Inertial-Aided Vision-Based Localization and Mapping in a Riverine Environment with Reflection Measurements

Junho Yang *, Ashwin Dani †, Soon-Jo Chung ‡, and Seth Hutchinson §

University of Illinois at Urbana-Champaign, Urbana, IL 61801

This paper presents an inertial-aided vision-based localization and mapping algorithm for an unmanned aerial vehicle (UAV) that can operate in a GPS-denied riverine environment. We take vision measurements from the features surrounding the river and their corresponding points reflected in the river. We apply a robot-centric mapping framework to let the uncertainty of the features be referenced to the UAV body frame and estimate the 3D positions of point features while estimating the location of the UAV. We demonstrate the localization and mapping results with sensors on our quadcopter UAV platform in the University of Illinois at Urbana Champaign Boneyard Creek. The UAV is equipped with a light weight monocular camera, an inertial measurement unit (IMU) which contains a magnetometer, an ultrasound altimeter, and an on-board computer. To our knowledge, we report the first result of performing localization and mapping by exploiting multiple views with reflections of features in a river-like environment.

I. INTRODUCTION

Recent advances in navigation technologies using on-board local sensing modalities are enabling intelligence, surveillance, and reconnaissance (ISR) missions by UAVs in a range of diverse environments [1–5] where GPS signal is intermittent or is not available. Our goal in this research is to further expand the scope of future ISR missions by developing a localization and mapping algorithm using on-board sensing particularly for navigation in GPS-denied riverine environments. To perform navigation for ISR missions while saving power and payload of the UAV, we present a localization and mapping algorithm which uses a lightweight monocular camera, as well as an IMU which has a magnetometer, and an ultrasound altimeter, which are typically available on-board for the autopilot control system of a UAV.

A contribution of this work is specific to the riverine environments or to the environments where reflection of the periphery can be observed. The reflection of the surrounding environment on the river surface shown in Figure 1 is an important aspect of riverine environments which can be exploited in the estimation algorithm. A two-view geometry formulation can be exploited by considering the point features from a real object and its reflection in the river. The feature point matching information can be used along with the attitude and altitude information of the UAV. One should consider a scenario where all the features observed from the camera are in long distance when the UAV is navigating along the river. Neither monocular SLAM methods that only rely on motion parallax nor a stereo camera with limited baseline distance will perform well. Light detection and ranging (LIDAR) sensor that are capable of ranging distant features will add too much load to the UAV for long missions. On the other hand, our localization and mapping method exploiting the reflection measurements can be highly effective for navigation of a UAV with distant features along the river.

A. Paper Organization

The rest of this paper is organized as follows. In Section II, we describe where our work stands among the related work. In Section III, we give an overview of our navigation system and explain the dynamics and the
measurement model for our system. In Section IV, we formulate an EKF estimator for UAV localization and point feature mapping. In Section V, we analyze the effectiveness of the proposed algorithm with numerical simulation results. Finally, in Section VI, we show offline experimental results of vision-based navigation of a UAV in a river-like environment and summarize our work in Section VII.

II. Related Work

A necessary precursor for navigating in an unknown environment with local measurements is to progressively construct a map of its surroundings while localizing itself within the map, a process often known as simultaneous localization and mapping (SLAM) [6–8]. Specifically for riverine environments, SLAM was performed in [9] with an autonomous surface-craft to build a map of both above and below the water surface. A sonar was used for subsurface mapping, and LIDAR, camera, and radar sensors were used for terrestrial mapping to account for degradation of GPS. In [10], a vision-based SLAM algorithm exploiting planar ground assumption on the border of the water and the ground was presented for localization and mapping in riverine environments. In [11], an on-board vision-based river detection and navigation algorithm was presented for an autonomous UAV to monitor a river from high altitude. In [12], a navigation algorithm was presented to map a river from a boat fully equipped with stereo camera, LIDAR, IMU, and intermittent GPS. A graph-based representation was used to estimate the vehicle’s state with vision and GPS, while mapping the river with a self-supervised river detection algorithm and the obstacles with a LIDAR.

