Development of a Visual-Teach-and-Repeat based Navigation Technique on Quadrotor Aerial Vehicle

by

©Trung Nguyen

A Thesis submitted to the School of Graduate Studies
in partial fulfillment of
the requirements for the degree of

Master of Engineering

Faculty of Engineering and Applied Science

Memorial University of Newfoundland

October, 2014

St. John's Newfoundland

Abstract

The objective of this thesis is to develop a vision-based navigation and control technique for quadrotor to operate in GPS-denied environments. The navigating technique has been developed while using *Visual-Teach-and-Repeat* (VT&R) method. This method is qualitative where the position of the quadrotor is estimated based on a set of reference images. These reference images are collected while taking the quadrotor manually along a desired route. Each image, collected in the database, represents one segment of the desired route. The features are extracted from these images using a well-known method, *Speeded-Up Robust Features* (SURF) [1].

When the quadrotor is navigated along the desired route (repeat mode), the quadrotor performs self-localization. Three methods of self-localization are presented. In method I, the SURF features observed on the current image are matched with the SURF features of the reference images to compute the probability value of each segment in the desired route. The segment that provides the best probability value is chosen as the current segment of the quadrotor. To improve the accuracy of localization, in the method II, the condition of feature-size relation with spatial distance is imposed. In the method III, the estimation of the current segment of the quadrotor is built on Bayes's rule.

Based on the appearance-based error of feature coordinates, the system computes qualitative motion control commands (desired yaw and height) for the next movement

in order to control the quadrotor to follow the desired route. This computation is developed on Funnel Lane theory, which was originally proposed in [2], in order to 2D navigate ground vehicle following the desired route. The thesis extends it to 3D navigation for the quadrotor. Funnel Lane theory qualitatively defines possible positions where the vehicle can fly straight by the constraints of features coordinates between the current image and the reference image. If the quadrotor locates outside the funnel lane, it will be navigated back to the funnel lane.

A nonlinear geometric controller has been developed to convert the motion control commands, generated basing on VT&R technique, into control inputs necessary for the four rotors in the quadrotor. The design of proposed controller is simplified by concentrating on the errors of rotational matrix, instead of attempting to access the errors of each degree of freedom.

The quadrotor for this thesis is chosen as the well-known AR.Drone model [3]. The whole system is modeled and simulated in Gazebo simulator using *Robot Operating System* (ROS). Four image databases have been used for testing self-localization: two databases around Engineering building of Memorial University of Newfoundland, COLD database and New College database. With proposed VT&R technique, the quadrotor is able to independently follow a long route without GPS-information or the support from an external tracking system. The proposed system has a simple implementation, inexpensive computation and high potential for exploring and searching-and-rescuing missions.

To my Mom, Dad and Brother

Acknowledgements

It is an exclusive privilege of being advised and working in *Intelligent Systems Lab* (ISLab), under the supervision of Dr. George K. I. Mann and Dr. Raymond G. Gosine at my Master of Engineering degree. Their invaluable lessons, academic guidance as well as encouragement greatly improve my research skills. I would like to express my sincere and deepest appreciation to them and their values they give me in order to finish my thesis.

To ISLab members, Mr. Oscar De Silva, Mr. Thumeera R. Wanasinghe and Mr. Mohamed Mehrez, I would like to thank them for their great assistances, patience and inspirations.

Finally, this research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), C-CORE J. I. Clark Chair and Memorial University of Newfoundland, Canada.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

FL Fixed Landmark

GPS Global Positioning System

IMU Inertial Measurement Unit

KLT Kanade-Lucas-Tomasi

LQR Linear Quadratic Regulator

MPC Model Predictive Control

MSE Mean Square Error

PID Proportional-Integral-Derivative

ROS Robot Operating System

SIFT Scale-Invariant Feature Transform

SLAM Simultaneous Localizing And Mapping

SURF Speeded-Up Robust Features

UAV Unmanned Aerial Vehicle

VT&R Visual Teach and Repeat

Chapter 1

Introduction and Overview

1.1 Introduction

In recent years, *Unmanned Aerial Vehicle* (UAV), especially quadrotor type micro aerial vehicles, has become one of the fastest developing technologies of unmanned vehicles. Numerous commercial models of quadrotors [3, 5] have been produced by utilizing advanced mechanical and electrical technologies in conjunction with accurate sensor measurements and high-speed processing. Although quadrotor has demonstrated unique potentials to replace humans performing difficult tasks [6, 7, 8, 9], the widespread use of quadrotor in industries and commercials is still limited due to numerous navigational issues. The utilization of *Global Positioning System* (GPS) for localization and control does not satisfy the necessary requirements of fully autonomous applications. GPS technology shows some unreliability and unavailability in several specific environments such as indoor and in urban canyons.

Localization of quadrotor attempts to estimate the current position of quadrotor with respect to a fixed coordinate system in the working environment. Basically, if the initial position of quadrotor is defined, the measurements of traveling speeds over

elapsed time and course can reveal the current position of quadrotor. The research works, presented in [10, 11], apply dead-reckoning technique through visual odometry on the camera of quadrotor. This technique can produce acceptable results only for short-term operations due to experimental issues in cumulative errors and measuring drifts. Combining these measurements with other external measurements taken from a ground station [11] by an External Kalman Filter can improve the accuracy of localization. However, the ability of independent operations of quadrotor is reduced. One of popular strategies, including both predicting and updating step, is Simultaneous Localizing And Mapping (SLAM) technique. The SLAM technique has been implemented in quadrotors either laser-based [12, 13] or vision-based [14, 15, 16]. Such implementations require considerable payload to carry exteroceptive sensors and demand higher computational cost. Running SLAM filters in computationally and hardware constrained systems is expensive and limits the applications in long term operations. Therefore, the practical applications of SLAM technique on quadrotor need the support from ground vehicle [17] where most of computations are performed. Sharing computations among multiple vehicles is considered in the scheme of relative localization technique [18, 19, 20]. Although these kinds of systems have demonstrated practical applications to investigate indoor environment, the working volume of quadrotor is limited to a restricted space defined by the maximum measured distance from ground vehicles. Another class of techniques addresses the problem of localization from computer vision. Processing an image feedback determines and tracks specific objects such as long and straight parallel lines [21], artificial markers [22, 23]. The drawbacks of these techniques are the assumption of specific working environments which should contain the tracked specific objects.

To fulfill both the requirements of the practical implementation and also to overcome the limitations of above techniques, *Visual Teach and Repeat* (VT&R) tech-

nique [24, 25, 26, 27, 28] emerges as one of effective solutions to enable quadrotor to autonomously follow a desired route in GPS-denied environment. VT&R technique can be considered as a simpler form of SLAM technique, which reduces the demand of higher computational cost and complexity in implementation. Navigating while comparing reference images along the route, VT&R technique does not show the global consistent problem as in dead-reckoning technique.

As implied in the name, VT&R technique consists of two phases: learning phase and replaying phase. In the learning phase, quadrotor is controlled by users (using joystick, keyboard, teleoperation) along a desired route to collect numerous reference images, which become the database of the desired route. In the replaying phase, quadrotor will compare the current image, reflecting the environment at the moment, with the reference images, reflecting the environment in the past, in order to produce appropriate motions.

The advantages and practical applications of VT&R technique can be listed:

- *Independent operation*: The vehicle can work independently by itself in GPS-denied environment without the supports and connections from other vehicles or external tracking system for updating information and rectification.

- *Reusable database of the desired route*: The database of the desired route can be reused to navigate other vehicles in condition that the changes of working environment is trivial as long as the camera configurations of both vehicles are reasonably similar. Additionally, the database of the desired route can also be updated when the vehicle performs route following.

- *Visual-guiding*: VT&R technique is developed on the comparison between the current image and the reference images, which helps to navigate and guide the vehicle in surveillance operation, delivery, remote inspection and other au-

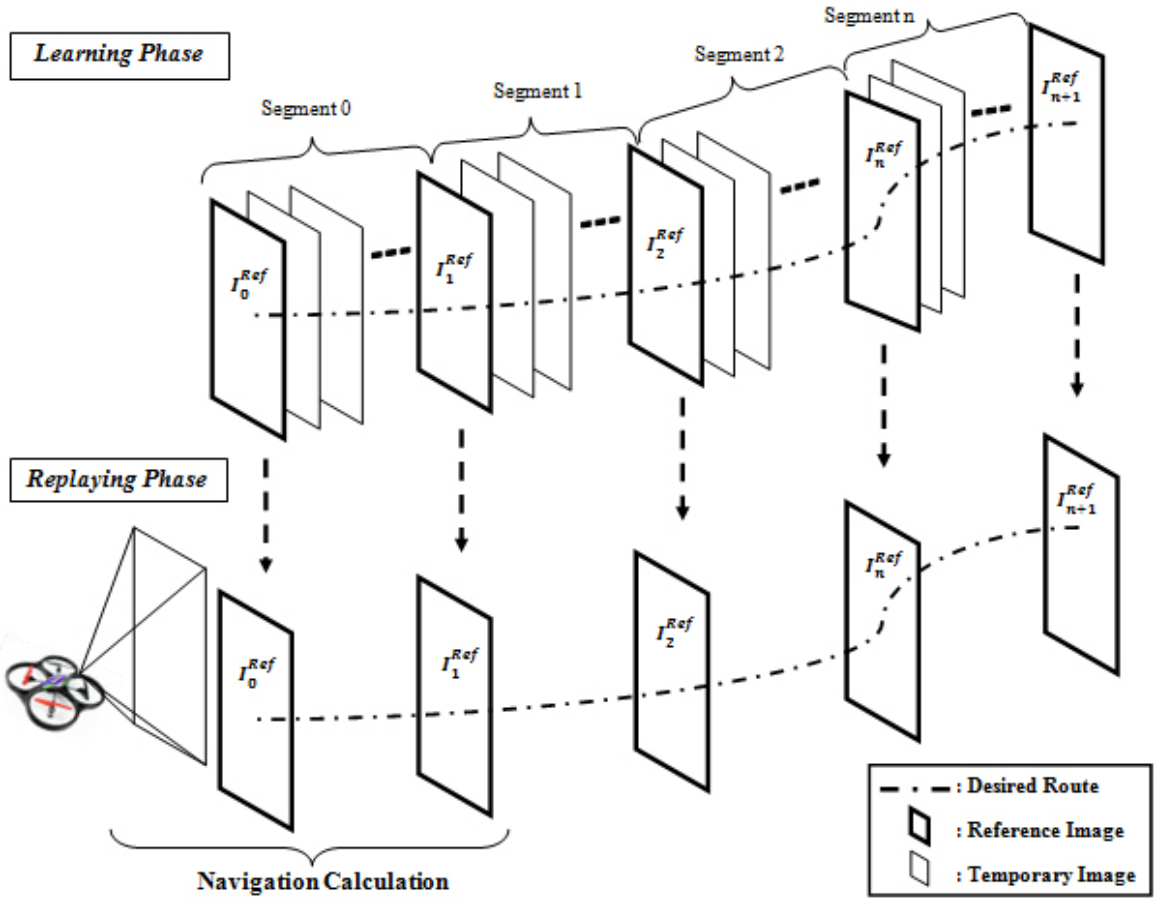

Figure 1.1: Learning Phase and Replaying Phase of VT&R aerial system

tonomous tasks without the information of GPS measurement.

- *Visual-homing*: One of VT&R applications aims to guide the vehicle back to the start position basing on the available set of reference images. Especially in ground-aerial multi-vehicle systems [20], the aerial vehicle is tracked by a sensor system, which is placed on the ground vehicle. When the aerial vehicle needs to exceed its working-area limit, VT&R technique can navigate it back to the ground vehicle. Additionally, VT&R system can play a contingency plan in case of lost connection with the ground vehicle.

If the VT&R systems are classified by the calculating approach of motion control commands, two general strategies have been developed for VT&R system: quantitative approach [27][24] and qualitative approach [26][2].

- *The quantitative or pose-based approach* reconstructs the position of the vehicle, the detected landmarks and the desired route in the same global coordinate frame basing on the current image and a set of reference images. Although the robustness of the route-following technique can be achieved, the quantitative approach still shows higher computational cost. These properties become disadvantageous for some specific systems, such as micro aerial vehicles, which possess strict constraint of energy consumption or consider VT&R as secondary plan for visual homing. Hence, this technique is limited to travel within short distances.

- *The qualitative or appearance-based approach* is an appearance-based approach which navigates on the qualitative comparison between the current image and the reference images. In qualitative approach, navigation is simply performed by comparing the current image with the reference image. Self-localization (or place recognition) firstly defines which image in database of the desired route

is appropriate for navigation. After that, basing on the coordinate alternation of features between two images, motion control commands are produced to navigate the vehicle to follow the desired route.

Compared to the quantitative approach, the qualitative approach demonstrates more advantages in computational cost, convenient in implementation as well as the low demand in energy consumption. These properties will become considerable benefits in controlling quadrotor aerial vehicle, which normally encountering the limitation of working time, payload, and reliable measurement for localization.

