

A DAISY-CHAINING APPROACH FOR VISION-BASED CONTROL AND
ESTIMATION

By

SIDDHARTHA S. MEHTA

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2010

© 2010 Siddhartha S. Mehta

To my parents Satish and Sulabha, and my friends and family members who constantly filled me with motivation and joy.

ACKNOWLEDGMENTS

I express my most sincere appreciation to Dr. Warren E. Dixon and Dr. Thomas F. Burks. Their contribution to my current and ensuing career cannot be overemphasized. I thank them for the education, advice, and introducing me with the interesting field of vision-based control. Special thanks go to Dr. Prabir Barooah for his technical insights, support, and active participation in the work on robust pose estimation. I would like to thank Dr. Richard Lind and Dr. Ryan Causey for the collaborative research on autonomous aerial refueling and Dr. Carl Crane and Dr. Donald MacArthur for active participation on the daisy-chaining control research.

I would like to thank my colleagues from Nonlinear Controls and Robotics group for the technical discussions and frequent nerdy conversations which have been insightful and thought provoking. Also, I would like to thank my colleagues from Agricultural Robotics and Mechatronics Group (ARMG) for their support and encouragement.

TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS	4
LIST OF TABLES	7
LIST OF FIGURES	8
ABSTRACT	12
CHAPTER	
1 INTRODUCTION AND MOTIVATION	15
1.1 Vision-based Pose Estimation	15
1.2 Vision-based Control of Autonomous Systems	17
1.3 Dissertation Outline and Contributions	20
2 PEGUS: A NOVEL ALGORITHM FOR POSE ESTIMATION	23
2.1 Introduction	23
2.2 Related Work	24
2.3 Problem Statement and Approach	26
2.4 Proposed Algorithm	28
2.4.1 Rotation Estimation	28
2.4.2 Estimating Translation	31
2.4.2.1 Case A: unit translation	31
2.4.2.2 Case B: translation	32
2.5 Performance Evaluation	32
2.6 Discussion	35
2.6.1 Sampling of the Hypotheses	35
2.6.2 Robustness to Outliers	37
2.7 Conclusion	37
3 VISUAL SERVO CONTROL OF AN UNMANNED GROUND VEHICLE VIA A MOVING AIRBORNE MONOCULAR CAMERA	41
3.1 Introduction	41
3.2 Daisy-Chaining Based Regulation Control	42
3.2.1 Geometric Model	42
3.2.2 Euclidean Reconstruction	47
3.2.3 UGV Kinematics	50
3.2.4 Control Objective	51
3.2.5 Control Development	52
3.2.6 Stability Analysis	53
3.3 Multi-Reference Visual Servo Control of an Unmanned Ground Vehicle	55
3.3.1 Geometric Model	55

3.3.2	Euclidean Reconstruction	60
3.3.3	UGV Kinematics	64
3.3.4	Simulation Results	65
3.3.5	Concluding Remarks	67
4	A DAISY-CHAINING VISUAL SERVOING APPROACH WITH APPLICATIONS IN TRACKING, LOCALIZATION, AND MAPPING	71
4.1	Introduction	71
4.2	Daisy-Chaining Based Tracking Control	71
4.2.1	Problem Scenario	72
4.2.2	Geometric Relationships	74
4.2.3	Euclidean Reconstruction	77
4.2.4	Control Objective	80
4.2.5	Control Development	83
4.2.5.1	Open-loop error system	83
4.2.5.2	Closed-loop error system	83
4.2.6	Stability Analysis	84
4.3	Cooperative Tracking Control of a Nonholonomic Unmanned Ground Vehicle	85
4.3.1	Problem Scenario	86
4.3.2	Geometric Relationships	88
4.3.3	Euclidean Reconstruction	89
4.3.4	Control Objective	94
4.3.5	Control Development	96
4.4	Simultaneous Tracking, Localization and Mapping	98
4.4.1	Problem Scenario	98
4.4.2	Geometric Relationships	99
4.4.3	Euclidean Reconstruction	99
4.4.4	Tracking and Mapping	104
4.4.5	Simulation Results	104
4.5	Error Propagation in Daisy-Chaining	109
4.6	Concluding Remarks	112
5	CONCLUSIONS	117
5.1	Research Summary	117
5.2	Recommendations for Future Work	120
	REFERENCES	121
	BIOGRAPHICAL SKETCH	127

LIST OF TABLES

<u>Table</u>	<u>page</u>
2-1 Comparison of mean and variance of estimation error: (A) PEGUS (B) RANSAC+least squares, (C) non-linear mean shift.	36
3-1 Coordinate frames relationships for UGV regulation control.	44
3-2 Coordinate frames relationships for multi-reference UGV regulation control. . .	57
4-1 Coordinate frames relationships for 6-DOF planar object tracking control. . . .	75
4-2 Coordinate frame relationships for UGV tracking control.	89
4-3 Coordinate frame relationships for multi-reference UGV tracking control.	99

LIST OF FIGURES

<u>Figure</u>	<u>page</u>
2-1 Two views of a scene and the matched feature points between the images.	25
2-2 Histogram of the Euler angle data obtained from 5000 rotation hypotheses between the two images shown in Figure 2-1.	26
2-3 Multi-modal distribution of 5000 unit-translation hypotheses between the two images shown in Figure 2-1.	26
2-4 Histogram of the average overlap in 100 hypotheses generated.	29
2-5 Number of feature points tracked during a sequence of 970 images taken by a camera. The number of pairs of matched feature points between the first and the second image pair is 31 and that between the first and the 970-th image is 9.	33
2-6 Evaluation of the estimation accuracy achieved by the proposed PEGUS algorithm and its comparison with that achieved by RANSAC+LS and non-linear mean shift algorithm of [1]. RANSAC+LS means outlier rejection by RANSAC followed by re-estimation of the pose by feeding all the inliers to the normalized 8-point algorithm. Pmf of the rotation and unit translation estimation errors are computed from 9000 samples of the error obtained from the 9000 image pairs.	35
2-7 Comparison of the computation time required by the 3 algorithms. The pmf is estimated from 9000 samples of computation time.	35
2-8 Robustness comparison of the presented PEGUS algorithm with RANSAC+LS and non-linear mean shift algorithm in terms of the rotation estimation accuracy using synthetic data with (a) 10% feature outliers, (b) 20% feature outliers, (c) 30% feature outliers, (d) 40% feature outliers, (e) 50% feature outliers, (f) 60% feature outliers, (g) 70% feature outliers, (h) 80% feature outliers, and (i) 90% feature outliers. Pmf of the rotation estimation error is computed from 100 samples of the error obtained from the 100 image pairs.	38
2-9 Robustness comparison of the presented PEGUS algorithm with RANSAC+LS and non-linear mean shift algorithm in terms of the translation estimation accuracy using synthetic data with (a) 10% feature outliers, (b) 20% feature outliers, (c) 30% feature outliers, (d) 40% feature outliers, (e) 50% feature outliers, (f) 60% feature outliers, (g) 70% feature outliers, (h) 80% feature outliers, and (i) 90% feature outliers. Pmf of the translation estimation error is computed from 100 samples of the error obtained from the 100 image pairs.	39
2-10 (a) Mean rotation estimation error and (b) mean translation estimation error for the presented PEGUS algorithm, RANSAC+LS, and non-linear mean shift algorithm using the pose estimation results for synthetic data of varying feature outliers (10% - 90%) presented in Figs. 2-8 and 2-9.	40

3-1	Camera coordinate frame relationships: A moving airborne monocular camera (coordinate frame \mathcal{I}) hovering above an UGV (coordinate frame \mathcal{F}) while viewing a fixed reference object (coordinate frame \mathcal{F}^*) regulates an UGV to the desired pose (coordinate frame \mathcal{F}_d) captured by a priori located camera (coordinate frame \mathcal{I}_R).	43
3-2	Urban scenario describing regulation of an UGV to the desired pose using an airborne camera.	45
3-3	Camera to reference object relationships: A monocular camera (coordinate frame \mathcal{I}) viewing a stationary reference object (coordinate frame \mathcal{F}_i^*) such that a stationary object can leave the camera FOV as the new object enters the FOV, while the stationary reference camera (coordinate frame \mathcal{I}_R) is assumed to view the stationary reference object \mathcal{F}_1^* .	56
3-4	Camera to UGV relationships: A monocular camera (coordinate frame \mathcal{I}) hovering above an UGV (coordinate frame \mathcal{F}) while viewing a stationary reference object (coordinate frame \mathcal{F}_n^*) regulates an UGV to the desired pose (coordinate frame \mathcal{F}_d) known a priori in the reference camera (coordinate frame \mathcal{I}_R). The stationary pose (coordinate frame \mathcal{F}_r) corresponds to a snapshot of the UGV visible from the reference camera (coordinate frame \mathcal{I}_R).	57
3-5	Euclidean space trajectory of the moving camera \mathcal{I} , initial and final position of the time-varying UGV $\mathcal{F}(t)$, and desired UGV \mathcal{F}_d . $\mathcal{F}(0)$ denotes the initial position of the UGV, $\mathcal{I}(0)$ denotes the initial position of the moving camera, $\mathcal{I}(t)$ denotes the time-varying position of the moving camera, \mathcal{F}_i^* where $i = 1, 2, \dots, 7$ denotes the stationary reference objects, and $\mathcal{F}(t)$ denotes the regulated position of the UGV coincident with the desired UGV \mathcal{F}_d .	68
3-6	Linear (i.e. $e_1(t)$ and $e_2(t)$) and angular (i.e. $e_3(t)$) regulation error.	69
3-7	Linear (i.e. $v_c(t)$) and angular (i.e. $\omega_c(t)$) velocity control inputs.	69
3-8	Linear (i.e. $e_1(t)$ and $e_2(t)$) and angular (i.e. $e_3(t)$) regulation error in presence of an additive white Gaussian noise.	70
3-9	Linear (i.e. $v_c(t)$) and angular (i.e. $\omega_c(t)$) velocity control inputs in presence of an additive white Gaussian noise.	70
4-1	Geometric model for a moving camera (coordinate frame \mathcal{I}), moving target (coordinate frame \mathcal{F}) and stationary reference camera (coordinate frame \mathcal{I}_R).	73

4-2	Geometric model for a moving camera, moving UGV, and stationary reference camera: A moving camera (coordinate frame \mathcal{I}_M) records the desired trajectory of an UGV (coordinate frame $\mathcal{F}_d(t)$) with respect to the stationary reference object \mathcal{F}^* while stationary coordinate frame \mathcal{F}_s represents a snapshot of an UGV along the desired trajectory taken by $\mathcal{I}_R = \mathcal{I}_M(t) _{t=T}$. A moving camera (coordinate frame \mathcal{I}) views the current UGV (coordinate frame $\mathcal{F}(t)$) and the stationary reference object \mathcal{F}^*	87
4-3	Geometric model showing a snapshot of an UGV along the desired trajectory (coordinate frame \mathcal{F}_s) taken by $\mathcal{I}_R = \mathcal{I}_M(t) _{t=T}$. A current camera (coordinate frame \mathcal{I}) viewing the time-varying UGV (coordinate frame \mathcal{F}) while observing the set of feature points attached to \mathcal{F}^*	89
4-4	Geometric model showing a moving camera (coordinate frame \mathcal{I}_M) recording the desired trajectory of an UGV (coordinate frame $\mathcal{F}_d(t)$) with respect to the stationary reference object \mathcal{F}^* while stationary coordinate frame \mathcal{F}_s represents a snapshot of an UGV along the desired trajectory taken by $\mathcal{I}_R = \mathcal{I}_M(t) _{t=T}$	90
4-5	Geometric model for a moving camera, moving UGV, and stationary reference camera: A moving camera (coordinate frame \mathcal{I}_M) records the desired trajectory of an UGV (coordinate frame $\mathcal{F}_d(t)$) with respect to the stationary reference object \mathcal{F}_1^* while stationary coordinate frame \mathcal{F}_s represents a snapshot of an UGV along the desired trajectory taken by $\mathcal{I}_R = \mathcal{I}_M(t) _{t=T}$. A moving camera (coordinate frame \mathcal{I}) views the current UGV (coordinate frame $\mathcal{F}(t)$) and the stationary reference object \mathcal{F}_j^*	100
4-6	A simplified equivalent model showing a moving camera (coordinate frame \mathcal{I}) observing the current UGV (coordinate frame $\mathcal{F}(t)$) and the stationary reference object \mathcal{F}_j^* and the pose of \mathcal{F}_j^* is expressed in terms of \mathcal{I}_R	103
4-7	Euclidean space trajectory of the feature points attached to the current (i.e. $\mathcal{F}(t)$) and desired (i.e. $\mathcal{F}_d(t)$) UGV taken by \mathcal{I} and \mathcal{I}_M , respectively and the time-varying trajectory of the current and reference camera, \mathcal{I} and \mathcal{I}_M , respectively. $\mathcal{F}(0)$ denotes the initial position of the current UGV, $\mathcal{F}(t)$ denotes the time-varying position of the current UGV, $\mathcal{F}_d(0)$ denotes the initial position of the desired UGV, $\mathcal{I}(0)$ denotes the initial position of the current camera, $\mathcal{I}(t)$ denotes the time-varying position of the current camera, $\mathcal{I}_M(0)$ denotes the initial position of the time-varying reference camera, $\mathcal{I}_M(t)$ denotes the time-varying position of the time-varying reference camera, and \mathcal{F}_1^* , \mathcal{F}_2^* , and \mathcal{F}_3^* denote the position of the stationary reference objects.	106
4-8	Linear (i.e., $e_1(t)$ and $e_2(t)$) and angular (i.e., $e_3(t)$) tracking error.	107
4-9	Linear (i.e., $v_c(t)$) and angular (i.e., $\omega_c(t)$) velocity control inputs.	107
4-10	Linear (i.e., $e_1(t)$ and $e_2(t)$) and angular (i.e., $e_3(t)$) tracking error in presence of an additive white Gaussian image noise.	108

4-11	Linear (i.e., $v_c(t)$) and angular (i.e., $\omega_c(t)$) velocity control inputs in presence of an additive white Gaussian image noise.	108
4-12	Results of localization of the current UGV attached to $\mathcal{F}(t)$ and mapping of reference targets attached to \mathcal{F}_1^* , \mathcal{F}_2^* , and \mathcal{F}_3^* expressed in constant reference frame \mathcal{I}_R . Specifically, trajectory (1) shows the time-varying pose of the moving camera attached to $\mathcal{I}(t)$, trajectory (2) shows the time-varying pose of the moving camera attached to $\mathcal{I}_M(t)$, and trajectory (3) shows the time-varying pose of the current UGV attached to $\mathcal{F}(t)$ measured in the stationary reference camera frame \mathcal{I}_R . $\mathcal{F}(0)$ denotes the initial position of the current UGV and \mathcal{F}_1^* , \mathcal{F}_2^* , and \mathcal{F}_3^* denote the position of the stationary reference objects.	109
4-13	A simulation scenario depicting the circular trajectory of camera and a set of stationary reference objects \mathcal{F}_i^* where $i = 1, \dots, 8$	111
4-14	Error propagation in daisy-chaining pose estimation method in absence of feature point noise after 240 daisy-chains by traversing the circular trajectory 30 times.	112
4-15	A simulation scenario depicting the estimated camera trajectory in presence of white Gaussian image noise and a set of stationary reference objects \mathcal{F}_i^* where $i = 1, \dots, 8$	113
4-16	Error propagation in daisy-chaining pose estimation method in presence of white Gaussian noise after 240 daisy-chains by traversing the circular trajectory 30 times.	114
4-17	A simulation scenario depicting the estimated camera trajectory in presence of white Gaussian image noise by updating the camera position at the end of each circular trajectory and a set of stationary reference objects \mathcal{F}_i^* where $i = 1, \dots, 8$	115
4-18	Error propagation in daisy-chaining pose estimation method in presence of white Gaussian noise by updating the camera position at the end of each circular trajectory.	116

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

A DAISY-CHAINING APPROACH FOR VISION-BASED CONTROL AND
ESTIMATION

By

Siddhartha S. Mehta

April 2010

Chair: Dr. Warren E. Dixon
Co-Chair: Dr. Prabir Barooah
Major: Mechanical Engineering

The research presented in this dissertation monograph lies within the general scope of guidance, navigation, and control of autonomous systems and centers around the design and analysis of visual servo control strategies and vision-based robust position and orientation (i.e., pose) estimation. The motivation behind the presented research is to enable a vision system to provide robust navigation and control of autonomous agents operating over a large area. In order to enable vision systems to provide pose estimates over a large area, a new *daisy-chaining* method is developed. By developing multi-view geometry, or photogrammetry, based concepts relationships are established between the current pose of an agent and the desired agent pose, when the desired agent is out of the camera field-of-view (FOV). The daisy-chaining method is limited by the need to maintain a single reference object that is contained in both the current view and the final view of the desired pose of the vehicle. To overcome this limitation, the daisy-chaining method is extended to allow multiple reference objects to enter and leave the camera FOV, allowing theoretically infinite daisy-chaining and hence an unrestricted applicative area for an UGV. Error propagation analysis for the daisy-chaining method, which resembles a ‘dead-reckoning’ scheme, shows the method is susceptible to image noise and feature point outliers. To address the local pose estimation problem, a statistical method is

The accuracy of any vision-based control and estimation problem largely depends on accurate feature point information. Feature point errors will result in an erroneous pose estimation that could potentially affect the stability and performance of the control and estimation methods. An accurate pose estimation is a non-trivial problem, especially when real-time requirements prohibit computationally complex algorithms. Chapter 2 illustrates a novel method, PEGUS, for estimating the relative pose between two images captured by a calibrated camera. The method, based on the statistical theory, utilizes redundant feature points in the captured images to develop a robust pose estimate. Experimental results indicate markedly better performance over existing popular methods such as RANSAC and nonlinear mean shift algorithm, and the non-iterative structure of the algorithm makes it suitable in real-time applications.

Control of a moving object using a stationary camera and *vice versa* are well attended problems in the literature of visual servo control and various solutions exist for a class of autonomous systems. However, control of a moving object using the image feedback from a moving camera has been a well-known problem due to the unknown relative velocity associated with moving camera and moving object. In Chapter 3, a collaborative visual servo controller, the daisy-chaining method, is developed with an objective to regulate a sensor-less unmanned ground vehicle (UGV) to a desired pose utilizing the feedback from a moving airborne monocular camera system. Multi-view photogrammetric methods are used to develop relationships between different camera frames and UGV coordinate systems, and Lyapunov-based methods are used to prove asymptotic regulation of an UGV.

Another technical challenge when using a vision system for autonomous systems is that the given feature points can leave the camera FOV. In order to address the issue of features leaving the FOV an extension of the method developed in Chapter 3 is provided by considering multiple reseeding feature points. The presented multi-reference daisy-chaining scheme enables the UGV/camera pair to operate over an arbitrarily large

area. Simulation results are provided that illustrate the performance of the developed cooperative control scheme.

Building on the results in Chapter 3, the complex problem of cooperative visual servo tracking control is formulated in Chapter 4 with an objective to enable an UGV to follow a desired trajectory encoded as a sequence of images utilizing the image feedback from a moving airborne monocular camera system. The association as well as the relative velocity problem is addressed by introducing a daisy-chaining structure to link a series of projective homographies and expressing them in a constant reference frame. An adaptive parameter update law is employed to actively compensate for the lack of object model and depth measurements. Based on the open-loop error system, a tracking control law is developed through the application of Extended Barbalat's lemma in the Lyapunov-based framework to yield an asymptotic stability. The tracking results are extended to include reseeded stationary feature points by formulating additional projective homography relationships to provide an unrestricted applicative area for the UGV/camera pair. Simulation results are provided demonstrating the tracking control of an UGV in presence of multiple stationary reference objects and visual simultaneous localization and mapping (vSLAM) results are achieved by fusing the daisy-chaining method with the geometric reconstruction scheme.

Since the development provided in Chapters 3 and 4 assumes a stationary object can leave the camera FOV and a new reference object enters the FOV, it is necessary to determine the pose of the new reference object with respect to the receding object in order to provide the pose information of a moving agent such as an UGV or the camera itself. Therefore, the error in pose measurement between the stationary reference objects could propagate through the subsequent reference objects leading to large localization errors. The error propagation is analyzed in Chapter 4 by performing a numerical simulation and possible solutions are provided, along with simulation results, to mitigate the error propagation in daisy-chaining.

CHAPTER 1 INTRODUCTION AND MOTIVATION

The research presented in this dissertation monograph lies within the general scope of guidance, navigation, and control of autonomous systems and centers around the design and analysis of visual servo control strategies and vision-based robust position and orientation (i.e., pose) estimation. The Euclidean pose of an agent is typically required for autonomous navigation and control. Often the pose of an agent is determined by a global positioning system (GPS) or an inertial measurement unit (IMU). However, GPS may not be available in many environments, and IMUs can drift and accumulate errors over time in a similar manner as dead reckoning. Given recent advances in image extraction/interpretation technology, an interesting approach to overcome the pose measurement problem is to utilize a vision system. Specifically, rather than obtain an inertial measurement of the agent, vision systems can be used to recast some navigation and control problems in terms of the image space where the goal pose is compared to the relative pose via multiple images.

The aim of this chapter is to provide a reader with background in the area of navigation and control of autonomous systems using vision (i.e., camera) as a sensor modality. A motivation behind the presented research is established by describing the problem scenarios and posing the open problems. The chapter is organized in three sections; Sections 1.1 and 1.2 provide introduction to vision-based pose estimation and control of autonomous systems, and the outline of dissertation along with the contributions of presented research are detailed in Section 1.3.

1.1 Vision-based Pose Estimation

Motivated by practical applications such as autonomous guidance, navigation, and control, various techniques have been developed such as visual servo control, visual odometry, structure from motion, etc. Common to all these methods is the problem of estimating the relative pose (rotation and translation) between two images. For a

monocular camera, the rotation and direction of translation are estimated, whereas in case of a stereo camera system, the rotation and the translation vector are estimated.

Existing methods for pose estimation use point correspondence between the two views, which is provided by a feature-tracking algorithm, such as the KLT algorithm [2]. Given a minimal set of point correspondence, the relative pose can be estimated by a number of algorithms (the eight point algorithm [3], the five point algorithm [4], etc.). However, point correspondences as the output of the feature tracker invariably contain gross mismatches or large errors in feature point locations, which are commonly referred to as *outliers*. A central issue in accurate pose estimation is devising robust estimators that can reject such outliers. The most popular solution to this problem has been hypothesize-and-test methods, such as RANSAC [5] and its variants: MLESAC [6], PROSAC [7], GOODSAC [8], pre-emptive RANSAC [9], etc. In these methods, hypotheses are generated by randomly choosing a minimal set of corresponding feature point pairs that are required to generate a hypothesis. A hypothesis is typically scored based on how many of the observations are well-explained by it, and the one with the best score is declared as the desired estimate. Most of the extensions to the basic RANSAC scheme focus on reducing the computation time, since generating a large number of hypotheses (which is required to obtain a good estimate with high probability) and scoring them is computationally expensive.

RANSAC and other hypothesize-and-test methods choose only one of the many hypotheses that are or can be generated. All other hypotheses are ignored, even those that may be quite close to the true pose. Each hypothesis can be thought of as a noisy “measurement” of the relative pose that is to be estimated. In principle, one should be able to average these measurements in an appropriate sense to compute a more accurate estimate than any of the individual measurements (i.e., hypotheses).

In Chapter 2, a novel robust pose estimation algorithm is presented based on the idea above. There are two hurdles that impede the development of this idea. First, many of

the pose hypotheses will be corrupted by outliers, which will have poor accuracy, so that including them in the averaging process may lead to little improvement, if any. The second difficulty is that since a pose is not an element of a vector space, it is not clear how to average multiple noisy measurements of a pose.

To address these challenges, pose estimation problem is treated as estimating the rotation and (unit) translation separately. By expressing the rotation hypotheses as unit quaternions that lie on the unit sphere in 4-dimensions, and computing a histogram of this data by dividing the sphere into bins, the dominant cluster, or mode, of the rotations is identified. A subset of the rotations that are within a small geodesic distance of the mode is then extracted. These “low-noise” rotation hypotheses are then averaged using a method developed by Moakher [10] to produce the estimate of the rotation. Estimating unit translations proceed in exactly the same way, except now the data lies on the surface of the unit sphere in 3 dimensions. When translation (direction as well as magnitude) is available, say from a stereo camera, mode estimation and averaging is simpler since the data lies in a vector space. Because of the role played by gridding of the unit sphere in 3 or 4 dimensions, the proposed algorithm is called the Pose Estimation by Gridding of Unit Spheres (PEGUS) algorithm.

1.2 Vision-based Control of Autonomous Systems

The Euclidean position and orientation (i.e., pose) of an unmanned ground vehicle (UGV) is typically required for autonomous navigation and control. The vision-based control schemes can be benefitted from the robust pose estimation method developed in Chapter 2.

Some examples of image-based visual servo control of mobile vehicles include: [11–25]. Previous pure image-based visual servo control results have a known problem with potential singularities in the image-Jacobian, and since the feedback is only in the image-space, these methods may require impossible Euclidean motions. Motivated by the desire to eliminate these issues, some efforts have been developed that combine

reconstructed Euclidean information and image-space information in the control design. The Euclidean information can be reconstructed by decoupling the interaction between translation and rotation components of a homography matrix. This homography-based method yields an invertible triangular image-Jacobian with realizable Euclidean motion. Homography-based visual servo control results that have been developed for UGV include: [26–31]. In [29], a visual servo controller was developed to asymptotically regulate the pose of an UGV to a constant pose defined by a goal image, where the camera was mounted on-board an UGV (i.e., the camera-in-hand problem). The camera on-board result in [29] was extended in [26] to address the more general tracking problem. In [27], a stationary overhead camera (i.e., the camera-to-hand or fixed camera configuration) was used to regulate a UGV to a desired pose.

The development in Chapter 3 is motivated by the desire to address the well-known problem of controlling a moving object using a moving camera. A moving airborne monocular camera (e.g., a camera attached to a remote controlled aircraft, a camera mounted on a satellite) is used to provide pose measurements of a moving sensorless UGV relative to a goal configuration. Distinguishing the relative velocity between the moving UGV and the moving camera presents a significant challenge. Geometric constructs developed for traditional camera-in-hand problems are fused with fixed-camera geometry to develop a set of Euclidean homographies so that a measurable error system for the nonholonomic UGV can be developed. The resulting open-loop error system is expressed in a form that is amenable to a variety of UGV controllers.

In addition to vision-based control problem, image feedback can be used to localize and map the environment (i.e., visual simultaneous localization and mapping - vSLAM) [32–38]. vSLAM is used in the applications where the camera is the main sensor used to estimate the location of a robot in the world, as well as estimate and maintain estimates of surrounding terrain or features. Often a measure of estimate uncertainty is also

maintained. vSLAM is a broad topic with varied approaches. See [32] and references therein for a recent survey.

There are many overlapping ways to categorize vSLAM approaches. Some authors (e.g., [33, 34, 36]) make a distinction between “local vSLAM” and “global vSLAM”. In this categorization, local vSLAM is concerned with estimating the current state of the robot and world map through matching visual features from frame to frame, and global vSLAM is concerned with recognizing when features have been previously encountered and updating estimates of the robot and map (sometimes referred to as “closing loops”). To address both issues, many researchers use pose invariant features, such as SIFT [39], which can be accurately matched from frame to frame or from multiple camera viewpoints. Many vSLAM approaches use probabilistic filters (e.g., extended Kalman filter or particle filter) [32, 33, 36–38], typically estimating a state vector composed of the camera/robot position, orientation, and velocity, and the 3D coordinates of visual features in the world frame. An option to a filtered based approach is the use of epipolar geometry [34, 35]. A final possible category are methods that build a true 3D map (i.e., a map that is easily interpreted by a human being such as walls or topography) [33, 34, 36–38], and those that build a more abstract map that is designed to allow the camera/robot to accurately navigate and recognize its location, but not designed for human interpretation.

Chapter 4 utilizes a new daisy-chaining method developed in Chapter 3 for vision-based tracking control of a UGV, while also providing localization of the moving camera and moving object in the world frame, and mapping the location of static landmarks in the world frame. Hence, this approach can be used in vSLAM of the UGV, with applications toward path planning, real time trajectory generation, obstacle avoidance, multi-vehicle coordination control and task assignment, etc. By using the daisy-chaining strategy, the coordinates of static features out of the field-of-view (FOV) can also be estimated. The estimates of static features can be maintained as a map, or can be used as measurements in existing vSLAM methods.

1.3 Dissertation Outline and Contributions

Chapter 2 describes a novel robust algorithm for estimation of the relative pose between two calibrated images, called Pose Estimation by Gridding of Unit Spheres (PEGUS). The focus of this chapter is to develop a computationally deterministic pose estimation method that is robust to feature outliers. Pose estimation results using PEGUS are compared with popular methods such as RANSAC and nonlinear mean-shift algorithm using an indoor image sequence and synthetic feature point data. The results in Chapter 2 demonstrate an improved performance of PEGUS against RANSAC+least squares as well as non-linear mean shift method, both in terms of the estimation accuracy and computation time. By virtue of non-iterative formulation underlying the deterministic structure of PEGUS, the computation time is more predictable than that of RANSAC and non-linear mean shift algorithm, thus making it amenable to a variety of real-time applications. Vision-based control of autonomous systems typically require pose estimation between multiple views captured by a camera system. Robust pose estimation results developed in Chapter 2 can be used for such applications.

Control of a moving object using a stationary camera and vice versa are well attended problems in the literature of visual servo control and numerous solutions exist for a general class of autonomous systems. However, control of a moving object using the image feedback from a moving camera has been a well-known problem due to the unknown relative velocity associated with moving camera and moving object. In Chapter 3, a collaborative visual servo controller, which is coined the daisy-chaining method, is developed with an objective to regulate a sensor-less unmanned ground vehicle (UGV) to a desired pose utilizing the feedback from a moving airborne monocular camera system.

The contribution of research in Chapter 3 is the development of multi-view geometry, or photogrammetry, based concepts to relate the coordinate frames attached to moving camera, moving UGV, and desired UGV pose specified by an a priori image. Geometric constructs developed for traditional camera-in-hand problem are fused with fixed-camera

geometry to develop a set of Euclidean homographies. Due to intrinsic physical constraints, one of the resulting Euclidean homographies is not measurable through a set of spatiotemporal images as the corresponding projective homography can not be developed. Hence, the new geometric formulations, termed virtual homography, are conceived to solve for the homography in order to develop a measurable error system for the nonholonomic UGV. Asymptotic regulation results are proved using the Lyapunov-based stability analysis.

Chapter 3 also illustrates a framework to achieve asymptotic regulation of an UGV based on the scenario that the given reference objects can leave camera FOV while another reference object enters FOV. The controller is developed - with the underlying geometrical constructs that daisy-chain multiple reference objects - such that the airborne camera does not require to maintain a view of the static reference object; therefore the airborne camera/UGV pair can navigate over an arbitrarily large area. Also, by taking leverage of the geometric reconstruction method, the assumption of equal Euclidean distance of the features for UGV and reference object is relaxed.

Building on the results in Chapter 3, the complex problem of cooperative visual servo tracking control is formulated with an objective to enable an UGV to follow a desired trajectory encoded as a sequence of images utilizing the image feedback from a moving airborne monocular camera system. The desired trajectory of an UGV is recorded by a moving airborne monocular camera \mathcal{I}_M traversing an unknown time-varying trajectory. The control objective is to track an UGV along the desired trajectory using the image feedback from a moving airborne camera \mathcal{I} that may traverse different trajectory than that of \mathcal{I}_M . The association as well as the relative velocity problem is addressed by introducing a daisy-chaining structure to link a series of projective homographies and expressing them in a constant reference frame. An adaptive parameter update law is employed to actively compensate for the lack of object model and depth measurements. Based on the open-loop error system, a tracking control law is developed through the

application of Extended Barbalats lemma in the Lyapunov-based framework to yield an asymptotic stability.

The tracking results are extended to include the reseeding reference object by formulating an additional projective homography relationship to provide an unbounded applicative area of operation. The theoretical development in Chapter 3 manifests the coalescence of daisy-chaining controller and newly formed geometric reconstruction technique towards application in visual simultaneous localization and mapping (vSLAM). The chapter also provides simulation results demonstrating the propagation of pose estimation error in daisy-chaining control scheme and discusses suitable methods for limiting the error propagation.

Summary of research and future work recommendations are provided in Chapter 5.

CHAPTER 2 PEGUS: A NOVEL ALGORITHM FOR POSE ESTIMATION

2.1 Introduction

The focus of research in this chapter is to develop a computationally deterministic pose estimation method that is robust to feature outliers. A novel robust algorithm is developed for estimation of the relative pose between two calibrated images, which is coined as *Pose Estimation by Gridding of Unit Spheres* (PEGUS). The key idea behind the method is, if there are M matched pairs of feature points between two views, one can compute a maximum of $\binom{M}{P}$ possible pose hypotheses by using a P -point algorithm. The developed algorithm selects a subset of “low-noise” hypotheses by empirically estimating the probability density function of the rotation and translation random variables, and averages them, conforming manifold constraints, to compute a pose estimate.

In contrast to hypothesize-and-test methods such as RANSAC [5], the proposed algorithm averages the information from a number of hypotheses that are likely to be close to the true pose. As a result, it comes up with a more accurate estimate than that returned by RANSAC-type methods. The proposed algorithm has certain similarities with the non-linear mean shift algorithm proposed in [1]; the similarities and differences between the two are discussed in Section 2.2. Another advantage of the PEGUS algorithm is that it does not involve any iterative search, so that the time required for its execution is not only quite small but also highly predictable. This aspect of the algorithm makes it suitable for real-time applications.

In tests with real image data, the proposed algorithm significantly outperforms RANSAC as well as the non-linear mean shift algorithm of [1]. Improvement is seen not only in estimation accuracy but also in computational time and predictability of execution time. Robustness of the presented algorithm is compared with RANSAC and non-linear mean shift algorithm by varying the number of outliers from 10% to 90%.

The rest of the chapter is organized as follows: Section 2.2 describes some of the prior work that is relevant to the presented approach. Section 2.3 explains the approach behind the proposed algorithm. The proposed algorithm is described in detail in Section 2.4. Experimental results are presented in Section 2.5 and concluding remarks are presented in Section 2.7.

2.2 Related Work

There are certain similarities between PEGUS and the non-linear mean shift algorithm by Subbarao *et. al.* [1], in which a set of generated hypotheses are used to construct a kernel-based estimate of the probability density function (pdf) of the pose hypothesis in $SE(3)$. A non-linear version of the mean-shift algorithm is then used to iteratively search for the mode of this pdf starting from an arbitrary initial condition. The identified mode is declared the pose estimate. Since all the hypotheses used to construct the pdf contributes to the mode, and the mode may not coincide with any of the hypotheses, the resulting estimate can be thought of as an average of the hypotheses, though the averaging is of an implicit nature. In short, the approaches in the proposed PEGUS algorithm as well as that in [1] treat the pose estimation problem as a clustering problem. Both construct estimates of the probability density (or mass) function from a set of generated hypotheses and returns an averaged hypothesis as the pose estimate rather than a single hypotheses from those generated.

Despite these similarities between the two approaches, there are significant differences between the proposed PEGUS algorithm and the non-linear mean shift algorithm of [1]. First, the PEGUS algorithm is far more robust to multi-modal densities of the generated hypotheses than mean shift. Experimental evidence suggests that the distribution of these hypotheses are typically multi-modal. As an example, all the possible hypotheses from 31 matched feature points between the image pair shown in Fig. 2-1 are computed using the normalized 8-point algorithm. Fig. 2-2 shows a histogram of the Euler angles obtained from the rotation hypotheses and Fig. 2-3 shows the unit translation

hypotheses (direction of translation), which are points on the surface of the unit sphere in 3-dimensions. Multi-modality of the distribution is clear from the figures. In such a situation, the iterative search involved in the mean shift algorithm will converge to a local maximum depending on the initial condition. In contrast, a histogram-based estimate of the pmf (probability mass function) of the hypotheses makes locating the global mode a trivial problem even with multi-modal densities. The pmf for the rotation hypothesis is constructed by gridding the surface of the unit sphere in 4 dimensions, on which unit quaternion representations of the rotations lie. The same approach works for unit translations as well, by gridding the surface of the unit sphere in 3 dimensions. If both magnitude and direction of translation can be estimated, the histogram is constructed by dividing a region of \mathbb{R}^3 into a number of cells.

