VISION BASED ESTIMATION, LOCALIZATION, AND MAPPING FOR AUTONOMOUS VEHICLES

BY

JUNHO YANG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Doctoral Committee:

Professor Andrew Alleyne, Chair
Associate Professor Soon-Jo Chung, Director of Research
Professor Seth Hutchinson, Director of Research
Assistant Professor Derek Hoiem

Abstract

In this dissertation, we focus on developing simultaneous localization and mapping (SLAM) algorithms with a robot-centric estimation framework primarily using monocular vision sensors. A primary contribution of this work is to use a robot-centric mapping framework concurrently with a world-centric localization method. We exploit the differential equation of motion of the normalized pixel coordinates of each point feature in the robot body frame. Another contribution of our work is to exploit a multiple-view geometry formulation with initial and current view projection of point features. We extract the features from objects surrounding the river and their reflections. The correspondences of the features are used along with the attitude and altitude information of the robot. We demonstrate that the observability of the estimation system is improved by applying our robot-centric mapping framework and multiple-view measurements.

Using the robot-centric mapping framework and multiple-view measurements including reflection of features, we present a vision based localization and mapping algorithm that we developed for an unmanned aerial vehicle (UAV) flying in a riverine environment. Our algorithm estimates the 3D positions of point features along a river and the pose of the UAV. Our UAV is equipped with a lightweight monocular camera, an inertial measurement unit (IMU), a magnetometer, an altimeter, and an onboard computer. To our knowledge, we report the first result that exploits the reflections of features in a riverine environment for localization and mapping.

We also present an omnidirectional vision based localization and mapping system for a lawn mowing robot. Our algorithm can detect whether the robotic mower is contained in a permitted area. Our robotic mower is modified with an omnidirectional camera, an IMU, a magnetometer, and a vehicle speed sensor. Here, we also exploit the robot-centric mapping framework. The estimator in our system generates a 3D point based map with landmarks. Concurrently, the estimator defines a boundary of the mowing area by using the estimated trajectory of the mower. The estimated boundary and the landmark map are provided for the estimation of the mowing location and for the containment detection. First, we derive a nonlinear observer with contraction analysis and pseudo-measurements of the depth of each landmark to prevent the map estimator from diverging. Of particular interest for this work is ensuring that the estimator for

localization and mapping will not fail due to the nonlinearity of the system model. For batch estimation, we design a hybrid extended Kalman smoother for our localization and robot-centric mapping model. Finally, we present a single camera based SLAM algorithm using a convex optimization based nonlinear estimator. We validate the effectiveness of our algorithms through numerical simulations and outdoor experiments.

Acknowledgments

I would like to express sincere gratitude to my advisors, Professor Soon-Jo Chung and Professor Seth Hutchinson. I feel fortunate for having the opportunity to work with them and I would like to thank them for the guidance on my research and the financial support they have provided me during my Ph.D study at the University of Illinois at Urbana-Champaign (UIUC).

I am grateful to my doctoral committee members, Professor Andrew Alleyne and Professor Derek Hoiem for their suggestions on this dissertation. Dr. Michio Kise and David Johnson of John Deere provided me technical support for my research, and I thank their help and encouragement. Many thanks to Professor Ashwin Dani for the stimulating discussions that helped improve the quality of my research.

I would also like to thank my current and former lab members at Aerospace Robotics and Control Lab. Especially, thanks to my colleagues who helped me implement algorithms and conduct experiments for this dissertation. I also appreciate the friendship and encouragement my friends at UIUC and at the Korean church provided to me.

Finally, and most importantly, I would like to express my gratitude to my family. I appreciate my parents and my sister for their endless love and support. My wife, Hyunju, has been raising my two sons, Joshua and Julian, while patiently supporting me during my Ph.D study. I would not have been able to finish my degree without her and I cannot thank her enough.

Contents

List of Figures

Chapter 1

Introduction

Recent advances in navigation technologies using onboard local sensing modalities are allowing autonomous vehicles to execute missions in a range of diverse environments [1, 2, 3]. The problem for navigating a robot without a priori knowledge of the surroundings is solved by progressively constructing a map while estimating the location of the robot. The process is often known as simultaneous localization and mapping (SLAM) [4, 5, 6]

In this dissertation, we develop localization and mapping algorithms that primarily use monocular vision sensors. Solving a monocular-vision-based SLAM problem is particularly difficult because the depth of the scene cannot be estimated with a single view from a monocular camera. In particular, we seek methods of enhancing the accuracy of the vision-based localization and mapping results for two interesting applications: the riverine mapping with an unmanned aerial vehicle (UAV) and the autonomous mowing with a robotic mower. A goal in our research is to further expand the scope of future intelligence, surveillance, and reconnaissance (ISR) missions by developing a localization and mapping algorithm particularly for a riverine environment. Another goal is to allow autonomous robots to help us in our daily lives by investigating a vision-based localization and mapping scheme and solving the containment problem for autonomous lawn mowing.

1.1 Main Contributions

A primary contribution of this dissertation is to use a robot-centric mapping framework concurrently with a world-centric localization method to improve the accuracy of localization and mapping results. We exploit the differential equation of motion of the normalized pixel coordinates of each point feature in the robot body frame. This prevents the estimator from underestimating the error [7] in the feature estimate. Prior work of robot-centric SLAM [8, 9, 10] estimates both the initial robot body frame and the features with respect to the robot's current pose indirectly through a composition stage without properly considering the actual dynamics of the features. Another category of robot-centric work [7, 11, 12, 13, 14] estimates

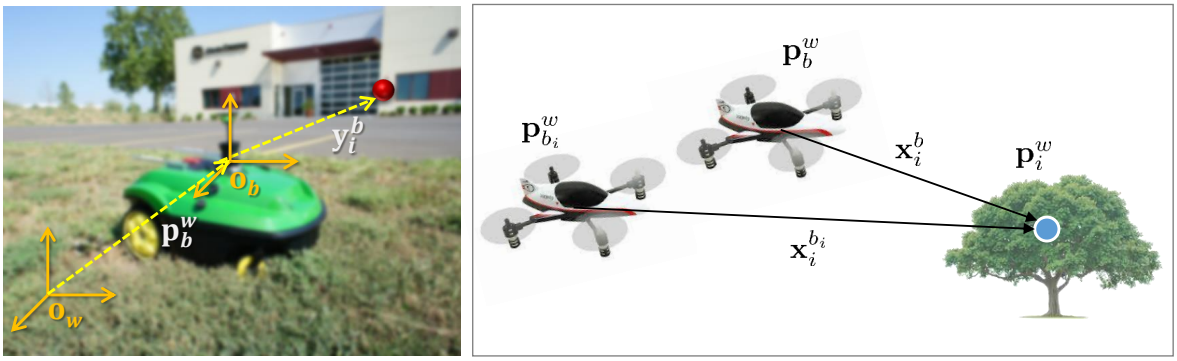


Figure 1.1: Illustration of the robot-centric mapping and multiple view measurements.

the features with respect to the robot or the camera by using a dynamic model with velocity and angular velocity information without estimating the pose of the robot. In contrast, we formulate a system model that exploits the robot-centric mapping framework for localization of the robot with respect to its initial pose. Using the robot-centric estimation framework, we report the first experimental results of containment detection with an omnidirectional camera for robotic mowing applications.

Another contribution of our work is to exploit a multiple-view geometry formulation with initial and current view projection of point features. For the problem of solving localization and mapping in riverine environments, we augment the measurements with the features that are extracted from real objects surrounding the river and their reflections. The correspondences of the features are used along with the attitude and altitude information that are available from our UAV. We demonstrate that the observability of the estimation system is improved by applying our proposed methods and show enhanced localization and mapping results in both numerical simulations and real-world experiments in a riverine environment. We report the first result that exploits the reflections of features in a riverine environment for localization and mapping.

Finally, we present a single camera based SLAM algorithm using convex optimization. We formulate a convex optimization problem with linear matrix inequality (LMI) constraints for the convergence of the state estimate to the true state. We reduce the mean-squared estimation error by minimizing an objective function that is derived using a contraction analysis approach. Using the convex optimization based estimator, we enhance the performance of our localization and robot-centric mapping algorithm.

1.2 Literature Review

The problem of navigating a vehicle in an unknown environment is addressed by SLAM. In the field of robotics, SLAM algorithms [4, 5, 6] have been studied extensively for over a decade using various types of sensors. In this dissertation we developed SLAM algorithms that primarily use a monocular camera and an IMU. We compare our work with the recent research in monocular camera based SLAM, riverine environment SLAM, observability of SLAM, and robot-centric estimation.

One of the most well known early work in monocular vision based SLAM [15] solves the localization and mapping problem by using Cartesian coordinates of features in the world reference frame and by sequentially updating the feature measurements from different locations. Instead of estimating the Cartesian coordinates of features in the world reference frame, some recent work [16, 17] defines the locations of the moving camera (anchor locations) where a set of point features is first observed. The point features are parameterized using these stationary anchor locations, the direction of each feature with respect to the world reference frame, and the inverse-distance between the feature and the anchor. Such methods reduce the 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 in the anchor-based methods [16, 17, 18] to alleviate the nonlinearity of the measurement model and to introduce new features to the map immediately. The inverse-depth method ameliorates the known problem of the EKF-based mono-vision SLAM, which often appears when the features are estimated in Cartesian coordinates [16]. We shall compare our localization and mapping approach against an anchored IDP method.

The computational issues of SLAM are addressed by keyframe-based optimization [19, 20, 21, 22] and sub-mapping [23]. In the scope of keyframe optimization-based research, parallel tracking and mapping (PTAM) [24] achieves real-time processing by separating the tracking of the camera and mapping of the environment into two parallel tasks. The UAV navigation [1, 25] and surveillance [26] problems are addressed based on the PTAM method.

A navigation algorithm particularly suited for riverine environments is presented in [27] with a graph-based state estimation framework [28] to estimate the vehicle's state with vision and limited GPS, while mapping the river with a self-supervised river detection algorithm and finding obstacles with a LIDAR sensor. In [29], a LIDAR and a stereo camera is used to demonstrate autonomous flight in riverine environments. A sub-mapping approach is applied in [30] to address the SLAM problem with an autonomous surface-craft that builds a map above and below the water's surface. A 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 [31], a surface-craft equipped with an acoustic modem is used to support the

localization of autonomous underwater vehicles.

Observations of known points of a vehicle through a mirror are used to estimate the 6-DOF pose of the camera with a maximum-likelihood estimator in [32]. 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 [33]. In [34], 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. We exploit geometrical constraints from reflection measurements in a natural environment for localization and mapping.

The observability problems of SLAM [35, 36, 37], and particularly, monocular vision-aided inertial navigation [1, 38, 39, 40] have been studied in the literature. In general, a-priori knowledge of the position of a set of features in the map are required for the system to be observable. The 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 [39]. An observability constrained EKF [37], 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 [39] to improve the consistency of the estimation. We formulate a world-centric localization and robot-centric mapping system model with multiple-view measurements and enhance the observability. We analyze the local weak observability of our system model and we present the enhancement in the degree of observability.

Robot-centric estimation, as opposed to world-centric SLAM, has been used with different meanings and purposes. Robot-centric SLAM for both localization and mapping is introduced in [8] and applied to monocular visual odometry in [9, 10]. The method defines the origin on the current robot frame and estimates 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 [7, 11] 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 to circumvent the observability issue in SLAM. Nonlinear observers are derived in [12, 13, 14] for feature tracking and inverse-depth estimation, which can also be viewed as robot-centric mapping with a monocular camera. We exploit the differential equation of motion of the normalized pixel coordinates in the body frame for each point feature in contrast with prior work using robot-centric SLAM, which estimates the robot's previous pose with respect to the current pose to indirectly acquire the location of each feature. We demonstrate that the degree of observability of the system is improved by applying the proposed robot-centric mapping strategy.

1.3 Dissertation Outline

This dissertation is organized as follows. In Chapter 2, we present our early work which influenced our research in vision based localization and mapping using a UAV [41]. We use a planar ground assumption and the epipolar geometry with a fast SLAM algorithm. In Chapter 3, we use multiple views from current and initial camera projection of features and reflections of features for localization and robot-centric mapping in a riverine environment using a UAV [42, 43]. We analyze the observability of our system model and present numerical simulation and real-world experimental results. In Chapter 4, we design a nonlinear observer with the pseudo measurement of feature's depth for robot-centric localization and mapping by using contraction analysis [44]. We present our scheme of separating the task for autonomous mowing into a teaching phase and a mowing phase. We show simulation results of the two phases and present preliminary experimental results of the teaching phase. In Chapter 5, we apply a hybrid extended Kalman smoother for batch processing of the localization and robot-centric mapping in the teaching phase [45]. We show experimental results in different environments for both teaching and mowing. In Chapter 6, we design a convex optimization based SLAM estimator for localization and robot-centric mapping. In Chapter 7, we summarize our work with concluding remarks.

Chapter 2

Monocular-Vision-Based Localization and Mapping for a UAV with a Planar Ground Assumption

2.1 Chapter Objective

This chapter presents a monocular vision based simultaneous localization and mapping (SLAM) algorithm with a particular focus on navigation of a unmanned aerial vehicle (UAV) operating in multiple environments. We exploit the so-called planar ground assumption, which holds for many environments. The proposed methods include image segmentation, epipolar geometry, and a variation of the FastSLAM algorithm in order to estimate a trajectory of a UAV while building a map by using only a monocular camera and an altitude sensor. Results of experimentation show the effectiveness of the proposed algorithms in an outdoor environment, an indoor corridor setting, and a river-like environment.

This chapter is organized as follows: Section 2.2 describes our attitude determination method using epipolar geometry and Kalman filtering; Section 2.3 explains our ground plane feature extraction and landmark ranging methods; Section 2.4 presents the new FastSLAM formulation with the converted measurement Kalman filter (CMKF); Section 2.5 illustrates the experimental setup and presents the results from our monocular vision SLAM algorithm in both indoor and outdoor environments; and Section 2.6 presents our conclusions.

2.2 Attitude Determination using Epipolar Geometry

The monocular vision algorithm outlined in this chapter presents an approach for localization and mapping in environments that hold the planar ground assumption. The operational steps are shown in Figure 2.2.

Past works [46, 47] solved the depth measurement problem by using planar features in an orthogonal indoor environment, with an altimeter measurement of the height of the camera above the ground, which constrained the geometry sufficiently to enable immediate landmark initialization. Navigation and mapping results were produced in an orthogonal indoor environment like in Figure 2.1 (a), but the work couldn't be used to reliably navigate in outdoor environments not satisfying the orthogonality condition as shown in

(a) Building corridor (b) Outdoor quad (c) Creek

Figure 2.1: Diverse environments that a UAV may need to navigate.

Figures 2.1 (b) and (c). This placed an immediate constraint on the utility of a UAV; any new environment that it explores requires a different navigation algorithm. Our work in this chapter extends this to an approach that is suitable for outdoor environments and others not satisfying the orthogonality assumption.

First, the path that a UAV can navigate through is segmented from the planar ground, and feature points on this segmented plane are extracted to be used as landmarks in the ranging algorithm. In parallel, features that are not necessarily from the ground are additionally used to initialize and update the attitude of the UAV, using epipolar geometry. Given the attitude information, the segmented plane, and altimeter data, the algorithm can then identify the position of landmarks on the ground plane. These are used iteratively within the FastSLAM framework [48] to localize the position of the UAV while mapping the structure of the path for the vehicle. By separating the attitude determination and ranging process, the algorithm can produce effective results in localization and mapping. The attitude determination can use large numbers of feature points in the whole scene in order to maintain good accuracy, while the ranging and mapping algorithm only needs points on the ground, leading to more structured and less cluttered maps built from visual data.

Our work here attempts to build an algorithm that allows for navigation in multiple environments, by not making excessive assumptions on the structure of the environment, and by showing results from a range of settings a UAV can fly through, namely an outdoor scene, an indoor hallway environment, and a river-like environment. Results of experiments show that our algorithm can be used successfully in all of these settings to produce an effective navigation estimate along with a coherent, uncluttered map by using a single camera and an altitude sensor. We focus on developing a lightweight algorithm that can represent the environment with a two-dimensional map while constructing a 6 DOF trajectories of the UAV, whereas many other vision based navigation research for UAVs are more concerned about precisely measuring the pose of the UAV for its attitude and tracking controls. Our algorithm can be effective when a UAV is sent out to explore an unknown environment and provide a map of an open path for a follower.

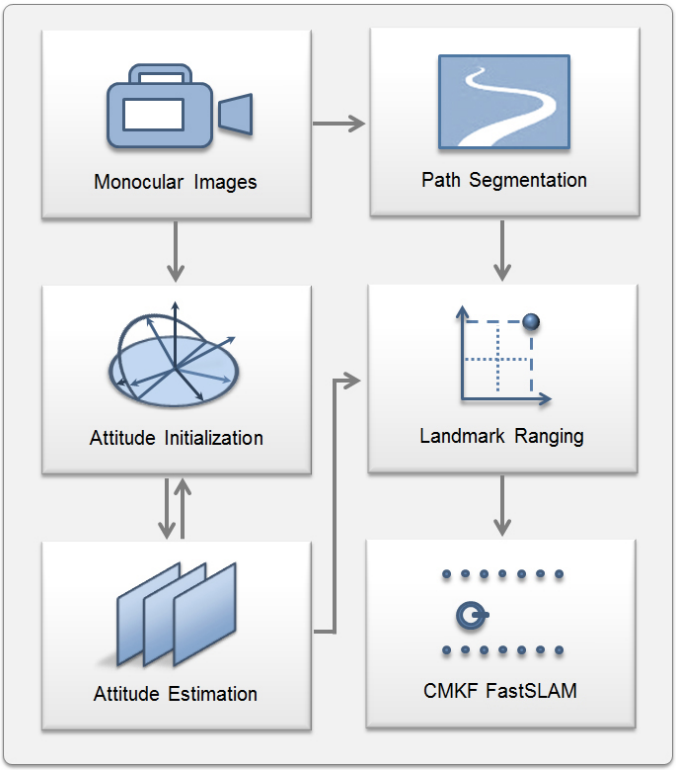


Figure 2.2: Operational steps of the proposed algorithm.

Our approach is composed of attitude estimation and landmark ranging process to estimate the pose of a UAV while building a map that can represent the environment. In this chapter, we exploit the well known epipolar geometry to perceive the attitude of a camera fixed on a UAV. With this attitude information, the range and bearing to landmarks on the ground plane can be determined as described in Section 2.3.

2.2.1 Focus of Expansion Based Attitude Measurement

Before updating the successive rotation of the camera, our monocular vision algorithm initializes the attitude of a camera with epipolar geometry, or more specifically, by using focus of expansion (FOE). Epipolar geometry is concerned with the projective geometry of different camera views. Figure 2.3 shows the epipolar geometry in forward motion of a single pinhole camera. A feature X is projected in two different image frames as $\mathbf{x}$ and $\mathbf{x}'$. The epipole $\mathbf{e}$ is the image of a camera center C' in the view of another camera that has C as its center. The line extending the feature $\mathbf{x}$ in the image plane and the epipole $\mathbf{e}$ is the epipolar line l. The epipolar lines $\mathbf{e}-\mathbf{x}$ and $\mathbf{e}'-\mathbf{x}'$ in the two images are coplanar and they define an epipolar plane CXC'. While typically used for stereo vision, a sequence of images from a monocular camera can also exploit these properties.

The fundamental matrix F maps points in an image frame to a line in another image frame [49]. It is computed from the corresponding points $\mathbf{x} = [u\ v\ 1]^T$ in the image frame as

$$\mathbf{x'}^T F \mathbf{x} = 0 \tag{2.1}$$

$$\begin{bmatrix} u'_1u_1 & u'_1v_1 & u'_1 & v'_1u_1 & v'_1v_1 & v'_1 & u_1 & v_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u'_nu_n & u'_nv_n & u'_n & v'_nu_n & v'_nv_n & v'_n & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0 \tag{2.2}$$

where there are n correspondences, and F_{ij} are elements of the fundamental matrix.

Given these geometric constraints, it is necessary to extract point matches to derive the fundamental matrix from which the attitude is extracted. The speeded up robust features (SURF) [50] algorithm is used in our system to obtain the corresponding feature points. The SURF algorithm is a real time feature detector that is robust to image transformation and variance in scale. The algorithm first finds unique keypoints in the images and represents its neighborhood by a descriptor vector. Then it matches the vectors by comparing their Euclidean distance. Figures 2.4 (a) and 2.5 (a) show the SURF keypoints. Figures 2.4 (b) and (c) and Figures 2.5 (b) and (c) show the point matching between an image from an earlier time-step and the current image.

Since the fundamental matrix has nine elements and the common scaling is not important, the fundamental matrix has 8 DOF. Therefore, the fundamental matrix can be found with eight point correspondences between two images [51]. When more points are available, the RANSAC algorithm [52] is applied to find the best fitting combination of points. The algorithm achieves this by using a random subset of the points and then iteratively taking the particular solution closest to the average, thereby recognizing and discarding outliers.

By definition, the epipolar line can be expressed as $l = F^T\mathbf{x}'$ and $(\mathbf{x'}^T F)\mathbf{e} = 0$. The epipole can be derived from the relation with the fundamental matrix, $F\mathbf{e} = 0$. When the camera is purely in translational forward motion, the epipoles $\mathbf{e}$ and $\mathbf{e}'$ coincide with each other, and in this particular case the epipole is

called as the FOE. When the horizontal and vertical coordinates of the epipole (e_u, e_v) in the image frame are close to being stationary, and thus, translate less than thresholds ϵ_1 and ϵ_2 as shown in Eq. (2.3), the epipole can be assumed as an FOE for the corresponding time interval.

$$\Delta e_u < \epsilon_1,\ \Delta e_v < \epsilon_2 \tag{2.3}$$

We assume that the UAV is translating forward during this interval. For a UAV that purely translates forward through a corridor, the FOE is identical to the vanishing point utilized in a previous work [47] for attitude measurement. However, while this previous work extracted straight lines from the environment to estimate the vanishing point, the current algorithm can estimate the FOE with a series of feature points regardless of the structure of the environment.

Figures 2.4 (d) and 2.5 (d) show the corresponding point matches (cyan), epipolar lines (orange), and the epipole (green) in an image. The point matches in a sequence of images are first found with the SURF algorithm. The distance between the corresponding points are computed, and the point matches that have much longer distance compared to other matches are excluded. When an FOE is detected, we start measuring the attitude with it, to initialize the camera attitude. The initial yaw ψ and pitch θ angles of the camera frame relative to its heading direction can be estimated from the FOE by

$$\psi = -\tan^{-1}\left(\frac{e_u}{f/s_u}\right) = -\tan^{-1}\left(\frac{e_u}{\alpha_x}\right) \tag{2.4}$$

$$\theta = \tan^{-1}\left(\frac{e_v \cos\psi}{f/s_v}\right) = \tan^{-1}\left(\frac{e_v \cos\psi}{\alpha_y}\right) \tag{2.5}$$

where f is the focal length, s_u and s_v are the pixel sizes in horizontal and vertical directions, and $\alpha_x = f/s_u$ and $\alpha_y = f/s_v$ are the focal length of the camera in pixels.

While updating the camera attitude through the process presented in Section 2.2.2, the measurement of the pitch angle θ of the camera is updated in the EKF with Eq. (2.5) every time an FOE is detected. We also take advantage of the fact that the roll angle ϕ can generally be assumed to be zero for the short intervals where the UAV translates forward, since the UAV is tilted in the flight direction to create a thrust toward that direction [53]. Without computationally heavy bundle adjustment techniques [54], this prevents drift in the roll angle ϕ and pitch angle θ, which is critical in our feature ranging method.

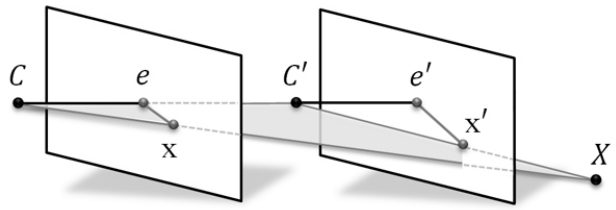


Figure 2.3: Epipolar geometry during pure translational motion.

2.2.2 Attitude Estimation

Once the initial attitude of the camera mounted is derived from the FOE, the attitude is updated from a sequence of frames by estimating the rotation from the essential matrix E. The essential matrix is a special version of the fundamental matrix. It is defined as $\hat{\mathbf{x}}'^T E\hat{\mathbf{x}} = 0$, where $\hat{\mathbf{x}} = K^{-1}\mathbf{x}$ and K is the camera calibration matrix [49]. The essential matrix can be applied to Eq. (2.1) as $\mathbf{x}'^T K^{-T} E K^{-1}\mathbf{x} = 0$. Therefore, if the intrinsic camera parameters for the camera calibration matrix are known, the essential matrix can be derived from the fundamental matrix as

$$E = K^T F K \tag{2.6}$$

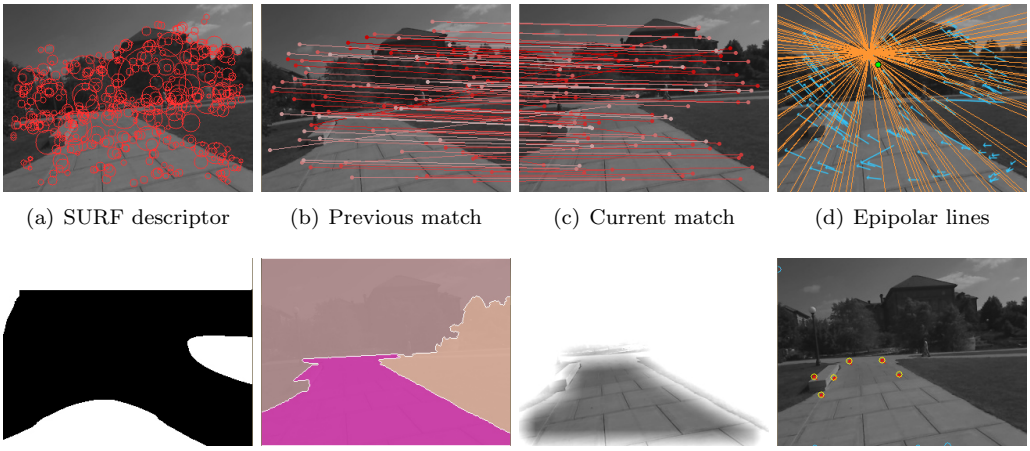
By definition, the essential matrix depends on the external camera parameters as $E = SR_{c_1}^{c_0}$, where S is a skew symmetric matrix that represents the translation with its elements, and $R_{c_1}^{c_0}$ is the rotation matrix representing the incremental rotation of the camera (see Figure 2.6). The matrix S can be decomposed as $kUZU^T$, where the matrix Z is also skew symmetric and U is the left singular vector matrix of E. Here, k is the mean of singular values from the essential matrix.

To represent the incremental rotation $R_{c_1}^{c_0}$ of the camera with a coordinate frame generally used for aerial vehicles, $R_{a_1}^{a_0}$ can be derived from Eq. (2.10). In order to obtain the rotation matrix $R_{c_1}^{c_0}$, the singular value decomposition (SVD) is applied to the essential matrix E, which is a singular matrix with rank 2.

$$\begin{aligned} E &= UDkWU^T R_{c_1}^{c_0} \\ &= UkDV^T \end{aligned} \tag{2.7}$$

Here, the matrix $Z = DW$, where $kD = \mathrm{diag}(k, k, 0)$ is the singular value matrix of E, and W is an orthogonal matrix. The skew symmetric matrix Z and the orthogonal matrix W are defined as

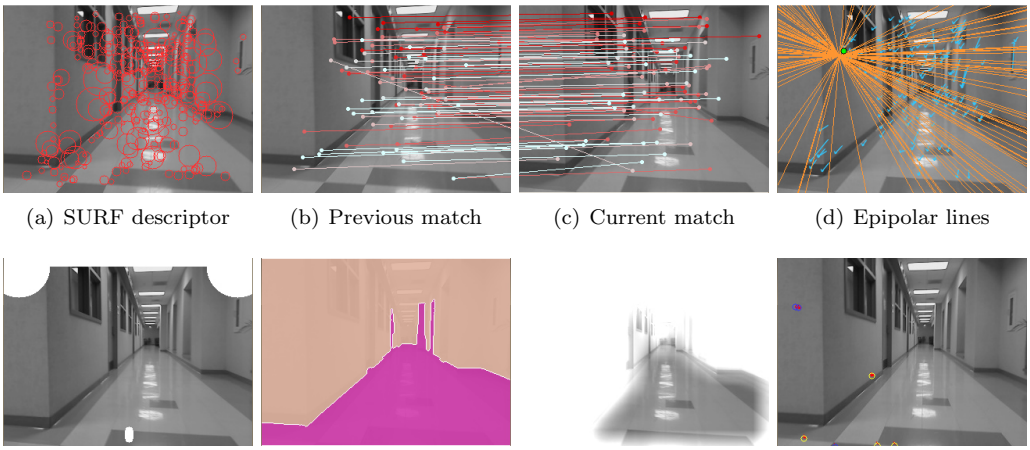
$$Z = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$



(a) SURF descriptor (b) Previous match (c) Current match (d) Epipolar lines

(e) Markers (f) Segments (g) Pavement (h) Features

Figure 2.4: Processing vision data collected in outdoor environments.



(a) SURF descriptor (b) Previous match (c) Current match (d) Epipolar lines

(e) Markers (f) Segments (g) Floor segment (h) Features

Figure 2.5: Processing vision data collected in indoor environments.

The SVD of the essential matrix is not unique because two of the singular values are equal. Thus, two sets of rotation matrices are derived as

$$R_{c_1}^{c_0} = UWV^T \text{ and } UW^TV^T \tag{2.9}$$

The actual rotation can be distinguished from its reflection by considering the determinant of $R_{c_1}^{c_0}$ and the amount of rotation.

Finally, the rotation of the camera as shown in Figure 2.6 can be expressed as

$$R_{a_1}^{a_0}(\Delta\psi, \Delta\theta, \Delta\phi) := R_{c_0}^{a_0} R_{c_1}^{c_0} R_{a_1}^{c_1} \tag{2.10}$$

where

$$R_{a_1}^{a_0}(\Delta\psi, \Delta\theta, \Delta\phi) = R_{z_c,\Delta\psi} R_{y_c,\Delta\theta} R_{x_c,\Delta\phi} \tag{2.11}$$

$$R_{c_0}^{a_0} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \; R_{a_1}^{c_1} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \tag{2.12}$$

Here, $R_{c_0}^{a_0}$ and $R_{a_1}^{c_1}$ shows the relationship between the two coordinate frames shown in Figure 2.6, and $c(\ )$ and $s(\ )$ are abbreviations of cos( ) and sin( ). The incremental roll $\Delta\phi$, pitch $\Delta\theta$, and yaw $\Delta\psi$ angles of the successive rotation of the camera are derived from the rotation matrix $R_{a_1}^{a_0}$.

The integrated roll ϕ, pitch θ, and yaw ψ angles of the camera frame relative to the inertial frame are defined in Figure 2.8 and are obtained by integrating the incremental angles with an EKF. The equations for the propagation step of the EKF is given by

$$\begin{pmatrix} \Theta_{n+1} \\ \bar{\omega}_{n+1} \end{pmatrix} = \begin{pmatrix} \Theta_n + T(\Delta\phi_n, \Delta\psi_n, \Delta\theta_n)\left(\bar{\omega}_n + \alpha_n \Delta t_n\right)\Delta t_n \\ \bar{\omega}_n + \alpha_n \Delta t_n \end{pmatrix} \tag{2.13}$$

where angular acceleration α is modeled as a zero mean Gaussian distribution. Here, $\Theta = [\phi \; \theta \; \psi]^T$ includes the roll, pitch, and yaw angles of the camera, $\bar{\omega}$ denotes the 3 DOF body rotation rate. The transformation matrix T that transforms the body rotation rate to the Euler angle rate in Eq. (2.13)is given by

$$T(\Delta\phi_n, \Delta\psi_n, \Delta\theta_n) = \begin{pmatrix} 1 & \sin\Delta\phi_n \tan\Delta\theta_n & \cos\Delta\phi_n \tan\Delta\theta_n \\ 0 & \cos\Delta\phi_n & -\sin\Delta\phi_n \\ 0 & \sin\Delta\phi_n \sec\Delta\theta_n & \cos\Delta\phi_n \sec\Delta\theta_n \end{pmatrix} \tag{2.14}$$

The measurement Eq. is given by

$$\begin{aligned} z_n &= \bar{\omega}_n \\ &= T^{-1}(\Delta\phi_n, \Delta\psi_n, \Delta\theta_n) \begin{pmatrix} \Delta\phi_n & \Delta\theta_n & \Delta\psi_n \end{pmatrix}^T / \Delta t_n \end{aligned} \tag{2.15}$$

where Δt_n denotes the time intervals for computing the essential matrix. Whenever the condition given in Eq. (2.3) is satisfied, the pitch angle θ of the UAV is measured from Eq. (2.5) and updated in the EKF. As mentioned in Section 2.2, the roll angle ϕ can generally be assumed to be zero for intervals where it translates forward.

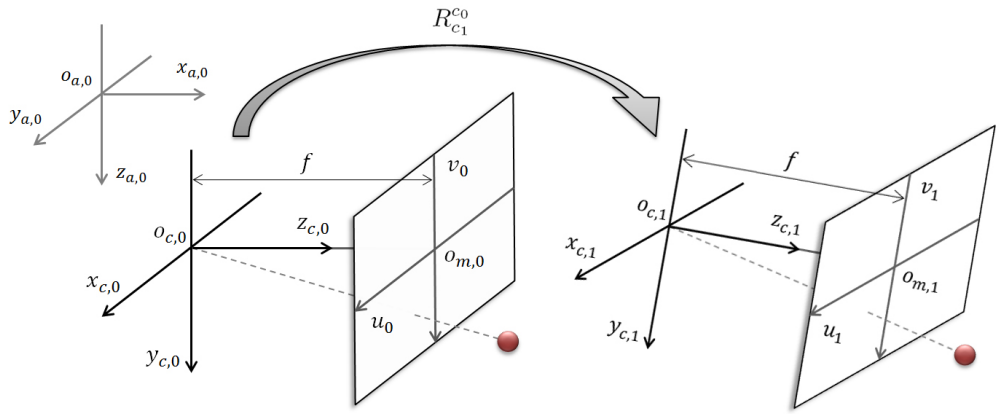


Figure 2.6: Rotation between a sequence of images.

2.3 Landmark Extraction and Ranging

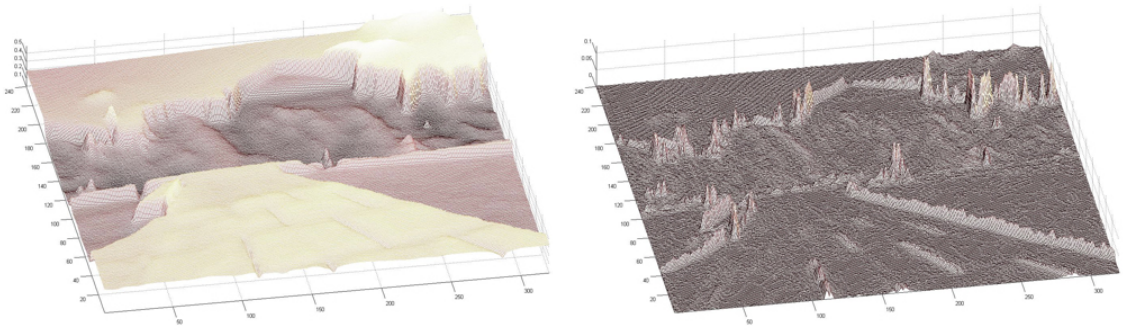
To determine the depth of landmarks satisfying the planar ground assumption, the range estimation method using rotation between the image frame, the camera frame, and the inertial frame is derived in this section. We explain how we utilize the structural commonalities of multiple environments.

2.3.1 Ground Plane Segmentation

To utilize features from the ground plane as landmarks, we first segment the ground plane region from its surroundings using a morphological segmentation method called the watershed transformation [55]. First, the gradient norm of the gray scale intensity image is acquired to locate dominant edges that are called ranges and relatively uniform surfaces that are called catchment basins. Figures 2.7 (a) and (b) show the gray scale intensity and the gradient norm of an image from the UIUC Engineering Quad shown in Figure 2.4.

Watershed markers are specified in the image, and the gradient norm image is then immersed starting from the watershed markers to select the ground plane. In our algorithm, locations and shapes of the watershed markers can be automatically updated by considering the intensity difference of regions in the image. For example, gray scale intensity of the ground and the rest of the scene are different as shown in Figure 2.7 (a). Figure 2.4 (e) shows a watershed marker being updated automatically, while Figure 2.5 (e) shows a watershed marker manually initialized in an image.