1.2 Problem Statement

The objective of the thesis is to design and develop qualitative VT&R navigation technique on quadrotor. The model of quadrotor is Ar.Drone [3], equipped with single, forward-looking camera, *Inertial Measurement Unit* (IMU) and altimeter. The design of qualitative VT&R system is a combination of three different parts: self-localization, qualitative motion control command, and controller:

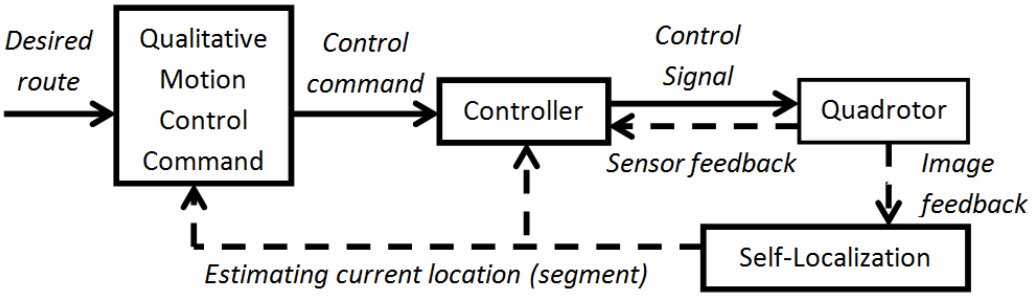

Figure 1.2: The block diagram of proposed qualitative VT&R aerial system

- *Self-localization*: The task of self-localization is to define which segment of the desired route, where quadrotor is locating. Appropriate image in the database of the desired route is loaded as reference for navigation. Self-localization is

developed on matched Speeded-Up Robust Features (SURF) features and the relation between spatial distance and the size of feature.

- *Qualitative motion control command*: Comparing the current image and the loaded reference image, qualitative motion control command part is developed to calculate the desired yaw angle and the desired height for the next movement. It is developed on Funnel Lane theory, developed in [2] for ground vehicles. Funnel Lane theory defines directly possible positions of quadrotor to fly straight, fly up, fly down, turn left and turn right while observing multiple landmark features that are extracted from the image feedback.

- *Controller*: The controller converts the motion control commands of the desired yaw and height, generated by qualitative motion control command part, into control inputs necessary for the four rotors in quadrotor. The controller is designed using nonlinear geometric control theory. The proposed design of the controller can directly calculate the control signals on the exact errors of rotational matrix without linear-approximating step in order to preserve the underlying geometric dynamic properties of the system.

1.3 Contribution

The contribution of the thesis has following aspects:

- The thesis has been able to develop VT&R based navigation technique for quadrotor aerial vehicle in order to work in GPS-denied environment.

- The thesis advances the state of the art in monocular visual self-localization. Three methods of self-localization are proposed by utilizing a number of matched SURF features, and the feature-size relation with spatial distance. Validations

of three proposed methods are conducted on four databases, collected by the author and by other research groups. The proposed methods can be applied into solving place recognition as well as image retrieval problem.

- Qualitative motion control command system is developed based on the coordinate changes of the matched features. Funnel Lane theory (proposed by the work [2]) is adapted and expanded into 3D for computing the appropriate motion control commands for the next movement.

- Nonlinear geometric controller is employed in VT&R system to develop a controller for the quadrotor. The controller is modeled using Bond-Graph modeling method and 20Sim simulation program.

- The proposed design of VT&R aerial system is implemented in Robot Operating System (ROS) [29] for simulations and real-time experiments.

1.4 Thesis Outline

The thesis consists of 6 chapters:

- *Chapter 1* introduces about the VT&R technique as well as its advantages and disadvantages. The qualitative approach shows more considerable advantages than the quantitative approach when applied into quadrotor aerial vehicle. The objective and some contributions of the thesis are defined.

- *Chapter 2* summarizes the main works related to the three main problems addressed in this thesis: self-localization, motion control command generation and the quadrotor controller.

- *Chapter 3* assesses the problem of self-localization. Three methods are proposed and analyzed in order to improve the estimation of the current segment in the

desired route. Experimental validations are conducted with offline and online processing.

- *Chapter 4* describes the design of qualitative motion control command part. Funnel Lane theory is used to present the geometric relation between the matched landmark features and possible positions where quadrotor can fly straight. Funnel-lane 3D motion control algorithm is proposed for computing the appropriate motion control commands for the next movement during the replaying phase.

- *Chapter 5* mentions about the controller part of quadrotor, which converts the motion control commands into control inputs necessary for the four rotors in quadrotor. The controller part is developed by nonlinear geometric control theory. The controller is designed and tested by bond-graph modeling method and 20Sim simulator before implementing into the VT&R system.

- *Chapter 6* presents some conclusion and future work.

Besides, some publications of the research is attached in the appendix:

- Trung Nguyen, George K. I. Mann and Raymond G. Gosine, "Vision-Based Qualitative Path-Following Control of Quadrotor Aerial Vehicle", The International Conference on Unmanned Aircraft Systems, 2014, Florida, USA. (Appendix 1)

- Trung Nguyen, George K. I. Mann and Raymond G. Gosine, "Vision-Based Qualitative Path-Following Control of Quadrotor Aerial Vehicle with Speeded-Up Robust Features", The Conference on Computer and Robot Vision, 2014, Quebec, Canada. (Appendix 2)

- Trung Nguyen, George K. I. Mann and Raymond G. Gosine, "Vision-Based 3D Navigation Technique for Quadrotor Aerial Vehicle in Path Following Missions",

The 22^{nd} Annual Newfoundland Electrical and Computer Engineering Conference, 2013, St.John's, Newfoundland, Canada.

- Trung Nguyen, George K. I. Mann and Raymond G. Gosine, "A Study on Nonlinear Trajectory Control of Quadrotor Aerial Vehicles", The 21^{st} Annual Newfoundland Electrical and Computer Engineering Conference, 2012, St.John's, Newfoundland, Canada.

Chapter 2

Related Works

About this chapter: This chapter summarizes the main works related to the three main problems addressed in this thesis. These include: self-localization, motion control command generation and quadrotor controller. The chapter mainly focuses on vision-based approaches used to address these problems.

2.1 Self-localization

Self-localization is the problem of finding the position of the vehicle in an environment. For quadrotor platforms, this problem is harder because of the dynamic nature of the platform [30]. To solve this problem, the following main methods are reported in literature.

2.1.1 External Tracking System

In outdoor settings, GPS is a viable solution for the problem. But as reference [24] points out, GPS systems suffer from poor accuracies and unavailability in certain environments such as indoors, and urban canyons. Therefore, GPS is not suitable to support autonomous tasks performed by the quadrotor, such as visual inspection and

navigation in cluttered environments.

Most of quadrotor trajectory control applications in GPS-denied settings uses an external tracking system such as VICON^{TM} motion capture system [5, 31, 32, 33]. The system allows to simultaneously track multiple quadrotors with sub millimeter localization accuracies at typical update rates of $\sim$ 300Hz [5]. This localization precision allows an implementation of aggressive flight path controllers, producing accurate reference tracking [5]. Solutions based on static external tracking systems are impractical in real-life surveillance and exploration tasks, due to the demanding infrastructure and investment required for enabling a static external tracking system covering the whole surveillance region. Furthermore, in applications such as disaster relief [34], the use of a static external tracking systems is impossible.

To overcome the limitations associated with static systems like VICON^{TM}, the references [18, 19, 20] attempt to set up the external tracking system on mobile vehicles. The external tracking system will include numerous sensor nodes to track the position of quadrotor, while each node is placed on a small mobile vehicle. The approach is recognized as the relative localization technique. The approach can be expanded to other unmanned vehicles for exploration and search-and-rescue applications. For example, in Intelligent Systems Lab, Faculty of Engineering and Applied Science, Memorial University of Newfoundland, a relative localization approach is developed in ground-aerial vehicles [20] to perform mapping, visual inspection in indoor environments. However, the working area of quadrotor is still limited by the measurable distance of the ground vehicle.

2.1.2 Dead Reckoning

Knowing the initial position as well as traveling speeds over elapsed time and course, the vehicle can estimate its current position. This type of localization is termed

dead reckoning. The dead-reckoning systems would inherently accumulate errors over time, which makes its estimation unreliable after a particular time period has elapsed. Therefore, this strategy is effective only for a short distance of traveling. For long term operations, it needs a reference point to correct the cumulative errors in its estimated current position. For example, in underwater applications [35], the dead-reckoning system of the vehicle is corrected when the vehicle surfaces to acquire a GPS measurement. In quadrotor applications, dead reckoning is mainly performed by optical flow detection with an onboard vision sensor [11, 10]. In [11], dead reckoning is fused with other measurements from ultrasound and IMU by an Extended Kalman Filter (EKF) to improve the estimation of current position. There is a growing area of research on performing dead-reckoning using image sensors. This is termed visual odometry. In visual odometry, the features between successive images are matched to solve the problem of finding the camera pose. This provides better accuracies than conventional optical flow systems. However, all methods which are conceptually based only on odometry, accumulate localization error over time. Therefore, dead reckoning is only applicable for quadrotor platforms for short periods of time.

2.1.3 Simultaneous Localizing and Mapping (SLAM)

SLAM is proposed as another solution to estimate the position of the vehicle [15, 17, 16, 36, 37]. The vehicle can localize itself by building up and updating the map of the environment, and simultaneously determining its relative position in the environment. SLAM employs the exteroceptive sensor data observed in the working environment during navigation. In order to fully capture the working environment, visual SLAM [14, 15, 16], has received much scientific consideration as a viable SLAM solution for quadrotors. Visual SLAM senses visual landmarks of the environment, and attempts to reconstruct all landmarks and quadrotor positions in a common global frame of

reference. This method is a development from 2D [38] to 3D SLAM [17]. Although SLAM allows a quadrotor to independently work in exploration missions, this strategy is not effective in the absence of a powerful sensory system. SLAM demands higher energy, computational power and sometime requires offline reconstruction [25]. The time-consuming computational processes prevents SLAM from some applications, which require high traveling velocity of the quadrotor. Additionally, requiring massive computational resources makes SLAM ineffective when investigating large-scale areas and following a long route. As a result, the SLAM strategy on quadrotors is mostly applied to investigate small constrained areas in limited periods of time. For example, in [17][34], quadrotor is operated with a ground vehicle and only used to map small areas, where the ground vehicle can not reach. The processing of SLAM is performed on the ground vehicle after receiving essential data from the quadrotor through wireless communication.

Some studies simplify the localization and mapping problem by establishing the map prior to localization. In [39], the environment is represented by a map of carefully placed markers in the environment. Localization of the quadrotor is performed by detecting the markers which are placed at known positions, by using the downward looking camera of the quadrotor. These approaches produce highly accurate results, and is mainly used as benchmarking methods to evaluate SLAM algorithms [39]. Therefore, these approaches are not applicable in practical environments where artificial marker placement is not viable and landmark locations are not known in advance.

2.1.4 Self-localization in VT&R Methods

VT&R technique uses a database of a desired route to self-localize the quadrotor before computing suitable motion control commands. Therefore, it does not use artificial markers and known landmark positions of the environment to perform localiza-

tion. Alternatively, it stores a set of reference images to use for localization, which is practical for many environments that the quadrotor operates. In VT&R techniques, the self-localization is performed using two main approaches. One is a pose-based approach, where the quadrotor estimates its full pose relative to the reference image in the database. The second method is an appearance-based approach where the quadrotor only estimates its current segment. Additionally, self-localization in VT&R also needs a method to perform transition between two successive segments, i.e., the quadrotor reaches the end of the current segment, it should determine when to load the reference image of the next segment, in order to continuously navigate quadrotor to follow the desired route.

2.1.4.1 Pose-based Approach

Works, reported in [27, 25, 24, 40, 41, 42], are examples of pose-based VT&R systems. In these methods, the relative pose between the current image frame and the reference image frame is estimated based on matched features between these images. The shortest relative pose determines the appropriate reference image to be used for navigation. The initialization is performed using all reference images in order to define the initial position of the vehicle. This is a highly computationally expensive task. Then self-localization is performed using few reference images (2 or 3) for the transition between two successive segments [43, 27]. As a result, the time of calculation is considerably decreased when the vehicle is following the desired route. However, the ability to handle kidnapped-robot scenarios and recovery from considerable deviations from the desire route is reduced. The work in [28] uses visual odometry to estimate the quadrotor current pose, which is compared with the reference poses of the desired route for localization. Instead of using the reference images, the method uses the reference poses to overcome the memory consumption problem.

Due to the computational cost of the pose-based approach, in some studies, the calculation is performed on a ground station (desktop) before sending the motion commands to quadrotor [43]. The limitations of the quadrotor hardware do not allow the onboard calculation. Although the pose-based approaches show good performance in terms of self-localization, the estimation still requires high computational cost when simultaneously processing many reference images to find the pose and the reference image. Additionally, the pose-based approaches require calibrated cameras and scale factor estimation for metric localization within a segment.

2.1.4.2 Appearance-based Approach

A simpler approach for self-localization in VT&R systems is to use appearance-based methods. Appearance-based approach does not use camera parameters, or attempts to extract exact pose of the platform, rather it estimates the segment where the quadrotor is operating. This information is sufficient to implement appearance-based controllers to navigate the quadrotor in the desired route.

Most of the proposed methods such as [44, 45, 46] use matched features to determine the current segment of the vehicle. These methods heavily rely on reliable feature-matching results for self-localization. Therefore, the development of the feature detection and matching techniques has received much attention as means of improving self-localization in VT&R systems. Work, reported in [44], aims at optimizing the combination of feature detection methods and feature descriptor methods. Although this improves the performance of feature matching, the practical applicability of the method is questionable, due to the absence of the validation of the uncommon feature-detection methods used in different environments such as indoor scenarios. Majdik et al [45] propose to produce a virtual view of the current image using an image database, in order to improve the performance of the feature-matching step. This

work reports validation of this strategy for outdoor scenarios by using the Google-Street-View database to generate the virtual view.