Localization and Mapping with monocular vision is challenging because the distance to a feature cannot be estimated from a single measurement. EKF has been widely used for SLAM but it does not cope well with nonlinear systems [13], especially when the features are estimated in Cartesian coordinates with monocular vision measurements. In [14], a monocular vision based SLAM problem was solved by sequentially updating measurements from different locations. The positions of features were added to the map after having enough parallax from the initial observation. As an alternative, different parametrization such as the inverse depth parametrization [15] and the anchored homogeneous point [13] have been proposed. Instead of estimating the Cartesian coordinates of features in a world reference frame, these methods define an anchor location which is the location of the camera where a set of point features are first observed. They parameterize the point features using these anchor locations, the direction of the feature with respect to the world reference frame, and the distance between the feature and the anchor. The inverse of the distance was used to alleviate the nonlinearity of the measurement model and to introduce new features to the map immediately. Robot-centric estimation opposed to world-centric estimation has also been used recently in vision-based SLAM [16–18] to
alleviate the linearization error in the EKF-based estimation.

Localization and mapping using higher level features has also been explored recently. Curves along the border of the riverbank or on the edge of a road were used in [19]. Their approach used cubic Bézier curves as stereo vision primitives and formulated a SLAM algorithm to estimate 6 degree-of-freedom (DOF) pose of a camera pair and produce a map with the curves. Image moments of region features were used in [20] instead of using a large number of feature points. A new nonlinear estimator [21] which computes an optimal state dependent coefficient form was applied to the moment-based SLAM method.

In the scope of observer-based structure from motion (SFM) feature tracking, a nonlinear observer was designed in [22] with Lyapunov theory to guarantee stability of the range estimation with a single perspective system and inertial measurements. This can be viewed as a camera-centric mapping problem. The dynamics of the system were derived for a system which has the normalized coordinates of a feature and the inverse of its depth as its state. The work was extended in [23] to a structure and motion problem where only one linear velocity and a corresponding acceleration of a camera were given. A reduced-order observer was designed to estimate the depth of a feature and the motion of the camera. In optimization-based SFM research, parallel tracking and mapping (PTAM) [24] enabled real-time processing by separating the tracking of the camera and mapping of the environment into two parallel tasks. One was tracking the motion of the camera, and the other was recovering a feature based map at keyframes with both local and global bundle adjustments. Their algorithm was demonstrated with a UAV guidance and surveillance application [2].

In [25], a method using a mirror was presented to compute the 6 DOF pose of the camera. The algorithm used observations of known points through the mirror and refined the results with a maximum-likelihood estimator. In [26], an approach for estimating 6 DOF transformation between an IMU and a camera, and intrinsic camera parameters using a mirror was proposed. A sigma-point Kalman filter was applied to estimate the transformation and the camera parameters. In [27], an algorithm using epipolar geometry was presented to compute the location of a camera and reconstruct a 3D scene. A catadioptric system was formulated with multiple planar mirrors in a small indoor experiment setup.

The work presented in this paper establishes the robot-centric mapping framework using normalized coordinates to localization and mapping, specifically for riverine environments. In contrast with the aforementioned SLAM algorithms, we build a map of the unknown environment by composing the measurement model with multiple views of point features including their reflections and estimate the location of the UAV. To our knowledge, we report the first result of performing localization and mapping by using multiple views with reflections of features.

### III. LOCALIZATION AND MAPPING SYSTEM

In this section, we describe the architecture of the overall system. We present the motion model for localization of the UAV and robot-centric mapping of point features. We derive the measurement model with a current view, an initial view, and reflections of point features.

#### A. OVERVIEW OF THE SYSTEM

The work presented in this paper focuses on the UAV localization and riverine mapping with on-board sensing. Figure 2 shows the block diagram of our UAV navigation framework. To alleviate the nonlinearity of the system, on-board sensor readings are used in the propagation stage as well as in the measurement update stage. The proposed work should be applied to close the navigation and control loop in the future (dashed arrow).

Figure 3 shows the quadcopter UAV we used to demonstrate the capability of our algorithm. The UAV contains a lightweight monocular camera facing forward with a resolution of 640 × 480 pixels, an IMU which has a three-axis magnetometer, an ultrasound altimeter, and an on-board computer. We use the sensor measurements in both the propagation stage and the measurement update stage in our EKF estimator. In the propagation stage, the UAV motion model is formulated with IMU and magnetometer readings, and the vision motion model is derived using IMU measurements. In the measurement update stage, the measurement model is formulated with multiple views. We project the features to the camera when they are first observed and at the current timestep. The measurement model is augmented with observations of corresponding reflection points combined with attitude and altitude readings of the UAV.
Figure 2. Block diagram of our vision based UAV localization and mapping.