Watersheds are made in the image where catchment basins meet together. Each catchment basin is then associated with one of the markers. The catchment basins that meet together along a marker are considered as a single region. Finally, the ground plane is extracted by increasing the immersion level until only the



(a) Intensity image (b) Gradient norm image

Figure 2.7: Gray scale intensity and its gradient norm of an image from the UIUC Engineering Quad.

regions corresponding to each markers is left. Here, we take advantage of the fact that the ground region is on the bottom side of the image. Figures 2.4 (f), 2.4 (g), 2.5 (f), and 2.5 (g) show the segmentation results.

Once the ground plane is segmented, Shi and Tomasi's method [56] is used to extract feature points from the ground segment in the original image. These points are used as landmarks that represent the environment through the SLAM algorithm. The method searches for feature points by computing eigenvalues with the second order derivatives of the image. Features are selected as good landmarks if the smaller eigenvalue is larger than a threshold, meaning that the feature must have strong texture and large contrast in two directions. The algorithm then tracks the features via Lucas-Kanade optical flow technique [57], which estimates the velocity of image patches around a given set of feature points. In our algorithm, data association is assisted by tracking the features until they move out from the image. New features are then extracted to keep a certain number of features in the view and enable continuous localization. Figures 2.4 (h) and 2.5 (h) show the resulting features extracted and tracked from the planar ground.

2.3.2 Landmark Ranging

In a previous work [47], it has been shown that the depth of landmarks can be calculated from two-dimensional pixel coordinates of an image when the locations of landmarks are constrained on an orthogonal planar ground. In contrast to the previous work, we do not need to identify a vanishing point from parallel lines. Further, we acquire camera attitude through the method proposed in Section 2.2 and perform localization and mapping only with the coplanar landmarks.

Figure 2.8 shows the rotation between the camera coordinate frame and the inertial frame, which is used for measuring the depth of each landmark. The rotation between the camera and the inertial frame can be

described as

$$\mathcal{F}_c = \begin{pmatrix} x_c & y_c & z_c \end{pmatrix}^T \tag{2.16}$$

$$\begin{pmatrix} x_i & y_i & z_i \end{pmatrix}^T = R_c^r(\psi, \theta, \phi)\mathcal{F}_c \tag{2.17}$$

where $\mathcal{F}_c$ is the camera frame. The three-dimensional rotation matrix $R_c^r(\psi, \theta, \phi)$ represents the rotation between the camera frame and the inertial frame, where the incremental angles $\Delta\phi$, $\Delta\theta$, and $\Delta\psi$ in Eq. (2.11) are substituted with the Euler angles ψ, θ, and ϕ derived from Eq. (2.13).

The relation between the image frame and the camera frame $o_c x_c y_c z_c$ can be written as

$$\begin{aligned} u &= \frac{f}{s_u}\left(\frac{y_c}{x_c}\right) = \alpha_x \frac{y_c}{x_c} \\ v &= \frac{f}{s_v}\left(\frac{z_c}{x_c}\right) = \alpha_y \frac{z_c}{x_c} \end{aligned} \tag{2.18}$$

where u and v are the horizontal and vertical pixel coordinates of a landmark, f is the focal length, s_u and s_v are the pixel sizes in horizontal and vertical directions, and α_x and α_y are the focal lengths in pixels.

The coordinates of the landmark are described in the camera frame as

$$\begin{aligned} x_c &= \frac{\alpha_y}{v} z_c \\ y_c &= \frac{u}{\alpha_x} x_c = \frac{\alpha_y u}{\alpha_x v} z_c \end{aligned} \tag{2.19}$$

The landmarks expressed in the camera frame can then be derived in the inertial frame from

$$\begin{aligned} z_i &= h \\ &= R_{31}x_c + R_{32}y_c + R_{33}z_c \\ &= -s\theta x_c + c\theta s\phi y_c + c\theta c\phi z_c \\ &= \left(-s\theta \frac{\alpha_y}{v} + c\theta s\phi \frac{\alpha_y u}{\alpha_x v} + c\theta c\phi\right) z_c \end{aligned} \tag{2.20}$$

where h is the altitude of the camera, which can be obtained from the onboard altimeter on the UAV, and R_{ij} are the elements of $R_c^r(\psi, \theta, \phi)$.

The longitudinal distance x and the transverse distance y to a landmark are found from the rotation

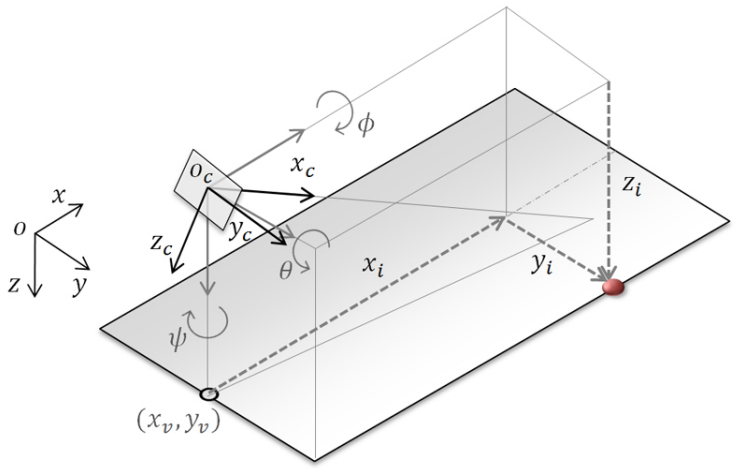


Figure 2.8: Rotation of the camera frame from the inertial frame.

shown in Eq. (2.17) as follows:

$$\begin{aligned}
x_i &= R_{11}x_c + R_{12}y_c + R_{13}z_c \\
&= \left(\frac{R_{11}\alpha_x\alpha_y + R_{12}\alpha_y u + R_{13}\alpha_x v}{R_{31}\alpha_x\alpha_y + R_{32}\alpha_y u + R_{33}\alpha_x v}\right) h \\
&= \left(\frac{c\psi c\theta\alpha_x\alpha_y + (-s\psi c\phi + c\psi s\theta s\phi)\alpha_y u + (s\psi s\phi + c\psi s\theta c\phi)\alpha_x v}{-s\theta\alpha_x\alpha_y + c\theta s\phi\alpha_y u + c\theta c\phi\alpha_x v}\right) h \\
y_i &= R_{21}x_c + R_{22}y_c + R_{23}z_c \\
&= \left(\frac{R_{21}\alpha_x\alpha_y + R_{22}\alpha_y u + R_{23}\alpha_x v}{R_{31}\alpha_x\alpha_y + R_{32}\alpha_y u + R_{33}\alpha_x v}\right) h \\
&= \left(\frac{s\psi c\theta\alpha_x\alpha_y + (c\psi c\phi + s\psi s\theta s\phi)\alpha_y u + (-c\psi s\phi + s\psi s\theta c\phi)\alpha_x v}{-s\theta\alpha_x\alpha_y + c\theta s\phi\alpha_y u + c\theta c\phi\alpha_x v}\right) h
\end{aligned} \tag{2.21}$$

2.4 CMKF Based FastSLAM

The SLAM algorithm utilizes the attitude and ranging estimates from Sections 2.2 and 2.3 to build the environment map while localizing the vehicle. It is known that conventional SLAM algorithms based only on EKF has problems in environments with large number of landmarks. Therefore, we apply the FastSLAM algorithm [48] for localization and mapping. The algorithm decomposes the vehicle pose posterior into a set of conditionally independent estimates, one corresponding to each landmark in the environment.

The FastSLAM algorithm uses a particle filter which represents the vehicle pose posterior. Each particle represents a vehicle pose, and each landmark is represented independently with the mean and covariance from an EKF. At each time-step, the set of particles are sampled based on the probabilistic motion model, and each landmark observed has its corresponding EKF updated based on the measurement. Finally, the

particles are resampled based on their respective probabilistic weights. Particles that are highly consistent with landmark measurements are redrawn, while inconsistent particles are sampled out in this process.

The probabilistic motion model we use for the particle filter is a standard kinematic vehicle model [4] with a state $\mathbf{x}_v = [x_v \ y_v \ \psi]^T$ that represent the 2D location of the UAV and its heading direction in the world frame as shown below.

$$\mathbf{x}_{v,k+1} = \begin{pmatrix} x_{v,k+1} \\ y_{v,k+1} \\ \psi_{k+1} \end{pmatrix} = \begin{pmatrix} x_{v,k} + V_{1,k}\Delta t_k \cos\psi_k \\ y_{v,k} + V_{1,k}\Delta t_k \sin\psi_k \\ \psi_k + V_{2,k}\Delta t_k \end{pmatrix} \tag{2.22}$$

Here, V_1 and V_2 are the forward and angular velocities and Δt_k is the time intervals for processing the Fast-SLAM algorithm. The motion propagates toward a heading direction ψ, which is estimated from Eq. (2.13).

The measurement equation related with the UAV's states and the i-th landmark at each time-step is given by

$$\mathbf{h}\left(\mathbf{x}_{i,k}\right) = \begin{pmatrix} x_{i,k} \\ y_{i,k} \end{pmatrix} = \begin{pmatrix} x_{w_i,k} - x_{v,k} \\ y_{w_i,k} - y_{v,k} \end{pmatrix} \tag{2.23}$$

where x_i and y_i denote the i-th landmark's location derived in Eq. (2.21), and x_{w_i} and y_{w_i} are their location in the inertial frame.

The range r_i and bearing ϑ_i to a landmark can also be derived from Eq. (2.21) as

$$\begin{aligned} r_i &= \sqrt{{x_i}^2 + {y_i}^2} \\ \vartheta_i &= \tan^{-1}\left(\frac{y_i}{x_i}\right) - \psi \end{aligned} \tag{2.24}$$

The measurement equation in polar coordinates is given by

$$\mathbf{h}\left(\mathbf{x}_{i,k}\right) = \begin{pmatrix} r_{i,k} \\ \vartheta_{i,k} \end{pmatrix} = \begin{pmatrix} \sqrt{\left(x_{w_i,k} - x_{v,k}\right)^2 + \left(y_{w_i,k} - y_{v,k}\right)^2} \\ \tan^{-1}\left(\frac{\left(y_{w_i,k} - y_{v,k}\right)}{\left(x_{w_i,k} - x_{v,k}\right)}\right) - \psi_k \end{pmatrix} \tag{2.25}$$

Either $(x_{w_i,k}, y_{w_i,k})$ or (r_i, ϑ_i) can be used by modeling the measurement noise as a zero-mean Gaussian distribution in the chosen coordinate system. The polar coordinate representation can have a more relevant approximation of the noise characteristic since monocular vision is known to possess greater uncertainty in range than in bearing. However, the polar coordinate model has nonlinear terms in the equation, as shown in Eq. (2.25), and it requires a linearization step to estimate the location of each map feature. If an EKF is used to approximate the nonlinear terms, a Jacobian matrix has to be updated for each landmark and the

performance of the estimator will depend on the accuracy of the linearization.

In order to avoid the linearization error of the range and bearing measurements while ensuring a better approximation of the noise characteristics, we adopt the converted measurement Kalman filter (CMKF) [58]. We change the Cartesian measurements to polar coordinates, and then after applying the CMKF, we convert the results back to Cartesian coordinates. The CMKF considers the correlated error in the Cartesian components, which appears when polar coordinates are converted to Cartesian coordinates. The converted measurement covariance is updated consistently at each time-step and a nearly optimal estimator gain is derived from the correct covariance. As a result, it produces a smaller estimation error than the EKF when it estimates the state of each landmark, especially with a long range. In a recent study [59], it was also shown that the CMKF is more robust to inconsistent measurements than an EKF.

The average true bias of the converted measurements used in CMKF is

$$\tilde{\mu}_{i,k} = \begin{pmatrix} r_{i,k}\cos\left(\vartheta_{i,k}+\psi_k\right)\left(e^{-\sigma_\vartheta^2} - e^{-\sigma_\vartheta^2/2}\right) \\ r_{i,k}\sin\left(\vartheta_{i,k}+\psi_k\right)\left(e^{-\sigma_\vartheta^2} - e^{-\sigma_\vartheta^2/2}\right) \end{pmatrix} \tag{2.26}$$

where ψ is the yaw angle of the camera derived from Eq. (2.13), and σ_ϑ is the standard deviation of the vision based bearing measurement. We used $\sigma_\vartheta = 1.5^\circ$ for our experiments in Section 2.5.

The measurement model for our feature mapping algorithm is derived as

$$\mathbf{h}\left(\mathbf{x}_{i,k}\right) = \begin{pmatrix} x_{i,k} \\ y_{i,k} \end{pmatrix} = \begin{pmatrix} r_{i,k}\cos\left(\vartheta_{i,k}+\psi_k\right) \\ r_{i,k}\sin\left(\vartheta_{i,k}+\psi_k\right) \end{pmatrix} - \tilde{\mu}_{i,k} \tag{2.27}$$

The covariance of the noise term represented in cartesian coordinates is called the converted measurement covariance $\mathcal{R}_k$, which is given by [58] as

$$\begin{aligned}
\mathcal{R}_{11,k} &= r_{i,k}^2 e^{-\sigma_\vartheta^2}\left[\cos^2\left(\vartheta_{i,k}+\psi_k\right)\left(\cosh 2\sigma_\vartheta^2 - \cosh\sigma_\vartheta^2\right) + \sin^2\left(\vartheta_{i,k}+\psi_k\right)\right. \\
&\quad \left.\left(\sinh 2\sigma_\vartheta^2 - \sinh\sigma_\vartheta^2\right)\right] + \sigma_r^2 e^{-2\sigma_\vartheta^2}\left[\cos^2\left(\vartheta_{i,k}+\psi_k\right)\left(2\cosh 2\sigma_\vartheta^2 - \cosh\sigma_\vartheta^2\right)\right. \\
&\quad \left.+ \sin^2\left(\vartheta_{i,k}+\psi_k\right)\left(2\sinh 2\sigma_\vartheta^2 - \sinh\sigma_\vartheta^2\right)\right] \\
\mathcal{R}_{12,k} &= \sin\left(\vartheta_{i,k}+\psi_k\right)\cos\left(\vartheta_{i,k}+\psi_k\right) e^{-4\sigma_\vartheta^2}\left[\sigma_r^2 + \left(r_{i,k}^2+\sigma_r^2\right)\left(1 - e^{\sigma_\vartheta^2}\right)\right] \\
\mathcal{R}_{22,k} &= r_{i,k}^2 e^{-\sigma_\vartheta^2}\left[\sin^2\left(\vartheta_{i,k}+\psi_k\right)\left(\cosh 2\sigma_\vartheta^2 - \cosh\sigma_\vartheta^2\right) + \cos^2\left(\vartheta_{i,k}+\psi_k\right)\right. \\
&\quad \left.\left(\sinh 2\sigma_\vartheta^2 - \sinh\sigma_\vartheta^2\right)\right] + \sigma_r^2 e^{-2\sigma_\vartheta^2}\left[\sin^2\left(\vartheta_{i,k}+\psi_k\right)\left(2\cosh 2\sigma_\vartheta^2 - \cosh\sigma_\vartheta^2\right)\right. \\
&\quad \left.+ \cos^2\left(\vartheta_{i,k}+\psi_k\right)\left(2\sinh 2\sigma_\vartheta^2 - \sinh\sigma_\vartheta^2\right)\right]
\end{aligned} \tag{2.28}$$

Figure 2.9: The UAV used for data acquisition.

where $\mathcal{R}_{ij,k}$ denotes the elements of the converted measurement covariance $\mathcal{R}_k$ at time-step k, and σ_r is the standard deviation of the vision based range measurement. When the CMKF computes the Kalman gain to estimate the location of a map feature, the average true covariance is added in the predicted measurement covariance as

$$\mathcal{K}_k = \bar{\Sigma}_k C_k^T \left(C_k \bar{\Sigma}_k C_k^T + \mathcal{R}_k \right)^{-1} \tag{2.29}$$

where $\mathcal{K}_k$ is the Kalman gain, C_k is the measurement matrix, and $\bar{\Sigma}_k$ is the covariance of the states' prediction.

By performing the series of conversions between measurements in polar and Cartesian coordinates and applying the CMKF to estimate the location of each feature in FastSLAM, we can avoid dealing with nonlinearities and systematically model the measurement noise characteristic.

2.5 Experimental Results and Discussions

A number of experiments were performed in different environments to gauge the effectiveness and robustness of the algorithm by only using a monocular camera and an onboard ultrasonic altimeter of the UAV shown in Figure 2.9. Figure 2.10 shows the results of the navigation experiment conducted at the UIUC Engineering Quad (see Figure 2.1 (a)), which is approximately 60 m $\times$ 60 m. Compared to methods that require GPS and IMU, we only need to know the height of the camera. The initial attitude of the UAV was determined from the image coordinates of the FOE as described in Section 2.2.1 while flying forward for a short period. Subsequent rotations were measured by accumulating the rotations between sets of images as shown in Section 2.2.2. Drifting in the roll and pitch angles was prevented by initializing them every time an FOE was detected. Here, we used 40 pixels for ϵ_1 and ϵ_2 in Eq. (2.3). Preventing the drift in these two measurements is critical since it can be detrimental to the ranging method, while the yaw ψ drift only

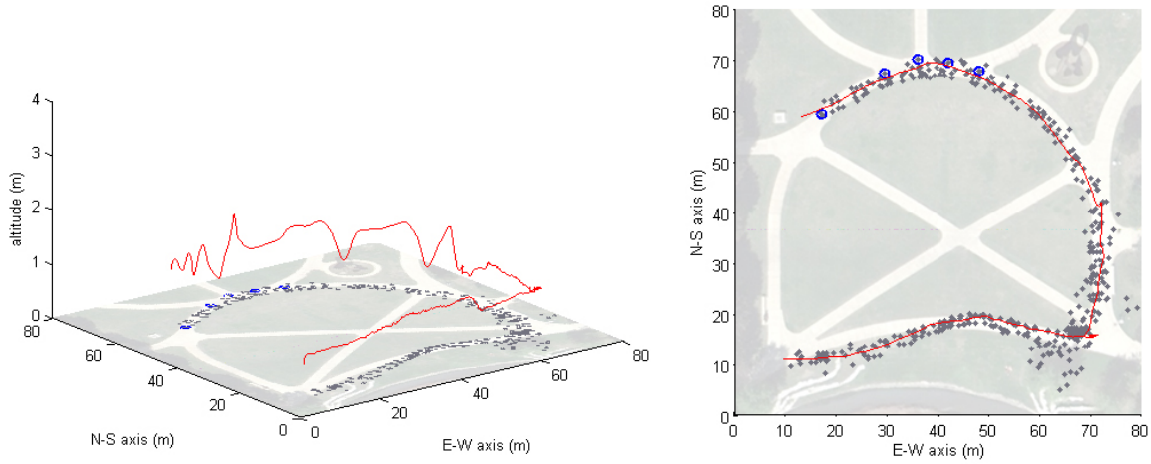


Figure 2.10: Localization and mapping results from the Engineering Quad.

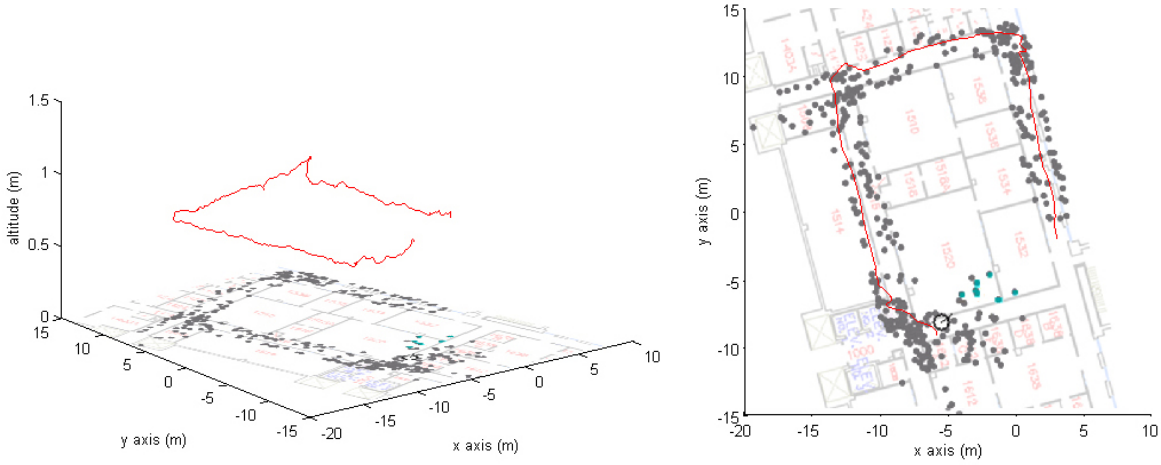


Figure 2.11: Localization and mapping results from inside the building of the Beckman Institute.

manifests as a more minor drift in the trajectory, as shown in Figure 2.10.

The time-varying height of the UAV was measured with an onboard altimeter and synchronized with each image. Landmarks shown on the map with gray dots primarily represent the footpath at the Engineering Quad, composed of features from the ground plane segmentation step. The three-dimensional trajectory of the UAV was reproduced by using ground features and the height of the UAV measured at each time-step. There were barely any distinguishable objects around the trajectory, but the algorithm was still able to represent the path by extracting point features from the pavement. This was possible because there were contrast between the pavement and the lawn in the quad and the vision sensor was able to detect the texture of the pavement. This is an advantage of the vision algorithms, and laser range finder based

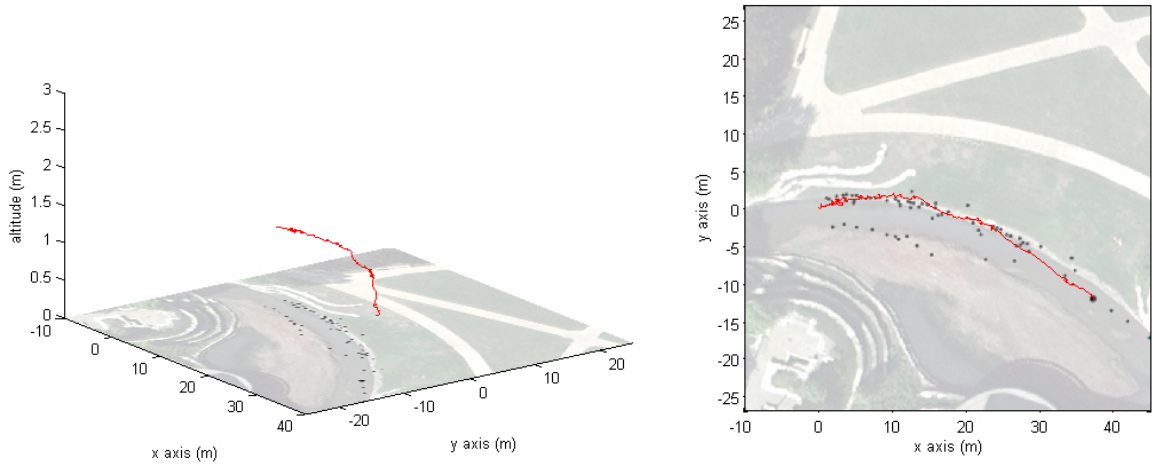


Figure 2.12: Localization and mapping results from the Boneyard Creek.

methods would not work for environments like in Figure 2.1 (b) and Figure 2.10. The results suggest that the algorithm is able to perform localization and mapping without relying on additional sensors for camera attitude measurement.

Figure 2.10 was generated by matching a set of map features at the beginning of the trajectory (marked with blue circles in Figure 2.10) in the SLAM result and the Google satellite image. The SLAM result was overlaid on the satellite image by computing the overlay that minimized the overall position error of the selected features. The attitude measurement was less accurate in the later part of the trajectory; the SLAM result is more deviated from the pavement in the southern part of the Engineering Quad. Nevertheless, the mapping results clearly show the shape and outline of the path, and the overlay with the satellite image shows the effectiveness of the algorithm.

Figure 2.11 shows the localization and mapping results from the inside of the building of the UIUC Beckman Institute (see Figure 2.1 (b)). Although feature points were sparse in this indoor environment, the feature tracking algorithm performed robustly. Given the lack of difference in intensity between the floor and the wall, the pixel position of the watershed algorithm markers was manually initialized (see Figure 2.5 (e)) to segment the floor in the corridor. The middle part of the floor segment in the images was automatically removed to avoid tracking erroneous dynamic feature points caused by specular reflections. During the period where the UAV turned in the corridor, the attitude measurement was updated more frequently to maintain enough point matches and estimate the rotation properly. The mapping results overlaid on the building floor plan show that the algorithm can perform well in indoor environments as well.

Figure 2.12 shows the results of the localization and mapping algorithm in the UIUC Boneyard Creek (see Figure 2.1 (c)). For this experiment, the camera was held approximately at a constant height, but the

altitude data was not available. This introduced scale ambiguity in the map. Further, the segmented plane was not perfectly planar due to intrusion into the regions surrounding the water surface. Nonetheless, the landmarks extracted around the water surface were able to represent the outline of the creek. Note that the landmarks predominantly consist of features from the river's edge (shown on the map as a series of curved points), but a number of these include points from the reflections on the surface of the river and the surroundings. Such measurements need to be carefully considered in future research since they can induce errors into the navigation solution. Nevertheless, the produced map illustrates the outline of the creek, and the results of experiments show that our algorithm can be applied to navigation in river-like environments as well.

2.6 Conclusions

In this chapter, we presented a monocular-vision-based algorithm with a particular focus on navigation of a UAV in multiple environments. Our method exploited the planar ground assumption in multiple environments. In the presence of coplanar features and the knowledge of the camera height, we have shown that the range and bearing to the landmarks on the ground plane can be measured instantaneously. We estimated the attitude of the UAV separately by exploiting the epipolar geometry with multiple features that are not required to be included in the FastSLAM estimation state vector. Localization and mapping was performed by applying the CMKF based FastSLAM algorithm to the attitude and range estimation. The results were obtained in an indoor environment from the Beckman Institute at UIUC, and in outdoor environments such as the Engineering Quad and the Boneyard Creek at UIUC. It was demonstrated that with our algorithms, a monocular based system is able to perform visual SLAM and generate a map of the ground plane and obtain a pose estimate of the UAV.

Chapter 3

Vision-Based Localization and Robot-Centric Mapping in Riverine Environments

3.1 Chapter Objective

In this chapter, we present a vision-based localization and mapping algorithm developed for an unmanned aerial vehicle (UAV) which can operate in a riverine environment. Our algorithm estimates the 3D 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 (IMU), a magnetometer, an altimeter, and an onboard computer. To our knowledge, we report the first result that exploits the reflections of features in a riverine environment for localization and mapping.

This chapter is organized as follows. In Section 3.2, 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 3.3, we formulate an extended Kalman filter (EKF) estimator for UAV localization and point feature mapping. In Section 3.4, we analyze the observability of our estimation system under various conditions and show the advantage of our method. In Section 3.5, we validate the performance of our algorithm with numerical simulation results. In Section 3.6, we show experimental results of our monocular vision-based localization and mapping algorithm at Crystal Lake Park in Urbana, Illinois. In Section 3.7, we summarize our work with concluding remarks.

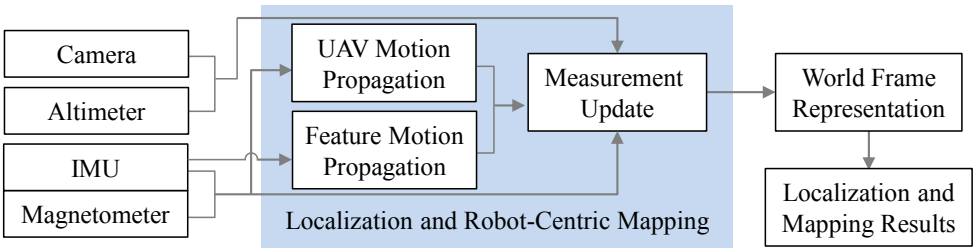


Figure 3.1: The block diagram of our riverine localization and mapping system.

3.2 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.

3.2.1 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 3.1 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 downwards along the gravity vector (see Figure 3.3). We set the origin of the UAV body frame on the center of the IMU which is mounted on the UAV and 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 3.2 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.

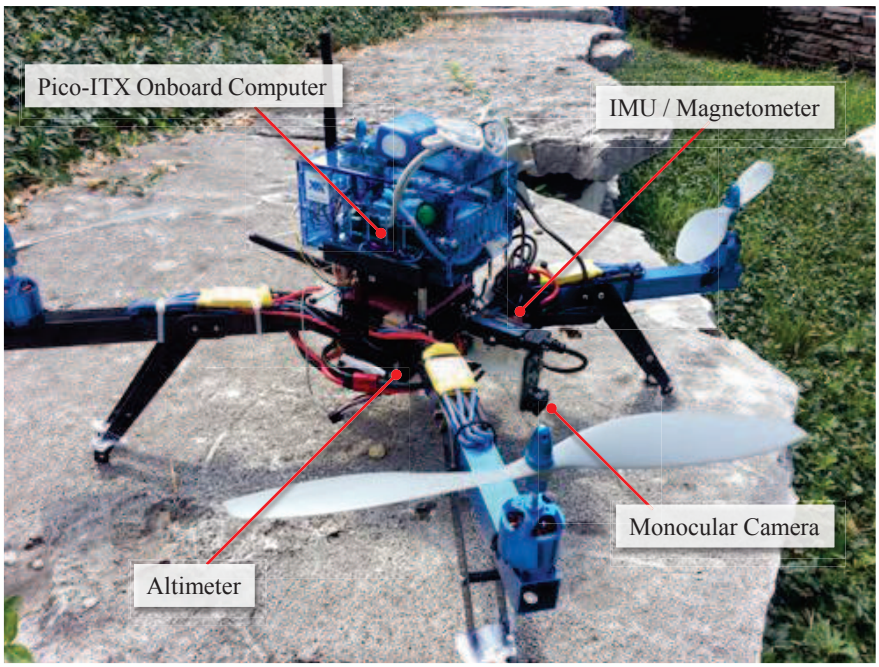


Figure 3.2: Our quadcopter is equipped with a lightweight monocular camera, an IMU, a magnetometer, an altimeter, and a compact Pico-ITX onboard computer.

3.2.2 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.

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 i-th 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 [38].

The dynamic model for our estimation system is given by

$$\frac{d}{dt}\begin{pmatrix} \mathbf{p}_b^w \\ \mathbf{v}^b \\ \mathbf{b}_a^b \\ \mathbf{q}_b^w \\ \mathbf{b}_g^b \\ \mathbf{x}_1^b \\ \vdots \\ \mathbf{x}_n^b \end{pmatrix} = \begin{pmatrix} R(\mathbf{q}_b^w)\mathbf{v}^b \\ -[\boldsymbol{\omega}^b]_\times \mathbf{v}^b + \mathbf{a}^b + R^T(\mathbf{q}_b^w)\mathbf{g}^w \\ \mathbf{0} \\ \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}^b)\mathbf{q}_b^w \\ \mathbf{0} \\ \mathbf{f}(\mathbf{x}_1^b,\, \mathbf{v}^b,\, \boldsymbol{\omega}^b) \\ \vdots \\ \mathbf{f}(\mathbf{x}_n^b,\, \mathbf{v}^b,\, \boldsymbol{\omega}^b) \end{pmatrix} \tag{3.1}$$

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 i-th 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

$$\Omega(\boldsymbol{\omega}^b) \equiv \begin{pmatrix} -[\boldsymbol{\omega}^b]_\times & \boldsymbol{\omega}^b \\ -\left(\boldsymbol{\omega}^b\right)^T & 0 \end{pmatrix} \tag{3.2}$$

The motion model for each feature $\mathbf{x}_i^b$ in Eq. (3.1) requires the velocity of the UAV in the UAV body frame of reference as shall be shown in Eq. (3.4). 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.

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 3.4.

The position of each point feature is first estimated in the UAV body frame. The dynamics of the i-th feature in Cartesian coordinates is given in [60] as follows:

$$\frac{d}{dt}\mathbf{p}_i^b = -[\boldsymbol{\omega}^b]_\times \mathbf{p}_i^b - \mathbf{v}^b \tag{3.3}$$

where $\mathbf{p}_i^b$ is the location of the i-th 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 [14], 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 i-th feature referenced with respect to the UAV body frame from Eq. (3.3) as

$$\frac{d}{dt}\begin{pmatrix} h_{1,i}^b \\ h_{2,i}^b \\ \rho_i^b \end{pmatrix} = \begin{pmatrix} \left(-v_2 + h_{1,i}^b v_1\right)\rho_i + h_{2,i}^b\omega_1 - \left(1 + \left(h_{1,i}^b\right)^2\right)\omega_3 + h_{1,i}^b h_{2,i}^b\omega_2 \\ \left(-v_3 + h_{2,i}^b v_1\right)\rho_i - h_{1,i}^b\omega_1 + \left(1 + \left(h_{2,i}^b\right)^2\right)\omega_2 - h_{1,i}^b h_{2,i}^b\omega_3 \\ \left(-\omega_3 h_{1,i}^b + \omega_2 h_{2,i}^b\right)\rho_i^b + v_1\left(\rho_i^b\right)^2 \end{pmatrix} \tag{3.4}$$

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

The model in Eq. (3.4) is similar to the one presented in [14], but the model is augmented with the UAV localization part. We construct the motion model in Eq. (3.1) for the localization and mapping by combining the dynamic model of the UAV and the vision motion model given by Eq. (3.4). The estimator that we will present in Section 3.3 exploits the motion model given by Eqs. (3.1) and (3.4).

3.2.3 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.

Projected Measurements of Features

We compute the normalized pixel coordinates $\mathbf{h}_i^c$ of the i-th 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 [60]. 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 camera is negligible. Here, $\mathbf{q}_c^b$ is the orientation quaternion of the camera with respect to the UAV body frame,

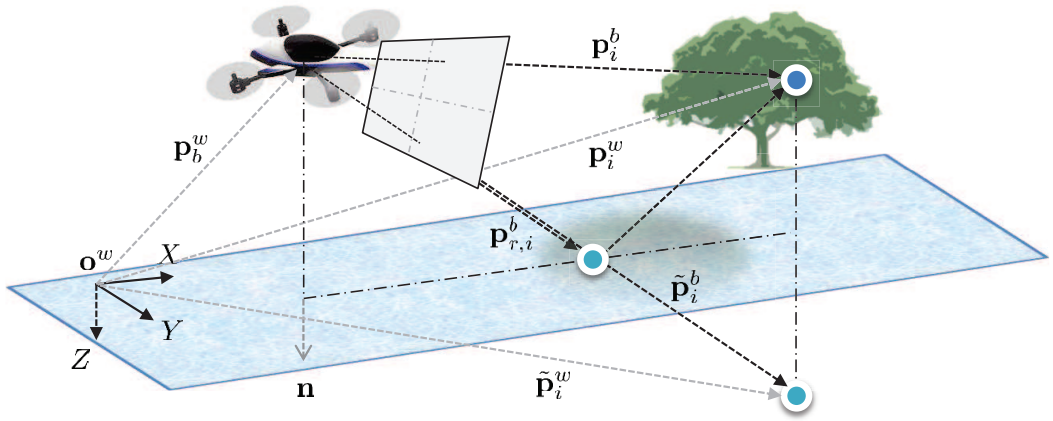


Figure 3.3: Illustration of the vision measurements of a real object and its reflection. The vector $\mathbf{p}_i^w$ of a point feature from a real object in the world frame is symmetric to the vector $\tilde{\mathbf{p}}_i^w$ of its mirrored point with respect to the river surface (X-Y plane). The measurement of the reflection is a camera projection of the vector $\tilde{\mathbf{p}}_i^b$.

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 i-th 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 i-th 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 i-th 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

$$\mathbf{p}_i^{bi} = R^T(\mathbf{q}_{bi}^w)\,(\mathbf{p}_b^w - \mathbf{p}_{bi}^w) + R^T(\mathbf{q}_{bi}^w)R(\mathbf{q}_b^w)\mathbf{p}_i^b \tag{3.5}$$

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 i-th feature in the measurement vector and exploit multiple views as shall be seen in Section 3.3.2. The initial normalized coordinates $\mathbf{h}_i^{bi}$ define a constant vector which is identical to the normalized coordinates $\mathbf{h}_i^b$ of the i-th feature upon its first observation.

Measurements of Reflections of Features

Reflection of the surrounding environment is an important aspect of riverine environments. We express the reflection of the i-th 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 define a mirrored point as $\tilde{\mathbf{p}}_i^w = S\mathbf{p}_i^w \in \mathbb{R}^3$, where $S = I - 2\mathbf{n}\mathbf{n}^T \in \mathbb{R}^{3\times 3}$ is the householder transformation matrix that describes a reflection about a plane and $\mathbf{n} = (0,\, 0,\, 1)^T$ [32]. 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 3.3.