Another problem faced by VT&R self-localization modules is developing robust methods for automatic transition between two successive segments. The works, described in [47, 48, 49], propose to set a switching threshold for *Mean Square Error* (MSE) of feature coordinates in the images, which tends to decrease when the vehicle moves closely to the reference image. The simple method satisfies the requirement of the transition without consuming much computation. In work reported in [2], the probability computation of the transition between two successive segments consists of multiple information sources (i.e. matched features, distance traveled and heading angle). Accurate and robust performances along different long routes have been demonstrated using this method. The self-localization is performed using *Kanade-Lucas-Tomasi* (KLT) features, which are sensitive with the ambient lighting, rotation and scale of the viewpoint. This method cannot be directly employed in quadrotor applications because the odometric measures used to locate the vehicle in the segment are highly unreliable. However, the study suggests that the use of multiple sources to support transition significantly improves the self-localization capability of vehicles. As improved models of the transition computation, the references use Bayes filter [50], Kalman filter [51] or Markov filter [52]. Here, the estimation of self-localization highly depends on the estimation of the previous location and the accuracy of traveling measurements. The estimating errors, which occur in previous estimation, can not be fixed in the current estimation.

The work in [46] self-localizes the vehicle on matched *Scale-Invariant Feature Transform* (SIFT) features of omnidirectional camera images. The relation between SIFT feature size and spatial distance is used in the computation of the location probability values using a Bayes filter. The approach produces very good results, and is compared

with the method of using average percentage of matched features so as to validate its performance. The method is a viable solution to improve image transition in VT&R systems. It should be adopted to quadrotor systems such as the Ar.Drone model which do not have the capability of omnidirectional visual perception. The method, reported in [46], motivates this thesis to apply the feature-size relation with spatial distance to improve self-localization of the quadrotor.

Besides using matched features, Dame and Marchand [53], are interested in mutual entropy information between the current image and the reference images to perform the transition between two successive segments. This method can overcome the problems of occlusions and illumination variations. Nevertheless, the appearance of unexpected obstacles and the significant changes of environment negatively effect the accuracy. The method is performed in condition of 2D variation of camera rotation, which does not show potentials for 3D applications.

To summarize, self-localization in VT&R systems can be solved using appearance-based approaches. The most crucial system components are the feature detection, matching, and the method used for transition between two successive segments. These components should be properly designed to suit the specific limitations and complexities faced by quadrotor platforms. For the purpose of inheritance, the matched features generated during the self-localization step can be reused to compute motion control commands and obstacle avoidance. The next section discusses viable approaches to generate motion control commands after self-localization.

2.2 Motion Control Command

After performing self-localization, VT&R systems calculate the motion control commands (velocity commands) to control the quadrotor to reach the end of the segment

and eventually follow the desired route. This section discusses both pose-based and appearance-based approaches for motion-control-command generation.

2.2.1 Pose-based Approach

Self-localization with a pose-based approach provides the current pose of the quadrotor with respect to reference image frames. The metric positional errors between the current quadrotor frame and the reference image frame are easily computed in pose-based methods. Classical method such as Proportional-Integral-Derivative (PID) [5, 27, 28] is applied to control the quadrotor reach desired positions. However, the PID controller is susceptible to overshoot and exhibits instability particularly in vision-based control methods.

Work in reference [31], focuses on trajectory generation (or path planning) to smoothly reach the desired position while producing small reference errors for the PID to track. Optimal trajectory is computed based on considering the constraints of four rotors, flight time and the working environments. Advanced solutions proposed in [54, 55, 56, 56] apply *Model Predictive Control* (MPC) for motion-control-command generation. MPC position control has a prediction model, and system constraint information to assist the generation of optimal trajectories to reach the desired position. This can also incorporate obstacle avoidance which is important in practical applications. It is crucial that the method of motion-control-command generation is selected so as to satisfy onboard processing on quadrotors.

2.2.2 Appearance-based Approach

Appearance-based methods directly use the image feedback in order to generate motion control commands without explicitly estimating the position of the quadrotor. The motion-control-command generation is performed using specific features of the

image. For example, Bills et al [21] classify the type of indoor environment for quadrotor to perform autonomous flying. Using perspective cues appearing on single image feedback, the desired direction is estimated to keep the quadrotor following corridors and staircases of buildings, containing numerous long and straight parallel lines. The work in [57] applies the visual object tracking technique for the generation of quadrotor motion control commands. Processing the image feedback from forward-looking camera, the Ar.Drone quadrotor model is controlled to detect and track objects such as humans and cars. These methods address the problem of the motion-control-command generation through computer vision approaches with the assumption of specific and well-structure environments and conditions. In VT&R techniques, the environment is captured in the learning phase, and thus need not be well-structured or well-constrained to support autonomous navigation.

The works in [47, 2] propose Funnel Lane theory in order to command a ground vehicle follow the desired route. The method first qualitatively defines possible positions, where the vehicle can go straight, by the constraints of feature coordinates between the current image and the reference image. The generation of motion control commands is based on funnel-lane guided motion. If the vehicle locates itself outside the funnel lane, it will be commanded back to the funnel lane. Robust performance and inexpensive calculation motivate the thesis to apply this strategy into quadrotor navigation. However, the calculation of motion control commands needs to be extended for 3D cases.

2.3 Quadrotor Controller

2.3.1 Quadrotor Dynamic Modeling

The controller of the quadrotor receives motion control commands, which are used to produce the desired angular velocities for each rotor. Analyzing the motions of the quadrotor, the dynamic model has four inputs (thrust of the rotors), but contains six outputs (translational and rotational degrees of freedom). Thus, the quadrotor can be classified as an under-actuated system. The dynamic model exhibits highly nonlinear behaviors due to the aerodynamic effects and the 3D dynamics of the quadrotor.

The first step to design a quadrotor controller is building its dynamic model. Commonly used methods for modeling are Euler-Lagrange [58] and Newton-Euler [59], which finally produce a set of complicated nonlinear equations. Simplification steps are required to eliminate unnecessary elements such as rotational drag forces, blade flapping, surrounding wind velocities, and mechanical-electrical structure of motor. These assumptions are acceptable in an ideal working condition and advantageous for simulation. Conversely, these two methods find it difficult to expand to more complicated mechanical structures of the quadrotor such as small manipulators assembled on the quadrotor [60]. At this point, bond-graph modeling method [61] is considered. Bond-graph method is a graphically natural representation of a physical dynamic system. By generating equations in a systematic way as well as numerically integrating them, bond-graph method finds it more convenient for modeling complicated system. The thesis will apply bond-graph method to model quadrotor's motions, test the controller and simulate on 20Sim software.

2.3.2 Quadrotor Controller Designs

The design of a typical quadrotor controller follows the cascade architecture with two control layers [30]: position control and attitude control. Attitude controller is placed at a lower layer to control the quadrotor rotational angles (Roll - Pitch - Yaw), while position controller is placed at a higher level to manage trajectory control, position control or velocity control. Therefore, the higher level controller is equivalent to the motion-control-command generator in VT&R systems.

Starting from the linearization and simplification of a complicated dynamic model, PID control is applied into controlling position and attitude [5]. Six PID controllers should be used to control travel in X, Y, Z directions and Roll - Pitch - Yaw angles. This simple method is sufficient for simple applications [62] as well as possibly aggressive manoeuvres with support of a VICON^{TM} motion capture system [5]. As improvements, optimizing PID controller's gains is studied to enable quadrotor working in more complex conditions [63]. *Linear Quadratic Regulator* (LQR) is proposed as a design to control the full dynamic model using state feedback [64]. MPC controller [54, 55] shows potential to improve the performance of quadrotor controller by incorporating system constraints and accurate system prediction models. The output control efforts of MPC controller include the control actions according to the prediction of the future events, as well as optimizing steps basing on the quadrotor system constraints such as motor constraints. Nevertheless, MPC controller consumes considerable computational resources, which limits its applicability in onboard computers of quadrotors.

These above designs of controllers overcome the complication introduced by the nonlinearity of the quadrotor system model, by taking linear approximation and approximation of the underlying geometric properties of quadrotor. As an improved solution, T. Lee [65] proposes to apply geometric control theory in the design of quadrotor con-

troller. Geometric control theory [66], a coordinate-free control approach, converts the difficult problem of quadrotor control design into a straightforward linear problem. The nonlinear geometric controller concentrates on the errors of the rotational matrix, instead of attempting to access the errors of each degree of freedom as in PID controllers. Therefore, the controller structure is simple enough to implement into the embedded systems of the quadrotor. This work motivates the thesis to apply geometric control theory in VT&R system.

Chapter 3

Self-localization

About this chapter: The first section 3.1 describes initial notations. SURF features are chosen for navigation, which is explained in section 3.2. The thesis proposes three methods of self-localization (section 3.3, 3.4 and 3.5). Section 3.6 provides the experimental validations with offline and online processing.

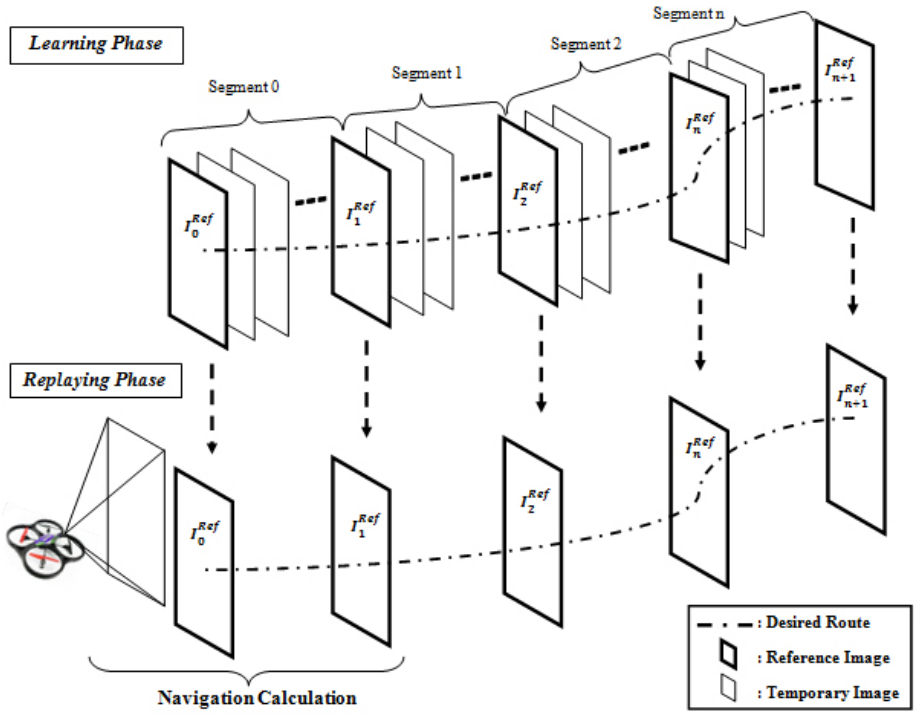

Figure 3.1: Two phases of VT&R aerial system: learning phase and replaying phase

3.1 Visual-Teach-and-Repeat Aerial System

The VT&R aerial system is equipped with monocular forward-looking camera that can be found in numerous other commercial and research quadrotor models [44, 3, 39]. Each reference image represents for one segment of the desired route as in Fig. 3.1. Some notations used in the thesis are presented as follows.

Ψ is the desired route of VT&R system.

$\{Seg_s | s \in \{1, 2, ..., n\}\}$ are several continuous segments constructing the desired route Ψ.

I_s^{Ref} is the reference image of the segment s in the desired route,
$\{I_s^{Ref} | s \in \{1, 2, ..., n+1\}\}$.

I^C is the current image of quadrotor.

$F_{s,j}^{Ref}$ is the j^{th} feature detected in the reference image I_s^{Ref}.

$F_{s+1,j}^{Ref}$ is the j^{th} feature detected in the reference image I_{s+1}^{Ref}.

F_j^C is the j^{th} feature detected in the current image I^C.

P_s^{Ref} is the position snapshot of the reference image I_s^{Ref}.

P_{s+1}^{Ref} is the position snapshot of the reference image I_{s+1}^{Ref}.

P^C is the current position of quadrotor for I^C.

The VT&R system is built on the observation of visual landmarks in the working environment. These visual landmarks appear as interest points (or features) on the image plane. These features are detected and matched between the current image and the reference images. In order to receive useful features, the landmark should contain following properties:

- **Being stationary and repeatable in the working environment for both learning phase and replaying phase**: Landmarks on moving objects such as other vehicles and humans should not become observed features. It is very complicated to navigate basing on these dynamic features. The feature matching technique can possibly reject the unwanted features of moving objects. However, fully rejecting the unwanted features of moving objects become an interesting topic to study. Additionally, the features should appear in both phases of VT&R technique. If some features only appear in the replaying phase, these features will possibly present landmark on obstacle.

- **Being robust in case of different lighting and visual noise**: The changes of working environment between the learning phase and the replaying phase are inevitable. However, when considering the changes of ambient lighting and the effects of noise, the landmark should be still robust enough for the vehicle to detect and to match its features.

- **Being distinctive in the working environment**: Because of the difference of the heading angles, the features detected in both the current image and the reference images should present only one landmark.