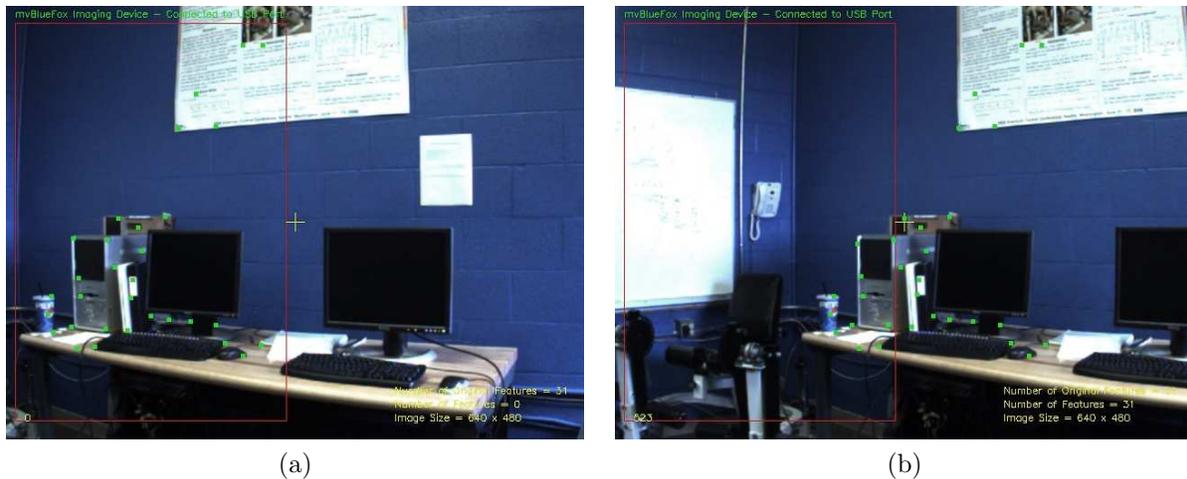


Figure 2-1. Two views of a scene and the matched feature points between the images.

The second major difference is that the non-linear mean shift algorithm returns the mode as the estimate, whereas the proposed method uses the mode only to identify a set of hypotheses that are likely to be close to the true pose. These “low-noise” hypotheses are then explicitly averaged in an appropriate manner to construct the final estimate. In addition, the proposed method does not involve iterative computation, whereas the mean-shift algorithm requires an iterative search for the mode. On the other hand, the non linear mean-shift algorithm is applicable to a wide variety of estimation problems

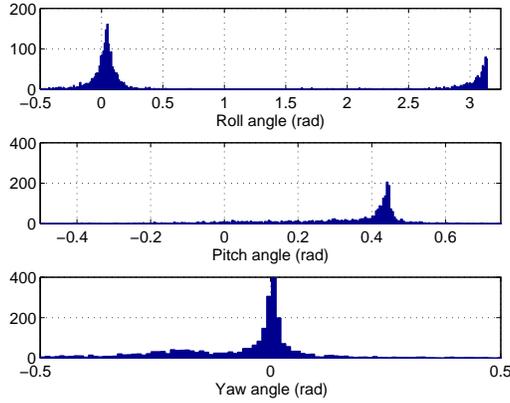


Figure 2-2. Histogram of the Euler angle data obtained from 5000 rotation hypotheses between the two images shown in Figure 2-1.

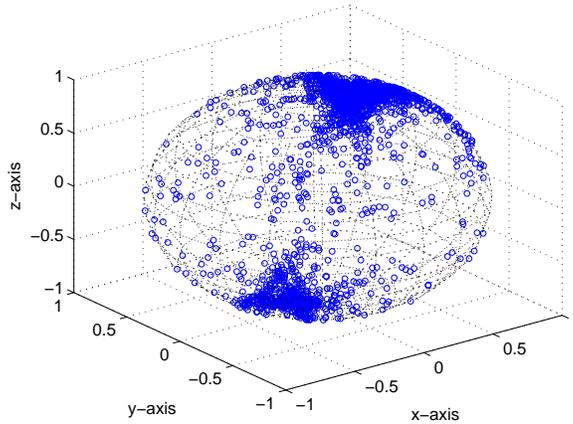


Figure 2-3. Multi-modal distribution of 5000 unit-translation hypotheses between the two images shown in Figure 2-1.

in which data lies on Riemannian manifolds [40], whereas the proposed method is only applicable to problems in which the data lies on spherical surfaces or real coordinate spaces.

2.3 Problem Statement and Approach

The objective is to develop a robust pose estimation algorithm using two images captured by a monocular camera (or four images if a pair of cameras are used) and without the knowledge of the scene. Let R denote the *true rotation* between two views

and t be the *true translation*. The translation can be a *unit* translation if scale information is not available.

If there are M pairs of feature points between two views captured by the camera and the minimal number of feature point pairs need to generate a hypothesis is P , then the total number of pose hypotheses that can be computed is $N_{\max} := \binom{M}{P}$. First n such hypotheses are generated, where n is typically much smaller than N_{\max} . Each pair of generated rotation and translation hypothesis is a “noisy measurement” of the true rotation R and true (unit) translation t , respectively. Some of these measurements, i.e., hypotheses, suffer from large inaccuracy, as seen from Figs. 2-2 and 2-3. The proposed approach is to select a subset of “low-noise” hypotheses from the set of all possible hypotheses so that they are close to the true rotation and translation, respectively. The low-noise hypotheses are then appropriately averaged to compute a pose estimate.

To facilitate extraction of the low-noise hypotheses, each rotation hypothesis is expressed in terms of its unit-quaternion representation. Since the unit quaternions q and $-q$ represent the same rotation, it is ensured that the unit-quaternion representation of a rotation hypothesis has the first component positive. That is, if $q = q_r + iq_1 + jq_2 + kq_3$, then $q_r > 0$. A unit quaternion representation of a rotation matrix can now be thought of as a unit-norm vector in \mathbb{R}^4 whose first component is positive. That is, it lies on the “top” half of the 3-sphere \mathbb{S}^3 . The d -sphere \mathbb{S}^d is defined as

$$\mathbb{S}^d := \{x = [x_1, \dots, x_{d+1}]^T \in \mathbb{R}^{d+1} \mid \|x\| = 1\} \quad (2-1)$$

where $\|\cdot\|$ denotes the Euclidean norm. Similarly, define

$$\mathbb{S}^{d+} = \{x \in \mathbb{R}^{d+1} \mid \|x\| = 1, x_1 \geq 0\}. \quad (2-2)$$

Therefore, each rotation hypothesis is an element of \mathbb{S}^{3+} . Similarly, each hypothesis of unit translation is an element of \mathbb{S}^2 . If scale information is available, translation hypotheses are elements of \mathbb{R}^3 instead of \mathbb{S}^2 .

Since each rotation hypothesis is a noisy measurement of the true rotation, the rotation hypotheses can be thought of as realizations of a random variable whose distribution is defined over the half-sphere \mathbb{S}^{3+} . By dividing the surface of the sphere \mathbb{S}^3 and counting the number of rotation hypotheses (rather, their unit-quaternion representation), the pmf of this random variable can be estimated. The mode of the pmf is a point in the bin that has the largest number of unit-quaternions. A subset of these quaternions that are within a predetermined geodesic distance of the mode is selected, and then averaged in an appropriate manner to obtain the final estimate of the rotation. Estimation of translations proceed in a similar manner. The algorithm is described in detail in the next section.

2.4 Proposed Algorithm

2.4.1 Rotation Estimation

Step 1: Hypotheses generation engine: The total number of possible pose hypotheses, N_{\max} is typically extremely large, since $N_{\max} = \binom{M}{P}$, where M is the number of point correspondence and P is the minimal number needed to generate a hypothesis. For example, even a small value of M , e.g., 21, with $P = 8$ yields $N_{\max} = 203490$. Processing such a large number of hypotheses is computationally infeasible. In addition, processing all of them is not necessary since most of these hypotheses are “correlated”, as they are generated from overlapping feature point sets. A sampling with replacement strategy is used to generate a number of hypotheses that have small correlations among one another. The number of such hypotheses to be generated, n , is a design parameter that has to be specified a-priori. However, even with a small value of n (≈ 100) the method yields beneficial results. The sampling strategy consists of selecting the first feature point pair at random from the M pairs, then selecting the second pair from the remaining $M - 1$ pairs, and so on until the P -th pair is selected. These P pairs of point correspondence are used to generate a hypothesis. This sampling procedure is repeated n times to generate n hypotheses, which are denoted by q_i, t_i , where q_i is an unit-quaternion and t_i is a

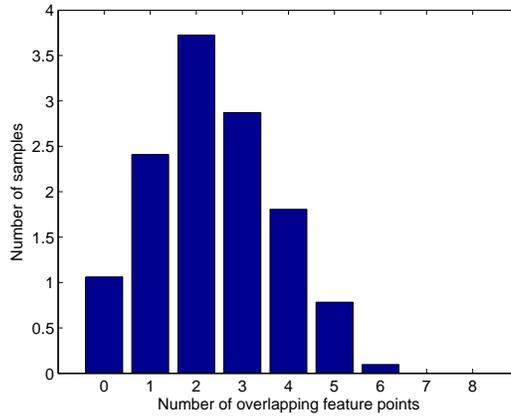


Figure 2-4. Histogram of the average overlap in 100 hypotheses generated.

translation vector (unit-norm or otherwise), for $i = 1, \dots, n$. The set of these n rotation hypotheses is denoted by S_q , and the set of translation hypotheses is denoted by S_t .

Figure 2-4 provides evidence of the small correlation among the hypotheses generated by the sampling strategy mentioned above. A set of hypotheses q_1, \dots, q_j is said to have an overlap of ℓ if there are exactly ℓ feature point pairs that are common to the sets of points used to generate the hypotheses q_1, \dots, q_j . The metric for measuring correlations among hypotheses is the number of overlap among them. The figure shows a histogram of the overlap between 100 hypotheses generated in this fashion, computed based on the average of 1000 random trials of the hypotheses generation.

Step 2: Estimating the mode: Each q_i is imagined to be the realization of a random variable \mathbf{q} with an unknown distribution defined over \mathbb{S}^{3+} . The 3-sphere \mathbb{S}^3 is divided into a number of regions of equal area, or bins, that are denoted by B_j , $j = 1, \dots, K_q$, where K_q is the (appropriately chosen) number of regions. The algorithm described in [41] is used for this purpose. The pmf of the random variable \mathbf{q} over the bins B_j , which is denoted by $p^{(q)}$, is an array of K_q numbers: $p_j^{(q)} = P(\mathbf{q} \in B_j)$, where P denotes probability. A histogram estimate $\hat{p}^{(q)}$ of the pmf $p^{(q)}$ is computed by counting the number of points q_i inside each bin: $\hat{p}_j^{(q)} = \frac{1}{n} \sum_{i=1}^n I_{B_j}(q_i)$, where $I_A(x)$ is the indicator function of the set A . That is, $I_A(x) = 1$ if $x \in A$ and 0 otherwise. A useful property of the histogram-based

estimate is that $\hat{p}_j^{(q)}$ is an unbiased estimate of $p_j^{(q)}$ even if the samples used to construct the estimates are correlated. Let B_{j^*} be the bin in which the pmf attains its maximum value, i.e., $j^* = \arg \max_j (\hat{p}_j^{(q)})$. If the number of bins K_q is large, then the center of B_{j^*} is taken as the estimate of the mode of the pmf $p^{(q)}$, which is denoted by $q^* \in \mathbb{S}^{3+}$. If a small K_q is chosen so that each bin may be large, the geodesic distances between the center of B_{j^*} and the q_i 's lying in B_{j^*} are computed and then averaged to identify the dominant cluster of points inside this bin. If there is such a set of points that form a dominant cluster, their center is chosen as the mode. The center is computed by taking the arithmetic mean of the unit-quaternions (thinking of them as 4 dimensional vectors) within the cluster and then normalizing the mean.

Step 3: Extracting low-noise measurements: Once the mode is identified, a subset $Q_q \subset S_q$ is selected that consists of those $\hat{q}_i \in S$ that satisfies

$$d_q(q^*, q_i) < \varepsilon_q, \quad (2-3)$$

where the distance function $d_q(\cdot, \cdot)$ is the Riemannian distance. The Riemannian distance between two rotations $q_1, q_2 \in \mathbb{S}^{3+}$ is given by

$$d(R_1, R_2) = \frac{1}{\sqrt{2}} \|\log(R_1^T R_2)\|_F, \quad (2-4)$$

where $R_1, R_2 \in SO(3)$ are the rotation matrix representation of q_1, q_2 , and the subscript F refers to the Frobenius norm.

Step 4: Averaging low-noise data: Let N_1 be the number of elements in the low-noise data set Q_q of rotations obtained as described above, and let R_i denote the rotation matrix corresponding to $q_i \in Q_q$. The *optimal* average of the rotation matrices $R_1 \dots R_{N_1}$ in the Euclidean sense is the matrix \hat{R} that satisfies [10]

$$\hat{R} = \operatorname{argmin}_{R \in SO(3)} \sum_{i=1}^{N_1} \|R_i - R\|_F^2. \quad (2-5)$$

It was shown by Moakher [10] that \hat{R} defined by (2-5) can be computed by the orthogonal projection of the arithmetic average $\bar{R} = \sum_{i=1}^{N_1} \frac{R_i}{N_1}$ onto the special orthogonal group $SO(3)$ by

$$\hat{R} = \bar{R}U \text{diag}\left(\frac{1}{\sqrt{\Lambda_1}}, \frac{1}{\sqrt{\Lambda_2}}, \frac{s}{\sqrt{\Lambda_3}}\right)U^T, \quad (2-6)$$

where the orthogonal matrix U is such that

$$\bar{R}^T \bar{R} = U^T D U \text{ and } D = \text{diag}(\Lambda_1, \Lambda_2, \Lambda_3), \quad (2-7)$$

and $s = 1$ if $\det \bar{R} > 0$ and $s = -1$ otherwise.

The matrix \hat{R} computed using (2-6) is the desired estimate of the true rotation R .

2.4.2 Estimating Translation

2.4.2.1 Case A: unit translation

The estimation scheme for unit translation is very similar to that for the rotation. The unit translation data $t_i \in S_t$, $i = 1, \dots, n$ represent realizations of the random variable \mathbf{t} with an unknown distribution defined over the 2-sphere S^2 . The 2-sphere S^2 is divided into a number of bins of equal area B_j , $j = 1, \dots, K_t$, where K_t is the (appropriately chosen) integer, by using the method described in [41]. A histogram estimate $\hat{p}^{(t)}$ of the pmf $p^{(t)}$, where $p_j^{(t)} := P(\mathbf{t} \in B_j)$ is then computed by counting the number of points t_i in B_j . When K_t is large, the mode of the unit translation distribution, denoted by t^* , is taken as the center of the bin B_{j^*} in which the pmf takes its maximum value: $j^* = \arg \max_j \hat{p}_j^{(t)}$. When the number of bins K_t is small, a method similar to the one used for rotations is used to estimate the mode. Once the mode t^* is identified, the low-noise data set Q_t is selected by choosing those $t_i \in S_t$ that satisfies

$$d_t(t^*, t_i) < \varepsilon_t, \quad (2-8)$$

where ε_t is a pre-specified small positive number, and $d_t(t_1, t_2)$ is the geodesic distance between the unit translation vectors t_1 and t_2 . Let N_2 be the number of elements in the

low-noise data set Q_t of the unit translations obtained above. The normalized arithmetic mean of the unit translations in the set Q_t , which is given by

$$\hat{t} = \frac{\sum_{i=1}^{N_2} \frac{t_i}{N_2}}{\left\| \sum_{i=1}^{N_2} \frac{t_i}{N_2} \right\|} \quad (2-9)$$

is taken as the estimate of the unit translation t .

2.4.2.2 Case B: translation

When scale information is available, e.g., using a stereo camera pair, t_i 's are hypotheses of the translation between the two views and are elements of \mathbb{R}^3 , not of \mathbb{S}^2 . In this case, histogram construction, mode estimation and low-noise hypotheses extraction is carried out in \mathbb{R}^3 , by dividing a particular volume of \mathbb{R}^3 into K_t bins of equal volume, with each bin being a cube with equal sides. The volume to grid is chosen so that all the hypotheses lie in it. The rest of the algorithm stays the same, except that the Euclidean norm of the difference is used as the distance function $d_t(\cdot, \cdot)$ in (2-8), and the normalization step in (2-9) is omitted.

2.5 Performance Evaluation

To test the performance of the proposed algorithm, 970 images of a stationary scene are captured by a moving monocular camera (make: Matrix Vision GmbH, model: mvBlueFox-120aC, resolution: 640×480) mounted on a 2-link planar manipulator. The 2-link manipulator is equipped with rotary encoders on each motor with 614,400 readings/revolution, which provides the ground truth of rotation and translation between any two time instants. Fig. 2-1 shows the initial and the final images of the scenes captured by the camera.

A multi-scale version of Kanade-Lucas-Tomasi (KLT) [2] feature point tracker is implemented to track the features in successive images as the camera undergoes rotation and translation. 31 feature points were detected in the first image of the sequence. Due to the camera motion, restricted FOV, feature point tracking errors, etc., the number of tracked feature points reduces with time (see Fig. 2-5).

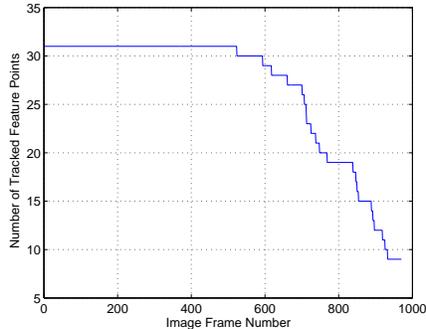


Figure 2-5. Number of feature points tracked during a sequence of 970 images taken by a camera. The number of pairs of matched feature points between the first and the second image pair is 31 and that between the first and the 970-th image is 9.

The performance of the PEGUS algorithm is evaluated by comparing the relative pose estimates to the ground truth. The performance of the proposed algorithm is also compared with that of RANSAC+least squares (outlier rejection by RANSAC followed by re-estimation of the pose by feeding all the inlier data into the normalized 8-point algorithm) and the non-linear mean shift algorithm [1]. For all the three methods, a normalized 8-point algorithm is used to generate relative pose hypothesis (rotation and unit translation) between the two frames from a minimal number of 8 pairs of matched feature points. The same set of hypotheses are used as inputs to the PEGUS algorithm and the non-linear mean shift.

The parameters used for the proposed method are $K_q = 11$, $\epsilon_q = 0.0223$, $K_t = 7$, $\epsilon_t = 0.017$. The parameter n is nominally set to 100, except when the number of matched feature points M between two views is so small that $N_{\max} < n$, in which case $n = N_{\max}$ is used. For the non-linear mean shift algorithm, a bandwidth of 0.7 and a Gaussian kernel with mean 0 and variance 1 was used, with a stopping criteria that the norm of the gradient of the mean-shift vector is less than 7×10^{-4} . For RANSAC, a Sampson distance of 0.005 as a threshold between the data points and the model is used.

The performance of each of the three pose estimation algorithms (PEGUS, RANSAC+least squares, non-linear mean shift) is evaluated in the following manner. A total of 9000

image pairs are selected arbitrarily from the 970 images captured. The true rotation and translation between the frames in the i -th image pair are denoted as $R(i)$ and $t(i)$, respectively, which are obtained from encoder measurements. The rotation and translation estimation error for the i -th image pair are defined as

$$\begin{aligned} e_R(i) &= \|I - R(i)^T \hat{R}(i)\| \\ e_t(i) &= \|t(i) - \hat{t}(i)\|, \end{aligned} \tag{2-10}$$

where $\hat{R}(i)$ and $\hat{t}(i)$ are the estimates of the rotation $R(i)$ and unit translation $t(i)$, $\|\cdot\|$ denotes the 2-norm, and I denotes a $\mathbb{R}^{3 \times 3}$ identity matrix. For each of the three algorithms, the 9000 samples of the errors (2-10) are computed, and are used to estimate the pmf of the rotation and translation errors that results from the particular algorithm. The error probabilities for the three methods: PEGUS, RANSAC+least squares, and non-linear mean shift, are shown in Fig. 2-10(a) and (b). The mean and standard deviation of the rotation and translation errors are tabulated in Table 2-1. The figure and the table shows that the estimation error with the proposed method is significantly lower than that with RANSAC as well as non-linear mean shift.

A computational time comparison for the three methods is shown in Fig. 2-7, which indicates that the proposed algorithm is also faster than both RANSAC and mean shift algorithms. All computations were done using MATLAB in a desktop Linux machine. In addition, the computational time of PEGUS is more predictable compared to that of RANSAC and mean-shift. The reason is that in contrast to both RANSAC and mean shift, PEGUS does not involve iterative search. It should be noted that these trends may change depending on the parameters of the algorithms. For example, execution time of the mean shift algorithm may be reduced by increasing the bandwidth, which may affect its accuracy.

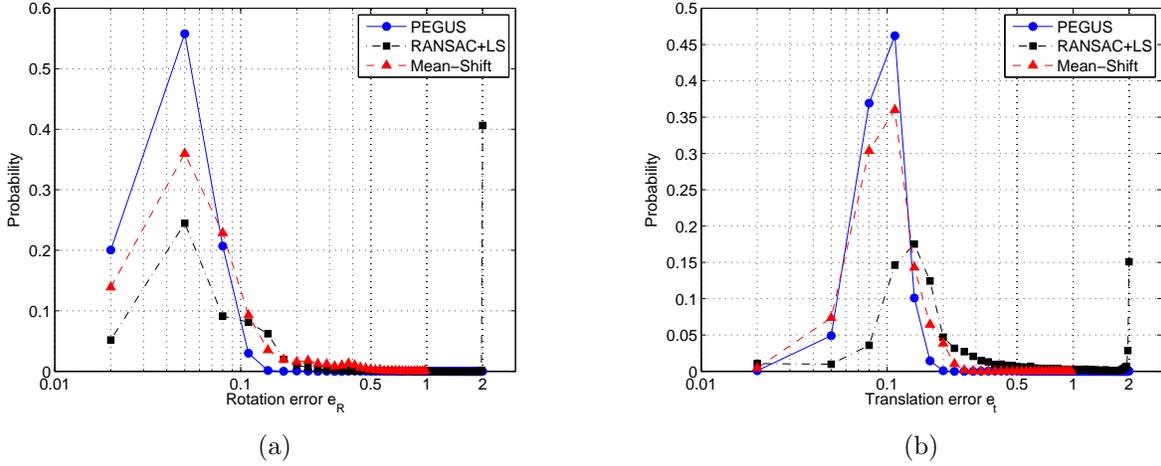


Figure 2-6. Evaluation of the estimation accuracy achieved by the proposed PEGUS algorithm and its comparison with that achieved by RANSAC+LS and non-linear mean shift algorithm of [1]. RANSAC+LS means outlier rejection by RANSAC followed by re-estimation of the pose by feeding all the inliers to the normalized 8-point algorithm. Pmf of the rotation and unit translation estimation errors are computed from 9000 samples of the error obtained from the 9000 image pairs.

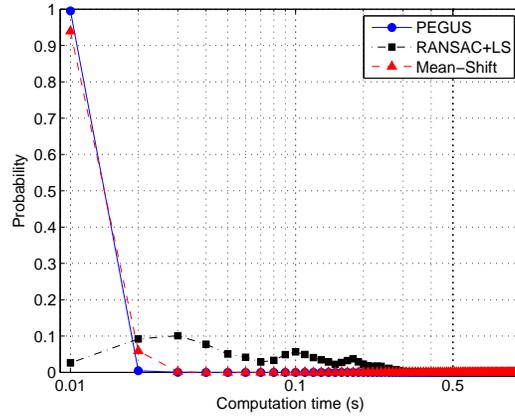


Figure 2-7. Comparison of the computation time required by the 3 algorithms. The pmf is estimated from 9000 samples of computation time.

2.6 Discussion

2.6.1 Sampling of the Hypotheses

The accuracy of the estimate obtained by PEGUS algorithm depends on the pose hypotheses that are generated in the beginning. In this section some of the properties of

Table 2-1. Comparison of mean and variance of estimation error: (A) PEGUS (B) RANSAC+least squares, (C) non-linear mean shift.

Algorithm	Rot. Error		Trans. Error	
	Mean	Std. Dev.	Mean	Std. Dev.
A	0.0533	0.0390	0.1010	0.0438
B	0.8726	0.9383	0.6177	0.7340
C	0.1016	0.1163	0.1092	0.0371

the hypotheses generation scheme used in the algorithm are described, and discuss why the scheme used leads to good estimates.

Note that since each distinct P pairs of point correspondence leads to a distinct hypotheses of q and t .

Let \mathbf{h} be the random variable representing the hypothesis that is obtained when the Simple Random Sampling With Replacement (SRSWR) scheme is executed. The possible values that \mathbf{h} can take are denoted by $h_i, i = 1, N_{\max}$. Each h_i is a pair q_i, t_i . Since there is a mapping from each set of P feature point pairs to hypotheses, (e.g., the 8-point algorithm).

Proposition 1. *The SRSWR scheme for hypotheses generation ensures that each possible hypothesis is obtained with equal probability, i.e., $P(\mathbf{h} = h_i) = \frac{1}{N_{\max}}$.*

Proof. A hypothesis \mathbf{h} is uniquely defined by the P point correspondence used to generate it, which are denoted by $\mathbf{f}^1, \mathbf{f}^2, \dots, \mathbf{f}^P$. Assume that the all feature point pairs are sorted to have increasing index from 1 through M .

$$\begin{aligned}
 P(\mathbf{h} = h_i) &= P(\mathbf{f}^1 = h_i^1, \mathbf{f}^2 = h_i^2, \dots, \mathbf{f}^P = h_i^P) \\
 &= \prod_{k=2}^8 P(\mathbf{f}^k = h_i^k | \mathbf{f}^{k-1} = h_i^{k-1}, \dots, \mathbf{f}^1 = h_i^1) \\
 &\quad \times P(\mathbf{f}^1 = h_i^1) \\
 &= \frac{1}{M-7} \frac{1}{M-6} \cdots \frac{1}{M}
 \end{aligned}$$

where the second equality follows from the chain rule of conditional probability, and the third equality follows from the fact that once the first k point correspondences are picked, the probability of picking a specific point among the remaining ones is $1/(M - k)$. \square

2.6.2 Robustness to Outliers

Robustness of the presented algorithm to feature outliers was analyzed using synthetic feature point data. A random cloud of 100 Euclidean points was generated and these Euclidean points were projected on the image plane using a pin-hole camera model. The Euclidean point cloud is viewed from two distinct camera positions with the known relative rotation and translation serving as a ground truth. In each trial, white Gaussian noise of 0.01 standard deviation was added to the image data. In order to verify the robustness of the three pose estimation algorithms (PEGUS, RANSAC+LS, and non-linear mean shift), the percentage of outliers was varied from 10% to 90% with an increment of 10% and the outliers were added randomly to the synthetic data. For each case, the experiment was repeated 100 times and pose estimates were obtained using PEGUS, RANSAC+LS, and non-linear mean shift algorithms. Using 2-10, the rotation and translation estimation errors were obtained as shown in Figs. 2-8 and 2-9. Fig. 2-10 shows the mean of the rotation and translation errors obtained for different number of feature outliers. It can be seen that the mean pose estimation error obtained using PEGUS is minimum among these algorithms.

2.7 Conclusion

In this chapter a novel robust two-view relative pose estimation algorithm is presented. Hypothesize-and-test methods such as RANSAC ignore all but one of the good hypotheses, whereas the proposed algorithm identifies a set of “low-noise” pose hypotheses among the large number of possible ones and then averages them appropriately to compute an estimate. Identification of the “low-noise” set of hypotheses is simplified by expressing rotations as unit-quaternions that lie on the 3-sphere \mathbb{S}^3 and constructing a histogram by gridding \mathbb{S}^3 . The same technique is used for unit-translations, except that

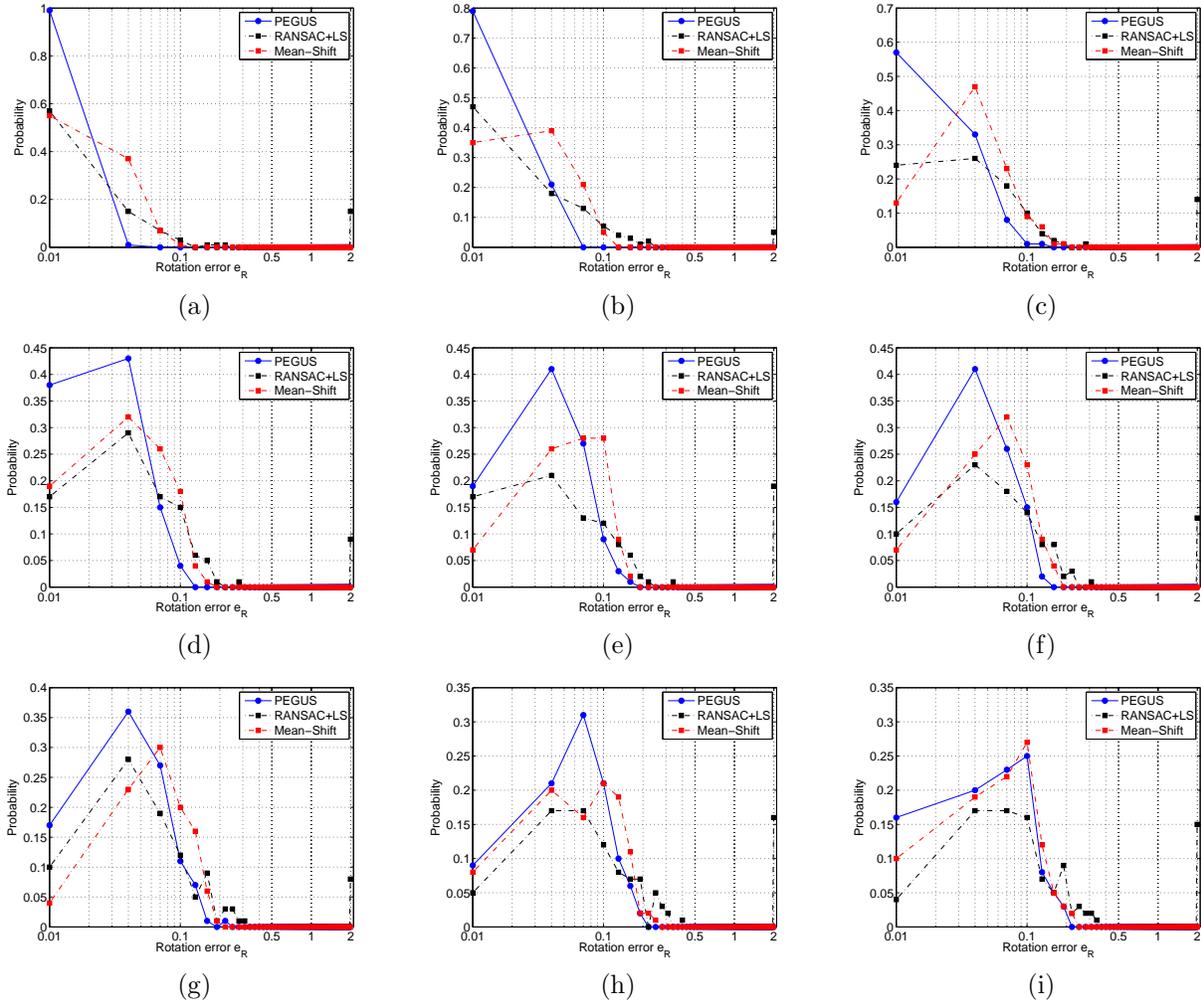


Figure 2-8. Robustness comparison of the presented PEGUS algorithm with RANSAC+LS and non-linear mean shift algorithm in terms of the rotation estimation accuracy using synthetic data with (a) 10% feature outliers, (b) 20% feature outliers, (c) 30% feature outliers, (d) 40% feature outliers, (e) 50% feature outliers, (f) 60% feature outliers, (g) 70% feature outliers, (h) 80% feature outliers, and (i) 90% feature outliers. Pmf of the rotation estimation error is computed from 100 samples of the error obtained from the 100 image pairs.

the hypotheses are now points on the unit sphere in 3-dimensions. Experimental results demonstrate improved performance of the proposed method against RANSAC+least squares method as well as non-linear mean shift, in terms of both estimation accuracy and computation time. Since the proposed method does not involve any iterative search, its computation time is more predictable than that of RANSAC and non-linear mean shift. Subsequent chapters will focus on development and analysis of visual servo control

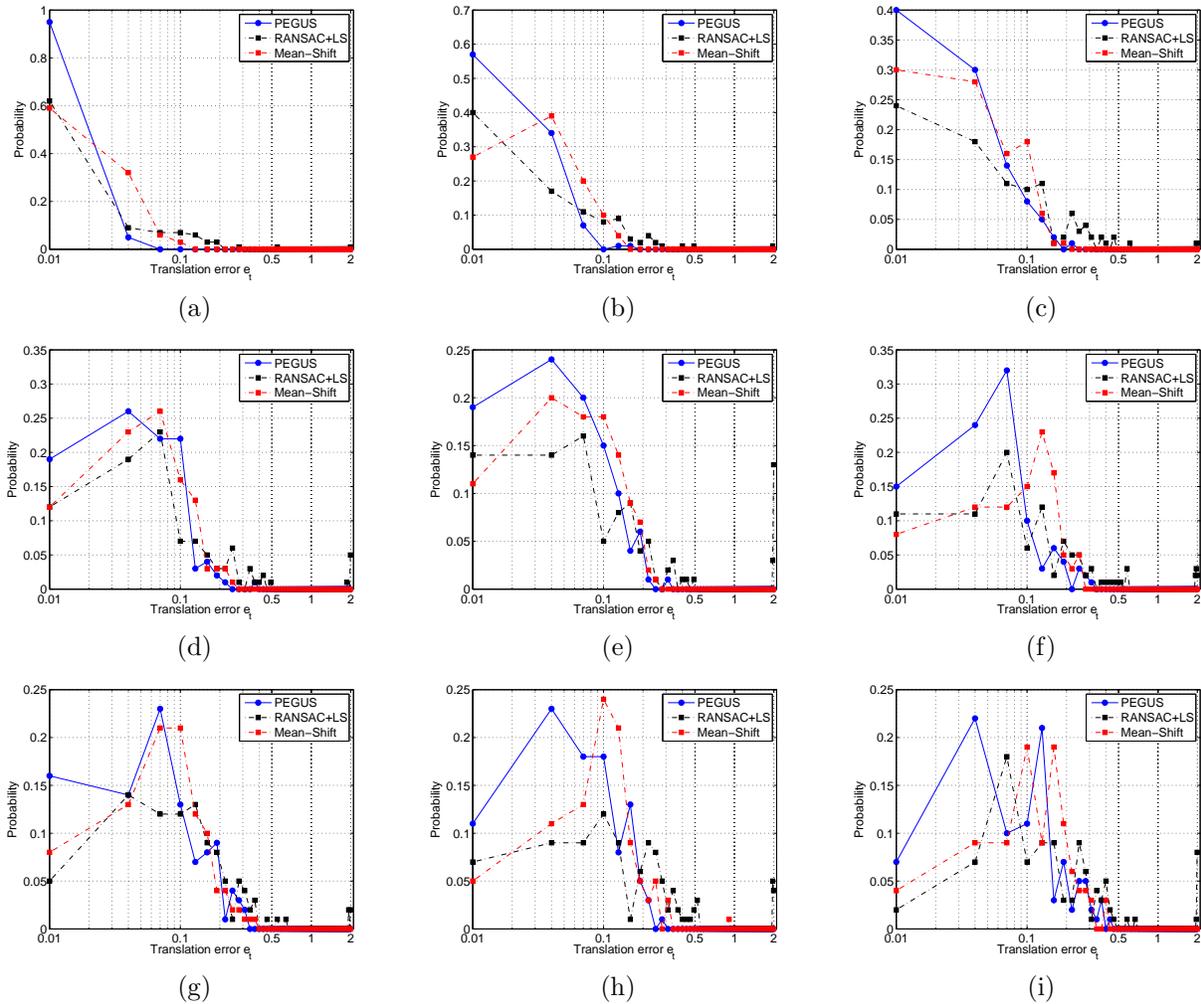
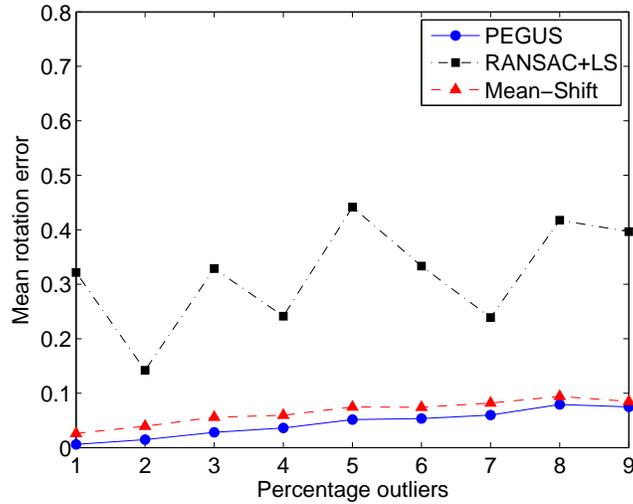


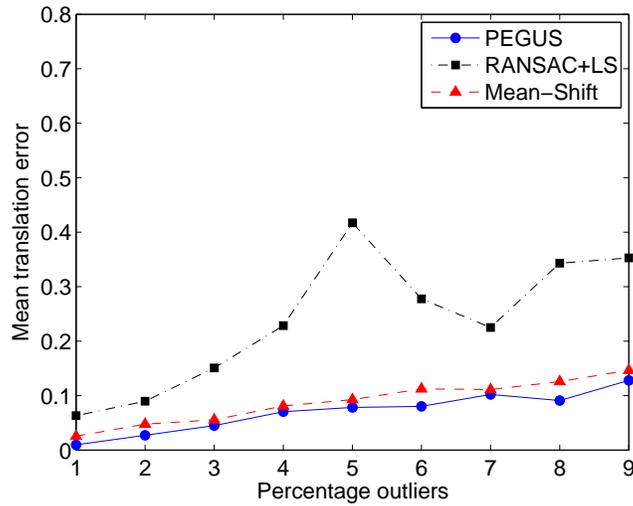
Figure 2-9. Robustness comparison of the presented PEGUS algorithm with RANSAC+LS and non-linear mean shift algorithm in terms of the translation estimation accuracy using synthetic data with (a) 10% feature outliers, (b) 20% feature outliers, (c) 30% feature outliers, (d) 40% feature outliers, (e) 50% feature outliers, (f) 60% feature outliers, (g) 70% feature outliers, (h) 80% feature outliers, and (i) 90% feature outliers. Pmf of the translation estimation error is computed from 100 samples of the error obtained from the 100 image pairs.

of autonomous systems wherein vision-based pose estimation acts as the foundation.

