows our quadcopter, which contains a lightweight monocular camera facing forward with a resolution of 640×480 pixels, a three-axis IMU and a magnetometer, an ultrasound/barometric altimeter, and a compact Pico-ITX onboard computer equipped with a 64-bit VIA Eden X2 dual core processor and a VIA VX900H media system processor. The distance between the UAV and the surface of the river is measured with the altimeter. For the propagation stage of the filter, the motion model of the UAV is derived to use the IMU and magnetometer readings, and the motion model of each feature incorporates gyroscope measurements. In the measurement update stage, the measurement model is formulated with multiple views as follows. We project the features to the camera upon their first and current observations. The measurement model is augmented with observations of corresponding reflection points and the altitude readings of the UAV.

3.2. Dynamic Model

We describe the motion model used for localization of the UAV in the world reference frame and the estimation of point features with respect to the UAV body frame.

3.2.1. Dynamic Model for the Riverine Localization and Mapping

The state vector for our estimation system consists of $\mathbf{p}_b^w \equiv (x_b^w, y_b^w, z_b^w)^T \in \mathbb{R}^3$, $\mathbf{v}^b \equiv (v_1, v_2, v_3)^T \in \mathbb{R}^3$, $\mathbf{b}_a^b \in \mathbb{R}^3$, $\mathbf{q}_b^w \in \mathbb{H}$, $\mathbf{b}_g^b \in \mathbb{R}^3$, and $\mathbf{x}_i^b \equiv ((\mathbf{h}_i^b)^T, \rho_i^b)^T \in \mathbb{R}^3$, where $\mathbf{p}_b^w$ and $\mathbf{q}_b^w$ are the location and the attitude quaternion of the UAV's body

with respect to the world reference frame, $\mathbf{v}^b$ is the velocity of the UAV with respect to the UAV body frame, and $\mathbf{b}_a^b$ and $\mathbf{b}_g^b$ are the bias of the accelerometer and the gyroscope. The subscript (or superscript) b denotes the UAV body frame, and w represents the world reference frame. The vector $\mathbf{x}_i^b$ for the ith landmark consists of its normalized coordinates $\mathbf{h}_i^b = (h_{1,i}^b, h_{2,i}^b)^T = (y_i^b/x_i^b, z_i^b/x_i^b)^T \in \mathbb{R}^2$ and its inverse-depth $\rho_i^b = 1/x_i^b \in \mathbb{R}^+$ from the UAV along the X-axis of the UAV body frame, where the vector $\mathbf{p}_i^b = (x_i^b, y_i^b, z_i^b)^T \in \mathbb{R}^3$ is the Cartesian coordinates of the feature with respect to the UAV body frame. We get the acceleration $\mathbf{a}^b = \tilde{\mathbf{a}}^b - \mathbf{b}_a^b \in \mathbb{R}^3$ by subtracting the accelerometer bias $\mathbf{b}_a^b$ from the accelerometer readings $\tilde{\mathbf{a}}^b \in \mathbb{R}^3$, and the angular velocity $\boldsymbol{\omega}^b \equiv (\omega_1, \omega_2, \omega_3)^T = \tilde{\boldsymbol{\omega}}^b - \mathbf{b}_g^b \in \mathbb{R}^3$ by subtracting the gyroscope bias $\mathbf{b}_g^b$ from the gyroscope readings $\tilde{\boldsymbol{\omega}}^b \in \mathbb{R}^3$ as shown in Kelly, & Sukhatme (2011).

The dynamic model for our estimation system is given by

$$\frac{d}{dt}\begin{pmatrix} \mathbf{p}_b^w \\ \mathbf{v}^b \\ \mathbf{b}_a^b \\ \mathbf{q}_b^w \\ \mathbf{b}_g^b \\ \mathbf{x}_1^b \\ \vdots \\ \mathbf{x}_n^b \end{pmatrix} = \begin{pmatrix} R\left(\mathbf{q}_b^w\right)\mathbf{v}^b \\ -[\boldsymbol{\omega}^b]_\times \mathbf{v}^b + \mathbf{a}^b + R^T\left(\mathbf{q}_b^w\right)\mathbf{g}^w \\ \mathbf{0} \\ \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}^b)\mathbf{q}_b^w \\ \mathbf{0} \\ \mathbf{f}\left(\mathbf{x}_1^b, \mathbf{v}^b, \boldsymbol{\omega}^b\right) \\ \vdots \\ \mathbf{f}\left(\mathbf{x}_n^b, \mathbf{v}^b, \boldsymbol{\omega}^b\right) \end{pmatrix}, \tag{1}$$

where $(\mathbf{x}_1^b \cdots \mathbf{x}_n^b)$ are the state vectors of n point features, and $\mathbf{g}^w \in \mathbb{R}^3$ is the gravity vector in the world reference frame. We shall define the motion model $\mathbf{f}(\mathbf{x}_i^b, \mathbf{v}^b, \boldsymbol{\omega}^b)$ of the ith feature in Section 3.2.2. The skew-symmetric matrix $[\boldsymbol{\omega}^b]_\times \in$ so(3) is constructed from the angular velocity vector $\boldsymbol{\omega}^b$, and $\Omega(\boldsymbol{\omega}^b)$ is given by

$$\Omega(\boldsymbol{\omega}^b) \equiv \begin{pmatrix} -[\boldsymbol{\omega}^b]_\times & \boldsymbol{\omega}^b \\ -\left(\boldsymbol{\omega}^b\right)^T & 0 \end{pmatrix}. \tag{2}$$

The motion model for each feature $\mathbf{x}_i^b$ in Eq. (1) requires the velocity of the UAV in the UAV body frame of reference, as shall be shown in Eq. (4). Therefore, we employ the time derivative of the UAV's velocity, which considers the acceleration $\mathbf{a}^b$ and the angular velocity $\boldsymbol{\omega}^b$ in the UAV body frame instead of integrating the acceleration $\mathbf{a}^w$ in the world reference frame.

3.2.2. Vision Motion Model for the Robot-centric Mapping

We perform robot-centric mapping to generate a 3D point feature-based map. The method references the point features to the UAV body frame and mainly considers the current motion of the UAV to estimate the position of the features. We provide the observability analysis of our estimation system in Section 5.

The position of each point feature is first estimated in the UAV body frame. The dynamics of the ith feature in Cartesian coordinates is given in Chaumette & Hutchinson (2006) as follows:

$$\frac{d}{dt}\mathbf{p}_i^b = -[\boldsymbol{\omega}^b]_\times \mathbf{p}_i^b - \mathbf{v}^b, \tag{3}$$

where $\mathbf{p}_i^b$ is the location of the ith feature with respect to the UAV body frame. We represent the vector $\mathbf{p}_i^b$ of the feature with normalized coordinates $\mathbf{h}_i^b \equiv (h_{1,i}^b, h_{2,i}^b)^T$ and the inverse-depth ρ_i^b. In Dani et al. (2012), a model that consists of the normalized pixel coordinates $\mathbf{h}_i^c \equiv (h_{1,i}^c, h_{2,i}^c)^T \in \mathbb{R}^2$ and the inverse depth $\rho_i^c \in \mathbb{R}^+$ of a point feature with respect to the camera coordinate frame is used to estimate the location of the point, along with the angular velocity and two of the velocity components of the camera. In this work, we employ the robot-centric mapping framework and formulate a system for both localization and mapping. We derive the dynamics $[\dot{\mathbf{x}}_i^b = \mathbf{f}(\mathbf{x}_i^b, \mathbf{v}^b, \boldsymbol{\omega}^b)]$ of the ith feature referenced with respect to the UAV body frame from Eq. (3) as

$$\begin{aligned} &\frac{d}{dt}\begin{pmatrix} h_{1,i}^b \\ h_{2,i}^b \\ \rho_i^b \end{pmatrix} \\ &= \begin{pmatrix} \left(-v_2 + h_{1,i}^b v_1\right)\rho_i + h_{2,i}^b\omega_1 - \left(1 + \left(h_{1,i}^b\right)^2\right)\omega_3 + h_{1,i}^b h_{2,i}^b \omega_2 \\ \left(-v_3 + h_{2,i}^b v_1\right)\rho_i - h_{1,i}^b\omega_1 + \left(1 + \left(h_{2,i}^b\right)^2\right)\omega_2 - h_{1,i}^b h_{2,i}^b \omega_3 \\ \left(-\omega_3 h_{1,i}^b + \omega_2 h_{2,i}^b\right)\rho_i^b + v_1\left(\rho_i^b\right)^2 \end{pmatrix}, \end{aligned} \tag{4}$$

where $\mathbf{x}_i^b \equiv ((\mathbf{h}_i^b)^T, \rho_i^b)^T$ represents the vector of the ith landmark from the UAV body frame.

The model in Eq. (4) is similar to the one presented in Dani et al. (2012), but the model is augmented with the UAV localization part. We construct the motion model in Eq. (1) for the localization and mapping by combining the dynamic model of the UAV and the vision motion model given by Eq. (4). The estimator that we will present in Section 4 exploits the motion model given by Eqs. (1) and (4).

3.3. Vision Measurement Model

We describe our vision measurement model for our estimation system. The vision measurements consist of the projection of each point feature at the first and current observations and its reflection.

3.3.1. Projected Measurements of Features

We compute the normalized pixel coordinates $\mathbf{h}_i^c$ of the ith point feature in the camera coordinate frame with $\mathbf{h}_i^c = ((x_i^m - x_0^m)/\lambda\alpha, (y_i^m - y_0^m)/\lambda)^T$, where (x_i^m, y_i^m) is the

pixel coordinates of the feature, (x_0^m, y_0^m) is the coordinates of the principal point, λ is the focal length of the camera lens, and α is the ratio of the pixel dimensions (Chaumette & Hutchinson, 2006). The camera coordinate frame is assigned with a rightward-pointing X-axis, a downward-pointing Y-axis, which forms the basis for the image plane, and a Z-axis perpendicular to the image plane along the optical axis. Also, the camera coordinate frame has an origin located at distance λ behind the image plane. We compute the unit vector $\mathbf{p}_{s,i}^c \equiv (x_{s,i}^c, y_{s,i}^c, z_{s,i}^c)^T \in \mathbb{R}^3$ to the feature with respect to the camera coordinate frame from the normalized pixel coordinates $\mathbf{h}_i^c$. The subscript s stands for the unit sphere projection of a vector. We get the unit vector $\mathbf{p}_{s,i}^b \equiv (x_{s,i}^b, y_{s,i}^b, z_{s,i}^b)^T \in \mathbb{R}^3$ to the feature with respect to the UAV body frame from $\mathbf{p}_{s,i}^b = R(\mathbf{q}_c^b)\mathbf{p}_{s,i}^c$ since the distance between our IMU and the camera is negligible. Here, $\mathbf{q}_c^b$ is the orientation quaternion of the camera with respect to the UAV body frame, which we get from the IMU-camera calibration. Note that $\mathbf{p}_{s,i}^b$ is a unit sphere projection of the vector $\mathbf{p}_i^b \equiv (x_i^b, y_i^b, z_i^b)^T \in \mathbb{R}^3$ of the feature, which is referenced with respect to the UAV body frame. We compute the normalized coordinates $\mathbf{h}_i^b \equiv (h_{1,i}^b, h_{2,i}^b)^T = (y_i^b/x_i^b, z_i^b/x_i^b)^T = (y_{s,i}^b/x_{s,i}^b, z_{s,i}^b/x_{s,i}^b)^T$ of the ith feature in the UAV body frame with the elements of the unit vector $\mathbf{p}_{s,i}^b$.

We define $\mathbf{p}_{bi}^w \in \mathbb{R}^3$ and $\mathbf{q}_{bi}^w \in \mathbb{H}$ as the location and the attitude quaternion of the UAV when the estimator first incorporates the ith feature to the state vector. The current location of the UAV with respect to $(\mathbf{p}_{bi}^w, \mathbf{q}_{bi}^w)$ is given by $\mathbf{p}_b^{bi} = R^T(\mathbf{q}_{bi}^w)(\mathbf{p}_b^w - \mathbf{p}_{bi}^w) \in \mathbb{R}^3$, and the current attitude quaternion of the UAV with respect to $\mathbf{q}_{bi}^w$ is denoted by $\mathbf{q}_b^{bi} \in \mathbb{H}$, where $R(\mathbf{q}_b^{bi}) = R^T(\mathbf{q}_{bi}^w)R(\mathbf{q}_b^w)$. We reference the ith feature with respect to $(\mathbf{p}_{bi}^w, \mathbf{q}_{bi}^w)$ as $\mathbf{p}_i^{bi} \equiv (x_i^{bi}, y_i^{bi}, z_i^{bi})^T \in \mathbb{R}^3$ and express it in terms of the state of the UAV and the feature itself as

$$\mathbf{p}_i^{bi} = R^T(\mathbf{q}_{bi}^w)(\mathbf{p}_b^w - \mathbf{p}_{bi}^w) + R^T(\mathbf{q}_{bi}^w)R(\mathbf{q}_b^w)\mathbf{p}_i^b, \tag{5}$$

where the vector $\mathbf{p}_i^b$ of the feature with respect to the UAV body frame is given by $\mathbf{p}_i^b = (1/\rho_i^b, h_{1,i}^b/\rho_i^b, h_{2,i}^b/\rho_i^b)^T$. We include the initial normalized coordinates $\mathbf{h}_i^{bi} = (y_i^{bi}/x_i^{bi}, z_i^{bi}/x_i^{bi})^T \in \mathbb{R}^2$ of the ith feature in the measurement vector and exploit multiple views, as shall be seen in Section 4.2. The initial normalized coordinates $\mathbf{h}_i^{bi}$ define a constant vector that is identical to the normalized coordinates $\mathbf{h}_i^b$ of the ith feature upon its first observation.