3.2 Feature Detection and Matching

3.2.1 Speeded-Up Robust Features

Scale- and rotation-invariant feature detectors, such as Speeded-Up Robust Features (SURF)[1] and Scale-Invariant Feature Transform (SIFT)[4] features, have recently demonstrated their useful applications in computer vision as well as robotic technologies. Although the SIFT features are capable of matching features across multiple images and invariant to the large scale changes, SIFT feature calculation is comparatively slow. Thus, the use of SIFT features is inappropriate for real-time applications such as online SLAM and visual odometry.

Generally, feature detection accesses the scale-space processing through image pyramid to reach the scale-invariant property of features. The scale-space consists of many octaves, where an octave contains many levels according to the increasing or decreasing of the scale σ values (Fig. 3.3). With multiple values of scaling σ parameters, Laplacian of Gaussian method (LoG), a scale-space filtering, is performed for different octaves to define local maxima across scale and space. In order to cope with the high computational cost associated with LoG, SIFT feature detection approximates LoG with Difference of Gaussians (DoG) (Fig. 3.2). And then these local maxima are defined by comparing one pixel in an image with its 8 neighbors, 9 pixels in the next level and 9 pixels in the previous level. These defined local maxima need to pass an evaluation step to become scale-invariant features. As a result, the approach of SIFT feature detection still requires large savings in memory to store the entire image pyramid. The advent of SURF features is partly inspired by the requirement of a speeded-up version for SIFT features. H. Bay et al [1] use Hessian box filter to approximate LoG (Fig. 3.3) in the support of integral images. More details about SURF and SIFT features can be found in the work, presented in [1] and [4] respectively.

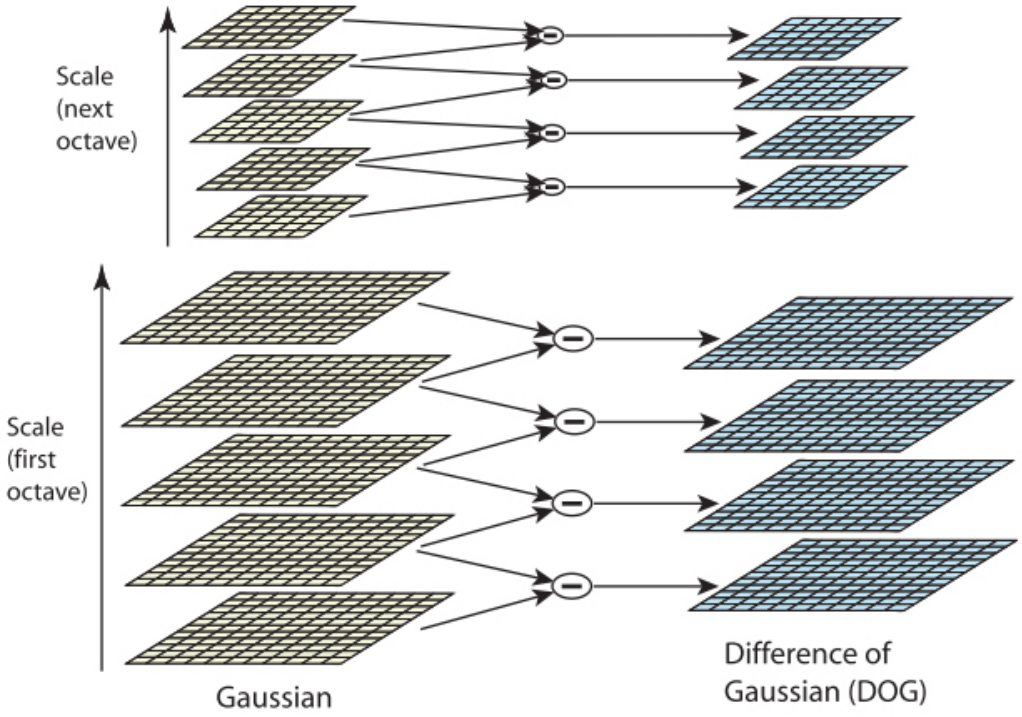

Figure 3.2: The scale-space filtering is performed in the approximation of LoG with DoG in SIFT feature detection [4]

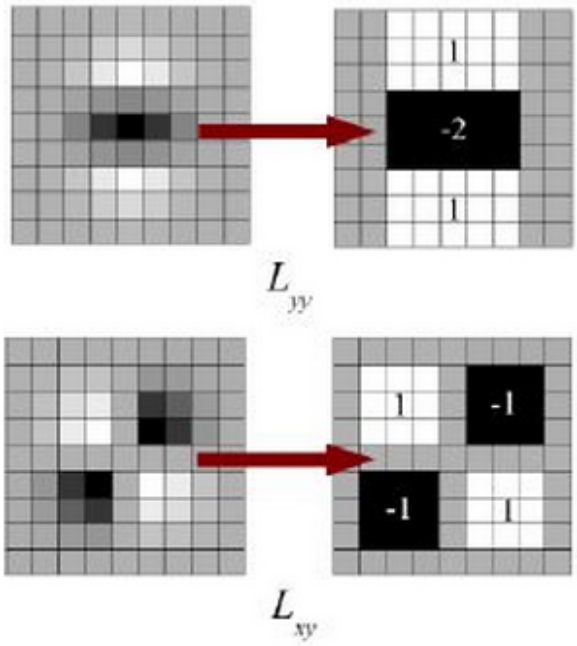

Figure 3.3: The approximation of LoG with Box filter in SURF feature detection [1]

The processing steps of SURF feature detection are summarized as follows:

- Step 1: The integral image [67] is computed to utilize the fast computation of the Hessian box filters.
- Step 2: The scale-space is divided into many octaves where an octave is defined as a series of filters as in Fig. 3.4. This approach allows the simultaneously processing of multiple levels in the scale-space pyramid without subsampling the image. The determinant response map is computed up to a certain scale-space level with the use of determinant of Hessian matrix, and then scale-normalized.

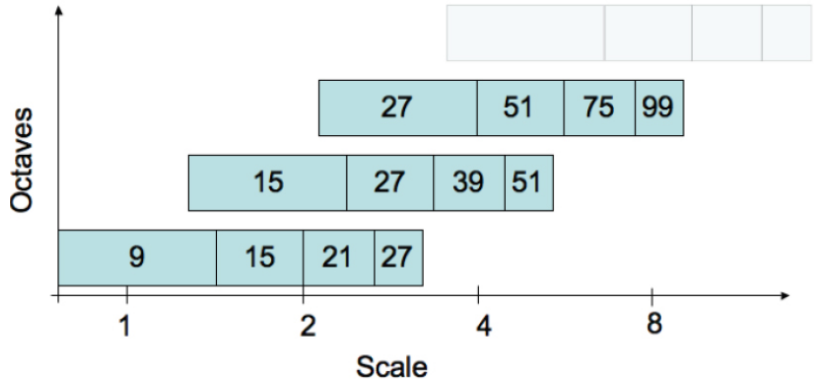

Figure 3.4: The construction of scale-space according to SURF feature detection [1]. The scale-space is represented by 3 octaves with different filter side lengths

- Step 3: With these determinant response maps, non-maximal suppression in a $3 \times 3 \times 3$ neighborhood is performed to localize interest points in the image.
- Step 4: As the scaling difference between the first levels of every octave is relatively large, the scale-space interpolation of the nearby data is performed to provide SURF features at the correct scaling value σ.
- Step 5: The orientation of each SURF feature is defined by the use of Haar Transforms [1].

In the view of VT&R aerial vehicle design, the scale-invariant and rotation-invariant properties of SURF features provide considerable benefits as quadrotor maneuvers in 3D and is subjected to image noise and viewpoint disparity. The work [46] proposes to use SIFT features for the VT&R ground system. Although SIFT and SURF features have the same principals in terms of the definitions, SURF features show faster calculation than SIFT features, and still can satisfy the accuracy and stability for self-localization of the quadrotor. The scale-space representation of the SURF features [1, 68, 69, 70] is approximately estimated to decrease the average calculational time. Besides these advantages, another noticeable property of SURF feature is the relation between spatial distance and the feature size (or scale). When the vehicle moves closely to the landmarks, the size of the landmark features tends to increase. For example, the position snapshot of I^C locates between the position snapshot of I_s^{Ref} and the position snapshot of I_{s+1}^{Ref} as in Fig. 3.5. Matched features of I^C should have larger size than the features of I_s^{Ref} but smaller size than the features of I_{s+1}^{Ref}. Therefore, this thesis has chosen SURF features for the VT&R aerial system.

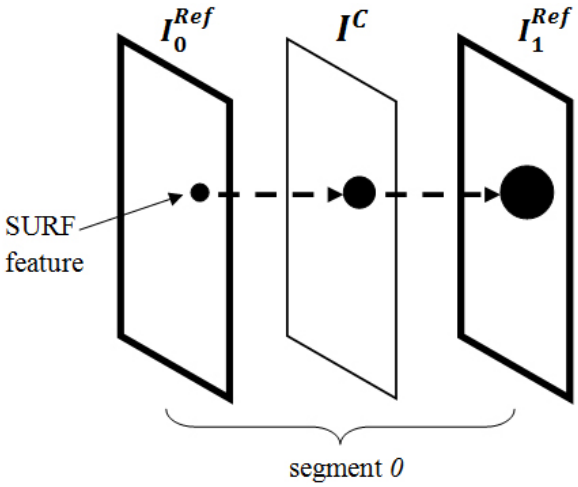

Figure 3.5: The SURF feature size relation with spatial distance

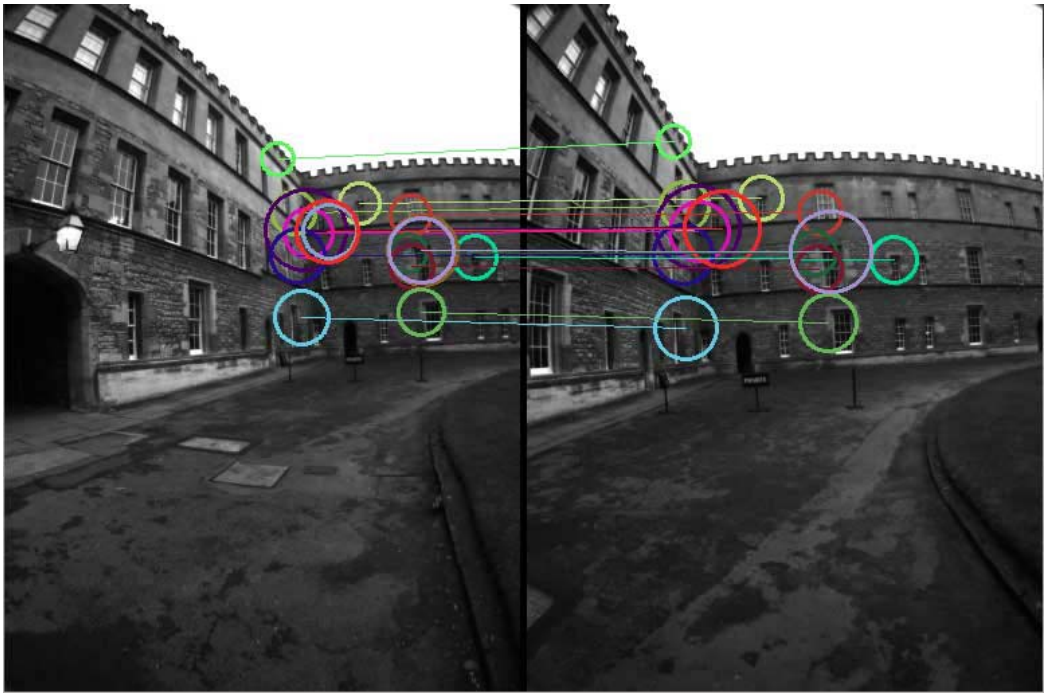

Figure 3.6: Test the feature-size relation on images from New College Database. LEFT: I_s^{Ref}, RIGHT: I^C

3.2.2 Binary Robust Independent Elementary Feature

After detecting SURF features in the current image and the reference images, feature matching can be performed by SURF feature descriptors. However, the descriptor of SURF features is still calculated in the form of floating-point numbers as well as contains 64-dim descriptor vector. As a result, the memory footprint of one SURF descriptor requires at least 256 bytes. Considering computational capabilities of the embedded systems on the quadrotor, the use of SURF descriptor will take considerable memory for thousands or hundreds of features. Hence, the thesis proposes to use Binary Robust Independent Elementary Feature (BRIEF) descriptor in order to improve the efficient computation of matching feature step [71, 72].

BRIEF descriptor is one of many binary descriptors, which considering the statistical properties of image region around the detected feature to perform feature matching.

BRIEF descriptor has shown fast computation and compact presentation. Additionally, its processing is simple to implement. To summarize, the processing steps of BRIEF descriptor are presented:

- Step 1 (Pre-processing): Prior to computing BRIEF descriptors, the image patch is smoothed against noise by gaussian or box filter. Notably, if this step has been performed when detecting SURF feature, it can be skipped.

- Step 2 (Binary Feature Descriptor Extraction): Points p_1, p_2 are randomly selected in the region around the feature, which is detected by other feature detection technique such as SURF, SIFT. A test $\tau(p_1, p_2)$ is computed [71]:

$$\tau(p_1, p_2) := \begin{cases} 1, & \text{if} I(p_1) < I(p_2) \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

  The BRIEF descriptor d_n is computed in the form of the n-dimensional bit string with $n \in \{128, 256, 512\}$ [71]:

$$d_n := \sum_{i=1}^{n} 2^{i-1} \tau(p_1^i, p_2^i) \tag{3.2}$$

- Step 3 (Binary Descriptor Matching): After computing BRIEF descriptors for all features by step 2, the matching between these BRIEF descriptors is performed by Hamming distance [73].