Therefore, the development provided in this chapter would be beneficial to such systems requiring robust pose estimation in a deterministic fashion.



(a)



(b)

Figure 2-10. (a) Mean rotation estimation error and (b) mean translation estimation error for the presented PEGUS algorithm, RANSAC+LS, and non-linear mean shift algorithm using the pose estimation results for synthetic data of varying feature outliers (10% - 90%) presented in Figs. 2-8 and 2-9.

CHAPTER 3
VISUAL SERVO CONTROL OF AN UNMANNED GROUND VEHICLE VIA A
MOVING AIRBORNE MONOCULAR CAMERA

3.1 Introduction

The development in this chapter is motivated by the desire to address the well-known problem of controlling a moving target using a moving camera. A moving airborne monocular camera (e.g., a camera attached to a remote controlled aircraft, a camera mounted on a satellite) is used to provide pose measurements of a moving sensorless UGV relative to a goal configuration. The relative velocity between the moving UGV and the moving camera presents a significant challenge. The contribution of this chapter is the development of multi-view geometry concepts (i.e., photogrammetry) to relate coordinate frames attached to the moving camera, moving UGV, and the desired UGV pose specified by an a priori image, the control scheme is coined as *daisy-chaining method*.

For the results in [42, 43], the pose measurements are taken with respect to a stationary reference object and restrictions are imposed on the area of operation motion for the UGV so that the stationary reference object never leaves the field-of-view of an on-board camera. Also, the method presented in [42, 43] assumes that the known Euclidean distance of the feature points on the UGV and the stationary reference object are identical, which imposes practical limitations on the implementation of the visual servo controller. The result in this chapter paper further develops the daisy-chaining method to achieve asymptotic regulation of the UGV based on the assumption that the given reference objects can leave the field of view while another reference object enters the field of view.

This chapter is organized in two parts; daisy-chaining method is developed in the first part with an objective of regulating an UGV to the desired position using image feedback from a moving monocular camera. The second part of chapter presents the development of a multi-reference visual servo control scheme in presence of the reseeding stationary

reference objects. Simulation results are provided to demonstrate the performance of the multi-reference daisy-chaining method for regulation control.

3.2 Daisy-Chaining Based Regulation Control

In this section, a collaborative visual servo controller is developed with the objective to regulate an UGV to a desired pose utilizing the feedback from a moving airborne monocular camera system. In contrast to typical camera configurations used for visual servo control problems, the controller in this paper is developed using a moving on-board camera viewing a moving target. Multi-view photogrammetric methods are used to develop relationships between difference camera frames and UGV coordinate systems. Geometric constructs developed for traditional camera-in-hand problems are fused with fixed-camera geometry to develop a set of Euclidean homographies. One of the resulting Euclidean homographies is not measurable through a set of spatiotemporal images (i.e., a corresponding projective homography can not be developed as in previous results). Hence, new geometric relationships are formulated to solve for the homography so that a measurable error system for the nonholonomic UGV can be developed. The resulting open-loop error system is expressed in a form that is amenable to a variety of UGV controllers. A benchmark controller originally proposed in [44] is proven to yield the asymptotic regulation result through a Lyapunov-based stability analysis.

3.2.1 Geometric Model

Consider a single camera that is navigating (e.g., by remote controlled aircraft) above¹ the planar motion of an UGV as depicted in Fig. 3-1 and Fig. 3-2. The moving coordinate frame \mathcal{I} is attached to the airborne camera and the moving coordinate frame \mathcal{F} is attached to the UGV at the center of the rear wheel axis (for simplicity and without

¹ No assumptions are made with regard to the alignment of the WMR plane of motion and the focal axis of the camera as in [26].

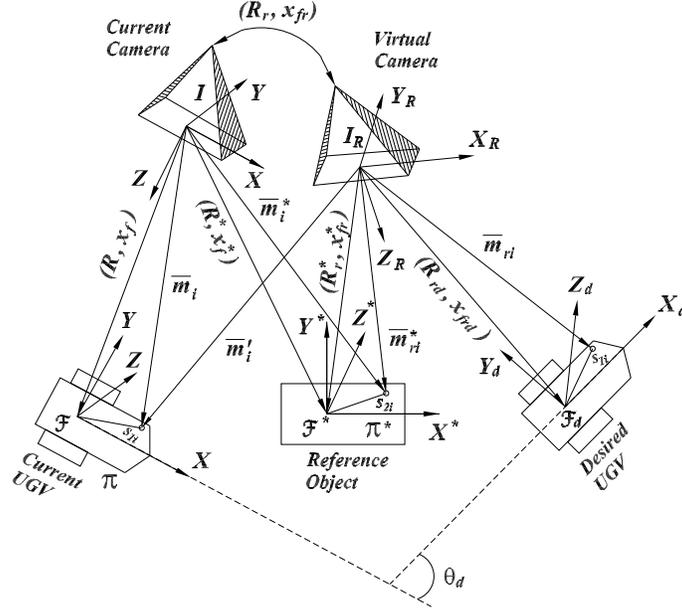


Figure 3-1. Camera coordinate frame relationships: A moving airborne monocular camera (coordinate frame \mathcal{I}) hovering above an UGV (coordinate frame \mathcal{F}) while viewing a fixed reference object (coordinate frame \mathcal{F}^*) regulates an UGV to the desired pose (coordinate frame \mathcal{F}_d) captured by a priori located camera (coordinate frame \mathcal{I}_R).

loss of generality). The UGV is represented in the camera image by four² feature points that are coplanar and not collinear. The Euclidean distance (i.e., $s_{1i} \in \mathbb{R}^3 \forall i = 1, 2, 3, 4$) from the origin of \mathcal{F} to one of the feature points is assumed to be known. The plane defined by the UGV motion (i.e., the plane defined by the xy -axis of \mathcal{F}) and the UGV feature points is denoted as π . The linear velocity of the UGV along the x -axis is denoted by $v_c(t) \in \mathbb{R}$, and the angular velocity $\omega_c(t) \in \mathbb{R}$ is about the z -axis of \mathcal{F} (see Figure 3-1).

While viewing the feature points of the UGV, the camera is assumed to also view four additional coplanar and noncollinear feature points of a stationary reference object. The

² Image analysis methods can be used to determine planar objects (e.g. through color, texture differences). These traditional computer vision methods can be used to help determine and isolate the four coplanar feature points. If four coplanar target points are not available then the subsequent development can exploit the classic eight-points algorithm [45] with no four of the eight target points being coplanar.

four additional feature points define the plane π^* in Fig. 3-1 and Fig. 3-2. The stationary coordinate frame \mathcal{F}^* is attached to the object where distance (i.e., $s_{2i} \in \mathbb{R}^3 \forall i = 1, 2, 3, 4$) from the origin of the coordinate frame to one of the feature points is assumed to be known. The plane π^* is assumed to be parallel to the plane π . The feature points that define π^* are also assumed to be visible when the camera is a priori located coincident with the position and orientation (i.e., pose) of the stationary coordinate frame \mathcal{I}_r . When the camera is coincident with \mathcal{I}_r , the desired pose of the UGV is assumed to be in the camera's field-of-view. When the UGV is located at the desired pose, the coordinate frame \mathcal{F} is coincident with the coordinate frame \mathcal{F}_d . Table 3-1 shows the relationships between various coordinate frames.

Table 3-1. Coordinate frames relationships for UGV regulation control.

Motion	Frames
$R(t), x_f(t)$	\mathcal{F} to \mathcal{I}
$R^*(t), x_f^*(t)$	\mathcal{F}^* to \mathcal{I}
$R_r(t), x_{fr}(t)$	\mathcal{I} to \mathcal{I}_R
R_r^*, x_{fr}^*	\mathcal{F}^* to \mathcal{I}_R
R_{rd}, x_{frd}	\mathcal{F}_d to \mathcal{I}_R
$R'(t), x'_{fr}(t)$	\mathcal{F} to \mathcal{I}_R in \mathcal{I}_R

To relate the coordinate systems, let $R(t), R^*(t), R_r(t), R_{rd}, R_r^* \in SO(3)$ denote the rotation from \mathcal{F} to \mathcal{I} , \mathcal{F}^* to \mathcal{I} , \mathcal{I} to \mathcal{I}_R , \mathcal{F}_d to \mathcal{I}_R , and \mathcal{F}^* to \mathcal{I}_R , respectively, $x_f(t), x_f^*(t) \in \mathbb{R}^3$ denote the respective time-varying translation from \mathcal{F} to \mathcal{I} and from \mathcal{F}^* to \mathcal{I} with coordinates expressed in \mathcal{I} , and $x_{fr}(t), x'_{fr}(t), x_{frd}, x_{fr}^* \in \mathbb{R}^3$ denote the respective translation from \mathcal{I} to \mathcal{I}_R , \mathcal{F} to \mathcal{I}_R , \mathcal{F}_d to \mathcal{I}_R , and from \mathcal{F}^* to \mathcal{I}_R expressed in the coordinates of \mathcal{I}_R . From the geometry between the coordinate frames depicted in Fig. 3-1 and Fig. 3-2, the following relationships can be developed

$$\bar{m}_i = x_f + R s_{1i} \quad \bar{m}_{rdi} = x_{frd} + R_{rd} s_{1i} \quad (3-1)$$

$$\bar{m}_i^* = x_f^* + R^* s_{2i} \quad \bar{m}_{ri}^* = x_{fr}^* + R_r^* s_{2i} \quad (3-2)$$

$$\bar{m}'_i(t) = x'_{fr} + R_r^* R^{*T} R s_{1i} \quad (3-3)$$

and $\bar{m}_{ri}^* \in \mathbb{R}^3$ denotes the constant Euclidean coordinates of the feature points on the plane π^* expressed in \mathcal{I}_R as

$$\bar{m}_{ri}^* \triangleq \begin{bmatrix} x_{ri}^* & y_{ri}^* & z_{ri}^* \end{bmatrix}^T. \quad (3-8)$$

After some algebraic manipulation, the expressions for $\bar{m}_i(t)$, \bar{m}_{rdi} , $\bar{m}_{ri}^*(t)$, and $\bar{m}'_i(t)$ in (3-1)-(3-3) can be rewritten as

$$\bar{m}_i = \bar{x}_f + \bar{R}\bar{m}_i^* \quad \bar{m}_{rdi} = \bar{x}_{f rd} + \bar{R}_{rd}\bar{m}_{ri}^* \quad (3-9)$$

$$\bar{m}_{ri}^* = x_{fr} + R_r\bar{m}_i^* \quad \bar{m}'_i(t) = x_{fr} + R_r\bar{m}_i \quad (3-10)$$

where $\bar{R}(t)$, \bar{R}_{rd} , $\bar{R}_r \in SO(3)$ and $\bar{x}_f(t)$, $\bar{x}_{f rd}$, $\bar{x}'_{fr}(t) \in \mathbb{R}^3$ are new rotational and translational variables, respectively, defined as

$$\bar{R} = RR^{*T} \quad \bar{R}_{rd} = R_{rd}R_r^{*T} \quad R_r = R_r^*R^{*T} \quad (3-11)$$

$$\bar{x}_f = x_f - \bar{R}(x_f^* + R^*(s_{2i} - s_{1i})) \quad (3-12)$$

$$\bar{x}_{f rd} = x_{f rd} - \bar{R}_{rd}(x_{fr}^* + R_r^*(s_{2i} - s_{1i})) \quad (3-13)$$

$$x_{fr} = x_{fr}^* - R_r x_f^* = x'_{fr} - R_r x_f. \quad (3-14)$$

By using the projective relationships (see Fig. 3-3 and Fig. 3-4)

$$d(t) = n^{*T}\bar{m}_i \quad d^*(t) = n^{*T}\bar{m}_i^* \quad d_r^* = n^{*T}\bar{m}_{ri}^* \quad (3-15)$$

the relationships in (3-9) and (3-10) can be expressed as

$$\bar{m}_i = \left(\bar{R} + \frac{\bar{x}_f n^{*T}}{d^*} \right) \bar{m}_i^* \quad (3-16)$$

$$\bar{m}_{rdi} = \left(\bar{R}_{rd} + \frac{\bar{x}_{f rd} n^{*T}}{d_r^*} \right) \bar{m}_{ri}^* \quad (3-17)$$

$$\bar{m}_{ri}^* = \left(R_r + \frac{x_{fr} n^{*T}}{d^*} \right) \bar{m}_i^* \quad (3-18)$$

$$\bar{m}'_i = \left(R_r + \frac{x_{fr} n^{*T}}{d} \right) \bar{m}_i. \quad (3-19)$$

In (3-15)-(3-19), $d(t)$, $d^*(t)$, $d_r^* > \varepsilon$ for some positive constant $\varepsilon \in \mathbb{R}$, and $n^* \in \mathbb{R}^3$ denotes the constant unit normal to the planes π and π^* .

Remark 1. *As in [46], the subsequent development requires that the constant rotation matrix R_r^* be known. The constant rotation matrix R_r^* can be obtained a priori using various methods (e.g., a second camera, Euclidean measurements).*

3.2.2 Euclidean Reconstruction

The relationships given by (3-16) and (3-19) provide a means to quantify a translation and rotation error between the different coordinate systems. Since the pose of \mathcal{F} , \mathcal{F}_d , and \mathcal{F}^* cannot be directly measured, a Euclidean reconstruction is developed in this section to obtain the position and rotational error information by comparing multiple images acquired from the hovering monocular vision system. Specifically, comparisons are made between the current UGV image and the reference image in terms of \mathcal{I} and between the a priori known UGV image and the reference image in terms of \mathcal{I}_R . To facilitate the subsequent development, the normalized Euclidean coordinates of the feature points for the current UGV image and the reference image can be expressed in terms of \mathcal{I} as $m_i(t)$ and $m_i^*(t) \in \mathbb{R}^3$, respectively, as follows:

$$m_i \triangleq \frac{\bar{m}_i}{z_i} \quad m_i^* \triangleq \frac{\bar{m}_i^*}{z_i^*}. \quad (3-20)$$

Similarly, the normalized Euclidean coordinates of the feature points for the current, goal, and reference image can be expressed in terms of \mathcal{I}_R as $m_i'(t)$, $m_{r di}$, $m_{r i}^* \in \mathbb{R}^3$, respectively, as follows:

$$m_i'(t) = \frac{\bar{m}_i'(t)}{z_i'(t)} \quad m_{r di} \triangleq \frac{\bar{m}_{r di}}{z_{r di}} \quad m_{r i}^* \triangleq \frac{\bar{m}_{r i}^*}{z_{r i}^*}. \quad (3-21)$$

From the expressions given in (3-16) and (3-20), the rotation and translation between the coordinate systems \mathcal{F} and \mathcal{F}^* can now be related in terms of the normalized Euclidean coordinates as follows:

$$m_i = \underbrace{\frac{z_i^*}{z_i}}_{\alpha_i} \underbrace{(\bar{R} + x_h n^{*T})}_{H} m_i^*. \quad (3-22)$$

In a similar manner, (3-17)-(3-21) can be used to relate the rotation and translation between m_{ri}^* and m_{rdi} as

$$m_{rdi} = \underbrace{\frac{z_{ri}^*}{z_{rdi}}}_{\alpha_{rdi}} \underbrace{\left(\bar{R}_{rd} + x_{hrd} n^{*T} \right)}_{H_{rd}} m_{ri}^* \quad (3-23)$$

and between $m_i^*(t)$ and m_{ri}^* as

$$m_{ri}^* = \underbrace{\frac{z_i^*}{z_{ri}^*}}_{\alpha_{ri}} \underbrace{\left(R_r + x_{hr} n^{*T} \right)}_{H_r} m_i^*. \quad (3-24)$$

The development provided in the Remark 2 can be used to relate $m_i(t)$ to $m_i'(t)$ as

$$m_i' = \frac{z_i}{z_i'} \underbrace{\left(R_r + x_{hr} \alpha_i \frac{n^{*T} m_i^*}{n^{*T} m_i} \right)}_{H_r'} m_i. \quad (3-25)$$

In (3-22)-(3-25), $\alpha_i(t)$, α_{rdi} , $\alpha_{ri}(t) \in \mathbb{R}$ denote depth ratios, $H(t)$, H_{rd} , $H_r(t)$, $H_r'(t) \in \mathbb{R}^{3 \times 3}$ denote Euclidean homographies [47], and $x_h(t)$, x_{hrd} , $x_{hr}(t) \in \mathbb{R}^3$ denote scaled translation vectors that are defined as follows

$$x_h = \frac{\bar{x}_f}{d^*} \quad x_{hrd} = \frac{\bar{x}_{frd}}{d_r^*} \quad x_{hr} = \frac{x_{fr}}{d^*} \quad (3-26)$$

Remark 2. In order to find the relationship between the normalized Euclidean coordinates $m_i(t)$ and $m_i'(t)$, termed as virtual homography, (3-19) is expressed as

$$\bar{m}_i' = \left(R_r + \frac{x_{fr} n^{*T} d^*}{d^*} \frac{d^*}{d} \right) \bar{m}_i. \quad (3-27)$$

By substituting $x_{fr}(t)$ from (3-26) into (3-27) the following expression can be obtained

$$\bar{m}_i' = \left(R_r + x_{hr} \frac{d^*}{d} n^{*T} \right) \bar{m}_i. \quad (3-28)$$

Utilizing (3-15), (3-20), and (3-22) and rearranging the terms yields

$$\bar{m}'_i = \left(R_r + x_{hr} \alpha_i \frac{n^{*T} m_i^*}{n^{*T} m_i} n^{*T} \right) \bar{m}_i. \quad (3-29)$$

The above expression can be written in terms of the normalized Euclidean coordinates by using (3-20)-(3-21) as follows

$$m'_i = \frac{z_i}{z'_i} \left(R_r + x_{hr} \alpha_i \frac{n^{*T} m_i^*}{n^{*T} m_i} n^{*T} \right) m_i. \quad (3-30)$$

Each Euclidean feature point will have a projected pixel coordinate expressed in terms of \mathcal{I} as

$$p_i \triangleq \begin{bmatrix} u_i & v_i & 1 \end{bmatrix}^T \quad p_i^* \triangleq \begin{bmatrix} u_i^* & v_i^* & 1 \end{bmatrix}^T \quad (3-31)$$

where $p_i(t)$ and $p_i^*(t) \in \mathbb{R}^3$ represents the image-space coordinates of the time-varying feature points of the UGV and reference object, respectively, and $u_i(t), v_i(t), u_i^*(t), v_i^*(t) \in \mathbb{R}$. Similarly, the projected pixel coordinate of the Euclidean features in the reference image can be expressed in terms of \mathcal{I}_R as

$$p_{rdi} \triangleq \begin{bmatrix} u_{rdi} & v_{rdi} & 1 \end{bmatrix}^T \quad p_{ri}^* \triangleq \begin{bmatrix} u_{ri}^* & v_{ri}^* & 1 \end{bmatrix}^T \quad (3-32)$$

where p_{rdi} and $p_{ri}^* \in \mathbb{R}^3$ represents the constant image-space coordinates of the goal UGV and the reference object, respectively, and $u_{rdi}, v_{rdi}, u_{ri}^*, v_{ri}^* \in \mathbb{R}$. To calculate the Euclidean homographies given in (3-22)-(3-25) from pixel information, the projected pixel coordinates are related to $m_i(t), m_i^*(t), m_{rdi}$ and m_{ri}^* by the pin-hole camera model as

$$p_i = A m_i \quad p_i^* = A m_i^* \quad (3-33)$$

$$p_{rdi} = A m_{rdi} \quad p_{ri}^* = A m_{ri}^* \quad (3-34)$$

where $A \in \mathbb{R}^{3 \times 3}$ is a known, constant, and invertible intrinsic camera calibration matrix. By using (3-22)-(3-25), (3-33), and (3-34), the following relationships can be developed:

$$p_i = \alpha_i \underbrace{(AHA^{-1})}_{G} p_i^* \quad p_{rdi} = \alpha_{rdi} \underbrace{(AH_{rd}A^{-1})}_{G_{rd}} p_{ri}^* \quad (3-35)$$

$$p_{ri}^* = \alpha_{ri} \underbrace{(AH_rA^{-1})}_{G_r} p_i^* \quad (3-36)$$

where $G(t) = [g_{ij}(t)]$, $G_{rd} = [g_{rdij}]$, $G_r = [g_{rij}] \quad \forall i, j = 1, 2, 3 \in \mathbb{R}^{3 \times 3}$ denote projective homographies.

Sets of linear equations can be developed from (3-35) and (3-36) to determine the projective homographies up to a scalar multiple. Various techniques can be used (e.g., see [48, 49]) to decompose the Euclidean homographies, to obtain $\alpha_i(t)$, α_{rdi} , $\alpha_{ri}(t)$, $x_h(t)$, x_{hrd} , $x_{hr}(t)$, $\bar{R}(t)$, \bar{R}_{rd} , $R_r(t)$. Given that the constant rotation matrix R_r^* is assumed to be known, the expressions for \bar{R}_{rd} and $R_r(t)$ in (3-11) can be used to determine R_{rd} and $R^*(t)$. Once $R^*(t)$ is determined, the expression for $\bar{R}(t)$ in (3-11) can be used to determine $R(t)$.

3.2.3 UGV Kinematics

The kinematic model for the UGV can be determined from Fig. 3-1 as

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_d \end{bmatrix} = \begin{bmatrix} \cos \theta_d & 0 \\ \sin \theta_d & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} \quad (3-37)$$

where \dot{x}_c , \dot{y}_c , and $\dot{\theta}_d$ denote the time derivatives of $x_c(t)$, $y_c(t)$, and $\theta_d(t) \in \mathbb{R}$, respectively, where $x_c(t)$ and $y_c(t)$ denote the planar position of \mathcal{F} expressed in \mathcal{F}_d , and $\theta_d(t) \in \mathbb{R}$ denotes the right-handed rotation angle about the z-axis of \mathcal{F} that aligns \mathcal{F} with \mathcal{F}_d , and $v_c(t)$ and $\omega_c(t)$ were introduced in Section 3.2.1 and are depicted in Fig. 3-1 and Fig. 3-2. Based on the definitions for $R(t)$, $R^*(t)$, R_{rd} , R_r^* , and $R_r(t)$ provided in the previous

development (see also Table 3-1), the rotation from \mathcal{F} to \mathcal{F}_d can be developed as

$$R_{rd}^T R_r^* R^{*T} R = \begin{bmatrix} \cos \theta_d & -\sin \theta_d & 0 \\ \sin \theta_d & \cos \theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3-38)$$

Based on the fact that $R(t)$, $R^*(t)$, R_r^* , and R_{rd} are known from the homography decomposition, it is clear from (3-38) that $\theta_d(t)$ is a known signal that can be used in the subsequent control development.

The geometric relationships between the coordinate frames can be used to develop the following expression

$$\begin{bmatrix} x_c & y_c & 0 \end{bmatrix}^T = R_{rd}^T (x'_{fr} - x_{frd}). \quad (3-39)$$

After utilizing, (3-1), (3-3), (3-25), and the assumption (as in [46]) that $s_{11} = [0, 0, 0]^T$, the following expression can be obtained³

$$\begin{bmatrix} \frac{x_c}{z_{rd1}} & \frac{y_c}{z_{rd1}} & 0 \end{bmatrix}^T = R_{rd}^T \left(\frac{\alpha_{r1} \alpha_{rd1}}{\alpha_1} H'_r m_1 - m_{rd1} \right). \quad (3-40)$$

After utilizing (3-33) and (3-34), the expression in (3-40) can be rewritten as follows

$$\begin{bmatrix} \frac{x_c}{z_{rd1}} & \frac{y_c}{z_{rd1}} & 0 \end{bmatrix}^T = R_{rd}^T A^{-1} \left(\frac{\alpha_{r1} \alpha_{rd1}}{\alpha_1} H'_r p_1 - p_{rd1} \right). \quad (3-41)$$

Since all terms on the right-hand side of (3-41) are measurable or known, then $\frac{x_c(t)}{z_{rd1}}$ and $\frac{y_c(t)}{z_{rd1}}$ can be used in the subsequent control development.

3.2.4 Control Objective

The objective considered in this chapter is to develop a visual servo controller that ensures that the pose of a UGV is regulated to a desired pose. A challenging aspect of this problem is that the UGV pose information is supplied by a moving airborne monocular

³ Any point s_{1i} , s_{2i} can be utilized in the subsequent development; however, to reduce the notational complexity, we have elected to select the image point s_{11} , s_{21} , and hence, the subscript 1 is utilized in lieu of i in the subsequent development.

camera system. That is, unlike traditional camera-in-hand configurations or fixed camera configurations, the problem considered in this chapter involves a moving airborne camera observing a moving ground vehicle. Mathematically, the objective can be expressed as the desire to regulate $\bar{m}_i(t)$ to \bar{m}_{di} (or stated otherwise for $x'_{fr}(t) \rightarrow x_{frd}$ and $\theta_d(t) \rightarrow 0$). Based on (3-37)-(3-39) the objective can be quantified by a regulation error $e(t) \in \mathbb{R}^3$, defined by the following global diffeomorphism

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \triangleq \begin{bmatrix} \cos \theta_d & \sin \theta_d & 0 \\ -\sin \theta_d & \cos \theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{x_c}{z_{rd1}} \\ \frac{y_c}{z_{rd1}} \\ \theta_d \end{bmatrix}. \quad (3-42)$$

If $\|e(t)\| \rightarrow 0$, then (3-39) and (3-42) can be used to conclude that $x'_{fr}(t) \rightarrow x_{frd}$ and $\theta_d(t) \rightarrow 0$. Based on (3-38) and (3-41), it is clear that $e(t)$ is measurable.

3.2.5 Control Development

After taking the time derivative of (3-42) and using (3-37), the open-loop error system for $e(t)$ can be determined as

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \frac{v_c}{z_{rd1}} + \omega_c e_2 \\ -\omega_c e_1 \\ \omega_c \end{bmatrix}. \quad (3-43)$$

A variety of controllers could now be proposed to yield the regulation result based on the manner in which the open-loop error system given by (3-43) has been developed. Several controllers are provided in [50] including an explanation of how the UGV dynamics could also be easily incorporated into the control design. The following benchmark controller proposed in [44] is an example that can be used to achieve asymptotic regulation:

$$v_c \triangleq -k_v e_1 \quad (3-44)$$

$$\omega_c \triangleq -k_\omega e_3 + e_2^2 \sin t \quad (3-45)$$

where $k_v, k_\omega \in \mathbb{R}$ denote positive, constant control gains. After substituting the controller designed in (3-44) and (3-45) into (3-43), the following closed-loop error system is obtained:

$$\begin{aligned} z_{rd1}\dot{e}_1 &= -k_v e_1 + z_{rd1}\omega_c e_2 \\ \dot{e}_2 &= -\omega_c e_1 \\ \dot{e}_3 &= -k_\omega e_3 + e_2^2 \sin t . \end{aligned} \tag{3-46}$$

Remark 3. As stated in [44] and [50], the closed-loop dynamics for $e_3(t)$ given in (3-46) represent a stable linear system subject to an additive disturbance given by the product $e_2^2(t) \sin(t)$. If the additive disturbance is bounded (i.e., if $e_2(t) \in \mathcal{L}_\infty$), then it is clear that $e_3(t) \in \mathcal{L}_\infty$. Furthermore, if the additive disturbance asymptotically vanishes (i.e., if $e_2(t) \rightarrow 0$) then it is clear from standard linear control arguments [51] that $e_3(t) \rightarrow 0$.

3.2.6 Stability Analysis

Theorem 1. The kinematic controller given in (3-44) and (3-45) ensure asymptotic pose regulation of the UGV in the sense that

$$\lim_{t \rightarrow \infty} \|e(t)\| = 0 \tag{3-47}$$

Proof. Let $V(t) \in \mathbb{R}$ denote the following non-negative function:

$$V \triangleq \frac{1}{2} z_{rd1} (e_1^2 + e_2^2) . \tag{3-48}$$

The following simplified expression can be obtained by taking the time derivative of (3-48), substituting the closed-loop dynamics from (3-46) into the resulting expression, and then cancelling common terms

$$\dot{V} = -k_v e_1^2 . \tag{3-49}$$

Based on (3-48) and (3-49), it is clear that $e_1(t), e_2(t) \in \mathcal{L}_\infty$ and that $e_1(t) \in \mathcal{L}_2$. Remark 3 can be used along with the fact that $e_2(t) \in \mathcal{L}_\infty$ to conclude that $e_3(t) \in \mathcal{L}_\infty$. Based on the fact that $e_1(t), e_2(t), e_3(t) \in \mathcal{L}_\infty$, (3-44), (3-45) and (3-46) can be used to prove

that $v_c(t), \omega_c(t), \dot{e}_1(t), \dot{e}_2(t), \dot{e}_3(t) \in \mathcal{L}_\infty$. The fact that $\dot{e}_1(t), \dot{e}_2(t), \dot{e}_3(t) \in \mathcal{L}_\infty$ is a sufficient condition for $e_1(t), e_2(t), e_3(t)$ to be uniformly continuous. After taking the time derivative of (3-44) and (3-45) and utilizing the aforementioned facts, we can show that $\dot{v}_c(t), \dot{\omega}_c(t) \in \mathcal{L}_\infty$, and hence, $v_c(t)$ and $\omega_c(t)$ are uniformly continuous. Based on the facts that $e_1(t), \dot{e}_1(t) \in \mathcal{L}_\infty$ and $e_1(t) \in \mathcal{L}_2$, Barbalat's Lemma can be used to prove that

$$\lim_{t \rightarrow \infty} e_1(t) = 0. \quad (3-50)$$

After taking the time derivative of the product $e_1(t)e_2(t)$ and then substituting (3-46) into the resulting expression for the time derivative of $e_1(t)$, the following expression is obtained

$$\frac{d}{dt}(e_1 e_2) = [e_2^2 \omega_c] + e_1(\dot{e}_2 - \frac{k_v e_2}{z_{rd1}}). \quad (3-51)$$

Given the facts that $\lim_{t \rightarrow \infty} e_1(t) = 0$ and the bracketed term in (3-51) is uniformly continuous (i.e., $e_2(t)$ and $\omega_c(t)$ are uniformly continuous), we can invoke extended Barbalat's lemma [50] to conclude that

$$\lim_{t \rightarrow \infty} \frac{d}{dt}(e_1(t)e_2(t)) = 0 \quad \lim_{t \rightarrow \infty} e_2^2(t)\omega_c(t) = 0. \quad (3-52)$$

After utilizing (3-44), (3-46), (3-50), and (3-52), we can conclude that

$$\lim_{t \rightarrow \infty} v_c(t) = 0 \quad \lim_{t \rightarrow \infty} \dot{e}_1(t) = 0 \quad \lim_{t \rightarrow \infty} \dot{e}_2(t) = 0 \quad (3-53)$$

To facilitate further analysis, we take the time derivative of the product $e_2(t)\omega_c(t)$ and utilize (3-43), (3-44) and (3-45) to obtain the following expression

$$\frac{d}{dt}(e_2 \omega_c) = [e_2^3 \cos(t)] + \dot{e}_2(\omega_c + 2e_2^2 \sin(t)) - k_\omega e_2 \omega_c. \quad (3-54)$$

Since the bracketed term in (3-54) is uniformly continuous, we can invoke extended Barbalat's lemma to conclude that

$$e_2^3(t) \cos(t) = 0; \quad (3-55)$$

hence,

$$\lim_{t \rightarrow \infty} e_2(t) = 0. \quad (3-56)$$

Based on (3-56), it is clear from Remark 3 that

$$\lim_{t \rightarrow \infty} e_3(t) = 0. \quad (3-57)$$

After utilizing (3-50), (3-56), and (3-57), the asymptotic regulation result given in (3-47) is obtained. \square

3.3 Multi-Reference Visual Servo Control of an Unmanned Ground Vehicle

The result in this section further develops the daisy chaining method introduced in Section 3.2 to achieve asymptotic regulation of the UGV based on the assumption that the given reference objects can leave the field of view while another reference object enters the field of view. The contribution of this research is that since the controller development is based on the ability to daisy chain multiple reference objects, the restrictions on the applicative area of operation are removed. That is, since the control development does not require the airborne camera to maintain a view of a static reference object, the airborne camera/UGV pair are able to navigate over an arbitrarily large area. The presented work also relaxes the assumption of having identical Euclidean distance of the features for the UGV and the reference object by leveraging on the geometric reconstruction method proposed by Dupree et al.[52].

3.3.1 Geometric Model

The geometric model for multi-reference regulation control is similar to that presented in Section 3.2. While viewing the feature points of the UGV, the camera is assumed to also view four additional coplanar and non-collinear feature points of a stationary reference object, such that at any instant of time along the camera motion trajectory at least one such reference target is in the field of view. The four additional feature points define the plane π_n^* in Fig. 3-3 and Fig. 3-4. The stationary coordinate frame \mathcal{F}_n^* ($n = 1, 2, \dots, m$) is attached to the object where distance from the origin of the coordinate

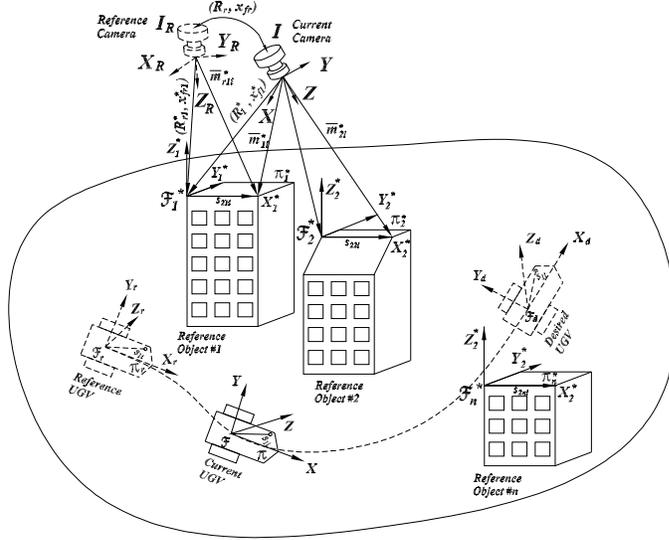


Figure 3-3. Camera to reference object relationships: A monocular camera (coordinate frame \mathcal{I}) viewing a stationary reference object (coordinate frame \mathcal{F}_i^*) such that a stationary object can leave the camera FOV as the new object enters the FOV, while the stationary reference camera (coordinate frame \mathcal{I}_R) is assumed to view the stationary reference object \mathcal{F}_1^* .

frame to one of the feature points is assumed to be known, i.e., $s_{2ni} \in \mathbb{R}^3 \forall i = 1, 2, 3, 4$. The plane π_n^* is assumed to be parallel to the plane π . The feature points that define π_1^* , corresponding to a reference object \mathcal{F}_1^* (i.e. \mathcal{F}_n^* corresponding to $n = 1$), are also assumed to be visible when the camera is a priori located coincident with the position and orientation (i.e., pose) of the stationary coordinate frame \mathcal{I}_R . The stationary pose \mathcal{F}_r corresponds to a snapshot of the UGV (e.g. at the starting location) visible from the reference camera coordinate system \mathcal{I}_R . When the camera is coincident with \mathcal{I}_R , the desired pose of the UGV \mathcal{F}_d is assumed to be known. When the UGV is located at the desired pose, the coordinate frame \mathcal{F} is coincident with the coordinate frame \mathcal{F}_d .