3.3.2. Measurements of Reflections of Features

Reflection of the surrounding environment is an important aspect of riverine environments. We express the reflection of the ith point feature $\mathbf{p}_i^w$ that we measure with the camera as $\mathbf{p}_{r,i}^b \equiv (x_{r,i}^b, y_{r,i}^b, z_{r,i}^b)^T \in \mathbb{R}^3$ in the UAV body frame, and we define a mirrored point as $\tilde{\mathbf{p}}_i^w = S\mathbf{p}_i^w \in \mathbb{R}^3$, where $S = I - 2\mathbf{n}\mathbf{n}^T \in \mathbb{R}^{3\times3}$ is the householder transformation matrix that describes a reflection about a plane and $\mathbf{n} = (0, 0, 1)^T$ (Hesch et al., 2009). The point feature $\mathbf{p}_i^w$ in the world reference frame is symmetric to its mirrored point in the world reference frame $\tilde{\mathbf{p}}_i^w$ with respect to the river surface. We define the X-Y plane of the world reference frame as the river surface, as shown in Figure 3.

The measurement of the reflection can be expressed in terms of a projection of the vector $\tilde{\mathbf{p}}_i^b \equiv (\tilde{x}_i^b, \tilde{y}_i^b, \tilde{z}_i^b)^T \in \mathbb{R}^3$, which is the position of the mirrored point with respect to the UAV body frame. The equality of the normalized coordinates $\tilde{\mathbf{h}}_i^b \equiv (\tilde{h}_{1,i}^b, \tilde{h}_{2,i}^b)^T = (\tilde{y}_i^b/\tilde{x}_i^b, \tilde{z}_i^b/\tilde{x}_i^b)^T = (y_{r,i}^b/x_{r,i}^b, z_{r,i}^b/x_{r,i}^b)^T \in \mathbb{R}^2$ holds, where $(\tilde{y}_i^b/\tilde{x}_i^b, \tilde{z}_i^b/\tilde{x}_i^b)^T$ is the normalized coordinates of the mirrored point $\tilde{\mathbf{p}}_i^b$, and $(y_{r,i}^b/x_{r,i}^b, z_{r,i}^b/x_{r,i}^b)^T$ is the normalized coordinates of the reflection $\mathbf{p}_{r,i}^b$. The position of the mirrored point with respect to the world reference frame is $\tilde{\mathbf{p}}_i^w = S\mathbf{p}_i^w = S[\mathbf{p}_b^w + R(\mathbf{q}_b^w)\mathbf{p}_i^b]$. The position of the mirrored point with respect to the UAV body frame is given by (Hesch et al., 2009)

$$\tilde{\mathbf{p}}_i^b = R^T(\mathbf{q}_b^w)\left[S\left(\mathbf{p}_b^w + R(\mathbf{q}_b^w)\mathbf{p}_i^b\right) - \mathbf{p}_b^w\right]. \tag{6}$$

Figure 3 shows an illustration of the projection of the vector to a point feature $\mathbf{p}_i^b$ and the vector to its reflection $\mathbf{p}_{r,i}^b$ from the UAV body frame. We include the two-view

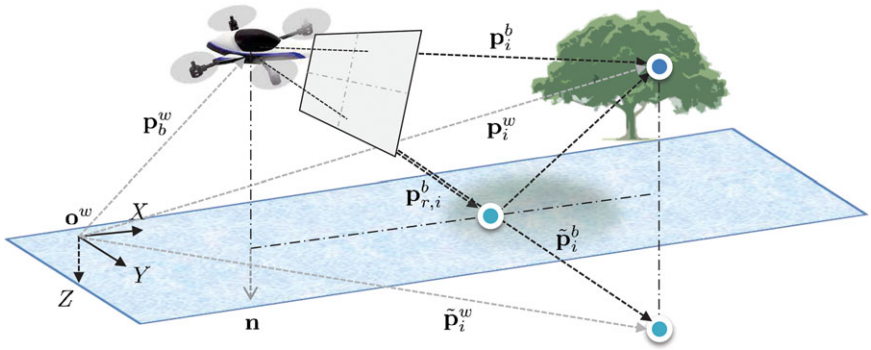


Figure 3. Illustration of the vision measurements of a real object and its reflection. The vector $\mathbf{p}_i^w$ of a point feature from a real object in the world frame is symmetric to the vector $\tilde{\mathbf{p}}_i^w$ of its mirrored point with respect to the river surface (X-Y plane). The measurement of the reflection is a camera projection of the vector $\tilde{\mathbf{p}}_i^b$.

Algorithm 1 Reflection matching in riverine environments

Input: image data T_0 and camera orientation $\mathbf{q}_c^w$
Output: normalized pixel coordinates of real objects and their reflections
while *image data are present* **do**
  Select good features to track with the Shi and Tomasi method.
  Select an image patch around each feature and invert the patch vertically.
  Slide each patch T on the source image T_0 and compute the NCC given by Eq. (8).
  Match each patch T in the source image T_0 based on the NCC.
  Compute the reference slope θ_0 and the matching slope θ_i given by Eq. (9).
  if $|\theta_0 - \theta_i| > \eta$ **then**
    Reject the matching result.
  end if
  Acquire reflection measurements from the matching results.
  Track the feature and its reflection with the KLT algorithm.
end while

measurements ($\mathbf{h}_i^b$, $\mathbf{h}_i^{bi}$) and the reflection measurement $\tilde{\mathbf{h}}_i^b$ in the measurement model and enhance the observability of our estimation system (see Section 5 for the observability analysis).

3.3.3. Vision-data Processing and Reflection Matching

We implemented an algorithm that matches the points from the objects around the river to the points from the reflections in the image by using the normalized correlation coefficients (NCC) (Scherer et al., 2012). The algorithm discards false matches by using the UAV's attitude information. The pseudocode of the reflection matching algorithm is shown in Algorithm 1.

The algorithm first selects good features to track by using Shi and Tomasi's method (Shi & Tomasi, 1994), which computes the minimum eigenvalue of the autocorrelation matrix of the Hessian over a small window in the intensity image. Algorithm 1 extracts an image patch around each point feature and inverts the image patch vertically to take account for the reflection symmetry. We compute the correlation coefficient of the two intensity image patches by

$$M(\tilde{x}^m, \tilde{y}^m) = \sum_{x'=1}^{50}\sum_{y'=1}^{50}[T(x', y') - \bar{T}][T_0(\tilde{x}^m + x', \tilde{y}^m + y') - \bar{T}_0], \tag{7}$$

where T_0 is the original intensity image, T is a 50×50 pixels patch from the image T_0, which is vertically inverted, and $\bar{T}$ and $\bar{T}_0$ are the means of T and T_0. The coordinates of the pixels that form the patch T are (x', y'), and the coordinates of the first upper left pixel in the image T_0 are $(\tilde{x}^m, \tilde{y}^m)$. The results are normalized to reduce the effects of lighting differences. The NCC is given by (Haralick & Shapiro, 1993)

$$N(\tilde{x}^m, \tilde{y}^m) = M(\tilde{x}^m, \tilde{y}^m)\left(\sum_{x'=1}^{50}\sum_{y'=1}^{50}[T(x', y') - \bar{T}]^2 \cdot \sum_{x'=1}^{50}\sum_{y'=1}^{50}\left[T_0(\tilde{x}^m + x', \tilde{y}^m + y') - \bar{T}_0\right]^2\right)^{-1/2}. \tag{8}$$

Algorithm 1 then finds the corresponding location in the source image that has the highest NCC. The methods proposed in Zhang, Guo, & Cao (2010) and Zhong, Liu, Liu, & Li (2013) could also be considered as cues for reflection detection.

To reject incorrect matches, we define a reference slope θ_0, which is computed based on the camera orientation, across the source image. Algorithm 1 computes the reflection matching slope θ_i with the pixel coordinates of the object and its reflection. If the difference between the reference slope and the matching slope exceeds a threshold η, the algorithm rejects the matched reflection. The reference slope θ_0 and the matching slope θ_i are given by

$$\theta_0 = \operatorname{atan2}\left(y_{s,0}^c - \tilde{y}_{s,0}^c, x_{s,0}^c - \tilde{x}_{s,0}^c\right),$$
$$\theta_i = \operatorname{atan2}\left(y_i^m - \tilde{y}_i^m, x_i^m - \tilde{x}_i^m\right), \tag{9}$$

where (x_i^m, y_i^m) are the pixel coordinates of the ith feature, and $(\tilde{x}_i^m, \tilde{y}_i^m)$ are the coordinates of the candidate for the reflection of the feature. We compute the unit vector $\mathbf{p}_{s,0}^c$ with a unit sphere projection of an arbitrary point in the image. The reflection corresponding to the unit vector $\mathbf{p}_{s,0}^c$ is given by $\tilde{\mathbf{p}}_{s,0}^c = (\tilde{x}_{s,0}^c, \tilde{y}_{s,0}^c, \tilde{z}_{s,0}^c)^T = R^T(\mathbf{q}_c^b)R^T(\mathbf{q}_b^w)SR(\mathbf{q}_b^w)R(\mathbf{q}_c^b)\mathbf{p}_{s,0}^c \in \mathbb{R}^3$ in the camera frame.

Figure 4 shows an example of matching real objects and their reflections at Crystal Lake. The algorithm tracks the center of the inverted image patch and its matched image patch over the sequence of image data with the pyramid KLT tracking algorithm (Bradski & Kaehler, 2008). The KLT algorithm solves an optical flow equation by the least-squares criterion while assuming that the flow is locally constant. We assume that the UAV does not perform acrobatic maneuvers, so the pixel coordinates of real objects lie above their reflections in the image.

4. EKF ESTIMATOR

In this section, we formulate a discrete-time EKF (Reif, Gunther, Yaz, & Unbehauen, 1999) to estimate the location $\hat{\mathbf{p}}_b^w \in \mathbb{R}^3$ of the UAV in the world reference frame; the velocity $\hat{\mathbf{v}}^b \equiv (\hat{v}_1, \hat{v}_2, \hat{v}_3)^T \in \mathbb{R}^3$ of the UAV, accelerometer bias $\hat{\mathbf{b}}_a^b \in \mathbb{R}^3$, and each vector of the ith point feature $\hat{\mathbf{x}}_i^b \equiv (\hat{h}_{1,i}^b, \hat{h}_{2,i}^b \hat{\rho}_i^b)^T \in \mathbb{R}^3$ with respect to the UAV body frame, where the hat operator $\hat{(.)}$ indicates an estimated value.

4.1. Motion Propagation

Let us denote the reduced-order state estimate by $\boldsymbol{\mu} \equiv ((\hat{\mathbf{p}}_b^w)^T, (\hat{\mathbf{v}}^b)^T, (\hat{\mathbf{b}}_a^b)^T, (\hat{\mathbf{x}}_{1:n}^b)^T)^T \in \mathbb{R}^{9+3n}$, where $\hat{\mathbf{x}}_{1:n}^b$ denotes n features. We denote the predicted state estimate by $\boldsymbol{\mu}_k$ at time-step k and the corrected state estimate after the measurement update by $\boldsymbol{\mu}_k^+$ in discrete time. We denote the estimate covariance by $\Sigma_k \in \mathbb{R}^{(9+3n)\times(9+3n)}$ at time-step k. The state estimate of the UAV is propagated through the dynamic model based on Eqs. (1) and (4) as follows:

$$\boldsymbol{\mu}_k = \boldsymbol{f}\left(\boldsymbol{\mu}_{k-1}^+, \mathbf{q}_{b,k-1}^w, \boldsymbol{\omega}_{k-1}^b, \tilde{\mathbf{a}}_{k-1}^b\right), \tag{10}$$

where

$$\boldsymbol{f}\left(\boldsymbol{\mu}_{k-1}^+, \mathbf{q}_{b,k-1}^w, \boldsymbol{\omega}_{k-1}^b, \tilde{\mathbf{a}}_{k-1}^b\right) = \begin{pmatrix} \hat{\mathbf{p}}_{b,k-1}^{w+} + R\left(\mathbf{q}_{b,k-1}^w\right)\hat{\mathbf{v}}_{k-1}^{b+}\Delta t \\ \hat{\mathbf{v}}_{k-1}^{b+} + \left(-[\boldsymbol{\omega}_{k-1}^b]_\times \hat{\mathbf{v}}_{k-1}^{b+} + \tilde{\mathbf{a}}_{k-1}^b - \hat{\mathbf{b}}_{a,k-1}^{b+} + R^T(\mathbf{q}_{b,k-1}^w)\mathbf{g}^w\right)\Delta t \\ \hat{\mathbf{b}}_{a,k-1}^{b+} \\ \hat{\mathbf{x}}_{1,k-1}^{b+} + \mathbf{f}_1\left(\boldsymbol{\mu}_{k-1}^+, \boldsymbol{\omega}_{k-1}^b\right)\Delta t \\ \vdots \\ \hat{\mathbf{x}}_{n,k-1}^{b+} + \mathbf{f}_n\left(\boldsymbol{\mu}_{k-1}^+, \boldsymbol{\omega}_{k-1}^b\right)\Delta t \end{pmatrix}, \tag{11}$$