Each segment of the desired route is represented by the reference image as in Fig. 3.1. Image processing for feature detection, feature description and feature matching is performed in the support of OpenCV library [74]. Self-localization is performed by analyzing the reference images to determine the current segment where the quadrotor is locating. Three methods of self-localization are proposed as follows:

- Method I (section 3.3) is a traditional method, which uses matches of the SURF features in the current image with the SURF features in the reference images to compute the probability value of each segment in the desired route. The segment receives the best probability value of matched features will be chosen as the current segment.

- Method II (section 3.4) a development of method I in the additional use of the SURF feature-size relation with spatial distance for outlier rejection when matching SURF features.

- Method III (section 3.5) is an adaptation and implementation of self-localization method, proposed in [46], in case of monocular front camera. In other words, the difference between method III and method II is at the probability model of calculation following Bayesian probability inference.

3.3 Method I

Normally, self-localization is popularly performed by comparing the number of matched features between the current image and the reference images [44, 45, 46]. After matching SURF features between images by its descriptors, the results contains considerable incorrect matches. Some filtering methods of outlier rejection are employed in order to provide the reliable matched features for self-localization. One of filtering methods is RANdom SAmple Consensus (RANSAC), which eliminates incorrect feature matches by finding homographies between two images [75]. Matching SURF features of method I is presented step by step as follows:

Algorithm 1: Matching features of method I

Input: $\{F_j^C\}$, $\{F_{s,j}^{Ref}\}$, I^C and $\{I_{s,j}^{Ref}\}$

Output: Matching SURF Feature Result: $\{F_{s,j}^{M,3*}\}$

1 Calculate Descriptor: $\{F_j^C.descriptor\}$ and $\{F_{s,j}^{Ref}.descriptor\}$;

2 Match feature descriptors between $\{F_j^C.descriptor\}$ and $\{F_{s,j}^{Ref}.descriptor\}$ in order to produce $\{F_{s,j}^{M,1}\}$;

3 Eliminate incorrect matched results by its distance (in feature space) to have $\{F_{s,j}^{M,2}\}$: $\{F_{s,j}^{M,1}.distance\} > 0.5 * Mean(\{F_{s,j}^{M,1}.distance\})$;

4 RANSAC feature filtering of $\{F_{s,j}^{M,2}\}$ to have $\{F_{s,j}^{M,3*}\}$;

SURF features of I^C are matched with those of I_s^{Ref} to have $\{F_{s,j}^{M,3*}\}$ and $\{F_{s,j}^{Ref}\}$. SURF features of I^C are matched with those of I_{s+1}^{Ref} to have $\{F_{s+1,j}^{M,3*}\}$ and $\{F_{s+1,j}^{Ref}\}$. The probability calculation of self-localization is performed by Eq. 3.3 in order to estimate the segment where the vehicle is locating. The segment receiving the maximum value of percent estimation will provide its reference image for navigation.

$$MethodI\{Seg_s\} = \frac{1}{2}\left(\frac{\{F_{s,j}^{M,3*}\}}{\{F_{s,j}^{Ref}\}} + \frac{\{F_{s+1,j}^{M,3*}\}}{\{F_{s+1,j}^{Ref}\}}\right) \tag{3.3}$$

3.4 Method II

Method II intends to improve method I while using additional condition of the relation between spatial distance and the size of feature. Some incorrect features are eliminated in term of this feature-size relation. It means that the $\{F_j^C\}$ has larger size than $\{F_{s,j}^{Ref}\}$ but smaller size than $\{F_{s+1,j}^{Ref}\}$ as in Fig. 3.5. Reasonably, $\{F_{s,j}^{Ref}\}$ and $\{F_{s+1,j}^{Ref}\}$ are detected at the start position and the end position of the segment. Any features of $\{F_j^C\}$ should have the value of size between these constraints. Matching SURF feature of method II is presented as in algorithm 2.

Algorithm 2: Matching features of method II

Input: $\{F_j^C\}$, $\{F_{s,j}^{Ref}\}$, I^C and $\{I_{s,j}^{Ref}\}$
Output: Matching SURF Feature Result: $\{F_{s,j}^{M,4}\}$
1 Calculate Descriptor: $\{F_j^C.descriptor\}$ and $\{F_{s,j}^{Ref}.descriptor\}$;
2 Match feature descriptors between $\{F_j^C.descriptor\}$ and $\{F_{s,j}^{Ref}.descriptor\}$ in order to produce $\{F_{s,j}^{M,1}\}$;
3 Eliminate incorrect matched results by its distance (in feature space) to have $\{F_{s,j}^{M,2}\}$: $\{F_{s,j}^{M,1}.distance\} > 0.5 * Mean(\{F_{s,j}^{M,1}.distance\})$;
4 Check the feature-size relation with spatial distance to have $\{F_{s,j}^{M,3}\}$: $\{F_{s,j}^{M,2}.size\} >= \{F_{s,j}^{Ref}.size\}$ (or $\{F_{s+1,j}^{M,2}.size\} <= \{F_{s+1,j}^{Ref}.size\}$) ;
5 RANSAC feature filtering of $\{F_{s,j}^{M,3}\}$ to have $\{F_{s,j}^{M,4}\}$;

SURF features of I^C are matched with those of I_s^{Ref} to have $\{F_{s,j}^{M,4}\}$ and $\{F_{s,j}^{Ref}\}$. SURF features of I^C are matched with those of I_{s+1}^{Ref} to have $\{F_{s+1,j}^{M,4}\}$ and $\{F_{s+1,j}^{Ref}\}$. Then, the probability calculation of self-localization is performed by Eq. 3.4 in order to estimate the segment of vehicle location. The segment receiving the maximum value of percent estimation becomes the segment where the vehicle is locating.

$$MethodII\{Seg_s\} = \frac{1}{2}\left(\frac{\{F_{s,j}^{M,4}\}}{\{F_{s,j}^{Ref}\}} + \frac{\{F_{s+1,j}^{M,4}\}}{\{F_{s+1,j}^{Ref}\}}\right) \tag{3.4}$$

Results of matching SURF feature step-by-step is presented in Fig. 3.10.

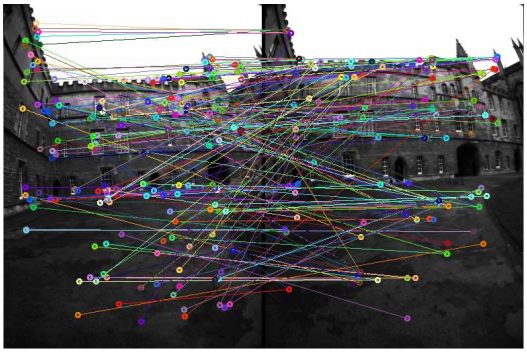

Figure 3.7: Results of matching features after step 2

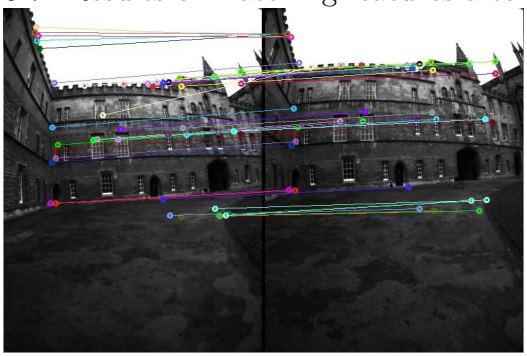

Figure 3.8: Results of matching features after step 3

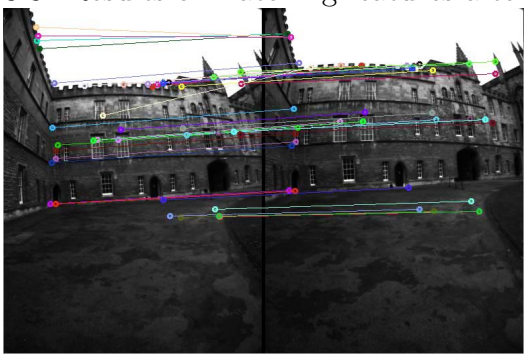

Figure 3.9: Results of matching features after step 4

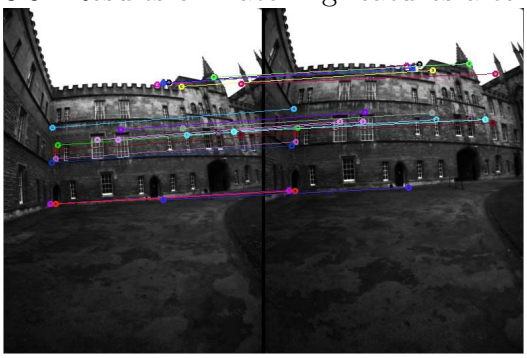

Figure 3.10: Results of matching features after step 5

3.5 Method III

Method III is an adaptation and implementation of self-localization method, proposed in the work [46], into monocular camera. Method III reuses the SURF feature matching technique in method II. Segment estimation is calculated by Bayes' theorem. Basically, in order to perform self-localization, two dependent events are considered as number of features, $\{F_j^M\}$, and segment estimation, $\{Seg_s\}$. The conditional probability of their relationship obeys Bayes' theorem [76]:

$$p(Seg_s|\{F_j^M\}) = \frac{p(\{F_j^M\}|Seg_s)p(Seg_s)}{p(\{F_j^M\})} \tag{3.5}$$

$$p(Seg_s|\{F_j^M\}) \propto p(\{F_j^M\}|Seg_s)\ p(Seg_s) \tag{3.6}$$

$p(Seg_s)$ presents the belief of specific segment s. This information is available if the quadrotor global position during localization is provided. As the VT&R system's configuration excludes global positioning system, $p(Seg_s)$ is set to be the same for every segment. Additionally, $p(\{F_j^M\})$ is set to 1 or eliminated since the maximum value of $p(Seg_s|\{F_j^M\})$ is considered. $p(\{F_j^M\}|Seg_s)$ is calculated by assuming that the quadrotor is at segment s of the desired route between two reference images I_s^{Ref} and I_{s+1}^{Ref} in order to define $p(Seg_s|\{F_j^M\})$.

$$p(\{F_j^M\}|Seg_{s=i}) = p(\{F_j^M\}|Seg_{s>i-1}).p(\{F_j^M\}|Seg_{s<i+1}) \tag{3.7}$$

$\{F_j^C\}$ are matched with $\{F_{s,j}^{Ref}\}$ to produce matched features $\{F_{s,j}^{M,4}\}$. $\{F_j^C\}$ are matched with $\{F_{s+1,j}^{Ref}\}$ to produce matched features $\{F_{s+1,j}^{M,4}\}$. Notably, when matching SURF feature, the feature-size relation is added in considering the $\{I_s^{Ref}\}$ or $\{I_{s+1}^{Ref}\}$.

Probability calculation of $p(\{F_{s,j}^{M}\}|Seg_{s>i-1})$ and $p(\{F_{s,j}^{M}\}|Seg_{s<i+1})$ are computed:

$$p(\{F_{j}^{M}\}|Seg_{s>i-1}) = \frac{|\{F_{s,j}^{M,4}\}|}{|\{F_{s,j}^{Ref}\}|} \tag{3.8}$$

$$p(\{F_{j}^{M}\}|Seg_{s<i+1}) = \frac{|\{F_{s+1,j}^{M,4}\}|}{|\{F_{s+1,j}^{Ref}\}|} \tag{3.9}$$

As a result, $p(Seg_s|\{F_j^M\})$ is defined as Eq. 3.10.

$$p(\{F_{j}^{M}\}|Seg_{s=i}) = \frac{|\{F_{s,j}^{M,4}\}|}{|\{F_{s,j}^{Ref}\}|}\frac{|\{F_{s+1,j}^{M,4}\}|}{|\{F_{s+1,j}^{Ref}\}|} \tag{3.10}$$

$$MethodIII\{Seg_s\} = p(\{F_{j}^{M}\}|Seg_{s=i}) \tag{3.11}$$

The segment receiving maximum percent value of estimation $MethodIII\{Seg_s\}$ will provide data as reference image for navigation.

Algorithm 3: Matching features of method III

Input: $\{F_j^C\}$, $\{F_{s,j}^{Ref}\}$, I^C and $\{I_{s,j}^{Ref}\}$

Output: Matching SURF Feature Result: $\{F_{s,j}^{M,4}\}$

1 Calculate Descriptor: $\{F_j^C.descriptor\}$ and $\{F_{s,j}^{Ref}.descriptor\}$;

2 Match feature descriptors between $\{F_j^C.descriptor\}$ and $\{F_{s,j}^{Ref}.descriptor\}$ in order to produce $\{F_{s,j}^{M,1}\}$;

3 Eliminate incorrect matched results by its distance (in feature space) to have $\{F_{s,j}^{M,2}\}$: $\{F_{s,j}^{M,1}.distance\} > 0.5 * Mean(\{F_{s,j}^{M,1}.distance\})$;

4 Check the feature-size relation with spatial distance to have $\{F_j^{M,3}\}$: $\{F_{s,j}^{M,2}.size\} >= \{F_{s,j}^{Ref}.size\}$ or $\{F_{s+1,j}^{M,2}.size\} <= \{F_{s+1,j}^{Ref}.size\}$;

5 RANSAC feature filtering of $\{F_{s,j}^{M,3}\}$ to have $\{F_{s,j}^{M,4}\}$;

3.6 Experiments of Self-Localization Technique

3.6.1 Offline Testing

3.6.1.1 Database Acquisition at MUN

The thesis performs the learning phase of VT&R technique by collecting images in the Engineering building. Two databases are collected by *Intelligent Systems Lab* (ISLab) and listed as follows:

- **Route A**: Second Floor of Faculty of Engineering and Applied Science including Engineering Lobby, Cafeteria and Engineering Lounge, images were collected at 11:00 am on May 17, 2014, in cloudy weather. Image details are, dimensions: 2592x1936, resolution: 72 dpi, bit depth: 24, environment: indoor in Fig. 3.11.