Figure 3. Our quadcopter UAV which contains the on-board sensors for the experiments.

B. SYSTEM DYNAMICS

1. Dynamic Model for UAV Localization

We define the UAV body frame with the frame of the IMU which is mounted on the UAV. Our world reference frame is defined on the river surface with the projection of the x and y-axis of the UAV body frame when the estimation starts. The z-axis points down along the gravity vector. We estimate the location of the UAV with respect to the world reference frame. The state vector for the UAV is $(x^w_b, v^b) \in \mathbb{R}^6$, where $x^w_b \in \mathbb{R}^3$ is the location of the UAV in the world reference frame and $v^b = (v_1, v_2, v_3) \in \mathbb{R}^3$ is the linear velocity of the UAV in the body frame. The dynamic model for UAV localization is given by

$$\frac{d}{dt} \begin{pmatrix} x^w_b \\ v^b \end{pmatrix} = \begin{pmatrix} R(q^w_b) v^b \\ -[\omega^b]_x v^b + a^b + R(q^w_b) g \end{pmatrix}$$

(1)

where $q^w_b \in \mathbb{R}^4$ is the quaternion attitude of the UAV, which we get separately from the gyroscope and magnetometer readings, $\omega^b = (\omega_1, \omega_2, \omega_3) \in \mathbb{R}^3$ is the angular velocity of the UAV we measure with a gyroscope, $a^b \in \mathbb{R}^3$ is the linear acceleration of the UAV we measure with an accelerometer, $g \in \mathbb{R}^3$ is the gravity vector in the world reference frame. We treat the time-varying bias in angular velocity and linear
acceleration as an uncertainty of the motion model in the estimator.

2. Vision Motion Model for Mapping

For point feature mapping, we exploit robot-centric mapping which allows the state of each point feature to be referenced to the current UAV body frame. It has been shown [13] that an anchor based method has an advantage over a world-centric framework of representing each point feature’s uncertainty in a anchor location which is closer to the point feature than the world reference frame. By using a robot-centric framework, we reference the uncertainty of each point feature to the UAV body frame which becomes the closest to the point features when the UAV approach the features.

The position of each point feature is estimated in the UAV body frame and converted to the world reference frame. In Cartesian coordinates, the dynamics of the i-th feature is given by [28]

\[
\frac{d}{dt} \begin{pmatrix}
  x_i^b \\
  y_i^b \\
  z_i^b
\end{pmatrix} = -\begin{pmatrix}
  0 & -\omega_3 & \omega_2 \\
  \omega_3 & 0 & -\omega_1 \\
  -\omega_2 & \omega_1 & 0
\end{pmatrix} \begin{pmatrix}
  x_i^b \\
  y_i^b \\
  z_i^b
\end{pmatrix} - \begin{pmatrix}
  v_1 \\
  v_2 \\
  v_3
\end{pmatrix}
\]

(2)

where \(x_i^b = (x_i^b, y_i^b, z_i^b)^T \in \mathbb{R}^3\) is the location of the i-th feature with respect to the UAV body frame. The skew-symmetric matrix \([\omega^b]_x \in \text{so}(3)\) is formed from the angular velocity vector \(\omega^b\). A point feature can be described with normalized coordinates \(h_i^b = (h_{i,1}^b, h_{i,2}^b)^T = (y_i^b/x_i^b, z_i^b/x_i^b)^T \in \mathbb{R}^2\) and the distance from the UAV along the x-axis in the UAV body frame. We form the corresponding state vector as \(p_i^b = (h_i^b, \rho_i^b)^T\), where \(\rho_i^b = 1/x_i^b\). The dynamics of the system can be derived from Equation 2 as

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

(3)

Our state vector for localization and mapping is composed of the state of the UAV and the point features on the map. We combine the dynamic model of the UAV and the vision motion model for our estimation algorithm presented in Section IV.