and

$$\mathbf{f}_i\left(\boldsymbol{\mu}_{k-1}^+, \boldsymbol{\omega}_{k-1}^b\right) = \begin{pmatrix} \left(-\hat{v}_{2,k-1}^+ + \hat{h}_{1,i,k-1}^{b+}\hat{v}_{1,k-1}^+\right)\hat{\rho}_{i,k-1}^{b+} + \hat{h}_{2,i,k-1}^{b+}\omega_{1,k-1} - \left(1 + \left(\hat{h}_{1,i,k-1}^{b+}\right)^2\right)\omega_{3,k-1} + \hat{h}_{1,i,k-1}^{b+}\hat{h}_{2,i,k-1}^{b+}\omega_{2,k-1} \\ \left(-\hat{v}_{3,k-1}^+ + \hat{h}_{2,i,k-1}^{b+}\hat{v}_{1,k-1}^+\right)\hat{\rho}_{i,k-1}^{b+} - \hat{h}_{1,i,k-1}^{b+}\omega_{1,k-1} + \left(1 + \left(\hat{h}_{2,i,k-1}^{b+}\right)^2\right)\omega_{2,k-1} - \hat{h}_{1,i,k-1}^{b+}\hat{h}_{2,i,k-1}^{b+}\omega_{3,k-1} \\ \left(-\omega_{3,k-1}\hat{h}_{1,i,k-1}^{b+} + \omega_{2,k-1}\hat{h}_{2,i,k-1}^{b+}\right)\hat{\rho}_{i,k-1}^{b+} + \hat{v}_{1,k-1}^+\left(\hat{\rho}_{i,k-1}^{b+}\right)^2 \end{pmatrix}. \tag{12}$$

Here, $\boldsymbol{\mu}_{k-1}^+$ is the state estimate from the previous time-step; $\mathbf{q}_{b,k-1}^w$ is the attitude quaternion of the UAV, and $\tilde{\mathbf{a}}_{k-1}^b$ and $\boldsymbol{\omega}_{k-1}^b$ are the acceleration and the bias free angular velocity measurements, which are provided by the magnetometer and the IMU at time-step $k-1$.

The covariance matrix is propagated through $\Sigma_k = F_{k-1}\Sigma_{k-1}^+ F_{k-1}^T + W_{k-1}$, where F_{k-1} is the Jacobian of the motion model $\boldsymbol{f}(\boldsymbol{\mu}_{k-1}^+, \mathbf{q}_{b,k-1}^w, \boldsymbol{\omega}_{k-1}^b, \tilde{\mathbf{a}}_{k-1}^b)$ in Eq. (11) evaluated at $\boldsymbol{\mu}_{k-1}^+$, and W_{k-1} represents the covariance of the process noise.

The prediction of the error angle vector $\delta\hat{\boldsymbol{\theta}}_b^w \in \mathbb{R}^3$ and the gyroscope bias error $\Delta\hat{\mathbf{b}}_g^b = \mathbf{b}_g^b - \hat{\mathbf{b}}_g^b \in \mathbb{R}^3$ (Crassidis, Landis Markley, & Cheng, 2007; Kelly & Sukhatme, 2011) can be included in Eq. (11) as

$$\begin{aligned} \delta\hat{\boldsymbol{\theta}}_{b,k}^w &= \delta\hat{\boldsymbol{\theta}}_{b,k-1}^{w+} - \left[\left(\left[\tilde{\boldsymbol{\omega}}_{k-1}^b\right]_\times - \left[\hat{\mathbf{b}}_{g,k-1}^{b+}\right]_\times\right)\delta\hat{\boldsymbol{\theta}}_{b,k-1}^{w+} \right. \\ &\quad \left. + \Delta\hat{\mathbf{b}}_{g,k-1}^{b+}\right]\Delta t, \\ \Delta\hat{\mathbf{b}}_{g,k}^b &= \Delta\hat{\mathbf{b}}_{g,k-1}^{b+}, \end{aligned} \tag{13}$$

where $\tilde{\boldsymbol{\omega}}_{k-1}^b$ is the gyroscope measurement that includes a bias $\mathbf{b}_g^b$, and $\hat{\mathbf{b}}_{g,k-1}^b$ is the estimate of the gyroscope bias. The predicted estimate of the attitude quaternion $\hat{\boldsymbol{q}}_{b,k}^w$ is given by

$$\hat{\boldsymbol{q}}_{b,k}^w = \hat{\boldsymbol{q}}_{b,k-1}^{w+} + \frac{1}{2}\left(\Omega(\tilde{\boldsymbol{\omega}}_{k-1}^b) - \Omega(\hat{\mathbf{b}}_{g,k-1}^{b+})\right)\hat{\boldsymbol{q}}_{b,k-1}^{w+}\Delta t. \tag{14}$$

The error angle vector $\delta\hat{\boldsymbol{\theta}}_b^w$ is a minimal representation derived with a small-angle approximation of the error quaternion, $\delta\hat{\mathbf{q}}_b^w = \mathbf{q}_b^w \otimes (\hat{\mathbf{q}}_b^w)^{-1}$ (Crassidis et al., 2007; Kelly & Sukhatme, 2011), where $\otimes$ denotes quaternion multiplication. For the case of including the attitude in the estimation state vector, the estimate of the UAV's attitude $\hat{\boldsymbol{q}}_{b,k-1}^w$ and the angular velocity $\hat{\boldsymbol{\omega}}_{k-1}^b \equiv (\hat{\omega}_{1,k-1}, \hat{\omega}_{2,k-1}, \hat{\omega}_{3,k-1})^T = \tilde{\boldsymbol{\omega}}_{k-1}^b - \hat{\mathbf{b}}_{g,k-1}^b$ should replace $\boldsymbol{q}_{b,k-1}^w$ and $\boldsymbol{\omega}_{k-1}^b$ in Eqs. (11) and (12), respectively.

It is possible to include the gyroscope bias error $\Delta\hat{\mathbf{b}}_g^b$ and the error angle vector $\delta\hat{\boldsymbol{\theta}}_b^w$ in the estimation state to estimate the attitude of the UAV while preserving the normalization constraint of the quaternion if the UAV's attitude information is not provided. Reduced-order estimators are often used (Dani et al., 2012; Dixon, Fang, Dawson, & Flynn, 2003; Jankovic & Ghosh, 1995) to solve an estimation problem concisely with directly measurable variables when it is not necessary to filter the measurements. We simplify the process and alleviate the nonlinearity of the model by

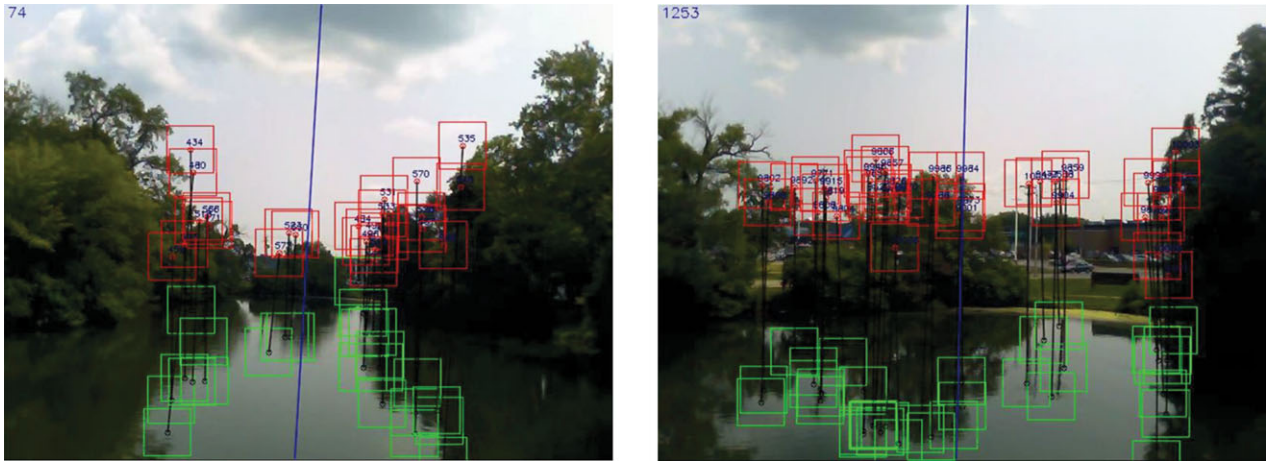


Figure 4. The results of reflection feature detection with the reflection matching Algorithm 1. The real objects (red boxes), the corresponding reflections (green boxes), the matching slope θ_i (black lines), and the reference slope θ_0 (blue line in the middle of the image) are shown.

acquiring the estimated attitude $\mathbf{q}_b^w$ of the UAV and the bias-compensated angular velocity from an IMU and a magnetometer and excluding the corresponding state variables from the estimation state vector.

4.2. Measurement Update

The predicted measurements of our estimation system that consist of the current view $\mathbf{h}_{1:n}^b$ of features, the observation $\mathbf{h}_{1:n}^b$ of the features from the initial feature detection positions (which we denote as the initial view $\mathbf{h}_{1:n}^{bi}$), the reflection view $\tilde{\mathbf{h}}_{1:n}^b$ of n point features, and the altitude $-\hat{z}_{b,k}^w$ are given by

$$\mathbf{h}(\boldsymbol{\mu}_k, \mathbf{q}_{b,k}^w, \mathbf{p}_{bi}^w, \mathbf{q}_{bi}^w) = \begin{pmatrix} \mathbf{h}_{1:n}^b(\boldsymbol{\mu}_k) \\ \mathbf{h}_{1:n}^{bi}(\boldsymbol{\mu}_k, \mathbf{q}_{b,k}^w, \mathbf{p}_{bi}^w, \mathbf{q}_{bi}^w) \\ \tilde{\mathbf{h}}_{1:n}^b(\boldsymbol{\mu}_k, \mathbf{q}_{b,k}^w) \\ -\hat{z}_{b,k}^w \end{pmatrix}, \quad (15)$$

where the altitude $-\hat{z}_{b,k}^w$ is measured by an altimeter. The current view of the ith point feature is given by

$$\mathbf{h}_i^b(\boldsymbol{\mu}_k) = \begin{pmatrix} \hat{h}_{1,i,k}^b & \hat{h}_{2,i,k}^b \end{pmatrix}^T. \quad (16)$$

We transform the measurements $\mathbf{h}_i^c$ in the camera coordinate frame to $\mathbf{h}_i^b$ in the UAV body frame as we described in Section 3.3.1.

The initial view of the ith point feature is given by

$$\mathbf{h}_i^{bi}(\boldsymbol{\mu}_k, \mathbf{q}_{b,k}^w, \mathbf{p}_{bi}^w, \mathbf{q}_{bi}^w) = \begin{pmatrix} \hat{y}_{i,k}^{bi}/\hat{x}_{i,k}^{bi} & \hat{z}_{i,k}^{bi}/\hat{x}_{i,k}^{bi} \end{pmatrix}^T, \quad (17)$$

where $\hat{\mathbf{p}}_{i,k}^{bi} = (\hat{x}_{i,k}^{bi}, \hat{y}_{i,k}^{bi}, \hat{z}_{i,k}^{bi})^T = R^T(\mathbf{q}_{bi}^w)(\hat{\mathbf{x}}_{b,k}^w - \hat{\mathbf{x}}_{bi}^w) + R^T(\mathbf{q}_{bi}^w) R(\mathbf{q}_{b,k}^w)\hat{\mathbf{p}}_{i,k}^b$ [which is given by Eq. (5)] is the estimated position of the feature with respect to $(\mathbf{p}_{bi}^w, \mathbf{q}_{bi}^w)$. The estimated location $\mathbf{p}_{bi}^w = \hat{\mathbf{p}}_{b,k_i}^w$ and the filtered attitude $\mathbf{q}_{bi}^w$ of the UAV are stored at time-step k_i when the ith feature is first measured. On the other hand, $\hat{\mathbf{p}}_{i,k}^b = (1/\hat{\rho}_{i,k}^b, \hat{h}_{1,i,k}^b/\hat{\rho}_{i,k}^b, \hat{h}_{2,i,k}^b/\hat{\rho}_{i,k}^b)^T$ is the estimated position of the feature with respect to the UAV body frame at the current time-step k.