To relate the coordinate systems, consider the coordinate frame relationships as given in Table 3-2.

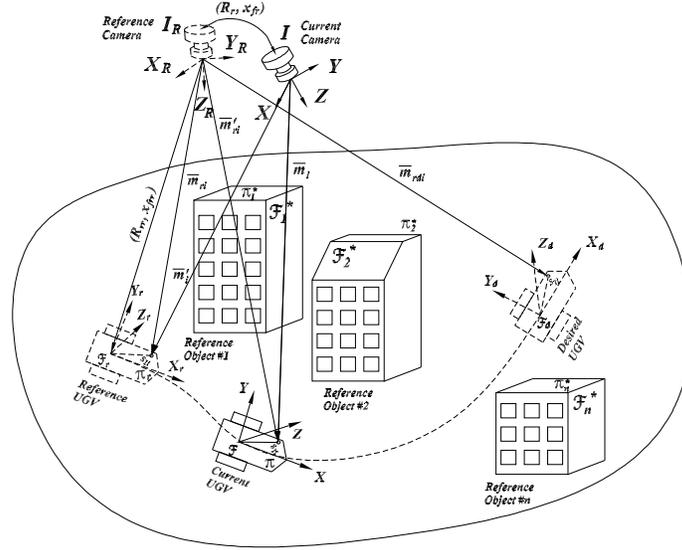


Figure 3-4. Camera to UGV relationships: A monocular camera (coordinate frame \mathcal{I}) hovering above an UGV (coordinate frame \mathcal{F}) while viewing a stationary reference object (coordinate frame \mathcal{F}_n^*) regulates an UGV to the desired pose (coordinate frame \mathcal{F}_d) known a priori in the reference camera (coordinate frame \mathcal{I}_R). The stationary pose (coordinate frame \mathcal{F}_r) corresponds to a snapshot of the UGV visible from the reference camera (coordinate frame \mathcal{I}_R).

Table 3-2. Coordinate frames relationships for multi-reference UGV regulation control.

Motion	Frames
$R(t), x_f(t)$	\mathcal{F} to \mathcal{I}
$R_n^*(t), x_{fn}^*(t)$	\mathcal{F}_n^* to \mathcal{I}
$R_r(t), x_{fr}(t)$	\mathcal{I} to \mathcal{I}_R
R_{rd}, x_{frd}	\mathcal{F}_d to \mathcal{I}_R
R_{rn}^*, x_{frn}^*	\mathcal{F}_n^* to \mathcal{I}_R
R_{rr}, x_{frr}	\mathcal{F}_r to \mathcal{I}_R
$R''(t), x_f''(t)$	\mathcal{F}_r to \mathcal{I}
$R'_r(t), x'_{fr}(t)$	\mathcal{F} to \mathcal{I}_R
$R'(t), x'_f(t)$	\mathcal{F} to \mathcal{F}_r
$R'_{rd}(t), x'_{frd}(t)$	\mathcal{F} to \mathcal{F}_d

From the geometry between the coordinate frames depicted in Fig. 3-3 and Fig. 3-4, the following relationships can be developed

$$\bar{m}_i = x_f + R s_{1i} \quad \bar{m}_{rdi} = x_{frd} + R_{rd} s_{1i} \quad (3-58)$$

$$\bar{m}_{ni}^* = x_{fn}^* + R_n^* s_{2ni} \quad \bar{m}_{rni}^* = x_{frn}^* + R_{rn}^* s_{2ni} \quad (3-59)$$

$$\bar{m}'_i = x_f'' + R'' s_{1i} \quad \bar{m}'_{ri} = x'_{fr} + R'_r s_{1i} \quad (3-60)$$

$$\bar{m}_{ri} = x_{frr} + R_{rr} s_{1i}. \quad (3-61)$$

In (3-58)-(3-61), $\bar{m}_i(t)$, $\bar{m}'_i(t)$, $\bar{m}_{ni}^*(t) \in \mathbb{R}^3$ denote the Euclidean coordinates of the feature points of the current UGV (i.e. \mathcal{F}), constant reference UGV position, and stationary reference object π_n^* ($n = 1, 2, \dots, m$), respectively, expressed in \mathcal{I} as

$$\bar{m}_i(t) = \begin{bmatrix} x_i(t) & y_i(t) & z_i(t) \end{bmatrix}^T \quad (3-62)$$

$$\bar{m}'_i(t) = \begin{bmatrix} x'_i(t) & y'_i(t) & z'_i(t) \end{bmatrix}^T \quad (3-63)$$

$$\bar{m}_{ni}^*(t) = \begin{bmatrix} x_{ni}^*(t) & y_{ni}^*(t) & z_{ni}^*(t) \end{bmatrix}^T, \quad (3-64)$$

\bar{m}_{ri} , $\bar{m}'_{ri}(t)$, $\bar{m}_{rdi} \in \mathbb{R}^3$ denote the Euclidean coordinates of the constant reference UGV, actual time varying current UGV, and constant desired UGV, respectively, expressed in \mathcal{I}_R as

$$\bar{m}_{ri} = \begin{bmatrix} x_{ri} & y_{ri} & z_{ri} \end{bmatrix}^T \quad (3-65)$$

$$\bar{m}'_{ri} = \begin{bmatrix} x'_{ri}(t) & y'_{ri}(t) & z'_{ri}(t) \end{bmatrix}^T \quad (3-66)$$

$$\bar{m}_{rdi} = \begin{bmatrix} x_{rdi} & y_{rdi} & z_{rdi} \end{bmatrix}^T, \quad (3-67)$$

and $\bar{m}_{rni}^* \in \mathbb{R}^3$ denotes the constant Euclidean coordinates of the feature points on the stationary reference plane π_n^* expressed in \mathcal{I}_R as

$$\bar{m}_{rni}^* = \begin{bmatrix} x_{rni}^* & y_{rni}^* & z_{rni}^* \end{bmatrix}^T. \quad (3-68)$$

For simplicity and without loss of generality, we consider two reference targets \mathcal{F}_n^* (where $n = 1, 2$). After some algebraic manipulation, the expressions for \bar{m}_{rni}^* , $\bar{m}'_{ri}(t)$, and $\bar{m}'_i(t)$ in (3-58)-(3-61) can be rewritten as

$$\bar{m}_{r1i}^* = x_{fr} + R_r \bar{m}_{1i}^* \quad \bar{m}_{r2i}^* = x_{fr} + R_r \bar{m}_{2i}^* \quad (3-69)$$

$$\bar{m}'_{ri} = x_{fr} + R_r \bar{m}_i \quad \bar{m}_{ri} = x_{fr} + R_r \bar{m}'_i \quad (3-70)$$

$$\bar{m}'_i = x'_f + R' \bar{m}_i \quad \bar{m}_{rdi} = x'_{frd} + R'_{rd} \bar{m}'_{ri} \quad (3-71)$$

where $R_r(t)$, $R'(t)$, $R'_{rd}(t) \in \mathbb{R}^{3 \times 3}$ and $x_{fr}(t)$, $x'_f(t)$, $x'_{frd}(t) \in \mathbb{R}^3$ denote new rotation and translation variables given as follows:

$$R_r = R_{rn}^* R_n^{*T} \quad x_{fr} = x_{frn}^* - R_r x_{fn}^* \quad (3-72)$$

$$R' = R'' R^T \quad x'_f = x''_f - R' x_f \quad (3-73)$$

$$R'_{rd} = R_{rd} R_r'^T \quad x'_{frd} = x_{frd} - R'_{rd} x'_{fr} \quad (3-74)$$

By using the projective relationships

$$d_1^* = n_1^{*T} \bar{m}_{1i}^* \quad d_2^* = n_2^{*T} \bar{m}_{2i}^* \quad (3-75)$$

$$d = n^T \bar{m}_i \quad d'_r = n_r'^T \bar{m}'_{ri} \quad (3-76)$$

the relationships in (3-69) and (3-71) can be expressed as

$$\bar{m}_{r1i}^* = \left(R_r + \frac{x_{fr} n_1^{*T}}{d_1^*} \right) \bar{m}_{1i}^* \quad (3-77)$$

$$\bar{m}_{r2i}^* = \left(R_r + \frac{x_{fr} n_2^{*T}}{d_2^*} \right) \bar{m}_{2i}^* \quad (3-78)$$

$$\bar{m}'_i = \left(R' + \frac{x'_f n^T}{d} \right) \bar{m}_i \quad (3-79)$$

$$\bar{m}_{rdi} = \left(R'_{rd} + \frac{x'_{frd} n_r'^T}{d'_r} \right) \bar{m}'_{ri}. \quad (3-80)$$

In (3-77)-(3-80), $d_1^*(t)$, $d_2^*(t)$, $d(t)$, $d'_r(t) > \varepsilon$ for some positive constant $\varepsilon \in \mathbb{R}$, and $n_1^*(t)$, $n_2^*(t)$, $n(t)$, $n'_r(t) \in \mathbb{R}^3$ denote the time varying unit normal to the planes π_1^* , π_2^* , and π , respectively.

3.3.2 Euclidean Reconstruction

The relationships given by (3-77)-(3-80) provide a means to quantify a translation and rotation error between the different coordinate systems. Comparisons are made between the current UGV image and the reference image in terms of \mathcal{I} , between the a priori known desired UGV pose and the current pose in terms of \mathcal{I}_R and between the images of the stationary reference object in terms of \mathcal{I} and \mathcal{I}_R . To facilitate the subsequent development, the normalized Euclidean coordinates of the feature points for the current UGV image, the reference UGV image, and the reference object images can be expressed in terms of \mathcal{I} as $m_i(t)$, $m_{1i}^*(t)$, and $m_{2i}^*(t) \in \mathbb{R}^3$, respectively, as

$$m_i = \frac{\bar{m}_i}{z_i} \quad m'_i = \frac{\bar{m}'_i}{z'_i} \quad (3-81)$$

$$m_{1i}^* = \frac{\bar{m}_{1i}^*}{z_{1i}^*} \quad m_{2i}^* = \frac{\bar{m}_{2i}^*}{z_{2i}^*}. \quad (3-82)$$

Similarly, the normalized Euclidean coordinates of the feature points for the current UGV, goal UGV, and reference object image can be expressed in terms of \mathcal{I}_R as $m'_{ri}(t)$, $m_{r di}$, m_{r1i}^* , and $m_{r2i}^* \in \mathbb{R}^3$, respectively, as

$$m'_{ri} = \frac{\bar{m}'_{ri}}{z'_{ri}} \quad m_{r di} = \frac{\bar{m}_{r di}}{z_{r di}} \quad (3-83)$$

$$m_{r1i}^* = \frac{\bar{m}_{r1i}^*}{z_{r1i}^*} \quad m_{r2i}^* = \frac{\bar{m}_{r2i}^*}{z_{r2i}^*}. \quad (3-84)$$

From the expressions given in (3-77), (3-82), and (3-84) the rotation and translation between the coordinate systems \mathcal{I} and \mathcal{I}_R can now be related in terms of the normalized Euclidean coordinates of the reference object \mathcal{F}_1^* as [42]

$$m_{r1i}^* = \underbrace{\frac{z_{1i}^*}{z_{r1i}^*}}_{\alpha_{ri}} \underbrace{\left(R_r + x_{hr} n_1^{*T} \right)}_{H_r} m_{1i}^*. \quad (3-85)$$

At a future instant in time, when the static reference object \mathcal{F}_2^* is in the field of view of the current camera (i.e. \mathcal{I}) and the daisy chaining method has been used to relate camera frames \mathcal{I} and \mathcal{I}_R in terms of the reference object \mathcal{F}_2^* , then (3-78), (3-82), and (3-84) can be used to relate the rotation and translation between \mathcal{I} and \mathcal{I}_R in terms of the normalized Euclidean coordinates of the reference object \mathcal{F}_2^* as⁴

$$m_{r2i}^* = \underbrace{\frac{z_{2i}^*}{z_{r2i}}}_{\alpha_{ri}} \underbrace{\left(R_r + x_{hr} n_2^{*T} \right)}_{H_r} m_{2i}^* \quad (3-86)$$

where $m_{r2i}^* \in \mathbb{R}^3$ represent virtual normalized Euclidean coordinates since the stationary reference object \mathcal{F}_2^* is not in the field of view of the stationary reference camera \mathcal{I}_R . The relationship between \mathcal{F} and \mathcal{F}_r can be expressed as [42]

$$m'_i = \underbrace{\frac{z_i}{z'_i}}_{\alpha'_i} \underbrace{\left(R' + x'_h n^T \right)}_{H'} m_i. \quad (3-87)$$

Similarly, using (3-80) and (4-25) the rotation and translation between the coordinate systems \mathcal{F} and \mathcal{F}_d can now be related in terms of the normalized Euclidean coordinates of the UGV expressed in \mathcal{I}_R as

$$m_{rdi} = \underbrace{\frac{z'_{ri}}{z_{rdi}}}_{\alpha_{rdi}} \underbrace{\left(R'_{rd} + x'_{hrd} n_r'^T \right)}_{H_{rd}} m'_{ri}. \quad (3-88)$$

In (3-85)-(3-88), $\alpha'_i(t)$, $\alpha_{rdi}(t)$, $\alpha_{ri}(t) \in \mathbb{R}$ denote depth ratios, $H'(t)$, $H_{rd}(t)$, $H_r(t) \in \mathbb{R}^{3 \times 3}$ denote Euclidean homographies [53], and $x'_h(t)$, $x'_{hrd}(t)$, $x_{hr}(t) \in \mathbb{R}^3$ denote scaled

⁴ Homography relationship in (3-86) relates camera frames \mathcal{I} and \mathcal{I}_R utilizing the static reference object \mathcal{F}_2^* however, given development can be generalized for any reference object \mathcal{F}_n^* ($n = 2, 3, ..m$).

translation vectors that are defined as follows

$$x'_h = \frac{x'_f}{d} \quad x'_{hrd} = \frac{x'_{frd}}{d'_r} \quad x_{hr} = \frac{x_{fr}}{d_1^*} \quad (3-89)$$

where the scaled translation $x_{hr}(t)$ is obtained when the relationship between \mathcal{I} and \mathcal{I}_R is expressed in terms of the static reference object \mathcal{F}_1^* ,

$$x_{hr} = \frac{x_{fr}}{d_2^*}. \quad (3-90)$$

In (3-90), the scaled translation $x_{hr}(t)$ is obtained when the static reference object \mathcal{F}_2^* is in the field of view of current camera frame (i.e. \mathcal{I}) and daisy chaining strategy has established a connection between camera frames \mathcal{I} and \mathcal{I}_R in terms of the reference object \mathcal{F}_2^* . Each Euclidean feature point will have a projected pixel coordinate expressed in terms of \mathcal{I} as

$$p_i = \begin{bmatrix} u_i & v_i & 1 \end{bmatrix}^T \quad p_{1i}^* = \begin{bmatrix} u_{1i}^* & v_{1i}^* & 1 \end{bmatrix}^T \quad (3-91)$$

$$p_{2i}^* = \begin{bmatrix} u_{2i}^* & v_{2i}^* & 1 \end{bmatrix}^T \quad (3-92)$$

where $p_i(t)$, $p_{1i}^*(t)$, and $p_{2i}^*(t) \in \mathbb{R}^3$ represents the image-space coordinates of the time-varying feature points of the UGV and reference objects \mathcal{F}_1^* and \mathcal{F}_2^* , respectively, and $u_i(t)$, $v_i(t)$, $u_{1i}^*(t)$, $v_{1i}^*(t)$, $u_{2i}^*(t)$, $v_{2i}^*(t) \in \mathbb{R}$. Similarly, the projected pixel coordinate of the Euclidean features in the reference image can be expressed in terms of \mathcal{I}_R as

$$p_{r1i}^* = \begin{bmatrix} u_{r1i}^* & v_{r1i}^* & 1 \end{bmatrix}^T \quad (3-93)$$

where $p_{r1i}^* \in \mathbb{R}^3$ represents the constant image-space coordinates of the stationary reference object \mathcal{F}_1^* and u_{r1i}^* , $v_{r1i}^* \in \mathbb{R}$. To calculate the Euclidean homographies given in (3-85)-(3-88) from pixel information, the projected pixel coordinates are related to $m_i(t)$,

$m_{1i}^*(t)$, $m_{2i}^*(t)$, and m_{r1i}^* by the pin-hole camera model as

$$p_i = Am_i \quad p_{1i}^* = Am_{1i}^* \quad (3-94)$$

$$p_{2i}^* = Am_{2i}^* \quad p_{r1i}^* = Am_{r1i}^* \quad (3-95)$$

Also, the pin-hole camera model relationship for the normalized Euclidean coordinates m_{r2i}^* , $m'_i(t)$, $m'_{ri}(t)$, and m_{rdi} can be formulated in terms of the virtual pixel coordinates p_{r2i}^* , $p'_i(t)$, $p'_{ri}(t)$, and p_{rdi} as follows:

$$p_{r2i}^* = Am_{r2i}^* \quad p'_i = Am'_i \quad (3-96)$$

$$p'_{ri} = Am'_{ri} \quad p_{rdi} = Am_{rdi} \quad (3-97)$$

where $A \in \mathbb{R}^{3 \times 3}$ is a known, constant, and invertible intrinsic camera calibration matrix. By using (3-85)-(3-88), (3-94), and (3-95), the following relationships can be developed:

$$p_{r1i}^* = \alpha_{ri} G_r p_{1i}^* \quad p_{r2i}^* = \alpha_{ri} G_r p_{2i}^* \quad (3-98)$$

$$p'_i = \alpha'_i G' p_i \quad p_{rdi} = \alpha_{rdi} G_{rd} p'_{ri} \quad (3-99)$$

where $G_r(t) = [g_{rij}(t)]$, $G'(t) = [g'_{ij}(t)]$, $G_{rd} = [g_{rdij}] \forall i, j = 1, 2, 3 \in \mathbb{R}^{3 \times 3}$ denote projective homographies.

Sets of linear equations can be developed from (3-98) and (3-99) to determine the projective homographies up to a scalar multiple. Various techniques can be used (e.g., see[48, 49]) to decompose the Euclidean homographies, to obtain $\alpha_{ri}(t)$, $\alpha'_i(t)$, $\alpha_{rdi}(t)$, $x_{hr}(t)$, $x'_h(t)$, $x'_{hrd}(t)$, $R_r(t)$, $R'(t)$, $R'_{rd}(t)$. Using the known geometric length s_{21i} and a unit normal n_1^* , obtained from homography decomposition of (3-85), geometric reconstruction method can be utilized to obtain $\bar{m}_{1i}^*(t)$ and $d_1^*(t)$. Hence, the translation $x_{fr}(t)$ between \mathcal{I} and \mathcal{I}_R can be recovered from (3-89). Also, the Euclidean coordinates \bar{m}_{ri} of the UGV corresponding to the stationary reference pose can be obtained from geometric reconstruction. Thus, $\bar{m}'_i(t)$ can be computed from (3-70). Using (3-81), (3-87), (3-96), and (3-99), the projective homography can be defined between $p'_i(t)$ and

$p_i(t)$, which can be decomposed to obtain a unit normal $n(t)$ and hence the time-varying Euclidean coordinates $\bar{m}_i(t)$. The Euclidean coordinates $\bar{m}'_{ri}(t)$, corresponding to the current UGV position as seen by reference camera \mathcal{I}_R , can be obtained using (3-70). Therefore, using (3-83) and (3-97) a projective homography relationship can be obtained between the current UGV (i.e. $\mathcal{F}(t)$) and the desired UGV (i.e. \mathcal{F}_d) in terms of a stationary reference camera coordinate system \mathcal{I}_R given by (3-99).

Further, when the reference object \mathcal{F}_2^* appears in the field of view of \mathcal{I} , the Euclidean position $\bar{m}_{2i}^*(t)$ can be obtained. Using (3-69), (3-82), (3-84), (3-86), and (3-98), a projective homography relationship can be defined between p_{r2i}^* and $p_{2i}^*(t)$, which can be decomposed to obtain rotation and translation $R_r(t)$, $x_{fr}(t)$ between \mathcal{I} and \mathcal{I}_R . Once $R_r(t)$ and $x_{fr}(t)$ have been determined, the future relationship can be expressed with respect to the new reference object (i.e. \mathcal{F}_2^*) and similarly, can be generalized for $n = 2, 3, \dots, m$.

3.3.3 UGV Kinematics

The kinematic model for the UGV can be determined from Fig. 3-3 as

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_d \end{bmatrix} = \begin{bmatrix} \cos \theta_d & 0 \\ \sin \theta_d & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} \quad (3-100)$$

where \dot{x}_c , \dot{y}_c , and $\dot{\theta}_d$ denote the time derivatives of $x_c(t)$, $y_c(t)$, and $\theta_d(t) \in \mathbb{R}$, respectively, where $x_c(t)$ and $y_c(t)$ denote the planar position of \mathcal{F} expressed in \mathcal{F}_d , and $\theta_d(t) \in \mathbb{R}$ denotes the right-handed rotation angle about the z-axis of \mathcal{F} that aligns \mathcal{F} with \mathcal{F}_d , and $v_c(t)$ and $\omega_c(t)$ were introduced in Section 3.2.1 and are depicted in Figs. 3-3 and 3-4. Based on the definition for $R'_{rd}(t)$ provided in the previous development, the rotation from

\mathcal{F} to \mathcal{F}_d can be developed as

$$R'_{rd} = \begin{bmatrix} \cos \theta_d & -\sin \theta_d & 0 \\ \sin \theta_d & \cos \theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3-101)$$

Based on the fact that $R'_{rd}(t)$ can be obtained from (3-88) and (3-99), it is clear from (3-101) that $\theta_d(t)$ is a known signal that can be used in the control development.

The geometric relationships between the coordinate frames can be used to develop the following expression

$$\begin{bmatrix} x_c & y_c & 0 \end{bmatrix}^T = R'_{rd}{}^T(x'_{fr} - x_{frd}). \quad (3-102)$$

After utilizing, (3-58), (3-60), (3-88), and the assumption (as in [46]) that $s_{11} = [0, 0, 0]^T$, the following expression can be obtained⁵

$$\begin{bmatrix} \frac{x_c}{z_{rd1}} & \frac{y_c}{z_{rd1}} & 0 \end{bmatrix}^T = R'_{rd}{}^T\left(\frac{z'_{r1}}{z_{rd1}}m'_{r1} - m_{rd1}\right). \quad (3-103)$$

Since the terms on the right-hand side of (3-103) are known or measurable (refer to Section 3.3.2), then $\frac{x_c(t)}{z_{rd1}}$ and $\frac{y_c(t)}{z_{rd1}}$ can be used in the control development.

Based on the form of (3-101 and 3-103), control development and Lyapunov-based stability analysis arguments provided in Section 3.2 can be used to prove asymptotic regulation of the UGV.

3.3.4 Simulation Results

A numerical simulation was performed to illustrate the performance of the multi-reference regulation control given the controller in (3-44) and (3-45). The simulation scenario is shown in Fig. 3-5, such that the pose of current UGV $\mathcal{F}(t)$ is estimated with respect

⁵ Any point s_{1i} , s_{2ni} can be utilized in the subsequent development; however, to reduce the notational complexity, we have elected to select the image point s_{11} , s_{2n1} , and hence, the subscript 1 is utilized in lieu of i in the subsequent development.

to four stationary reference objects \mathcal{F}_1^* , \mathcal{F}_2^* , \mathcal{F}_3^* , \mathcal{F}_4^* while regulating to the desired pose corresponding to the coordinate frame \mathcal{F}_d . The origins of the coordinate frames \mathcal{F} , \mathcal{F}_1^* , \mathcal{F}_2^* , \mathcal{F}_3^* , \mathcal{F}_4^* , and \mathcal{F}_d , and the four coplanar feature points on the planes π , π_1^* , π_2^* , π_3^* , π_4^* , and π_d are chosen such that the Euclidean coordinates of the feature points in \mathcal{F} , \mathcal{F}_1^* , \mathcal{F}_2^* , \mathcal{F}_3^* , \mathcal{F}_4^* , and \mathcal{F}_d are given by s_i , s_{1i} , s_{2i} , s_{3i} , s_{4i} , s_i (where $i = 1, 2, 3, 4$), respectively.

The initial pose of current UGV $\mathcal{F}(0) = \mathcal{F}(t)|_{t=0}$, stationary reference objects \mathcal{F}_j^* , $j = 1, 2, 3, 4$, and the desired pose \mathcal{F}_d were considered as

$$\begin{aligned} \mathcal{F}(0) &= \begin{bmatrix} \cos(60) & -\sin(60) & 0 & -3.40 \\ \sin(60) & \cos(60) & 0 & -1.00 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \mathcal{F}_1^* &= \begin{bmatrix} \cos(40) & -\sin(40) & 0 & -3.25 \\ \sin(40) & \cos(40) & 0 & 0.50 \\ 0 & 0 & 1 & -1.00 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ \mathcal{F}_2^* &= \begin{bmatrix} \cos(20) & -\sin(20) & 0 & -1.50 \\ \sin(20) & \cos(20) & 0 & -0.15 \\ 0 & 0 & 1 & -0.80 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \mathcal{F}_3^* &= \begin{bmatrix} \cos(00) & -\sin(00) & 0 & -1.25 \\ \sin(00) & \cos(00) & 0 & 1.50 \\ 0 & 0 & 1 & -1.50 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ \mathcal{F}_4^* &= \begin{bmatrix} \cos(10) & -\sin(10) & 0 & 0.15 \\ \sin(10) & \cos(10) & 0 & 0.15 \\ 0 & 0 & 1 & -2.25 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \mathcal{F}_d &= \begin{bmatrix} \cos(-30) & -\sin(-30) & 0 & 0.40 \\ \sin(-30) & \cos(-30) & 0 & 1.00 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

The control gains in (3-44) and (3-45) were selected as

$$k_\omega = 2.2025 \quad k_v = 12.9275.$$

The Euclidean space trajectory of the moving camera \mathcal{I} along with initial and final position of the time-varying UGV $\mathcal{F}(t)$ and desired UGV \mathcal{F}_d is shown in Fig. 3-5. The regulation errors are plotted in Fig. 3-6, which asymptotically approach zero. The linear

and angular velocity control inputs are shown in Fig. 3-7. Figs. 3-8 and 3-9 show the regulation results in presence of an additive white Gaussian noise of standard deviation $\sigma = 0.1 \text{ pixels}$.

3.3.5 Concluding Remarks

In this chapter, the pose of a moving sensorless UGV is regulated to a desired pose defined by a goal image using a collaborative visual servo control strategy. To achieve the result, multiple views of a reference object were used to develop Euclidean homographies. By decomposing the Euclidean homographies into separate translation and rotation components reconstructed Euclidean information was obtained for the control development. The impact of this research is a new framework to relate the pose of a moving target through images acquired by a moving camera. Further, the results are extended to include the scenario such that the stationary reference objects can leave the camera FOV and new objects enter the FOV, thus increasing the applicative area of the UGV/camera pair. The development in next chapter will target an extension of daisy-chaining method to a tracking control result.

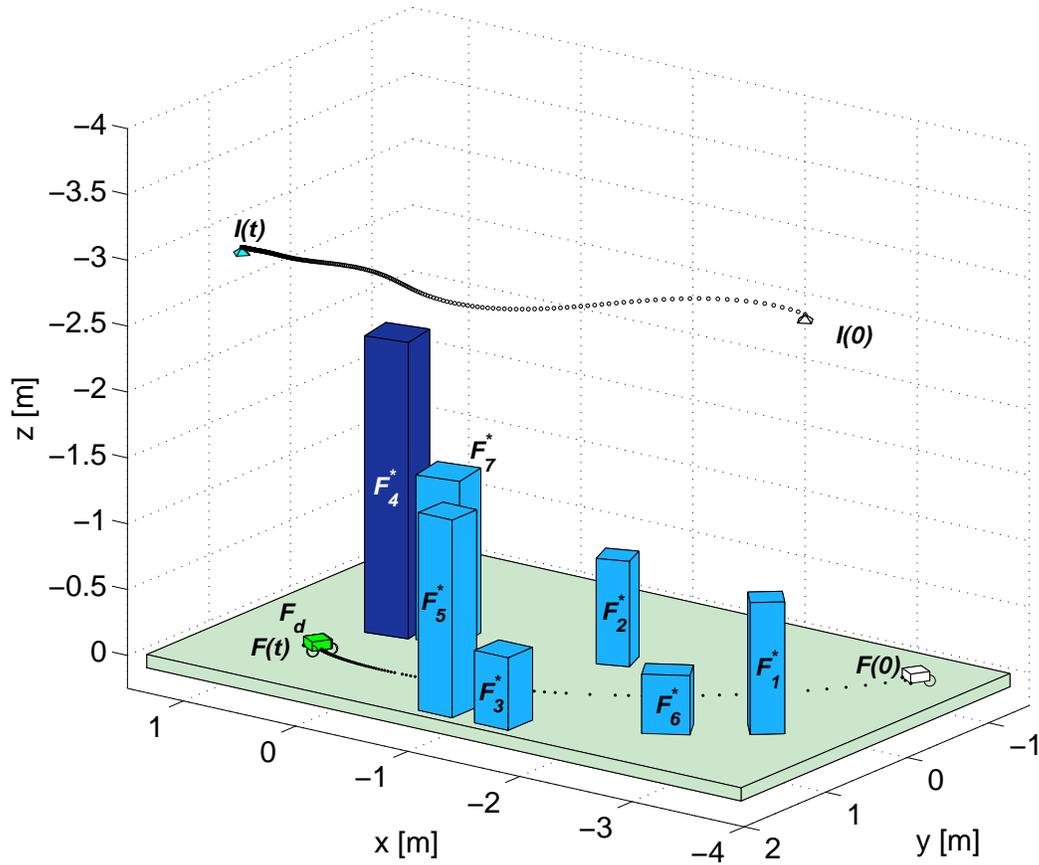


Figure 3-5. Euclidean space trajectory of the moving camera \mathcal{I} , initial and final position of the time-varying UGV $\mathcal{F}(t)$, and desired UGV \mathcal{F}_d . $\mathcal{F}(0)$ denotes the initial position of the UGV, $\mathcal{I}(0)$ denotes the initial position of the moving camera, $\mathcal{I}(t)$ denotes the time-varying position of the moving camera, \mathcal{F}_i^* where $i = 1, 2, \dots, 7$ denotes the stationary reference objects, and $\mathcal{F}(t)$ denotes the regulated position of the UGV coincident with the desired UGV \mathcal{F}_d

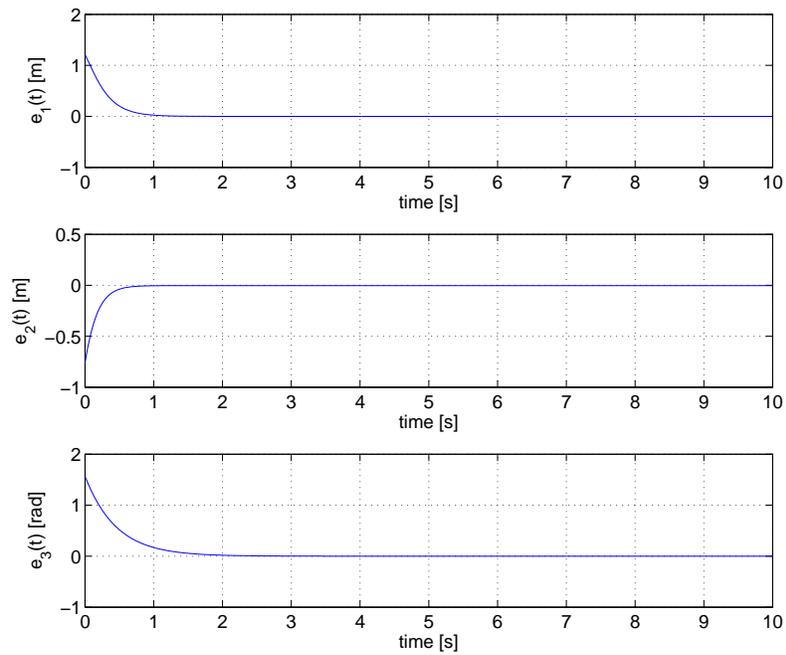


Figure 3-6. Linear (i.e. $e_1(t)$ and $e_2(t)$) and angular (i.e. $e_3(t)$) regulation error.

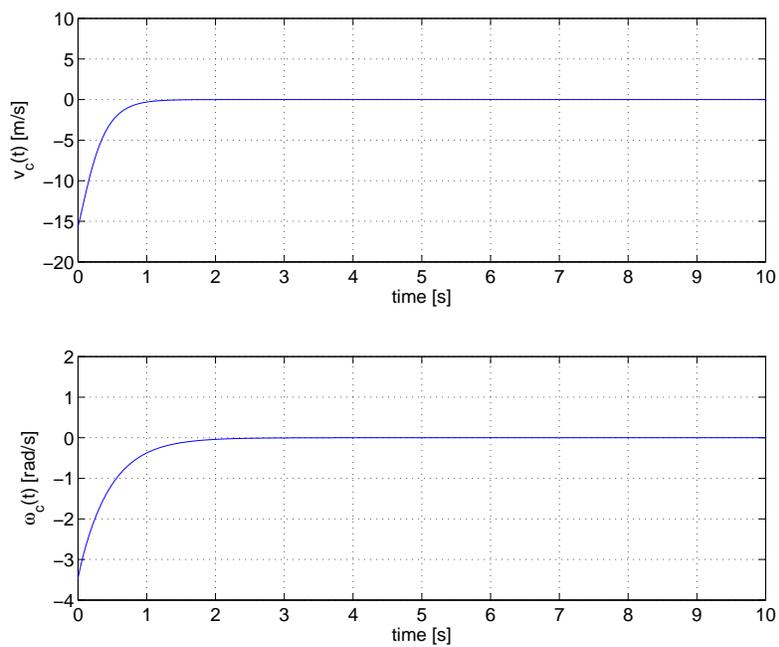


Figure 3-7. Linear (i.e. $v_c(t)$) and angular (i.e. $\omega_c(t)$) velocity control inputs.

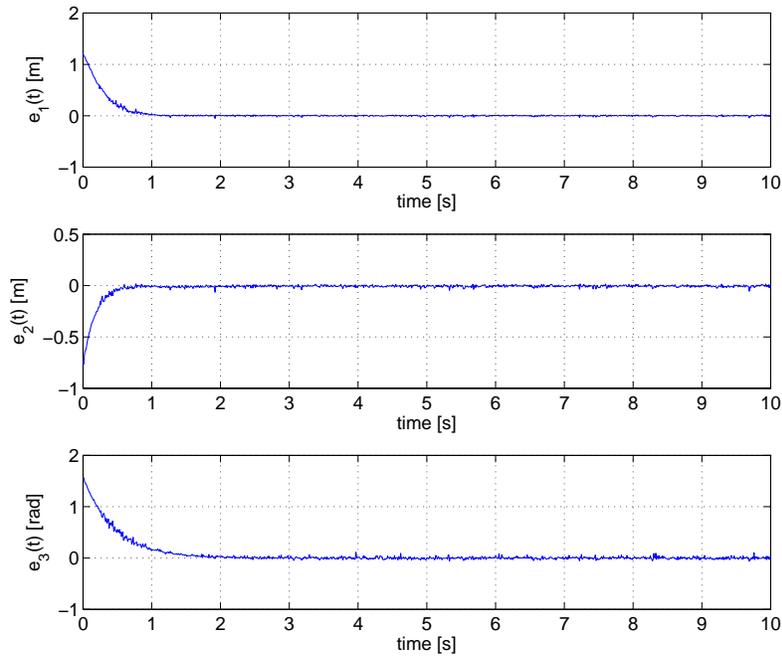


Figure 3-8. Linear (i.e. $e_1(t)$ and $e_2(t)$) and angular (i.e. $e_3(t)$) regulation error in presence of an additive white Gaussian noise.

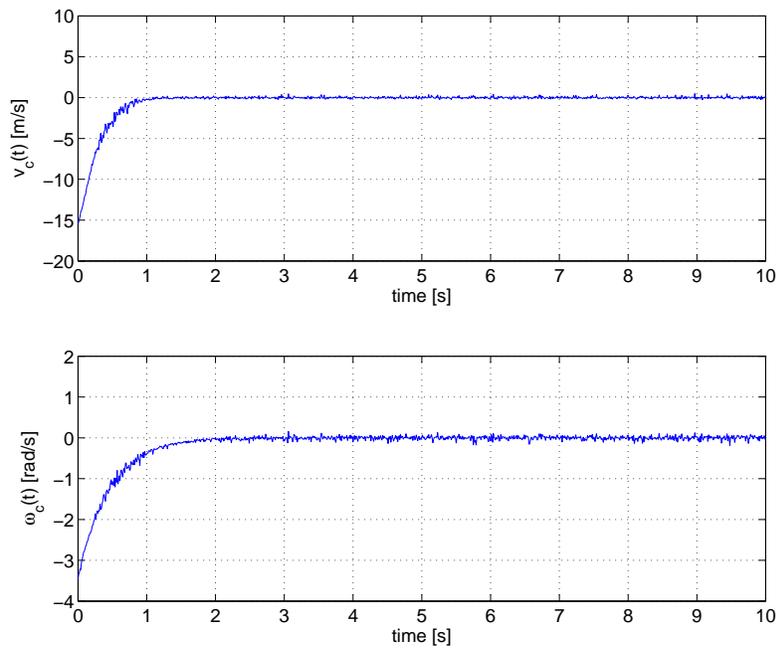


Figure 3-9. Linear (i.e. $v_c(t)$) and angular (i.e. $\omega_c(t)$) velocity control inputs in presence of an additive white Gaussian noise.