C. VISION MEASUREMENT MODEL

1. Projected Measurements of Features

The normalized pixel coordinates in the camera frame \(h_i^c = (h_{i,1}^c, h_{i,2}^c)^T = (x_i^c/z_i^c, y_i^c/z_i^c)^T \in \mathbb{R}^2\) can be computed using the image pixel of a point feature and the camera calibration matrix [28]. The vector projected on a unit sphere is \(x_{i,s}^b = (x_{i,s}^b, y_{i,s}^b, z_{i,s}^b)^T \in \mathbb{R}^3\). The unit sphere projection can be approximated in the UAV body frame \(x_{i,s}^b = (x_{i,s}^b, y_{i,s}^b, z_{i,s}^b)^T \in \mathbb{R}^3\) with \(x_{i,s} = R_i^b x_{i,s}\), where \(R_i^b\) is the rotation we get from IMU-camera calibration. We project this to a plane and get normalized coordinates \(h_i^b = (h_{i,1}^b, h_{i,2}^b)^T = (y_{i,s}^b/x_{i,s}^b, z_{i,s}^b/x_{i,s}^b)^T \in \mathbb{R}^2\) in the body frame.

We include the initial normalized coordinates of a feature \(h_i^b = (y_i^b/x_i^b, z_i^b/x_i^b)^T \in \mathbb{R}^2\) in the measurement vector and exploit two-view geometry. Let us denote the pose of the UAV when it first observes the i-th feature with \((x_{i}^w, q_{i}^w)\). The Cartesian coordinates of a feature can be described with respect to \((x_{i}^w, q_{i}^w)\) as \(x_i^b = (x_i^b, y_i^b, z_i^b)^T \in \mathbb{R}^3\) and be described in terms of the state of the UAV and the feature itself as

\[
x_i^b = x_i^w + R(q_i^w) x_i^b = R(q_i^w) (x_i^w - x_{i}^b) + R(q_i^w) R(q_i^w) x_i^b
\]

(4)

where \(x_i^b = (1/\rho_i^b, h_{i,1}^b/\rho_i^b, h_{i,2}^b/\rho_i^b)^T\).
2. Measurements of Features Matched with Reflections

Reflection of the surroundings is an important aspect of riverine environments. With the camera facing forward from our UAV, it is likely to see reflection of the environment in the river since the incidence angle of the observation vector surpasses a critical angle [29]. Let us denote the reflection of a point feature $x^b_i$ observed by a camera by $x^b_{i,r} = (x^b_{i,r}, y^b_{i,r}, z^b_{i,r})^T \in \mathbb{R}^3$ and define a virtual point $\tilde{x}^w_i = Sx^w_i \in \mathbb{R}^3$, where

$$S = I - 2nn^T \in \mathbb{R}^3$$
$$n = (0, 0, 1)^T.$$

The normalized coordinates of a reflection $(y^b_{i,r}/x^b_{i,r}, z^b_{i,r}/x^b_{i,r})^T$ is identical to that of a corresponding virtual point $\tilde{h}^b_i = (\tilde{y}^b_i/\tilde{x}^b_i, \tilde{z}^b_i/\tilde{x}^b_i)^T \in \mathbb{R}^2$ [25, 26], where

$$\tilde{x}^b_i = R(q^b_w)(\tilde{x}^w_i - x^w_i)$$
$$= R(q^b_w) \left( S(x^w_i + R(q^b_w)x^b_i) - x^w_i \right)$$

(5)

We compose the vision measurement model with two-view measurements $(h^b_i, h^w_i)$ and the reflection measurement $\tilde{h}^b_i$ and make the localization and mapping system observable.

3. Vision-Data Processing for Reflection Matching

We developed an algorithm that matches the points that are from the surrounding objects and the points from their reflections in the river. The pseudo-code of our algorithm is presented in Algorithm 1. Good features to track are selected in an image by computing the minimum eigenvalue of its structure tensor with Shi-Tomasi corner detector [30]. A 60 $\times$ 60 pixels image patch is selected around each feature point. We invert the image patch vertically to take account of the reflection. A normalized correlation coefficient is computed while the image patch slides on the source image for template matching [31]. We then find a location in the source image that has high matching probability.