The measurement of the reflection can be expressed with 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 $\tilde{\mathbf{p}}_i^w = S\mathbf{p}_i^w = S\left(\mathbf{p}_b^w + R(\mathbf{q}_b^w)\mathbf{p}_i^b\right)$. The position of the mirrored point with respect to the UAV body frame is given by [32]

$$\tilde{\mathbf{p}}_i^b = R^T(\mathbf{q}_b^w)\left(S\left(\mathbf{p}_b^w + R(\mathbf{q}_b^w)\mathbf{p}_i^b\right) - \mathbf{p}_b^w\right) \tag{3.6}$$

Figure 3.3 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 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 3.4 for the observability analysis).

Vision-Data Processing and Reflection Matching

We implement 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) [27]. The algorithm discards false matches by using the UAV's attitude information. The pseudo-code 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 [61], which computes the minimum eigenvalue of the auto-correlation 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

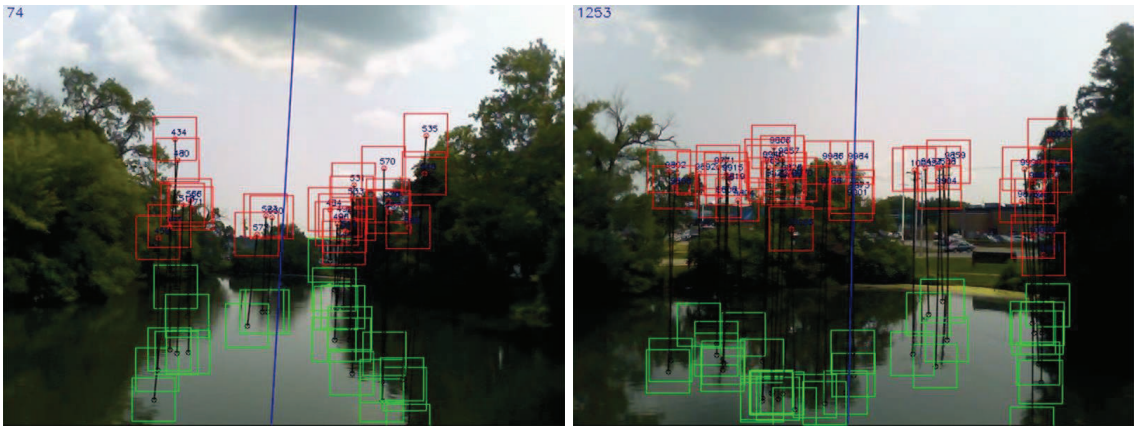


Figure 3.4: The results of reflection feature detection with the reflection matching Algorithm 1. The real objects (red boxes), the corresponding reflections (green boxes), the matching slope θ_i (black lines), and the reference slope θ_0 (blue line in the middle of the image) are shown.

patches by

$$M\left(\tilde{x}^m,\, \tilde{y}^m\right) = \sum_{x'=1}^{50} \sum_{y'=1}^{50} \left(T(x',\, y') - \bar{T}\right) \left(T_0(\tilde{x}^m + x',\, \tilde{y}^m + y') - \bar{T}_0\right) \tag{3.7}$$

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 [62]

$$N\left(\tilde{x}^m,\, \tilde{y}^m\right) = M\left(\tilde{x}^m,\, \tilde{y}^m\right) \left(\sum_{x'=1}^{50} \sum_{y'=1}^{50} \left(T(x',\, y') - \bar{T}\right)^2 \cdot \sum_{x'=1}^{50} \sum_{y'=1}^{50} \left(T_0(\tilde{x}^m + x',\, \tilde{y}^m + y') - \bar{T}_0\right)^2 \right)^{-1/2} \tag{3.8}$$

Algorithm 1 then finds the corresponding location in the source image that has the highest NCC. The methods proposed in [63, 64] 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

$$\theta_0 = \operatorname{atan2}\left(y^c_{s,0} - \tilde{y}^c_{s,0},\, x^c_{s,0} - \tilde{x}^c_{s,0}\right), \quad \theta_i = \operatorname{atan2}\left(y^m_i - \tilde{y}^m_i,\, x^m_i - \tilde{x}^m_i\right) \tag{3.9}$$

where $(x^m_i,\, y^m_i)$ are the pixel coordinates of the i-th feature, and $(\tilde{x}^m_i,\, \tilde{y}^m_i)$ are the coordinates of the

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
1. **while** *image data is present* **do**
2. Select good features to track with the Shi and Tomasi's method.
3. Select an image patch around each feature and invert the patch vertically.
4. Slide each patch T on the source image T_0 and compute the NCC given by Eq. (3.8).
5. Match each patch T in the source image T_0 based on the NCC.
6. Compute the reference slope θ_0 and the matching slope θ_i given by Eq. (3.9).
7. **if** $|\theta_0 - \theta_i| > \eta$ **then**
8. Reject the matching result.
9. **end if**
10. Acquire reflection measurements from the matching results.
11. Track the feature and its reflection with the KLT algorithm.
12. **end while**

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 3.4 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 [55]. 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.

3.3 EKF Estimator

In this section, we formulate a discrete-time EKF [65] 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 i-th 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.

3.3.1 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. (3.1) and (3.4) as follows

$$\boldsymbol{\mu}_k = \boldsymbol{f}(\boldsymbol{\mu}^+_{k-1},\, \mathbf{q}^w_{b,k-1},\, \boldsymbol{\omega}^b_{k-1},\, \tilde{\mathbf{a}}^b_{k-1}) \tag{3.10}$$

where

$$\boldsymbol{f}(\boldsymbol{\mu}^+_{k-1},\, \mathbf{q}^w_{b,k-1},\, \boldsymbol{\omega}^b_{k-1},\, \tilde{\mathbf{a}}^b_{k-1}) = \begin{pmatrix} \hat{\mathbf{p}}^{w+}_{b,k-1} + R(\mathbf{q}^w_{b,k-1})\hat{\mathbf{v}}^{b+}_{k-1}\Delta t \\ \hat{\mathbf{v}}^{b+}_{k-1} + \left(-[\boldsymbol{\omega}^b_{k-1}]_\times \hat{\mathbf{v}}^{b+}_{k-1} + \tilde{\mathbf{a}}^b_{k-1} - \hat{\mathbf{b}}^{b+}_{a,k-1} + R^T(\mathbf{q}^w_{b,k-1})\mathbf{g}^w\right)\Delta t \\ \hat{\mathbf{b}}^{b+}_{a,k-1} \\ \hat{\mathbf{x}}^{b+}_{1,k-1} + \mathbf{f}_1(\boldsymbol{\mu}^+_{k-1},\, \boldsymbol{\omega}^b_{k-1})\Delta t \\ \vdots \\ \hat{\mathbf{x}}^{b+}_{n,k-1} + \mathbf{f}_n(\boldsymbol{\mu}^+_{k-1},\, \boldsymbol{\omega}^b_{k-1})\Delta t \end{pmatrix} \tag{3.11}$$

and

$$\begin{aligned} &\mathbf{f}_i(\boldsymbol{\mu}^+_{k-1},\, \boldsymbol{\omega}^b_{k-1}) \\ &= \begin{pmatrix} \left(-\hat{v}^+_{2,k-1} + \hat{h}^{b+}_{1,i,k-1}\hat{v}^+_{1,k-1}\right)\hat{\rho}^{b+}_{i,k-1} + \hat{h}^{b+}_{2,i,k-1}\omega_{1,k-1} - \left(1 + \left(\hat{h}^{b+}_{1,i,k-1}\right)^2\right)\omega_{3,k-1} + \hat{h}^{b+}_{1,i,k-1}\hat{h}^{b+}_{2,i,k-1}\omega_{2,k-1} \\ \left(-\hat{v}^+{}_{3,k-1} + \hat{h}^{b+}_{2,i,k-1}\hat{v}^+_{1,k-1}\right)\hat{\rho}^{b+}_{i,k-1} - \hat{h}^{b+}_{1,i,k-1}\omega_{1,k-1} + \left(1 + \left(\hat{h}^{b+}_{2,i,k-1}\right)^2\right)\omega_{2,k-1} - \hat{h}^{b+}_{1,i,k-1}\hat{h}^{b+}_{2,i,k-1}\omega_{3,k-1} \\ \left(-\omega_{3,k-1}\hat{h}^{b+}_{1,i,k-1} + \omega_{2,k-1}\hat{h}^{b+}_{2,i,k-1}\right)\hat{\rho}^{b+}_{i,k-1} + \hat{v}^+_{1,k-1}\left(\hat{\rho}^{b+}_{i,k-1}\right)^2 \end{pmatrix} \end{aligned} \tag{3.12}$$

Here, $\boldsymbol{\mu}^+_{k-1}$ is the state estimate from the previous time-step; $\mathbf{q}^w_{b,k-1}$ is the attitude quaternion of the UAV, and $\tilde{\mathbf{a}}^b_{k-1}$ and $\boldsymbol{\omega}^b_{k-1}$ 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 $\Sigma_k = F_{k-1}\Sigma^+_{k-1}F^T_{k-1} + W_{k-1}$, where F_{k-1} is the Jacobian of the motion model $\boldsymbol{f}(\boldsymbol{\mu}^+_{k-1},\, \mathbf{q}^w_{b,k-1},\, \boldsymbol{\omega}^b_{k-1},\, \tilde{\mathbf{a}}^b_{k-1})$ in Eq. (3.11) evaluated at $\boldsymbol{\mu}^+_{k-1}$, and W_{k-1} represents the covariance of the process noise.

The prediction of the error angle vector $\delta\hat{\boldsymbol{\theta}}^w_b \in \mathbb{R}^3$ and the gyroscope bias error $\Delta\hat{\mathbf{b}}^b_g = \mathbf{b}^b_g - \hat{\mathbf{b}}^b_g \in \mathbb{R}^3$ [66, 38] can be included in Eq. (3.11) as

$$\begin{aligned} \delta\hat{\boldsymbol{\theta}}^w_{b,k} &= \delta\hat{\boldsymbol{\theta}}^{w+}_{b,k-1} - \left(\left([\tilde{\boldsymbol{\omega}}^b_{k-1}]_\times - [\hat{\mathbf{b}}^{b+}_{g,k-1}]_\times\right)\delta\hat{\boldsymbol{\theta}}^{w+}_{b,k-1} + \Delta\hat{\mathbf{b}}^{b+}_{g,k-1}\right)\Delta t \\ \Delta\hat{\mathbf{b}}^b_{g,k} &= \Delta\hat{\mathbf{b}}^{b+}_{g,k-1} \end{aligned} \tag{3.13}$$

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

$$\hat{\boldsymbol{q}}_{b,k}^{w} = \hat{\boldsymbol{q}}_{b,k-1}^{w+} + \frac{1}{2}\left(\Omega(\tilde{\boldsymbol{\omega}}_{k-1}^{b}) - \Omega(\hat{\mathbf{b}}_{g,k-1}^{b+})\right)\hat{\boldsymbol{q}}_{b,k-1}^{w+}\Delta t \tag{3.14}$$

The error angle vector $\delta\hat{\boldsymbol{\theta}}_{b}^{w}$ is a minimal representation derived with a small angle approximation of the error quaternion $\delta\hat{\mathbf{q}}_{b}^{w} = \mathbf{q}_{b}^{w} \otimes (\hat{\mathbf{q}}_{b}^{w})^{-1}$ [66, 38], 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. (3.11) and (3.12), 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 [12, 13, 14] 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 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.

3.3.2 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

$$\mathbf{h}(\boldsymbol{\mu}_{k},\, \mathbf{q}_{b,k}^{w},\, \mathbf{p}_{bi}^{w},\, \mathbf{q}_{bi}^{w}) = \begin{pmatrix} \mathbf{h}_{1:n}^{b}(\boldsymbol{\mu}_{k}) \\ \mathbf{h}_{1:n}^{bi}(\boldsymbol{\mu}_{k},\, \mathbf{q}_{b,k}^{w}, \mathbf{p}_{bi}^{w}, \mathbf{q}_{bi}^{w}) \\ \tilde{\mathbf{h}}_{1:n}^{b}(\boldsymbol{\mu}_{k},\, \mathbf{q}_{b,k}^{w}) \\ -\hat{z}_{b,k}^{w} \end{pmatrix} \tag{3.15}$$

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

$$\mathbf{h}_{i}^{b}(\boldsymbol{\mu}_{k}) = \begin{pmatrix} \hat{h}_{1,i,k}^{b} & \hat{h}_{2,i,k}^{b} \end{pmatrix}^{T} \tag{3.16}$$

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 3.2.3.

The initial view of the i-th point feature is given by

$$\mathbf{h}_i^{bi}(\boldsymbol{\mu}_k,\, \mathbf{q}_{b,k}^w, \mathbf{p}_{bi}^w, \mathbf{q}_{bi}^w) = \begin{pmatrix} \hat{y}_{i,k}^{bi}/\hat{x}_{i,k}^{bi} & \hat{z}_{i,k}^{bi}/\hat{x}_{i,k}^{bi} \end{pmatrix}^T \tag{3.17}$$

where $\hat{\mathbf{p}}_{i,k}^{bi} = (\hat{x}_{i,k}^{bi},\, \hat{y}_{i,k}^{bi},\, \hat{z}_{i,k}^{bi})^T = R^T(\mathbf{q}_{bi}^w)\left(\hat{\mathbf{x}}_{b,k}^w - \hat{\mathbf{x}}_{bi}^w\right) + R^T(\mathbf{q}_{bi}^w)R(\mathbf{q}_{b,k}^w)\hat{\mathbf{p}}_{i,k}^b$ (which is given by Eq. (3.5)) 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 i-th 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 i-th feature is given by

$$\tilde{\mathbf{h}}_i^b(\boldsymbol{\mu}_k,\, \mathbf{q}_{b,k}^w) = \begin{pmatrix} \hat{\tilde{x}}_{i,k}^b/\hat{\tilde{z}}_{i,k}^b & \hat{\tilde{y}}_{i,k}^b/\hat{\tilde{z}}_{i,k}^b \end{pmatrix}^T \tag{3.18}$$

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. (3.6)) is the estimated position of the mirrored point of the i-th feature with respect to the current UAV body frame (see Figure 3.3).

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

$$\begin{aligned} \boldsymbol{\mu}_k^+ &= \boldsymbol{\mu}_k + K_k\left(\mathbf{z}_k - \mathbf{h}(\boldsymbol{\mu}_k,\, \mathbf{q}_{b,k}^w,\, \mathbf{p}_{bi}^w,\, \mathbf{q}_{bi}^w)\right) \\ \Sigma_k^+ &= \Sigma_k - K_k H_k \Sigma_k \end{aligned} \tag{3.19}$$

where the Kalman gain is given by $K_k = \Sigma_k H_k^T\left(H_k\Sigma_k H_k^T + V_k\right)^{-1}$. 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. (3.15) 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. (3.11), the estimate of the UAV's attitude and the gyroscope bias can be updated by

$$\begin{aligned} \hat{\boldsymbol{q}}_{b,k}^{w+} &= \hat{\boldsymbol{q}}_{b,k}^w + \frac{1}{2}\Omega(\delta\hat{\boldsymbol{\theta}}_{b,k}^{w+})\hat{\boldsymbol{q}}_{b,k}^w \\ \hat{\mathbf{b}}_{g,k}^{b+} &= \hat{\mathbf{b}}_{g,k}^b + \Delta\hat{\mathbf{b}}_{g,k}^{b+} \end{aligned} \tag{3.20}$$

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.

3.3.3 World Reference Frame Representation

The motion model in Eq. (3.11) 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 3.4. 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

$$\hat{\mathbf{p}}^w_{i,k} = \hat{\mathbf{p}}^w_{b,k} + R(\mathbf{q}^w_{b,k})\hat{\mathbf{p}}^b_{i,k} \tag{3.21}$$

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 i-th 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.

3.4 Observability Analysis

The observability problem of VINS [1, 38, 39, 40] and SLAM [35, 36, 37] 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 3.2, 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.

3.4.1 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 $[t_0, t_f]$ [67]. 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 [68], the observability of nonlinear systems are categorized to be observable, locally observable, weakly observable, and locally weakly observable. Local weak observability is defined in [68] 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 [68] each state from its neighbors. In [36], it is stated that if the nonlinear system in 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. (3.15). The local weak observability can be analyzed with the rank of the nonlinear observability matrix $\mathcal{O}_{NL}$. We formulate the nonlinear observability matrix $\mathcal{O}_{NL}$ by recursively computing the Lie derivatives of the measurement function $\mathbf{h}$ in Eq. (3.15) 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. (3.11) and (3.12) as shown in [38]. The nonlinear observability matrix is given by

$$\mathcal{O}_{NL} = \nabla \begin{pmatrix} (L^0\mathbf{h})^T & (L^1_{\mathbf{f}_0}\mathbf{h})^T & \cdots & (L^\gamma_{\mathbf{f}_0\mathbf{f}_1\mathbf{f}_2}\mathbf{h})^T & \cdots \end{pmatrix}^T \tag{3.22}$$

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 γ-th 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.

3.4.2 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. (3.15) along with the motion model of our reduced-order system given by Eqs. (3.11) and (3.12). We consider situations where 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.

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

The nonlinear observability matrix $\mathcal{O}_{NL}$ for our estimation system given by Eqs. (3.11) and (3.15) 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. (3.15) provides sufficient constraints with the information from the motion model given by Eqs. (3.11) and (3.12).

Observability with Only Current View Measurements

We consider the case where 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 a

$$\text{span} \begin{pmatrix} I_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times2} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & -\frac{\left(\mathbf{v}^b\right)^T}{\rho_i^b} & -\frac{\left(\mathbf{b}^b\right)^T + \mathbf{g}^T R(\mathbf{q}_b^w)}{\rho_i^b} & \mathbf{0}_{1\times2} & 1 \end{pmatrix}^T \tag{3.23}$$

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. (3.24). 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

$$\text{span} \begin{pmatrix} I_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & I_{3\times3} \\ -\mathbf{g}^T[\mathbf{p}_b^w]_\times^T & -\mathbf{g}^T[\mathbf{v}_b^w]_\times^T & \mathbf{0}_{3\times3} & -\mathbf{g}^T[\mathbf{p}_i^w]_\times^T \end{pmatrix}^T \tag{3.24}$$

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. (3.24), 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 [39].

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

$$\text{span}\begin{pmatrix} \frac{(\mathbf{p}_{bi}^w - \mathbf{p}_b^w)^T}{\rho_i^b} & -\frac{\left(\mathbf{v}^b\right)^T}{\rho_i^b} & -\frac{\left(\mathbf{b}^b + \mathbf{g}\right)^T}{\rho_i^b} & \mathbf{0}_{1\times 2} & 1 \end{pmatrix}^T \tag{3.25}$$

For this case, we fixed the attitude of the UAV with $R(\mathbf{q}_b^w) = I_{3\times 3}$ 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. (3.15) except for the initial view $\mathbf{h}_i^{bi}$ of a feature, the null space of the nonlinear observability matrix $\mathcal{O}_{NL}$ can be found as

$$\text{span}\begin{pmatrix} I_{2\times 3} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 3} & \mathbf{0}_{2\times 3} & \mathbf{0}_{2\times 3} \end{pmatrix}^T \tag{3.26}$$

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. (3.23)-(3.26) 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 3.4.3, we will quantify the degree of observability of our estimation system.

3.4.3 Degree of Observability

In Section 3.4.2, we used the observability matrix to analytically determine whether the system is observable and 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 [69]. 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

$$\begin{aligned} \Psi \triangleq\; & H_k^T H_k + F_k^T H_{k+1}^T H_{k+1} F_k + F_k^T F_{k+1}^T H_{k+2}^T H_{k+2} F_{k+1} F_k + \\ & \cdots + F_k^T \cdots F_{k+m-1}^T H_{k+m}^T H_{k+m} F_{k+m-1} \cdots F_k \end{aligned} \tag{3.27}$$

where F_k and H_k are the Jacobian matrices of the dynamic function given Eqs. (3.11) and (3.12) and the measurement function given by Eq. (3.15), respectively, at time-step k. The smallest eigenvalue of $\Psi^{1/2}$ shows the degree of observability [69].

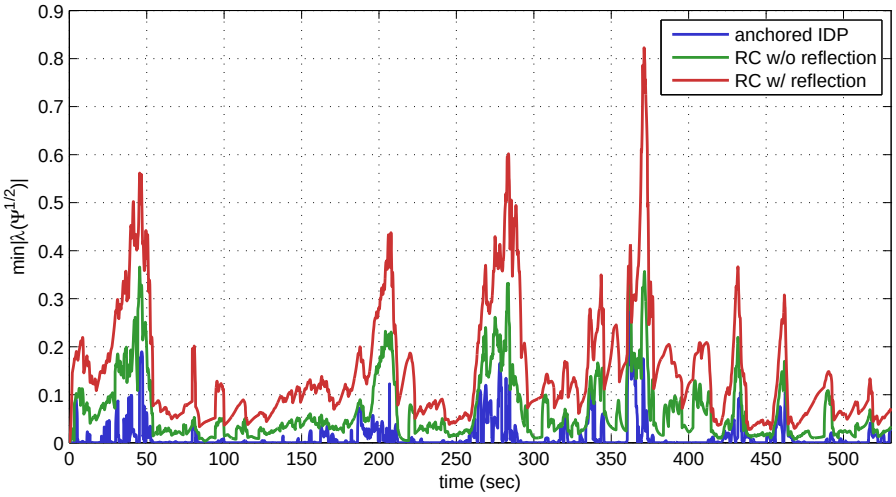


Figure 3.5: The degree of observability of our localization and robot-centric mapping system with reflection measurements and the anchored IDP SLAM system are shown, where the observability Gramian Ψ is defined in Eq. (3.27).

We compare our localization and robot-centric mapping system, which exploits the reflection measurements as presented in Sections 3.2 and 3.3, 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) [16, 17, 18] 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 3.5. Figure 3.5 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 3.5 with numerical simulation results of the localization and mapping.

3.5 Numerical Simulations

In this section, we present results of numerical simulation and analyze the performance of our riverine localization and mapping algorithm. In Section 3.6, we will present experimental results using real-world

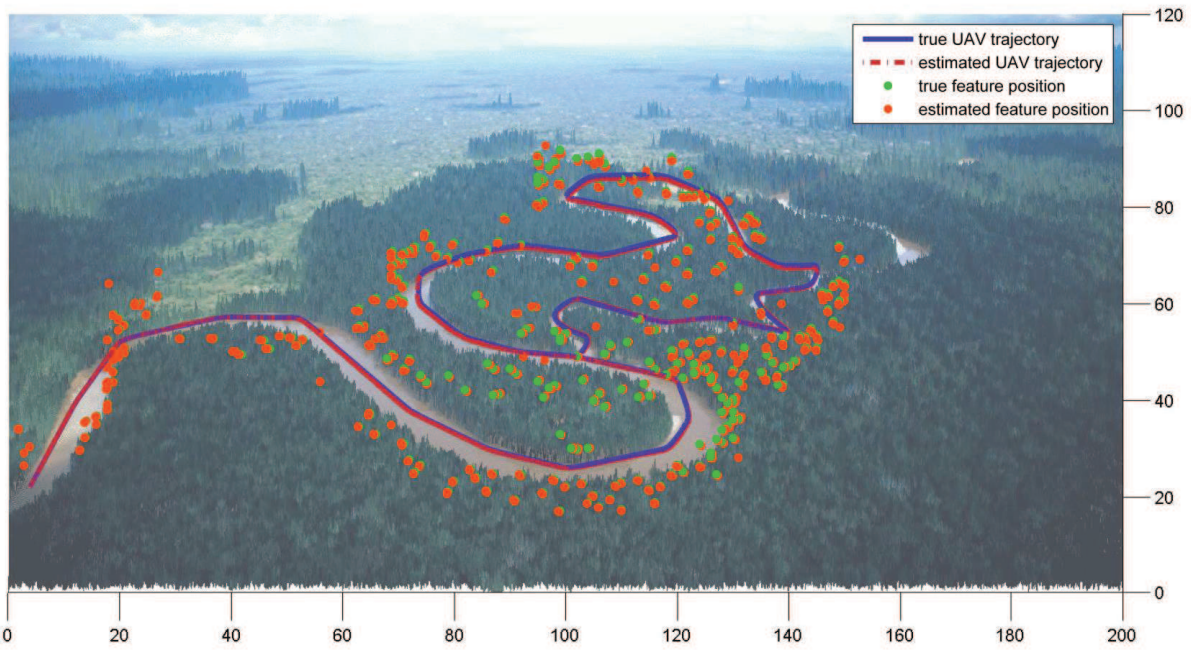
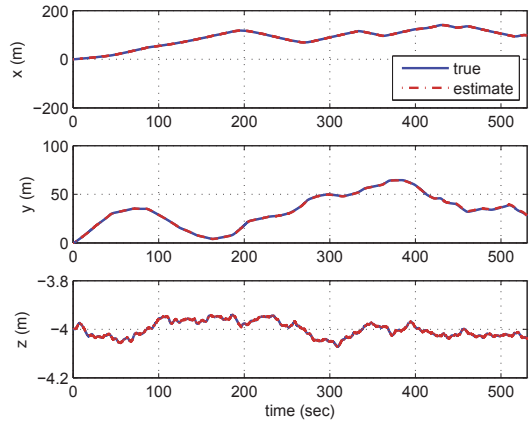


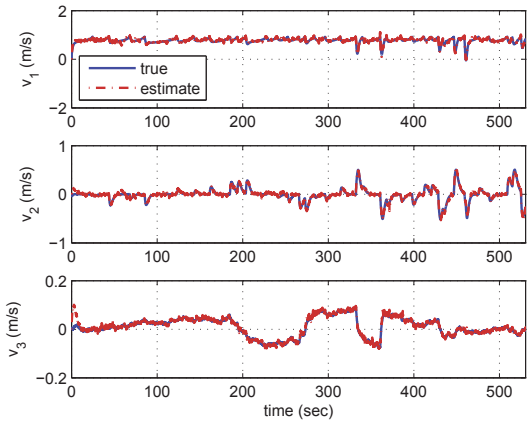
Figure 3.6: Results of the localization and mapping in a simulated riverine environment. The solid blue curve shows the trajectory of the UAV and the green dots are the 3D point features extracted from the trees. The dashed red curve is the time-history of the UAV's location estimate and the orange dots are the estimated locations of the features.

data. Here, we simulate a riverine environment with a river image [70] as shown in Figure 3.6. 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 seconds 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 height of the features are distributed with a uniform distribution on the interval $0 \sim 30$ m. The features that are between $5 \sim 20$ m away from the camera that has a 90 degree 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.

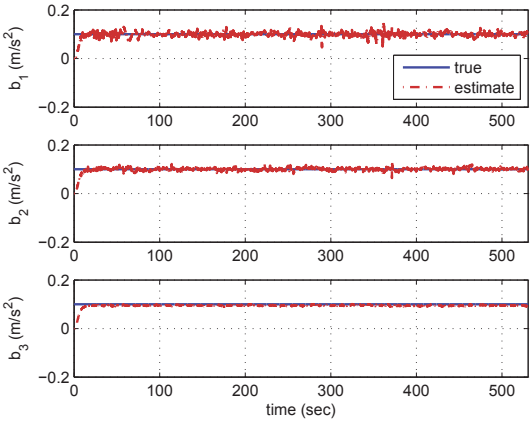
Gaussian white noise of standard deviation $\sigma = 0.01$ and $\sigma = 0.001$ is added to the acceleration and angular velocity readings, and the attitude and altitude measurements, respectively. The noise in camera pixel measurements is simulated as Gaussian white noise of $\sigma = 1$ considering 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



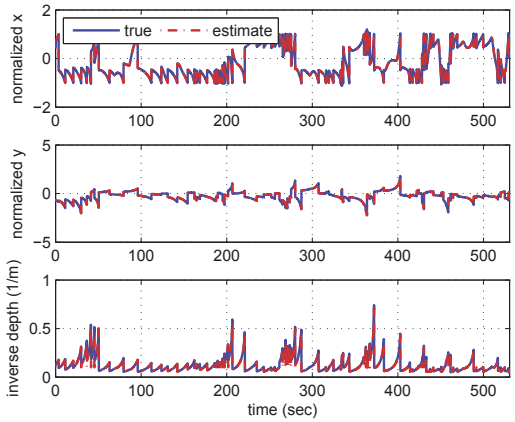
(a) Estimate of the UAV's location



(b) Estimate of the UAV's velocity



(c) Estimate of the accelerometer bias

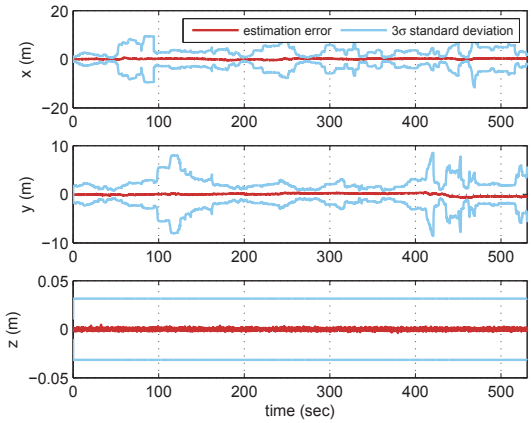


(d) Estimate of the features

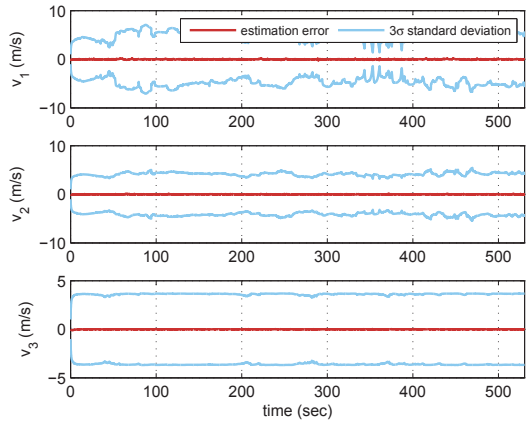
Figure 3.7: The location estimate of the UAV in the world reference frame, the velocity estimate of the UAV with respect to the UAV body frame, and the accelerometer bias estimate are shown. The estimates of the point features with respect to the UAV body frame are also shown.

are initialized as $\hat{\mathbf{v}}_0^b = \mathbf{0}$ and $\hat{\mathbf{b}}_0^b = \mathbf{0}$, respectively. The i-th point feature for Eq. (3.12) is initialized as $\hat{\mathbf{x}}_{i,0}^b = (h_{1,i,0}^b,\, h_{2,i,0}^b,\, 0.1)^T$, where $h_{1,i,0}^b$ and $h_{2,i,0}^b$ are the initial normalized coordinates of the feature.

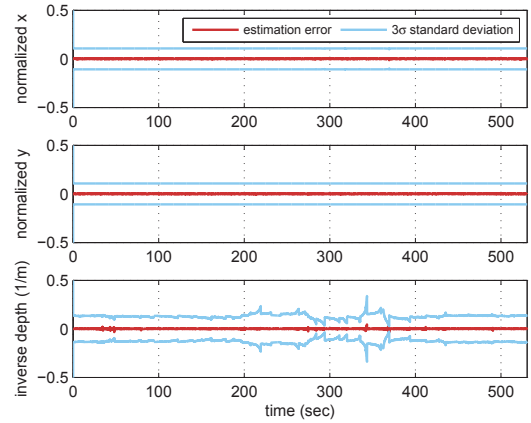
Figure 3.6 shows the localization and mapping results from our localization and robot-centric mapping system with reflection measurements projected on the simulated environment. The simulation results show the time-history of the location estimate of the UAV converging to the true trajectory of the UAV, and the estimated positions of the features converging to their true positions. Figure 3.7 also compares the estimates of the state variables and their true values. 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 ($\hat{\rho}_{i,k}^b = 1/\hat{x}_{i,k}^b$) of the features. The results show that the estimates converge to their true values. The estimates of the features, which are shown in Figure 3.7(d), are in the UAV body frame as described in Section 3.2.2. We



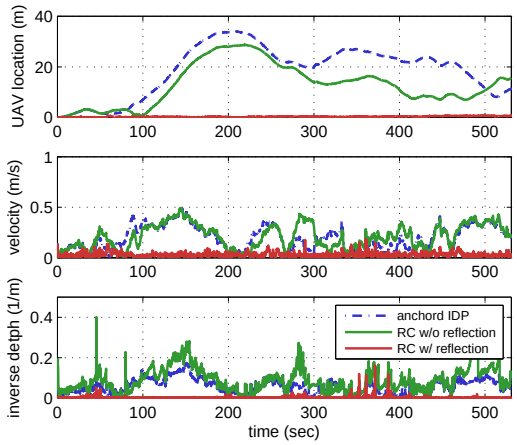
(a) Estimation error of the UAV's location



(b) Estimation error of the UAV's velocity



(c) Estimation error of the features



(d) Estimation error norm

Figure 3.8: 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.

represent the robot-centric results in the world reference frame as described in Section 3.3.3 and generate mapping results that are shown in Figure 3.6.

Figure 3.8 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 3.8(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 0.3155 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 18.7497 m for the location of the UAV, 0.1969 m/s for the velocity of the UAV, and 0.0589 (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 3.4.3.

3.6 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 3.9). 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 [16, 18, 17].
- 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 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.

3.6.1 Methodology

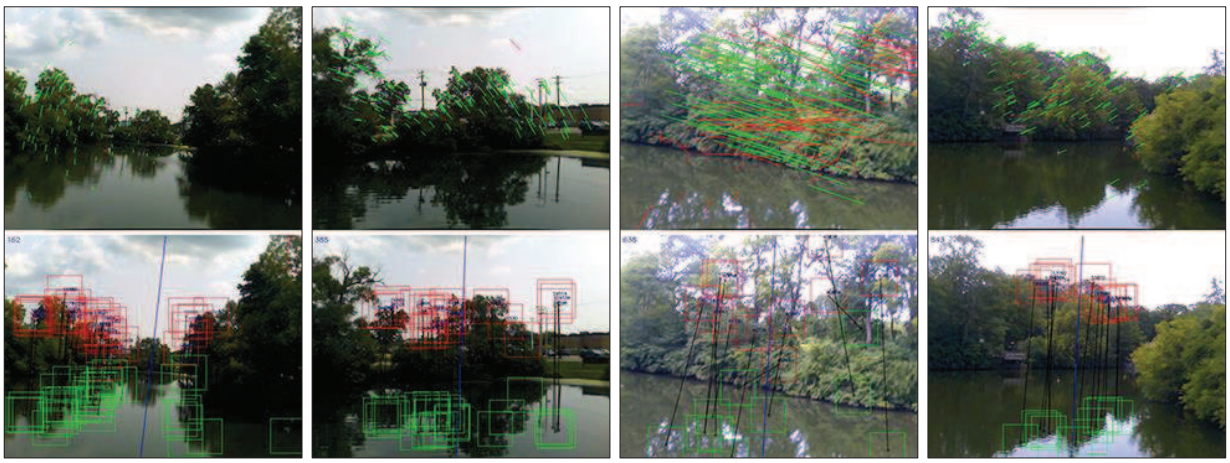
For all experiments, we flew our quadcopter UAV (described in Section 3.2.1) at Crystal Lake in Urbana, Illinois using the altitude hold mode of the onboard automatic flight control system that accepts a radio control pilot input for heading control. We performed a calibration of all parameters of the sensing system, including the camera intrinsic parameters (through camera calibration [71]) and the orientation between the IMU and the camera (with IMU-camera calibration [72]). 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

Figure 3.9: 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.

velocity. We used the Ublox Lea-6H GPS module that has an accuracy of 2.5 m and used the position results that are filtered with the inertial navigation system (INS) for the ground truth.