CHAPTER 4
A DAISY-CHAINING VISUAL SERVOING APPROACH WITH APPLICATIONS IN
TRACKING, LOCALIZATION, AND MAPPING

4.1 Introduction

A new daisy-chaining method developed in Chapter 3 is used for vision-based tracking control of a rigid-body object, such as an UGV, while providing localization of the moving camera and moving object in the world frame, and mapping the location of static landmarks in the world frame. Hence, this approach can be used in vSLAM of the UGV, with applications toward path planning, real time trajectory generation, obstacle avoidance, multi-vehicle coordination control and task assignment, etc. By using the daisy-chaining strategy, the coordinates of static features out of the FOV can also be estimated. The estimates of static features can be maintained as a map, or can be used as measurements in existing vSLAM methods.

For clarity, this chapter is presented in successive stages of increasing complexity. Section 4.2 introduces the imaging model, geometric model used in this chapter, as well as introduces the daisy-chaining method as applied to the case of controlling a six DOF planar object through visual data from a moving camera and fixed reference camera. These results are extended to the case of an UGV with nonholonomic constraints and a moving camera and moving reference camera in Section 4.3. The efforts of previous sections are then brought to bear on a tracking and mapping application, where the UGV is controlled to track a trajectory that takes the vehicle outside of the initial FOV of the camera. The daisy-chaining approach must be extended to allow for new fixed landmarks to enter the FOV and related to previous landmarks and the UGV.

4.2 Daisy-Chaining Based Tracking Control

In this section, a visual servo tracking controller is developed for a moving six DOF object based on daisy-chained image feedback from a moving camera. The control objective is to have the object track a desired trajectory determined by a sequence of prerecorded images from a stationary camera. To achieve this result, several technical

issues were resolved including: discriminating the relative velocity between the moving camera and the moving object, compensating for the unknown time-varying distance measurement from the camera to the object, relating the unknown attitude of the control object to some measurable signals, and using the unit quaternion to formulate the rotation motion and rotation error system. The relative velocity issue is resolved by utilizing multi-view image geometry to daisy-chain homography relationships between the moving camera frame and the moving object coordinate frames. By using the depth ratios obtained from the homography decomposition, the unknown depth information is related to an unknown constant that can be compensated for by a Lyapunov-based adaptive update law. Lyapunov-based methods are provided to prove the adaptive asymptotic tracking result.

4.2.1 Problem Scenario

Over the past decade, a variety of visual servo controllers have been addressed for both camera-to-hand and camera-in-hand configurations (e.g., see [54–57]). Typical camera-to-hand and camera-in-hand visual servo controllers have required that either the camera or the target remain stationary so that an absolute velocity can be determined and used in the control development. For the problem of a moving camera tracking a moving target (i.e. control of relative pose/velocity), integral control or predictive Kalman filters have been used to overcome the unknown target velocity [58, 59]. In contrast to these methods, the development in this section and our previous preliminary work in [42, 43, 60] is motivated by the problem when the camera and the target are moving. A practical example application of this scenario is an airborne camera attached to a remote controlled aircraft that is used to determine pose measurements of an UGV and then relay the information to the UGV for closed-loop control.

The scenario examined in this section is depicted in Fig. 4-1, where various coordinate frames are defined as a means to develop the subsequent Euclidean reconstruction and control methods. In Fig. 4-1, a stationary coordinate frame \mathcal{I}_R is attached to a

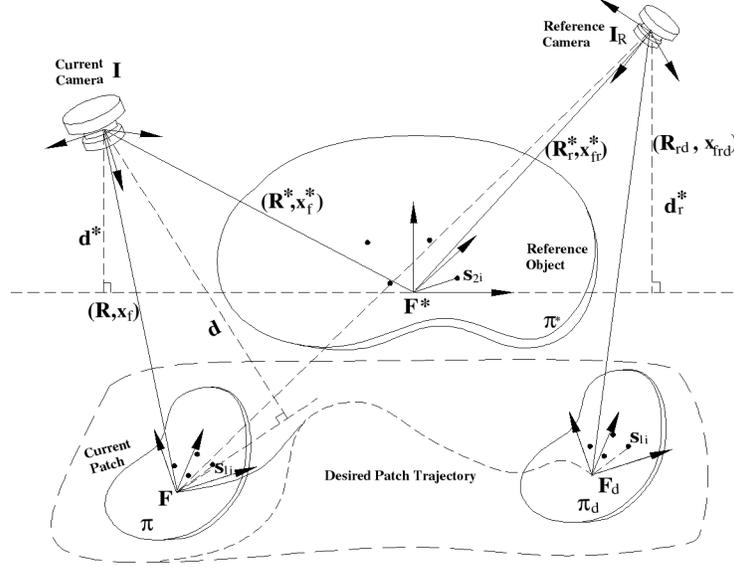


Figure 4-1. Geometric model for a moving camera (coordinate frame \mathcal{I}), moving target (coordinate frame \mathcal{F}) and stationary reference camera (coordinate frame \mathcal{I}_R).

camera and a time-varying coordinate frame \mathcal{F}_d is attached to some mobile object (e.g., an aircraft, a ground vehicle, a marine vessel). The object is identified by an image through a collection of feature points that are assumed (without loss of generality¹) to be coplanar and non-collinear (i.e., a planar patch of feature points). The camera attached to \mathcal{I}_R a priori records a series of snapshots (i.e., a video) of the motion of the coordinate frame \mathcal{F}_d until \mathcal{F}_d comes to rest. A stationary coordinate frame \mathcal{F}^* is attached to another planar patch of feature points that are assumed to be visible in every frame of the video recorded by the camera. For example, the camera attached to \mathcal{I}_R is on-board a “stationary” satellite that takes a series of snapshots of the relative motion of \mathcal{F}_d with respect to \mathcal{F}^* . Therefore, the desired motion of \mathcal{F}_d can be encoded as a series of relative translations and rotations with respect to the stationary frame \mathcal{F}^* a priori. Spline functions or filter

¹ Image processing techniques can often be used to select coplanar and non-collinear feature points within an image. However, if four coplanar target points are not available then the subsequent development can also exploit the virtual parallax method [45, 61] where the non-coplanar points are projected onto a virtual plane.

algorithms can then be used to generate a smooth desired feature point trajectory as described in [46].

Fig. 4-1 also depicts a time-varying coordinate frame \mathcal{I} that is attached to another camera (e.g., a camera attached to a remote controlled aircraft), and a time-varying coordinate frame \mathcal{F} that is attached to the current pose of the planar patch. The camera attached to \mathcal{I} captures snapshots of the planar patches associated with \mathcal{F} and \mathcal{F}^* , respectively. The a priori motion of \mathcal{F}_d represents the desired trajectory of the coordinate system \mathcal{F} , where \mathcal{F} and \mathcal{F}_d are attached to identical objects, but at different points in time. The camera attached to \mathcal{I}_R can be a different camera (with different calibration parameters) as the camera attached to \mathcal{I} . Based on these coordinate frame definitions, the problem considered in this section is to develop a kinematic controller for the object attached to \mathcal{F} so that the time-varying rotation and translation of \mathcal{F} converges to the desired time-varying rotation and translation of \mathcal{F}_d , where the motion of \mathcal{F} is determined from the time-varying overhead camera attached to \mathcal{I} .

4.2.2 Geometric Relationships

Relationships between the various coordinate frames are summarized in Table 4-1. In Table 4-1, $R(t)$, $R^*(t)$, $R_r(t)$, $R'(t)$, $R_{rd}(t)$, $R_r^* \in SO(3)$ denote rotation matrices, and $x_{fr}(t)$, $x'_{fr}(t)$, $x_{frd}(t)$, $x_{fr}^* \in \mathbb{R}^3$ denote translation vectors. From Fig. 4-1, the translation $x'_{fr}(t)$ and the rotation $R'(t)$ can be expressed as

$$\begin{aligned} x'_{fr} &= x_{fr}^* + R_r^* R^{*T} (x_f - x_f^*) \\ R' &= R_r^* R^{*T} R. \end{aligned} \tag{4-1}$$

As illustrated in Fig. 4-1, π , π_d and π^* denote the planes of feature points associated with \mathcal{F} , \mathcal{F}_d , and \mathcal{F}^* , respectively. The constant Euclidean coordinates of the i -th feature point in \mathcal{F} (and also \mathcal{F}_d) are denoted by $s_{1i} \in \mathbb{R}^3 \forall i = 1, 2, \dots, n$ ($n \geq 4$), and $s_{2i} \in \mathbb{R}^3 \forall i = 1, 2, \dots, n$ denotes the constant Euclidean coordinates of the i -th feature point in

Table 4-1. Coordinate frames relationships for 6-DOF planar object tracking control.

Motion	Frames
$R(t), x_f(t)$	\mathcal{F} to \mathcal{I} in \mathcal{I}
$R^*(t), x_f^*(t)$	\mathcal{F}^* to \mathcal{I} in \mathcal{I}
$R_r(t), x_{fr}(t)$	\mathcal{I} to \mathcal{I}_R
$R'(t), x_{fr}(t)$	\mathcal{F} to \mathcal{I}_R in \mathcal{I}_R
R_r^*, x_{fr}^*	\mathcal{F}^* to \mathcal{I}_R in \mathcal{I}_R
$R_{rd}(t), x_{frd}(t)$	\mathcal{F}_d to \mathcal{I}_R in \mathcal{I}_R

\mathcal{F}^* . From the geometry between the coordinate frames depicted in Fig. 4-1, the following relationships can be developed

$$\bar{m}_i = x_f + R s_{1i} \quad \bar{m}_{rdi} = x_{frd} + R_{rd} s_{1i} \quad (4-2)$$

$$\bar{m}_{ri}^* = x_{fr}^* + R_r^* s_{2i} \quad \bar{m}'_i = x'_{fr} + R' s_{1i} \quad (4-3)$$

$$\bar{m}_i^* = x_f^* + R^* s_{2i}. \quad (4-4)$$

In (4-2)-(4-4), $\bar{m}_i(t), \bar{m}_i^*(t) \in \mathbb{R}^3$ denote the Euclidean coordinates of the feature points on π and π^* , respectively, expressed in \mathcal{I} as

$$\bar{m}_i(t) \triangleq \begin{bmatrix} x_i(t) & y_i(t) & z_i(t) \end{bmatrix}^T \quad (4-5)$$

$$\bar{m}_i^*(t) \triangleq \begin{bmatrix} x_i^*(t) & y_i^*(t) & z_i^*(t) \end{bmatrix}^T, \quad (4-6)$$

$\bar{m}'_i(t), \bar{m}_{rdi}(t) \in \mathbb{R}^3$ denote the actual and desired time-varying Euclidean coordinates, respectively, of the feature points on π expressed in \mathcal{I}_R as

$$\bar{m}'_i(t) \triangleq \begin{bmatrix} x'_i(t) & y'_i(t) & z'_i(t) \end{bmatrix}^T \quad (4-7)$$

$$\bar{m}_{rdi}(t) \triangleq \begin{bmatrix} x_{rdi}(t) & y_{rdi}(t) & z_{rdi}(t) \end{bmatrix}^T, \quad (4-8)$$

and $\bar{m}_{ri}^* \in \mathbb{R}^3$ denotes the constant Euclidean coordinates of the feature points on the plane π^* expressed in \mathcal{I}_R as

$$\bar{m}_{ri}^* \triangleq \begin{bmatrix} x_{ri}^* & y_{ri}^* & z_{ri}^* \end{bmatrix}^T. \quad (4-9)$$

After some algebraic manipulation, the expressions in (4-2)-(4-4) can be rewritten as

$$\bar{m}_i^* = \bar{x}_n + R_n \bar{m}_i \quad (4-10)$$

$$\bar{m}_i = \bar{x}_f + \bar{R} \bar{m}_i^* \quad \bar{m}_{rdi} = \bar{x}_{f rd} + \bar{R}_{rd} \bar{m}_{ri}^* \quad (4-11)$$

$$\bar{m}_{ri}^* = x_{fr} + R_r \bar{m}_i^* \quad \bar{m}_i' = x_{fr} + R_r \bar{m}_i, \quad (4-12)$$

where $R_n(t)$, $\bar{R}(t)$, $\bar{R}_{rd}(t)$, $R_r(t) \in SO(3)$ and $\bar{x}_n(t)$, $\bar{x}_f(t)$, $\bar{x}_{f rd}(t)$, $x_{fr}(t) \in \mathbb{R}^3$ are new rotation and translation variables, respectively, defined as²

$$R_n = R^* R^T \quad \bar{R} = R R^{*T} \quad (4-13)$$

$$\bar{R}_{rd} = R_{rd} R_r^{*T} \quad R_r = R_r^* R^{*T}$$

$$\bar{x}_n = x_f^* - R_n (x_f - R(s_{2i} - s_{1i})) \quad (4-14)$$

$$\bar{x}_f = x_f - \bar{R} (x_f^* + R^* (s_{2i} - s_{1i})) \quad (4-15)$$

$$\bar{x}_{f rd} = x_{f rd} - \bar{R}_{rd} (x_{fr}^* + R_r^* (s_{2i} - s_{1i})) \quad (4-16)$$

$$x_{fr} = x_{fr}^* - R_r x_f^* = x_{fr}' - R_r x_f. \quad (4-17)$$

To facilitate the development of a relationship between the actual Euclidean translation of \mathcal{F} to the Euclidean translation that is reconstructed from the image information, projective relationships are developed from Fig. 4-1 as

$$d(t) = n^T \bar{m}_i \quad d^*(t) = n^{*T} \bar{m}_i^* \quad d_r^* = n_r^{*T} \bar{m}_{ri}^*, \quad (4-18)$$

where $d(t) \in \mathbb{R}$ represents the distance from the origin of \mathcal{I} to π along the unit normal (expressed in \mathcal{I}) to π denoted as $n(t) \in \mathbb{R}^3$, $d^*(t) \in \mathbb{R}$ represents the distance from the

² Note that $R_n(t)$, $\bar{R}(t)$ and $\bar{R}_{rd}(t)$ in (4-13) are the rotation matrices between \mathcal{F} and \mathcal{F}^* , \mathcal{F}^* and \mathcal{F} , and \mathcal{F}^* and \mathcal{F}_d , respectively, but $\bar{x}_n(t)$, $\bar{x}_f(t)$ and $\bar{x}_{f rd}(t)$ in (4-14)-(4-16) are not the translation vectors between the corresponding coordinate frames. Only the rotation matrices will be used in the controller development.

origin of \mathcal{I} to π^* along the unit normal (expressed in \mathcal{I}) to π^* denoted as $n^*(t) \in \mathbb{R}^3$, and $d_r^* \in \mathbb{R}$ represents the distance from the origin of \mathcal{I}_R to π^* along the unit normal (expressed in \mathcal{I}_R) to π^* denoted as $n_r^* \in \mathbb{R}^3$ where $n^*(t) = R_r^T(t)n_r^*$. In (4-18), $d(t)$, $d^*(t)$, $d_r^* > \varepsilon$ for some positive constant $\varepsilon \in \mathbb{R}$. Based on (4-18), the relationships in (4-10)-(4-12) can be expressed as

$$\bar{m}_i^* = \left(R_n + \frac{\bar{x}_n}{d} n^T \right) \bar{m}_i \quad (4-19)$$

$$\bar{m}_i = \left(\bar{R} + \frac{\bar{x}_f}{d^*} n^{*T} \right) \bar{m}_i^* \quad (4-20)$$

$$\bar{m}_{r di} = \left(\bar{R}_{rd} + \frac{\bar{x}_{f rd}}{d_r^*} n_r^{*T} \right) \bar{m}_{r i}^* \quad (4-21)$$

$$\bar{m}_{r i}^* = \left(R_r + \frac{x_{f r}}{d^*} n^{*T} \right) \bar{m}_i^* \quad (4-22)$$

$$\bar{m}_i' = \left(R_r + \frac{x_{f r}}{d} n^T \right) \bar{m}_i. \quad (4-23)$$

As in [46], the subsequent development requires that the constant rotation matrix R_r^* be known. The constant rotation matrix R_r^* can be obtained a priori using various methods (e.g., a second camera, additional on-board sensors, off-line calibration, Euclidean measurements). The subsequent development also assumes that the difference between the Euclidean distances ($s_{2i} - s_{1i}$) is a constant $\forall i = 1, \dots, n$. While there are many practical applications that satisfy this assumption (e.g., a simple scenario is that the objects attached to \mathcal{F} and \mathcal{F}^* are the identical objects), the assumption is generally restrictive and is the focus of future research. As described in our preliminary work in [62], each of these assumptions can be avoided by using the geometric reconstruction approach in [52, 63, 64] under an alternative assumption that the Euclidean distance between two feature points is precisely known.

4.2.3 Euclidean Reconstruction

The relationships given by (4-19)-(4-23) provide a means to quantify a translation and rotation error between the different coordinate systems. Since the pose of \mathcal{F} , \mathcal{F}_d , and \mathcal{F}^* cannot be directly measured, a Euclidean reconstruction is developed to obtain the

pose error by comparing multiple images acquired from the hovering monocular vision system. To facilitate the subsequent development, the normalized Euclidean coordinates of the feature points in π and π^* can be expressed in terms of \mathcal{I} as $m_i(t), m_i^*(t) \in \mathbb{R}^3$, respectively, as

$$m_i \triangleq \frac{\bar{m}_i}{z_i} \quad m_i^* \triangleq \frac{\bar{m}_i^*}{z_i^*}. \quad (4-24)$$

Similarly, the normalized Euclidean coordinates of the feature points in π, π_d and π^* can be expressed in terms of \mathcal{I}_R as $m_i'(t), m_{rdi}(t), m_{ri}^* \in \mathbb{R}^3$, respectively, as

$$m_i'(t) \triangleq \frac{\bar{m}_i'(t)}{z_i'(t)} \quad m_{rdi}(t) \triangleq \frac{\bar{m}_{rdi}(t)}{z_{rdi}(t)} \quad m_{ri}^* \triangleq \frac{\bar{m}_{ri}^*}{z_{ri}^*}. \quad (4-25)$$

From the expressions given in (4-20) and (4-24), the rotation and translation between the coordinate systems \mathcal{F} and \mathcal{F}^* , between \mathcal{F}^* and \mathcal{F}_d , and between \mathcal{I} and \mathcal{I}_R can now be related in terms of the normalized Euclidean coordinates as

$$m_i = \alpha_i (\bar{R} + x_h n^{*T}) m_i^*, \quad (4-26)$$

$$m_i^* = \frac{1}{\alpha_i} (R_n + x_{nh} n^T) m_i \quad (4-27)$$

$$m_{rdi} = \alpha_{rdi} (\bar{R}_{rd} + x_{hrd} n_r^{*T}) m_{ri}^*, \quad (4-28)$$

$$m_{ri}^* = \alpha_{ri} (R_r + x_{hr} n^{*T}) m_i^*, \quad (4-29)$$

where $\alpha_i(t), \alpha_{rdi}(t), \alpha_{ri}(t) \in \mathbb{R}$ denote depth ratios defined as

$$\alpha_i = \frac{z_i^*}{z_i} \quad \alpha_{rdi} = \frac{z_{ri}^*}{z_{rdi}} \quad \alpha_{ri} = \frac{z_i^*}{z_{ri}^*},$$

and $x_h(t), x_{nh}(t), x_{hrd}(t), x_{hr}(t) \in \mathbb{R}^3$ denote scaled translation vectors that are defined as

$$\begin{aligned} x_h &= \frac{\bar{x}_f}{d^*} & x_{nh} &= \frac{\bar{x}_n}{d} \\ x_{hrd} &= \frac{\bar{x}_{frd}}{d_r^*} & x_{hr} &= \frac{x_{fr}}{d^*}. \end{aligned} \quad (4-30)$$

Since the normalized Euclidean coordinates in (4-26)-(4-29) can not be directly measured, the following relationships (i.e., the pin-hole camera model) are used to

determine the normalized Euclidean coordinates from pixel information

$$p_i = A_1 m_i \quad p_i^* = A_1 m_i^* \quad (4-31)$$

$$p_{rdi} = A_2 m_{rdi} \quad p_{ri}^* = A_2 m_{ri}^*, \quad (4-32)$$

where $A_1, A_2 \in \mathbb{R}^{3 \times 3}$ are known, constant, and invertible intrinsic camera calibration matrices of the current camera and the reference camera, respectively. In (4-31) and (4-32), $p_i(t), p_i^*(t) \in \mathbb{R}^3$ represent the image-space coordinates of the Euclidean feature points on π and π^* expressed in terms of \mathcal{I} as

$$p_i \triangleq \begin{bmatrix} u_i & v_i & 1 \end{bmatrix}^T \quad p_i^* \triangleq \begin{bmatrix} u_i^* & v_i^* & 1 \end{bmatrix}^T, \quad (4-33)$$

respectively, where $u_i(t), v_i(t), u_i^*(t), v_i^*(t) \in \mathbb{R}$. Similarly, $p_{rdi}(t), p_{ri}^* \in \mathbb{R}^3$ represent the image-space coordinates of the Euclidean features on π_d and π^* expressed in terms of \mathcal{I}_R as

$$p_{rdi} \triangleq \begin{bmatrix} u_{rdi} & v_{rdi} & 1 \end{bmatrix}^T \quad p_{ri}^* \triangleq \begin{bmatrix} u_{ri}^* & v_{ri}^* & 1 \end{bmatrix}^T \quad (4-34)$$

respectively, where $u_{rdi}(t), v_{rdi}(t), u_{ri}^*, v_{ri}^* \in \mathbb{R}$. By using (4-26)-(4-29) and (4-31)-(4-34), the following relationships can be developed:

$$p_i = \alpha_i \underbrace{(A_1 (\bar{R} + x_h n^{*T}) A_1^{-1})}_{G} p_i^* \quad (4-35)$$

$$p_i^* = \frac{1}{\alpha_i} \underbrace{(A_1 (R_n + x_{nh} n^T) A_1^{-1})}_{G_n} p_i \quad (4-36)$$

$$p_{rdi} = \alpha_{rdi} \underbrace{(A_2 (\bar{R}_{rd} + x_{hrd} n_r^{*T}) A_2^{-1})}_{G_{rd}} p_{ri}^* \quad (4-37)$$

$$p_{ri}^* = \alpha_{ri} \underbrace{(A_2 (R_r + x_{hr} n^{*T}) A_2^{-1})}_{G_r} p_{rdi}, \quad (4-38)$$

where $G(t), G_n(t), G_{rd}(t), G_r(t) \in \mathbb{R}^{3 \times 3}$ denote projective homographies. Sets of linear equations can be developed from (4-35)-(4-38) to determine the projective homographies up to a scalar multiple. Various techniques can be used (e.g., see [49, 65]) to decompose the Euclidean homographies, to obtain $\alpha_i(t), \alpha_{rdi}(t), \alpha_{ri}(t), x_h(t), x_{nh}(t), x_{hrd}(t), x_{hr}(t), \bar{R}(t), R_n(t), \bar{R}_{rd}(t), R_r(t), n^*(t), n_r^*, n(t)$. Given that the constant rotation matrix R_r^* is assumed to be known, the expressions for $\bar{R}_{rd}(t)$ and $R_r(t)$ in (4-13) can be used to determine $R_{rd}(t)$ and $R^*(t)$. Once $R^*(t)$ is determined, the expression for $\bar{R}(t)$ in (4-13) can be used to determine $R(t)$. Also, once $R_r^*, R^{*T}(t)$, and $R(t)$ have been determined, (4-1) can be used to determine $R'(t)$. Since $R_r(t), x_{hr}(t), \alpha_i(t), n^*(t), n_r^*, n(t), m_i^*(t)$, and $m_i(t)$ can be determined, the following relationship can be used to determine $m_i'(t)$:

$$m_i' = \frac{z_i}{z_i'} \left(R_r + x_{hr} \alpha_i \frac{n^{*T} m_i^*}{n^T m_i} n^T \right) m_i, \quad (4-39)$$

where the inverse of the ratio $\frac{z_i(t)}{z_i'(t)}$ can be determined as

$$\frac{z_i'}{z_i} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \left(R_r + x_{hr} \alpha_i \frac{n^{*T} m_i^*}{n^T m_i} n^T \right) m_i. \quad (4-40)$$

4.2.4 Control Objective

The control objective is for a controlled object (e.g., an UGV or an UAV) to track a desired trajectory that is determined by a sequence of images. This objective is based on the assumption that the control object is physically able to follow the desired image trajectory, that the linear and angular velocities of the camera are control inputs that can be independently controlled (i.e., unconstrained motion), and that the reference and desired cameras are calibrated (i.e., A_1 and A_2 are known). The control objective can be stated as the desire for the Euclidean feature points on π to track the corresponding feature points on π_d , which can be mathematically stated as the desire for $\bar{m}_i'(t) \rightarrow \bar{m}_{rdi}(t)$. Equivalently, the control objective can also be stated in terms of the rotation and translation of the object as the desire for $x'_{fr}(t) \rightarrow x_{frd}(t)$ and $R'(t) \rightarrow R_{rd}(t)$.

As stated previously, $R'(t)$ and $R_{rd}(t)$ can be computed by decomposing the projective homographies in (4-35)-(4-38) and using (4-1). Once these rotation matrices have been determined, a variety of parameterizations can be used to describe the rotation. The unit quaternion parameterization is used to describe the rotation in the subsequent problem formulation, control development, and stability analysis since the unit quaternion provides a globally nonsingular parameterization of the corresponding rotation matrices.

The unit quaternion is a four dimensional vector, which can be defined as

$$q \triangleq \begin{bmatrix} q_0 & q_v^T \end{bmatrix}^T \quad q_v \triangleq \begin{bmatrix} q_{v1} & q_{v2} & q_{v3} \end{bmatrix}^T, \quad (4-41)$$

where $q_0(t), q_{vi}(t) \in \mathbb{R} \forall i = 1, 2, 3$ satisfy the following nonlinear constraint

$$q^T q = 1. \quad (4-42)$$

Given the rotation matrices $R'(t)$ and $R_{rd}(t)$, the corresponding unit quaternions $q(t)$ and $q_d(t)$ can be calculated by using the numerically robust method presented in [31] and [66] based on the corresponding relationships

$$R' = (q_0^2 - q_v^T q_v) I_3 + 2q_v q_v^T + 2q_0 q_v^\times \quad (4-43)$$

$$R_{rd} = (q_{0d}^2 - q_{vd}^T q_{vd}) I_3 + 2q_{vd} q_{vd}^T + 2q_{0d} q_{vd}^\times \quad (4-44)$$

where I_3 is the 3×3 identity matrix, and the notation $q_v^\times(t)$ denotes a skew-symmetric form of the vector $q_v(t)$ as

$$q_v^\times = \begin{bmatrix} 0 & -q_{v3} & q_{v2} \\ q_{v3} & 0 & -q_{v1} \\ -q_{v2} & q_{v1} & 0 \end{bmatrix}, \quad \forall q_v = \begin{bmatrix} q_{v1} \\ q_{v2} \\ q_{v3} \end{bmatrix}. \quad (4-45)$$

To quantify the rotation error between the feature points on π and π_d , the multiplicative error between rotation matrices $R'(t)$ and $R_{rd}(t)$ is defined as

$$\tilde{R} = R'^T R_{rd} = (\tilde{q}_0^2 - \tilde{q}_v^T \tilde{q}_v) I_3 + 2\tilde{q}_v \tilde{q}_v^T - 2\tilde{q}_0 \tilde{q}_v^\times, \quad (4-46)$$

where the error quaternion $\tilde{q}(t) = (\tilde{q}_0(t), \tilde{q}_v^T(t))^T$ is defined as

$$\tilde{q} = \begin{bmatrix} \tilde{q}_0 \\ \tilde{q}_v \end{bmatrix} = \begin{bmatrix} q_0 q_{0d} + q_v^T q_{vd} \\ q_{0d} q_v - q_0 q_{vd} + q_v^\times q_{vd} \end{bmatrix}. \quad (4-47)$$

Since $\tilde{q}(t)$ is a unit quaternion, (4-46) can be used to quantify the rotation tracking objective as

$$\|\tilde{q}_v(t)\| \rightarrow 0 \implies \tilde{R}(t) \rightarrow I_3 \quad \text{as } t \rightarrow \infty. \quad (4-48)$$

The translation error, denoted by $e(t) \in \mathbb{R}^3$, is defined as

$$e = m_e - m_{ed} \quad (4-49)$$

where $m_e(t), m_{ed}(t) \in \mathbb{R}^3$ are defined as³

$$\begin{aligned} m_e &= \left[\frac{x'_1}{z'_1} \quad \frac{y'_1}{z'_1} \quad \ln\left(\frac{z'_1}{z_{r1}^*}\right) \right]^T \\ m_{ed} &= \left[\frac{x_{rd1}}{z_{rd1}} \quad \frac{y_{rd1}}{z_{rd1}} \quad \ln\left(\frac{z_{rd1}}{z_{r1}^*}\right) \right]^T. \end{aligned} \quad (4-50)$$

In (4-50), $\frac{z'_1(t)}{z_{r1}^*(t)}$ and $\frac{z_{rd1}(t)}{z_{r1}^*(t)}$ can be expressed in terms of known signals as

$$\begin{aligned} \frac{z'_1}{z_{r1}^*} &= \frac{z'_1 z_1 z_1^*}{z_1 z_1^* z_{r1}^*} = \frac{z'_1}{z_1} \frac{1}{\alpha_1} \alpha_{r1} \\ \frac{z_{rd1}}{z_{r1}^*} &= \frac{1}{\alpha_{rd1}}. \end{aligned}$$

Based on (4-48) and (4-49), the subsequent control development targets the following objectives:

$$\|\tilde{q}_v(t)\| \rightarrow 0 \quad \text{and} \quad \|e(t)\| \rightarrow 0 \quad \text{as } t \rightarrow \infty. \quad (4-51)$$

³ Any point O_i can be utilized in the subsequent development; however, to reduce the notational complexity, we have elected to select the image point O_1 , and hence, the subscript 1 is utilized in lieu of i in the subsequent development.

4.2.5 Control Development

4.2.5.1 Open-loop error system

Based on (4-46) and (4-47), the open-loop rotation error system can be developed as [67]

$$\dot{\tilde{q}} = \frac{1}{2} \begin{bmatrix} -\tilde{q}_v^T \\ \tilde{q}_0 I_3 + \tilde{q}_v^\times \end{bmatrix} (\omega_c - \tilde{R}\omega_{cd}), \quad (4-52)$$

where $\omega_{cd}(t)$ denotes the angular velocity of π_d expressed in \mathcal{F}_d that can be calculated as [67]

$$\omega_{cd} = 2(q_{0d}\dot{q}_{vd} - q_{vd}\dot{q}_{0d}) - 2q_{vd}^\times\dot{q}_{vd}, \quad (4-53)$$

where $(q_{0d}(t), q_{vd}^T(t))^T$, $(\dot{q}_{0d}(t), \dot{q}_{vd}^T(t))^T$ are assumed to be bounded; hence, $\omega_{cd}(t)$ is also bounded. The open-loop translation error system can be derived as

$$z_{r1}^* \dot{e} = \frac{z_{r1}^*}{z_1'} L_v' R' (v_c + \omega_c^\times s_1) - z_{r1}^* \dot{m}_{ed}, \quad (4-54)$$

where $v_c(t), \omega_c(t) \in \mathbb{R}^3$ denote the linear and angular velocity of π expressed in \mathcal{F} , respectively, and the auxiliary measurable term $L_v'(t) \in \mathbb{R}^{3 \times 3}$ is defined as

$$L_v' = \begin{bmatrix} 1 & 0 & -\frac{x_1'}{z_1'} \\ 0 & 1 & -\frac{y_1'}{z_1'} \\ 0 & 0 & 1 \end{bmatrix}.$$

4.2.5.2 Closed-loop error system

Based on the open-loop rotation error system in (4-52) and the subsequent Lyapunov-based stability analysis, the angular velocity controller is designed as

$$\omega_c = -K_\omega \tilde{q}_v + \tilde{R}\omega_{cd}, \quad (4-55)$$

where $K_\omega \in \mathbb{R}^{3 \times 3}$ denotes a diagonal matrix of positive constant control gains. From (4-52) and (4-55), the rotation closed-loop error system can be determined as

$$\begin{aligned}\dot{\tilde{q}}_0 &= \frac{1}{2} \tilde{q}_v^T K_\omega \tilde{q}_v \\ \dot{\tilde{q}}_v &= -\frac{1}{2} (\tilde{q}_0 I_3 + \tilde{q}_v^\times) K_\omega \tilde{q}_v = -\frac{1}{2} K_\omega \tilde{q}_0 \tilde{q}_v.\end{aligned}\tag{4-56}$$

Based on (4-54), the translation control input $v_c(t)$ is designed as

$$v_c = -\frac{z_1'}{z_{r1}^*} R'^T L_v'^{-1} (K_v e - \hat{z}_{r1}^* \dot{m}_{ed}) - \omega_c^\times s_1,\tag{4-57}$$

where $K_v \in \mathbb{R}^{3 \times 3}$ denotes a diagonal matrix of positive constant control gains. In (4-57), the parameter estimate $\hat{z}_{r1}^*(t) \in \mathbb{R}$ for the unknown constant z_{r1}^* is designed as

$$\dot{\hat{z}}_{r1}^* = -\gamma e^T \dot{m}_{ed},\tag{4-58}$$

where $\gamma \in \mathbb{R}$ denotes a positive constant adaptation gain. By using (4-54) and (4-57), the translation closed-loop error system is

$$z_{r1}^* \dot{e} = -K_v e - \tilde{z}_{r1}^* \dot{m}_{ed},\tag{4-59}$$

where $\tilde{z}_{r1}^*(t) \in \mathbb{R}$ denotes the parameter estimation error

$$\tilde{z}_{r1}^* = z_{r1}^* - \hat{z}_{r1}^*.\tag{4-60}$$

4.2.6 Stability Analysis

Theorem 1. *The controller given in (4-55) and (4-57), along with the adaptive update law in (4-58) ensures asymptotic tracking in the sense that*

$$\|\tilde{q}_v(t)\| \rightarrow 0, \quad \|e(t)\| \rightarrow 0, \quad \text{as } t \rightarrow \infty.\tag{4-61}$$

Proof. Let $V(t) \in \mathbb{R}$ denote the following differentiable non-negative function (i.e., a Lyapunov candidate):

$$V = \tilde{q}_v^T \tilde{q}_v + (1 - \tilde{q}_0)^2 + \frac{z_{r1}^*}{2} e^T e + \frac{1}{2\gamma} \tilde{z}_{r1}^{*2}. \quad (4-62)$$

The time-derivative of $V(t)$ can be determined as

$$\begin{aligned} \dot{V} &= -\tilde{q}_v^T K_\omega \tilde{q}_0 \tilde{q}_v - (1 - \tilde{q}_0) \tilde{q}_v^T K_\omega \tilde{q}_v - e^T K_v e \\ &\quad + e^T (-K_v e - \tilde{z}_{r1}^* \dot{m}_{ed}) + \tilde{z}_{r1}^* e^T \dot{m}_{ed} \\ &= -\tilde{q}_v^T (\tilde{q}_0 I_3 + (1 - \tilde{q}_0) I_3) K_\omega \tilde{q}_v - e^T K_v e \\ &= -\tilde{q}_v^T K_\omega \tilde{q}_v - e^T K_v e, \end{aligned} \quad (4-63)$$

where (4-56) and (4-58)-(4-60) were utilized. Based on (4-62) and (4-63), $e(t)$, $\tilde{q}_v(t)$, $\tilde{q}_0(t)$, $\tilde{z}_{r1}^*(t) \in \mathcal{L}_\infty$ and $e(t)$, $\tilde{q}_v(t) \in \mathcal{L}_2$. Since $\tilde{z}_{r1}^*(t) \in \mathcal{L}_\infty$, it is clear from (4-60) that $\hat{z}_{r1}^*(t) \in \mathcal{L}_\infty$.