The current view $\tilde{\mathbf{h}}_{1:n}^b(\boldsymbol{\mu}_k, \mathbf{q}_{b,k}^w)$ of a reflection of the ith feature is given by

$$\tilde{\mathbf{h}}_i^b(\boldsymbol{\mu}_k, \mathbf{q}_{b,k}^w) = \begin{pmatrix} \hat{\tilde{x}}_{i,k}^b/\hat{\tilde{z}}_{i,k}^b & \hat{\tilde{y}}_{i,k}^b/\hat{\tilde{z}}_{i,k}^b \end{pmatrix}^T, \quad (18)$$

where $\hat{\tilde{\mathbf{p}}}_{i,k}^b \equiv (\hat{\tilde{x}}_i^b, \hat{\tilde{y}}_i^b, \hat{\tilde{z}}_i^b)^T = R^T(\mathbf{q}_{b,k}^w)\{S[\hat{\mathbf{p}}_{b,k}^w + R(\mathbf{q}_{b,k}^w)\hat{\mathbf{p}}_{i,k}^b] - \hat{\mathbf{p}}_{b,k}^w\}$ [which is given by Eq. (6)] is the estimated position of the mirrored point of the ith feature with respect to the current UAV body frame (see Figure 3).

The state estimate and the estimate covariance are updated with vision measurements by

$$\begin{aligned} \boldsymbol{\mu}_k^+ &= \boldsymbol{\mu}_k + K_k\left(\mathbf{z}_k - \mathbf{h}\left(\boldsymbol{\mu}_k, \mathbf{q}_{b,k}^w, \mathbf{p}_{bi}^w, \mathbf{q}_{bi}^w\right)\right), \\ \Sigma_k^+ &= \Sigma_k - K_k H_k \Sigma_k, \end{aligned} \quad (19)$$

where the Kalman gain is given by $K_k = \Sigma_k H_k^T (H_k \Sigma_k H_k^T + V_k)^{-1}$. Here, $\mathbf{z}_k$ is the measurement vector at time-step k, H_k is the Jacobian of the measurement model $\mathbf{h}(\boldsymbol{\mu}_k, \mathbf{q}_{b,k}^w, \mathbf{p}_{bi}^w, \mathbf{q}_{bi}^w)$ in Eq. (15) evaluated at $\boldsymbol{\mu}_k$, and V_k is the covariance of the measurement noise.

If we include the error angle vector $\delta\hat{\boldsymbol{\theta}}_b^w$ and the gyroscope bias $\Delta\hat{\mathbf{b}}_g^b$ in Eq. (11), the estimate of the UAV's attitude and the gyroscope bias can be updated by

$$\begin{aligned} \hat{\boldsymbol{q}}_{b,k}^{w+} &= \hat{\boldsymbol{q}}_{b,k}^w + \frac{1}{2}\Omega\left(\delta\hat{\boldsymbol{\theta}}_{b,k}^{w+}\right)\hat{\boldsymbol{q}}_{b,k}^w, \\ \hat{\mathbf{b}}_{g,k}^{b+} &= \hat{\mathbf{b}}_{g,k}^b + \Delta\hat{\mathbf{b}}_{g,k}^{b+}, \end{aligned} \quad (20)$$

where $(\hat{\boldsymbol{q}}_{b,k}^w, \hat{\mathbf{b}}_{g,k}^b)$ are the predicted estimates of the attitude and the gyroscope bias, and $(\delta\hat{\boldsymbol{\theta}}_{b,k}^{w+}, \Delta\hat{\mathbf{b}}_{g,k}^{b+})$ are the updated error angle vector and the gyroscope bias error.

4.3. World Reference Frame Representation

The motion model in Eq. (11) includes the dynamics of each feature in the UAV body frame. By using this robot-centric approach, we are able to estimate the position of each point feature with respect to the UAV body frame and enhance the observability of the estimation system, as shall be shown in Section 5. After the measurement update of each EKF cycle, we express the estimates of the point features with respect to the world reference frame as follows:

$$\hat{\mathbf{p}}^w_{i,k} = \hat{\mathbf{p}}^w_{b,k} + R(\mathbf{q}^w_{b,k})\hat{\mathbf{p}}^b_{i,k}, \tag{21}$$

where $\hat{\mathbf{p}}^w_{b,k}$ is the estimated location of the UAV, $\hat{\mathbf{p}}^b_{i,k}$ is the estimated position of the ith feature, and $\mathbf{q}^w_{b,k}$ is the attitude of the UAV. By representing the estimated $\hat{\mathbf{p}}^w_{i,k}$ in the world reference frame, we are able to generate a map in a global frame instead of showing the time-varying trajectories of point features in the UAV body frame. We estimate the vector $\hat{\mathbf{x}}^b_{i,k}$ of the feature that is being measured and discard the features that go out of sight to maintain the size of the state vector, thereby reducing the computational load.

5. OBSERVABILITY ANALYSIS

The observability problem of VINS (Hesch et al., 2014; Kelly & Sukhatme, 2011; Martinelli, 2014; Weiss et al., 2013) and SLAM (Bryson & Sukkarieh, 2008; Huang et al., 2010; Lee et al., 2006) have been studied in the literature. It has been shown that VINS and SLAM require *a priori* knowledge of the position of a set of features in the map in order to make the system observable. In Section 3, we presented an estimation system for world-centric localization and robot-centric mapping, which includes water reflections and feature point locations referenced to initial-view robot positions. In this section, we analyze the observability property of the estimation system under various conditions.

5.1. Methods of Observability Analysis

First, we state the definition of the observability. A system is observable if there exists $t_0 \leq t_f$ such that the state $\mathbf{x}_0$ of the system at time t_0 can be determined from the knowledge of the system's output over the interval $[t_0, t_f]$ (Franklin, Powell, & Emami-Naeini, 2001). Here, $\mathbf{x}_0$ is the state vector at time t_0. Observability implies that the current state of the system can be determined from the present and past output measurements and input commands.

In Hermann & Krener (1977), the observability of nonlinear systems is categorized to be observable, locally observable, weakly observable, and locally weakly observable. Local weak observability is defined in Hermann & Krener (1977) as follows:

Definition 1. (***Local weak observability***): *A system is locally weakly observable at* $\mathbf{x}_0$ *if there exists an open neighborhood* U *of* $\mathbf{x}_0$ *such that for every open neighborhood* V *of* $\mathbf{x}_0$ *contained in* U, $\mathbf{x}_0$ *is distinguishable from any other point in* V.

The local weak observability from Definition 1 implies that we can instantaneously distinguish (Hermann & Krener, 1977) each state from its neighbors. In Huang et al. (2010), it is stated that if the nonlinear system is not locally weakly observable, the linearized system can gain spurious information along the unobservable direction and degrade the performance. Therefore, we first check the local weak observability of our estimation system and verify the role of the measurements included in Eq. (15). The local weak observability can be analyzed with the rank of the nonlinear observability matrix $\mathcal{O}_{\text{NL}}$. We formulate the nonlinear observability matrix $\mathcal{O}_{\text{NL}}$ by recursively computing the Lie derivatives of the measurement function $\mathbf{h}$ in Eq. (15) with respect to the affine form of the dynamic function $\mathbf{f} = \mathbf{f}_0 + \mathbf{f}_1\mathbf{a}^b + \mathbf{f}_2\boldsymbol{\omega}^b$ presented in Eqs. (11) and (12) as shown in Kelly, & Sukhatme (2011). The nonlinear observability matrix is given by

$$\mathcal{O}_{\text{NL}} = \nabla\left((L^0\mathbf{h})^T \left(L^1_{\mathbf{f}_0}\mathbf{h}\right)^T \cdots \left(L^\gamma_{\mathbf{f}_0\mathbf{f}_1\mathbf{f}_2}\mathbf{h}\right)^T \cdots\right)^T, \tag{22}$$

where ∇ is the gradient operator with respect to our state, $L^0\mathbf{h} = \mathbf{h}$, $L^\gamma_{\mathbf{f}_0}\mathbf{h} = \nabla L^{\gamma-1}\mathbf{h}\cdot\mathbf{f}_0$ for the γth-order Lie derivative, and $L^\gamma_{\mathbf{f}_0\mathbf{f}_1\mathbf{f}_2}\mathbf{h} = \nabla L^{\gamma-1}_{\mathbf{f}_1\mathbf{f}_2}\mathbf{h}\cdot\mathbf{f}_0$ for mixed Lie derivatives.

5.2. Observability Analysis of the System

We analyze the observability of our estimation system and show the advantage of employing the measurement model given by Eq. (15) along with the motion model of our reduced-order system given by Eqs. (11) and (12). We consider situations in which we do not acquire the reflection measurement, the initial view measurement, which is the observation of a feature from the initial feature detection location, and the altitude measurement to show the necessity of each type of measurements.

5.2.1. Observability with Current View, Initial View, Reflection, and Altitude Measurements

The nonlinear observability matrix $\mathcal{O}_{\text{NL}}$ for our estimation system given by Eqs. (11) and (15) satisfies the observability rank condition. The linear observability matrix for our estimation system also satisfies the rank condition. Therefore, the nonlinear system is locally weakly observable, and the linearized model computed for the EKF estimator is completely observable. The reflection measurements $\tilde{\mathbf{h}}^b_{1:n}$ allow the observability results to hold even if a single feature is measured without any *a priori* knowledge of the feature's position and the UAV is stationary without any motion parallax provided for the feature. The multiple measurements in the model given by Eq. (15) provide sufficient constraints with the information from the motion model given by Eqs. (11) and (12).

5.2.2. Observability with Only Current View Measurements

We consider the case in which the altitude $-z_b^w$ measurement is not available, and the reflection view $\tilde{\mathbf{h}}_i^b$ and the initial view $\mathbf{h}_i^{bi}$ measurements are not used in order to show the role of these measurements. If we only include the current view $\mathbf{h}_i^b$ of a single feature to the measurement function and omit the rest, the null space of the nonlinear observability matrix can be found as

$$\text{span}\begin{pmatrix} I_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times2} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & -\frac{(\mathbf{v}^b)^T}{\rho_i^b} & -\frac{(\mathbf{b}_a^b)^T + \mathbf{g}^T R(\mathbf{q}_b^w)}{\rho_i^b} & \mathbf{0}_{1\times2} & 1 \end{pmatrix}^T, \tag{23}$$

where the state vector of the reduced-order system is composed of $(\mathbf{p}_b^w, \mathbf{v}^b, \mathbf{b}_a^b, \mathbf{x}_i^b)$. Note that the same state vector is used in this section except for Eq. (24). We treat the attitude quaternion $\mathbf{q}_b^w$ of the UAV as a known vector since we acquire the estimate of the UAV's attitude from the IMU and the magnetometer. The null space shows the unobservable modes. The location of the UAV is unobservable. Also, the velocity of the UAV, the bias of the accelerometer, and the inverse-depth of the feature constitute the unobservable modes. The normalized coordinates $(h_{1,i}^b, h_{2,i}^b)$ of the feature, which are directly measured, are observable.

It is known that the monocular-vision SLAM with IMU measurements is also unobservable when the position $\mathbf{p}_i^w = (x_i^w, y_i^w, z_i^w)^T \in \mathbb{R}^3$ of the feature is estimated with respect to the world reference frame. The null space of the nonlinear observability matrix for the visual-inertial SLAM with a single feature prescribed in the world frame is given as

$$\text{span}\begin{pmatrix} I_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & I_{3\times3} \\ -\mathbf{g}^T[\mathbf{p}_b^w]_\times^T & -\mathbf{g}^T[\mathbf{v}_b^w]_\times^T & \mathbf{0}_{3\times3} & -\mathbf{g}^T[\mathbf{p}_i^w]_\times^T \end{pmatrix}^T, \tag{24}$$

where the state vector of the reduced-order system is composed of $(\mathbf{p}_b^w, \mathbf{v}^b, \mathbf{b}_a^b, \mathbf{p}_i^w)$.

From Eq. (24), we can see that the relative 3D location of the robot and the feature location are unobservable. The location and the velocity of the robot and the position of the feature also form the unobservable modes for a world-centric 6-DOF localization and 3D mapping system that uses a monocular camera and inertial sensors. Furthermore, the attitude of the UAV along the gravity vector, i.e., yaw, is unobservable for visual-inertial SLAM if we do not acquire the attitude information from the IMU and the magnetometer (Hesch et al., 2014).

5.2.3. Observability with Partial Measurements

If we measure the current view $\mathbf{h}_i^b$ and the initial view $\mathbf{h}_i^{bi}$ of a single feature but not the reflection $\tilde{\mathbf{h}}_i^b$ and the altitude $-z_b^w$ after the initialization, the null space of the nonlinear observability matrix can be found as

$$\text{span}\begin{pmatrix} \frac{(\mathbf{p}_{bi}^w - \mathbf{p}_b^w)^T}{\rho_i^b} & -\frac{(\mathbf{v}^b)^T}{\rho_i^b} & -\frac{(\mathbf{b}^b + \mathbf{g})^T}{\rho_i^b} & \mathbf{0}_{1\times2} & 1 \end{pmatrix}^T. \tag{25}$$

For this case, we fixed the attitude of the UAV with $R(\mathbf{q}_b^w) = I_{3\times3}$ for simplicity. The null space shows that the translation of the UAV from the initial-view position, the velocity of the UAV, and the bias of the accelerometer constitute the unobservable modes.