Figure 3.10 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 [61] and found their reflections with Algorithm 1, which we presented in Section 3.2.3. The algorithm tracked the features with the pyramid KLT method [55] and sorted outliers with random sample consensus (RANSAC) [55] 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 quad-core computer with the features shown in Figures 3.10 and 3.11. We simplified the estimation process by estimating the vector $\hat{\mathbf{x}}_{i,k}^{b}$ 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 $\hat{\mathbf{p}}_{i,k}^{w}$ of each feature which is derived from the



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

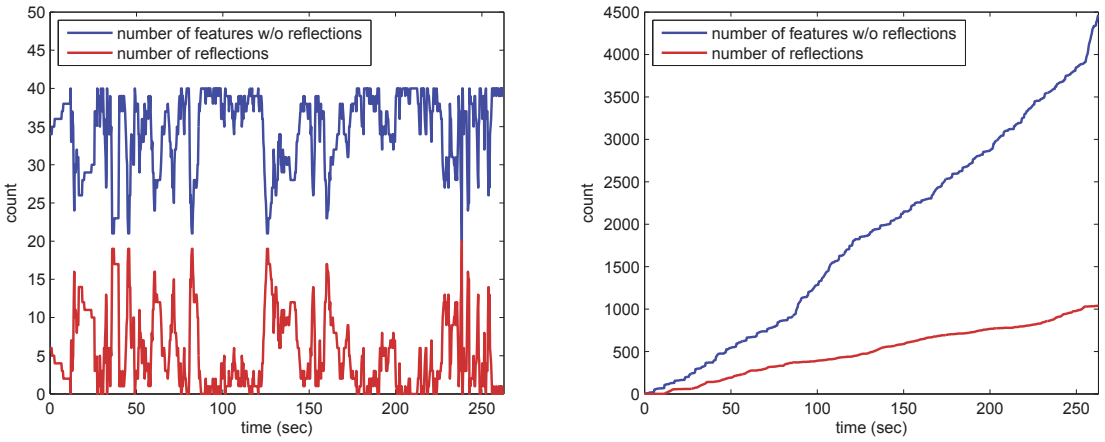
Figure 3.10: Feature tracking on image data from Crystal Lake. Feature tracking results (green lines) with the pyramid KLT method and outliers (red lines) are shown in the first row. Matching of the reflections (green boxes) corresponding to real objects (red boxes) with Algorithm 1 are shown in the second row.

state estimate $\hat{\mathbf{x}}_{i,k}^{b}$ of the feature. We initialized the location estimate and the velocity estimate of the UAV as $\hat{\mathbf{p}}_{b,0}^{w} = (0,\, 0,\, -z_{b,0}^{w})^{T}$ and $\hat{\mathbf{v}}_{0}^{b} = \mathbf{0}$, respectively, and the accelerometer bias estimate as $\hat{\mathbf{b}}_{0}^{b} = \mathbf{0}$. We initialized the altitude $-z_{b,0}^{w}$ of the UAV over the water with the measurements from the altimeter.

State-of-the-art SLAM methods [1, 27, 39] rely on loop closing to prevent drift over time. Therefore, we have implemented a simple vision-based algorithm to detect loop closure, and incorporated a post-processing stage to minimize the final error between our UAV's location estimate and the ground truth. Our algorithm used speeded-up robust features (SURF) [55] to find the best match between image data that was acquired from the starting location and from when our UAV quadcopter revisited the starting point. Then, our algorithm used a Kalman smoother [73] to constrain the two location estimates to coincide and filtered the entire trajectory.

3.6.2 Experimental Results

Figure 3.12 gives a quantitative summary of our results. Figure 3.12(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 3.12(b) shows the estimation errors of the three aforementioned methods, relative to the GPS/INS data. Figure 3.12(c) shows the estimate of the UAV's velocity. Figure 3.12(d) shows the normalized coordinates and the inverse-depth estimates of the features. In Figure 3.12(d), the estimation of old features that move out of sight are re-initialized with new features as we stated in Section 3.6.1.



(a) Number of features in each image

(b) Accumulated number of features

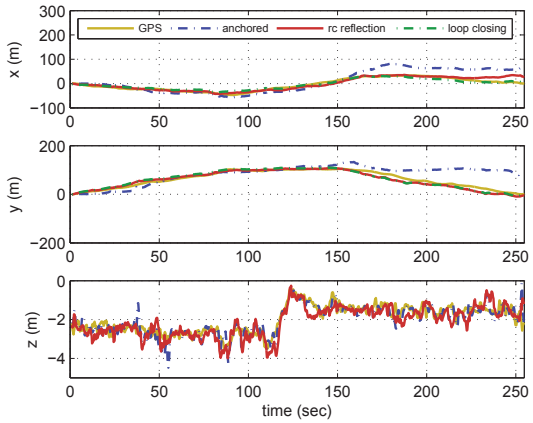
Figure 3.11: The number of features incorporated in the measurement vector with and without reflection measurements.

The estimation results and GPS/INS data are overlaid on a satellite image of Crystal Lake provided by Google Maps in Figure 3.13. According to the GPS data, the quadcopter traveled approximately 343.48 m for 253 seconds. 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 makes the unobservable rotation directly measurable. The reduced-order state estimator we presented in Section 3.3 uses the drift-free attitude information acquired by the IMU and the magnetometer.

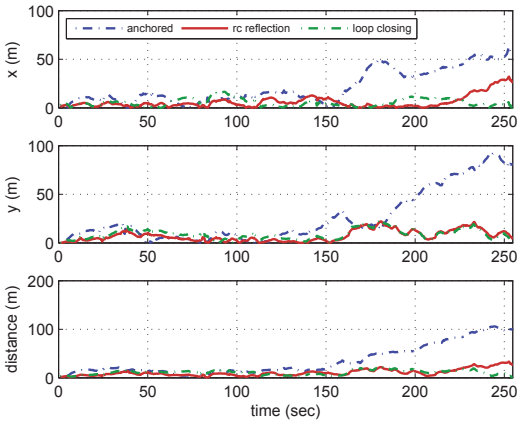
The final error between the GPS data and the estimated location of the UAV was 26.26 m for our localization and robot-centric mapping system with reflection measurements, 100.36 m for the anchored IDP SLAM method without reflection measurements, and 0.67 m for our method with loop closing. The average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

3.6.3 Lessons Learned

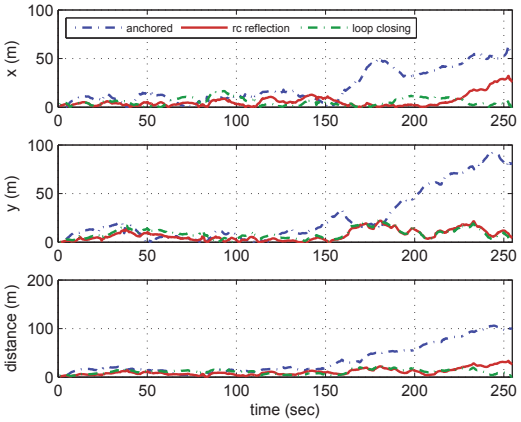
As can be seen in both Figures 3.12(b) and 3.13, our method outperforms the anchored IDP method, and incorporating loop closure provides further improvement. In particular, the drift along the X-Y plane is reduced when we used our localization and robot-centric mapping system which uses reflection measurements, and 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



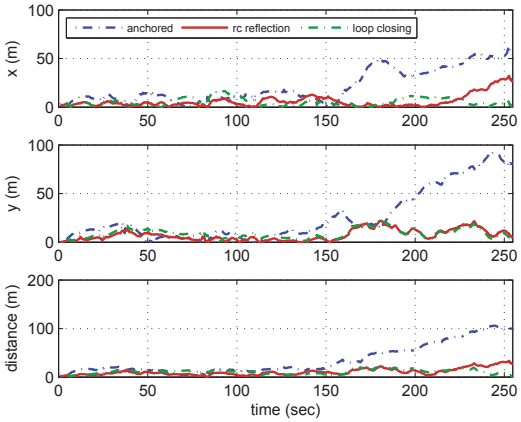
(a) Estimate of the UAV's location



(b) Estimation error of the UAV's location



(c) Estimate of the UAV's velocity



(d) [Estimates of the features

Figure 3.12: 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.

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 3.4).

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

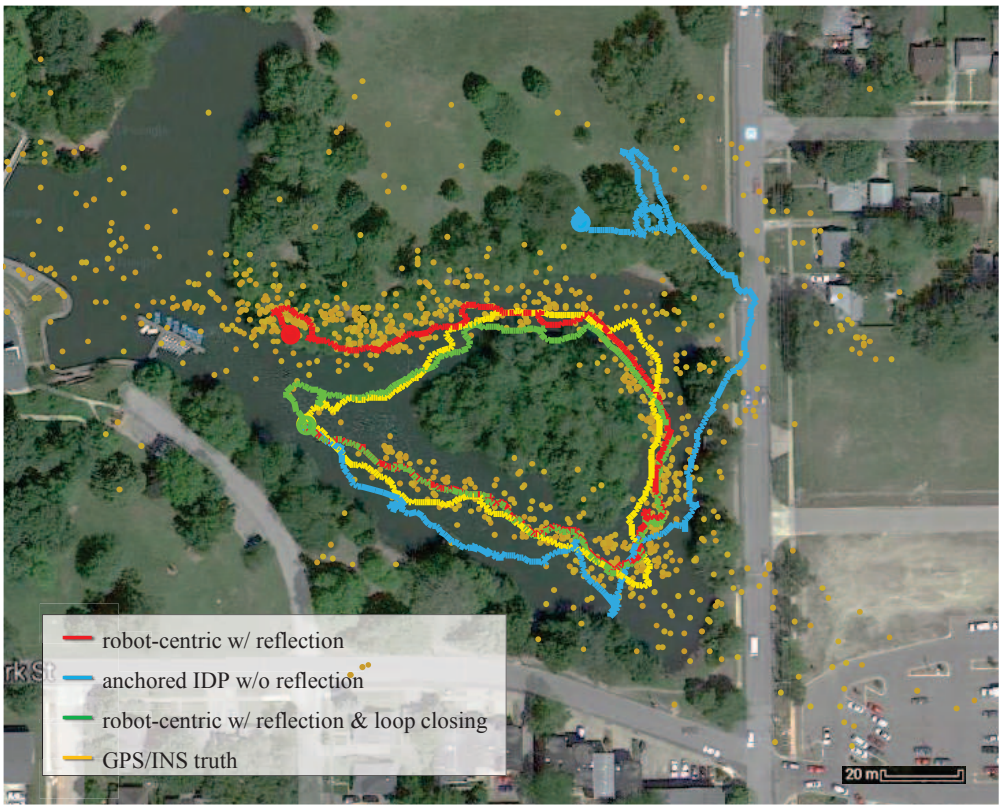


Figure 3.13: The experimental results are overlaid on a satellite image of Crystal Lake provided by Google Maps. The time-history of the UAV's location estimate from our robot-centric method with reflections (red) and the anchored IDP method without reflections (blue) and the position estimate of the features from our method with reflections (orange dots) are shown. 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.

which Algorithm 1 was unable to find reflections. This can be seen in Figure 3.11(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 3.10(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∼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 3.14. At the positions indicated by the blue circles, significant pose estimation error

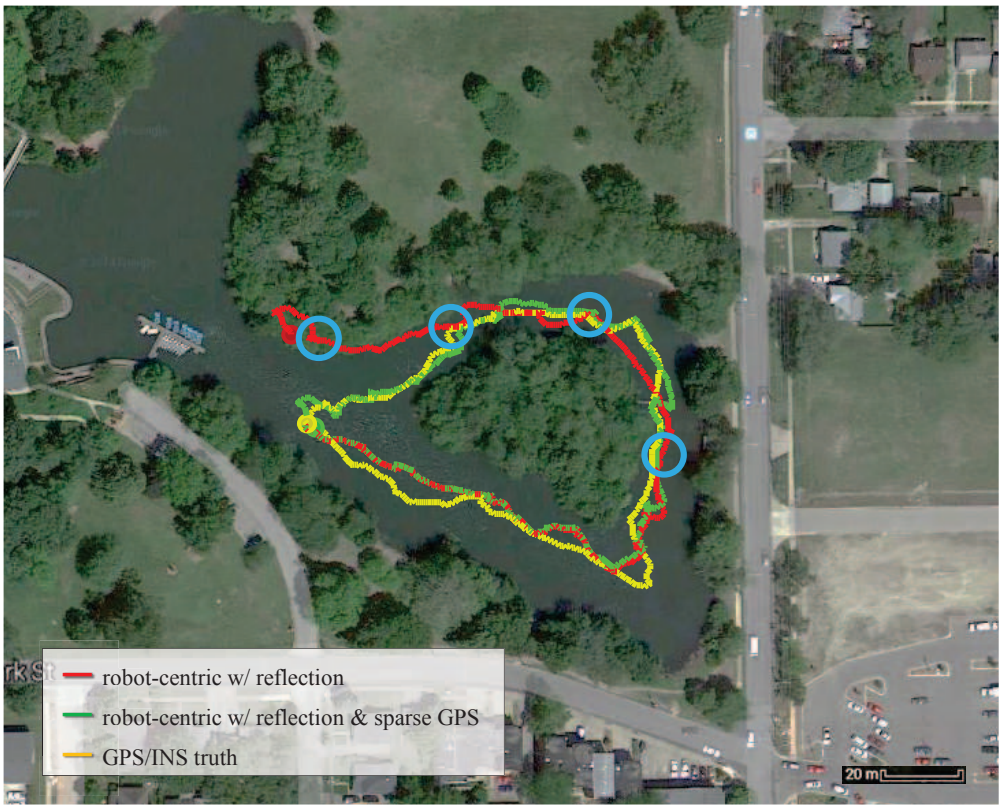


Figure 3.14: The experimental results show that a short sequences of badly estimated poses (blue circles) can cause the pose estimates to drift (red). The localization result that is obtained when GPS data are provided as measurements to the smoothing filter around these points is also shown (green).

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 both the detrimental consequences of even a small number of pose estimation errors, as well as pointing to the utility of our method in situations for which intermittent GPS data might be available.

3.7 Conclusions

In this chapter, 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.

Chapter 4

Observer Design for Localization and Mapping with an Omnidirectional Camera for Autonomous Mowing

4.1 Chapter Objective

In this chapter, we present an omnidirectional-vision-based localization and mapping algorithm with an application to autonomous mowing. We divide the task for robotic mowing into two separate phases, a teaching phase and a mowing phase. During the teaching phase, the mower estimates the 3D positions of landmarks and defines a boundary in the lawn with an estimate of its own trajectory. During the mowing phase, the location of the mower is estimated using the landmark and boundary map acquired from the teaching phase. Of particular interest for our work is ensuring that the estimator for landmark mapping will not fail due to the nonlinearity of the system during the teaching phase. A nonlinear observer is designed with pseudo-measurements of each landmark's depth to prevent the map estimator from diverging. Simultaneously, the boundary is estimated with an EKF. Measurements taken from an omnidirectional camera, an IMU, and a vehicle speed sensor are used for the estimation. Numerical simulations and offline teaching phase experiments with our autonomous mower demonstrate the potential of our algorithm.

The rest of this chapter is organized as follows. In Section 4.2, we give an overview of our system. In Section 4.3, we describe the dynamics and measurements for landmark mapping, and design a nonlinear observer with hybrid contraction analysis. In Section 4.4, an EKF estimator is presented to estimate the trajectory of the mower and define a boundary in the lawn. In Section 4.5, the landmarks are used to estimate the location of the autonomous mower and solve the containment problem with the boundary information. Numerical simulations are shown in Section 4.6. Offline experiments of the teaching phase are presented in Section 4.7. We conclude with plans for future work in Section 4.8.

4.2 Overview of the System

In this work, we used a robotic mower from John Deere which is shown in Figure 4.1. Our autonomous mower is equipped with a ground speed sensor and modified with an omnidirectional vision sensor and an

Figure 4.1: Our autonomous mower modified with an omnidirectional camera and an IMU for experiments.

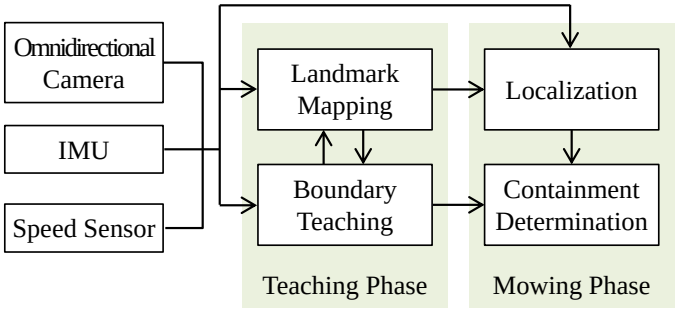


Figure 4.2: Block diagram of our localization and mapping strategy developed for the autonomous mowing.

IMU. The task for robotic mowing can be divided into two phases, a teaching phase and a mowing phase as shown in Figure 4.2. During the teaching phase, the mower can follow a boundary wire temporarily set-up in the lawn or can be tele-operated by a user. Our algorithm defines a boundary by estimating the trajectory of the mower with an EKF while generating a point feature-based map of its surrounding landmarks with an observer. We designed a nonlinear observer to estimate the 3D positions of landmarks with respect to the robot's body frame.

There have been several research prototypes as well as manufactured products developed for robotic lawn mowing [74, 75, 76]. Boundary wires are widely used to ensure containment in available products. However, they require users to add infrastructure to the environment which increases set-up time and decreases portability. GPS has been widely used for navigation purposes [77] but performs best in a wide open area. It can be difficult to get accurate position estimation results with the GPS in a residential area occluded by

walls and tree canopies. RF and infrared signal based methods have also been demonstrated for localization [78, 76] but they can require costly infrastructure. Vision sensors are an attractive option for mowing robots for their potential to enable no-infrastructure installations and perform sensing functions beyond positioning such as determining and diagnosing turf problems.

4.3 Observer Design for Robot-Centric Landmark Mapping

In this section, we describe the dynamics of the vision system for robot-centric mapping and derive pseudo-measurements of the depth of each landmark. A nonlinear observer is designed for mapping, and the convergence of the estimates is proved with hybrid contraction analysis.

4.3.1 Dynamic Model for Landmark Mapping

The dynamic model of each landmark in the robot's body frame is given by [79]

$$\dot{\mathbf{x}}_i = -[\boldsymbol{\omega}]_\times \mathbf{x}_i - \mathbf{v} \tag{4.1}$$

where $\mathbf{x}_i \in \mathbb{R}^3$ is the location of the i-th landmark with respect to the robot's body frame. The x-axis of the robot's body frame is pointing towards the front of the mower, and the z-axis is pointing up from the mower. For simplicity, all the variables without a superscript are in the robot's body frame in this chapter.

The linear and angular velocities of the robot measured in the robot's body frame are denoted by $\mathbf{v} \in \mathbb{R}^3$ and $\boldsymbol{\omega} \in \mathbb{R}^3$. The skew-symmetric matrix $[\boldsymbol{\omega}]_\times \in so(3)$ is formed from the angular velocity vector $\boldsymbol{\omega}$.

We estimate the location of each landmark in the robot's body frame and let the measurements be linear with respect to the states. Similar to [80], a landmark can be described with a unit vector $\mathbf{y} = \mathbf{x}_i / \|\mathbf{x}_i\|_2 \in \mathbb{R}^3$ from the robot and its distance $d = \|\mathbf{x}_i\|_2 \in \mathbb{R}$. The state vector of each landmark is $\mathbf{z}^T = (d,\ \mathbf{y}^T)^T$, and the dynamics of the system is given by

$$\frac{d}{dt}\begin{pmatrix} d \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} -\mathbf{v}^T\mathbf{y} \\ -(I - \mathbf{y}\mathbf{y}^T)\mathbf{v}d^{-1} - [\boldsymbol{\omega}]_\times \mathbf{y} \end{pmatrix} + \eta \tag{4.2}$$

where $\eta \in \mathbb{R}^4$ denotes the disturbance.

Assumption 1 *The Euclidean distance d between the camera and a landmark is lower bounded by a known positive constant. Therefore, we assume that $\underline{d} \leq d$, where $\underline{d} \in \mathbb{R}^+$ is a known constant parameter.*

Remark 1 *Assumptions 1 is satisfied due to physical constraints of the system.*

Measurements of system in Eq. (4.2) is linear to the states. The unit vector $\mathbf{y}$ is directly measurable, whereas the depth d of a landmark is not directly measurable from a single image. However, pseudo-measurements of d can be formulated as we shall describe in Section 4.3.2.

4.3.2 Pseudo-Measurements of a Landmark's Depth

We formulate pseudo-measurements $d_k \in \mathbb{R}$ at current time-step k to acquire the depth of each landmark as

$$d_k = \|\hat{\mathbf{x}}_b^{b_j}\| \left(1 - \left(\frac{\hat{\mathbf{x}}_b^{b_j}}{\|\hat{\mathbf{x}}_b^{b_j}\|} \cdot \mathbf{y}_j\right)^2\right)^{\frac{1}{2}} \left(1 - \left(R^T(\hat{\mathbf{q}}_{b_j})\mathbf{y}_k \cdot \mathbf{y}_j\right)^2\right)^{-\frac{1}{2}} + \xi_{d,k} \tag{4.3}$$

where $\mathbf{y}_k$ and $\mathbf{y}_j$ are the direction vector measurements of a landmark at time-step k and at a previous time-step j. Noise in the pseudo-measurements d_k is denoted by $\xi_{d,k} \in \mathbb{R}$, and $R(\cdot)$ is a rotation matrix. In Section 4.4, we estimate the location $\hat{\mathbf{x}}_{b_j}$ and orientation quaternions $\hat{\mathbf{q}}_{b_j}$ of the robot's body frame at time-step j with respect to its current body frame. The current location of the mower with respect to its body frame at time-step j, which is used in Eq. (4.3), is $\hat{\mathbf{x}}_b^{b_j} = -R^T(\mathbf{q}_{b_j})\hat{\mathbf{x}}_{b_j}$.

4.3.3 Observer Design with Hybrid Contraction Analysis

The observer presented in this section updates the estimate of the state using vision measurements at discrete-time instances and propagates the motion between the measurements in continuous-time. We use dwell-time $\Delta t_k = t_k - t_{k-1}$ for vision measurements $\mathbf{y}_k$ since image processing can be much slower than the inertial measurements which is used in the dynamic model. One can also consider using the direction estimates $\hat{\mathbf{y}}_k$ to improve the vision tracking algorithm. We can allow sufficiently long dwell-time to track landmarks which are instantaneously occluded in images.

Estimation of landmarks is decoupled by using separate observers. This gives us a potential to increase the number of landmarks for the map. It is shown in [19] that increasing the number of landmarks is more profitable than increasing the measurement rate in order to enhance the accuracy of the estimates. We prove that our observer is guaranteed to be globally exponentially stable by using contraction theory.

4.3.4 Observer Design and Stability Analysis

The observer for each landmark is given by

$$\frac{d}{dt}\begin{pmatrix}\hat{d}\\ \hat{\mathbf{y}}\end{pmatrix} = \begin{pmatrix}-\mathbf{v}^T\hat{\mathbf{y}}\\ -(I-\hat{\mathbf{y}}\hat{\mathbf{y}}^T)\mathbf{v}\hat{d}^{-1} - [\boldsymbol{\omega}]_\times\hat{\mathbf{y}}\end{pmatrix} \tag{4.4}$$

$$\begin{pmatrix}\hat{d}_k^+\\ \hat{\mathbf{y}}_k^+\end{pmatrix} = \begin{pmatrix}\hat{d}_k^- + l_{k,1}(d_k - \hat{d}_k^-)\\ \hat{\mathbf{y}}_k^- + Il_{k,2}(\mathbf{y}_k - \hat{\mathbf{y}}_k^-)\end{pmatrix} \tag{4.5}$$

where $\hat{d} \in \mathbb{R}$ is an estimate of a landmark's depth between vision measurements, $\hat{d}_k^- \in \mathbb{R}$ and $\hat{d}_k^+ \in \mathbb{R}$ are estimates of the depth before and after the measurement update at time-step k, $\hat{\mathbf{y}} \in \mathbb{R}^3$ is an estimate of a unit vector from the robot's body frame to the landmark between its measurements, $\hat{\mathbf{y}}_k^- \in \mathbb{R}^3$ and $\hat{\mathbf{y}}_k^+ \in \mathbb{R}^3$ are estimates of the unit vector before and after the measurement update at time-step k, and $I \in \mathbb{R}^{3\times3}$ is an identity matrix. The user can select observer gains $l_{k,1} \in \mathbb{R}$ and $l_{k,2} \in \mathbb{R}$ for $\hat{d}_k$ and $\hat{\mathbf{y}}_k$ respectively.

The continuous system for prediction of the state $\hat{\mathbf{z}}^T = (\hat{d},\ \hat{\mathbf{y}}^T)^T \in \mathbb{R}^4$ is switched to the discrete system in Eq. (4.5) at every Δt_k to update the states with measurements.

The estimation error for the hybrid system is defined as $\mathbf{e}_k \triangleq \mathbf{z}_k - \hat{\mathbf{z}}_k \in \mathbb{R}^4$, where $\mathbf{z}_k \in \mathbb{R}^4$ is the ground truth of the state.

Theorem 1 *The observer in Eqs.* (4.4) *and* (4.5) *is globally exponentially stable such that*

$$\|\mathbf{e}_{k+1}\| \leq \bar{\sigma}_k\|\mathbf{e}_k\|\exp\left(\bar{\lambda}\Delta t_k\right) \tag{4.6}$$

if Assumption 1 is satisfied and the state is constrained by $\|\hat{\mathbf{y}}\| = 1$ *and* $\hat{d} > \underline{d}$*, and if the observer gain is given by*

$$l_{k,m} = 1 - \exp\left(\frac{1}{2}\left(\gamma_m - \bar{\lambda}\right)\Delta t_k\right), \quad m \in \{1,2\} \tag{4.7}$$

where $\gamma_m \in \mathbb{R}^-$ *is defined by the user for* $\hat{d}_k$ *and* $\hat{\mathbf{y}}_k$*. The convergence rate of the system in the prediction stage in Eq.* (4.4) *is given by* $\bar{\lambda} = \lambda_{\max}(F^T + F)$*, where* $\lambda_{\max}$ *denotes the maximum eigenvalue and* $F \in \mathbb{R}^{4\times4}$ *is the Jacobian matrix in Eq.* (4.8)*.*

Proof 1 *We can write the first variation of system in Eqs.* (4.4) *and* (4.5) *as*

$$\frac{d}{dt}\begin{pmatrix}\delta\hat{d}\\ \delta\hat{\mathbf{y}}\end{pmatrix}=\begin{pmatrix}0 & -\mathbf{v}^T\\ (I-\hat{\mathbf{y}}\hat{\mathbf{y}}^T)\mathbf{v}\hat{d}^{-2} & -[\boldsymbol{\omega}]_\times+\left(\hat{\mathbf{y}}\mathbf{v}^T+\mathbf{v}^T\hat{\mathbf{y}}I\right)\hat{d}^{-1}\end{pmatrix}\begin{pmatrix}\delta\hat{d}\\ \delta\hat{\mathbf{y}}\end{pmatrix} \tag{4.8}$$

$$\begin{pmatrix}\delta\hat{d}_k^+\\ \delta\hat{\mathbf{y}}_k^+\end{pmatrix}=\begin{pmatrix}1-l_{k,1} & 0\\ 0 & I(1-l_{k,2})\end{pmatrix}\begin{pmatrix}\delta\hat{d}_k^-\\ \delta\hat{\mathbf{y}}_k^-\end{pmatrix} \tag{4.9}$$

where the virtual displacement $\delta\hat{d}\in\mathbb{R}$ *and* $\delta\hat{\mathbf{y}}\in\mathbb{R}^3$ *are infinitesimal displacements [81] at a fixed time instance, and* $I\in\mathbb{R}^{3\times3}$ *is an identity matrix. Let* $F_k\in\mathbb{R}^{4\times4}$ *denote the Jacobian matrix in Eq.* (4.9)*, and let* $\gamma=\max\{\gamma_1,\gamma_2\}$*. The convergence rate of the measurement update stage is given by*

$$\bar{\sigma}_k=\lambda_{\max}(F_k^TF_k)=(1-l_k)^2 \tag{4.10}$$

where $l_k=l_{k,m}$ *with* $\gamma_m=\gamma$*.*

Consider the observer given by Eqs. (4.4) *and* (4.5)*. The condition for the hybrid system to be contracting [81, 82, 83] is satisfied since*

$$\begin{aligned}\bar{\lambda}+\frac{\ln\bar{\sigma}_k}{\Delta t_k}&=\bar{\lambda}+\frac{\ln\exp\left(\gamma\Delta t_k-\bar{\lambda}\Delta t_k\right)}{\Delta t_k}\\ &=\gamma\end{aligned} \tag{4.11}$$

where γ *is selected to be negative. We then have*

$$\|\delta\hat{\mathbf{z}}_{k+1}\|\le\bar{\sigma}_k\|\delta\hat{\mathbf{z}}_k\|\exp(\bar{\lambda}\Delta t_k) \tag{4.12}$$

and since $\delta\hat{\mathbf{z}}_k$ *converges to zero, the estimated state converges to its true value globally exponentially fast.*

4.3.5 Uncertainty Bound on the Estimation Error

Uncertainty bound on estimation error can be analyzed by considering the disturbance η and measurement noise $\xi_k=(\xi_{d,k},\xi_{y,k}^T)^T\in\mathbb{R}^4$, where $\xi_{y,k}\in\mathbb{R}^3$ is the noise in the unit vector vision measurement. Let $\bar{\mathbf{e}}_k=\int_{\hat{\mathbf{z}}}^{\mathbf{z}}\|\delta\mathbf{x}_k\|\in\mathbb{R}$ be the quadratic bound of the observer error [81] which considers the uncertainty. Then

$$\bar{\mathbf{e}}_{k+1}\le\bar{\sigma}_k\bar{\mathbf{e}}_k\exp\left(\bar{\lambda}\Delta t_k\right)+\|\eta\Delta t_k+l_k\xi_k\|_\infty \tag{4.13}$$

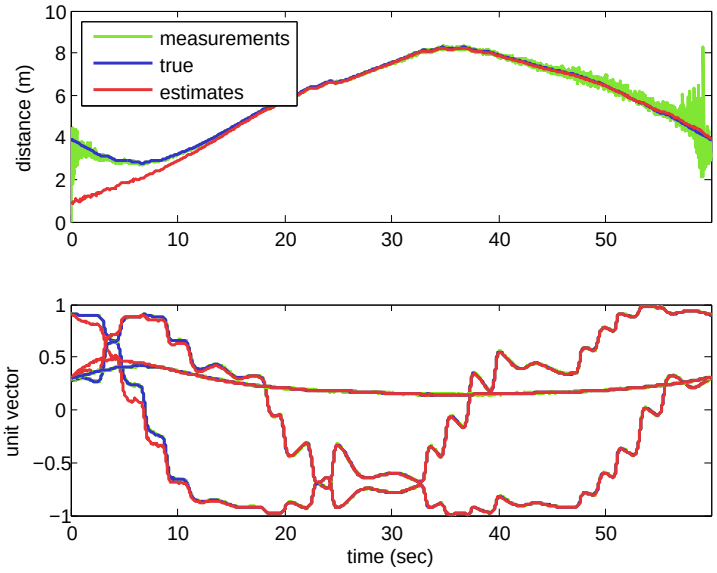


Figure 4.3: Simulation results of the landmark depth and direction estimation.

where l_k is the observer gain. The observer gain can be designed to take into account the magnitude of the noise in the measurements. When the observer gain l_k is increased, the estimation error converges to zero faster and the estimates are less sensitive to disturbance η. When the observer gain l_k is decreased, the estimates will be affected less by the measurement noise ξ_k.

4.4 Robot-Centric Localization for Boundary Estimation

In this section, we estimate the trajectory of the mower with a robot-centric system which allows us to use a linear motion model. The mower traverses a boundary in the lawn during the teaching phase. It can either follow a boundary wire temporarily set-up in the lawn or be tele-operated by a user.

Consider a model with a state vector $\mathbf{x} = ((\mathbf{x}_{b_j})^T, (\mathbf{q}_{b_j})^T)^T$, where $\mathbf{x}_{b_j} \in \mathbb{R}^3$ and $\mathbf{q}_{b_j} \in \mathbb{H}$ are the location and orientation quaternions of the mower at an instance for time-step j with respect to its current body frame.

The motion model is given by

$$\frac{d}{dt}\begin{pmatrix}\mathbf{x}_{b_j}\\ \mathbf{q}_{b_j}\end{pmatrix} = \begin{pmatrix}-[\boldsymbol{\omega}]_\times & 0\\ 0 & \frac{1}{2}\Omega(\boldsymbol{\omega})\end{pmatrix}\begin{pmatrix}\mathbf{x}_{b_j}\\ \mathbf{q}_{b_j}\end{pmatrix} - \begin{pmatrix}\mathbf{v}\\ 0\end{pmatrix} \tag{4.14}$$

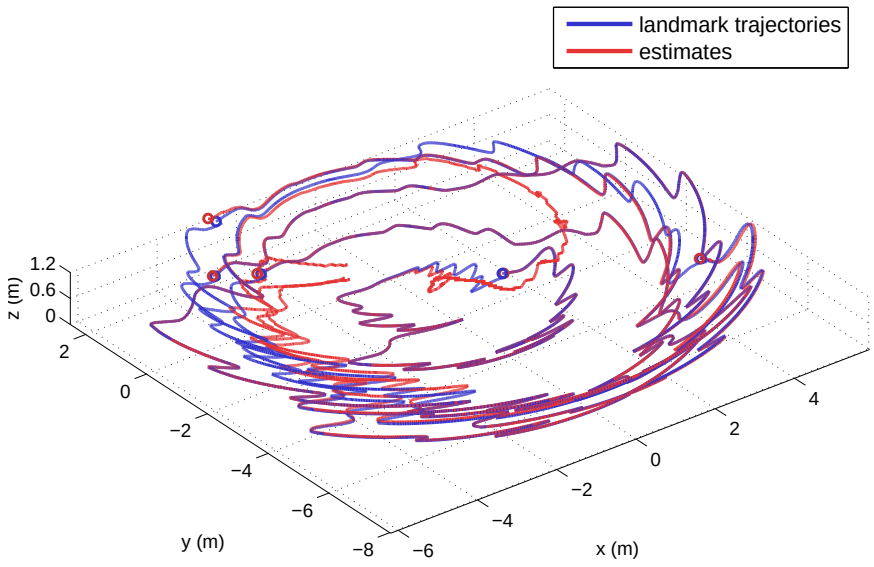


Figure 4.4: Simulation results of the estimation of landmarks in the robot's body frame.

where $\Omega(\boldsymbol{\omega}) \in \mathbb{R}^{4\times4}$ is a skew symmetric matrix

$$\Omega(\boldsymbol{\omega}) = \begin{pmatrix} 0 & \boldsymbol{\omega}^T \\ -\boldsymbol{\omega} & -[\boldsymbol{\omega}]_\times \end{pmatrix} \tag{4.15}$$

The measurement model $h(\mathbf{x})$ is a stacked vector of measurements of each landmark given by

$$\begin{aligned} h_i(\mathbf{x}) &= \mathbf{y}_i^{b_j} \\ &= \mathbf{x}_i^{b_j} / \|\mathbf{x}_i^{b_j}\|_2, \quad \forall i \in \{1, 2, \cdots, n\} \end{aligned} \tag{4.16}$$

Here, n is the number of landmarks. The direction vector measurement $\mathbf{y}_i^{b_j}$ of the i-th landmark is taken when the landmark was first observed at time-step j.

Note that $\mathbf{x}_i^{b_j} = R^T(\mathbf{q}_{b_j})(\mathbf{x}_i - \mathbf{x}_{b_j})$, where, $\mathbf{x}_i = \mathbf{y}_i d \in \mathbb{R}^3$ is the position of the i-th landmark we estimate in the robot's body frame with the nonlinear observer given by Eqs. (4.4) and (4.5).

An EKF estimator for boundary estimation is written as

$$\begin{aligned} \dot{\hat{\mathbf{x}}} &= A\hat{\mathbf{x}} + \mathbf{u} + K\left(\mathbf{h} - h(\hat{\mathbf{x}})\right) \\ \dot{P} &= AP + PA^T - PH^TV^{-1}HP + W \end{aligned} \tag{4.17}$$

where P is the covariance of the state, $A \in \mathbb{R}^{7\times7}$ is the state transition matrix in Eq. (4.14), $\mathbf{u} = \left(-\mathbf{v}^T, 0\right)^T \in$

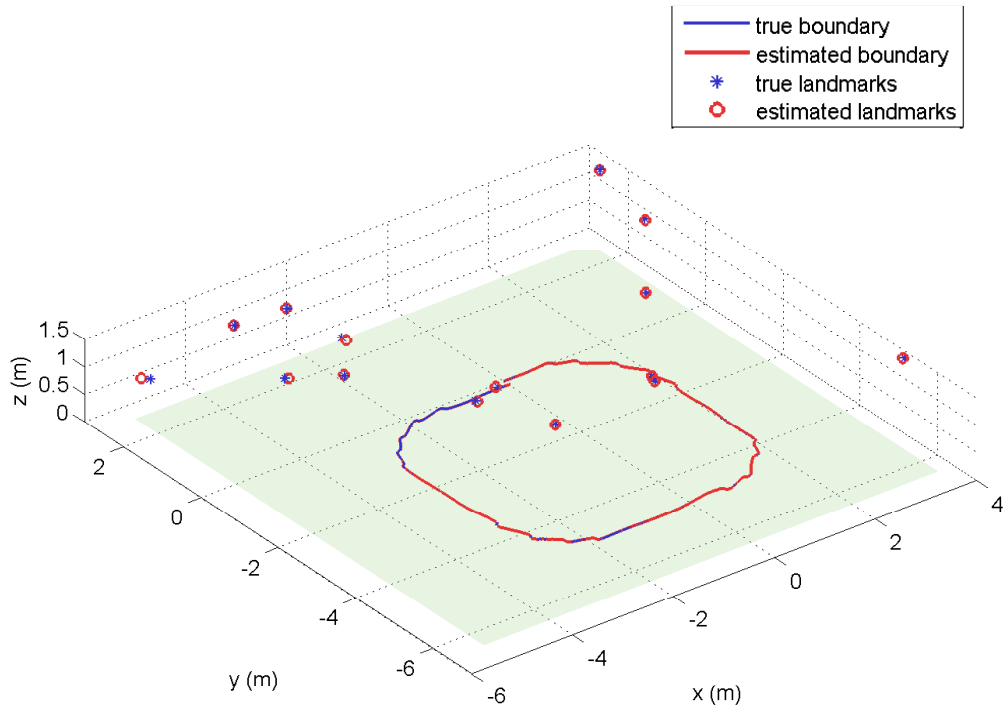


Figure 4.5: Simulation results of the boundary estimation and landmark mapping.