Based on the fact that $e(t) \in \mathcal{L}_\infty$, (4-49) and (4-50) can be used to prove that $m_1'(t) \in \mathcal{L}_\infty$, and then $L_v'(t)$, $L_v'^{-1}(t) \in \mathcal{L}_\infty$. Based on the fact that $\tilde{q}_v(t) \in \mathcal{L}_\infty$ and $\omega_{cd}(t)$ is a bounded function, (4-55) can be used to conclude that $\omega_c(t) \in \mathcal{L}_\infty$. Since $\hat{z}_{r1}^*(t)$, $e(t)$, $m_1'(t)$, $L_v'(t)$, $L_v'^{-1}(t) \in \mathcal{L}_\infty$ and $\dot{m}_{ed}(t)$ is bounded, (4-57) can be utilized to prove that $v_c(t) \in \mathcal{L}_\infty$. From the previous results, (4-52)-(4-54) can be used to prove that $\dot{e}(t)$, $\dot{\tilde{q}}_v(t) \in \mathcal{L}_\infty$. Since $e(t)$, $\tilde{q}_v(t) \in \mathcal{L}_\infty \cap \mathcal{L}_2$, and $\dot{e}(t)$, $\dot{\tilde{q}}_v(t) \in \mathcal{L}_\infty$, we can utilize a corollary to Barbalat's Lemma [68] to conclude the result given in (4-61). \square

4.3 Cooperative Tracking Control of a Nonholonomic Unmanned Ground Vehicle

In the previous section, a visual servo tracking controller is developed for a moving six DOF object based on daisy-chained image feedback from a moving camera where a stationary reference camera was used to encode a desired video. The development in this section and our preliminary work in [43] extends the previous section by allowing the reference camera to also move. The example of a reference camera in the previous section

was a “stationary” satellite that was used to encode the desired trajectory. In this section, the desired trajectory could be encoded by a moving camera (e.g., attached to a moving satellite, a dirigible, or another UAV). In addition, instead of considering the general six DOF control object, the control object in this section is a nonholonomic constrained UGV. The control objective is for the UGV to track a desired trajectory determined by a sequence of prerecorded images from some moving overhead camera. An additional technical issue resolved in this section is the challenge of comparing the relative velocity between a moving camera and a moving UGV to the relative desired trajectory recorded by a moving camera.

4.3.1 Problem Scenario

Recent advances in image extraction/interpretation technology and advances in control theory have motivated results such as [11, 15–17, 21, 24, 27, 28, 30, 69, 70] and others, where camera-based vision systems are the sole sensor used for autonomous navigation of an UGV. See [28] for a detailed review of these and other related results. Typically these results are focused on the regulation result, and in all the results the targets are static with respect to the moving camera or the camera is stationary and recording images of the moving UGV. In contrast to these methods, the development in this section and our previous preliminary work in [43] is motivated by the problem when a moving camera is recording images of a moving UGV so a second UGV can track a desired image trajectory. A practical example application of this scenario is an airborne camera attached to a remote controlled aircraft that is used to determine a desired video of an UGV moving in a terrain, and then another moving camera (which does not have to follow the same trajectory as the previous camera) is used to relate and control the pose of a moving UGV with respect to the recorded video.

The scenario examined in this section is depicted in Fig. 4-2, where various coordinate frames are defined again as a means to develop the subsequent Euclidean reconstruction and control methods. In Fig. 4-2, a single camera that is navigating above

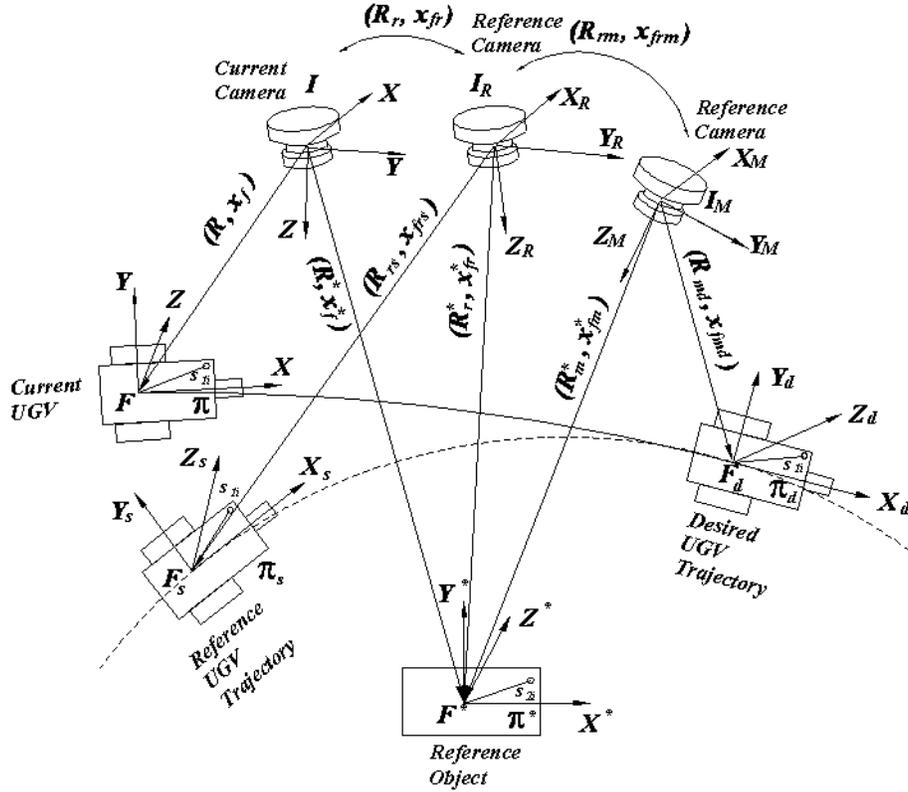


Figure 4-2. Geometric model for a moving camera, moving UGV, and stationary reference camera: A moving camera (coordinate frame \mathcal{I}_M) records the desired trajectory of an UGV (coordinate frame $\mathcal{F}_d(t)$) with respect to the stationary reference object \mathcal{F}^* while stationary coordinate frame \mathcal{F}_s represents a snapshot of an UGV along the desired trajectory taken by $\mathcal{I}_R = \mathcal{I}_M(t)|_{t=T}$. A moving camera (coordinate frame \mathcal{I}) views the current UGV (coordinate frame $\mathcal{F}(t)$) and the stationary reference object \mathcal{F}^* .

the planar motion of an UGV. The moving coordinate frame \mathcal{I} is attached to an overhead camera, which records an images for real-time tracking control. The moving coordinate frame \mathcal{I}_M is attached to the overhead camera that recorded the desired image sequence, and the fixed coordinate frame \mathcal{I}_R is some single snapshot of \mathcal{I}_M .

Geometric model for a moving camera (coordinate frame \mathcal{I}), moving UGV (coordinate frame \mathcal{F}) and stationary reference camera (coordinate frame \mathcal{I}_R)

The moving coordinate frame \mathcal{F} is attached to the UGV at the center of the rear wheel axis (for simplicity and without loss of generality). The UGV is represented in the camera image by four feature points that are coplanar and not collinear. The Euclidean

distance (i.e., $s_{1i} \in \mathbb{R}^3 \forall i = 1, 2, 3, 4$) from the origin of \mathcal{F} to one of the feature points is assumed to be known. A priori information (such as a known target in the initial FOV [32]) is sometimes used in vSLAM methods to establish scale. The plane defined by the UGV motion (i.e., the plane defined by the xy-axis of \mathcal{F}) and the UGV feature points is denoted by π . The linear velocity of the UGV along the x-axis of \mathcal{F} is denoted by $v_c(t) \in \mathbb{R}$, and the angular velocity $\omega_c(t) \in \mathbb{R}$ is about the z-axis of \mathcal{F} .

While viewing the feature points of the UGV, the camera is assumed to also view four additional coplanar and noncollinear feature points of a stationary reference object. The four additional feature points define the plane π^* in Fig. 4-2. The stationary coordinate frame \mathcal{F}^* is attached to the object where a distance from the origin of the coordinate frame to one of the feature points (i.e., $s_{2i} \in \mathbb{R}^3$) is assumed to be known. The plane π^* is assumed to be parallel to the plane π . When the camera is coincident with \mathcal{I}_R , a fixed (i.e., a single snapshot) reference pose of the UGV, denoted by \mathcal{F}_s , is assumed to be in the camera's FOV. A desired trajectory is defined by a prerecorded time-varying trajectory of \mathcal{F}_d that is assumed to be second-order differentiable where $v_{cd}(t), \omega_{cd}(t) \in \mathbb{R}$ denote the desired linear and angular velocity of \mathcal{F}_d , respectively. The feature points that define π^* are also assumed to be visible when the camera is a priori located coincident with the pose of the stationary coordinate frame \mathcal{I}_R and the time-varying coordinate frame \mathcal{I}_M . Based on these coordinate frame definitions, the problem considered in this section is to develop a kinematic controller for the object attached to \mathcal{F} so that the time-varying rotation and translation of \mathcal{F} converges to the desired time-varying rotation and translation of \mathcal{F}_d , where the motion of \mathcal{F} is determined from the time-varying overhead camera attached to \mathcal{I} .

4.3.2 Geometric Relationships

The rotation matrices and translation vectors in Table 4-1 (except the last line) are also valid for this section. Additional relationships between the various coordinate frames are summarized in Table 4-2. In Table 4-2, $R_{rs}, R_{md}(t), R_m^*(t), R_{rm}(t), R'_{md}(t) \in$

Table 4-2. Coordinate frame relationships for UGV tracking control.

Motion	Frames
R_{rs}, x_{frs}	\mathcal{F}_s to \mathcal{I}_R
$R_{md}(t), x_{fmd}(t)$	\mathcal{F}_d to \mathcal{I}_M
$R_m^*(t), x_{fm}^*(t)$	\mathcal{F}^* to \mathcal{I}_M
$R_{rm}(t), x_{frm}(t)$	\mathcal{I}_M to \mathcal{I}_R
$R'_{md}(t), x'_{frm}(t)$	\mathcal{F} to \mathcal{I}_R in \mathcal{I}_M

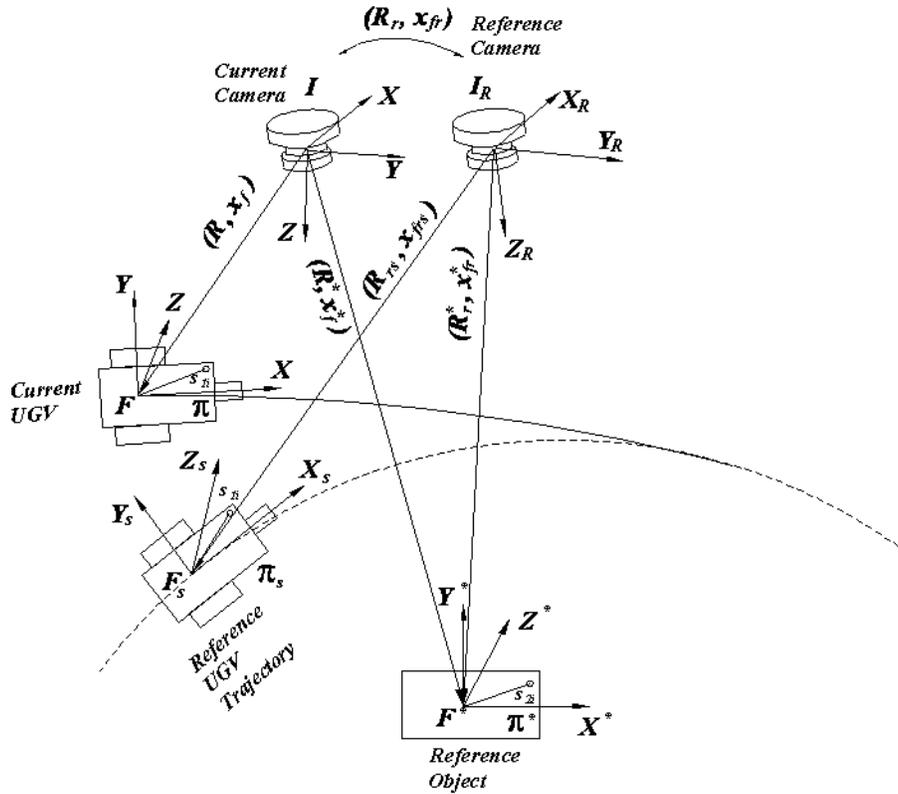


Figure 4-3. Geometric model showing a snapshot of an UGV along the desired trajectory (coordinate frame \mathcal{F}_s) taken by $\mathcal{I}_R = \mathcal{I}_M(t)|_{t=T}$. A current camera (coordinate frame \mathcal{I}) viewing the time-varying UGV (coordinate frame \mathcal{F}) while observing the set of feature points attached to \mathcal{F}^* .

$SO(3)$ denote rotation matrices, and $x_{frs}, x_{fmd}(t), x_{fm}^*(t), x_{frm}(t), x'_{frm}(t) \in \mathbb{R}^3$ denote translation vectors.

4.3.3 Euclidean Reconstruction

The coordinate frame representation in Fig. 4-2 can be separated into Figs. 4-3 and 4-4 to relate \mathcal{I} to \mathcal{I}_R and \mathcal{I}_R to \mathcal{I}_M , respectively. The coordinate frames in each figure

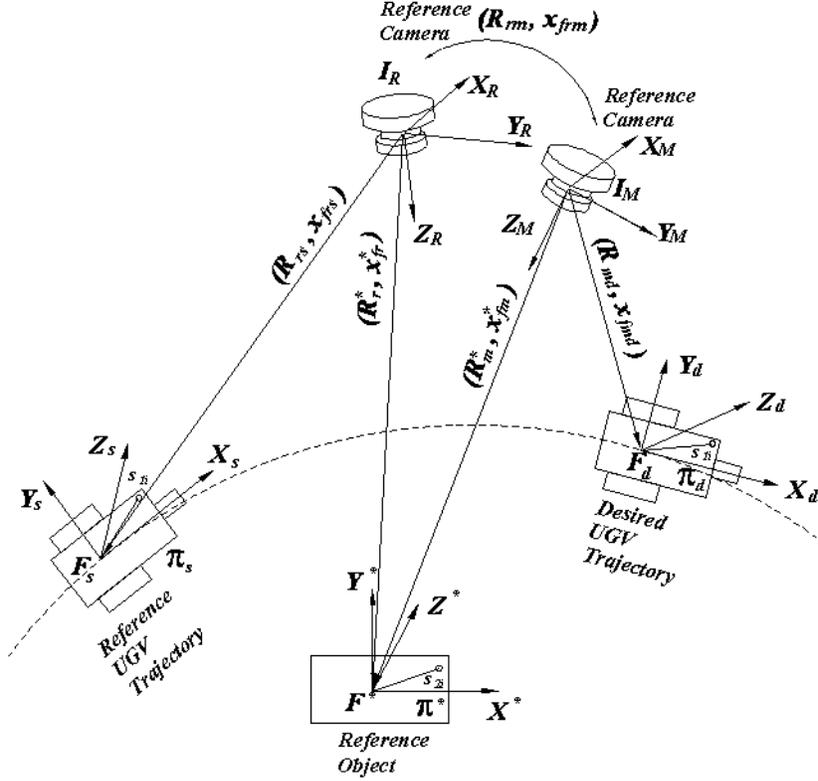


Figure 4-4. Geometric model showing a moving camera (coordinate frame \mathcal{I}_M) recording the desired trajectory of an UGV (coordinate frame $\mathcal{F}_d(t)$) with respect to the stationary reference object \mathcal{F}^* while stationary coordinate frame \mathcal{F}_s represents a snapshot of an UGV along the desired trajectory taken by $\mathcal{I}_R = \mathcal{I}_M(t)|_{t=T}$.

have the same relationships as in Fig. 4-1. Therefore, the same Euclidean reconstruction process as presented in Section 4.2.1-4.2.3 can be used twice to build the Euclidean relationships for this example.

To reconstruct the Euclidean relationship for the geometric model as shown in Fig. 4-3, let $\bar{m}_{rsi} \in \mathbb{R}^3$ denote the constant reference Euclidean coordinates of the feature points on π_s expressed in \mathcal{I}_R as

$$\bar{m}_{rsi} \triangleq \begin{bmatrix} x_{rsi} & y_{rsi} & z_{rsi} \end{bmatrix}^T,$$

and let $p_{rsi} \in \mathbb{R}^3$ represent the constant image-space coordinates of the feature points on π_s taken by the camera attached to \mathcal{I}_M when \mathcal{I}_M is coincident with \mathcal{I}_R

$$p_{rsi} \triangleq \begin{bmatrix} u_{rsi} & v_{rsi} & 1 \end{bmatrix}^T.$$

Following the development in Section 4.2.2 and 4.2.3, relationships can be obtained to determine the homographies and depth ratios as⁴

$$p_i = \alpha_i \underbrace{\left(A \left(\bar{R} + x_h n^{*T} \right) A^{-1} \right)}_G p_i^* \quad (4-64)$$

$$p_{rsi} = \alpha_{rsi} \underbrace{\left(A \left(\bar{R}_{rs} + x_{hrs} n_r^{*T} \right) A^{-1} \right)}_{G_{rs}} p_{rsi}^* \quad (4-65)$$

$$p_{ri}^* = \alpha_{ri} \underbrace{\left(A \left(R_r + x_{hr} n^{*T} \right) A^{-1} \right)}_{G_r} p_i^* \quad (4-66)$$

where

$$\begin{aligned} \alpha_i &= \frac{z_i^*}{z_i} & \alpha_{rsi} &= \frac{z_{rsi}^*}{z_{rsi}} & \alpha_{ri} &= \frac{z_i^*}{z_{ri}^*} \\ \bar{R} &= RR^{*T} & \bar{R}_{rs} &= R_{rs}R_r^{*T} & R_r &= R_r^*R_r^{*T}. \end{aligned} \quad (4-67)$$

Furthermore, the normalize Euclidean coordinates $m_i(t)$ can be related to $m_i'(t)$ as

$$m_i' = \frac{z_i}{z_i'} \left(R_r + x_{hr} \alpha_i \frac{n^{*T} m_i^*}{n^{*T} m_i} n^{*T} \right) m_i \quad (4-68)$$

$$\begin{aligned} \frac{z_i'}{z_i} &= \left[\begin{array}{ccc} 0 & 0 & 1 \end{array} \right] \left(R_r \right. \\ &\quad \left. + x_{hr} \alpha_i \frac{n^{*T} m_i^*}{n^{*T} m_i} n^{*T} \right) m_i. \end{aligned} \quad (4-69)$$

To reconstruct the Euclidean relationship for the geometric model as shown in Fig.

4-4, let $\bar{m}_{mdi}(t), \bar{m}_{mi}^*(t) \in \mathbb{R}^3$ denote the Euclidean coordinates of the feature points on π_d

⁴ To simplify the notations, the cameras are assumed to have the same calibration matrix A in the following development. The readers can refer to Section 4.2.1 for the deductions that the calibration matrices are different.

and π^* expressed in \mathcal{I}_M as

$$\begin{aligned}\bar{m}_{mdi}(t) &\triangleq \begin{bmatrix} x_{mdi}(t) & y_{mdi}(t) & z_{mdi}(t) \end{bmatrix}^T \\ \bar{m}_{mi}^*(t) &\triangleq \begin{bmatrix} x_{mi}^*(t) & y_{mi}^*(t) & z_{mi}^*(t) \end{bmatrix}^T,\end{aligned}$$

let $\bar{m}'_{mdi}(t) \in \mathbb{R}^3$ denote the desired Euclidean coordinates of the feature points on π_d expressed in \mathcal{I}_R as

$$\bar{m}'_{mdi}(t) \triangleq \begin{bmatrix} x'_{mdi}(t) & y'_{mdi}(t) & z'_{mdi}(t) \end{bmatrix}^T,$$

and let $p_{mdi}(t), p_{mi}^*(t) \in \mathbb{R}^3$ represent the image-space coordinates of the feature points on π_d and π^* captured by the camera attached to \mathcal{I}_M , respectively, as

$$p_{mdi} \triangleq \begin{bmatrix} u_{mdi} & v_{mdi} & 1 \end{bmatrix}^T \quad p_{mi}^* \triangleq \begin{bmatrix} u_{mi}^* & v_{mi}^* & 1 \end{bmatrix}^T.$$

The normalized coordinates of $\bar{m}'_{mdi}(t)$ and $\bar{m}_{mdi}(t)$, denoted as $m'_{mdi}(t), m_{mdi}(t) \in \mathbb{R}^3$, respectively, are defined as

$$m'_{mdi}(t) \triangleq \frac{\bar{m}'_{mdi}(t)}{z'_{mdi}(t)} \quad m_{mdi}(t) \triangleq \frac{\bar{m}_{mdi}(t)}{z_{mdi}(t)}. \quad (4-70)$$

Following the development in Section 4.2.2 and 4.2.3, relationships can be developed to compute the homographies and depth ratios as

$$p_{mdi} = \alpha_{mdi} \underbrace{\left(A \left(\bar{R}_{md} + x_{hmd} n_m^{*T} \right) A^{-1} \right)}_{G_{md}} p_{mi}^* \quad (4-71)$$

$$p_{ri}^* = \alpha_{rmi} \underbrace{\left(A \left(R_{rm} + x_{hrm} n_m^{*T} \right) A^{-1} \right)}_{G_{rm}} p_{mi}^*, \quad (4-72)$$

where

$$\begin{aligned}\alpha_{mdi} &= \frac{z_{mi}^*}{z_{mdi}} & \alpha_{rmi} &= \frac{z_{mi}^*}{z_{ri}^*} \\ \bar{R}_{md} &= R_{md} R_m^{*T} & R_{rm} &= R_r^* R_m^{*T}.\end{aligned} \quad (4-73)$$

The equations to relate $m_{mdi}(t)$ to $m'_{mdi}(t)$ can be developed as

$$m'_{mdi} = \frac{z_{mdi}}{z'_{mdi}} (R_{rm} \quad (4-74)$$

$$+ x_{hrm} \alpha_{mdi} \frac{n_m^{*T} m_{mi}^*}{n_m^{*T} m_{mdi}} n_m^{*T}) m_{mdi}$$

$$\frac{z'_{mdi}}{z_{mdi}} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} (R_{rm} \quad (4-75)$$

$$+ x_{hrm} \alpha_{mdi} \frac{n_m^{*T} m_{mi}^*}{n_m^{*T} m_{mdi}} n_m^{*T}) m_{mdi}.$$

In (4-64)-(4-72), $n^*(t)$, $n_m^*(t)$, and $n_r^* \in \mathbb{R}^3$ denote the constant unit normal to the planes π and π^* as expressed in \mathcal{I} , \mathcal{I}_M , and \mathcal{I}_R respectively, $x_h(t)$, $x_{hrs}(t)$, $x_{hr}(t)$, $x_{hmd}(t)$, $x_{hrm}(t) \in \mathbb{R}^3$ denote the corresponding scaled translation vectors, and $G(t)$, G_{rs} , $G_r(t)$, $G_{md}(t)$, $G_{rm}(t) \in R^{3 \times 3}$ denote projective homographies.

Sets of linear equations in (4-64)-(4-66), (4-71) and (4-72) can be used to determine and decompose homographies to obtain $\alpha_i(t)$, α_{rsi} , $\alpha_{mdi}(t)$, $\alpha_{ri}(t)$, $\alpha_{rmi}(t)$, $x_h(t)$, x_{hrs} , $x_{hr}(t)$, $x_{hmd}(t)$, $x_{hrm}(t)$, $\bar{R}(t)$, \bar{R}_{rs} , $R_r(t)$, $\bar{R}_{md}(t)$, and $R_{rm}(t)$. Given that the rotation matrix $R_r^*(t)$ is assumed to be known, the expressions for $\bar{R}_{rs}(t)$ and $R_r(t)$ in (4-67) can be used to determine $R_{rs}(t)$ and $R^*(t)$. Once $R^*(t)$ is determined, the expression for $\bar{R}(t)$ and $R_{rm}(t)$ in (4-67) and (4-73) can be used to determine $R(t)$ and $R_m^*(t)$. The rotation $R_m^*(t)$ can then be used to calculate $R_{md}(t)$ from the relationship for \bar{R}_{md} in (4-73). Based on the definitions for $R(t)$, $R^*(t)$, $R_{md}(t)$, $R_m^*(t)$, R_r^* , and R_{rs} provided in the previous development, the rotation from \mathcal{F} to \mathcal{F}_s and from \mathcal{F}_d to \mathcal{F}_s , denoted by $R_1(t)$, $R_{d1}(t) \in SO(3)$, respectively, are defined as

$$R_1(t) = R_{rs}^T R_r^* R^{*T}(t) R(t) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4-76)$$

$$\begin{aligned}
R_{d1}(t) &= R_{rs}^T R_r^* R_m^{*T}(t) R_{md}(t) \\
&= \begin{bmatrix} \cos \theta_d & \sin \theta_d & 0 \\ -\sin \theta_d & \cos \theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix},
\end{aligned} \tag{4-77}$$

where $\theta(t) \in \mathbb{R}$ denotes the right-handed rotation angle about the z-axis that aligns \mathcal{F} with \mathcal{F}_s , and $\theta_d(t) \in \mathbb{R}$ denotes the right-handed rotation angle about the z-axis that aligns \mathcal{F}_d with \mathcal{F}_s . From the definitions of $\theta(t)$ and $\theta_d(t)$, it is clear that

$$\dot{\theta} = \omega_c \quad \dot{\theta}_d = \omega_{cd} \tag{4-78}$$

where $\omega_c(t), \omega_{cd}(t) \in \mathbb{R}$ denote the desired angular velocities of \mathcal{F} and \mathcal{F}_d , respectively.

Based on the fact that $R(t), R^*(t), R_{md}(t), R_m^*(t), R_r^*$, and R_{rs} are known, it is clear from (4-76)-(4-78) that $\theta(t)$ and $\theta_d(t)$ are known signals that can be used in the subsequent control development. To facilitate the subsequent development, $\theta(t)$ and $\theta_d(t)$ are assumed to be confined to the following regions

$$-\pi < \theta(t) \leq \pi \quad -\pi < \theta_d(t) \leq \pi. \tag{4-79}$$

4.3.4 Control Objective

The objective is to develop a visual servo controller that ensures that the coordinate system \mathcal{F} tracks the time-varying trajectory of \mathcal{F}_d (i.e., $\bar{m}_i(t)$ measured in \mathcal{I} tracks $\bar{m}_{mdi}(t)$ measured in \mathcal{I}_M). To ensure that $\bar{m}_i(t)$ tracks $\bar{m}_{mdi}(t)$, the control objective can be stated by using the Euclidean reconstruction given in (4-64)-(4-72) as the desire for $\bar{m}'_1(t) \rightarrow \bar{m}'_{md1}(t)$. To quantify the control objective, translation and rotation tracking error, denoted by $e(t) \triangleq [e_1(t), e_2(t), e_3(t)]^T \in \mathbb{R}^3$, are defined as [67]

$$\begin{aligned}
e_1 &\triangleq \eta_1 - \eta_{d1} \\
e_2 &\triangleq \eta_2 - \eta_{d2} \\
e_3 &\triangleq \theta - \theta_d
\end{aligned} \tag{4-80}$$

where $\theta(t)$ and $\theta_d(t)$ are introduced in (4-76) and (4-77), respectively, and the auxiliary signals $\eta(t) \triangleq [\eta_1(t), \eta_2(t), \eta_3(t)]^T$, $\eta_d(t) \triangleq [\eta_{d1}(t), \eta_{d2}(t), \eta_{d3}(t)]^T \in \mathbb{R}^3$ are defined as

$$\begin{aligned}\eta(t) &\triangleq \frac{1}{z_{r1}^*} R^T(t) R^*(t) R_r^{*T} \bar{m}'_1(t) \\ \eta_d(t) &\triangleq \frac{1}{z_{r1}^*} R_{md}^T(t) R_m^*(t) R_r^{*T} \bar{m}'_{md1}(t).\end{aligned}\quad (4-81)$$

Also, the normal unit vector n_r^* is defined as [43]

$$\begin{aligned}n_r^* &= R_r^* R_r^{*T}(t) R(t) \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T \\ &= R_r^* R_m^{*T}(t) R_{md}(t) \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T.\end{aligned}\quad (4-82)$$

The expressions in (4-82) and (4-81) can be used to determine that

$$\eta_3 = \eta_{d3} = \frac{-d_r}{z_{r1}^*}.\quad (4-83)$$

The expressions in (4-64)-(4-75) can be used to rewrite $\eta(t)$ and $\eta_d(t)$ in terms of the measurable signals $\alpha_1(t)$, $\alpha_{r1}(t)$, $\alpha_{rm1}(t)$, $\alpha_{md1}(t)$, $R(t)$, $R^*(t)$, R_r^* , $R_{md}(t)$, $R_m^*(t)$, $p_1(t)$, and $p_{md1}(t)$ as

$$\begin{aligned}\eta(t) &= \frac{\alpha_{r1}}{\alpha_1} R^T(t) R^*(t) R_r^{*T} H_r' A^{-1} p_1 \\ \eta_d(t) &= \frac{\alpha_{rm1}}{\alpha_{md1}} R_{md}^T(t) R_m^*(t) R_r^{*T} H_{rm}' A^{-1} p_{md1}.\end{aligned}\quad (4-84)$$

Based on (4-80), (4-84), and the fact that $\theta(t)$ and $\theta_d(t)$ are measurable, it is clear that $e(t)$ is measurable. By examining (4-80)-(4-83), the control objective is achieved if $\|e(t)\| \rightarrow 0$. Specifically, if $e_3(t) \rightarrow 0$, then it is clear from (4-80) that $R_1(t) \rightarrow R_{d1}(t)$. If $e_1(t) \rightarrow 0$ and $e_2(t) \rightarrow 0$, then from (4-80) and (4-83) it is clear that $\eta(t) \rightarrow \eta_d(t)$. Given that $R_1(t) \rightarrow R_{d1}(t)$ and that $\eta(t) \rightarrow \eta_d(t)$, then (4-81) can be used to conclude that $m'_1(t) \rightarrow m'_{md1}(t)$. If $m'_1(t) \rightarrow m'_{md1}(t)$ and $R_1(t) \rightarrow R_{d1}(t)$, then the Euclidean relationships in the geometric model can be used to prove that $\bar{m}_i(t)$ measured in terms of $\mathcal{I} \rightarrow \bar{m}_{mdi}(t)$ measured in terms of \mathcal{I}_M .

4.3.5 Control Development

The open-loop error system can be obtained by taking the time derivative of (4-81) as

$$\dot{\eta} = \frac{v}{z_{r1}^*} + \left[\eta - \frac{s_{11}}{z_{r1}^*} \right]^\times \omega \quad (4-85)$$

where $v(t), \omega(t) \in \mathbb{R}^3$ denote the respective linear and angular velocity of an UGV expressed in \mathcal{F} as

$$v \triangleq \begin{bmatrix} v_c & 0 & 0 \end{bmatrix}^T \quad \omega \triangleq \begin{bmatrix} 0 & 0 & \omega_c \end{bmatrix}^T. \quad (4-86)$$

Without loss of generality, the location of the feature point s_1 is taken as the origin of \mathcal{F} , so that $s_{11} = [0, 0, 0]^T$. Then, based on (4-85) and (4-86), the error system can be further written as

$$\begin{aligned} \dot{\eta}_1 &= \frac{v_c}{z_{r1}^*} + \eta_2 \omega_c \\ \dot{\eta}_2 &= -\eta_1 \omega_c. \end{aligned} \quad (4-87)$$

Since the desired trajectory is assumed to be generated in accordance with UGV motion constraints, a similar expression to (4-87) can be developed as

$$\begin{aligned} \dot{\eta}_{d1} &= \frac{v_{cd}}{z_{r1}^*} + \eta_{d2} \omega_{cd} \\ \dot{\eta}_{d2} &= -\eta_{d1} \omega_{cd}. \end{aligned} \quad (4-88)$$

where $v_{cd}(t) \in \mathbb{R}$ denotes the desired linear velocity of \mathcal{F}_d . From (4-78), (4-80), (4-85) and (4-87), the open-loop error system can be obtained as

$$\begin{aligned} z_{r1}^* \dot{e}_1 &= v_c + z_{r1}^* (\eta_2 \omega_c - \dot{\eta}_{d1}) \\ \dot{e}_2 &= -\eta_1 \omega_c + \eta_{d1} \dot{\theta}_d \\ \dot{e}_3 &= \omega_c - \dot{\theta}_d. \end{aligned} \quad (4-89)$$

To facilitate the subsequent development, the auxiliary variable $\bar{e}_2(t) \in \mathbb{R}$ is defined as

$$\bar{e}_2 \triangleq e_2 + \eta_{d1} e_3. \quad (4-90)$$

After taking the time derivative of (4-90) and utilizing (4-89), the following expression is obtained:

$$\dot{\bar{e}}_2 = -e_1\omega_c + \dot{\eta}_{d1}e_3. \quad (4-91)$$

Based on (4-90), it is clear that if $\bar{e}_2(t), e_3(t) \rightarrow 0$, then $e_2(t) \rightarrow 0$. Based on this observation and the open-loop dynamics given in (4-91), the following control development is based on the desire to show that $e_1(t), \bar{e}_2(t), e_3(t)$ are asymptotically driven to zero.

Based on the open-loop error systems in (4-89) and (4-91), the linear and angular velocity control inputs for an UGV are designed as

$$v_c \triangleq -k_v e_1 + \bar{e}_2 \omega_c - \hat{z}_{r1}^* (\eta_2 \omega_c - \dot{\eta}_{d1}) \quad (4-92)$$

$$\omega_c \triangleq -k_\omega e_3 + \dot{\theta}_d - \dot{\eta}_{d1} \bar{e}_2 \quad (4-93)$$

where $k_v, k_\omega \in \mathbb{R}$ denote positive, constant control gains. In (4-92), the parameter update law $\hat{z}_{r1}^*(t) \in \mathbb{R}$ is generated by the differential equation

$$\dot{\hat{z}}_{r1}^* = \gamma_1 e_1 (\eta_2 \omega_c - \dot{\eta}_{d1}) \quad (4-94)$$

where $\gamma_1 \in \mathbb{R}$ is a positive, constant adaptation gain. After substituting the kinematic control signals designed in (4-92) and (4-93) into (4-89), the following closed-loop error systems are obtained:

$$\begin{aligned} z_{r1}^* \dot{e}_1 &= -k_v e_1 + \bar{e}_2 \omega_c + \tilde{z}_{r1}^* (\eta_2 \omega_c - \dot{\eta}_{d1}) \\ \dot{\bar{e}}_2 &= -e_1 \omega_c + \dot{\eta}_{d1} e_3 \\ \dot{e}_3 &= -k_\omega e_3 - \dot{\eta}_{d1} \bar{e}_2 \end{aligned} \quad (4-95)$$

where (4-91) was utilized, and the depth-related parameter estimation error, denoted by $\tilde{z}_{r1}^*(t) \in \mathbb{R}$, is defined as

$$\tilde{z}_{r1}^* \triangleq z_{r1}^* - \hat{z}_{r1}^*. \quad (4-96)$$

Theorem 2. *The control input designed in (4-92) and (4-93) along with the adaptive update law defined in (4-94) ensure asymptotic tracking for UGV in the sense that*

$$\|e(t)\| \rightarrow 0 \quad (4-97)$$

provided the time derivative of the desired trajectory satisfies the following condition

$$\dot{\eta}_{d1} \rightarrow 0. \quad (4-98)$$

Lyapunov-based analysis method and Barbalat's lemma can be used to prove the theorem 2 based on a Lyapunov function $V(t) \in \mathbb{R}$ defined as [43]

$$V \triangleq \frac{1}{2}z_{r1}^*e_1^2 + \frac{1}{2}\bar{e}_2^2 + \frac{1}{2}e_3^2 + \frac{1}{2\gamma_1}z_{r1}^{*2} \dots \quad (4-99)$$

4.4 Simultaneous Tracking, Localization and Mapping

For vision-based autonomous systems applications (e.g., tracking, localization and mapping), the given reference object can leave the camera's FOV while another reference object enters the FOV. In comparison to the single reference object problem presented in Section 4.3, multiple reference objects are taken into consideration in this section. The daisy-chaining method is further developed to achieve asymptotic tracking of the UGV by mapping each reference object to a global coordinate system. Moreover, the time-varying Euclidean position of the UGV and the stationary position of the reference objects can be localized with respect to the global coordinate system. In addition to achieving the visual servo tracking and localization objectives, the developed method generates data for the SLAM problem.

4.4.1 Problem Scenario

The geometric model in this section is the same as in Section 4.3, except that multiple reference objects are taken into consideration. While viewing the feature points of the UGV, the camera is assumed to also view four additional coplanar and noncollinear

Table 4-3. Coordinate frame relationships for multi-reference UGV tracking control.