If we employ all the measurements in Eq. (15) except for the initial view $\mathbf{h}_i^{bi}$ of a feature, the null space of the nonlinear observability matrix $\mathcal{O}_{\text{NL}}$ can be found as

$$\text{span}\begin{pmatrix} I_{2\times3} & \mathbf{0}_{2\times1} & \mathbf{0}_{2\times3} & \mathbf{0}_{2\times3} & \mathbf{0}_{2\times3} \end{pmatrix}^T. \tag{26}$$

The null space shows that the location of the UAV is unobservable without the initial view $\mathbf{h}_i^{bi}$. The initial view provides a reference to estimate the translation of the UAV. The results in Eqs. (23)–(26) show the necessity of employing the initial view $\mathbf{h}_{1:n}^{bi}$, the reflection view $\tilde{\mathbf{h}}_{1:n}^b$, and the altitude $-z_b^w$ measurements for achieving observability. In Section 5.3, we will quantify the degree of observability of our estimation system.

5.3. Degree of Observability

In Section 5.2, we used the observability matrix to analytically determine whether the system is observable and to find the unobservable modes. In this section, we quantify the observability by computing the degree of observability with the eigenvalues related to the observability Gramian as given by Krener & Ide (2009). The degree of observability indicates how accurate the estimation results are with noisy measurements. The discrete time-varying observability Gramian over a time-step interval $[k, k+m]$ is given by

$$\begin{aligned}\Psi \triangleq\ & H_k^T H_k + F_k^T H_{k+1}^T H_{k+1} F_k + F_k^T F_{k+1}^T H_{k+2}^T H_{k+2} F_{k+1} F_k \\ & + \cdots + F_k^T \cdots F_{k+m-1}^T H_{k+m}^T H_{k+m} F_{k+m-1} \cdots F_k,\end{aligned} \tag{27}$$

where F_k and H_k are the Jacobian matrices of the dynamic function given Eqs. (11) and (12) and the measurement function given by Eq. (15), respectively, at time-step k. The smallest eigenvalue of $\Psi^{1/2}$ shows the degree of observability (Krener & Ide, 2009).

We compare our localization and robot-centric mapping system, which exploits the reflection measurements

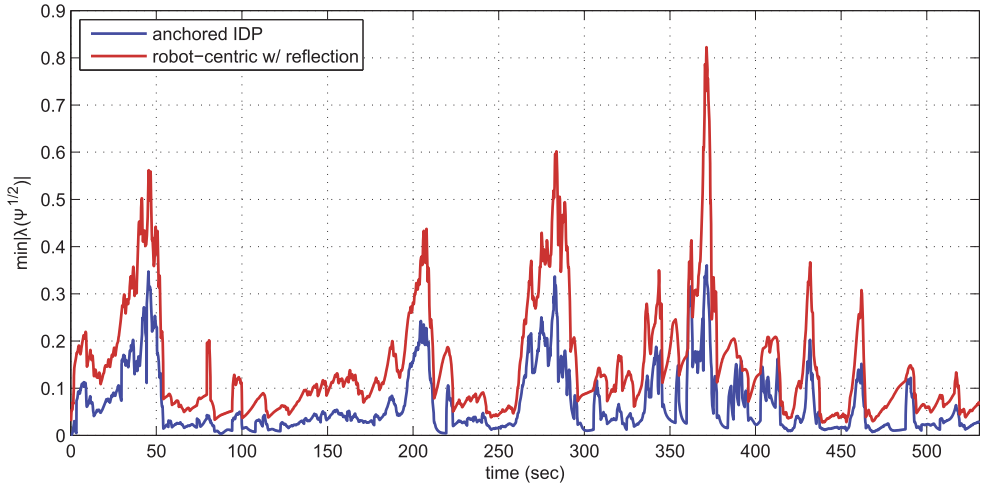


Figure 5. The degree of observability of our localization and robot-centric mapping system with reflection measurements and the anchored IDP SLAM system are shown, where the observability Gramian Ψ is defined in Eq. (27).

as presented in Sections 3 and 4, with a popular localization and mapping method that represents the features with respect to anchors in the world reference frame with the inverse depth parametrization (IDP) (Ceriani et al., 2011; Civera et al., 2008; Sola et al., 2012) by providing the anchored IDP SLAM method with the UAV's attitude and altitude information but without reflection measurements. The observability Gramian is computed with true state values for both of the systems in the simulation environment that shall be shown in Section 6. Figure 5 shows that our localization and robot-centric mapping system with reflection measurements has a larger degree of observability than the anchored IDP SLAM system, which does not use reflection measurements. The comparatively large degree of observability of our localization and robot-centric mapping system with reflection measurements shows that we can expect the estimation results from our estimation system to be more robust to measurement noise than the anchored IDP SLAM system. We shall demonstrate the superior performance of our localization and robot-centric mapping system with reflection measurements to the anchored IDP SLAM system in Section 6 with numerical simulation results of the localization and mapping.

6. NUMERICAL SIMULATIONS

In this section, we present results of numerical simulation and analyze the performance of our riverine localization and mapping algorithm. In Section 7, we will present experimental results using real-world data. Here, we simulate a riverine environment with a river image (Aroda, 2013) as shown in Figure 6. A trajectory along the river is defined by a sequence of way points and a potential field-style algorithm to generate the acceleration and angular velocity commands and execute a smooth 3D trajectory with roll, pitch, and yaw motions. Gaussian white noise of standard deviation $\sigma = 0.01$ is added to the acceleration and angular velocity commands as a disturbance. The UAV travels 418 m along the river for 530 s and extracts 330 point features from the trees around the river. The features are evenly distributed along the river 5 m apart from each other along the latitude and longitude directions. The heights of the features are distributed with a uniform distribution on the interval 0–30 m. The features that are between 5 and 20 m away from the camera that has a 90° field of view are considered as visible features. We allow the UAV to always measure four features at each step, where two of the features have reflections.

Gaussian white noise of standard deviation $\sigma = 0.01$ and 0.001 is added to the acceleration and angular velocity readings and the attitude and altitude measurements, respectively. The noise in camera pixel measurements is simulated as Gaussian white noise of $\sigma = 1$ considering a focal length of $\lambda = 770$ pixels. The location estimate of the UAV is initialized as $\hat{\mathbf{p}}^w_{b,0} = (0, 0, z^w_{b,0})^T$, where $-z^w_{b,0} \in \mathbb{R}^+$ is the initial altitude of the UAV over the river. The velocity estimate of the UAV and the accelerometer bias estimate are initialized as $\hat{\mathbf{v}}^b_0 = \mathbf{0}$ and $\hat{\mathbf{b}}^b_0 = \mathbf{0}$, respectively. The ith point feature for Eq. (12) is initialized as $\hat{\mathbf{x}}^b_{i,0} = (h^b_{1,i,0}, h^b_{2,i,0}, 0.1)^T$, where $h^b_{1,i,0}$ and $h^b_{2,i,0}$ are the initial normalized coordinates of the feature.

Figure 6 shows the localization and mapping results from our localization and robot-centric mapping system with reflection measurements on the simulated environment. The simulation results show the time-history of the location estimate of the UAV converging to the true trajectory of the UAV, and the estimated positions of the features converging to their true positions. Figure 7 also compares the estimates of the state variables and their true values.

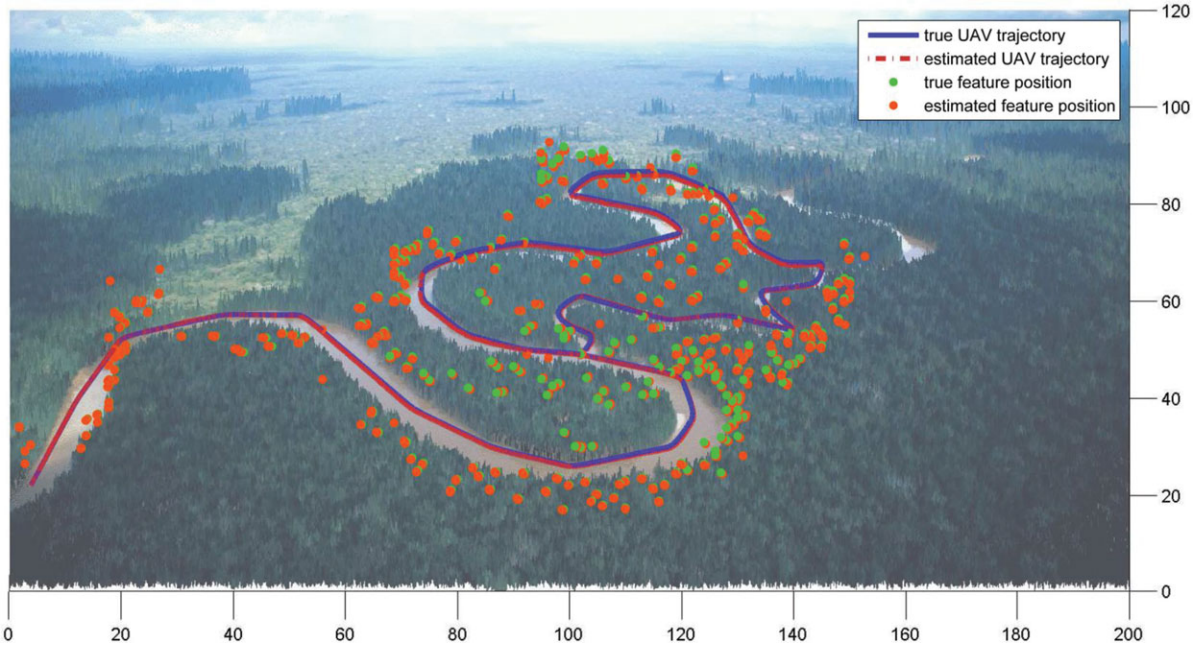


Figure 6. Results of the localization and mapping in a simulated riverine environment. The solid blue curve shows the trajectory of the UAV, and the green dots are the 3D point features extracted from the trees. The dashed red curve is the time-history of the UAV's location estimate, and the orange dots are the estimated locations of the features.

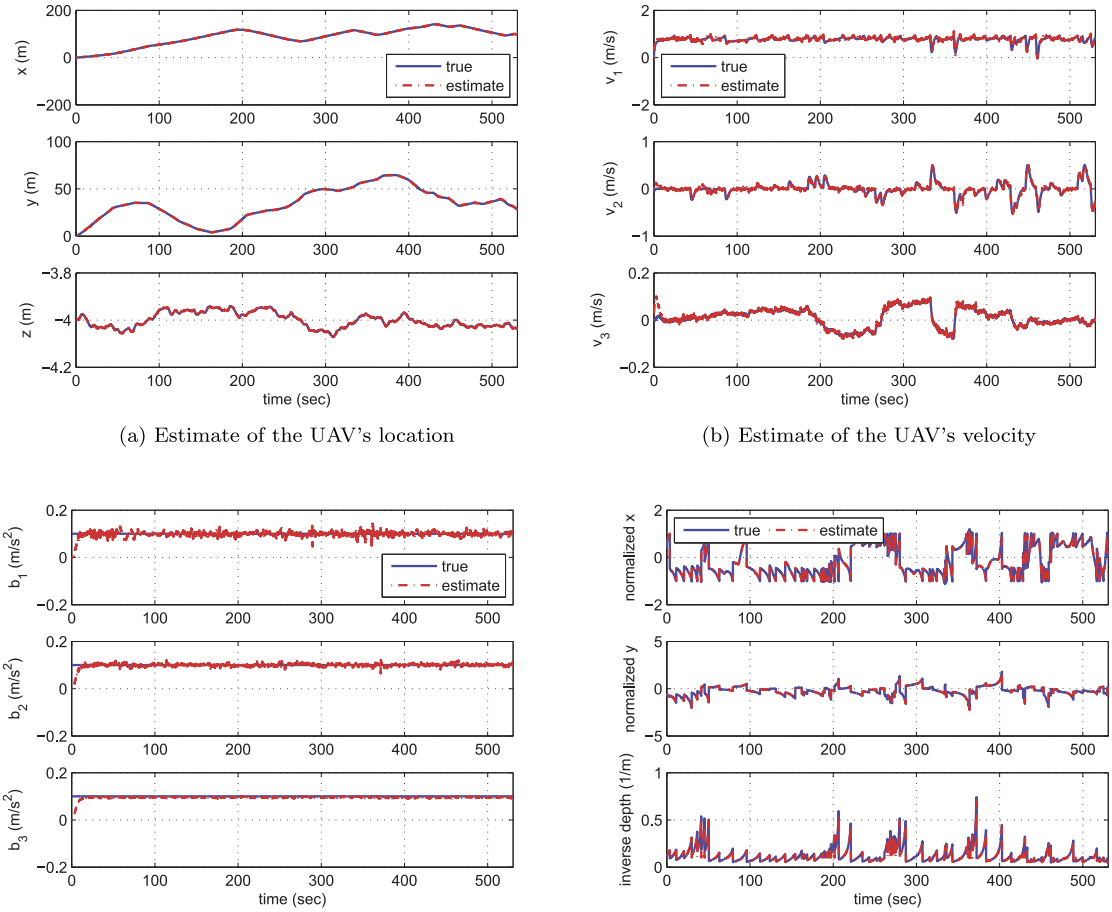
The estimation state includes the location and the velocity of the UAV, the bias of the accelerometer, and the normalized coordinates and the inverse-depth ($\hat{\rho}^b_{i,k} = 1/\hat{x}^b_{i,k}$) of the features. The results show that the estimates converge to their true values. The estimates of the features, which are shown in Figure 7(d), are in the UAV body frame as described in Section 3.2.2. We represent the robot-centric results in the world reference frame as described in Section 4.3 and generate mapping results that are shown in Figure 6.