- **Route B**: Outside Engineering Building, along Kerwin PI road, images are collected at 3:00 pm on May 17, 2014, in sunny weather. Image details are, dimensions: 2592x1936, resolution: 72 dpi, bit depth: 24, environment: outdoor in Fig. 3.12.

3.6.1.2 Databases Collected by Other Research Groups

Three methods are also tested with databases collected by other research groups:

- **CoSy Localization Database**[77] (COLD-Saarbrucken): Images were acquired at Language Technology Laboratory, German Research Center for Artificial Intelligence in Saarbrucken, Germany, on ActivMedia PeopleBot robotic platform. Image details are listed, dimensions: 640x480, resolution: 96 dpi, bit depth: 24, environment: indoor office, year: 2008, version: COLD-Saarbrucken - Part B - night condition. Robot travels with the speed 0.220 m/s. The analysis uses 160 images.

- **New College Database** [78]: Images were acquired at New College building by Oxford Mobile Robotics Group, University of Oxford, UK, on Segway robotic platform in Epoch A Campus at New College, Oxford, United Kingdom. The working environment is outdoor with sunny weather in May 2009. Image details are list, dimensions: 384x512 pixels, resolution: 72 dpi, bit depth: 24, environment: outdoor. 192 images at camera-0 are used.

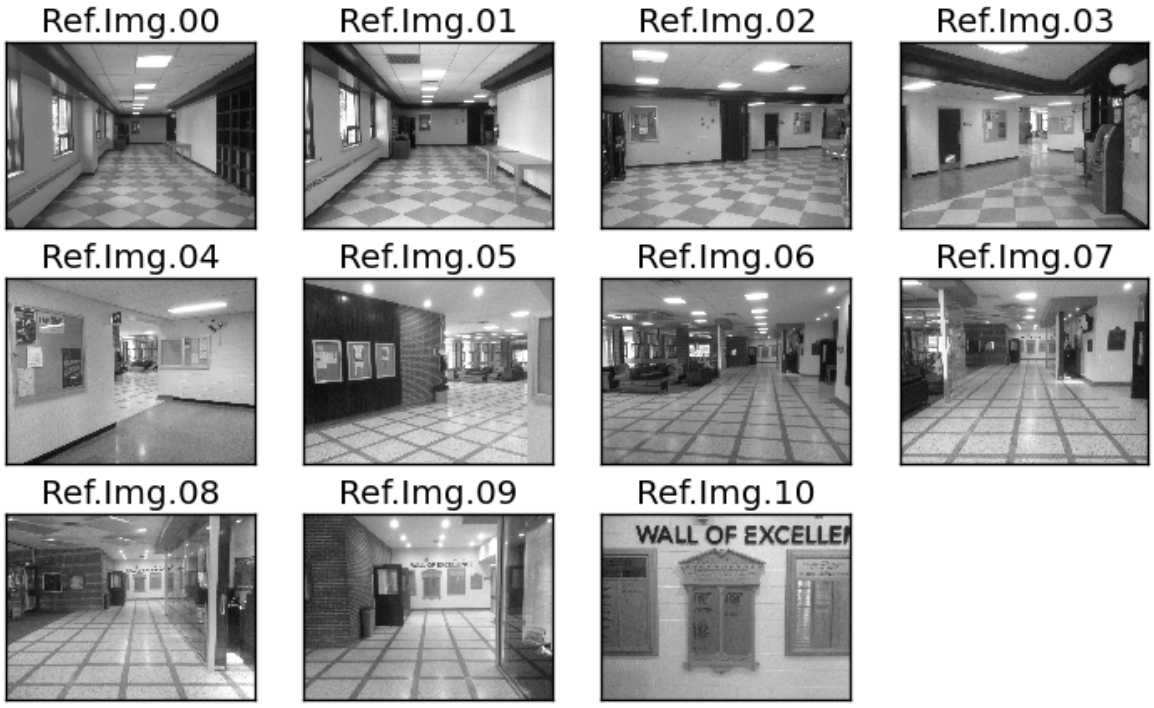

Figure 3.11: Reference images of the route A - Second Floor of Engineering building

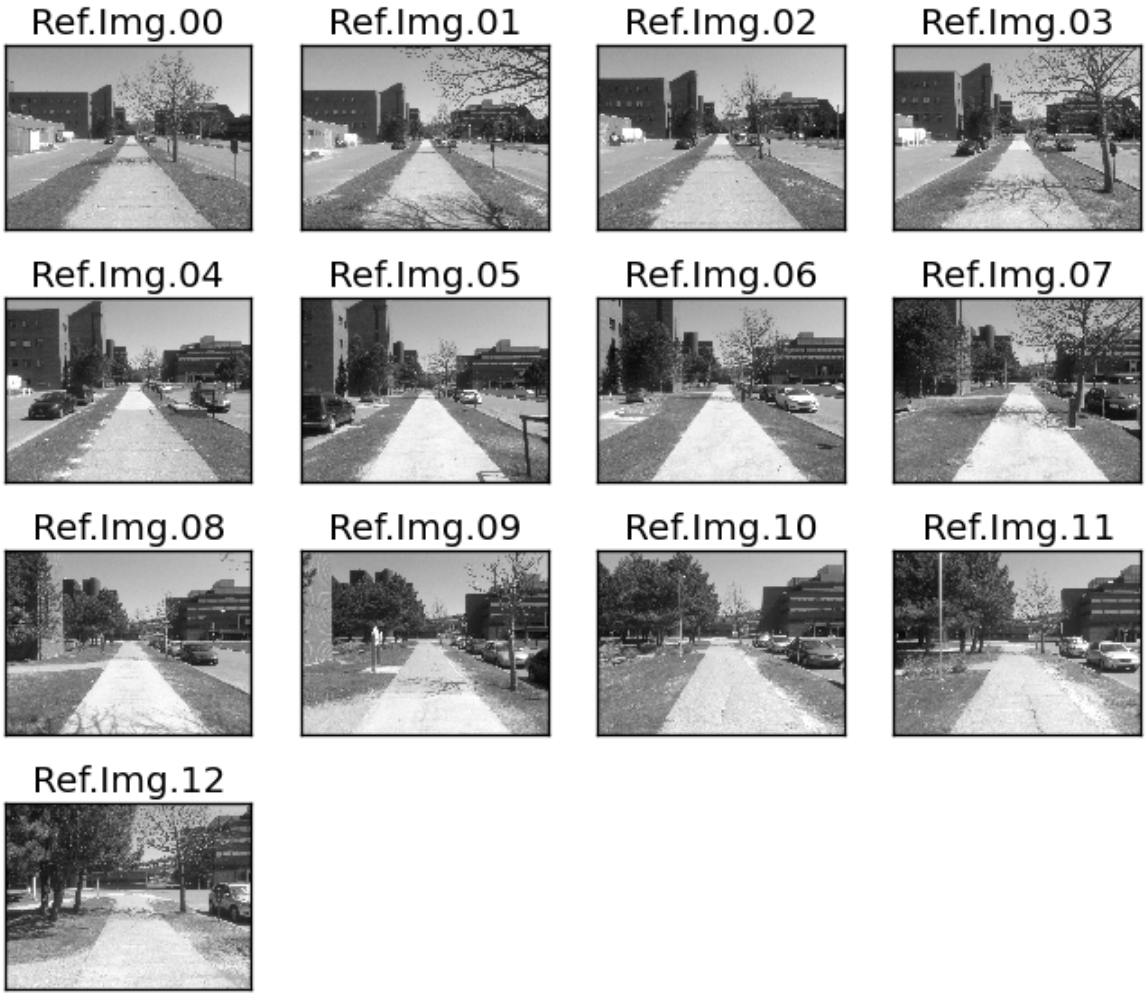

Figure 3.12: Reference images of the route B - Outside of Engineering building

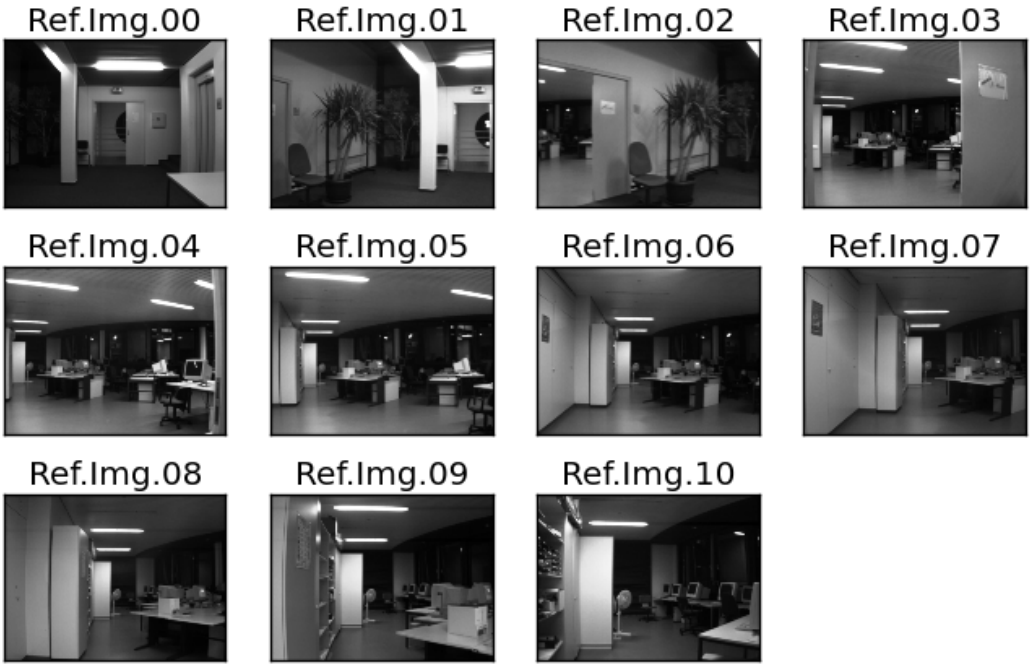

Figure 3.13: Reference images from COLD database

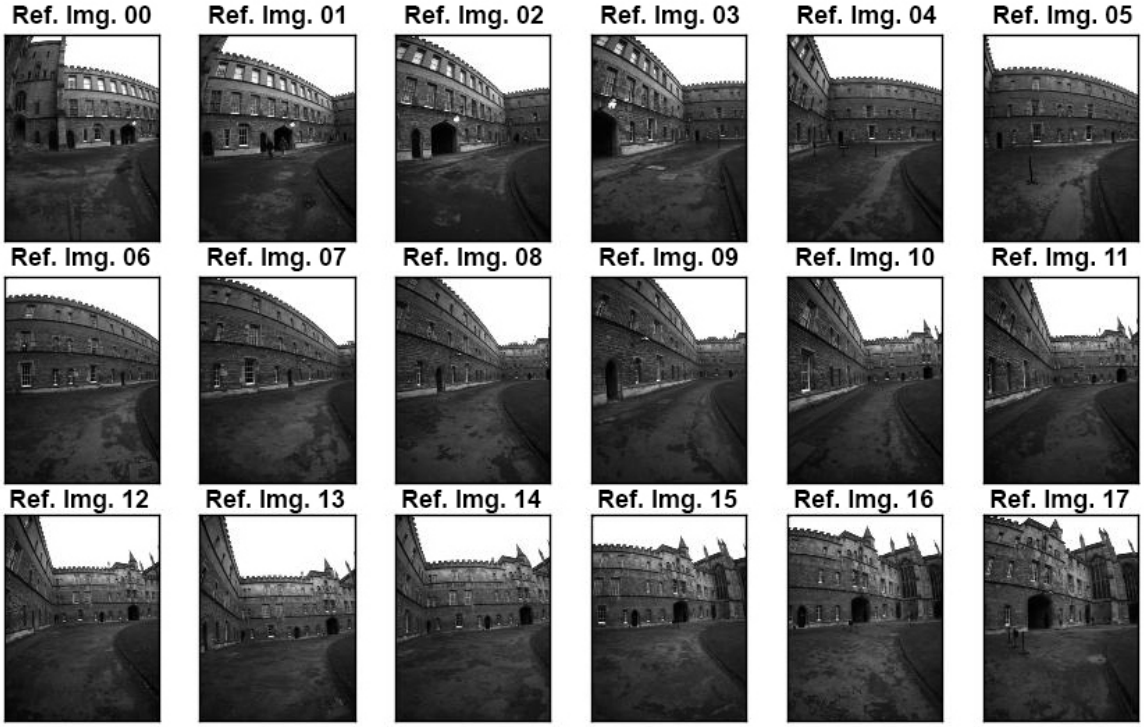

Figure 3.14: Reference images from New College database

3.6.1.3 Results of Offline Testing

Tested databases are collected when the vehicle is moving. Therefore, the images of databases show the motion blur in their content. When processing each database, some images are chosen as reference images, and the other images between chosen reference images are used to test the performance of self-localization technique (Fig. 3.15). The reference images are chosen so that the acquisition frequency of set is sufficient to achieve self-localization, where the distance of the segment is at least 1m. Three methods will be tested in using the same databases and the set of m tested images $\{I_i^C\}$ in order to have similar operating conditions.