To reject outliers, we set a vertical search region across the source image. The slope of the search region is given by

$$\theta = \tan^{-1}\left( \frac{y^c_{i,s} - \tilde{y}^c_{i,s}}{x^c_{i,s} - \tilde{x}^c_{i,s}} \right)$$

(6)

where the reflection of a point feature projected to a unit sphere in the camera frame $\tilde{x}^c_{i,s} = (\tilde{x}^c_{i,s}, \tilde{y}^c_{i,s}, \tilde{z}^c_{i,s})^T$ is given by $\tilde{x}^c_{i,s} = R(q^c_w)R(q^b_w)x^w_{i,s}$. The unit sphere projection of the reflection point is given by $\tilde{x}^w_{i,s} = Sx^w_{i,s}$ in the world reference frame, where $x^w_{i,s} = R(q^w_b)R(q^b_w)x^b_{i,s}$. The unit sphere projection of a point feature from a real object $x^c_{i,s}$ can be computed from the pixel coordinates of a point feature as described in Section

American Institute of Aeronautics and Astronautics
Algorithm 1 Reflection matching in riverine environments

Input: camera orientation and image data
Output: matching of a point feature from a real object and its reflection

1. while no match do
2. Select good features to track with the Shi Tomasi corner detector
3. Select an image patch around each feature and invert the patch
4. Slide the image patch on the source image and compute a normalized correlation coefficient
5. Find a location in the image with high matching probability
6. Compute a search region for each image patch using camera orientation data
7. Reject the match using the search region
8. Track the center of the inverted image patch and its match with a pyramid KLT algorithm
11. end while

III-C-1. The orientation of the camera with respect to the world frame is computed by using the IMU and magnetometer data, and IMU to camera calibration results. Slope $\theta$ of the search region in an image is shown in Figure 5. We reject false matches by using the search region. We then track the center of the inverted image patch and its match in the video sequence with a pyramid KLT algorithm [32]. Among the inverted image patch and its match, we assume that the pixel coordinates of a real object is higher than that of its reflection as long as the match is correct and the UAV is not performing acrobatic maneuver. Figure 5 shows an example of matching real objects and their reflections. The red boxes are the patches around real objects, the black boxes are the patches around its reflection, and the blue line shows the slope $\theta$ of the search region.

IV. ESTIMATOR FOR LOCALIZATION AND MAPPING

In this section, we formulate a discrete-time EKF to estimate the location of the UAV with respect to the world reference frame and the position of point features in the UAV body frame. We then represent the mapping results in the world reference frame.

A. Motion Propagation

Let us denote the mean of the estimates by $\mu = (\hat{x}_b^w)^T, (\hat{v}^b)^T, (\hat{p}_{1:n}^b)^T \in \mathbb{R}^{6+3n}$, where $\hat{x}_b^w \in \mathbb{R}^3$ is the estimated UAV location, $\hat{v}^b = (\hat{v}_1, \hat{v}_2, \hat{v}_3) \in \mathbb{R}^3$ is the estimated UAV linear velocity, and $\hat{p}_{1:n}^b \in \mathbb{R}^{3n}$ is the estimated state of $n$ point features. The covariance of the state is denoted by $\Sigma \in \mathbb{R}^{(6+3n) \times (6+3n)}$. The state of the UAV are propagated at timestep $k$ through the dynamic model given by

$$\mu_k = \mu_{k-1} + f(\mu_{k-1}, q_{w,k}^b, \omega_k^b, \epsilon_k^b) \Delta t$$

where
The measurement model is given by

\[
\begin{pmatrix}
R(q_{w,k}^b) \hat{v}_{k-1}^b \\
-\omega_{k}^b \times \hat{v}_{k-1}^b + a_k^b + R(q_{w,k}^b)g \\
\end{pmatrix}
\]