Figure 8 shows the error between the state estimate and the true state values of the UAV and the features. The errors converge close to zero, and the 3σ standard deviation are bounded because our measurements, which consist of the current view, the initial view, and the reflection view of each feature, along with the altitude of the UAV, provide sufficient information for the estimation. The spikes that appear in the error are due to the impulses in the acceleration, which are generated from the algorithm we used to define the trajectory of the UAV. Figure 8(d) shows the error norm of the location and the velocity of the UAV and the error norm of the inverse-depth of all the features. Our localization and robot-centric mapping system with reflection measurements has an average error norm of 0.3113 m for the location of the UAV, 0.0312 m/s for the velocity of the UAV, and 0.0029 (1/m) for the inverse depth of the features. The anchored IDP SLAM system has an average error norm of 12.9103 m for the location of the UAV, 0.1613 m/s for the velocity of the UAV, and 0.0419 (1/m) for the inverse depth of the features. The superior performance of our localization and robot-centric mapping system with reflection measurements compared to that of the anchored IDP SLAM system that does not have reflection measurements is related to the relatively large degree of observability, which is shown in Section 5.3.

7. EXPERIMENTAL RESULTS AND DISCUSSION

To demonstrate the effectiveness of our algorithm in a real environment, we conducted experiments at Crystal Lake Park in Urbana, Illinois (see Figure 9). To evaluate our method, we present five sets of results.

- We present results obtained using GPS and IMU data to serve as ground truth.
- To compare our method against existing methods, we present results obtained using an anchored IDP method (Ceriani et al., 2011; Civera et al., 2008; Sola et al., 2012).
- To demonstrate the effectiveness of our method, we present results obtained using reflection measurements, and that incorporate loop closure.
- To demonstrate the relative performance of our approach for localization and mapping, we present results for our method that do not exploit loop closure.
- To illustrate that a short sequences of badly estimated poses can cause the pose estimates to diverge, we show



(a) Estimate of the UAV's location

(b) Estimate of the UAV's velocity

(c) Estimate of the accelerometer bias

(d) Estimates of the features

Figure 7. The location estimate of the UAV in the world reference frame, the velocity estimate of the UAV with respect to the UAV body frame, and the accelerometer bias estimate are shown. The estimates of the point features with respect to the UAV body frame are also shown.

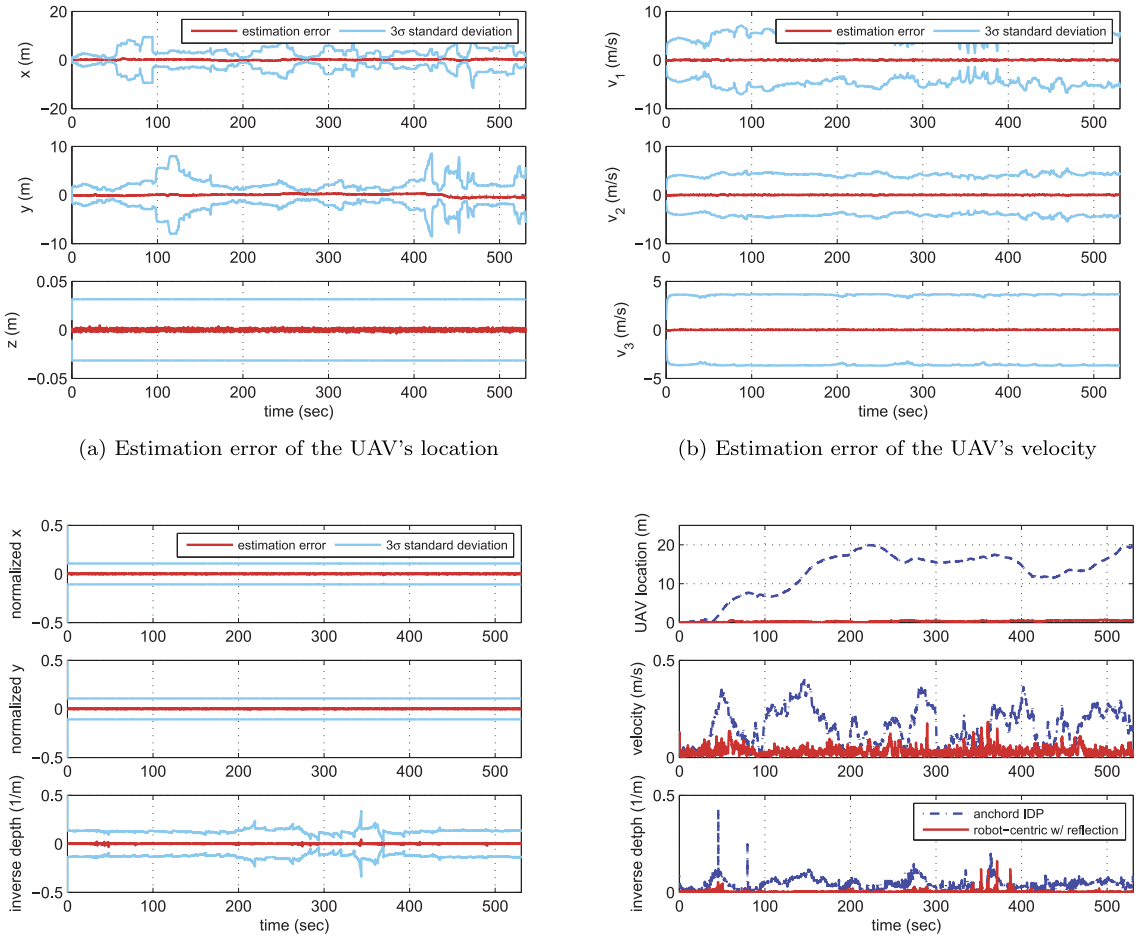
the results of our method obtained when GPS data are provided for those sequences.

Below, we describe our experimental methodology, present the results of our experiments, and discuss factors that influence the performance.

7.1. Methodology

For all experiments, we flew our quadcopter UAV (described in Section 3.1) at Crystal Lake in Urbana, Illinois using the altitude hold mode of the onboard automatic flight control system that accepts a radio control pilot input for heading control. We performed a calibration of all parameters of the sensing system, including the camera intrinsic parameters [through camera calibration Bouguet (2008)] and the orientation between the IMU and the camera [with IMU-camera calibration (Lobo & Dias, 2007)]. We removed the initial bias in the accelerometer by performing static calibration of the IMU at the beginning, and the IMU provided bias-compensated angular velocity. We used the Ublox Lea-6H GPS module that has an accuracy of 2.5 m, and we used the position results that are filtered with the inertial navigation system (INS) for the ground truth.

Figure 10 shows a characteristic set of images taken from the experimental data acquired at Crystal Lake with our quadcopter. We detected multiple point features from the image data automatically with Shi and Tomasi's method (Shi & Tomasi, 1994) and found their reflections with Algorithm 1, which we presented in Section 3.3.3. The algorithm tracked the features with the pyramid KLT method (Bradski & Kaehler, 2008) and sorted outliers with random sample consensus (RANSAC) (Bradski & Kaehler, 2008) on consecutive images. Algorithm 1 searched for a new pair of reflections per frame per core at 10 Hz. The rest of the estimation algorithm was capable to run at 100 Hz on a



(a) Estimation error of the UAV's location

(b) Estimation error of the UAV's velocity

(c) Estimation errors of the features

(d) Estimation error norm

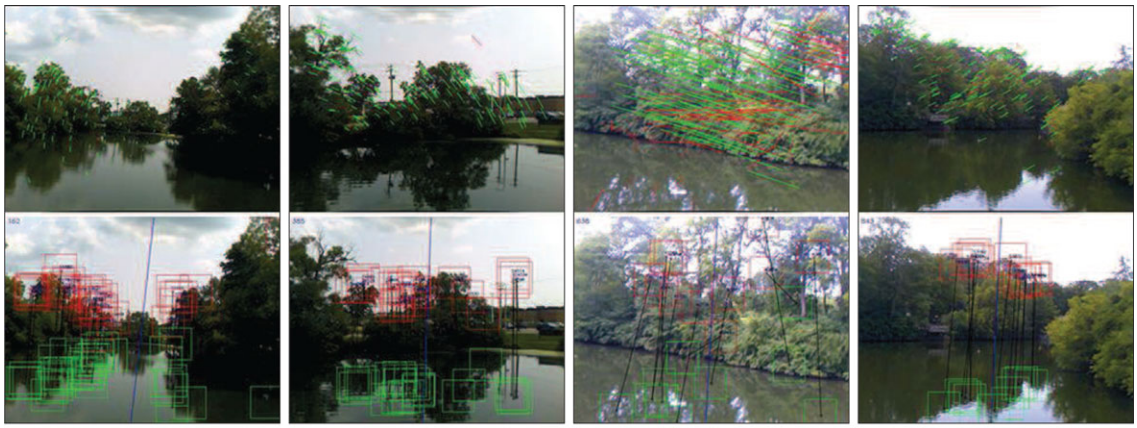
Figure 8. The estimation errors and the 3σ standard deviation estimates of the location and the velocity of the UAV and the features are shown. The error norms of the location and the velocity of the UAV and the inverse-depth of all the features are also shown.

quad-core computer with the features shown in Figures 10 and 11. We simplified the estimation process by estimating the vector $\hat{\mathbf{x}}^b_{i,k}$ of the feature that was being measured and by discarding the features that left the camera field of view in order to keep the size of the state vector small and to reduce the computational load. When an old feature was removed, a new feature was initialized in its place in the estimation state vector. We used a fixed number of features in the estimation for each frame (40 considering the process speed). Preference was given to features with matched reflections, and when there were not sufficiently many of these, features without matching reflections were used.

We updated the global map with the estimated location $\hat{\mathbf{p}}^w_{i,k}$ of each feature, which is derived from the state estimate $\hat{\mathbf{x}}^b_{i,k}$ of the feature. We initialized the location estimate and the velocity estimate of the UAV as $\hat{\mathbf{p}}^w_{b,0} = (0, 0, -z^w_{b,0})^T$ and $\hat{\mathbf{v}}^b_0 = \mathbf{0}$, respectively, and the accelerometer bias estimate as $\hat{\mathbf{b}}^b_0 = \mathbf{0}$. We initialized the altitude $-z^w_{b,0}$ of the UAV over the water with the measurements from the altimeter.

State-of-the-art SLAM methods (Hesch et al., 2014; Scherer et al., 2012; Weiss et al., 2013) rely on loop closing to prevent drift over time. Therefore, we have implemented a simple vision-based algorithm to detect loop closure, and we incorporated a postprocessing stage to minimize the final error between our UAV's location estimate and the ground truth. Our algorithm used speeded-up robust features (SURFs) (Bradski & Kaehler, 2008) to find the best match between image data that were acquired from the starting location and from when our UAV quadcopter revisited the starting point. Then, our algorithm used a Kalman smoother (Särkkä & Sarmavuori, 2013) to constrain

Figure 9. We acquired the real-environment data by using our quadcopter UAV (highlighted with a red circle). We flew our quadcopter UAV at Crystal Lake in Urbana, Illinois using the altitude hold mode of the onboard automatic flight control system.



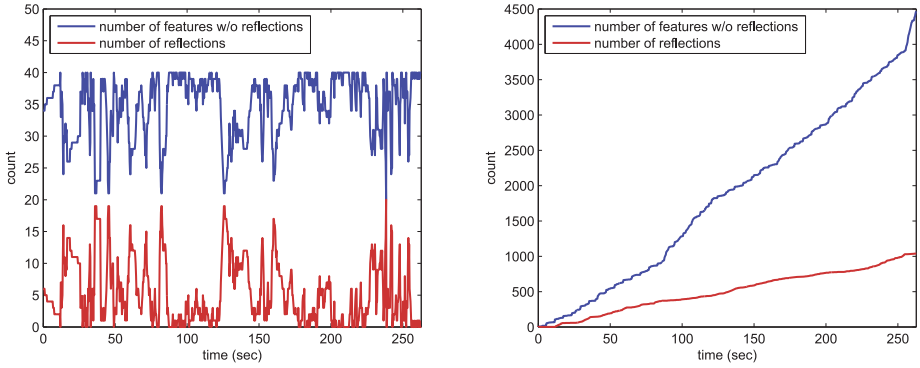
(a) 14 seconds (b) 75 seconds (c) 152 seconds (d) 211 seconds

Figure 10. Feature tracking on image data from Crystal Lake. Feature tracking results (green lines) with the pyramid KLT method and outliers (red lines) are shown in the first row. Matching of the reflections (green boxes) corresponding to real objects (red boxes) with Algorithm 1 are shown in the second row.

the two location estimates to coincide and filtered the entire trajectory.