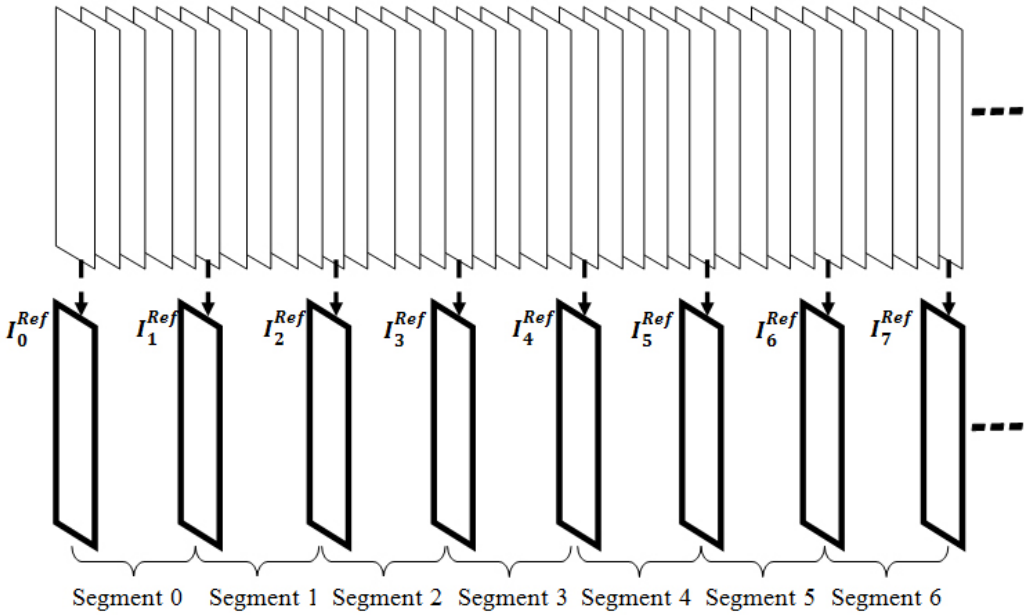

Figure 3.15: Processing database to choose reference images of the desired route

The results of estimating location are compared with the employed ground-truth data in order to define the percentage of success in each of the method. Additionally, entropy measurement is calculated as in Eq. 3.12 to measure the uniformity of the probability distribution in three proposed methods [46]. The sim in Eq. 3.12 is a output function calculated by method I or method II or method III. In comparison,

the method performing better has lower entropy value.

$$Entropy(\{I_i^C\}) = -\sum_{i=0}^{m} \text{sim}(I_i^C) \log_2 \text{sim}(I_i^C) \tag{3.12}$$

In addition, the thesis also observes the different rate between the current image and the reference images. The difference rate is computed in the form of correlation norm matching as Eq. 3.13 [74]. A value of D, closer to 1, represents a small difference or a good match, while a small D, closer to 0, is significantly different or a poor match. R, being reference image, and C, being current image, have the same size $w \times h$; $x' = 0...(w-1)$ and $y' = 0...(h-1)$.

$$D = \frac{\sum_{x',y'}(R(x',y').C(x+x',y+y'))}{\sqrt{\sum_{x',y'} R(x',y')^2 . \sum_{x',y'} C(x+x',y+y')^2}} \tag{3.13}$$

Table 3.1: Experiment results with multiple databases

		Method I	Method II	Method III
Route A (2592x1936 pixels)	FT	31/51	19/51	21/51
	PS	39.22%	62.75%	58.82%
	AE	3.2224	3.1089	2.4035
Route B (2592x1936 pixels)	FT	15/60	04/60	05/60
	PS	75%	93.33%	91.66%
	AE	3.3224	3.2327	2.3955
COLD Database (640x480 pixels)	FT	67/151	37/151	51/151
	PS	55.63%	75.50%	66.22%
	AE	2.8448	2.7503	1.9643
New College Database (384x512 pixels)	FT	46/175	7/175	20/175
	PS	73.71%	96%	88.57%
	AE	3.5758	3.3905	2.1477

Note: FT: Failure number over Trials; PS: Percent of Success; AE: Average Entropy;

Table 3.1 presents the results of self-localization by all three methods. Method II and method III provide better in estimating the current segment than method I. The use of the SURF feature size relation with spatial distance has eliminated some

incorrect features to improve the performance of self-localization. Method II has smallest failure numbers over trials. However, the estimating accuracy of method II is not much different as compared to method III, and suffers higher entropy evaluation. The performance of self-localization in segment 5 of New College database is analyzed. The segment 5 contains 10 current images, 2 reference images 5 and 6 (Fig. 3.16). Fig. 3.17 expresses the results of estimation (LEFT) and entropy (RIGHT) on the current images of segment 5 in New College database (3.16), while Fig. 3.18 expresses difference rate between current images (I^C) and reference images. In Fig. 3.18, Y-axis is the difference rate between I^C and I_{s+1}^{Ref}; X-axis is the different rate between I^C and I_s^{Ref}; the blue sign is successful self-localization and the red sign is unsuccessful self-localization. When the vehicle is at position 1 (Current Image 1), closest to the reference image I_s^{Ref}, the value of the estimation reaches the highest value. When the vehicle moves forward, far from P_s^{Ref} position, the value of the estimation tends to decrease. When the vehicle moves closed to P_{s+1}^{Ref} the value of the estimation tends to increase. Fig. 3.18 can show the changes of the different rate when the vehicle is moving from P_s^{Ref} to P_{s+1}^{Ref}. The property happens to be the same with the entropy diagram. These results are different from those of the work [46], where the centre position receives the maximum estimation. The difference is caused by the type of camera used in the system. The work [46] uses camera with hyperbolic mirror providing front and back images, while this thesis uses a monocular camera providing front images.

3.6.2 Online Testing

The simulation of experiments is conducted on ROS Fuerte [29] with Gazebo simulator, Linux Ubuntu 12.04. In Gazebo simulator, an indoor environment and Ar.Drone quadrotor model are created to sufficiently validate the performance of three proposed

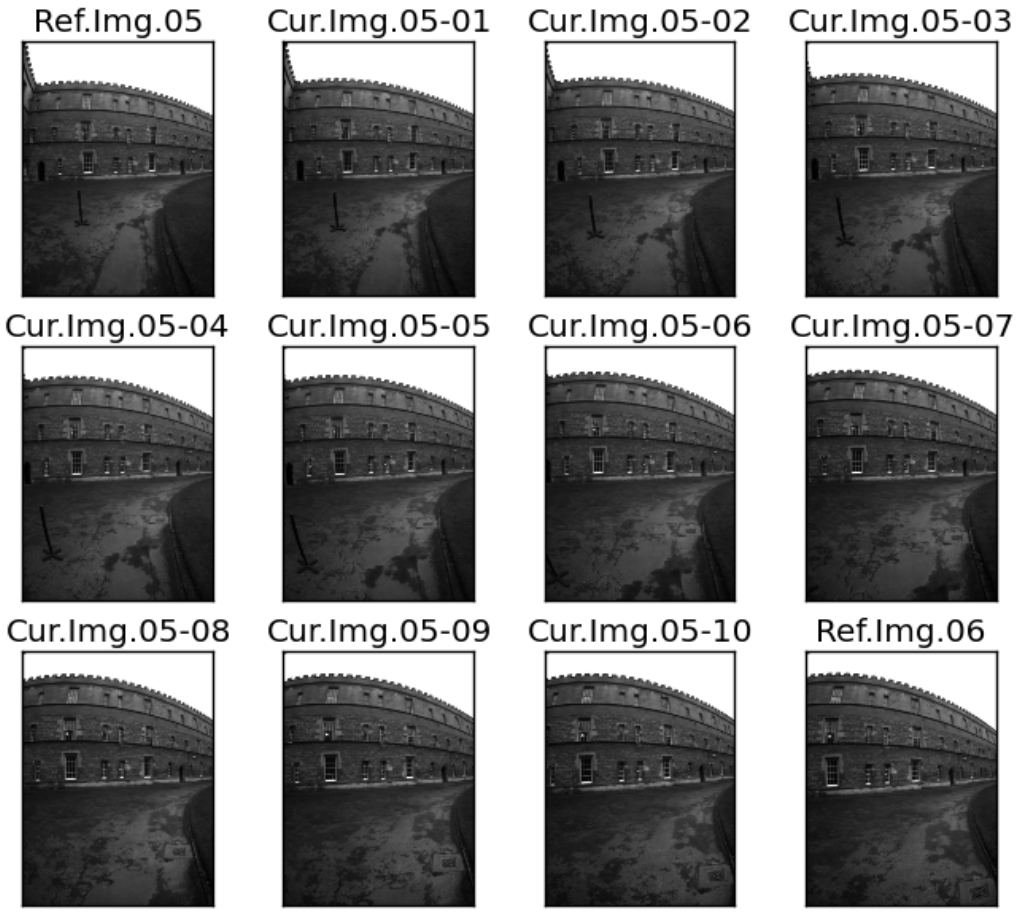

Figure 3.16: Reference images of segment 5, New College database

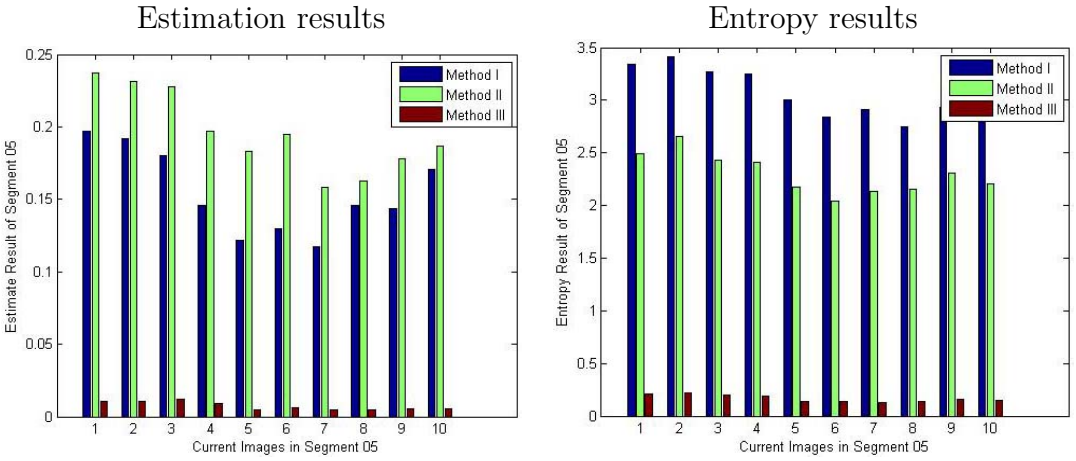

Figure 3.17: Estimation results and entropy results of Segment 5 - New College database. Estimation results are the calculation of $MethodI\{Seg_5\}$, $MethodII\{Seg_5\}$ and $MethodIII\{Seg_5\}$

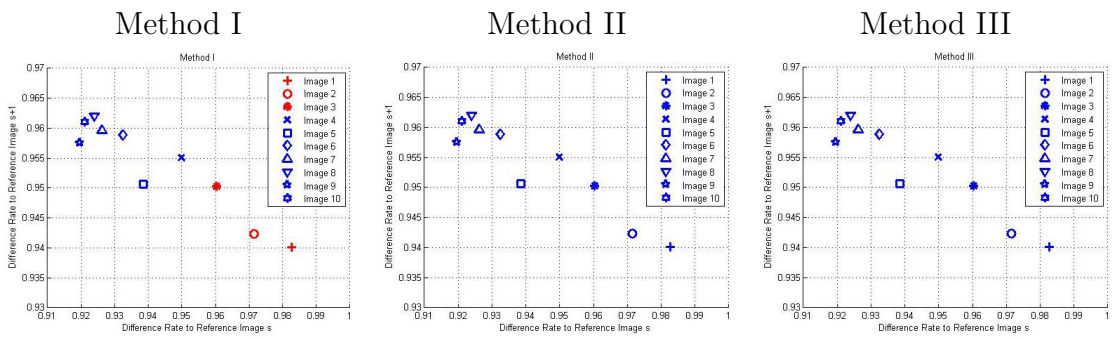

Figure 3.18: Estimation results by different rates of the current images in segment 5 - New College database

methods. Visual markers of the desired route (ground truth, segment notations) are made in Rviz application (Fig. 3.21) in order to check the error of the current-segment estimation. Fig. 3.20 describes the ROS system architecture of simulation. *CvBridge* node is defined to transfer the ROS image messages to OpenCV images in order to utilize OpenCV library for image processing. *ISLab Self-Localization* node is designed to perform self-localization on image feedback from quadrotor and reference images. *ISLab Controller* is the controller of quadrotor. In the learning phase, quadrotor is controlled along a desired circle route so that it can collect reference images at specific positions as in Fig. 3.19. The desired circle route of simulation has 8 segments, 8 reference images and 1.25m radius (Fig. 3.19). The center of the circle is at $(0, 0, 1.5)$. Quadrotor is manually controlled to follow the desired route with minimum errors of position to test the methods of self-localization.