Motion	Frames
$R_j^*(t), x_{fj}^*(t)$	\mathcal{F}_j^* to \mathcal{I} in \mathcal{I}
$R_{rj}^*(t), x_{frj}^*$	\mathcal{F}_j^* to \mathcal{I}_R in \mathcal{I}_R
$R_{mj}^*(t), x_{fmj}^*(t)$	\mathcal{F}_j^* to \mathcal{I}_M in \mathcal{I}_M

feature points of a stationary reference object, such that at any instant of time along the camera motion trajectory at least one such reference object is in the FOV. The four additional feature points define the plane π_j^* in Fig. 4-5. The stationary coordinate frame \mathcal{F}_j^* ($j = 1, 2, \dots, k$) is attached to the object where distance from the origin of the coordinate frame to one of the feature points is assumed to be known, i.e., $s_{2ji} \in \mathbb{R}^3 \forall i = 1, 2, 3, 4$. The plane π_j^* is assumed to be parallel to the plane π . The feature points that define π_1^* , corresponding to a reference object \mathcal{F}_1^* (i.e., \mathcal{F}_j^* corresponding to $j = 1$), are also assumed to be visible when the camera is a priori located coincident with the pose of the stationary coordinate frame \mathcal{I}_R . The fixed coordinate frame \mathcal{I}_R is a snapshot of \mathcal{I}_M at the time instant that the first reference object π_1^* is visible to the reference camera. The reference object π_1^* is visible to \mathcal{I}_R , but the other reference objects $\pi_j^*(j > 1)$ are not.

4.4.2 Geometric Relationships

In addition to the notations in Tables 4-1 and 4-2, more relationships between the various coordinate frames are summarized in Table 4-3. In Table 4-3, $R_j^*(t)$, $R_{rj}^*(t)$, $R_{mj}^*(t) \in SO(3)$ denote rotation matrices and $x_{fj}^*(t)$, x_{frj}^* , $x_{fmj}^*(t) \in \mathbb{R}^3$ denote translation vectors.

4.4.3 Euclidean Reconstruction

The Euclidean reconstruction for the geometric model in Fig. 4-5 can be separated into three cases. Case 1: a single reference object π_1^* is within the reference camera's FOV and therefore π_1^* is used as the reference object. Case 2: two reference objects (e.g., π_1^* and π_2^*) are within the camera's FOV, and the reference object in use is going to be switched from one to the other (e.g., from π_1^* to π_2^*). Case 3: π_j^* ($j \geq 2$) is used as the reference object.

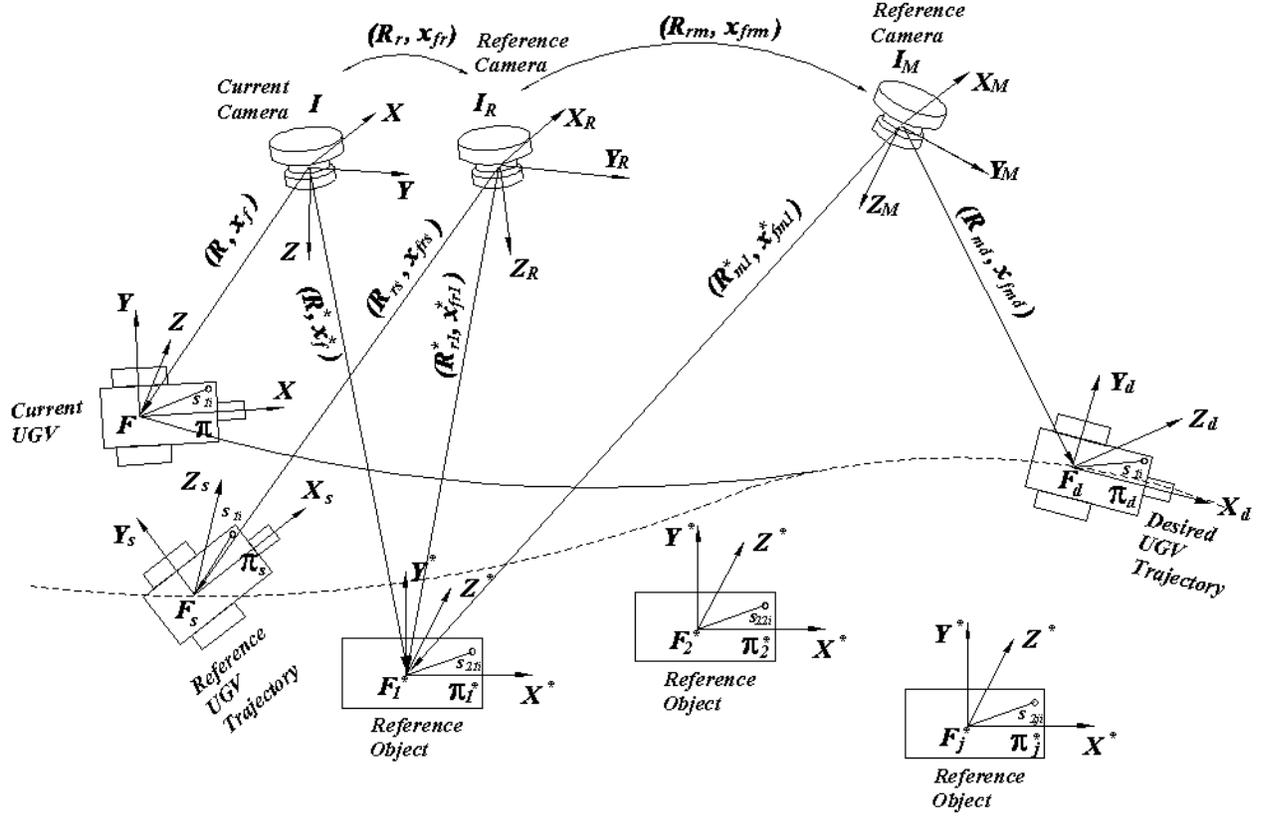


Figure 4-5. Geometric model for a moving camera, moving UGV, and stationary reference camera: A moving camera (coordinate frame \mathcal{I}_M) records the desired trajectory of an UGV (coordinate frame $\mathcal{F}_d(t)$) with respect to the stationary reference object \mathcal{F}_1^* while stationary coordinate frame \mathcal{F}_s represents a snapshot of an UGV along the desired trajectory taken by $\mathcal{I}_R = \mathcal{I}_M(t)|_{t=T}$. A moving camera (coordinate frame \mathcal{I}) views the current UGV (coordinate frame $\mathcal{F}(t)$) and the stationary reference object \mathcal{F}_j^* .

Let $\bar{m}_{mji}^*(t), \bar{m}'_{rji} \in \mathbb{R}^3$ denote the Euclidean coordinates of the feature points on π_j^* expressed in \mathcal{I}_M and \mathcal{I}_R , respectively, as

$$\bar{m}_{mji}^*(t) \triangleq \begin{bmatrix} x_{mji}^*(t) & y_{mji}^*(t) & z_{mji}^*(t) \end{bmatrix}^T$$

$$\bar{m}'_{rji} \triangleq \begin{bmatrix} x'_{rji} & y'_{rji} & z'_{rji} \end{bmatrix}^T.$$

Since the feature point plane π_1^* is visible to the reference camera when \mathcal{I}_M is coincident with \mathcal{I}_R , $\bar{m}'_{r1i} \triangleq \begin{bmatrix} x'_{r1i} & y'_{r1i} & z'_{r1i} \end{bmatrix}^T$ can also be written as $\bar{m}_{r1i}^* \triangleq \begin{bmatrix} x_{r1i}^* & y_{r1i}^* & z_{r1i}^* \end{bmatrix}^T$.

Let $p_{mji}^*(t), p_{rji}'^* \in \mathbb{R}^3$ represent the image-space coordinates of the feature points on π_j^* captured by the reference camera attached to \mathcal{I}_M and \mathcal{I}_R , respectively, as

$$\begin{aligned} p_{mji}^*(t) &\triangleq \begin{bmatrix} u_{mji}^*(t) & v_{mji}^*(t) & 1 \end{bmatrix}^T \\ p_{rji}'^* &\triangleq \begin{bmatrix} u_{rji}'^* & v_{rji}'^* & 1 \end{bmatrix}^T. \end{aligned}$$

When $j = 1$, $p_{r1i}'^* \triangleq \begin{bmatrix} u_{r1i}'^* & v_{r1i}'^* & 1 \end{bmatrix}^T$ can be written as $p_{r1i}^* \triangleq \begin{bmatrix} u_{r1i}^* & v_{r1i}^* & 1 \end{bmatrix}^T$, which is measurable. When $j > 1$, $p_{rji}'^*$ can not be measured directly. It needs to be computed based on the corresponding normalized coordinates obtained from the daisy-chaining multi-view geometry. The normalized coordinates of $\bar{m}_{mji}^*(t)$ and $\bar{m}_{rji}'^*$, denoted as $m_{mji}^*(t), m_{rji}'^* \in \mathbb{R}^3$, respectively, are defined as

$$m_{mji}^*(t) \triangleq \frac{m_{mji}^*(t)}{z_{mji}^*(t)} \quad m_{rji}'^* \triangleq \frac{\bar{m}_{rji}'^*}{z_{rji}'^*}.$$

For the first case, the Euclidean reconstruction is exactly the same as that in Section 4.3. For the second case, consider the feature point planes π_1^* and π_2^* as an example. Similar to the Euclidean reconstruction development in Section 4.3.3, relationships can be obtained to determine the homographies and depth ratios among the coordinate frames $\mathcal{F}_1^*, \mathcal{F}_2^*, \mathcal{I}_R$, and \mathcal{I}_M as

$$p_{m2i}^* = \alpha_{21i} \underbrace{\left(A \left(R_{21} + x_{h21} n_{m1}^{*T} \right) A^{-1} \right)}_{G_{21}} p_{m1i}^* \quad (4-100)$$

$$p_{r1i}^* = \alpha_{rm1i} \underbrace{\left(A \left(R_{rm1} + x_{hrm1} n_{m1}^{*T} \right) A^{-1} \right)}_{G_{rm1}} p_{m1i}^* \quad (4-101)$$

where

$$\alpha_{21i} = \frac{z_{m1i}^*}{z_{m2i}^*} \quad \alpha_{rm1i} = \frac{z_{m1i}^*}{z_{r1i}^*} \quad (4-102)$$

$$R_{21} = R_{m2}^* R_{m1}^{*T} \quad R_{rm1} = R_{r1}^* R_{m1}^{*T}. \quad (4-103)$$

The equations to relate $m'_{r2i}(t)$ to $m^*_{r1i}(t)$ can be developed as

$$m'_{r2i} = \frac{z^*_{r1i}}{z^*_{r2i}} \left(R_{21} + x_{h21} \alpha_{rm1i} \frac{n_{m1}^{*T} m_{m1i}^* n_{r1}^{*T}}{n_{r1}^{*T} m_{r1i}^*} \right) m^*_{r1i} \quad (4-104)$$

$$\frac{z'_{r2i}}{z^*_{r1i}} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \left(R_{21} + x_{h21} \alpha_{rm1i} \frac{n_{m1}^{*T} m_{m1i}^* n_{r1}^{*T}}{n_{r1}^{*T} m_{r1i}^*} \right) m^*_{r1i}. \quad (4-105)$$

In (4-100)-(4-105), $n_{m1}^*(t)$, and $n_{r1}^* \in \mathbb{R}^3$ denote the unit normal to the planes π_1^* expressed in \mathcal{I}_M and \mathcal{I}_R , respectively, $x_{h21}(t)$, $x_{hrm1}(t) \in \mathbb{R}^3$ denote the corresponding scaled translation vectors, and $G_{21}(t)$, $G_{rm1}(t) \in \mathbb{R}^{3 \times 3}$ denote projective homographies. Linear equations in (4-100) and (4-101) can be used to determine and decompose homographies to obtain $\alpha_{21i}(t)$, α_{rm1i} , $x_{h21}(t)$, x_{hrm1} , $R_{21}(t)$, R_{rm1} . From $m'_{r2i}(t)$ in (4-104), the virtual pixel coordinates $p'_{r2i}(t)$ can be computed. Based on (4-104) and (4-105), the Euclidean coordinates of the feature points on π_2^* can be related to fixed coordinate frame \mathcal{I}_R . Following the same idea as used to relate π_2^* and π_1^* , π_j^* can be related to π_{j-1}^* ($j = 3, \dots, k$) based on the following projective homographies:

$$p_{mji}^* = \alpha_{j(j-1)i} G_{j(j-1)} p_{m(j-1)i}^* \quad (4-106)$$

$$p_{r(j-1)i}^* = \alpha_{rm(j-1)i} G_{rm(j-1)} p_{m(j-1)i}^* \quad (4-107)$$

where $G_{j(j-1)}$ and $G_{rm(j-1)}$ are respectively defined as

$$G_{j(j-1)} = A \left(R_{j(j-1)} + x_{hj(j-1)} n_{m(j-1)}^{*T} \right) A^{-1}$$

$$G_{rm(j-1)} = A \left(R_{rm(j-1)} + x_{hrm(j-1)} n_{m(j-1)}^{*T} \right) A^{-1}$$

and

$$\alpha_{j(j-1)i} = \frac{z_{m(j-1)i}^*}{z_{mji}^*} \quad \alpha_{rm(j-1)i} = \frac{z_{m(j-1)i}^*}{z_{r(j-1)i}^*} \quad (4-108)$$

$$R_{j(j-1)} = R_{mj}^* R_{m(j-1)}^{*T} \quad (4-109)$$

$$R_{rm(j-1)} = R_{r(j-1)}^* R_{m(j-1)}^{*T}. \quad (4-110)$$

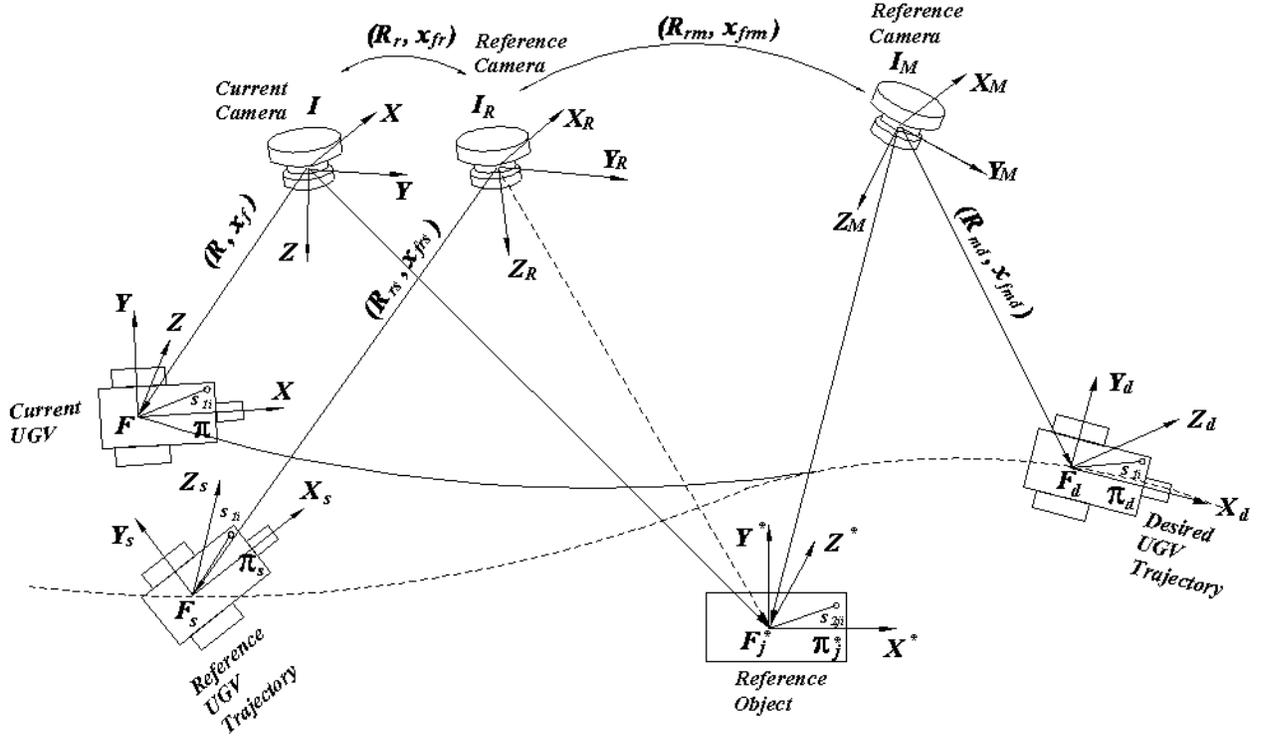


Figure 4-6. A simplified equivalent model showing a moving camera (coordinate frame \mathcal{I}) observing the current UGV (coordinate frame $\mathcal{F}(t)$) and the stationary reference object \mathcal{F}_j^* and the pose of \mathcal{F}_j^* is expressed in terms of \mathcal{I}_R .

Relationships can also be developed in terms of the normalized Euclidean coordinates as

$$m'_{rji} = \frac{z_{r(j-1)i}^*}{z_{rji}^{I*}} \left(R_{j(j-1)} + x_{hj(j-1)} \alpha_{rm(j-1)i} \right. \quad (4-111)$$

$$\left. \cdot \frac{n_{m(j-1)}^{*T} m_{m(j-1)i}^*}{n_{r(j-1)}^{*T} m_{r(j-1)i}^*} n_{r(j-1)}^{*T} \right) m_{r(j-1)i}^*$$

$$\frac{z_{rji}^{I*}}{z_{r(j-1)i}^*} = \left[\begin{array}{ccc} 0 & 0 & 1 \end{array} \right] \left(R_{j(j-1)} + x_{hj(j-1)} \alpha_{rm(j-1)i} \right. \quad (4-112)$$

$$\left. \cdot \frac{n_{m(j-1)}^{*T} m_{m(j-1)i}^*}{n_{r(j-1)}^{*T} m_{r(j-1)i}^*} n_{r(j-1)}^{*T} \right) m_{r(j-1)i}^*.$$

Recursively, from (4-100)-(4-112), $m'_{rji}(t)$ can be related to the known normalized Euclidean coordinate m_{r1i}^* . For the third case, the geometric model can be simplified as depicted in Fig. 4-6.

Once $m'_{rji}{}^*(t)$ is computed based on the deductions in case 2, the geometric model in Fig. 4-6 is equivalent to that in Fig. 4-2. Therefore, the Euclidean reconstruction in Section 4.3 can be used to build the Euclidean relationships among different coordinate frames.

4.4.4 Tracking and Mapping

The tracking control design is the same as that in Section 4.3, once the Euclidean relationship between \mathcal{F} and \mathcal{F}_d is obtained based on the Euclidean reconstruction analysis as shown in Section 4.4.3. The time-varying Euclidean position of the UGV and the stationary position of the reference objects can be localized with respect to the global coordinate system \mathcal{I}_R . Using the known geometric length s_{21i} and a unit normal n_{r1}^* (i.e., the normal to π_1^* expressed in \mathcal{I}_R), the geometric reconstruction method in [52, 63, 64] can be utilized to obtain $\bar{m}_{r1i}^*(t)$. Based on the computed $\bar{m}_{r1i}^*(t)$, (4-105) can be used to find z_{r2i}^* , and then (4-104) can be used to find $\bar{m}'_{r2i}{}^*(t)$. Recursively, based on (4-106)-(4-112), the Euclidean coordinates of the other reference objects denoted as $\bar{m}'_{rji}{}^*(t)$ ($j = 3, \dots, k$) can be computed. Similarly, using the known geometric length s_{1i} and a unit normal $n(t)$ (i.e., the normal to π expressed in \mathcal{I}), the geometric reconstruction method in [63] can also be utilized to obtain $\bar{m}_i(t)$. Based on the computed $\bar{m}_i(t)$, (4-68) and (4-69) can be used to find $\bar{m}'_i(t)$.

4.4.5 Simulation Results

A numerical simulation was performed to illustrate the localization and mapping performance given the controller in (4-92), (4-93), and the adaptive update law in (4-94). The simulation scenario is shown in Fig. 4-7, such that the pose of current UGV $\mathcal{F}(t)$ is estimated with respect to three stationary reference objects \mathcal{F}_1^* , \mathcal{F}_2^* , and \mathcal{F}_3^* while tracking the desired trajectory \mathcal{F}_d encoded as a sequence of images. The origins of the coordinate frames \mathcal{F} , \mathcal{F}_1^* , \mathcal{F}_2^* , \mathcal{F}_3^* , and \mathcal{F}_d , and the four coplanar feature points on the planes π , π_1^* , π_2^* , π_3^* , and π_d are chosen such that the Euclidean coordinates of the feature points in \mathcal{F} , \mathcal{F}_1^* , \mathcal{F}_2^* , \mathcal{F}_3^* , and \mathcal{F}_d are given by s_i , s_{1i} , s_{2i} , s_{3i} , and s_i (where $i = 1, 2, 3, 4$), respectively.

The initial pose of current UGV $\mathcal{F}(0) = \mathcal{F}(t)|_{t=0}$, stationary reference objects $\mathcal{F}_j^*, j = 1, 2, 3$, and the initial position of the time-varying desired UGV $\mathcal{F}_d(0) = \mathcal{F}_d|_{t=0}$ were considered as

$$\mathcal{F}(0) = \begin{bmatrix} \cos(27) & -\sin(27) & 0 & -2.50 \\ \sin(27) & \cos(27) & 0 & 1.20 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathcal{F}_1^* = \begin{bmatrix} \cos(45) & -\sin(45) & 0 & -1.30 \\ \sin(45) & \cos(45) & 0 & 0.80 \\ 0 & 0 & 1 & -0.50 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathcal{F}_2^* = \begin{bmatrix} \cos(35) & -\sin(35) & 0 & -1.25 \\ \sin(35) & \cos(35) & 0 & 1.90 \\ 0 & 0 & 1 & -1.50 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathcal{F}_3^* = \begin{bmatrix} \cos(25) & -\sin(25) & 0 & 0.50 \\ \sin(25) & \cos(25) & 0 & 2.75 \\ 0 & 0 & 1 & -2.00 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathcal{F}_d(0) = \begin{bmatrix} \cos(60) & -\sin(60) & 0 & -2.10 \\ \sin(60) & \cos(60) & 0 & 0.30 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The control gains in (4-92) and (4-93) and the adaptation gain in (4-94) were selected as

$$k_\omega = 1.0038 \quad k_v = 3.88 \quad \gamma_1 = 10.$$

The performance of the visual servo tracking controller is shown in Fig. 4-7, which shows the Euclidean space trajectory of the feature points attached to the planes π and π_d , taken by \mathcal{I} and \mathcal{I}_M , respectively and the time-varying trajectory of the current and reference camera, \mathcal{I} and \mathcal{I}_M , respectively. From Fig. 4-7, it can be seen that the current trajectory corresponding to the time-varying UGV $\mathcal{F}(t)$ is indistinguishable from the desired trajectory corresponding to the time-varying UGV $\mathcal{F}_d(t)$ due to relatively low tracking error (see Fig. 4-8). The resulting tracking errors are plotted in Fig. 4-8, which asymptotically approach zero. The linear and angular velocity control inputs are shown

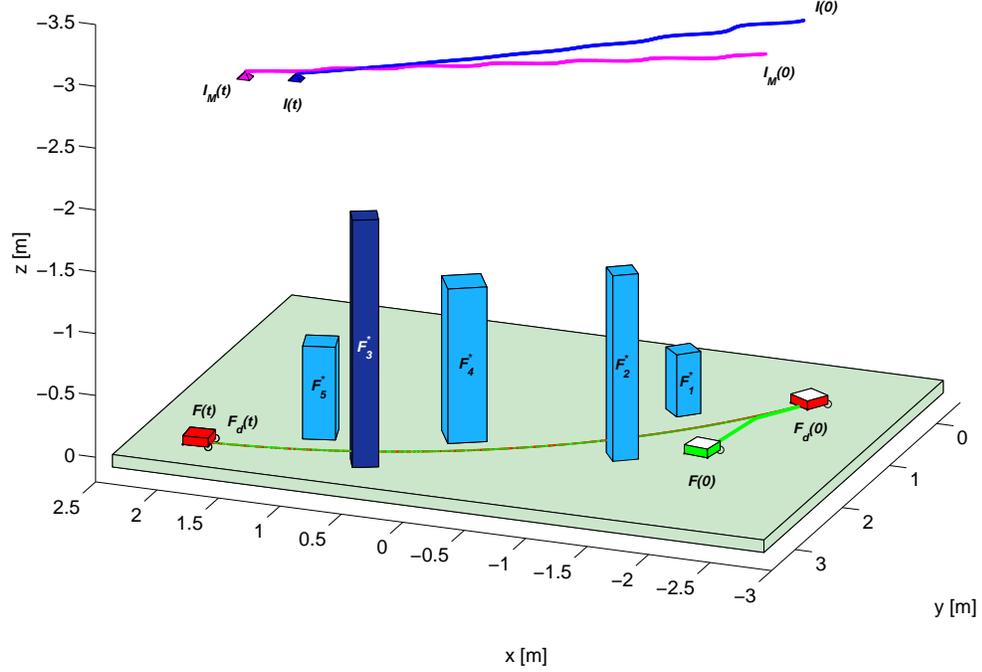


Figure 4-7. Euclidean space trajectory of the feature points attached to the current (i.e. $\mathcal{F}(t)$) and desired (i.e. $\mathcal{F}_d(t)$) UGV taken by \mathcal{I} and \mathcal{I}_M , respectively and the time-varying trajectory of the current and reference camera, \mathcal{I} and \mathcal{I}_M , respectively. $\mathcal{F}(0)$ denotes the initial position of the current UGV, $\mathcal{F}(t)$ denotes the time-varying position of the current UGV, $\mathcal{F}_d(0)$ denotes the initial position of the desired UGV, $\mathcal{I}(0)$ denotes the initial position of the current camera, $\mathcal{I}(t)$ denotes the time-varying position of the current camera, $\mathcal{I}_M(0)$ denotes the initial position of the time-varying reference camera, $\mathcal{I}_M(t)$ denotes the time-varying position of the time-varying reference camera, and \mathcal{F}_1^* , \mathcal{F}_2^* , and \mathcal{F}_3^* denote the position of the stationary reference objects.

in Fig. 4-9. Figs. 4-10 and 4-11 show the regulation results in presence of an additive white Gaussian noise of standard deviation $\sigma = 0.1$ pixels. Fig. 4-12 shows the results of localization of the current UGV attached to $\mathcal{F}(t)$ and mapping of reference targets attached to \mathcal{F}_1^* , \mathcal{F}_2^* , and \mathcal{F}_3^* expressed in constant reference frame \mathcal{I}_R .

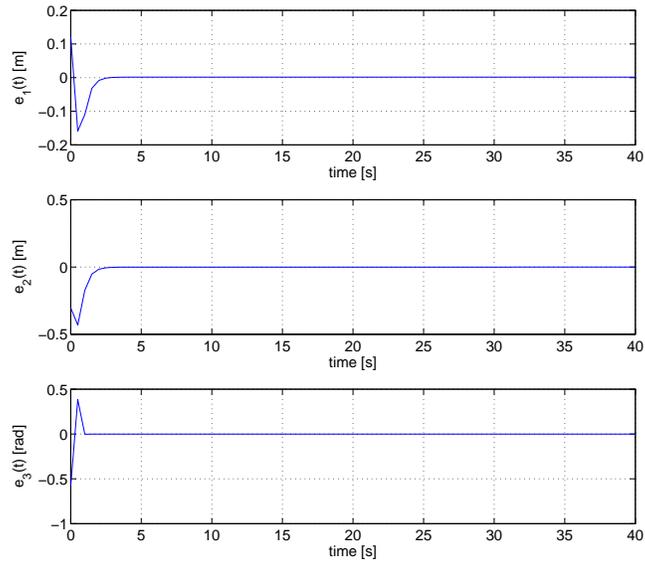


Figure 4-8. Linear (i.e., $e_1(t)$ and $e_2(t)$) and angular (i.e., $e_3(t)$) tracking error.

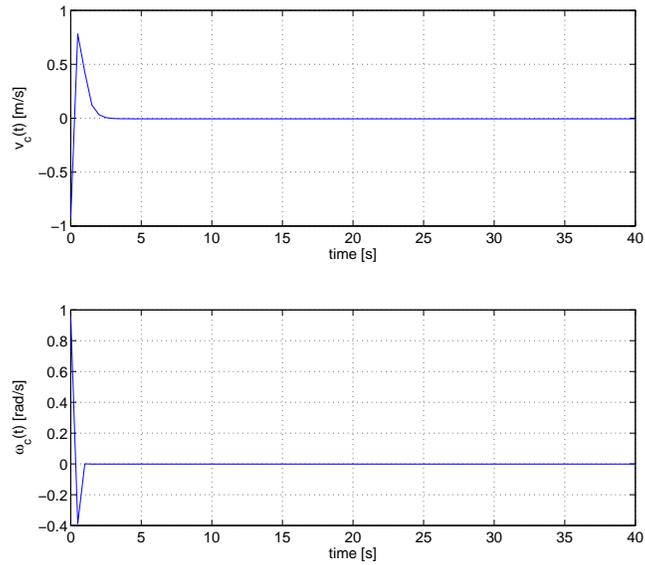


Figure 4-9. Linear (i.e., $v_c(t)$) and angular (i.e., $\omega_c(t)$) velocity control inputs.

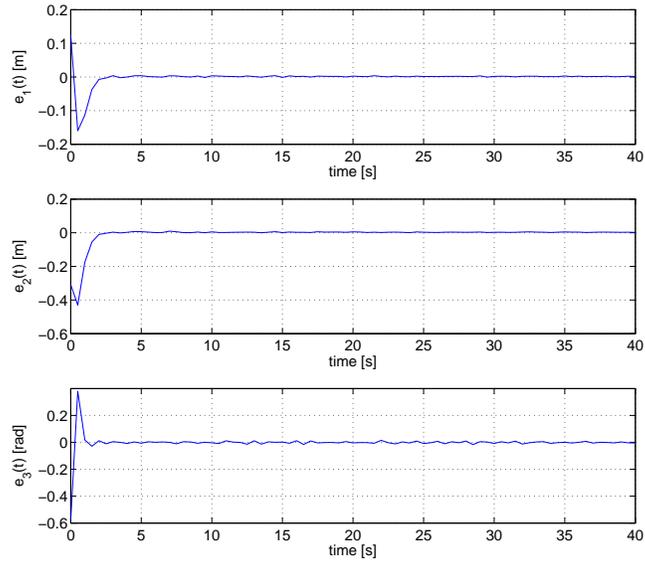


Figure 4-10. Linear (i.e., $e_1(t)$ and $e_2(t)$) and angular (i.e., $e_3(t)$) tracking error in presence of an additive white Gaussian image noise.

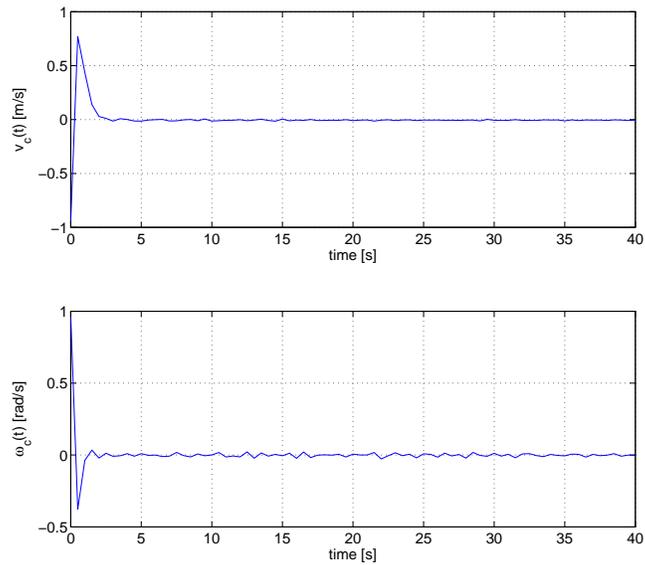


Figure 4-11. Linear (i.e., $v_c(t)$) and angular (i.e., $\omega_c(t)$) velocity control inputs in presence of an additive white Gaussian image noise.

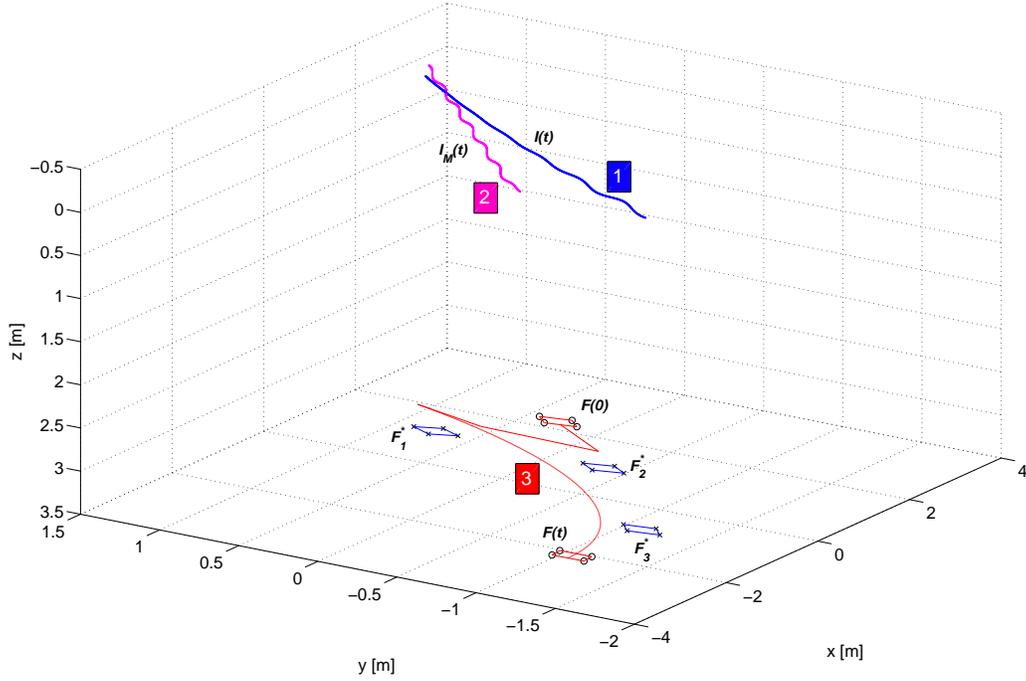


Figure 4-12. Results of localization of the current UGV attached to $\mathcal{F}(t)$ and mapping of reference targets attached to \mathcal{F}_1^* , \mathcal{F}_2^* , and \mathcal{F}_3^* expressed in constant reference frame \mathcal{I}_R . Specifically, trajectory (1) shows the time-varying pose of the moving camera attached to $\mathcal{I}(t)$, trajectory (2) shows the time-varying pose of the moving camera attached to $\mathcal{I}_M(t)$, and trajectory (3) shows the time-varying pose of the current UGV attached to $\mathcal{F}(t)$ measured in the stationary reference camera frame \mathcal{I}_R . $\mathcal{F}(0)$ denotes the initial position of the current UGV and \mathcal{F}_1^* , \mathcal{F}_2^* , and \mathcal{F}_3^* denote the position of the stationary reference objects.

4.5 Error Propagation in Daisy-Chaining

The daisy-chaining based control scheme developed in Section 4.4 is based on estimating the pose of an UGV with respect to stationary reference objects \mathcal{F}_i^* where $i = 1, \dots, n$ using a moving monocular camera. Since the stationary reference object can leave the camera FOV and a new reference object enters the FOV, it is necessary to determine the pose of the new reference object with respect to the receding object in order to provide the pose information of a moving agent such as an UGV or the camera itself. For example, if \mathcal{F}_j^* is leaving the FOV and \mathcal{F}_{j+1}^* is entering the FOV, then a homography

relationship is obtained between \mathcal{F}_j^* and \mathcal{F}_{j+1}^* . The errors in estimating the pose of \mathcal{F}_{j+1}^* would be propagated to subsequent reference objects \mathcal{F}_{j+n}^* where $n = 2, \dots, m$. The control law is established by comparing the current pose of an UGV with the desired pose measured in the stationary reference object. Therefore, the pose measurement error propagated through multiple stationary reference objects can result in erroneous control input and possibly lead to system instability.