\[
\begin{pmatrix}
(-\hat{v}_{2,k-1} + \hat{h}_{1,1,k-1}^b \hat{v}_{1,k-1}^b) \hat{\rho}_{n,k-1}^b + \hat{h}_{1,2,k-1}^b \omega_{1,k} + \frac{1}{2} \left( \hat{h}_{1,1,k-1}^b \right)^2 \omega_{3,k} + \hat{h}_{1,1,k-1}^b \hat{h}_{1,2,k-1}^b \omega_{2,k} \\
(-\hat{v}_{3,k-1} + \hat{h}_{1,2,k-1}^b \hat{v}_{1,k-1}^b) \hat{\rho}_{n,k-1}^b - \hat{h}_{1,1,k-1}^b \omega_{1,k} + \frac{1}{2} \left( \hat{h}_{1,2,k-1}^b \right)^2 \omega_{2,k} - \hat{h}_{1,1,k-1}^b \hat{h}_{1,2,k-1}^b \omega_{3,k} \\
\vdots \\
(-\hat{v}_{n,k-1} + \hat{h}_{1,n,k-1}^b \hat{v}_{1,k-1}^b) \hat{\rho}_{n,k-1}^b + \hat{h}_{n,2,k-1}^b \omega_{1,k} + \frac{1}{2} \left( \hat{h}_{n,1,k-1}^b \right)^2 \omega_{3,k} + \hat{h}_{n,1,k-1}^b \hat{h}_{n,2,k-1}^b \omega_{2,k} \\
(-\hat{v}_{n,k-1} + \hat{h}_{n,2,k-1}^b \hat{v}_{1,k-1}^b) \hat{\rho}_{n,k-1}^b - \hat{h}_{n,1,k-1}^b \omega_{1,k} + \frac{1}{2} \left( \hat{h}_{n,2,k-1}^b \right)^2 \omega_{2,k} - \hat{h}_{n,1,k-1}^b \hat{h}_{n,2,k-1}^b \omega_{3,k} \\
\end{pmatrix}
\]

\[
(8)
\]

The covariance matrix is propagated through

\[
\Sigma_k = F_k \Sigma_{k-1} F_k^T + W_k
\]

where \( F_k \) is the Jacobian of the motion model \( f(\mu_{k-1}, q_w^{b,k}, \omega_k^b, a_k^b) \), and \( W_k \) represents the covariance of the process noise. Here, \( \mu_{k-1} \) is the estimated state from the previous timestep, \( q_w^{b,k} \) is the attitude of the UAV provided by the magnetometer and the gyroscope, \( \omega_k^b \) and \( a_k^b \) are the angular velocity and linear acceleration of the UAV measured with the IMU at current timestep \( k \).

**B. Measurement Update**

The measurement model is given by

\[
h(\hat{\mu}_k, q_w^{b,k}, q_w^{w,k}) = \begin{pmatrix}
\hat{h}_{1,n}^b(\hat{\mu}_k) \\
\hat{h}_{1,n}^w(\hat{\mu}_k, q_w^{b,k}, q_w^{w,k}) \\
\hat{h}_{1,n}^w(\hat{\mu}_k, q_w^{b,k}) \\
-\hat{z}_{w,k}^w
\end{pmatrix}
\]

\[
(10)
\]

for \( n \) point features, where the altitude \( -\hat{z}_{w,k}^w \) is measured from the ultrasound altimeter, and the current view \( \hat{h}_{1,n}^b(\mu_k) \) of the \( i \)-th point feature is given by

\[
h_i^b(\hat{\mu}_k) = \begin{pmatrix}
\hat{h}_{i,1,k}^b \\
\hat{h}_{i,2,k}^b
\end{pmatrix}
\]

\[
(11)
\]

The initial view \( \hat{h}_{1,n}^b(\mu_k) \) of the \( i \)-th point feature is given by

\[
h_i^b(\mu_k, q_w^{b,k}, q_w^{w,k}) = \begin{pmatrix}
\hat{q}_{i,k}^b/\hat{z}_{i,k}^b, \hat{z}_{i,k}^b/\hat{z}_{i,k}^b
\end{pmatrix}
\]

\[
(12)
\]

where

\[
\hat{z}_{i,k}^b = R(q_{w,k}^b)(\hat{x}_{b,k}^b - \hat{x}_{i,k}^w) + R(q_{w,k}^b)R(q_{w,k}^b)\hat{x}_{i,k}^b
\]

\[
\hat{x}_{i,k}^b = \begin{pmatrix}
1/\hat{\rho}_{i,k}, \hat{h}_{i,1,k}/\hat{\rho}_{i,k}, \hat{h}_{i,2,k}/\hat{\rho}_{i,k}
\end{pmatrix}
\]

\[
(13)
\]
Here, $\hat{x}_b^w_i$ is the estimated location of the UAV when it first observes the $i$-th feature and $q_b^w_i$ is the measured attitude of the UAV at that moment. The current view $\hat{h}_{1:n}^b(\mu_k, q_b^w_i, q_b^{w,b})$ of a reflection of the $i$-th feature is given by