$\mathbb{R}^7$ is the velocity input, $\mathbf{h}$ is the vision measurement, H is the Jacobian of the measurement function $h(\hat{\mathbf{x}})$, and V and W are the covariance matrices that approximate the measurement noise and the process noise.

The estimator gain K is given by

$$K = PH^T V^{-1} \tag{4.18}$$

The location and orientation of the world reference frame with respect to the current body frame are updated by

$$\begin{aligned} \hat{\mathbf{x}}_w &= \hat{\mathbf{x}}_{b_j} + R(\hat{\mathbf{q}}_{b_j})\hat{\mathbf{x}}_w^{b_j} \\ \hat{\mathbf{q}}_w &= \hat{\mathbf{q}}_{b_j} \otimes \hat{\mathbf{q}}_w^{b_j} \end{aligned} \tag{4.19}$$

where $\otimes$ is a quaternion multiplication.

Finally, the results of mower localization $\hat{\mathbf{x}}_b^w$ and landmark mapping $\hat{\mathbf{x}}_i^w$ are represented in the world reference frame by

$$\begin{pmatrix} \hat{\mathbf{x}}_b^w \\ \hat{\mathbf{x}}_i^w \end{pmatrix} = \begin{pmatrix} -R^T(\hat{\mathbf{q}}_w)\hat{\mathbf{x}}_w \\ R^T(\hat{\mathbf{q}}_w)\left(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_w\right) \end{pmatrix} \tag{4.20}$$

The boundary can be defined in the lawn based on the history of the estimated trajectory $\hat{\mathbf{x}}_b^w$ of the mower.

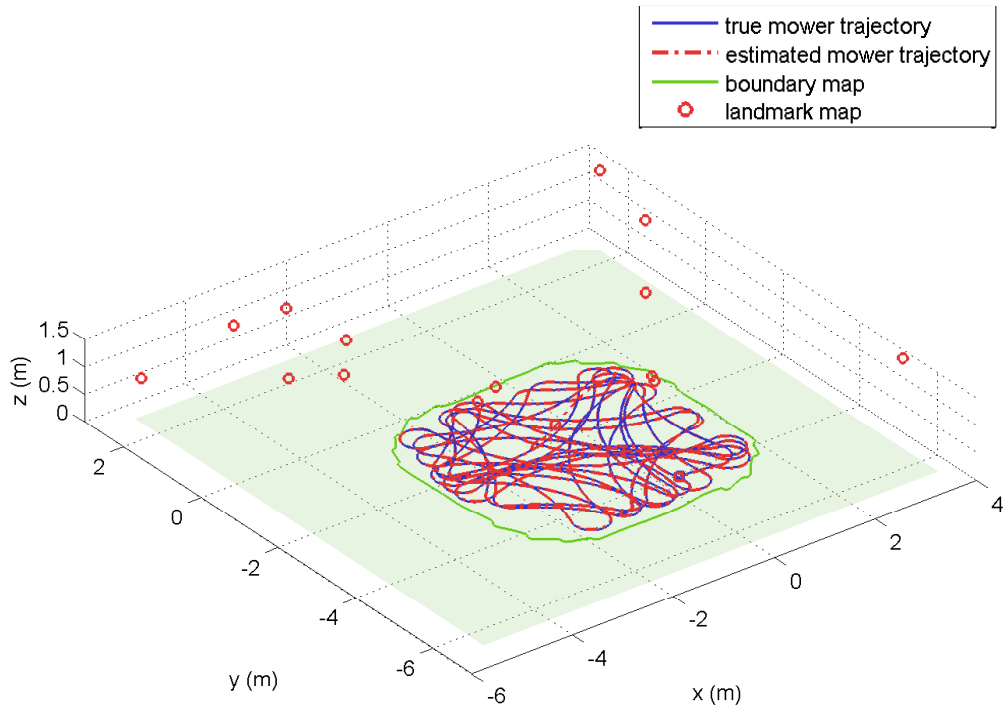


Figure 4.6: Simulation results of the robot containment based on the localization with information provided from the teaching phase.

4.5 Localization During Autonomous Mowing

During the mowing phase, localization of the autonomous mower can be performed with the information of the landmarks acquired through the teaching phase. The estimated location of the autonomous mower can be used to determine whether the mower is contained inside the estimated boundary.

The state vector of the autonomous mower is given by $((\mathbf{x}_b^w)^T, (\mathbf{q}_b^w)^T)^T$, where $\mathbf{x}_b^w \in \mathbb{R}^3$ is the current location of the mower, and $\mathbf{q}_b^w \in \mathbb{H}$ is the orientation quaternion of the mower. The pose $\mathbf{x}_b^w$ and $\mathbf{q}_b^w$ are described in the world reference frame.

The motion model of the system is given by

$$\frac{d}{dt}\begin{pmatrix}\mathbf{x}_b^w \\ \mathbf{q}_b^w\end{pmatrix} = \begin{pmatrix}R(\mathbf{q}_b^w)\mathbf{v} \\ \frac{1}{2}\mathbf{\Omega}(\omega)\mathbf{q}_b^w\end{pmatrix} \tag{4.21}$$

The measurement model $g(\mathbf{x}_b^w, \mathbf{q}_b^w)$ is a stacked vector of measurements of each landmark given by

$$\begin{aligned} g_i(\mathbf{x}_b^w, \mathbf{q}_b^w) &= \mathbf{y}_i \\ &= \mathbf{x}_i / \|\mathbf{x}_i\|_2, \quad \forall i \in \{1, 2, \cdots, n\} \end{aligned} \tag{4.22}$$

where the i-th landmark in the robot's body frame $\mathbf{x}_i$ is

$$\mathbf{x}_i = R(\mathbf{q}_w)\,(\mathbf{x}_i^w - \mathbf{x}_b^w) \tag{4.23}$$

the location of the i-th landmark $\mathbf{x}_i^w$ in the world reference frame is provided by the teaching phase results in Eq. (4.20).

The states of the autonomous mower can be estimated with an EKF estimator using the motion model in Eq. (4.21) and the measurement model in Eq. (4.22). A world-centric representation is used since the location of the landmarks $\mathbf{x}_i^w$ are provided by the teaching phase algorithm and the problem becomes a standard localization problem.

Containment of the robot can be determined by applying the estimated boundary and the estimate of the mower's current location to a point-in-polygon algorithm [84]. Consider spreading a set of rays from the mower's estimated location. The number of times the rays encounter the predefined boundary can be denoted as a winding number when the boundary is a single loop. If the winding number is odd, the mower is determined to be contained inside the boundary and it is permitted to continue mowing. If the winding number is even, the mower is outside of the boundary and mowing should be halted.

4.6 Simulation Results

Numerical simulation results are presented in this section. We distribute 15 landmark points randomly in a 3D space in a simulation. Gaussian white noise with standard deviation of 3 was added to the camera pixel measurements.

Figure 4.3 shows the estimated depth and direction vector of one of the landmarks converging to their true values. Figure 4.4 shows the trajectories of the landmarks in the robot's body frame converging to their true position. The motion of the robot and the scene can be understood when the estimates are converted to the world reference frame through Eq. (4.20). Figure 4.5 shows simulation results of boundary estimation and landmark mapping represented in the world reference frame. The estimated boundary follows the true trajectory of the mower, and the estimated landmarks converge to their true locations. The estimated trajectory of the mower is red, and the true trajectory of the mower is blue. The red circles are the estimated positions of the landmarks, and the blue stars are the true positions of the landmarks.

Figure 4.6 shows simulation results of containment during autonomous mowing. The estimated trajectory of the robot is red, and its true trajectory is blue. The boundary estimated during the teaching phase is green. The red circles are the positions of the landmarks estimated in the teaching phase. The mower

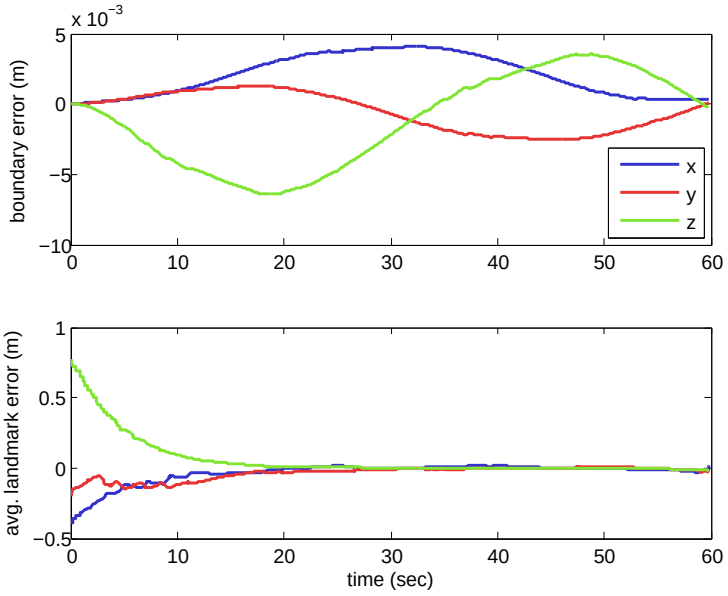


Figure 4.7: Estimation error in the location of the boundary and the average of the landmarks' positions during the teaching phase.

changes its heading direction randomly when it approaches the boundary. Threshold distance of 40cm from the boundary is used for changing the heading direction. Once the robot declares that it is inside the boundary, the mower randomly covers the given area while estimating the location of itself in the map. It is shown in Figure 4.6 that the estimated trajectory of the robot converges to its true trajectory.

Figure 4.7 shows the estimation error of the mower's location and orientation that are used to define the boundary and generate the map during the teaching phase. Figure 4.8 shows the error in the estimated location and the orientation quaternions of the mower during the mowing phase. The errors converge towards zero rapidly but oscillate continually because the mower abruptly changes its heading direction whenever it approaches the boundary.

Parameter	Value
Focal Length (f_u, f_v)	$(476.60667,\ 476.74991)$
Principal Point (u_0, v_0)	$(775.11715,\ 778.91684)$
Mirror Transformation ξ	0.92036
Skew α	0
Distortion (k_1, k_2, k_3, k_4)	$(-0.17357,\ 0.02025,$ $-0.00209,\ 0.00091,\ 0)$
Pixel Error (e_x, e_y)	$(0.50050,\ 0.51821)$

Table 4.1: Camera calibration results

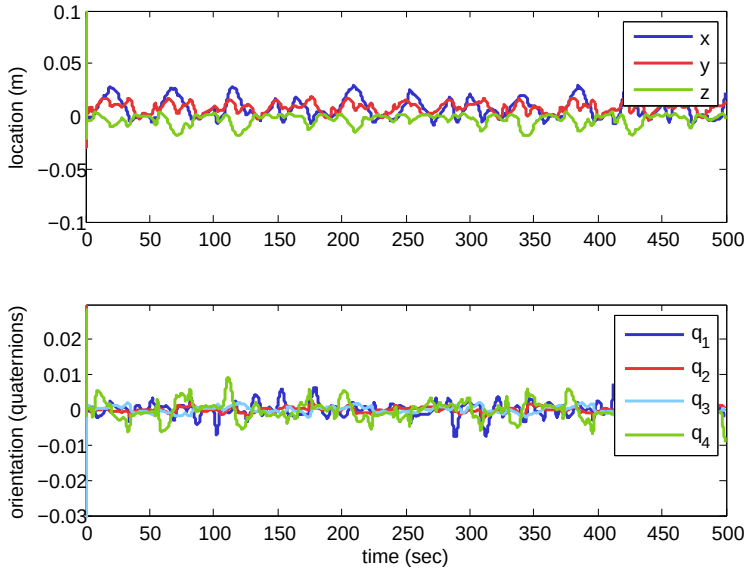


Figure 4.8: Estimation error in the location and the orientation quaternions of the mower during the mowing phase.

4.7 Teaching Phase Experiments

In this section, we demonstrate the potential of our teaching algorithm using a data set collected with our autonomous mower. Our autonomous mower is equipped with a 0-360 Panoramic Optic omnidirectional camera which can persistently capture landmarks in the scene with 360 degree field of view. A VM-100 Rugged IMU is mounted on the bottom of the camera. The vehicle speed measurements are provided from the mower.

During the teaching phase, our mower followed a boundary wire set-up in the lawn as shown in Figure 4.9. The mower made a full loop while traversing a boundary and captured 1540×1540 pixels omnidirectional camera images at 4Hz. We collected inertial measurements at 100Hz and filtered the measurements from the ground speed sensor at the same rate. Figure 4.11 shows the angular and linear velocities measured from the IMU and the vehicle speed sensor, respectively.

To demonstrate our algorithm, we manually selected 10 corners from the windows near the lawn as landmarks and tracked the points with the pyramid Lucas Kanade optical flow method [57]. Figure 4.10 shows a set of images collected using our autonomous mower. The landmarks are marked in Figure 4.10 with red dots. To extract the direction vector measurement $\mathbf{y}$, the pixel coordinates $\mathbf{p} = (p_u,\ p_v,\ 1)^T$ of each feature were transformed to normalized image coordinates with $\mathbf{p}_n = C^{-1}\mathbf{p} = (p_x,\ p_y,\ 1)^T$, where C is the camera projection matrix. The calibration parameters for our camera are shown in Table 4.1. The

Figure 4.9: Our autonomous mower following a boundary set-up in the backyard of our research building for the map estimation and the boundary teaching.

Figure 4.10: Tracking landmarks in a sequence of omnidirectional camera images.

normalized coordinates $\mathbf{p}_n$ were projected on a unit sphere through the method described in [85] which is given by

$$\mathbf{y} = \begin{pmatrix} \frac{\zeta+\sqrt{1+(1-\zeta^2)(p_x^2+p_y^2)}}{p_x^2+p_y^2+1} p_x \\ \frac{\zeta+\sqrt{1+(1-\zeta^2)(p_x^2+p_y^2)}}{p_x^2+p_y^2+1} p_y \\ \frac{\zeta+\sqrt{1+(1-\zeta^2)(p_x^2+p_y^2)}}{p_x^2+p_y^2+1} - \zeta \end{pmatrix} \tag{4.24}$$

where ζ is a mirror transformation parameter. Figure 4.12 shows the unit sphere projection of the landmarks measured with the omnidirectional camera at each time-step.

Figure 4.13 shows the depth of a landmark estimated with pseudo-measurements and the filtered direction of the unit vector vision measurement. Figure 4.14 demonstrates the landmark mapping and the boundary

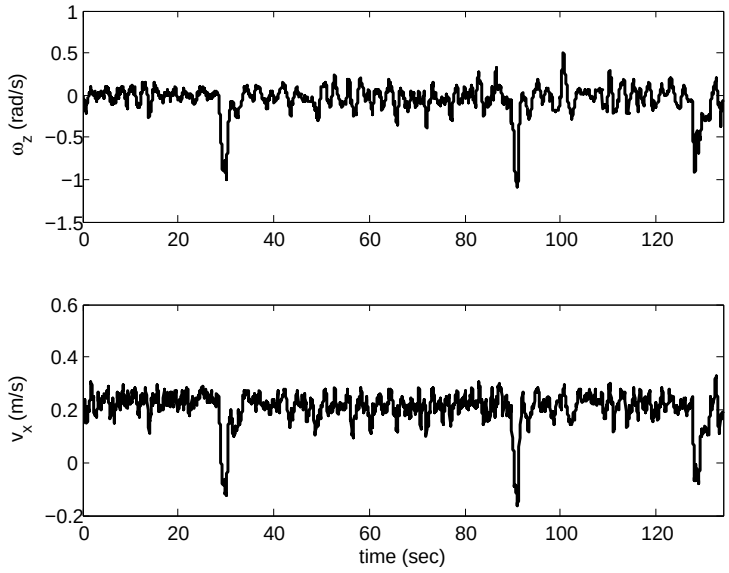


Figure 4.11: Angular and linear velocities of the mower collected during the experiments.

teaching with the data set collected using our mower. The estimated boundary, which the mower traversed, is marked with red. The blue circles are the estimated positions of the landmarks. Shape of the estimated boundary and the estimated positions of the landmarks resemble the actual configuration.

4.8 Conclusions

A vision based localization and mapping algorithm for an autonomous mower was presented in this chapter. A nonlinear observer was designed using pseudo-measurements of landmarks' depth for robot-centric landmark mapping. A boundary estimation strategy using localization results was described. We proposed to use the estimated boundary and landmark map to estimate the location of the mower for autonomous mowing. Numerical simulations illustrated the convergence of the estimates and the capability of using the estimates for containment of the mower. Preliminary experimental results showed boundary estimation and landmark mapping with a set of data collected with our autonomous mower.

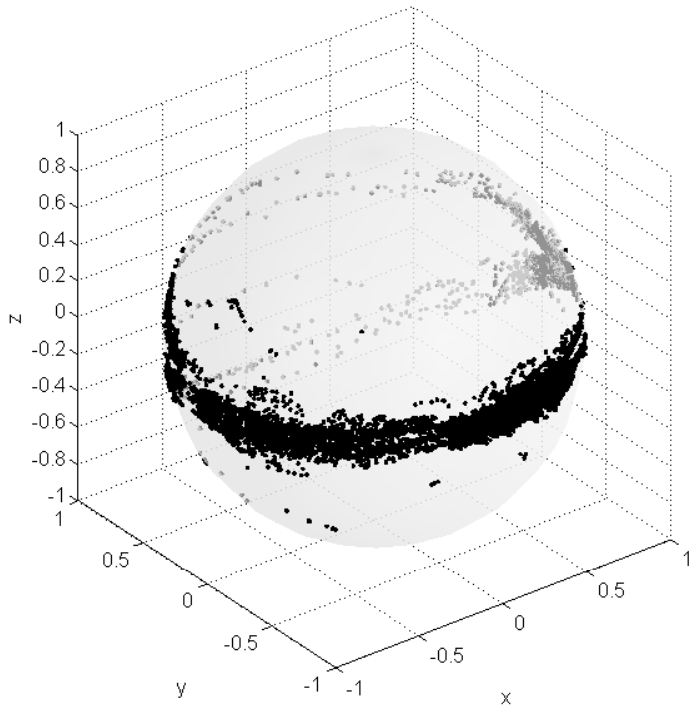


Figure 4.12: Unit sphere projection of landmark measurements at each time-step.

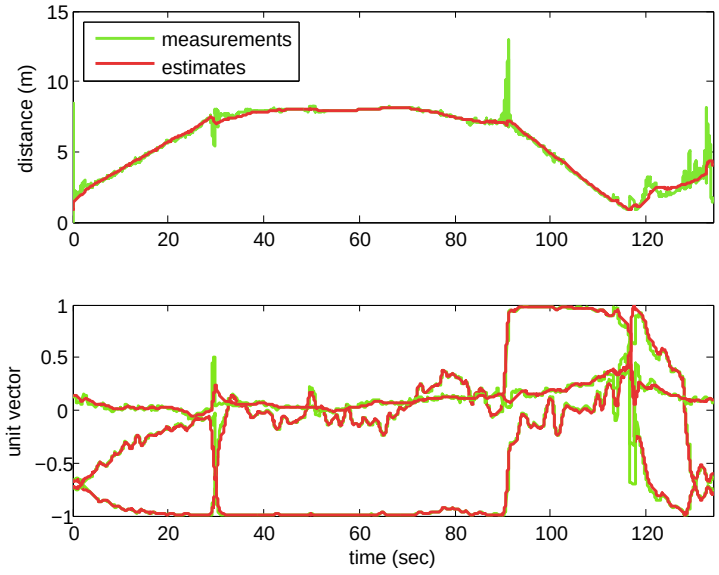


Figure 4.13: Experimental results of landmark depth and direction estimation.

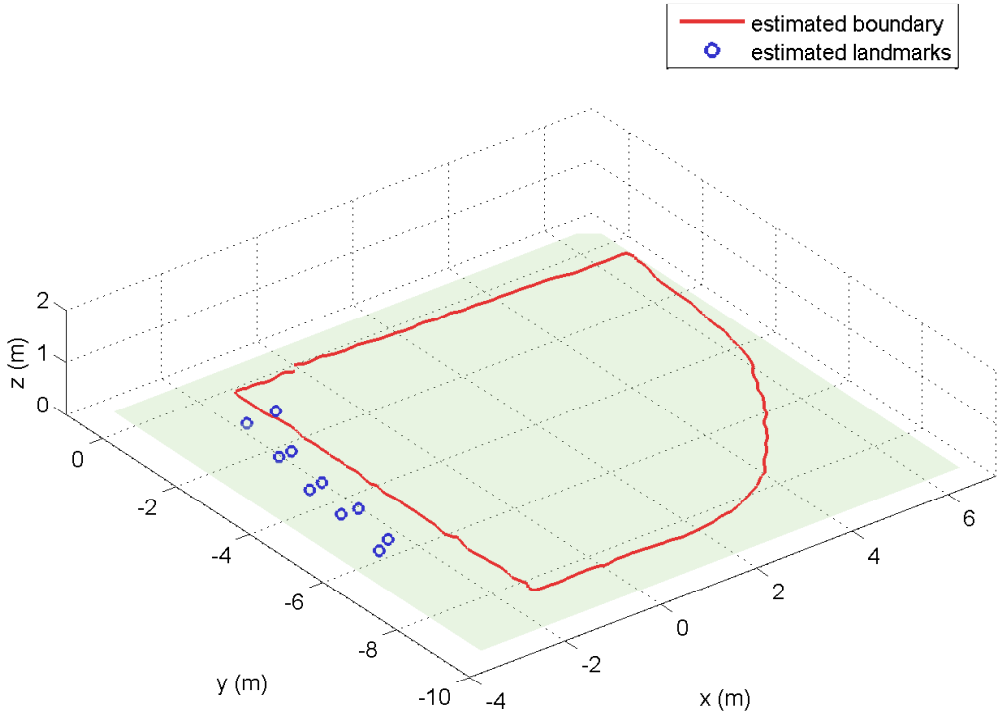


Figure 4.14: Experimental results of the boundary teaching and the landmark mapping with the data set collected using our autonomous mower.

Chapter 5

Omnidirectional-Vision-Based Estimation for Containment Detection of a Robotic Mower

5.1 Chapter Objective

In this chapter, we improve the omnidirectional-vision-based localization and mapping system with a batch estimator. We exploit the differential equation of motion of both the robot and the landmarks in the body frame with inertial measurements in contrast with prior work using robot-centric SLAM [8, 9, 10], which acquire the location of each landmark through a composition step by estimating the robot's previous pose with respect to the current pose. We analyze the performance and the degree of observability of our method with numerical simulations and present outdoor experimental results. To our knowledge, we report the first experimental results of containment detection with an omnidirectional camera for robotic mowing applications.

The rest of this chapter proceeds as follows. In Section 5.2, we give an overview of our system. In Section 5.3, we present our model for the boundary estimation and the landmark mapping. In Section 5.4, we show the model we use for the mowing location estimation and the algorithm we use to report the containment status. In Section 5.5, we describe the estimator we applied to our work. In Section 5.6, we compare the localization and mapping part of our algorithm with an existing method through numerical simulations. In Section 5.7, we present experimental results of the boundary estimation, landmark mapping, mowing location estimation, and containment detection of our robotic mower. In Section 5.8, we summarize our work with concluding remarks and future work.

5.2 Overview of the System

Our scheme for robotic mowing is separated into two phases. First, the mower generates a map of the boundary and the landmarks with the estimator described in this chapter. Our mower travels a boundary by following a wire temporarily set up over the lawn. A user can also tele-operate our mower to travel the boundary. We estimate the trajectory of the mower and define the area where it is permitted to mow.

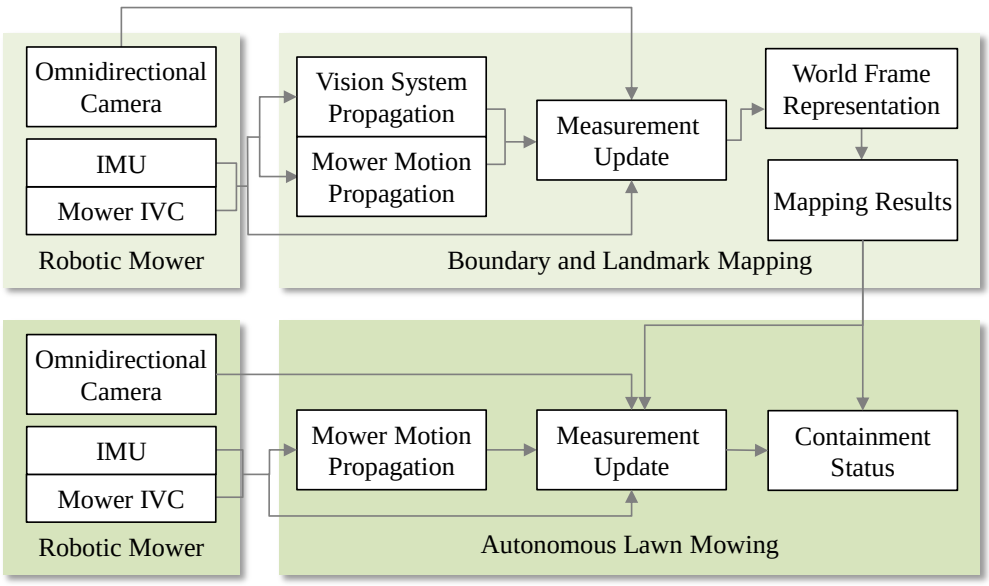


Figure 5.1: An overview of our localization and mapping algorithm and our containment detection scheme for robotic mowing.

Simultaneously, we estimate the 3D location of landmarks with our robot-centric scheme. We estimate the mowing location later with the landmark map and detect the containment of the mower with the boundary map (see Figure 5.1). We modified our John Deere robotic mower with a IDS UI-1490LE CMOS camera attached to a 360 PALNON omnidirectional camera lens, a VectorNav VN-100 Rugged IMU/magnetometer package (see Figure 5.2), and a Intel NUC5i5RYK computer for the data collection. The mower contains an IVC which computes its velocity based on encoder readings.

5.3 Boundary Estimation and Landmark Mapping

We present the model we used for the boundary estimation and landmark mapping in this section.

5.3.1 SLAM Motion Model

First, let us define the robot body frame and the world frame. We set the origin of the robot body frame on the center of the IMU which is attached to our robotic mower. The X-axis of the robot body frame points towards the front of the mower. The Z-axis of the robot body frame points upwards from the mower. The robot body frame at the initial time t_0 defines the world frame.

The estimation state of the mower consists of the location $\hat{\mathbf{p}}_b^w \in \mathbb{R}^3$ and the orientation quaternions $\hat{\mathbf{q}}_b^w \in \mathbb{H}$ of the robot body frame with respect to the world frame and the velocity $\hat{\mathbf{v}}^b \in \mathbb{R}^3$ of the mower

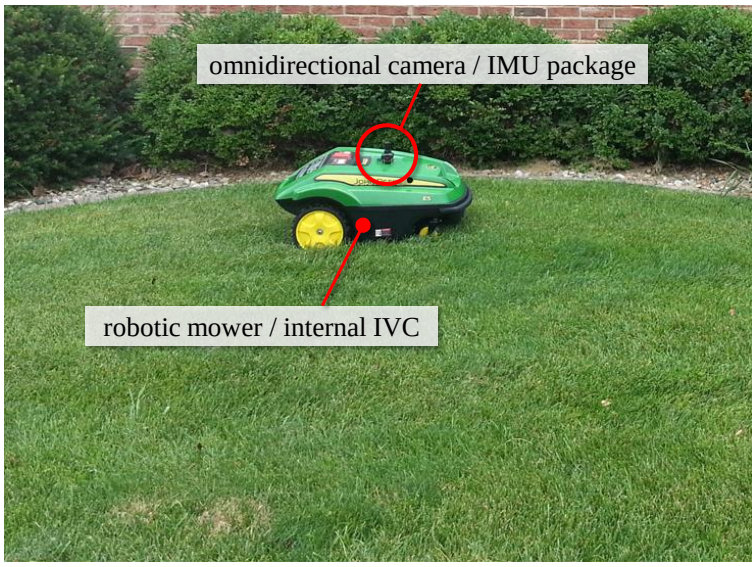


Figure 5.2: Our robotic mower is traveling a boundary in the lawn.

with respect to the current robot body frame. The state of the i-th landmark is composed of $\hat{\mathbf{x}}_i^b = (\hat{\mathbf{y}}_i^b,\ \rho_i^b)^T$, where $\hat{\mathbf{y}}_i^b = \hat{\mathbf{p}}_i^b \|\hat{\mathbf{p}}_i^b\|_2^{-1} \in \mathbb{R}^3$ is a unit vector, $\rho_i^b = \|\hat{\mathbf{p}}_i^b\|_2^{-1} \in \mathbb{R}^+$ is an inverse-distance, and $\hat{\mathbf{p}}_i^b \in \mathbb{R}^3$ is the estimated location of the landmark with respect to the robot body frame. The motion model for the mower and n landmarks is given by

$$\begin{cases} \dot{\hat{\mathbf{p}}}_b^w = R(\hat{\mathbf{q}}_b^w)\hat{\mathbf{v}}^b \\ \dot{\hat{\mathbf{q}}}_b^w = \frac{1}{2}\Omega(\boldsymbol{\omega}^b)\hat{\mathbf{q}}_b^w \\ \dot{\hat{\mathbf{v}}}^b = -[\boldsymbol{\omega}^b]_\times \hat{\mathbf{v}}^b + \mathbf{a}^b \\ \dot{\hat{\mathbf{x}}}_1^b = \mathbf{f}_1(\hat{\mathbf{y}}_1^b, \hat{\rho}_1^b, \hat{\mathbf{v}}^b, \boldsymbol{\omega}^b) \\ \quad \vdots \\ \dot{\hat{\mathbf{x}}}_n^b = \mathbf{f}_n(\hat{\mathbf{y}}_n^b, \hat{\rho}_n^b, \hat{\mathbf{v}}^b, \boldsymbol{\omega}^b) \end{cases} \tag{5.1}$$

The kinematic model of the i-th landmark $\mathbf{f}_i$ is described in Eq. (5.3). The skew symmetric matrices $[\boldsymbol{\omega}^b]_\times \in so(3)$ and $\Omega(\boldsymbol{\omega}^b) \in so(4)$ consist of the angular velocity $\boldsymbol{\omega}^b \in \mathbb{R}^3$ acquired from the IMU. The acceleration $\mathbf{a}^b \in \mathbb{R}^3$ of the mower is provided by the IMU/magnetometer package with a compensation of gravity. It is possible to include the IMU bias in the estimation by using error quaternions as shown in [38]. However, our sensor package internally compensates the gyroscope bias with a magnetometer, and the IVC in our mower provides velocity measurements which can accommodate the acceleration bias after a static calibration procedure.

5.3.2 Motion Model for Robot-Centric Mapping

The kinematic model of a landmark referenced with respect to the robot body frame in Cartesian coordinates is given in [60] as follows:

$$\dot{\hat{\mathbf{p}}}_i^b = -[\boldsymbol{\omega}^b]_\times \hat{\mathbf{p}}_i^b - \hat{\mathbf{v}}^b \tag{5.2}$$

where $\hat{\mathbf{p}}_i^b$ is the location estimate of the i-th landmark with respect to the robot body frame.

We estimate each landmark with respect to the robot body frame by using a unit vector and an inverse-distance. The kinematic model $\mathbf{f}_i(\hat{\mathbf{y}}_i^b, \hat{\rho}_i^b, \hat{\mathbf{v}}^b, \boldsymbol{\omega}^b)$ of the i-th landmark used in Eq. (5.1) with respect to the robot body frame is given in [86] as follows:

$$\begin{cases} \dot{\hat{\mathbf{y}}}_i^b = -[\boldsymbol{\omega}^b]_\times \hat{\mathbf{y}}_i^b + \left(I - \hat{\mathbf{y}}_i^b \hat{\mathbf{y}}_i^{bT}\right) \hat{\mathbf{v}}^b \hat{\rho}_i^b \\ \dot{\hat{\rho}}_i^b = \hat{\rho}_i^{b2} \hat{\mathbf{v}}^{bT} \hat{\mathbf{y}}_i^b \end{cases} \tag{5.3}$$

Note that we parameterize the landmarks with a unit vector and an inverse-distance from the robot instead of a normalized pixel coordinates and a depth along the optical axis, which we used in [43]. We are able to have a continuous parametrization of the landmarks that are acquired through an omnidirectional camera.

The state estimate ($\hat{\mathbf{y}}_i^b$, $\hat{\rho}_i^b$) of a landmark can be referenced with respect to the world frame by

$$\hat{\mathbf{p}}_i^w = \hat{\mathbf{p}}_b^w + R(\hat{\mathbf{q}}_b^w)\hat{\mathbf{y}}_i^b / \hat{\rho}_i^b \tag{5.4}$$

where $\hat{\mathbf{p}}_i^w \in \mathbb{R}^3$. By expressing the estimates of $\hat{\mathbf{y}}_i^b$ and $\hat{\rho}_i^b$ in the world frame, we are able to generate a stationary map from the estimated trajectories of the landmarks. We will analyze the advantage of our robot-centric method in Section 5.6.

5.3.3 Measurement Model

We define $\hat{\mathbf{p}}_{bi}^w \in \mathbb{R}^3$ and $\hat{\mathbf{q}}_{bi}^w \in \mathbb{H}$ as the estimates of the location and the orientation quaternions of the robot with respect to the world frame at time t_i, where t_i is the time the i-th feature is first observed and initialized in the state vector. The location estimate $\hat{\mathbf{p}}_i^{bi} \in \mathbb{R}^3$ of the i-th landmark is referenced with respect to the estimated pose $\hat{\mathbf{p}}_{bi}^w$ and $\hat{\mathbf{q}}_{bi}^w$ of the mower by

$$\hat{\mathbf{p}}_i^{bi} = R(\hat{\mathbf{q}}_{bi}^w)^T \left(\hat{\mathbf{p}}_b^w - \hat{\mathbf{p}}_{bi}^w\right) + R(\hat{\mathbf{q}}_{bi}^w)^T R(\hat{\mathbf{q}}_b^w)\hat{\mathbf{y}}_i^b / \hat{\rho}_i^b \tag{5.5}$$

We define a constant initial view unit vector $\hat{\mathbf{y}}_i^{bi} = \hat{\mathbf{p}}_i^{bi} \|\hat{\mathbf{p}}_i^{bi}\|_2^{-1} \in \mathbb{R}^3$ of the i-th landmark as the unit vector estimate $\hat{\mathbf{y}}_i^b$ of the landmark at time t_i.

The measurement model is given by

$$\mathbf{h} = \begin{pmatrix} \hat{\mathbf{y}}^{b\ T}_{1:n} & \hat{\mathbf{y}}^{bi\ T}_{1:n} & \hat{\mathbf{v}}^{bT} \end{pmatrix}^T \tag{5.6}$$

Note that we include the initial view unit vector $\hat{\mathbf{y}}^{bi}_i$ of the landmark along with the current view unit vector $\hat{\mathbf{y}}^b_i$ and formulate multiple view constraints on the mower's motion estimate ($\hat{\mathbf{p}}^b_w$, $\hat{\mathbf{q}}^b_w$) and the landmark estimate ($\hat{\mathbf{y}}^b_i$, $\hat{\rho}^b_i$).