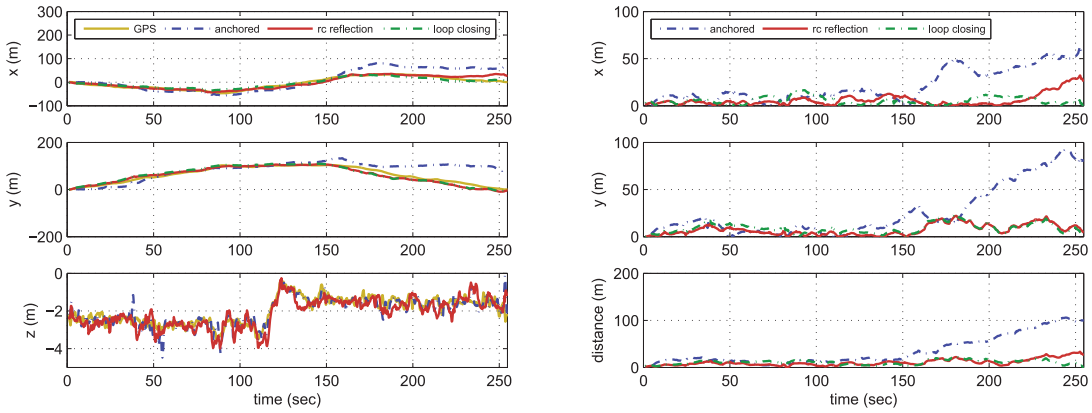
7.2. Experimental Results

Figure 12 gives a quantitative summary of our results. Figure 12(a) shows the ground truth (GPS/INS) location of our quadcopter UAV, the estimated location of our quadcopter from the anchored IDP SLAM method, our method of using reflection measurements (without loop closure), and our method augmented with loop closure detection. Figure 12(b) shows the estimation errors of the three aforementioned methods, relative to the GPS/INS data. Figure 12(c) shows the estimate of the UAV's velocity. Figure 12(d) shows the normalized coordinates and the inverse-depth estimates of the features. In Figure 12(d), the estimation of old features that move out of sight are reinitialized with new features, as we stated in Section 7.1.

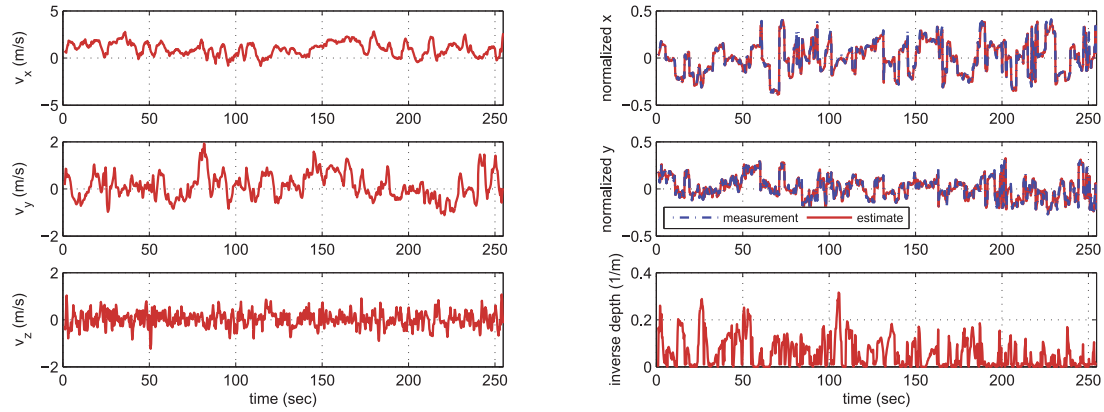


(a) Number of features in each image (b) Accumulated number of features

Figure 11. The number of features incorporated in the measurement vector with and without reflection measurements.



(a) Estimate of the UAV's location (b) Estimation error of the UAV's location



(c) Estimate of the UAV's velocity (d) Estimates of the features

Figure 12. The location estimate of the UAV with respect to the world reference frame, and the velocity estimate of the UAV and the estimates of the point features both with respect to the UAV body frame, are shown. The estimation error of the UAV's location relative to the GPS/INS data is also shown.

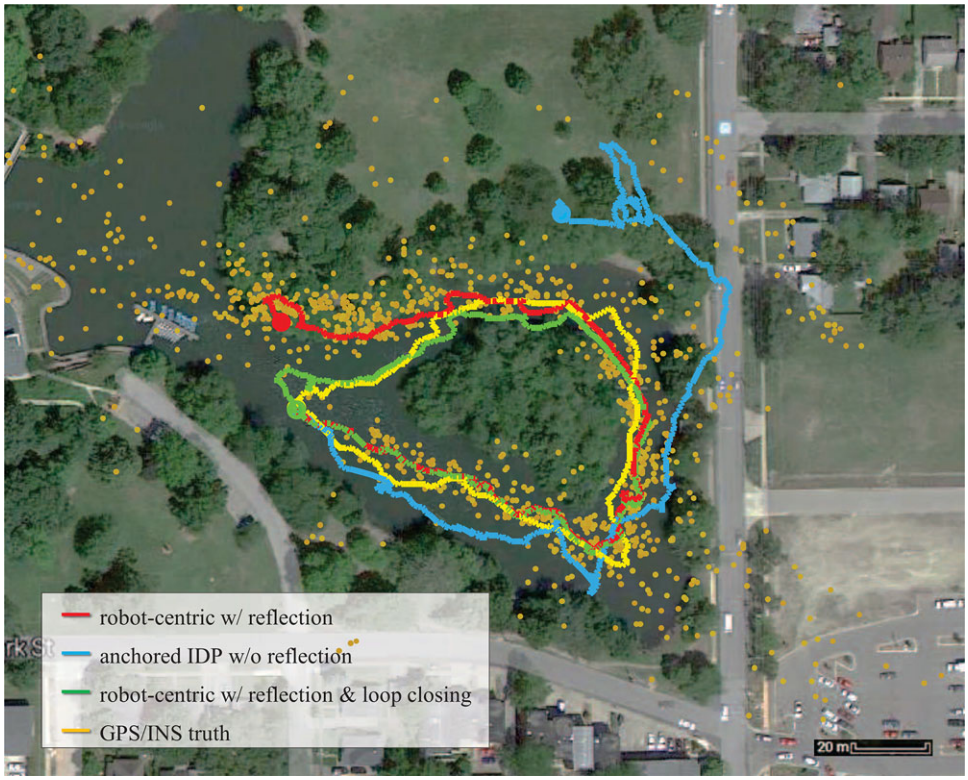


Figure 13. The experimental results are overlaid on a satellite image of Crystal Lake provided by Google Maps. The time-history of the UAV's location estimate from our robot-centric method with reflections (red) and the anchored IDP method without reflections (blue) and the position estimate of the features from our method with reflections (orange dots) are shown. The GPS/INS ground truth trajectory of the UAV (yellow) and the loop closing results with our method using reflections (green) are also shown. The ending locations are marked with circles.

The estimation results and GPS/INS data are overlaid on a satellite image of Crystal Lake provided by Google Maps in Figure 13. According to the GPS data, the quadcopter traveled approximately 343.48 m for 253 s. The rotation with respect to the gravity direction is unobservable in a pure visual-inertial navigation system. However, the sensor package we use compensates the gyro bias and provides angular rate and attitude estimates by using its gyroscope and accelerometer along with a magnetometer, and it makes the unobservable rotation directly measurable. The reduced-order state estimator we presented in Section 4 uses the drift-free attitude information acquired by the IMU and the magnetometer.

The final error between the GPS data and the estimated location of the UAV was 26.26 m for our localization and robot-centric mapping system with reflection measurements, 100.36 m for the anchored IDP SLAM method without reflection measurements, and 0.67 m for our method with loop closing. The average error norm of the UAV's location over the entire trajectory was 10.64 m from our localization and robot-centric mapping system with reflection measurements and 34.93 m from the anchored IDP SLAM system without reflection measurements.

7.3. Lessons Learned

As can be seen in both Figures 12(b) and 13, our method outperforms the anchored IDP method, and incorporating loop closure provides further improvement. In particular, the drift along the X-Y plane is reduced when we used our localization and robot-centric mapping system, which uses reflection measurements, and it is nearly eliminated when loop closure is exploited. We believe that the inaccuracies in the localization results for the anchored IDP method were due in part to inaccurate estimation of feature depth. Our method is able to exploit additional geometrical constraints imposed by using reflection measurements when estimating the depths of the features and the location of the UAV. A second advantage for our method is its larger degree of observability (Section 5).

Even though our method outperformed the anchored IDP method in real experiments, the difference in performance of our method for simulations versus real-world

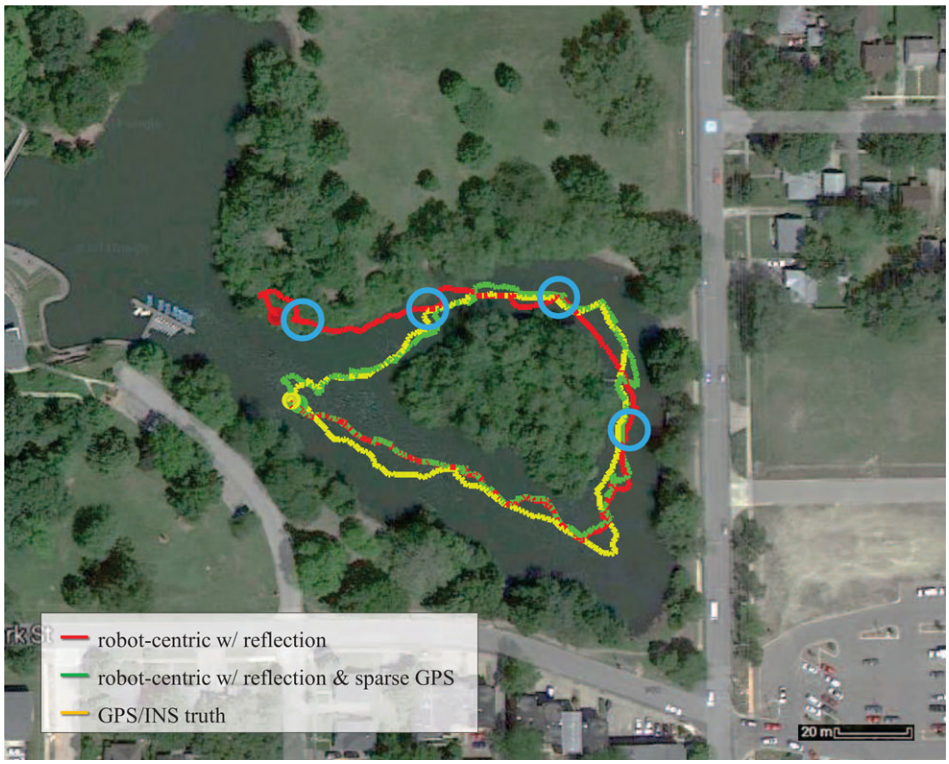


Figure 14. The experimental results show that a short sequences of badly estimated poses (blue circles) can cause the pose estimates to drift (red). The localization result that is obtained when GPS data are provided as measurements to the smoothing filter around these points is shown as a comparison (green).

experiments raises issues that merit discussion. The most significant cause for the difference between simulation and experimental performance is likely tied to the quality of feature matching, and consequent feature tracking error. For our simulations, we modeled the error in the vision measurements with Gaussian noise, but we did not model incorrect vision measurements caused by mismatch of reflections and drift in feature tracking results. In simulations, features with reflections were always visible to the UAV. In contrast, for our experiments, there were instances for which Algorithm 1 was unable to find reflections. This can be seen in Figure 11(a), which shows that the number of detected reflection features varied significantly over the course of the experiment. Further, in the experimental data, there were instances of incorrect feature matching and tracking, as shown in Figure 10(c).

A secondary factor in the mismatch between simulated and experimental results is related to the geometry of the environment. In simulations, features were located between 5 and 20 m away from the UAV, while for our experiments, the features that were available in the scene were sometimes significantly more distant. As features become more distant, the accuracy of our method decreases, and this can be seen in our experimental results.

Finally, as with all localization and mapping methods, the incremental nature of the pose estimation process is such that a short sequence of badly estimated poses can cause the pose estimates to diverge. This is illustrated in Figure 14. At the positions indicated by the blue circles, significant pose estimation error occurred, and from these points onward, the localization error begins to drift. To more fully illustrate this, we also show in green the localization result that is obtained when GPS data are provided as measurements to a Kalman smoother near these points in the trajectory to process the GPS data over a sequence of local intervals. This demonstrates the detrimental consequences of even a small number of pose estimation errors, and it points to the utility of our method in situations for which intermittent GPS data might be available.

8. CONCLUSION

In this paper, we presented a vision-based SLAM algorithm developed for riverine environments. To our knowledge, the water reflections of the surrounding features for SLAM are used for the first time. The performance of our visual SLAM algorithm has been validated through numerical simulations. We also demonstrated the effectiveness of our

algorithm with real-world experiments that we conducted at Crystal Lake. The numerical simulation results and the real-environment experimental results show that the accuracy in the estimation of the UAV's location along the X-Y plane in riverine environments is greatly improved by using our localization and robot-centric mapping framework with reflection measurements.

We believe that the water reflections of the surrounding features are important aspects of riverine environments. The localization results of our localization and robot-centric mapping system with reflection measurements outperformed the anchored IDP SLAM method because additional geometrical constraints are exploited by using reflection measurements to estimate the depths of the features and the location of the UAV. In contrast, without the geometrical constraints from the reflection measurements, the anchored IDP SLAM method lacked reliable depth information of the features that could improve the performance of the localization and mapping. The superior performance of our localization and robot-centric mapping system with reflection measurements was expected in the experiments due to its larger degree of observability compared to the anchored IDP SLAM method.