$$\hat{h}_i^b(\mu_k, q_b^w_i, q_b^{w,b}) = \left( \frac{\hat{z}_{i,k}^b}{z_{i,k}^b}, \frac{\hat{y}_{i,k}^b}{y_{i,k}^b}, \frac{\hat{z}_{i,k}^b}{z_{i,k}^b} \right)^T.$$  

where

$$\hat{x}_{i,k}^b = R(q_{w,k}^b) \left( S(\hat{x}_b^w_i) + R(\hat{q}_b^w)\hat{x}_{i,k}^b - \hat{x}_b^w_i \right).$$  

The mean and covariance of the state is updated with vision measurements by

$$\begin{align*}
\mu_k &= \hat{\mu}_k + K_k (h_k - h(\hat{\mu}_k, q_{b,k}^w, q_b^{b_i})) \\
\Sigma_k &= \hat{\Sigma}_k - K_k H_k \hat{\Sigma}_k
\end{align*}$$

where the Kalman gain is given by

$$K_k = \hat{\Sigma}_k H_k^T \left( H_k \hat{\Sigma}_k H_k^T + V_k \right)^{-1}$$

Here, $H_k$ is the Jacobian of the measurement model $h(\mu_k, q_{b,k}^w, q_b^{b_i})$, and $V_k$ is the covariance of the measurement noise.

### C. Conversion to World Reference Frame

We used a robot-centric mapping framework to estimate the position of the point features. By using the robot-centric approach, we are able to represent the uncertainty of each point feature with respect to the UAV body frame which becomes closer to the features than the world reference frame or an anchored frame.

After each EKF cycle, we convert the mean values of the estimated point features to the world reference frame by

$$\hat{x}_{b,k}^w = \hat{x}_{b,k}^w + R(q_{b,k}^w)\hat{x}_{b,k}^b$$

where $\hat{x}_{b,k}^w$ and $\hat{x}_{b,k}^b$ are the state estimated by the EKF, and $q_{b,k}^w$ is the attitude reading of the UAV. By converting the estimated mean to the world reference frame, we are able to generate a stationary map instead of having point features tracked in the UAV body frame.

### V. NUMERICAL SIMULATIONS

Numerical simulations are presented to analyze the performance of our localization and mapping algorithm. We consider a UAV flying in a riverine environment where 20 randomly distributed point features and their reflections are in the sight of the camera. We let the UAV fly forward with an initial linear velocity of 0.3 m/s and added random disturbance to generate the motion. We added Gaussian white noise with 0.01 standard deviation to the linear acceleration and angular velocity readings, and added 0.001 standard deviation to the attitude and altitude measurements. We corrupted the camera pixel measurements with Gaussian white noise with standard deviation of 1. The location state of the UAV is initialized as $\hat{x}_b^w = (0, 0, z_b^w)^T$, where $-z_b^w \in \mathbb{R}^+$ is measured by an on-board altitude sensor. The linear velocity state of the UAV is initialized as $\hat{v}_b^w = (0, 0, 0)^T$. The state for the $i$-th point feature is initialized as $\hat{p}_i^w = (h_{b,1}^i, h_{b,2}^i, 0.1)^T$, where $h_{b,1}^i$ and $h_{b,2}^i$ are from the vision measurements of the $i$-th feature.

Figure 6 shows the localization and mapping results. The light blue plane on the x-y plane is the water surface plane. The black curve and the square at the end of it shows the ground truth trajectory of the UAV and its current location. The ground truth position of the point features are marked with black stars. The virtual points described in Section III-C-2 are marked with cyan stars, and the positions of the reflections observed by the camera are marked with blue dots on the river surface plane. The red curve and the square is the estimated trajectory of the UAV and its current location. The estimated positions of the point features are marked with red stars. The simulation results show the convergence of the estimates to the ground truth.
Figure 6. Simulation results of localization and mapping in a riverine environment. The UAV is flying over the river and measuring 20 point features that have their reflection points in the river.
Figure 7. Estimation results of the UAV location with respect to the world reference frame, its linear velocity in the body frame, and the normalized coordinates and the inverse depth of one of the point features.