5.4 Robotic Mowing

In this section, we present the model we use to estimate the mowing location with the map. We also describe the algorithm which we use to detect the containment of the mower.

5.4.1 Localization of the Mower

The location of the mower during autonomous mowing can be estimated with the map composed of landmarks given by Eq. (5.4). The state vector is composed of the estimate of the location $\hat{\mathbf{p}}^w_b \in \mathbb{R}^3$ and the orientation quaternions $\hat{\mathbf{q}}^w_b \in \mathbb{H}$ of the mower with respect to the world frame, and the velocity $\hat{\mathbf{v}}^b$ of the mower. The motion model is given by

$$\begin{cases} \dot{\hat{\mathbf{p}}}^w_b = R(\hat{\mathbf{q}}^w_b)\hat{\mathbf{v}}^b \\ \dot{\hat{\mathbf{q}}}^w_b = \frac{1}{2}\Omega(\boldsymbol{\omega}^b)\hat{\mathbf{q}}^w_b \\ \dot{\hat{\mathbf{v}}}^b = -[\boldsymbol{\omega}^b]_\times \hat{\mathbf{v}}^b + \mathbf{a}^b \end{cases} \tag{5.7}$$

The measurement model is given by

$$\mathbf{h} = \begin{pmatrix} \hat{\mathbf{y}}^b_{1:n} & \hat{\mathbf{v}}^b \end{pmatrix}^T \tag{5.8}$$

where $\hat{\mathbf{y}}^b_{1:n}$ is composed of the unit vectors of n landmarks. The unit vector estimates of the i-th landmark is $\hat{\mathbf{y}}^b_i = \hat{\mathbf{p}}^b_i \|\hat{\mathbf{p}}^b_i\|_2^{-1}$, and the location of the i-th landmark with respect to the robot body frame is given by

$$\hat{\mathbf{p}}^b_i = R(\hat{\mathbf{q}}^w_b)^T (\hat{\mathbf{p}}^w_i - \hat{\mathbf{p}}^w_b) \tag{5.9}$$

Here, $\hat{\mathbf{p}}^w_i$ is the estimated location of the i-th landmark in the world referenced map given by Eq. (5.4). The

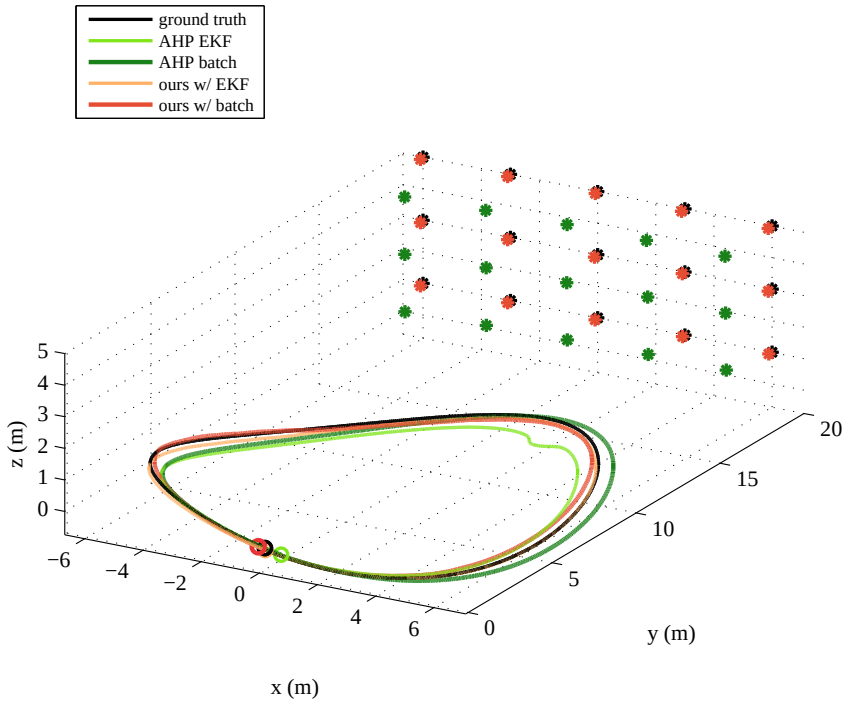


Figure 5.3: Localization and mapping simulation results.

map estimated with the model presented in Section 5.3 enables estimation of the mowing location $\hat{\mathbf{p}}_b^w$ from an arbitrary initial condition.

5.4.2 Containment Detection

Containment detection of the robotic mower can be approximated as a point-in-polygon problem [84] once we have a closed boundary map and the mowing location estimate. We use a ray casting [84] style algorithm and spread a ray from the estimated mowing location to an arbitrary direction along the X-Y plane. We report the number of intersections of the ray and the boundary projected to the X-Y plane. The mower is inside the boundary if the crossing number is odd and outside the boundary if the number is even. The mower should be allowed to continue to mow only when it is detected inside the boundary or close enough to the boundary.

5.5 Nonlinear Estimation

In this section, we describe an EKF and an extended Kalman smoother [73] we used with the models presented in Sections 5.3 and 5.4. We consider a resetting hybrid system [82] that has continuous state variables governed by differential equations and resetting laws that reset state variables at discrete time instances.

5.5.1 Continuous-Time Motion Propagation

Considering the high sampling rate of our IMU, we propagate the state estimate and the estimate covariance in continuous-time as follows:

$$\begin{aligned} \dot{\hat{\mathbf{x}}} &= \mathbf{f}(\hat{\mathbf{x}}, t) \\ \dot{P}(\hat{\mathbf{x}}, t) &= F(\hat{\mathbf{x}}, t)P(\hat{\mathbf{x}}, t) + P(\hat{\mathbf{x}}, t)F(\hat{\mathbf{x}}, t)^T + W \end{aligned} \tag{5.10}$$

where $\hat{\mathbf{x}} \in \mathbb{R}^r$ is the predicted state estimate, $P(\hat{\mathbf{x}}, t) \in \mathbb{R}^{r \times r}$ is the predicted estimate covariance, and $W \in \mathbb{R}^{r \times r}$ is the covariance approximating the process noise. Here, r is the dimension of the state vector of the motion model $\mathbf{f}(\hat{\mathbf{x}}, t)$ given by Eqs. (5.1) and (5.7) for the boundary estimation and landmark mapping and for the mowing location estimation, respectively. The Jacobian $F(\hat{\mathbf{x}}, t) \in \mathbb{R}^{r \times r}$ of the function $\mathbf{f}(\hat{\mathbf{x}}, t)$ is evaluated at $\hat{\mathbf{x}}$.

5.5.2 Discrete-Time Update of the Vision Measurements

Vision measurements are updated in discrete-time instances t_k considering the low sampling rate of our high-resolution omnidirectional camera. The subscript k is a piecewise constant index for the reset. Update of the state estimate and the estimate covariance with the measurements are given by

$$\begin{aligned} \hat{\mathbf{x}}^+ &= \hat{\mathbf{x}} + K(\hat{\mathbf{x}}, t_k)\left(\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}, t_k)\right) \\ P(\hat{\mathbf{x}}^+, t_k) &= \left(I - K(\hat{\mathbf{x}}, t_k)H(\hat{\mathbf{x}}, t_k)\right) P(\hat{\mathbf{x}}, t_k) \left(I - K(\hat{\mathbf{x}}, t_k)H(\hat{\mathbf{x}}, t_k)\right)^T + K(\hat{\mathbf{x}}, t_k)VK(\hat{\mathbf{x}}, t_k)^T \end{aligned} \tag{5.11}$$

where $\hat{\mathbf{x}}^+ \in \mathbb{R}^r$ is the updated state estimate, $P(\hat{\mathbf{x}}^+, t_k) \in \mathbb{R}^{r \times r}$ is the updated estimate covariance, $V^{s \times s}$ is the covariance approximating the measurement noise, and $\mathbf{z} \in \mathbb{R}^s$ is the vision measurement vector at time t_k. Here, s is the dimension of the measurement model $\mathbf{h}(\hat{\mathbf{x}}, t_k)$ given by Eqs. (5.6) and (5.8). The Jacobian $H(\hat{\mathbf{x}}, t_k) \in \mathbb{R}^{s \times r}$ of the function $\mathbf{h}(\hat{\mathbf{x}}, t_k)$ is evaluated at $\hat{\mathbf{x}}^+$. The estimator gain $K(\hat{\mathbf{x}}, t_k) \in \mathbb{R}^{r \times s}$ is given by

$$K(\hat{\mathbf{x}}, t_k) = P(\hat{\mathbf{x}}, t_k)H(\hat{\mathbf{x}}, t_k)^T \left(H(\hat{\mathbf{x}}, t_k)P(\hat{\mathbf{x}}, t_k)H(\hat{\mathbf{x}}, t_k)^T + V\right)^{-1} \tag{5.12}$$

The predicted state estimate and the predicted estimate covariance are corrected in the measurement update stage. It is understood that an EKF SLAM has a quadratic complexity in the size of the state [15]. We estimate the vectors of the features that are being measured and discard the features that go out of sight in order to maintain the size of the estimation state vector and reduce the computational load.

5.5.3 Batch Estimation

To provide an accurate map for robotic mowing, we use the entire batch of measurements over the time interval $[t_o,\, t_f]$ after the sequential estimation of the boundary and the map is completed. We process the estimates from Eqs. (5.10) and (5.11) with the batch stage of a Rauch-Tung-Striebel (RTS) smoother [87]. The batch stage starts at the final time t_f and proceeds backwards to t_0 recursively by

$$\begin{aligned} \dot{\hat{\mathbf{x}}}_s &= -\left(F(\hat{\mathbf{x}},t) + K(\hat{\mathbf{x}},t)\right)\left(\hat{\mathbf{x}}_s - \hat{\mathbf{x}}\right) - \mathbf{f}\left(\hat{\mathbf{x}},t\right) \\ \dot{P}_s(\hat{\mathbf{x}},t) &= -\left(F(\hat{\mathbf{x}},t) + K(\hat{\mathbf{x}},t)\right)P(\hat{\mathbf{x}},t) - P(\hat{\mathbf{x}},t)\left(F(\hat{\mathbf{x}},t) + K(\hat{\mathbf{x}},t)\right)^T + W \end{aligned} \tag{5.13}$$

where $\hat{\mathbf{x}}_s \in \mathbb{R}^r$ is the batch state estimate, and $P_s(\hat{\mathbf{x}},t) \in \mathbb{R}^{r\times r}$ is the batch estimate covariance. The estimator gain $K_s(\hat{\mathbf{x}},t) \in \mathbb{R}^{r\times s}$ is given by

$$K_s(\hat{\mathbf{x}},t) = WP(\hat{\mathbf{x}},t)^{-1} \tag{5.14}$$

After the mower travels the boundary and returns to its initial location, we recognize the location based on normalized correlations [55] of the images and trigger a loop closure. The initial location of the mower is added to the measurement model in Eq. (5.6) as a constraint, and the map of the boundary and the landmarks are corrected through the batch stage given by Eq. (5.13).

5.6 Numerical Simulation

In this section, we present results of numerical simulations and analyze the performance of our robot-centric framework in comparison to the world-centric SLAM that parameterizes landmarks with anchored homogeneous points (AHP) [17]. In Section 5.7, we will present experimental results by using real-world data acquired from the mower in an outdoor environment. During the simulation, the robot travels a 3D circular trajectory (see Figure 5.3) at an average velocity of 1 m/s and angular velocity of 0.15 rad/s and observes landmarks (marked with black dots) with 3σ Gaussian white noise in the pixel measurements. We use the normalized estimation error squared (NEES) and the root mean squared error (RMSE) to analyze the consistency and the accuracy of the estimation results. Figure 5.4 compares the NEES and the RMSE of our approach and the AHP. The results show that our approach outperforms the AHP with average RMSE of 0.2674 m for the boundary estimate, 0.0080 for the orientation quaternion, 0.0130 (1/m) for the inverse-distance, and has an average overall NEES of 0.0034. The AHP has average RMSE of 0.8948 m for the boundary estimate, 0.0143 for the orientation quaternion, 0.0371 (1/m) for the inverse-distance, and has

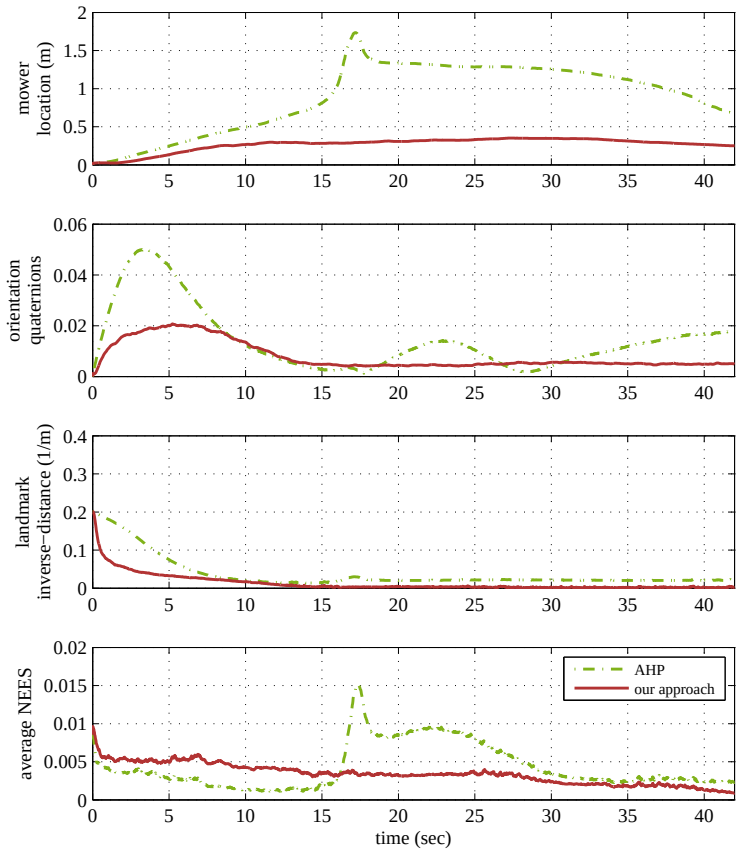


Figure 5.4: RMSE and NEES of our approach and the AHP.

an average overall NEES of 0.0043 due to the noise in the measurements.

We compute the degree of observability of the system models with the observability Gramian [69] to further analyze the difference between our approach and the AHP from a control theory perspective. We approximate our motion model given by Eq. (5.1) with a discrete-time model and use the discrete linear time-varying (LTV) observability Gramian which is given by

$$\Psi \triangleq H_{k_0}^T H_{k_0} + F_{k_0}^T H_{k_0+1}^T H_{k_0+1} F_{k_0} + \cdots + F_{k_0}^T \cdots F_{k_f-2}^T H_{k_f-1}^T H_{k_f-1} F_{k_f-2} \cdots F_{k_0} \tag{5.15}$$

for a time interval of $[t_0,\, t_f]$, where F_k and H_k are from the Jacobian matrices $F(\mathbf{x}, t_k)$ and $H(\mathbf{x}, t_k)$ of our model evaluated at the true state $\mathbf{x}$. The degree of observability is assessed with the smallest eigenvalue $|\lambda_{\min}(\Psi^{1/2})|$ during the simulation, where $\Psi^{1/2}$ is the square root of the observability Gramian [88]. Figure 5.5 shows that the matrix $\Psi^{1/2}$ of our approach has significantly higher degree of observability in comparison to the AHP. This indicates that measurement noise has a relatively small impact [69] on the estimation results of our approach.

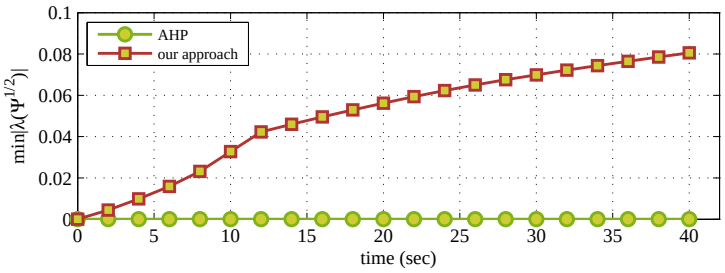


Figure 5.5: Comparison of the degree of observability.

5.7 Experimental Results

In this section, we present experimental results of containment detection with the real-world data which we collected with our robotic mower.

5.7.1 Vision Data Processing

We conducted outdoor experiments at the research park in the University of Illinois at Urbana-Champaign (UIUC) and at a front yard of a residential area. Our mower followed the boundary wire temporarily set-up in the lawn in both places. Our mower received omnidirectional camera images at 3Hz with 2500×2500 pixels resolutions, angular velocity $\boldsymbol{\omega}^b$ measurement from the IMU at 200Hz, and velocity measurement from the mower IVC, which was filtered with the IMU at 200Hz.

To extract the unit vector measurements $\mathbf{y}_{1:n}$ from the camera, we compensated the image distortion and converted the image with normalized image coordinates $\mathbf{p} = (p_x,\, p_y,\, 1)^T$. We projected the coordinates $\mathbf{p}$ onto a unit sphere [89] to get the unit vector $\mathbf{y} = (y_1,\, y_2,\, y_3)^T \in \mathbb{R}^3$ and onto a unit cylinder to get $\mathbf{y}_c \in \mathbb{R}^3$ of each pixel with the functions given by

$$\mathbf{y} = \begin{pmatrix} \frac{\zeta+\sqrt{1+(1-\zeta^2)(p_x^2+p_y^2)}}{p_x^2+p_y^2+1} p_x \\ \frac{\zeta+\sqrt{1+(1-\zeta^2)(p_x^2+p_y^2)}}{p_x^2+p_y^2+1} p_y \\ \frac{\zeta+\sqrt{1+(1-\zeta^2)(p_x^2+p_y^2)}}{p_x^2+p_y^2+1} - \zeta \end{pmatrix} \tag{5.16}$$

$$\mathbf{y}_c = \mathbf{y}\left(y_1^2 + y_2^2\right)^{-1/2}$$

where ζ is a calibration parameter for the omnidirectional camera mirror. We unwrapped the cylinder and extended each image to remove the discontinuity within each image.

We selected 18 landmarks from the research park data set (see Figure 5.6), and a pyramid KLT algorithm [57] tracked the landmarks with human assistance to reduce the drift error. From the residential area data set (see Figure 5.7), 126 landmarks were extracted and tracked without any assistance.

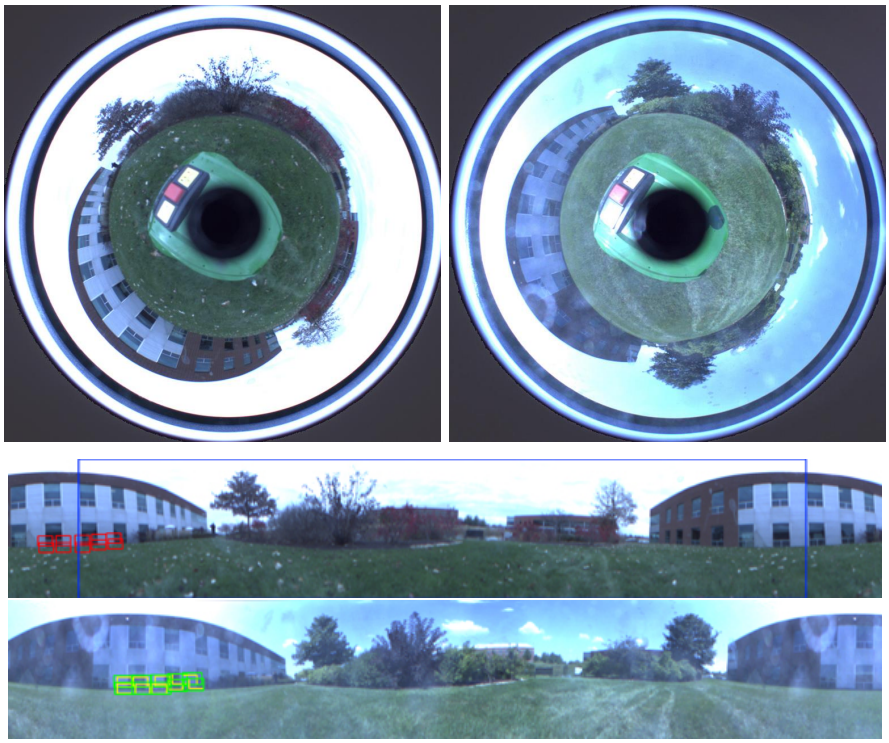


Figure 5.6: Omnidirectional camera images for the boundary estimation and landmark mapping (top left) and for the robotic mowing (top right) are projected to unwrapped cylinders. The landmarks are marked with red in the mapping image (middle row) and with yellow in the mowing image (bottom row)

After the landmarks were extracted, our algorithm computed the unit vector $\mathbf{y}$ with the pixel coordinates of the landmarks by using Eq. (5.16) for the measurement models given by Eqs. (5.6) and (5.8).

5.7.2 Containment Detection

The boundary was defined with the history of the mowing location estimate, and the inverse-distance to each landmark was estimated with the model presented in Section 5.3. The estimates of the landmarks are shown with respect to the world frame through Eq. (5.4). The map was provided to estimate the mowing location and report the containment status. The location and the orientation of the mower were estimated with the model presented in Section 5.4.1. The status of containment was reported with the ray-casting style algorithm described in Section 5.4.2 based on the boundary map and the estimates of the mowing location.

Figures 5.8 and 5.9 show the experimental results of boundary estimation and landmark mapping, mowing location estimation, and containment detection in a lawn near a building at the research park in the UIUC. The mower was contained inside the boundary during the mowing experiment by randomly changing the heading direction with a bang bang control style algorithm whenever the mower approached the boundary wire. The results show more false detections in one of the corners of the mowing area. The dilution of

Figure 5.7: Tracking on unwrapped cylinder image data from a front yard in a residential area.

precision (DOP) [90] overlaid on the results shows that the expected accuracy of the estimate is low in the corresponding region due to the lack of landmarks.

Figure 5.10 shows the experimental results of containment detection in a residential area. The mower's location was estimated along the boundary with the data used for boundary estimation and landmark mapping. We set an arbitrary initial condition for the location estimation and the location estimate quickly converged towards the starting location in the beginning as shown in Figures 5.10 and 5.11. The containment status was reported with a 10 cm margin. Figure 5.11 shows the boundary estimate, the estimate of the mowing location over the boundary, and the error between the two. The estimated error of the mowing location after one lap was 32.4 cm without loop closing. The average error between the estimate of the boundary and the mowing location along the X-Y plane, which was 45.7 cm, caused an error in the containment detection results. Although the results show that our algorithm is capable of reporting significant drift from the boundary, the accuracy of the estimation should be improved to report the drift more precisely. Using artificial landmarks can be an option to increase the DOP in the area where we do not have many landmarks and improve the accuracy of the estimation.

5.8 Conclusions

In this chapter, we have presented an omnidirectional-vision-based system for boundary estimation and landmark mapping, mowing location estimation, and containment detection of a robotic mower. As we showed with numerical simulation results, our method of using robot-centric mapping with initial view measurements can outperform existing methods due to its relatively high degree of observability. The outdoor experimental results demonstrated the effectiveness of our overall system although the accuracy of the estimation should be improved further.

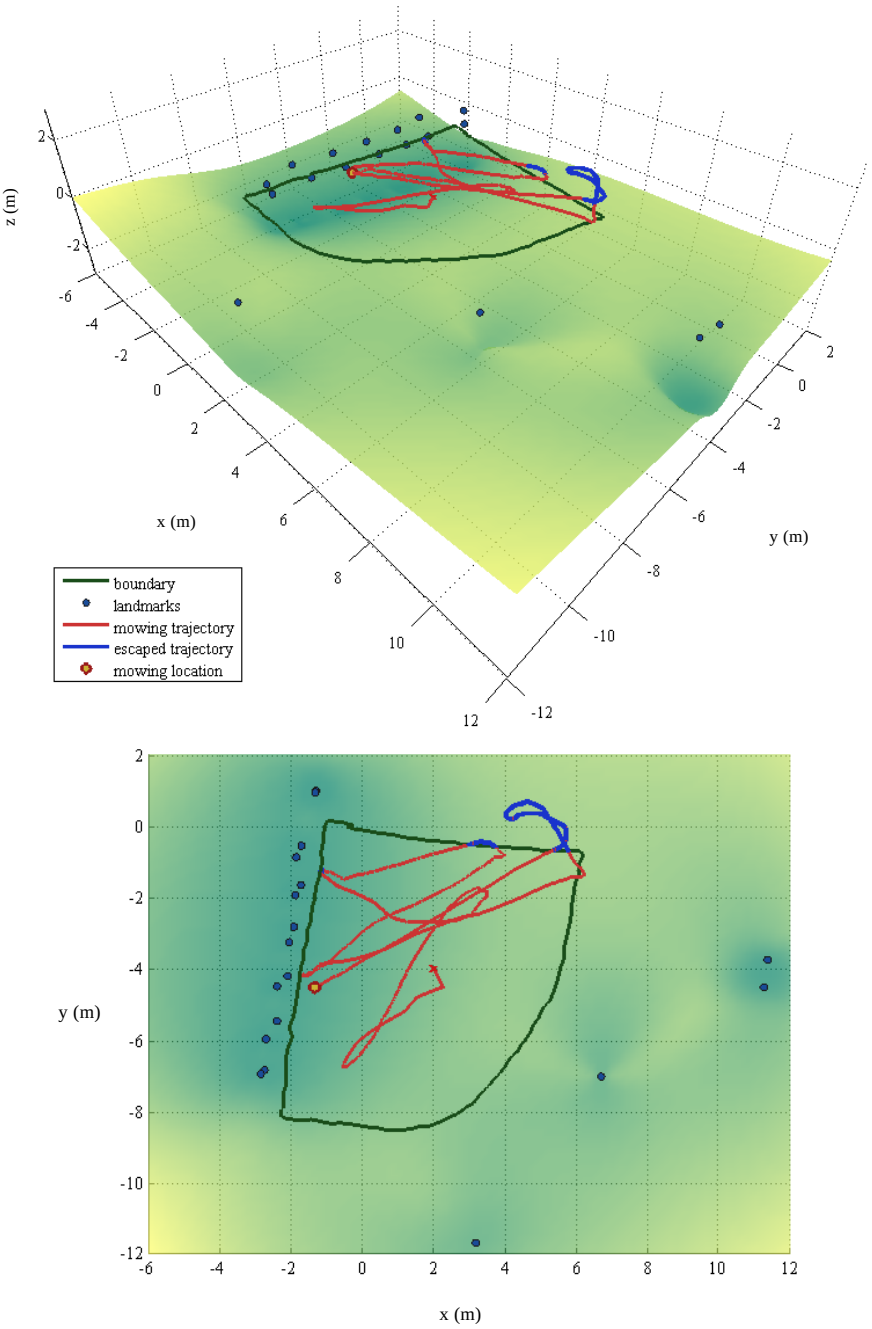


Figure 5.8: Experimental results of mowing location estimation and containment detection with mapping results from the UIUC research park (see Figure 5.6). The initial estimates (marked with red x) of the mowing location given in the middle of the lawn quickly converges to the true location (lower left corner). The DOP analysis results (green) overlaid on the map show the expected accuracy of the mowing location estimation. The accuracy degrades in the region where the DOP is high (bright).

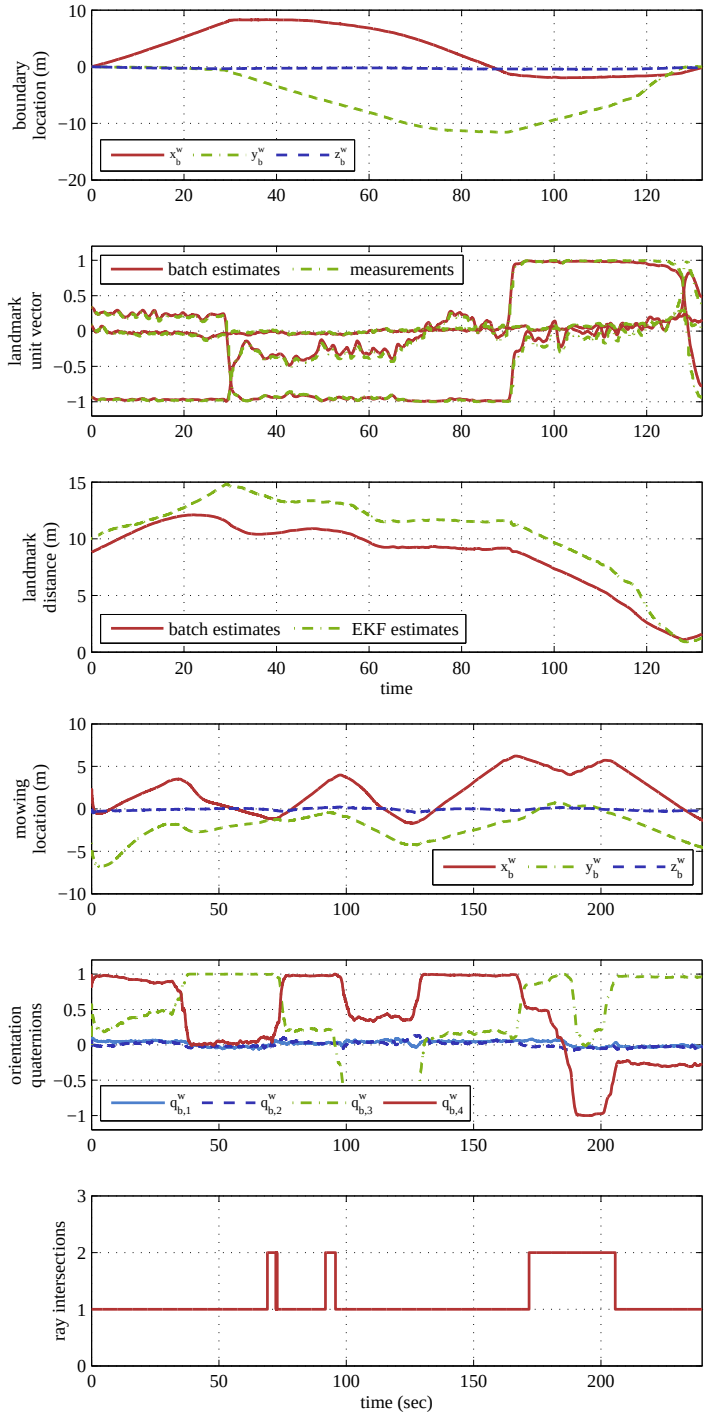


Figure 5.9: Experimental results of the boundary estimation, landmark mapping, mowing location estimation, and containment detection.

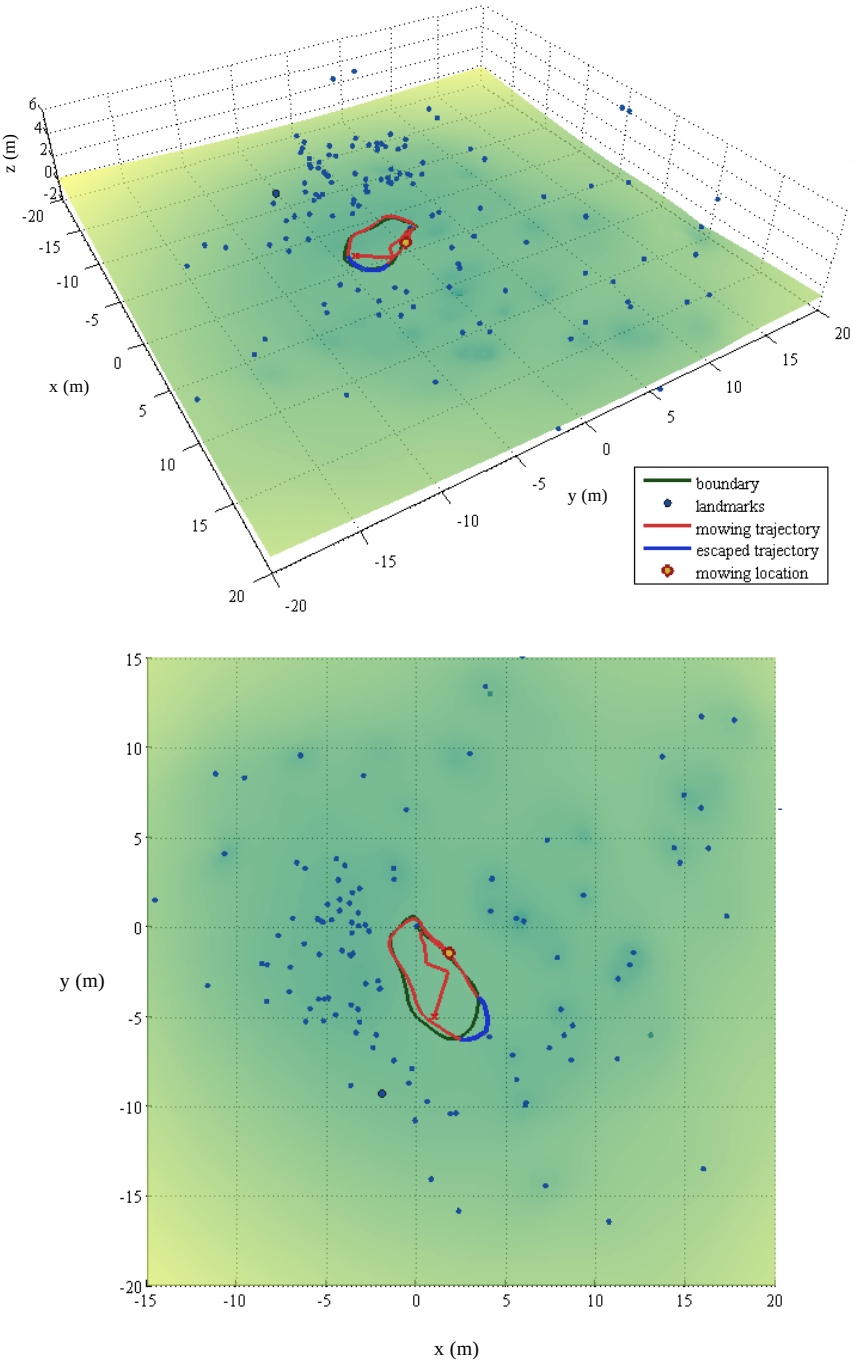


Figure 5.10: Experimental results of mowing location estimation and containment detection from a residential area (see Figure 5.7).

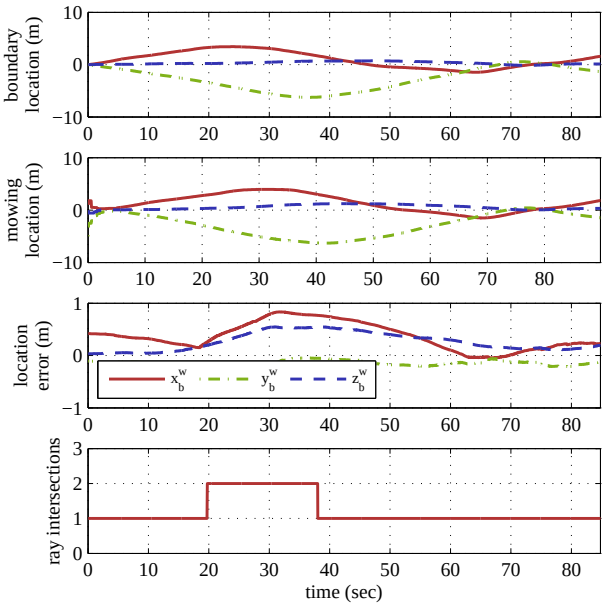


Figure 5.11: Location estimation error during boundary following experiments and the containment detection results.

Chapter 6

Convex Optimization Based SLAM

6.1 Chapter Objective

In this chapter, we present a convex optimization based simultaneous localization and mapping (SLAM) algorithm. We design a single camera based SLAM estimator using contraction analysis. We guarantee the stability of the estimator and minimize the mean squared estimation error.

Convex optimization has been studied in the SLAM literature [91, 92]. It is shown in [91] that a simplified SLAM problem can be approximated as a nonlinear least squares problem when relative position of features from the robot and odometry measurements are provided. In [92], a least squares pose SLAM problem is approximated as a convex optimization problem by introducing new variables. In contrast, we solve a single camera based SLAM problem by using a convex optimization based SLAM estimator. Linear matrix inequality (LMI) constraints for the convex optimization is derived using contraction theory to guarantee the stability of the estimator. Stochastic contraction analysis [93, 94] is used to derive the objective function that reduces the mean-squared error of the state estimate. The structure and the derivation of the nonlinear estimator used in our work is strongly influenced by the LMI state-dependent algebraic Riccati equation (SDARE) estimator [93]. In contrast to the LMI-SDARE, we use Jacobian matrices of the motion model and the measurement model instead of state dependent coefficient (SDC) matrices. Therefore, we relax an assumption in the LMI-SDARE estimator.