In this section, the error propagation in daisy-chaining based pose estimation is analysed by performing a numerical simulation. The simulation scenario consists of a moving airborne monocular camera (e.g., a camera attached to an UAV) with the coordinate frame $\mathcal{I}(t)$ travelling at an altitude of $100m$ and capturing the images of the stationary reference objects \mathcal{F}_i^* where $i = 1, \dots, 8$ as shown in Fig. 4-13. The camera is assumed to traverse a circular trajectory with a ground speed of $10m/s$. At time $t = 0$, pose of the camera is assumed to be known with respect to the world coordinate frame and subsequently the camera pose is estimated based on the reference objects \mathcal{F}_i^* using the daisy-chaining method. The goal is to determine the deviation of the estimated trajectory from the actual camera path in terms of position and orientation estimation errors after a finite number of loops. The position and orientation errors, $e_T(t)$ and $e_R(t)$, respectively, are defined as

$$e_T = \|t - \hat{t}\| \quad e_R = \|I - R^T \hat{R}\|, \quad (4-113)$$

where $t, \hat{t} \in \mathbb{R}^3$ denotes the actual and estimated camera position in the world frame, respectively. $R, \hat{R} \in \mathbb{R}^{3 \times 3}$ denotes the actual and estimated rotation of camera coordinate frame with respect to the world frame and $I \in \mathbb{R}^{3 \times 3}$ represents an identity matrix. Fig. 4-14 shows the pose estimation error after establishing 240 daisy-chains by traversing the circular trajectory 30 times. As seen from Fig. 4-14, in the absence of image noise the pose estimation error grows linearly and it is a result of numerical errors in estimating the rotation and translation during each daisy-chain that get propagated in the subsequent

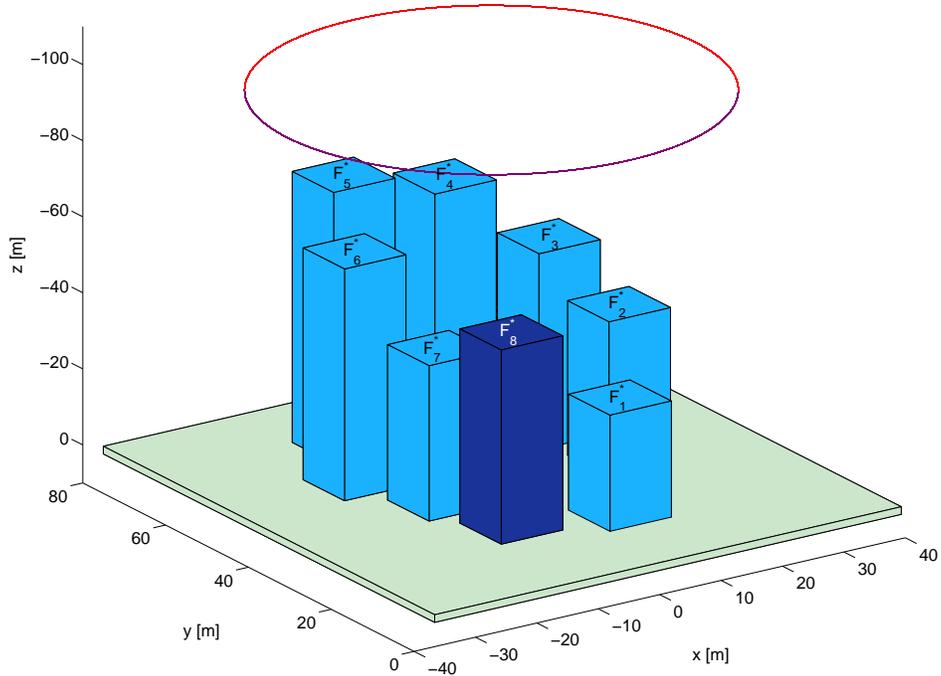


Figure 4-13. A simulation scenario depicting the circular trajectory of camera and a set of stationary reference objects \mathcal{F}_i^* where $i = 1, \dots, 8$.

daisy-chains. Fig. 4-16 shows the result of error propagation in presence of an additive white Gaussian noise with standard deviation $\sigma = 0.5 \text{ pixels}$, which demonstrates similarity with the ‘dead-reckoning’ systems. The estimated camera trajectory shown in Fig. 4-15 indicates deviation from the actual path in presence of noise when compared to Fig. 4-13. The error propagation problem can be addressed by mitigating the error using the known position of the camera in the scene or using an additional sensor. Specifically, the camera position can be updated when the camera revisits a known Euclidean point in the space, which can be determined by observing the known feature point ‘constellation’. Another approach based on sensor fusion could include an additional sensor (e.g., a GPS) to provide an absolute position of the camera when the measurement is available. Fig.

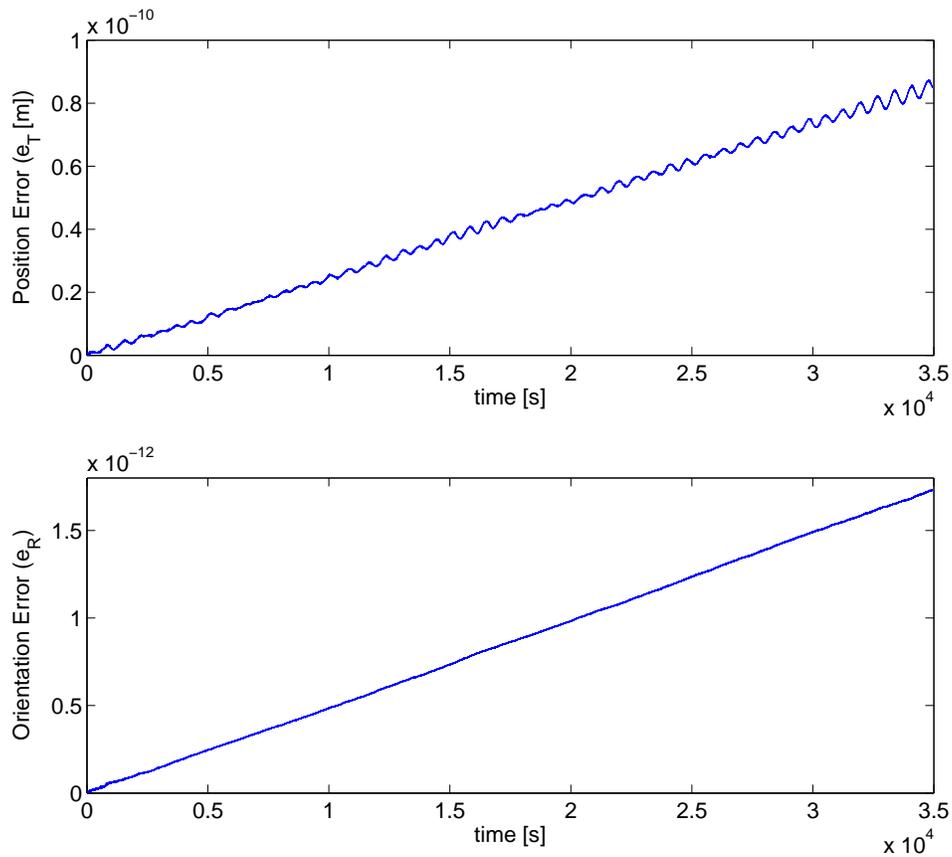


Figure 4-14. Error propagation in daisy-chaining pose estimation method in absence of feature point noise after 240 daisy-chains by traversing the circular trajectory 30 times.

4-15 shows the estimated trajectory of the camera and Fig. 4-18 shows the result of error propagation in presence of an additive white Gaussian noise with standard deviation $\sigma = 0.5 \text{ pixels}$ after updating the camera position at the end of each circular trajectory. It can be seen from Fig. 4-18 that the pose estimation error in daisy-chaining can be bounded by using an additional sensor or using the knowledge of the scene.

4.6 Concluding Remarks

In this chapter, a daisy-chaining vision-based control, localization and mapping approach has been presented. A visual servo tracking controller is first developed using this daisy-chaining approach to enable a control object to track a desired trajectory

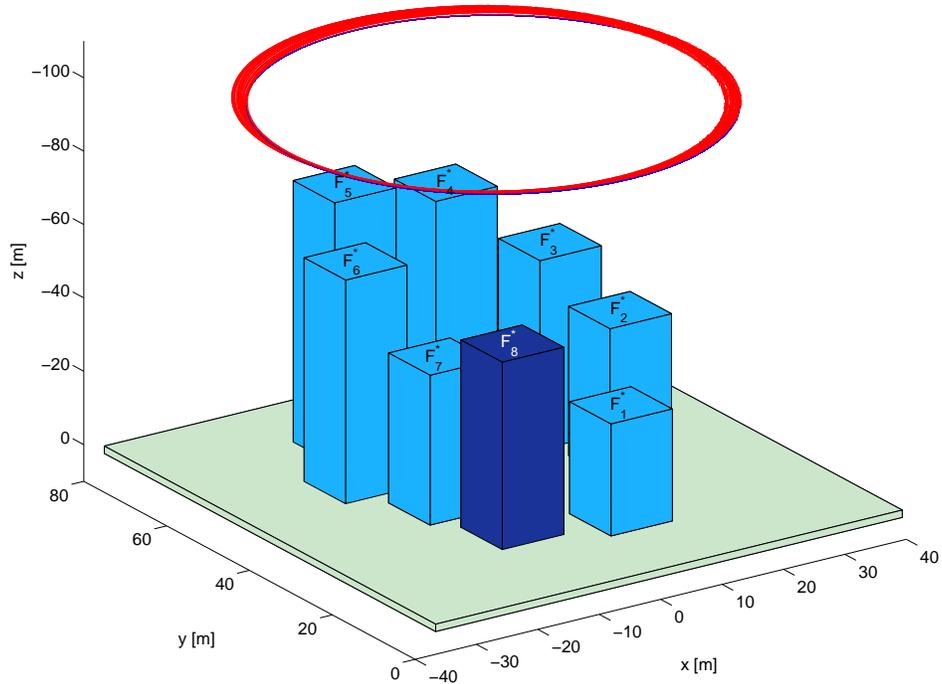


Figure 4-15. A simulation scenario depicting the estimated camera trajectory in presence of white Gaussian image noise and a set of stationary reference objects \mathcal{F}_i^* where $i = 1, \dots, 8$.

represented by a sequence of images. An example is followed to show its application in tracking control of a nonholonomic UGV. By fusing the daisy-chaining strategy with the geometric reconstruction method, the Euclidean position of the UGV and reference objects are identified to provide SLAM of the UGV.

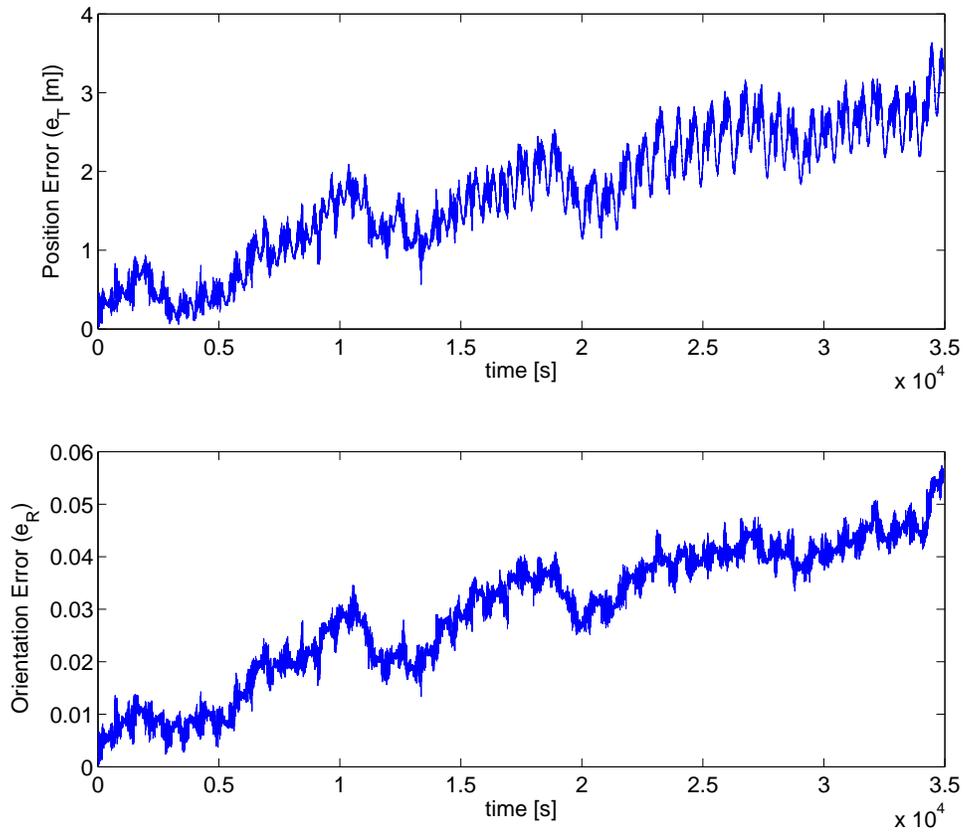


Figure 4-16. Error propagation in daisy-chaining pose estimation method in presence of white Gaussian noise after 240 daisy-chains by traversing the circular trajectory 30 times.

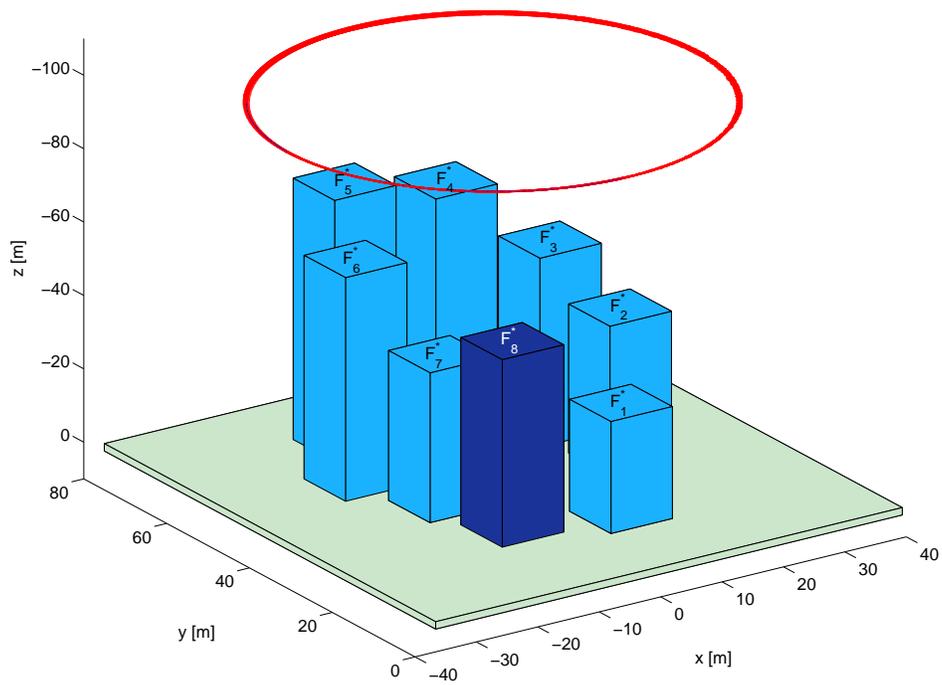


Figure 4-17. A simulation scenario depicting the estimated camera trajectory in presence of white Gaussian image noise by updating the camera position at the end of each circular trajectory and a set of stationary reference objects \mathcal{F}_i^* where $i = 1, \dots, 8$.

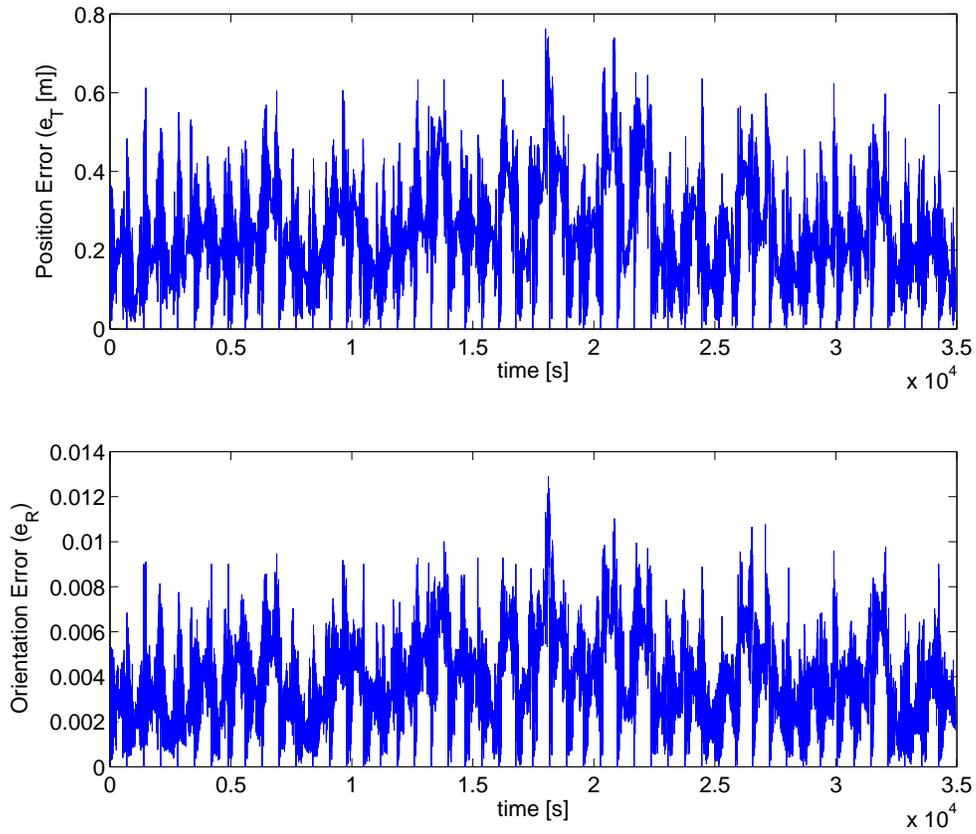


Figure 4-18. Error propagation in daisy-chaining pose estimation method in presence of white Gaussian noise by updating the camera position at the end of each circular trajectory.

CHAPTER 5 CONCLUSIONS

5.1 Research Summary

The research presented in this dissertation monograph centers around the design and analysis of visual servo control strategies and vision-based robust pose estimation with an objective of long range navigation and control of autonomous systems. The focus of the research in Chapter 2 is to develop a computationally deterministic pose estimation method that is robust to feature outliers. Chapter 2 presents the development of a novel robust algorithm for estimation of the relative pose between two calibrated images, which is coined as *Pose Estimation by Gridding of Unit Spheres* (PEGUS).

The key idea behind the method is, if there are M matched pairs of feature points between two views, one can compute a maximum of $\binom{M}{P}$ possible pose hypotheses by using a P -point algorithm. The developed algorithm selects a subset of “low-noise” hypotheses by empirically estimating the probability density function of the rotation and translation random variables, and averages them, conforming manifold constraints, to compute a pose estimate. The selection of low-noise hypotheses is facilitated by a unit-quaternion representation of rotation, which enables clustering of the rotation hypotheses on the 3-sphere \mathbb{S}^3 to identify the dominant cluster or mode. An identical approach facilitates estimation of unit translation that lie on the 2-sphere \mathbb{S}^2 .

The results in Chapter 2 demonstrate an improved performance of PEGUS against RANSAC+least squares as well as non-linear mean shift method, both in terms of the estimation accuracy and computation time. By virtue of non-iterative formulation underlying the deterministic structure of PEGUS, the computation time is more predictable than that of RANSAC and non-linear mean shift algorithm, thus making it amenable to a variety of real-time applications such as tracking control of an autonomous agent. Chapters 3 and 4 provide the development of vision-based control methods which can be benefitted from the robust pose estimation method given in Chapter 2.

Control of a moving object using a stationary camera and *vice versa* are well attended problems in the literature of visual servo control and various solutions exist for a class of autonomous systems. However, control of a moving object using the image feedback from a moving camera has been a well-known problem due to the unknown relative velocity associated with moving camera and moving object. In Chapter 3, a collaborative visual servo controller, which is coined the *daisy-chaining method*, is developed with an objective to regulate a sensor-less unmanned ground vehicle (UGV) to a desired pose utilizing the feedback from a moving airborne monocular camera system.

The contribution of research in Chapter 3 is the development of multi-view geometry, or photogrammetry, based concepts to relate the coordinate frames attached to moving camera, moving UGV, and desired UGV pose specified by an *a priori* image. Geometric constructs developed for traditional camera-in-hand problems are fused with fixed-camera geometry to develop a set of Euclidean homographies. Due to intrinsic physical constraints, one of the resulting Euclidean homographies is not measurable through a set of spatiotemporal images as the corresponding projective homography can not be developed. Hence, the new geometric formulations, termed *virtual homography*, are conceived to solve for the homography in order to develop a measurable error system for the nonholonomic UGV. Asymptotic regulation results are proved using the Lyapunov-based stability analysis.

Further, in Chapter 3, the results are extended to include asymptotic regulation of an UGV based on the scenario that the given reference objects can leave camera field-of-view (FOV) while another reference object enters FOV. The controller is developed - with the underlying geometrical constructs that daisy chain multiple reference objects - such that the airborne camera does not require to maintain a view of the static reference object; therefore the airborne camera/UGV pair can navigate over an arbitrarily large area. Simulation results are provided to demonstrate the performance of the daisy-chaining based control.

Building on the results in Chapter 3, the complex problem of cooperative visual servo tracking control is formulated in Chapter 4 with an objective to enable an UGV to follow a desired trajectory encoded as a sequence of images utilizing the image feedback from a moving airborne monocular camera system. The desired trajectory of an UGV is recorded by a moving airborne monocular camera \mathcal{I}_M traversing an unknown time-varying trajectory. The control objective is to track an UGV along the desired trajectory using the image feedback from a moving airborne camera \mathcal{I} that may traverse different trajectory than that of \mathcal{I}_M . The association as well as the relative velocity problem is addressed by introducing a daisy-chaining structure to link a series of projective homographies and expressing them in a constant reference frame. An adaptive parameter update law is employed to actively compensate for the lack of object model and depth measurements. Based on the open-loop error system, a tracking control law is developed through the application of Extended Barbalat's lemma in the Lyapunov-based framework to yield an asymptotic stability.

The tracking results are extended to reseed the stationary reference objects while formulating the additional projective homography relationship to provide an unrestricted applicative area of operation. The theoretical development in Chapter 4 manifests the coalescence of daisy-chaining controller and newly formed geometric reconstruction technique towards application in visual simultaneous localization and mapping (vSLAM). Simulation results are provided demonstrating the tracking control of an UGV in presence of multiple stationary reference objects and visual simultaneous localization and mapping (vSLAM) results are presented.

The daisy-chaining based control scheme developed in Chapters 3 and 4 is based on estimating the pose of an UGV with respect to stationary reference objects \mathcal{F}_i^* where $i = 1, \dots, n$ using a moving monocular camera. Since the stationary reference object can leave the camera FOV and a new reference object enters the FOV, it is necessary to determine the pose of the new reference object with respect to the receding object in

order to provide the pose information of a moving agent such as an UGV or the camera itself. Therefore, the error in pose measurement between the stationary reference objects would be propagated through the subsequent reference objects. The error propagation is analyzed in Chapter 4 by performing a numerical simulation, which shows that the pose estimation error grows linearly. Possible solutions are provided, along with simulation results, to mitigate the error propagation in daisy-chaining.

5.2 Recommendations for Future Work

Robust pose estimation method presented in Chapter 2 is based on estimating the probability density function (pdf) of rotation and translation random variables using the histogram density estimator by segmenting the 3-sphere and 2-sphere, respectively. Future work would focus on developing a method to determine the ‘best’ segmentation of sphere to minimize the pose estimation error and reduce the computation time. Also, an adaptive parameter law can be developed to determine the distance ϵ around the mode to extract the low-noise measurements.

The daisy-chaining method developed in Chapters 3 and 4 assumes a known geometric length on the moving agent and stationary reference objects. In practice, this assumption might be a too restrictive. Therefore, the future work would include a nonlinear observer-based range identification method in the daisy-chaining framework to estimate the required geometric length on the object. In order to estimate such a Euclidean parameter using the two-dimensional image information, an additional sensor would be included to provide the six degree-of-freedom camera velocity. The velocity measurements can also be fused with the pose estimates in a Kalman-like structure to mitigate the error propagation in daisy-chaining.

REFERENCES

- [1] R. Subbarao, Y. Genc, and P. Meer, “Nonlinear mean shift for robust pose estimation,” in *Proc. the Eighth IEEE Workshop on Applications of Computer Vision*. Washington, DC, USA: IEEE Computer Society, 2007, p. 6.
- [2] C. Tomasi and T. Kanade, “Detection and tracking of point features,” Carnegie Mellon University, Tech. Rep., 1991.
- [3] R. I. Hartley, “In defense of the eight-point algorithm,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, p. 580593, June 1997.
- [4] D. Nistér, “An efficient solution to the five-point relative pose problem,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, June 2004.
- [5] M. Fischler and L. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, pp. 381–385, 1981.
- [6] P. H. S. Torr and A. Zisserman, “Mlesac: a new robust estimator with application to estimating image geometry,” *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.
- [7] O. Chum and J. Matas, “Matching with PROSAC - progressive sample consensus,” in *Proc. IEEE Int. Conf. Computer Vision Pattern Recognition*, C. Schmid, S. Soatto, and C. Tomasi, Eds., vol. 1. Los Alamitos, USA: IEEE Computer Society, June 2005, pp. 220–226.
- [8] E. Michaelsen, W. von Hansen, M. Kirchhof, J. Meidow, and U. Stilla, “Estimating the essential matrix: GOODSAC versus RANSAC,” in *Photogrammetric Computer Vision (PCV06)*, 2006.
- [9] D. Nistér, “Preemptive RANSAC for live structure and motion estimation,” *Journal of Machine Vision and Applications*, vol. 16, pp. 321–329, December 2005.
- [10] M. Moakher, “Means and averaging in the group of rotations,” *SIAM Journal on Matrix Analysis and Applications*, vol. 24, 2002.
- [11] D. Burschka and G. Hager, “Vision-based control of mobile robots,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2001, pp. 1707–1713.
- [12] P. Corke and S. Hutchinson, “A new hybrid image-based visual servo control scheme,” in *Proc. IEEE Conf. Decision Control*, 2000, pp. 2521–2527.
- [13] —, “A new partitioned approach to image-based visual servo control,” *IEEE Trans. Robot. Automat.*, vol. 17, no. 4, pp. 507–515, 2001.

- [14] N. J. Cowan and D. Koditschek, “Planar image-based visual servoing as a navigation problem,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2000, pp. 1720–1725.
- [15] A. Das, R. Fierro, V. Kumar, B. Southall, J. Spletzer, and C. Taylor, “Real-time vision-based control of a nonholonomic mobile robot,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2001, pp. 1714–1719.
- [16] W. E. Dixon, D. M. Dawson, E. Zergeroglu, and A. Behal, “Adaptive tracking control of a wheeled mobile robot via an uncalibrated camera system,” *IEEE Trans. Syst., Man, Cybern. - Part B: Cybern.*, vol. 31, no. 3, pp. 341–352, 2001.
- [17] G. Hagar, D. Kriegman, A. Georghiades, and O. Ben-Shahar, “Toward domain-independent navigation: Dynamic vision and control,” in *Proc. IEEE Conf. Decision Control*, 1998, pp. 3257–3262.
- [18] T. Hamel and R. Mahony, “Image based visual servo control for a class of aerial robotic systems,” *Automatica*, vol. 43, no. 11, pp. 1975–1983, 2007.
- [19] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, “Manipulator control with image-based visual servo,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 1991, pp. 2267–2272.
- [20] N. Guenard, T. Hamel, and R. Mahony, “A practical visual servo control for a unmanned aerial vehicle,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2007, pp. 1342–1348.
- [21] Y. Ma, J. Kosecka, and S. Sastry, “Vision guided navigation for nonholonomic mobile robot,” *IEEE Trans. Robot.*, vol. 15, no. 3, pp. 521–536, 1999.
- [22] G. Mariottini, D. Prattichizzo, and G. Oriolo, “Image-based visual servoing for nonholonomic mobile robots with central catadioptric camera,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2006, pp. 538–544.
- [23] G. L. Mariottini, G. Oriolo, and D. Prattichizzo, “Image-based visual servoing for nonholonomic mobile robots using epipolar geometry,” *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 87–100, 2007.
- [24] K.-T. Song and J.-H. Huang, “Fast optical flow estimation and its application to real-time obstacle avoidance,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2001, pp. 2891–2896.
- [25] H. Y. Wang, S. Itani, T. Fukao, and N. Adachi, “Image-based visual adaptive tracking control of nonholonomic mobile robots,” in *Proc. of the IEEE/RJS International Conference on Intelligent Robots and Systems*, 2001, pp. 1–6.
- [26] J. Chen, W. E. Dixon, D. M. Dawson, and M. McIntire, “Homography-based visual servo tracking control of a wheeled mobile robot,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, pp. 1814–1819.

- [27] J. Chen, W. E. Dixon, D. M. Dawson, and V. Chitrakaran, "Visual servo tracking control of a wheeled mobile robot with a monocular fixed camera," in *Proc. IEEE Conf. Control Applications*, 2004, pp. 1061–1066.
- [28] J. Chen, W. E. Dixon, D. M. Dawson, and M. McIntyre, "Homography-based visual servo tracking control of a wheeled mobile robot," *IEEE Trans. Robot.*, vol. 22, no. 2, pp. 406–415, 2006.
- [29] Y. Fang, D. M. Dawson, W. E. Dixon, and M. S. de Queiroz, "2.5D visual servoing of wheeled mobile robots," in *Proc. IEEE Conf. Decision Control*, 2002, pp. 2866–2871.
- [30] Y. Fang, W. E. Dixon, D. M. Dawson, and P. Chawda, "Homography-based visual servoing of wheeled mobile robots," *IEEE Trans. Syst., Man, Cybern. - Part B: Cybern.*, vol. 35, no. 5, pp. 1041–1050, 2005.
- [31] G. Hu, W. E. Dixon, S. Gupta, and N. Fitz-coy, "A quaternion formulation for homography-based visual servo control," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2006, pp. 2391–2396.
- [32] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 6, pp. 1052–1067, 2007.
- [33] P. Jensfelt, D. Kragic, J. Folkesson, and M. Bjorkman, "A framework for vision based bearing only 3D SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2006, pp. 1944–1950.
- [34] S. Se, D. Lowe, and J. Little, "Global localization using distinctive visual features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2002, pp. 226–231.
- [35] L. Goncalves, E. di Bernardo, D. Benson, M. Svedman, J. Ostrowski, N. Karlsson, and P. Pirjanian, "A visual front-end for simultaneous localization and mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2005, pp. 44–49.
- [36] R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard, "Visually navigating the RMS titanic with SLAM information filters," in *Proc. Robotics: Science and Systems*, June 2005.
- [37] J.-H. Kim and S. Sukkarieh, "Airborne simultaneous localisation and map building," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2003, pp. 406–411.
- [38] I.-K. Jung and S. Lacroix, "High resolution terrain mapping using low attitude aerial stereo imagery," in *Proc. IEEE Int. Conf. Computer Vision*, 2003, pp. 946–951.
- [39] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Computer Vision*, 1999, pp. 1150–1157.

- [40] R. Subbarao and P. Meer, “Nonlinear mean shift over riemannian manifolds,” *Int. J. Computer Vision*, vol. 84, p. 120, 2009.
- [41] P. Leopardi, “A partition of the unit sphere into regions of equal area and small diameter,” *Electronic Trans. Numerical Analysis*, vol. 25, pp. 309–327, 2006.
- [42] S. Mehta, W. E. Dixon, D. MacArthur, and C. D. Crane, “Visual servo control of an unmanned ground vehicle via a moving airborne monocular camera,” in *Proc. American Control Conf.*, 2006, pp. 5276–5211.
- [43] S. Mehta, G. Hu, N. Gans, and W. E. Dixon, “Adaptive vision-based collaborative tracking control of an ugv via a moving airborne camera: A daisy chaining approach,” in *Proc. IEEE Conf. Decision Control*, 2006, pp. 3867–3872.
- [44] C. Samson, “Control of chained systems application to path following and time-varying point-stabilization of mobile robots,” *IEEE Trans. Automat. Contr.*, vol. 40, no. 1, pp. 64–77, 1995.
- [45] E. Malis and F. Chaumette, “2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement,” *Int. J. Computer Vision*, vol. 37, no. 1, pp. 79–97, 2000.
- [46] J. Chen, D. M. Dawson, W. E. Dixon, and A. Behal, “Adaptive homography-based visual servo tracking for a fixed camera configuration with a camera-in-hand extension,” *IEEE Trans. Contr. Syst. Technol.*, vol. 13, no. 5, pp. 814–825, 2005.
- [47] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*. Cambridge, Massachusetts: MIT Press, 1993.
- [48] O. Faugeras and F. Lustman, “Motion and structure from motion in a piecewise planar environment,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 2, pp. 485–508, 1988.
- [49] Z. Zhang and A. R. Hanson, “Scaled euclidean 3D reconstruction based on externally uncalibrated cameras,” in *IEEE Symp. Computer Vision*, 1995, pp. 37–42.
- [50] D. M. Dawson, E. Zergeroglu, A. Behal, and W. E. Dixon, *Nonlinear Control of Wheeled Mobile Robots*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2001.
- [51] C. A. Desoer and M. Vidyasagar, *Feedback Systems: Input-Output Properties*. Orlando, FL, USA: Academic Press, Inc., 1975.
- [52] K. Dupree, N. Gans, W. Mackunis, and W. E. Dixon, “Euclidean feature tracking for a rotating satellite,” in *Proc. American Control Conf.*, 2007, pp. 3874–3879.
- [53] O. Faugeras, *Three-Dimensional Computer Vision*. The MIT Press, Cambridge Massachusetts, 2001.

- [54] P. K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, “Automated tracking and grasping of a moving object with a robotic hand-eye system,” *IEEE Trans. Robot. Automat.*, vol. 9, no. 2, pp. 152–165, 1993.
- [55] G. D. Hager, W.-C. Chang, and A. S. Morse, “Robot hand-eye coordination based on stereo vision,” *IEEE Contr. Syst. Mag.*, vol. 15, no. 1, pp. 30–39, 1995.
- [56] S. Hutchinson, G. Hager, and P. Corke, “A tutorial on visual servo control,” *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 651–670, 1996.
- [57] S. Wiiesoma, D. Wolfe, and R. Richards, “Eye-to-hand coordination for vision-guided robot control applications,” *Int. J. Robot. Res.*, vol. 12, no. 1, pp. 65–78, 1993.
- [58] N. Papanikolopoulos, P. Khosla, and T. Kanade, “Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision,” *IEEE Trans. Robot. Automat.*, vol. 9, no. 1, pp. 14–35, 1993.
- [59] F. Bensalah and F. Chaumette, “Compensation of abrupt motion changes in target tracking by visual servoing,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1995, pp. 181–187.
- [60] G. Hu, S. Mehta, N. Gans, and W. E. Dixon, “Daisy chaining based visual servo control part I: Adaptive quaternion-based tracking control,” in *Proc. IEEE Multi-Conf. Syst. Control*, 2007, pp. 1474–1479.
- [61] B. Boufama and R. Mohr, “Epipole and fundamental matrix estimation using virtual parallax,” in *Proc. IEEE Int. Conf. Computer Vision*, 1995, pp. 1030–1036.
- [62] G. Hu, N. Gans, S. Mehta, and W. E. Dixon, “Daisy chaining based visual servo control part II: Extensions, applications and open problems,” in *Proc. IEEE Multi-Conf. Syst. Control*, 2007, pp. 729–734.
- [63] N. Gans, G. Hu, and W. E. Dixon, *Complexity and Nonlinearity in Autonomous Robotics, Encyclopedia of Complexity and System Science*. Springer, to appear 2008, ch. Image-Based State Estimation.
- [64] W. Mackunis, N. Gans, K. Kaiser, and W. E. Dixon, “Unified tracking and regulation visual servo control for wheeled mobile robots,” in *Proc. IEEE Multi-Conf. Syst. Control*, 2007, pp. 88–93.
- [65] O. Faugeras and F. Lustman, “Motion and structure from motion in a piecewise planar environment,” *Int. J. Pattern Recognition and Artificial Intelligence*, vol. 2, no. 3, pp. 485–508, 1988.
- [66] M. Shuster, “A survey of attitude representations,” *J. Astronautical Sciences*, vol. 41, no. 4, pp. 439–518, 1993.

- [67] W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti, *Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach*. Birkhuser Boston, 2003.
- [68] J. J. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliff, NJ: Prentice Hall, Inc., 1991.
- [69] S. Baker and S. Nayar, “A theory of single-viewpoint catadioptric image formation,” *Int. J. Computer Vision*, vol. 35, no. 2, pp. 175–196, 1999.
- [70] B. Kim, D. Roh, J. Lee, M. Lee, K. Son, M. Lee, J. Choi, and S. Han, “Localization of a mobile robot using images of a moving target,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2001, pp. 253–258.

BIOGRAPHICAL SKETCH

Siddhartha Mehta received Bachelor of Engineering degree in mechanical engineering from COEP, University of Pune, India. Subsequently, for a short period of a year he worked at Thermax Limited before joining University of Florida in 2003. He graduated with concurrent masters in mechanical engineering and agriculture engineering in 2007 with thesis titled ‘Vision-Based Control for Autonomous Robotic Citrus Harvesting’ under the guidance of Dr. Thomas F. Burks, which received the 2007 best thesis award in agriculture engineering. Later, he joined the Nonlinear Controls and Robotics (NCR) group in mechanical engineering to pursue Ph.D. under the advisement of Dr. Warren E. Dixon with focus on vision-based control and estimation. He is a student member of IEEE and AIAA.