Figure 8. RMSE of the location and velocity of the UAV and the inverse depth of all the point features.
Figure 9. Estimation error and standard deviation of the location and linear velocity of the UAV, and the average estimation error of all the point features.

Figure 7 shows the estimation results of each state. The estimates of the location and linear velocity of the UAV, normalized coordinates of a feature, and its inverse of the depth along the x-axis of the UAV body frame all converge to their true values. The results are converted to the world reference frame as shown in Section IV-C to generate the mapping results shown in Figure 6. Figure 8 shows the root mean squared error (RMSE) of the states of the UAV and the average RMSE of all the point features. Figure 9 shows the estimation error converging towards zero and the $3\sigma$ standard deviation. The standard deviation of the estimates are bounded since the system is observable with the multiple-view measurements which includes the vision measurements from reflections.

VI. EXPERIMENTS

To demonstrate our algorithm with real data, we conducted an experiment in the UIUC Engineering Campus at the Boneyard Creek shown in Figure 10. We collected data using our quadcopter UAV while traveling over the creek and processed the data offline. We calibrated the camera [33] and did the IMU to camera calibration [34] to covert the vision measurements to the UAV body frame. We calibrated the IMU measurements to remove the static bias in acceleration and angular velocity. Figure 1 shows the data we collected with the monocular camera attached to the UAV. We processed the data offline and manually selected points from real objects in the image and their corresponding reflections and tracked the points using Lucas Kanade optical flow [35]. The points are marked with red dots and their pixel coordinates are written in blue.
Figure 10. Experiments were conducted in the UIUC Engineering Campus at the Boneyard Creek.

Figure 11 shows the localization and mapping results. The light blue plane on the x-y plane is the water surface plane. The estimated trajectory of the UAV is marked with red, and the estimated location of the point features are marked with blue stars. The points close to the water surface plane are from the stones laid along the border of the creek. Distant points high above the water surface plane are from the building across the engineering quad as shown in Figure 1. Although we need a careful analysis with the ground truth, the distant point features seem to be estimated appropriately, despite the camera is moving roughly towards its optical axis direction. The estimation would fail in such a configuration if we remove the reflection measurements because the system will become almost unobservable without sideway motions.

Figure 12 shows the estimated location and linear velocity of the UAV. Figure 13 shows the estimated state of four point features selected from the data set. The results show the horizontal and vertical normalized coordinates of the point features filtered with their measurements and dynamics. The unmeasurable depth is estimated by its inverse along the x-axis of the UAV body frame. Figure 14 shows the attitude, angular velocity, linear acceleration, and altitude readings we used in the motion model.

VII. CONCLUSION

In this paper, we have presented an inertial-aided vision-based localization and mapping algorithm specifically built for a UAV operating in riverine environments. The reflection in the water is an important aspect of riverine environments, which we exploit in our algorithm. We derived a measurement model using multiple views from features, their initial observations, and their reflections. We used a robot-centric mapping strategy and estimated the point features with respect to the UAV body frame before we convert the mapping results to the world reference frame. We analyzed the performance of our algorithm through numerical simulations and showed the convergence of the estimates and the boundedness of the uncertainty. We demonstrated the capability of the algorithm through offline experiments with real data collected from a creek with our quadcopter UAV. To our knowledge, we report the first result of performing localization and mapping by exploiting multiple views with reflections of features in a river-like environment.

In the near future, we will apply the reflection matching algorithm we developed and conduct online experiments in a larger environment. We plan to improve the estimation results with an alternative nonlinear estimator. We aim to use the estimation results for autonomous guidance and navigation of our UAV in the future.

ACKNOWLEDGMENT

This material is based in part upon work supported by the Office of Naval Research (N00014-11-1-0088) and John Deere. We would like to thank Xichen Shi, Sunil Patel, Martin Miller, Simon Peter, Ghazaleh Panahandehmigjeh, and Shubham Gupta for their help.
Figure 11. Experimental results of localization and mapping.
Figure 12. Estimated values of the location of the UAV with respect to the world reference frame and the linear velocity of the UAV in its body frame from the experiment data set.

Figure 13. Estimated state of four of the point features extracted from the data set.
Figure 14. The attitude, angular velocity, linear acceleration, and altitude readings used in the motion model.
References