We employ the robot-centric feature mapping framework which we presented in Section 5. We consider an omnidirectional camera as our primary sensor but we can also directly apply our method to monocular camera based systems. We exploit the differential equation of motion of each feature in the robot body frame in contrast with existing work using robot-centric SLAM [9, 10]. We estimate the state of the robot and the features by formulating a measurement model with multiple views of point features. In Chapters 3 and 5, we showed world-centric localization and robot-centric mapping results using an EKF. In this chapter, we apply a convex optimization based nonlinear estimator to our single camera based robot-centric system model [45] and validate the effectiveness of our methods through numerical simulations. We report the first

result of convex optimization based SLAM using a single camera as a primary sensor.

The rest of this chapter proceeds as follows. In Section 6.2, we give an overview of our localization and robot-centric mapping system model. In Section 6.3, we describe the convex optimization based estimator, specifically for the single camera based localization and robot-centric mapping. In Section 6.4, we compare the performance of the convex optimization based estimator with an EKF estimator through numerical simulations. In Section 6.5, we summarize our work with concluding remarks.

6.2 Overview of Our System Model

In this section, we present the motion model for the 6 DOF world-centric localization of the robot and the 3D robot-centric mapping of point features. We also describe the measurement model with different views of point features.

We denote the state of our system by $\mathbf{x} \equiv ((\mathbf{p}_b^w)^T, (\mathbf{q}_b^w)^T, (\mathbf{x}_{1:n}^b)^T)^T \in \mathbb{R}^{7+4n}$, where $\mathbf{p}_b^w \in \mathbb{R}^3$ and $\mathbf{q}_b^w \in \mathbb{H}$ are the location and the orientation quaternion of the robot body frame in the world frame, respectively. The vector $\mathbf{x}_{1:n}^b \in \mathbb{R}^{4n}$ denotes n features, where $\mathbf{x}_i^b \equiv (\mathbf{y}_i^b, \rho_i^b)^T \in \mathbb{R}^4$ represents the i-th point feature with respect to the robot body frame (see Figure 6.1). The unit vector $\mathbf{y}_i^b = \mathbf{p}_i^b \|\mathbf{p}_i^b\|_2^{-1} \in \mathbb{R}^3$ is the direction, $\rho_i^b = \|\mathbf{p}_i^b\|_2^{-1} \in \mathbb{R}^+$ is the inverse-distance, and $\mathbf{p}_i^b \in \mathbb{R}^3$ is the location of the i-th feature in the robot body frame, respectively.

Consider our dynamic system represented by an Itô stochastic differential model [94] and our measurement model which are given by

$$\begin{aligned} d\mathbf{x} &= \mathbf{f}(\mathbf{x}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)dt + B(\mathbf{x}, t)dW_1(t) \\ \mathbf{z} &= \mathbf{h}(\mathbf{x}, t) + D(\mathbf{x}, t)\boldsymbol{\nu}(t) \end{aligned} \tag{6.1}$$

where $\mathbf{z}(t) \in \mathbb{R}^m$ is the vision measurement, and $\mathbf{h}(\mathbf{x}, t) : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^m$ is the measurement model. The vector $\boldsymbol{\nu}(t) \in \mathbb{R}^m$ is white noise of $dW_2(t) = \boldsymbol{\nu}(t)dt$, $W_1(t) \in \mathbb{R}^n$ and $W_2(t) \in \mathbb{R}^m$ are independent Wiener processes, $B(\mathbf{x}, t) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^{n \times n}$, and $D(\mathbf{x}, t) : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^{m \times m}$. The vectors $\mathbf{v}^b \in \mathbb{R}^3$ and $\boldsymbol{\omega}^b \in \mathbb{R}^3$ are the velocity and angular velocity, respectively.

The motion model $\mathbf{f}(\mathbf{x}, t) : \mathbb{R}^{7+4n} \times \mathbb{R} \to \mathbb{R}^{7+4n}$ of the robot and n features is given in Section 5 as follows:

$$\mathbf{f}(\mathbf{x}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) = \begin{pmatrix} \dot{\mathbf{p}}_b^w \\ \dot{\mathbf{q}}_b^w \\ \dot{\mathbf{x}}_1^b \\ \vdots \\ \dot{\mathbf{x}}_n^b \end{pmatrix} = \begin{pmatrix} R(\mathbf{q}_b^w)\mathbf{v}^b \\ \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}^b)\mathbf{q}_b^w \\ \boldsymbol{f}_1(\mathbf{x}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)) \\ \vdots \\ \boldsymbol{f}_n(\mathbf{x}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)) \end{pmatrix} \tag{6.2}$$

The skew-symmetric matrix $[\boldsymbol{\omega}^b]_\times \in so(3)$ is formed with the angular velocity $\boldsymbol{\omega}^b$, and $\Omega(\boldsymbol{\omega}^b)$ is given by

$$\Omega(\boldsymbol{\omega}^b) \equiv \begin{pmatrix} -[\boldsymbol{\omega}^b]_\times & \boldsymbol{\omega}^b \\ -\left(\boldsymbol{\omega}^b\right)^T & 0 \end{pmatrix} \tag{6.3}$$

The kinematic model $\boldsymbol{f}_i(\mathbf{x}_i^b,\, \mathbf{v}^b,\, \boldsymbol{\omega}^b,\, t)$ of the i-th feature used in Eq. (6.2) with respect to the robot body frame is given in Section 5 as follows:

$$\begin{aligned} &\boldsymbol{f}_i(\mathbf{x}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)) \\ &= \begin{pmatrix} \dot{\mathbf{y}}_i^b \\ \dot{\rho}_i^b \end{pmatrix} = \begin{pmatrix} -[\boldsymbol{\omega}^b]_\times \mathbf{y}_i^b + \left(I - \mathbf{y}_i^b \left(\mathbf{y}_i^b\right)^T\right) \mathbf{v}^b \\ \left(\rho_i^b\right)^2 \left(\mathbf{v}^b\right)^T \mathbf{y}_i^b \end{pmatrix} \end{aligned} \tag{6.4}$$

We use this representation to parameterize each feature without any discontinuity in the direction of the feature considering the case of using an omnidirectional camera, but we can also directly apply this parametrization to an ordinary monocular camera [86].

We formulate the measurement model as

$$\mathbf{h}(\mathbf{x}, t) = \begin{pmatrix} {\mathbf{y}_{1:n}^b}^T & {\mathbf{y}_{1:n}^{bi}}^T \end{pmatrix}^T \tag{6.5}$$

where $\mathbf{y}_i^{bi}$ is a unit vector of the i-th feature stored as a constant vector upon its first observation. The vector $\mathbf{y}_i^{bi}$ is identical to the unit vector measurement $\mathbf{y}_i^b$ of the i-th feature at time t_i .

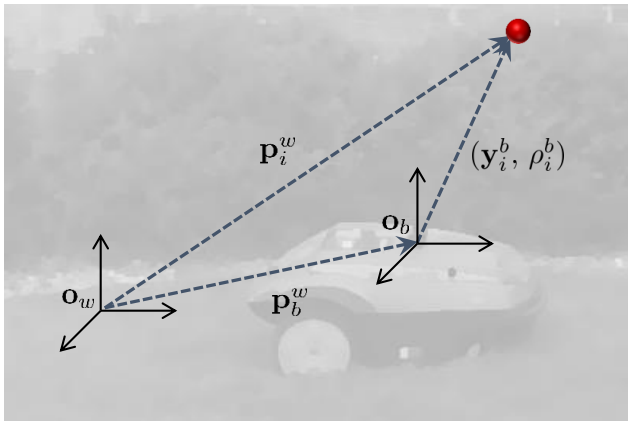


Figure 6.1: A feature seen from an omnidirectional camera and coordinate relationships.

6.3 Estimator Design and Stability Analysis

In this section, we design a nonlinear estimator to estimate the state $\mathbf{x}(t)$ of the system in Eq. (6.1). We denote the state estimate by $\hat{\mathbf{x}} \equiv ((\hat{\mathbf{p}}_b^w)^T, (\hat{\mathbf{q}}_b^w)^T, (\hat{\mathbf{x}}_{1:n}^b)^T)^T \in \mathbb{R}^{7+4n}$, where $\hat{\mathbf{p}}_b^w \in \mathbb{R}^3$ and $\hat{\mathbf{q}}_b^w \in \mathbb{H}$ are the estimate of the robot's location and orientation quaternion in the world frame, respectively. The vector $\hat{\mathbf{x}}_{1:n}^b \in \mathbb{R}^{4n}$ denotes the estimate of n features in the robot body frame. The vector $\hat{\mathbf{x}}_i^b \equiv (\hat{\mathbf{y}}_i^b,\ \hat{\rho}_i^b)^T \in \mathbb{R}^3$ represents the i-th point feature, where $\hat{\mathbf{y}}_i^b = \hat{\mathbf{p}}_i^b \|\hat{\mathbf{p}}_i^b\|_2^{-1} \in \mathbb{R}^3$ is a unit vector estimate, and $\hat{\rho}_i^b = \|\hat{\mathbf{p}}_i^b\|_2^{-1} \in \mathbb{R}^+$ is a inverse-distance estimate. The vector $\hat{\mathbf{p}}_i^b \in \mathbb{R}^3$ is the estimated location of the i-th feature with respect to the robot body frame.

6.3.1 Estimator Design

We design the estimator as

$$d\hat{\mathbf{x}} = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)dt + K(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)\big(\mathbf{h}(\mathbf{x}, t) - \mathbf{h}(\hat{\mathbf{x}}, t)\big)dt \tag{6.6}$$

The estimator gain in Eq. (6.6) is given by

$$K(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) = P(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)H^T(\hat{\mathbf{x}}, t)R(\hat{\mathbf{x}}, t)^{-1} \tag{6.7}$$

where $H(\hat{\mathbf{x}}, t) = \frac{\partial \mathbf{h}(\hat{\mathbf{x}}, t)}{\partial \hat{\mathbf{x}}}$ is the Jacobian of the measurement model $\mathbf{h}(\hat{\mathbf{x}}, t)$, and $R(\hat{\mathbf{x}}, t) = D(\hat{\mathbf{x}}, t)D^T(\hat{\mathbf{x}}, t)$ is a positive definite approximation of the measurement noise covariance matrix. The matrix $P(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) \in$

$\mathbb{R}^{(7+4n)\times(7+4n)}$ is a positive definite symmetric matrix and a solution to the equation given by

$$\begin{aligned} dP(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) = \Big(& F(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) P(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) + P(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) F^T(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) + 2\alpha P(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) \\ & - P(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) \left(-2\kappa I + H^T(\hat{\mathbf{x}}, t) R^{-1}(\hat{\mathbf{x}}, t) H(\hat{\mathbf{x}}, t) \right) P(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) \Big) dt \end{aligned} \tag{6.8}$$

where $\alpha > 0$ is a scalar tuning parameter, $\kappa > 0$ is a scalar solution to an optimization problem that we shall present in Section 6.3.3, and $I \in \mathbb{R}^{7+4n\times 7+4n}$ is an identity matrix. Here, n is the number of features in the state estimate. By considering Jacobian matrices, $F(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) = \frac{\partial \mathbf{f}(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)}{\partial \hat{\mathbf{x}}}$ to represent the motion model and $H(\hat{\mathbf{x}}, t) = \frac{\partial \mathbf{h}(\hat{\mathbf{x}}, t)}{\partial \hat{\mathbf{x}}}$ for the measurement model, we relax an assumption presented in the approach that uses state dependent coefficient (SDC) matrices [93]. We find the symmetric matrix $P(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)$ and compute the estimator gain $K(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)$ as shown in Eq. (6.7). The motion model for the robot and n features is given by

$$\mathbf{f}(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) = \begin{pmatrix} \dot{\hat{\mathbf{p}}}_b^w \\ \dot{\hat{\mathbf{q}}}_b^w \\ \dot{\hat{\mathbf{x}}}_1^b \\ \vdots \\ \dot{\hat{\mathbf{x}}}_n^b \end{pmatrix} = \begin{pmatrix} R(\mathbf{q}_b^w)\mathbf{v}^b \\ \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}^b)\mathbf{q}_b^w \\ \boldsymbol{f}_1(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) \\ \vdots \\ \boldsymbol{f}_n(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) \end{pmatrix} \tag{6.9}$$

where

$$\begin{aligned} & \boldsymbol{f}_i(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) \\ & = \begin{pmatrix} \dot{\hat{\mathbf{y}}}_i^b \\ \dot{\hat{\rho}}_i^b \end{pmatrix} = \begin{pmatrix} -[\boldsymbol{\omega}^b]_\times \hat{\mathbf{y}}_i^b + \left(I - \hat{\mathbf{y}}_i^b \left(\hat{\mathbf{y}}_i^b \right)^T \right) \mathbf{v}^b \\ \left(\hat{\rho}_i^b \right)^2 \left(\hat{\mathbf{v}}^b \right)^T \hat{\mathbf{y}}_i^b \end{pmatrix} \end{aligned} \tag{6.10}$$

We represent the estimate of each feature by using a unit vector $\hat{\mathbf{y}}_i^b$ and an inverse-distance $\hat{\rho}_i^b$ from the robot body frame as shown in [45, 86]. It is possible to use the error quaternions and include the IMU bias and the acceleration in the state estimate as shown in [38]. However, the gyroscope bias is removed from our sensor package output using a magnetometer, and we can neglect the acceleration since we consider using a robot that acquires velocity information.

We generate a stationary map derived from the estimated motion of the features $\hat{\mathbf{x}}_i^b \equiv (\hat{\mathbf{y}}_i^b, \hat{\rho}_i^b)$ with respect to the robot body frame. We reference the estimate of a feature with respect to the world frame by using

$$\hat{\mathbf{p}}_i^w = \hat{\mathbf{p}}_b^w + R(\hat{\mathbf{q}}_b^w)\hat{\mathbf{y}}_i^b/\hat{\rho}_i^b \tag{6.11}$$

where $\hat{\mathbf{p}}_i^w \in \mathbb{R}^3$ is the location of the feature with respect to the world frame. The advantage of the localization and robot-centric mapping method is analyzed in Chapters 3 and 5.

The measurement model for our estimation system is given by

$$\mathbf{h}(\hat{\mathbf{x}}, t) = \begin{pmatrix} \hat{\mathbf{y}}_{1:n}^{b\ T} & \hat{\mathbf{y}}_{1:n}^{bi\ T} \end{pmatrix}^T \tag{6.12}$$

Let us denote t_i as the time instance the i-th feature is measured for the first time. We define $\hat{\mathbf{p}}_{bi}^w \in \mathbb{R}^3$ as the robot's location estimate and $\hat{\mathbf{q}}_{bi}^w \in \mathbb{H}$ as the robot's quaternion orientation estimate, both referenced with respect to the world frame at time t_i. The estimate of the i-th feature $\hat{\mathbf{p}}_i^{bi} \in \mathbb{R}^3$ referenced with respect to the estimated pose $(\hat{\mathbf{p}}_{bi}^w, \hat{\mathbf{q}}_{bi}^w)$ of the robot is given by

$$\hat{\mathbf{p}}_i^{bi} = R(\mathbf{q}_{bi}^w)^T (\hat{\mathbf{p}}_b^w - \hat{\mathbf{p}}_{bi}^w) + R(\mathbf{q}_{bi}^w)^T R(\hat{\mathbf{q}}_b^w)\hat{\mathbf{y}}_i^b/\hat{\rho}_i^b \tag{6.13}$$

We define the initial view estimate $\hat{\mathbf{y}}_i^{bi} = \hat{\mathbf{p}}_i^{bi}\|\hat{\mathbf{p}}_i^{bi}\|_2^{-1} \in \mathbb{R}^3$ of the i-th feature as a constant vector that is equivalent to the unit vector estimate $\hat{\mathbf{y}}_i^b$ of the feature at time t_i.

6.3.2 Estimator Stability with Contraction Analysis

We analyze the stability of the estimator described in Section 6.3.1 using contraction theory. The trajectories of the system in Eq. (6.6) without noise and disturbance is given by

$$\begin{aligned} \dot{\mathbf{s}} &= \mathbf{f}_{CL}(\mathbf{s}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) \\ &= \mathbf{f}(\mathbf{s}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) + K(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)\big(\mathbf{h}(\mathbf{x}, t) - \mathbf{h}(\mathbf{s}, t)\big) \end{aligned} \tag{6.14}$$

where the particular solutions of the virtual system in Eq. (6.14) are $\mathbf{s} = \mathbf{x}$ and $\mathbf{s} = \hat{\mathbf{x}}$.

Assumption 2 *The Jacobian matrix $F(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)$ of the motion model $\mathbf{f}(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)$ and the Jacobian matrix $H(\hat{\mathbf{x}}, t)$ of the measurement model $\mathbf{h}(\hat{\mathbf{x}}, t)$ are uniformly observable.*

Assumption 3 *There exist positive scalars $\underline{\delta}_1$ and $\bar{\delta}_1$ such that $\underline{\delta}_1 \leq \|H(\hat{\mathbf{x}}, t)\| \leq \bar{\delta}_1$.*

Assumption 4 *There exist strictly positive constants $\underline{p}$ and $\bar{p}$ such that*

$$\underline{p}I \leq P^{-1}(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) \leq \bar{p}I, \qquad \forall t \geq 0 \tag{6.15}$$

Theorem 2 *(Deterministic Stability) The estimate $\hat{\mathbf{x}}(t)$ of the estimator in Eqs. (6.6)-(6.8) converges to the state $\mathbf{x}(t)$ globally and exponentially fast under Assumptions 2-4 if there exists a uniformly positive definite metric $P(\mathbf{s}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)$ that satisfies*

$$\begin{aligned}\dot{P}(\mathbf{s}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) = & F(\mathbf{s}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)P(\mathbf{s}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) + P(\mathbf{s}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)F^T(\mathbf{s}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) + 2\alpha P(\mathbf{s}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) \\ & - P(\mathbf{s}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)\left(-2\kappa I + H^T(\mathbf{s}, t)R^{-1}(\mathbf{s}, t)H(\mathbf{s}, t)\right)P(\mathbf{s}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) \\ & + \big(K(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) - K(\mathbf{s}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)\big)R(\mathbf{s}, t)\big(K(\hat{\mathbf{x}}, \mathbf{v}^b, \boldsymbol{\omega}^b, t) - K(\mathbf{s}, \mathbf{v}^b, \boldsymbol{\omega}^b, t)\big)^T\end{aligned} \tag{6.16}$$

Proof 2 *The virtual dynamics of Eq. (6.14) is given by*

$$\delta\dot{\mathbf{s}} = \big(F(\mathbf{s}, t) - K(\hat{\mathbf{x}}, t)H(\mathbf{s}, t)\big)\delta\mathbf{s} \tag{6.17}$$

where $\delta\mathbf{s}$ is the virtual displacement, which is an infinitesimal displacement of $\mathbf{s}$ at a fixed time t. We will not present $(\mathbf{v}^b, \boldsymbol{\omega}^b)$ from here for the sake of simplicity.

Consider the time derivative of the squared length of the virtual displacement $\delta\mathbf{s}$ with respect to the metric $P^{-1}(\mathbf{s}, t)$

$$\begin{aligned}\frac{d}{dt}\left(\delta\mathbf{s}^T P^{-1}(\mathbf{s}, t)\delta\mathbf{s}\right) = & \delta\mathbf{s}^T\big(F(\mathbf{s}, t) - K(\hat{\mathbf{x}}, t)H(\mathbf{s}, t)\big)^T P^{-1}(\mathbf{s}, t)\delta\mathbf{s} - \delta\mathbf{s}^T P^{-1}(\mathbf{s}, t)\dot{P}(\mathbf{s}, t)P^{-1}(\mathbf{s}, t)\delta\mathbf{s} \\ & + \delta\mathbf{s}^T P^{-1}(\mathbf{s}, t)\big(F(\mathbf{s}, t) - K(\hat{\mathbf{x}}, t)H(\mathbf{s}, t)\big)\delta\mathbf{s} \\ = & \delta\mathbf{s}^T P^{-1}(\mathbf{s}, t)\big(P(\mathbf{s}, t)F^T(\mathbf{s}, t) + F(\mathbf{s}, t)P(\mathbf{s}, t) - \dot{P}(\mathbf{s}, t) + (K(\hat{\mathbf{x}}, t) - K(\mathbf{s}, t))\,R(\mathbf{s}, t) \\ & \times (K(\hat{\mathbf{x}}, t) - K(\mathbf{s}, t))^T - K(\hat{\mathbf{x}}, t)R(\mathbf{s}, t)K^T(\hat{\mathbf{x}}, t) - K(\mathbf{s}, t)R(\mathbf{s}, t)K^T(\mathbf{s}, t)\big)P^{-1}(\mathbf{s}, t)\delta\mathbf{s}\end{aligned} \tag{6.18}$$

where $\dot{P}^{-1}(\mathbf{s}, t) = -P^{-1}(\mathbf{s}, t)\dot{P}(\mathbf{s}, t)P^{-1}(\mathbf{s}, t)$ [93]. By substituting Eq. (6.16), we can write Eq. (6.18) as

$$\begin{aligned}\frac{d}{dt}\left(\delta\mathbf{s}^T P^{-1}(\mathbf{s}, t)\delta\mathbf{s}\right) = & \delta\mathbf{s}^T P^{-1}(\mathbf{s}, t)\Big(P(\mathbf{s}, t)F^T(\mathbf{s}, t) + F(\mathbf{s}, t)P(\mathbf{s}, t) - \Big(F(\mathbf{s}, t)P(\mathbf{s}, t) + P(\mathbf{s}, t)F^T(\mathbf{s}, t) \\ & + 2\alpha P(\mathbf{s}, t) - P(\mathbf{s}, t)\left(-2\kappa I + H^T(\mathbf{s}, t)R^{-1}(\mathbf{s}, t)H(\mathbf{s}, t)\right)P(\mathbf{s}, t) \\ & + (K(\hat{\mathbf{x}}, t) - K(\mathbf{s}, t))\,R(\mathbf{s}, t)\,(K(\hat{\mathbf{x}}, t) - K(\mathbf{s}, t))^T\Big) \\ & + (K(\hat{\mathbf{x}}, t) - K(\mathbf{s}, t))\,R(\mathbf{s}, t)\big(K(\hat{\mathbf{x}}, t) - K(\mathbf{s}, t)\big)^T \\ & - K(\hat{\mathbf{x}}, t)R(\mathbf{s}, t)K^T(\hat{\mathbf{x}}, t) - K(\mathbf{s}, t)R(\mathbf{s}, t)K^T(\mathbf{s}, t)\Big)P^{-1}(\mathbf{s}, t)\delta\mathbf{s} \\ = & \delta\mathbf{s}^T P^{-1}(\mathbf{s}, t)\Big(-2\alpha P(\mathbf{s}, t) - 2\kappa P(\mathbf{s}, t)P(\mathbf{s}, t) - K(\hat{\mathbf{x}}, t)R(\mathbf{s}, t)K^T(\hat{\mathbf{x}}, t)\Big)P^{-1}(\mathbf{s}, t)\delta\mathbf{s}\end{aligned} \tag{6.19}$$

By considering the estimator gain given in Eq. (6.7), we can express the last term in Eq. (6.19) as

$$\begin{aligned} & P^{-1}(\mathbf{s},t)K(\hat{\mathbf{x}},t)R(\mathbf{s},t)K^T(\hat{\mathbf{x}},t)P^{-1}(\mathbf{s},t) \\ & = P^{-1}(\mathbf{s},t)P(\hat{\mathbf{x}},t)H^T(\hat{\mathbf{x}},t)R(\hat{\mathbf{x}},t)^{-1}R(\mathbf{s},t)R(\hat{\mathbf{x}},t)^{-1}H(\hat{\mathbf{x}},t)P(\hat{\mathbf{x}},t)P^{-1}(\mathbf{s},t) \\ & \geq \left(\frac{p\underline{\delta}_1}{\bar{p}\underline{r}}\right)^2 \bar{r}I \\ & = 2\kappa_1 I \end{aligned} \tag{6.20}$$

where $\bar{r}I \leq R(\mathbf{s},t) \leq \underline{r}I$*, and* $\underline{\delta}_1$ *is the lower bound of* $\|H(\hat{\mathbf{x}},t)\|$ *from Assumption 3. Then,*

$$\begin{aligned} \frac{d}{dt}\left(\delta\mathbf{s}^T P^{-1}(\mathbf{s},t)\delta\mathbf{s}\right) &\leq \delta\mathbf{s}^T\Big(-2\alpha P^{-1}(\mathbf{s},t) - 2\kappa I - 2\kappa_1 I\Big)\delta\mathbf{s} \\ &\leq -2\alpha_1 \delta\mathbf{s}^T P^{-1}(\mathbf{s},t)\delta\mathbf{s} \end{aligned} \tag{6.21}$$

where $\alpha_1 = \alpha + \frac{\kappa+\kappa_1}{\underline{p}}$*. Our proof follows the steps in [93], but we consider Jacobian matrices* $F(\hat{\mathbf{x}},\mathbf{v}^b,\boldsymbol{\omega}^b,t) = \frac{\partial \mathbf{f}(\hat{\mathbf{x}},\mathbf{v}^b,\boldsymbol{\omega}^b,t)}{\partial \hat{\mathbf{x}}}$ *and* $H(\hat{\mathbf{x}},t) = \frac{\partial \mathbf{h}(\hat{\mathbf{x}},t)}{\partial \hat{\mathbf{x}}}$ *to represent the motion model and the measurement model and relax the assumption that is required for SDC matrices.*

The bound on the squared length of the virtual displacement $\delta\mathbf{s}$ *with respect to the metric* $P^{-1}(\mathbf{s},t)$ *is given by*

$$\delta\mathbf{s}^T P^{-1}(\mathbf{s},t)\delta\mathbf{s} \leq \delta\mathbf{s}^T(0)P^{-1}(\mathbf{s}(0),0)\delta\mathbf{s}(0)e^{-2\alpha_1 t} \tag{6.22}$$

and

$$\|\delta\mathbf{s}(t)\| \leq \sqrt{\frac{\bar{p}}{\underline{p}}}\,\|\delta\mathbf{s}(0)\|\,e^{-\alpha_1 t} \tag{6.23}$$

which shows that the virtual system in Eq. (6.14) is contracting. Therefore, the state estimate $\hat{\mathbf{x}}(t)$ *of the estimator in Eqs. (6.6)-(6.8) converges to the true state* $\mathbf{x}(t)$ *globally and exponentially fast.* ■

Assumption 5 *Let* $\bar{p}_x = \sup_{t\geq 0,i,j} \left\|(P_{ij}^{-1})_{\mathbf{s}}\right\|$*,* $\bar{p}_{x^2} = \sup_{t\geq 0,i,j} \left\|\partial^2(P_{ij}^{-1})/\partial\mathbf{s}_i\partial\mathbf{s}_j\right\|$*, and* $\|B(\mathbf{x},t)\|_F \leq \bar{b}$*.*

Theorem 3 *(Stochastic Stability) The mean-squared estimation error of the estimator in Eqs. (6.6)-(6.8) is exponentially bounded with*

$$E\left[\|\mathbf{x}-\hat{\mathbf{x}}\|^2\right] \leq \left(E\left[V(\mathbf{s}(0),\delta\mathbf{s}(0),0)\right]\frac{1}{\underline{p}}e^{-2\alpha_2 t} + \frac{\delta_2}{2\underline{p}\alpha_2}\right) \tag{6.24}$$

under assumptions 2-5, where

$$\frac{\delta_2}{2\underline{p}\alpha_2} \geq \frac{\frac{\bar{p}_x}{\epsilon_1}\kappa_p + \bar{p}\bar{b}^2 + \left(\frac{\bar{\delta}_1}{\bar{r}}\right)^2 \underline{r}\,\mathrm{tr}\left(P(\hat{\mathbf{x}},t)\right)}{2\underline{p}\alpha_2} \tag{6.25}$$

The variable $\alpha_2 \equiv \alpha_1 - \frac{\kappa_p}{2\underline{p}}\left(\epsilon_1 \bar{p}_x + \frac{\bar{p}_{x^2}}{2}\right) > 0$, $\epsilon_1 > 0$, *and* $\kappa_p = \bar{b}^2 + \left(\frac{\bar{\delta}_1}{\bar{r}}\right)^2 \underline{r}\mathrm{tr}\left(P^2(\hat{\mathbf{x}}, t)\right)$ *as shown in [93], and* $\alpha_1 = \alpha + \frac{\kappa + \kappa_1}{\underline{p}}$ *is the convergence rate in Eq. (6.23).*

Proof 3 *Consider a Lyapunov-like function that is given by*

$$V(\mathbf{s}, \delta\mathbf{s}, t) = \int_0^1 \left(\frac{\partial \mathbf{s}}{\partial \mu}\right)^T P^{-1}(\mathbf{s}(\mu, t), t)\left(\frac{\partial \mathbf{s}}{\partial \mu}\right) d\mu \tag{6.26}$$

where $\mathbf{s}(\mu = 0) = \mathbf{x}$ *and* $\mathbf{s}(\mu = 1) = \hat{\mathbf{x}}$ *are the particular solutions of the virtual system in Eq. (6.14).*

The differential generator of $V(\mathbf{s}, \delta\mathbf{s}, t)$ *is given by*

$$\begin{aligned}
&\mathcal{L}V(\mathbf{s}, \delta\mathbf{s}, t) \\
&= \frac{d}{dt}\int_0^1 \left(\frac{\partial \mathbf{s}}{\partial \mu}\right)^T P^{-1}(\mathbf{s}, t)\frac{\partial \mathbf{s}}{\partial \mu} d\mu \\
&= \int_0^1 \left(\frac{\partial \mathbf{s}}{\partial \mu}\right)^T \frac{d}{dt} P^{-1}(\mathbf{s}, t)\frac{\partial \mathbf{s}}{\partial \mu} d\mu + \int_0^1 \frac{d}{dt}\left(\frac{\partial \mathbf{s}}{\partial \mu}\right)^T P^{-1}(\mathbf{s}, t)\frac{\partial \mathbf{s}}{\partial \mu} d\mu + \int_0^1 \left(\frac{\partial \mathbf{s}}{\partial \mu}\right)^T P^{-1}(\mathbf{s}, t)\frac{d}{dt}\frac{\partial \mathbf{s}}{\partial \mu} d\mu \\
&= \int_0^1 \left(\frac{\partial \mathbf{s}}{\partial \mu}\right)^T P^{-1}(\mathbf{s}, t)\dot{P}(\mathbf{s}, t)P^{-1}(\mathbf{s}, t)\frac{\partial \mathbf{s}}{\partial \mu} d\mu + V_1 + V_2 \\
&\leq \int_0^1 \left(\frac{\partial \mathbf{s}}{\partial \mu}\right)^T P^{-1}(\mathbf{s}, t)\dot{P}(\mathbf{s}, t)P^{-1}(\mathbf{s}, t)\frac{\partial \mathbf{s}}{\partial \mu} d\mu + \bar{V}_1 + \bar{V}_2
\end{aligned} \tag{6.27}$$

where

$$\begin{aligned}
V_1 &= \int_0^1 \left(\frac{\partial \mathbf{s}}{\partial \mu}\right)^T \left(\left(\frac{\partial \mathbf{f}_{CL}}{\partial \mu}\right) P^{-1}(\mathbf{s}, t) + P^{-1}(\mathbf{s}, t)\frac{\partial \mathbf{f}_{CL}}{\partial \mu}\right)\frac{\partial \mathbf{s}}{\partial \mu} d\mu \\
&= \int_0^1 \left(\frac{\partial \mathbf{s}}{\partial \mu}\right)^T U_1 \frac{\partial \mathbf{s}}{\partial \mu} d\mu \\
&\leq \int_0^1 \left(\frac{\partial \mathbf{s}}{\partial \mu}\right)^T \bar{U}_1 \frac{\partial \mathbf{s}}{\partial \mu} d\mu \\
&= \bar{V}_1
\end{aligned} \tag{6.28}$$

and

$$\begin{aligned}
U_1 &= \left(\frac{\partial \mathbf{f}_{CL}}{\partial \mu}\right)^T P^{-1}(\mathbf{s}, t) + P^{-1}(\mathbf{s}, t)\frac{\partial \mathbf{f}_{CL}}{\partial \mu} \\
&= \big(F(\mathbf{s}, t) - K(\hat{\mathbf{x}}, t)H(\mathbf{s}, t)\big)^T P^{-1}(\mathbf{s}, t) + P^{-1}(\mathbf{s}, t)\big(F(\mathbf{s}, t) - K(\hat{\mathbf{x}}, t)H(\mathbf{s}, t)\big) \\
&= P^{-1}(\mathbf{s}, t)\Big(P(\mathbf{s}, t)F^T(\mathbf{s}, t) + F(\mathbf{s}, t)P(\mathbf{s}, t) + \big(K(\mathbf{s}, t) - K(\hat{\mathbf{x}}, t)\big)R(\mathbf{s}, t)\big(K(\mathbf{s}, t) - K(\hat{\mathbf{x}}, t)\big)^T \\
&\quad - K(\mathbf{s}, t)R(\mathbf{s}, t)K^T(\mathbf{s}, t) - K(\hat{\mathbf{x}}, t)R(\mathbf{s}, t)K^T(\hat{\mathbf{x}}, t)\Big)P^{-1}(\mathbf{s}, t)
\end{aligned} \tag{6.29}$$

Here, $P^{-1}(\mathbf{s},t)K(\hat{\mathbf{x}},t)R(\mathbf{s},t)K^T(\hat{\mathbf{x}},t)P^{-1}(\mathbf{s},t) \geq 2\kappa_1 I$ as we consider in Eq. (6.20). Therefore,

$$\begin{aligned} U_1 &\leq P^{-1}(\mathbf{s},t)\Big(P(\mathbf{s},t)F^T(\mathbf{s},t) + F(\mathbf{s},t)P(\mathbf{s},t) + \big(K(\mathbf{s},t) - K(\hat{\mathbf{x}},t)\big)R(\mathbf{s},t)\big(K(\mathbf{s},t) - K(\hat{\mathbf{x}},t)\big)^T \\ &\quad - P(\mathbf{s},t)H^T(\mathbf{s},t)R^{-1}(\mathbf{s},t)H(\mathbf{s},t)P^T(\mathbf{s},t)\Big)P^{-1}(\mathbf{s},t) - 2\kappa_1 I \\ &= \bar{U}_1 \end{aligned} \tag{6.30}$$

The V_2 and $\bar{V}_2$ are given in [93]. Particularly, the upper bound of V_2 is given by

$$\begin{aligned} \bar{V}_2 &= \mathrm{tr}\Big(B^T(\mathbf{x},t)P^{-1}(\mathbf{x},t)B(\mathbf{x},t) + \big(K(\hat{\mathbf{x}},t)D(\mathbf{x},t)\big)^T P^{-1}(\hat{\mathbf{x}},t)K(\hat{\mathbf{x}},t)D(\mathbf{x},t)\Big) \\ &\quad + \bar{p}_x\left(\bar{b}^2 + \frac{\bar{\delta}_1^2 r}{\bar{r}^2}\mathrm{tr}\left(P^2(\hat{\mathbf{x}},t)\right)\right)\left(\int_0^1 \epsilon_1 \left\|\frac{\partial \mathbf{s}}{\partial \mu}\right\|^2 d\mu + \frac{1}{\epsilon_1}\right) + \frac{1}{2}\bar{p}_{x^2}\left(\bar{b}^2 + \frac{\bar{\delta}_1^2 r}{\bar{r}^2}\mathrm{tr}\left(P^2(\hat{\mathbf{x}},t)\right)\right)\int_0^1 \left\|\frac{\partial \mathbf{s}}{\partial \mu}\right\|^2 d\mu \end{aligned} \tag{6.31}$$