Future research could extend this work by employing other sensors and vision techniques. The improved estimation results from sensor fusion approaches could be applied for autonomous guidance and control of the UAV in a riverine environment.

ACKNOWLEDGMENTS

This material is based in part upon work supported by the Office of Naval Research (N00014-11-1-0088 and N00014-14-1-0265) and John Deere. We would like to thank Ghazaleh Panahandeh, Hong-bin Yoon, Martin Miller, Simon Peter, Sunil Patel, and Xichen Shi for their help.

APPENDIX: INDEX TO MULTIMEDIA EXTENSIONS

The video of our experimental results using Crystal Lake data is available in the online version of this article.

REFERENCES

Aroda, T. (2013). The rain forests of the mighty Amazon river. Retrieved January 16, 2014, from http://tomaroda.hubpages.com/hub/The-rain-forests-of-the-mighty-Amazon-river/.

Bachrach, A., Prentice, S., He, R., Henry, P., Huang, A. S., Krainin, M., Maturana, D., Fox, D., & Roy, N. (2012). Estimation, planning, and mapping for autonomous flight using an RGB-D camera in GPS-denied environments. The International Journal of Robotics Research, 31(11), 1320–1343.

Bailey, T., & Durrant-Whyte, H. (2006). Simultaneous localization and mapping (SLAM): Part II. IEEE Robotics Automation Magazine, 13(3), 108–117.

Boberg, A., Bishop, A., & Jensfelt, P. (2009). Robocentric mapping and localization in modified spherical coordinates with bearing measurements. In Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing (pp. 139–144), Melbourne, Australia.

Bouguet, J. Y. (2008). Camera calibration toolbox for Matlab. Retrieved February 19, 2013, from http://www.vision.caltech.edu/bouguetj/calib_doc/.

Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc.

Bryson, M., & Sukkarieh, S. (2007). Building a robust implementation of bearing-only inertial SLAM for a UAV. Journal of Field Robotics, 24(1-2), 113–143.

Bryson, M., & Sukkarieh, S. (2008). Observability analysis and active control for airborne SLAM. IEEE Transactions on Aerospace and Electronic Systems, 44(1), 261–280.

Castellanos, J. A., Martinez-cantin, R., Tardos, J. D., & Neira, J. (2007). Robocentric map joining: Improving the consistency of EKF-SLAM. Robotics and Autonomous Systems, 55, 21–29.

Ceriani, S., Marzorati, D., Matteucci, M., Migliore, D., & Sorrenti, D. G. (2011). On feature parameterization for EKF-based monocular SLAM. In Proceedings of the 18th World Congress of the International Federation of Automatic Control (pp. 6829–6834), Milan, Italy.

Chaumette, F., & Hutchinson, S. (2006). Visual servo control. I. Basic approaches. IEEE Robotics Automation Magazine, 13(4), 82–90.

Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., & Thrun, S. (2005). Principles of robot motion: Theory, algorithms, and implementations. Cambridge, MA: MIT Press.

Chowdhary, G., Johnson, E. N., Magree, D., Wu, A., & Shein, A. (2013). GPS-denied indoor and outdoor monocular vision aided navigation and control of unmanned aircraft. Journal of Field Robotics, 30(3), 1556–4967.

Civera, J., Davison, A., & Montiel, J. M. M. (2008). Inverse depth parametrization for monocular SLAM. IEEE Transactions on Robotics, 24(5), 932–945.

Civera, J., Grasa, O. G., Davison, A. J., & Montiel, J. M. M. (2010). 1-point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry. Journal of Field Robotics, 27(5), 609–631.

Clemente, L., Davison, A., Reid, I., Neira, J., & Tardós, J. D. (2007). Mapping large loops with a single hand-held camera. In Proceedings of Robotics: Science and Systems (pp. 297–304), Atlanta, GA.

Crassidis, J., Landis Markley, F., & Cheng, Y. (2007). Survey of nonlinear attitude estimation methods. Journal of Guidance, Control, and Dynamics, 30(1), 12–28.

Dani, A., Chung, S.-J., & Hutchinson, S. (2015). Observer design for stochastic nonlinear systems via contraction-based

incremental stability. IEEE Transactions on Automatic Control, 60(3), 700–714.
Dani, A., Fischer, N., & Dixon, W. (2012). Single camera structure and motion. IEEE Transactions on Automatic Control, 57(1), 238–243.
Dani, A., Panahandeh, G., Chung, S.-J., & Hutchinson, S. (2013). Image moments for higher-level feature based navigation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 602–609), Tokyo.
Davison, A., Reid, I., Molton, N., & Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(6), 1052–1067.
Dixon, W., Fang, Y., Dawson, D., & Flynn, T. (2003). Range identification for perspective vision systems. IEEE Transactions on Automatic Control, 48(12), 2232–2238.
Doitsidis, L., Weiss, S., Renzaglia, A., Achtelik, M., Kosmatopoulos, E., Siegwart, R., & Scaramuzza, D. (2012). Optimal surveillance coverage for teams of micro aerial vehicles in GPS-denied environments using onboard vision. Springer Tracts in Advanced Robotics, 33(1-2), 173–188.
Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous localization and mapping: Part I. IEEE Robotics Automation Magazine, 13(2), 99–110.
Fallon, M. F., Papadopoulos, G., Leonard, J. J., & Patrikalakis, N. M. (2010). Cooperative AUV navigation using a single maneuvering surface craft. The International Journal of Robotics Research, 29(12), 1461–1474.
Forster, C., Pizzoli, M., & Scaramuzza, D. (2014). Fast semi-direct monocular visual odometry. In Proceedings of the IEEE International Conference on Robotics and Automation (pp. 15–22), Hong Kong.
Franklin, G. F., Powell, D. J., & Emami-Naeini, A. (2001). Feedback control of dynamic systems, 4th ed. Upper Saddle River, NJ: Prentice Hall.
Haner, S., & Heyden, A. (2010). On-line structure and motion estimation based on a novel parameterized extended Kalman filter. In International Conference on Pattern Recognition (pp. 1836–1839), Istanbul, Turkey.
Haralick, R., & Shapiro, L. (1993). Computer and robot vision. Reading, MA: Addison-Wesleyx.
Hermann, R., & Krener, A. (1977). Nonlinear controllability and observability. IEEE Transactions on Automatic Control, 22(5), 728–740.
Hesch, J., Kottas, D., Bowman, S., & Roumeliotis, S. (2014). Consistency analysis and improvement of vision-aided inertial navigation. IEEE Transactions on Robotics, 30(1), 158–176.
Hesch, J., Mourikis, A., & Roumeliotis, S. (2009). Mirror-based extrinsic camera calibration. Springer Tracts in Advanced Robotics, 57, 285–299.
Huang, G. P., Mourikis, A. I., & Roumeliotis, S. I. (2010). Observability-based rules for designing consistent EKF SLAM estimators. The International Journal of Robotics Research, 29(5), 502–528.
Jain, S., Nuske, S., Chambers, A., Yoder, L., Cover, H., Chamberlain, L., Scherer, S., & Singh, S. (2013). Autonomous river exploration. In Proceedings of the International Conference on Field and Service Robotics (pp. 93–106), Brisbane. Australia.
Jankovic, M., & Ghosh, B. K. (1995). Visually guided ranging from observations of points, lines and curves via an identifier based nonlinear observer. Systems & Control Letters, 25(1), 63–73.
Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., & Dellaert, F. (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree. The International Journal of Robotics Research, 31(2), 216–235.
Kelly, J., & Sukhatme, G. S. (2011). Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. The International Journal of Robotics Research, 30(1), 56–79.
Kerl, C., Sturm, J., & Cremers, D. (2013). Dense visual SLAM for RGB-D cameras. In Proceedings of the International Conference on Intelligent Robot Systems (pp. 2100–2106), Tokyo.
Kim, A., & Eustice, R. M. (2013). Real-time visual SLAM for autonomous underwater hull inspection using visual saliency. IEEE Transactions on Robotics, 29(3), 719–733.
Klein, G., & Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (pp. 225–234), Nara, Japan.
Krener, A. J., & Ide, K. (2009). Measures of unobservability. In Proceedings of the IEEE Conference on Decision and Control (pp. 6401–6406), Shanghai.
Lee, K. W., Wijesoma, W. S., & Guzman, J. I. (2006). On the observability and observability analysis of SLAM. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 3569–3574), Beijing.
Leedekerken, J. C., Fallon, M. F., & Leonard, J. J. (2010). Mapping complex marine environments with autonomous surface craft. In Proceedings of the International Symposium on Experimental Robotics (pp. 331–340), Delhi, India.
Leutenegger, S., Furgale, P., Rabaud, V., Chli, M., Konolige, K., & Siegwart, R. (2013). Keyframe-based visual-inertial SLAM using nonlinear optimization. In Proceedings of Robotics: Science and Systems, Berlin.
Lobo, J., & Dias, J. (2007). Relative pose calibration between visual and inertial sensors. The International Journal of Robotics Research, 26(6), 561–575.
Mariottini, G. L., Scheggi, S., Morbidi, F., & Prattichizzo, D. (2012). Planar mirrors for image-based robot localization and 3-D reconstruction. Mechatronics—Special Issue on Visual Servoing, 22, 398–409.
Martinelli, A. (2014). Closed-form solution of visual-inertial structure from motion. International Journal of Computer Vision, 106(2), 138–152.
Mourikis, A., & Roumeliotis, S. (2007). A multi-state constraint Kalman filter for vision-aided inertial navigation. In Proceedings of the International Conference on Robotics and Automation (pp. 3565–3572), Rome.
Panahandeh, G., & Jansson, M. (2011). IMU-camera self-calibration using planar mirror reflection. In Proceedings

of the International Conference on Indoor Positioning and Indoor Navigation (pp. 1–7), Guimarães, Portugal.

Rao, D., Chung, S.-J., & Hutchinson, S. (2012). CurveSLAM: An approach for vision-based navigation without point features. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 4198–4204), Vilamoura, Portugal.

Rehder, J., Gupta, K., Nuske, S., & Singh, S. (2012). Global pose estimation with limited GPS and long range visual odometry. In Proceedings of the IEEE International Conference on Robotics and Automation (pp. 627–633), St. Paul, MN.

Reif, K., Gunther, S., Yaz, E., & Unbehauen, R. (1999). Stochastic stability of the discrete-time extended Kalman filter. IEEE Transactions on Automatic Control, 44(4), 714–728.

Scherer, S., Rehder, J., Achar, S., Cover, H., Chambers, A., Nuske, S., and Singh, S. (2012). River mapping from a flying robot: State estimation, river detection, and obstacle mapping. Autonomous Robots, 33(1-2), 189–214.

Shi, J., & Tomasi, C. (1994). Good features to track. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 593–600), Seattle.

Sola, J., Vidal-Calleja, T., Civera, J., & Montiel, J. M. M. (2012). Impact of landmark parametrization on monocular EKF-SLAM with points and lines. The International Journal of Computer Vision, 97(3), 339–368.

Strasdat, H., Montiel, J. M. M., & Davison, A. (2010). Real-time monocular SLAM: Why filter? In Proceedings of the IEEE International Conference on Robotics and Automation (pp. 2657–2664), Anchorage, AK.

Särkkä, S., & Sarmavuori, J. (2013). Gaussian filtering and smoothing for continuous-discrete dynamic systems. Signal Processing, 93(2), 500–510.

Weiss, S., Achtelik, M., Lynen, S., Achtelik, M., Kneip, L., Chli, M., and Siegwart, R. (2013). Monocular vision for long-term micro aerial vehicle state estimation: A compendium. Journal of Field Robotics, 30(5), 803–831.

Weiss, S., Scaramuzza, D., & Siegwart, R. (2011). Monocular-SLAM based navigation for autonomous micro helicopters in GPS-denied environments. Journal of Field Robotics, 28(6), 854–874.

Williams, B., & Reid, I. (2010). On combining visual SLAM and visual odometry. In Proceedings of the IEEE International Conference on Robotics and Automation (pp. 3494–3500), Anchorage, AK.

Yang, J., Chung, S.-J., Hutchinson, S., Johnson, D., & Kise, M. (2013a). Vision-based localization and mapping for an autonomous mower. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 3655–3662), Tokyo.

Yang, J., Dani, A., Chung, S.-J., & Hutchinson, S. (2013b). Inertial-aided vision-based localization and mapping in a riverine environment with reflection measurements. In Proceedings of the AIAA Guidance, Navigation, and Control Conference (pp. AIAA 2013–5246), Boston.

Yang, J., Rao, D., Chung, S.-J., & Hutchinson, S. (2011). Monocular vision based navigation in GPS denied riverine environments. In Proceedings of the AIAA Infotech at Aerospace Conference (pp. AIAA–2011–1403), St. Louis.

Zhang, H., Guo, X., & Cao, X. (2010). Water reflection detection using a flip invariant shape detector. In Proceedings of the International Conference on Pattern Recognition (pp. 633–636), Istanbul, Turkey.

Zhong, S.-H., Liu, Y., Liu, Y., & Li, C.-S. (2013). Water reflection recognition based on motion blur invariant moments in curvelet space. IEEE Transactions on Image Processing, 22(11), 4301–4313.