Therefore,

$$\begin{aligned} \mathcal{L}V(\mathbf{s},\delta\mathbf{s},t) &\leq \int_0^1 \left(\frac{\partial \mathbf{s}}{\partial \mu}\right)^T \Big(- P^{-1}(\mathbf{s},t)\Big(F(\mathbf{s},t)P(\mathbf{s},t) + P(\mathbf{s},t)F^T(\mathbf{s},t) + 2\alpha P(\mathbf{s},t) - P(\mathbf{s},t)\Big(- 2\kappa I \\ &\quad + H^T(\mathbf{s},t)R^{-1}(\mathbf{s},t)H(\mathbf{s},t)\Big)P(\mathbf{s},t) + \big(K(\hat{\mathbf{x}},t) - K(\mathbf{s},t)\big)R(\hat{\mathbf{x}},t)\big(K(\hat{\mathbf{x}},t) - K(\mathbf{s},t)\big)^T\Big)P^{-1}(\mathbf{s},t) \\ &\quad + P^{-1}(\mathbf{s},t)\Big(P(\mathbf{s},t)F^T(\mathbf{s},t) + F(\mathbf{s},t)P(\mathbf{s},t) + \big(K(\mathbf{s},t) - K(\hat{\mathbf{x}},t)\big)R(\mathbf{s},t)\big(K(\mathbf{s},t) - K(\hat{\mathbf{x}},t)\big)^T \\ &\quad - K(\mathbf{s},t)R(\mathbf{s},t)K^T(\mathbf{s},t)\Big)P^{-1}(\mathbf{s},t) - 2\kappa_1 I\Big)\frac{\partial \mathbf{s}}{\partial \mu}d\mu + \bar{V}_2 \\ &= \int_0^1 \left(\frac{\partial \mathbf{s}}{\partial \mu}\right)^T \left(-2\alpha P^{-1}(\mathbf{s},t) - 2\kappa I - 2\kappa_1 I\right)\frac{\partial \mathbf{s}}{\partial \mu}d\mu + \bar{V}_2 \\ &\leq -2\alpha_1 V(\mathbf{s},t) + \bar{V}_2 \\ &\leq -2\alpha_2 V(\mathbf{s},t) + \delta_2 \end{aligned} \tag{6.32}$$

where α_1 and α_2 are defined after Eq. (6.25), and a lower bound of δ_2 is given by Eq. (6.25). Our proof and assumptions follow the steps in [93], but we do not require the parameters that bounds SDCs. ■

6.3.3 LMI Formulation

We present an algorithm that computes the gain for the estimator given in Eqs. (6.6)-(6.8). We approximate $\dot{P}(\hat{\mathbf{x}},t)$ to its steady state value [93] and convert the differential Riccati equation in Eq. (6.8) to an algebraic

Riccati inequality given by

$$F(\hat{\mathbf{x}},t)P(\hat{\mathbf{x}},t)+P(\hat{\mathbf{x}},t)F^T(\hat{\mathbf{x}},t)+2\alpha P(\hat{\mathbf{x}},t)-P(\hat{\mathbf{x}},t)\left(-2\kappa I+H^T(\hat{\mathbf{x}},t)R^{-1}(\hat{\mathbf{x}},t)H(\hat{\mathbf{x}},t)\right)P(\hat{\mathbf{x}},t)\leq 0 \quad (6.33)$$

We multiply both side of Eq. (6.33) with $Q(\hat{\mathbf{x}},t)=P^{-1}(\hat{\mathbf{x}},t)$ to derive the algebraic Riccati inequality given by

$$Q(\hat{\mathbf{x}},t)F(\hat{\mathbf{x}},t)+F^T(\hat{\mathbf{x}},t)Q(\hat{\mathbf{x}},t)+2\alpha Q(\hat{\mathbf{x}},t)+2\kappa I-H^T(\hat{\mathbf{x}},t)R^{-1}(\hat{\mathbf{x}},t)H(\hat{\mathbf{x}},t)\leq 0 \quad (6.34)$$

We solve a convex optimization problem with the LMI constraint given by Eq. (6.34). We minimize the mean-squared estimation error $E\left[\|\mathbf{x}-\hat{\mathbf{x}}\|^2\right]$ which converges to $\frac{\delta_2}{2\underline{p}\alpha_2}$ as shown in Eq. (6.24). By approximating the covariance of the measurement noise with a constant matrix $R(\mathbf{s},t)$, Eq. (6.25) can be extended as

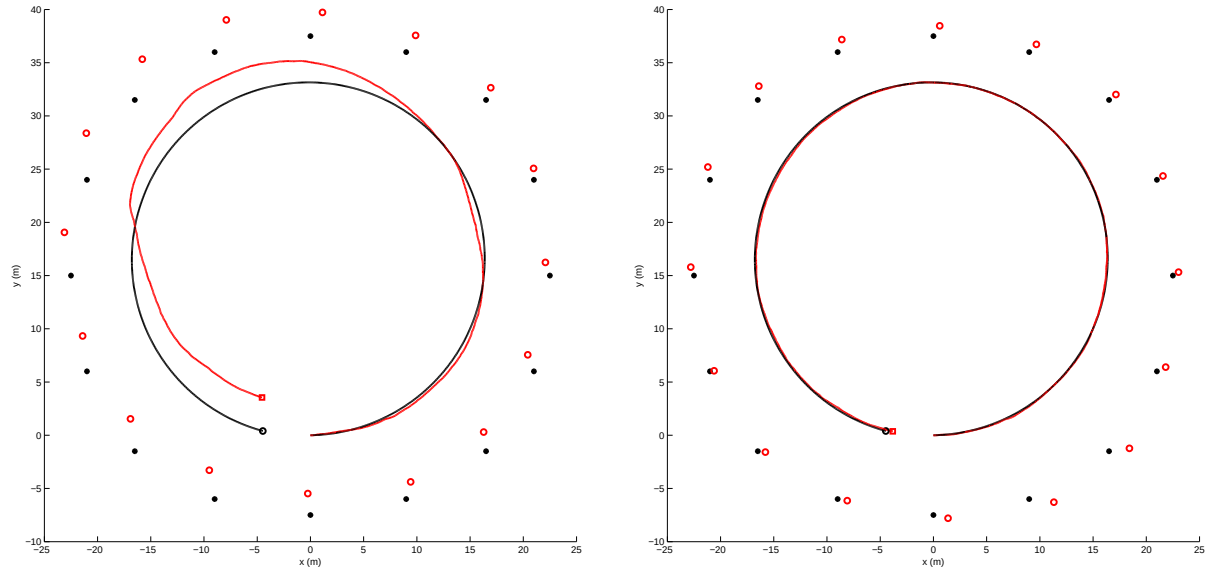
$$\begin{aligned}\frac{\delta_2}{2\underline{p}\alpha_2}&\geq\frac{\frac{\bar{p}_x}{\epsilon_1}\kappa_p+\bar{p}\bar{b}^2+\frac{\bar{\delta}_1^2}{\bar{r}}\mathrm{tr}\left(P(\hat{\mathbf{x}},t)\right)}{2\underline{p}\alpha_2}\\&\geq\frac{\left(\frac{\bar{p}_x}{\epsilon_1}+\bar{p}\right)\bar{b}^2+\left(\frac{\bar{p}_x}{\epsilon_1}\mathrm{tr}\left(P^2(\hat{\mathbf{x}},t)\right)+\mathrm{tr}\left(P(\hat{\mathbf{x}},t)\right)\right)\frac{\bar{\delta}_1^2}{\bar{r}}}{2\left(\underline{p}\alpha+\kappa+\frac{\left(\underline{p}\delta_1\right)^2}{\bar{p}^2\underline{r}}\right)-\left(\bar{b}^2+\frac{\bar{\delta}_1^2}{\bar{r}}\mathrm{tr}\left(P^2(\hat{\mathbf{x}},t)\right)\right)\left(\epsilon_1\bar{p}_x+\frac{\bar{p}_{x^2}}{2}\right)}\end{aligned} \quad (6.35)$$

To reduce the lower bound for $\frac{\delta_2}{2\underline{p}\alpha_2}$, we want to minimize $\frac{\bar{p}_x}{\epsilon_1}\mathrm{tr}\left(P^2(\hat{\mathbf{x}},t)\right)+\mathrm{tr}\left(P(\hat{\mathbf{x}},t)\right)$ and $\bar{b}^2+\frac{\bar{\delta}_1^2}{\bar{r}}\mathrm{tr}\left(P^2(\hat{\mathbf{x}},t)\right)$ by minimizing $\mathrm{tr}\left(P(\hat{\mathbf{x}},t)\right)^2$ and $\mathrm{tr}\left(P(\hat{\mathbf{x}},t)\right)$. We want to minimize $\left(\frac{\bar{p}_x}{\epsilon_1}+\bar{p}\right)\bar{b}^2$ and maximize $\underline{p}\alpha+\kappa+\frac{\left(\underline{p}\delta_1\right)^2}{\bar{p}^2\underline{r}}$ by minimizing $\lambda_{\max}\left(P^{-1}(\hat{\mathbf{x}},t)\right)$ and maximizing $\lambda_{\min}\left(P^{-1}(\hat{\mathbf{x}},t)\right)$ and κ. Since $\mathrm{tr}\left(P^2(\hat{\mathbf{x}},t)\right)$ is not a convex function of $P(\hat{\mathbf{x}},t)$, we minimize its convex upper bound $\mathrm{tr}\left(P(\hat{\mathbf{x}},t)\right)^2$.

Considering these conditions, we apply a convex objective function which as follows:

$$\begin{aligned}&\min\left(\Lambda_1\mathrm{tr}\left(Q(\hat{\mathbf{x}},t)^{-1}\right)^2+\Lambda_2\lambda_{\max}\left(Q(\hat{\mathbf{x}},t)\right)-\Lambda_3\lambda_{\min}\left(Q(\hat{\mathbf{x}},t)\right)-\Lambda_4\kappa\right)\\&\text{subject to Eq. (6.34)}\end{aligned} \quad (6.36)$$

Eq. (6.36) follows the structure of the convex optimization problem presented in [93] except for the SDC parametrization. The estimator in Eqs. (6.6)-(6.8) reduces the mean-squared estimation error using the solution of the convex optimization problem described in Eq. (6.36).



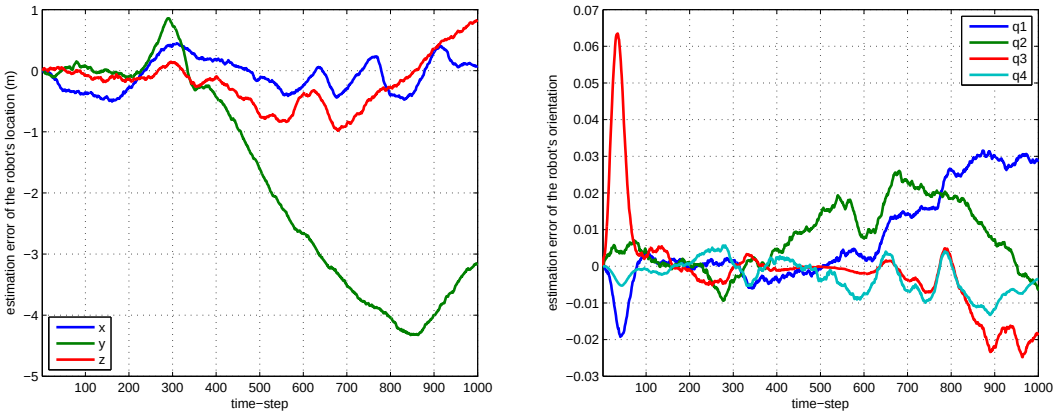
(a) Localization and mapping with EKF

(b) Localization and mapping with convex optimization

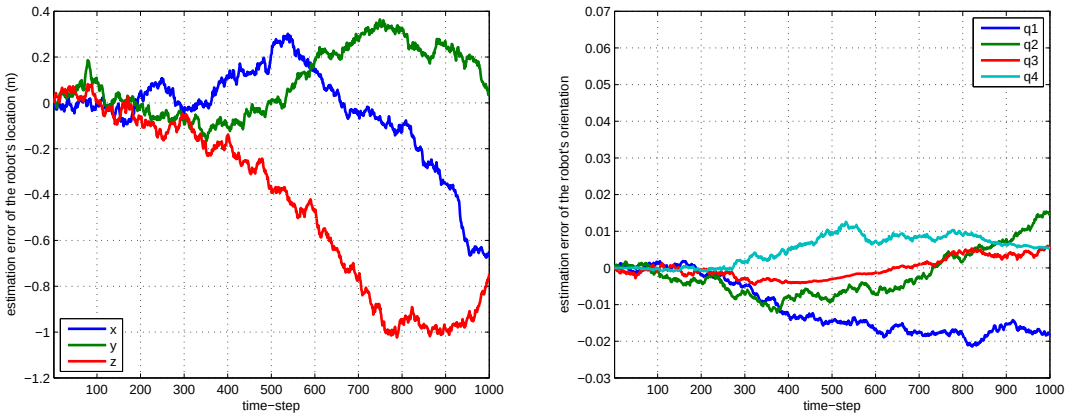
Figure 6.2: Numerical simulation results

6.4 Numerical Simulations

In this section, we present simulation results of localization and robot-centric mapping and analyze the performance of the convex optimization based SLAM estimator in comparison with an EKF based SLAM estimator. During the simulation, the robot travels a circular 3D trajectory for approximately 100 m (see Figure 6.2) at a velocity of approximately 0.5 m/s and angular velocity of approximately 0.03 rad/s and observes 16 features (marked with black dots) with 1σ Gaussian white noise in the pixel measurements. The convex optimization problem is solved using the CVX toolbox [95, 96] with the MOSEK solver [97] in MATLAB to obtain the positive definite metric P. To expedite the overall computation, the metric P is computed every 10 time-steps. The estimator gain K is computed every time-step using the piecewise constant metric P. Figures 6.3 and 6.4 compares the pose error of the robot and the RMSE using the EKF estimator and our estimator. The error norm of the EKF based SLAM is 3.2591 m, 0.0356, and 10.5795 m for the location and the quaternion orientation of the robot and the location of the features, respectively. The error norm of the convex optimization based SLAM is 1.0037 m, 0.023676, and 4.555 m for the location and the quaternion orientation of the robot and the location of the features, respectively. The accuracy of the EKF based SLAM degrades with an incorrect initial estimate covariance and the results can quickly diverge in the worse case. The results show that the trajectory of the robot is estimated more reliably using our convex optimization based estimator.

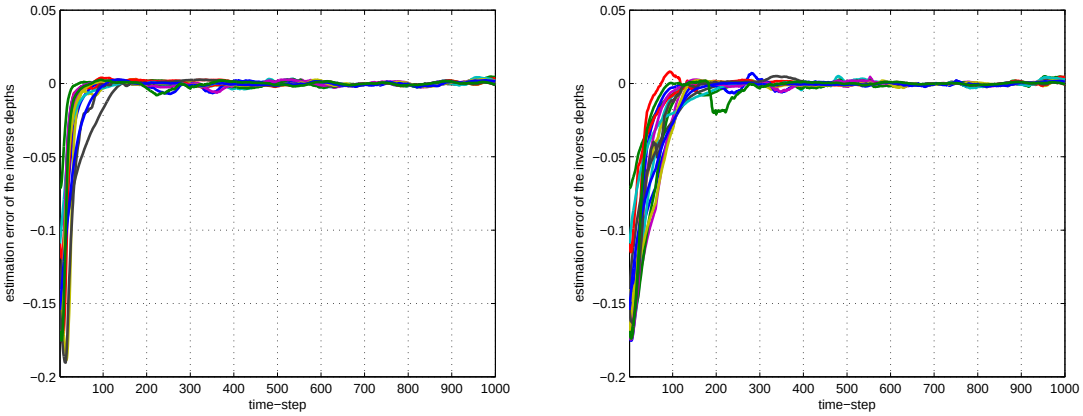


(a) Estimation error of the robot with EKF



(b) Estimation error of the robot with convex optimization

Figure 6.3: Error in the robot's pose estimate



(a) Estimation error of the inverse-depth with EKF

(b) Estimation error of the inverse-depth with convex optimization

Figure 6.4: Error in the inverse-depth estimates of the features

6.5 Conclusions

In this chapter, we presented a convex optimization based SLAM algorithm using our localization and robot-centric mapping system model and a nonlinear estimator implemented with convex optimization. Contraction analysis of the nonlinear estimator was used to derive an LMI constraint that guarantees exponential stability under the assumptions that we described. The stochastic incremental stability of the estimator was used to derive the objective function of the optimization problem that could reduce mean-squared estimation error.

We showed a comparison of the performance of our convex optimization based estimator and an EKF estimator via numerical simulations of the localization and mapping. From the simulation results, we concluded that the convex optimization based estimator could produce SLAM results more accurately than the EKF estimator without tuning the initial estimate covariance

Chapter 7

CONCLUSIONS AND FUTURE WORK

In this chapter, we provide a brief summary and a list of major contributions of this dissertation. We also recommend future work in the areas of estimator design and on-vehicle experiments.

7.1 Thesis Summary and Contributions

In this dissertation, we developed vision-based localization and mapping methods that can enhance the accuracy of the localization and mapping results for two different applications: the riverine mapping with a UAV and the autonomous mowing with a robotic mower. We formulated a system model that exploited the robot-centric mapping framework and localization of the robot with respect to its initial pose. We exploited a multiple-view geometry formulation with initial and current view projection of point features and the reflection of features for the riverine SLAM problem. We demonstrated that the observability of the estimation system is improved by applying our proposed methods and showed enhanced localization and mapping results in both numerical simulations and real-world experiments. We also presented a convex optimization based localization and mapping algorithm using incremental stability analysis. We assured the performance of the localization and mapping using the proposed convex optimization based estimator.

Monocular-Vision-Based Localization and Mapping for a UAV with a Planar Ground Assumption

We presented a monocular-vision-based algorithm with a particular focus on navigation of a UAV in multiple environments. Our method exploited the planar ground assumption in multiple environments. In the presence of coplanar features and the knowledge of the camera height, we have shown that the range and bearing to the landmarks on the ground plane can be measured instantaneously. We estimated the attitude of the UAV separately by exploiting the epipolar geometry with multiple features that are not required to be included in the FastSLAM estimation state vector. Localization and mapping was performed by applying the CMKF based FastSLAM algorithm to the attitude and range estimation. The results were obtained in an indoor environment from the Beckman Institute at the UIUC, and in outdoor environments at the

Engineering Quad and Boneyard Creek at the UIUC. It was demonstrated that with our algorithms, a monocular based system is able to perform visual SLAM that generates a structured map of the ground plane and obtains a pose estimate of the vehicle.

Vision-Based Localization and Robot-Centric Mapping in Riverine Environments

We presented a vision-based SLAM algorithm developed for riverine environments. The water reflections of the surrounding features were used for SLAM for the first time. The performance of our visual SLAM algorithm was 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 showed 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. 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.

Observer Design for Localization and Mapping with an Omnidirectional Camera for Autonomous Mowing

An omnidirectional vision based localization and mapping algorithm for an autonomous mower was presented. A nonlinear observer was designed using pseudo-measurements of landmarks' depth for robot-centric landmark mapping. A boundary estimation strategy using localization results was described. We proposed to use the estimated boundary and landmark map to estimate the location of the mower for autonomous mowing. Numerical simulations illustrated the convergence of the estimates and the capability of using the estimates for containment of the mower. Preliminary experimental results showed boundary estimation and landmark mapping with a set of data collected with our autonomous mower.

Omnidirectional-Vision-Based Estimation for Containment Detection of a Robotic Mower

The boundary estimation and landmark mapping, mowing location estimation, and containment detection of a robotic mower was presented with an omnidirectional-vision-based system. We designed a model for

the localization and mapping with a robot-centric framework which uses inertial measurements and velocity measurements, along with unit vector vision measurements. As we showed with numerical simulation results, our method could outperform a well-known exiting method due to its relatively high degree of observability. The outdoor experimental results demonstrated the process of our entire algorithm.

Convex Optimization Based SLAM

We presented a convex optimization based SLAM algorithm using our localization and robot-centric mapping system model and a nonlinear estimator implemented with convex optimization. Contraction analysis of the nonlinear estimator was used to derive an LMI constraint that guarantees exponential stability under the assumptions that we described. The stochastic incremental stability of the estimator was used to derive the objective function of the optimization problem that could reduce mean-squared estimation error.

We showed a comparison of the performance of our convex optimization based estimator and an EKF estimator via numerical simulations of the localization and mapping. From the simulation results, we concluded that the convex optimization based estimator could produce SLAM results more accurately than the EKF estimator without tuning the initial estimate covariance.

Summary of Contributions

The contributions of this dissertation are summarized as follows:

- We formulated a system model that exploited a differential equation of motion of robot-centric mapping for the localization of the robot with respect to its initial pose.
- Using the robot-centric estimation framework, we reported experimental results of containment detection with an omnidirectional camera for robotic mowing applications for the first time.
- We used a multiple-view geometry formulation with initial and current view projection of point features and reflection of features in a riverine environment.
- We reported the first result that exploited the reflections of features in a riverine environment for localization and mapping.
- Using incremental stability analysis, we formulated a convex optimization based localization and mapping algorithm that could outperform EKF based methods.

7.2 Recommended Future Work

We have shown the enhancement in the performance of the localization and mapping using the robot-centric mapping framework and multiple view measurements with real-world experimental results, but the results were obtained offline. Recommendations of future work to achieve successful on-vehicle experimental results are as follows:

- To perform precise localization in real-time, an accurate dense map has to be provided to the robot. Therefore, a large set of features should be extracted from the vision data with tracking and matching results that are highly reliable.
- Additional constraints from the surrounding environment can improve the estimation results. For example, a planar constraint can be used to enhance the accuracy of the results if we can extract regions from building walls.
- The localization and mapping results should be accurate and robust to the tracking error and to the error in the calibration of the vision sensor and the IMU used in our algorithms. Since the estimate of a feature in our algorithms include the unit vector direction or the normalized pixel coordinates of the feature, we can use the estimation results to predict the location of the feature in the next image frame and achieve robustness to abrupt motion.

We have presented a convex optimization based SLAM algorithm. Stochastic incremental stability has been used to derive the lower bound of the mean-squared estimation error of our SLAM estimator. We can investigate an objective function of the convex optimization problem that can represent the mean-squared estimation error more precisely and reduce the error more effectively.

Bibliography

[1] S. Weiss, M. Achtelik, S. Lynen, M. Achtelik, L. Kneip, M. Chli, and R. Siegwart, "Monocular vision for long-term micro aerial vehicle state estimation: A compendium," *Journal of Field Robotics*, vol. 30, no. 5, pp. 803–831, 2013.

[2] A. Kim and R. M. Eustice, "Real-time visual SLAM for autonomous underwater hull inspection using visual saliency," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 719–733, 2013.

[3] K. Choi, J. Park, Y.-H. Kim, and H.-K. Lee, "Monocular SLAM with undelayed initialization for an indoor robot," *Robotics and Autonomous Systems*, vol. 60, no. 6, pp. 841–851, 2012.

[4] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA: MIT Press, 2005.

[5] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[6] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): part II," *IEEE Robotics Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[7] S. Haner and A. Heyden, "On-line structure and motion estimation based on a novel parameterized extended Kalman filter," in *International Conference on Pattern Recognition*, 2010, pp. 1836–1839.

[8] J. A. Castellanos, R. Martinez-cantin, J. D. Tardos, and J. Neira, "Robocentric map joining: Improving the consistency of EKF-SLAM," *Robotics and Autonomous Systems*, vol. 55, pp. 21–29, 2007.

[9] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry," *Journal of Field Robotics*, vol. 27, no. 5, pp. 609–631, 2010.

[10] B. Williams and I. Reid, "On combining visual SLAM and visual odometry," in *Proc. IEEE International Conference on Robotics and Automation*, Anchorage, AK, May 2010, pp. 3494–3500.

[11] A. Boberg, A. Bishop, and P. Jensfelt, "Robocentric mapping and localization in modified spherical coordinates with bearing measurements," in *Proc. International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, Melbourne, Australia, Dec. 2009, pp. 139–144.

[12] M. Jankovic and B. K. Ghosh, "Visually guided ranging from observations of points, lines and curves via an identifier based nonlinear observer," *Systems & Control Letters*, vol. 25, no. 1, pp. 63–73, 1995.

[13] W. Dixon, Y. Fang, D. Dawson, and T. Flynn, "Range identification for perspective vision systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 12, pp. 2232–2238, 2003.

[14] A. Dani, N. Fischer, and W. Dixon, "Single camera structure and motion," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 238–243, 2012.

[15] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[16] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, 2008.

[17] J. Sola, T. Vidal-Calleja, J. Civera, and J. M. M. Montiel, "Impact of landmark parametrization on monocular EKF-SLAM with points and lines," *The International Journal of Computer Vision*, vol. 97, no. 3, pp. 339–368, 2012.

[18] M. M. D. M. Simone Ceriani, Daniele Marzorati and D. G. Sorrenti, "On feature parameterization for EKF-based monocular SLAM," in *Proc. 18th World Congress of the International Federation of Automatic Control*, Milan, Italy, Aug 2011, pp. 6829–6834.

[19] H. Strasdat, J. Montiel, and A. Davison, "Real-time monocular SLAM : Why filter?" in *Proc. IEEE International Conference on Robotics and Automation*, Anchorage, AK, May 2010, pp. 2657–2664.

[20] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, "Keyframe-based visual-inertial slam using nonlinear optimization," in *Proc. Robotics: Science and Systems*, Berlin, Germany, June 2013.

[21] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE International Conference on Robotics and Automation*, Hong Kong, China, May 2014, pp. 15–22.

[22] J. Engel, T. Schps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Computer Vision ECCV 2014*, ser. Lecture Notes in Computer Science, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8690. Springer International Publishing, 2014, pp. 834–849.

[23] L. Clemente, A. Davison, I. Reid, J. Neira, and J. D. Tardós, "Mapping large loops with a single hand-held camera," in *Proc. Robotics: Science and Systems*, June 2007.

[24] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. IEEE and ACM International Symposium on Mixed and Augmented Reality*, Nara, Japan, Nov. 2007, pp. 225–234.

[25] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-SLAM based navigation for autonomous micro helicopters in GPS-denied environments," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, 2011.

[26] L. Doitsidis, S. Weiss, A. Renzaglia, M. Achtelik, E. Kosmatopoulos, R. Siegwart, and D. Scaramuzza, "Optimal surveillance coverage for teams of micro aerial vehicles in GPS-denied environments using onboard vision," *Springer Tracts in Advanced Robotics*, vol. 33, no. 1-2, pp. 173–188, 2012.

[27] S. Scherer, J. Rehder, S. Achar, H. Cover, A. Chambers, S. Nuske, and S. Singh, "River mapping from a flying robot: state estimation, river detection, and obstacle mapping," *Autonomous Robots*, vol. 33, no. 1-2, pp. 189–214, 2012.

[28] J. Rehder, K. Gupta, S. Nuske, and S. Singh, "Global pose estimation with limited GPS and long range visual odometry," in *Proc. IEEE International Conference on Robotics and Automation*, St. Paul, MN, May 2012, pp. 627–633.

[29] S. Jain, S. Nuske, A. Chambers, L. Yoder, H. Cover, L. Chamberlain, S. Scherer, and S. Singh, "Autonomous river exploration," in *Proc. International Conference on Field and Service Robotics*, Dec 2013.

[30] J. C. Leedekerken, M. F. Fallon, and J. J. Leonard, "Mapping complex marine environments with autonomous surface craft," in *Proc. International Symposium on Experimental Robotics*, Delhi, India, Dec. 2010.

[31] M. F. Fallon, G. Papadopoulos, J. J. Leonard, and N. M. Patrikalakis, "Cooperative AUV navigation using a single maneuvering surface craft," *The International Journal of Robotics Research*, vol. 29, no. 12, pp. 1461–1474, 2010.

[32] J. Hesch, A. Mourikis, and S. Roumeliotis, "Mirror-based extrinsic camera calibration," *Springer Tracts in Advanced Robotics*, vol. 57, pp. 285–299, 2009.

[33] G. Panahandeh and M. Jansson, "IMU-camera self-calibration using planar mirror reflection," in *Proc. International Conference on Indoor Positioning and Indoor Navigation*, Guimares, Portugal, Sept. 2011, pp. 1–7.

[34] G. L. Mariottini, S. Scheggi, F. Morbidi, and D. Prattichizzo, "Planar mirrors for image-based robot localization and 3-D reconstruction," *Mechatronics - Special Issue on Visual Servoing*, vol. 22, pp. 398–409, 2012.

[35] K. W. Lee, W. S. Wijesoma, and J. I. Guzman, "On the observability and observability analysis of SLAM," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, Oct. 2006, pp. 3569–3574.

[36] M. Bryson and S. Sukkarieh, "Observability analysis and active control for airborne SLAM," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 1, pp. 261–280, 2008.

[37] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Observability-based rules for designing consistent EKF SLAM estimators," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 502–528, 2010.

[38] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *The International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011.

[39] J. Hesch, D. Kottas, S. Bowman, and S. Roumeliotis, "Consistency analysis and improvement of vision-aided inertial navigation," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 158–176, 2014.

[40] A. Martinelli, "Closed-form solution of visual-inertial structure from motion," *International Journal of Computer Vision*, vol. 106, no. 2, pp. 138–152, 2014.

[41] J. Yang, D. Rao, S.-J. Chung, and S. Hutchinson, "Monocular vision based navigation in GPS denied riverine environments," in *Proc. AIAA Infotech at Aerospace Conference*, St. Louis, MO, March 2011, AIAA 2011-1403.

[42] J. Yang, A. Dani, S.-J. Chung, and S. Hutchinson, "Inertial-aided vision-based localization and mapping in a riverine environment with reflection measurements," in *Proc. AIAA Guidance, Navigation, and Control Conference*, Boston, MA, Aug. 2013, AIAA 2013-5246.

[43] ——, "Vision-based localization and robot-centric mapping in riverine environments," *Journal of Field Robotics*, DOI: 10.1002/rob.21606, 2015.

[44] J. Yang, S.-J. Chung, S. Hutchinson, D. Johnson, and M. Kise, "Vision-based localization and mapping for an autonomous mower," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013, pp. 3655–3662.

[45] ——, "Omnidirectional-vision-based estimation for containment detection of a robotic mower," in *Proc. IEEE International Conference on Robotics and Automation*, Seatle, WA, 2015, pp. 6344–6351.

[46] K. Celik, S.-J. Chung, and A. Somani, "Mono-vision corner SLAM for indoor navigation," in *Proc. IEEE International Conference on Electro/Information Technology*, May 2008, pp. 343–348.

[47] K. Celik, S.-J. Chung, M. Clausman, and A. Somani, "Monocular vision SLAM for indoor aerial vehicles," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2009, pp. 1566–1573.

[48] "FastSLAM 1.0," in *FastSLAM*, ser. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2007, vol. 27, pp. 27–62.

[49] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[50] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[51] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, 1997.

[52] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[53] A. T. Conlisk, "Modern helicopter aerodynamics," *Annual Review of Fluid Mechanics*, vol. 29, no. 2, pp. 515–567, 1997.

[54] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robotics Automation Magazine*, vol. 18, no. 4, pp. 80 –92, 2011.

[55] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media Inc, 2008.

[56] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, 1994, pp. 593–600.

[57] B. D. Lucas and T. Kanade, "Iterative image registration technique with an application to stereo vision," in *Proc. International Joint Conference on Artificial Intelligence*, 1981, pp. 674–679.

[58] D. Lerro and Y. Bar-Shalom, "Tracking with debiased consistent converted measurements versus EKF," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 3, pp. 1015–1022, 1993.

[59] B. Balaji and Z. Dinga, "A performance comparison of nonlinear filtering techniques based on recorded radar datasets," in *Proc. SPIE - The International Society for Optical Engineering*, vol. 7445(74450Q), San Diego, CA, 2009.

[60] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics Automation Magazine*, vol. 13, no. 4, pp. 82–90, Dec 2006.

[61] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1994, pp. 593–600.

[62] R. Haralick and L. Shapiro, *Computer and robot vision*. MA: Addison-Wesley, 1993.

[63] H. Zhang, X. Guo, and X. Cao, "Water reflection detection using a flip invariant shape detector," in *Proc. International Conference on Pattern Recognition*, Aug 2010, pp. 633–636.

[64] S.-H. Zhong, Y. Liu, Y. Liu, and C.-S. Li, "Water reflection recognition based on motion blur invariant moments in curvelet space," *IEEE Transactions on Image Processing*, vol. 22, no. 11, pp. 4301–4313, Nov 2013.

[65] K. Reif, S. Gunther, E. Yaz, and R. Unbehauen, "Stochastic stability of the discrete-time extended Kalman filter," *IEEE Transactions on Automatic Control*, vol. 44, no. 4, pp. 714–728, Apr 1999.

[66] J. Crassidis, F. Landis Markley, and Y. Cheng, "Survey of nonlinear attitude estimation methods," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 12–28, 2007.

[67] G. F. Franklin, D. J. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 4th ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[68] R. Hermann and A. Krener, "Nonlinear controllability and observability," *IEEE Transactions on Automatic Control*, vol. 22, no. 5, pp. 728–740, 1977.

[69] A. J. Krener and K. Ide, "Measures of unobservability," in *Proc. IEEE Conference on Decision and Control*, Dec 2009, pp. 6401–6406.

[70] T. Aroda, "The rain forests of the mighty Amazon river," [Online]. Available: http://tomaroda.hubpages.com/hub/The-rain-forests-of-the-mighty-Amazon-river.

[71] J. Y. Bouguet, "Camera calibration toolbox for Matlab," 2008, [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/.

[72] J. Lobo and J. Dias, "Relative pose calibration between visual and inertial sensors," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 561–575, 2007.

[73] S. Srkk and J. Sarmavuori, "Gaussian filtering and smoothing for continuous-discrete dynamic systems," *Signal Processing*, vol. 93, no. 2, pp. 500 – 510, 2013.

[74] R. W. Hicks II and E. L. Hall, "Survey of robot lawn mowers," in *Proc. SPIE Intelligent Robots and Computer Vision XIX: Algorithms, Techniques, and Active Vision*, Boston, MA, Nov. 2000, pp. 262–269.

[75] H. Sahin and L. Guvenc, "Household robotics: autonomous devices for vacuuming and lawn mowing [applications of control]," *IEEE Control Systems*, vol. 27, no. 2, pp. 20–96, 2007.

[76] A. Smith, H. Chang, and E. Blanchard, "An outdoor high-accuracy local positioning system for an autonomous robotic golf greens mower," in *Proc. IEEE International Conference on Robotics and Automation*, St. Paul, MN, May 2012, pp. 2633–2639.

[77] S. Sukkarieh, E. Nebot, and H. Durrant-Whyte, "A high integrity IMU/GPS navigation loop for autonomous land vehicle applications," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 572 –578, June 1999.

[78] B. Ferris, D. Haehnel, and D. Fox, "Gaussian processes for signal strength-based location estimation," in *Proc. Robotics: Science and Systems*, Philadelphia, PA, Aug. 2006.

[79] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, Oct. 1996.

[80] A. Heyden and O. Dahl, "Provably convergent structure and motion estimation for perspective systems," in *Proc. IEEE Conference on Decision and Control*, Shanghai, China, Dec. 2009, pp. 4216–4221.

[81] W. Lohmiller and J.-J. E. Slotine, "Nonlinear process control using contraction theory," *AIChE Journal*, vol. 46, pp. 588–596, 2000.

[82] K. Rifai and J.-J. Slotine, "Compositional contraction analysis of resetting hybrid systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 9, pp. 1536–1541, 2006.

[83] Y. Zhao and J.-J. E. Slotine, "Discrete nonlinear observers for inertial navigation," *Systems & Control Letters*, vol. 54, no. 9, pp. 887–898, 2005.

[84] K. Hormann and A. Agathos, "The point in polygon problem for arbitrary polygons," *Computational Geometry*, vol. 20, no. 3, pp. 131 – 144, 2001.

[85] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids," in *Proc. IEEE International Conference on Robotics and Automation*, Roma, Italy, April 2007, pp. 3945–3950.

[86] O. Dahl, F. Nyberg, and A. Heyden, "Nonlinear and adaptive observers for perspective dynamic systems," in *Proc. American Control Conference*, New York City, NY, July 2007, pp. 966–971.

[87] H. Rauch, F. Tung, and C. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, pp. 1445 – 1450, 1965.

[88] G. Dullerud and F. Paganini, *A Course in Robust Control Theory: A Convex Approach.* Springer, 2000.

[89] C. Mei, S. Benhimane, E. Malis, and P. Rives, "Homography-based tracking for central catadioptric cameras," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2006, pp. 669–674.

[90] A. Leick, *GPS satellite surveying.* New York: John Wiley & Sons, 2004.

[91] S. Huang, Y. Lai, U. Frese, and G. Dissanayake, "How far is SLAM from a linear least squares problem?" in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 3011–3016.

[92] M. Liu, S. Huang, G. Dissanayake, and H. Wang, "A convex optimization based approach for pose SLAM problems," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 1898–1903.

[93] A. P. Dani, S.-J. Chung, and S. Hutchinson, "Observer design for stochastic nonlinear systems via contraction-based incremental stability," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 700–714, 2015.

[94] Q. Pham, N. Tabareau, and J. E. Slotine, "A contraction theory approach to stochastic incremental stability," *IEEE Transactions on Automatic Control*, vol. 54, no. 4, pp. 816–820, 2009.

[95] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," Mar. 2014.

[96] ——, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110.

[97] M. ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 7.1*, 2015.