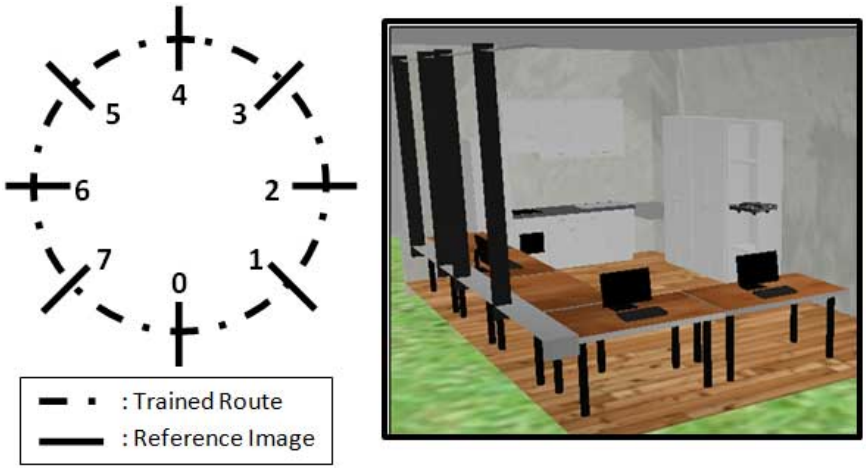

Figure 3.19: LEFT: The circle desired route; RIGHT: Gazebo simulator

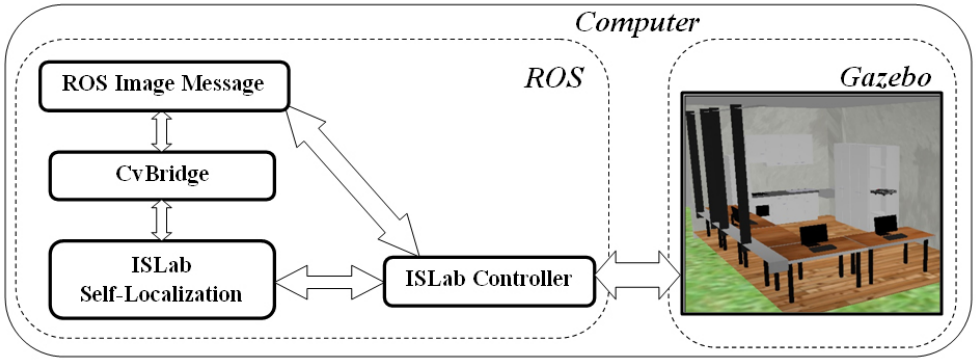

Figure 3.20: The ROS system architecture of simulation

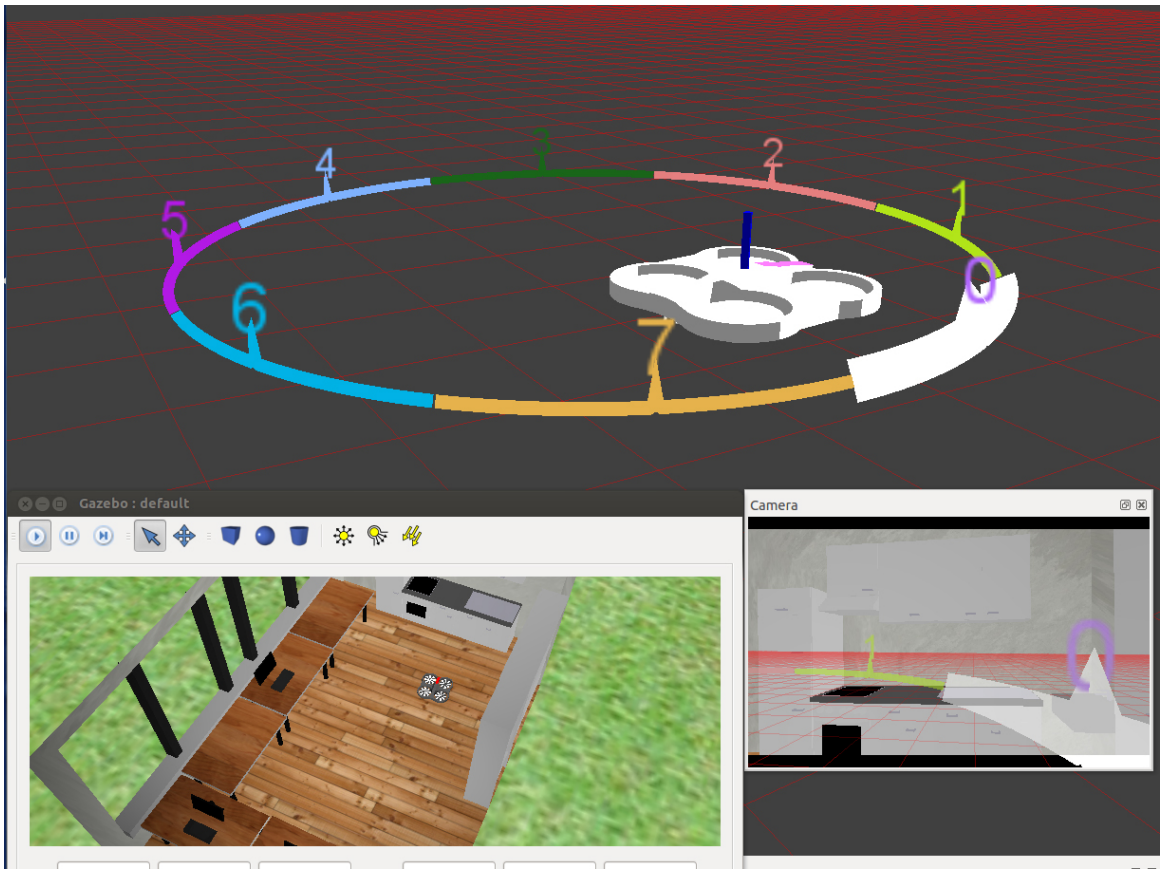

Figure 3.21: ROS-Gazebo simulation with Rviz application. Top: Rviz window shows Ar.Drone quadrotor and the desired route with its segments. Bottom-Left: Gazebo simulator window. Bottom-Right: the image feedback of quadrotor front camera.

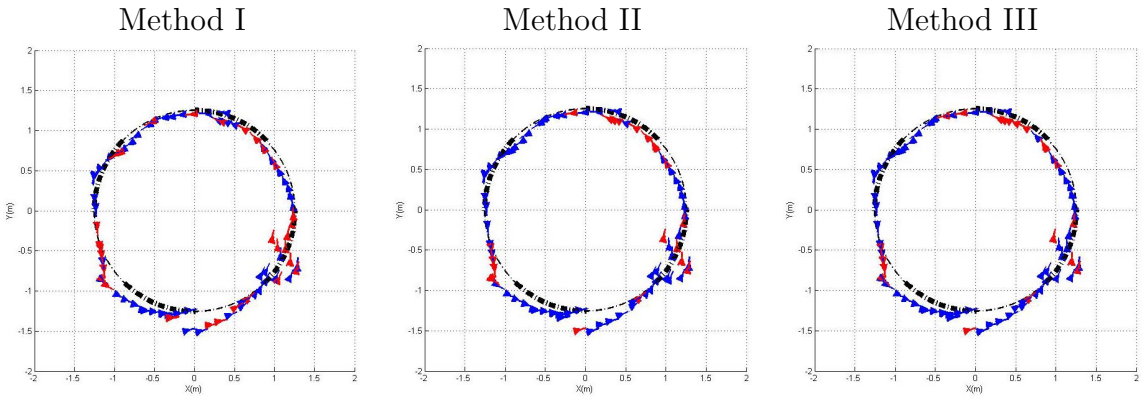

Figure 3.22: Online testing of three methods in simulation

The transition between segments as well as the errors of online self-localization are observed to evaluate the online performance of three methods. Fig. 3.22 presents the online performance of three methods. The dash-dot line is the desired route. Quadrotor with its heading angle is represented by triangle in blue color (true estimating location) and red color (false estimating location). Method II and III also provide better performance than method I. Remarkably, the use of feature-size relation with spatial distance improves the performance in the transition between segments. When quadrotor is traveling in segment 3, both of three methods provide incorrect estimations because of very few features in environment. This phenomenon also explains why the outdoor databases (route B and New College) receives higher percentage of success than indoor databases (route A and COLD). As method III is designed on Bayes' rule, its estimation produces very small errors. For example, if quadrotor is locating in segment 5, its incorrect estimation is segment 4 or 6.

Chapter 4

Qualitative Motion Control Command Based on Funnel Lane Theory

About this chapter: In the replaying phase, after performing self-localization, the reference images of the current segment are loaded to command the quadrotor to follow the desired route. The motion control command algorithm is built on Funnel Lane theory, described in section 4.1. The desired route combined with Funnel Lane theory is presented in section 4.2. The motion control command calculation, named Funnel-lane 3D motion control algorithm, is described in section 4.3. The works are published in *The 2014 International Conference on Unmanned Aircraft Systems* (ICUAS 2014) [49] and *The 2014 Conference on Computer and Robot Vision* (CRV 2014) [79]. In ICUAS 2014 conference, Funnel-Lane navigation technique is proposed and first results of adapting Funnel Lane theory into route following control of quadrotor using Kanade-Lucas-Tomasi (KLT) features. The paper of CRV 2014 conference is the development of the work of ICUAS 2014 paper by using SURF features.

4.1 The Theory of Funnel Lane

Funnel Lane theory was originally developed for 2D navigation by Z. Chen and S. Birchfield [2] in 2009 in order to navigate ground vehicle to follow the desired route. The method first qualitatively defines possible positions where the vehicle can go straight by the constraints of feature coordinates between the current image and the reference image. Navigation is based on funnel-lane guided motion. If the vehicle locates outside the funnel lane, it will be navigated back to the funnel lane. In the thesis, Funnel Lane theory is extended to 3D navigation. Applying Funnel Lane theory for the case of one *fixed landmark* (FL), UAV, which is locating at the current position, (P^C - point C), wants to reach the end of the segment s, (P_s^E - point E, $P^E = P_{s+1}^{Ref}$) as in Fig. 4.2. Notably, the origin of feature coordinates is assigned at the centre of the image plane, while the optical axis of the camera is parallel to the heading direction of the UAV. As shown in Fig. 4.1, the UAV at point E sees the FL (red point) at point $u^E(u_X^E, u_Y^E)$ in the destination image plane I_s^E. At point C, the UAV sees a landmark feature (red point) at point $u^C(u_X^C, u_Y^C)$ in the current image plane I^C. A funnel lane of one landmark feature is created by following definitions:

Definition 1: A funnel lane of a *fixed landmark* (FL) and an UAV at the end of the segment, point E, is the set of locations $\mathbb{F}_{FL,E}$ such that for each $C \in \mathbb{F}_{FL,E}$ four funnel lane constraints are satisfied:

$$|u_X^C| < |u_X^E| \qquad \text{(Horizontal Constraint 1)}$$

$$\text{sign}(u_X^C) = \text{sign}(u_X^E) \qquad \text{(Horizontal Constraint 2)}$$

$$|u_Y^C| < |u_Y^E| \qquad \text{(Vertical Constraint 3)}$$

$$\text{sign}(u_Y^C) = \text{sign}(u_Y^E) \qquad \text{(Vertical Constraint 4)}$$

Definition 2: A funnel lane of a *fixed landmark* (FL), an UAV position at point E, and a relative angle α is the set of positions $\mathbb{F}_{FL,E,\alpha} \subset \mathbb{F}_{FL,E}$ such that $\psi_s^C - \psi_s^E = \alpha$ for each $C \in \mathbb{F}_{FL,E,\alpha}$. A relative angle α is in the plane which is parallel with X-Y plane.

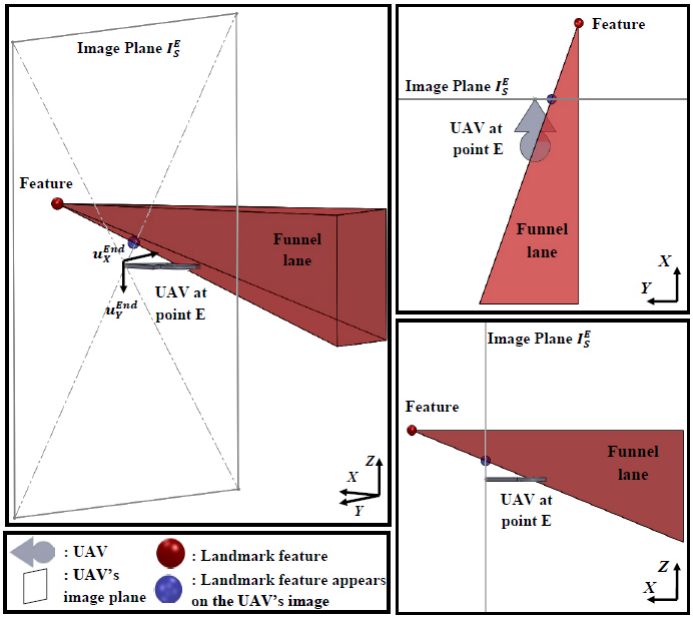

Figure 4.1: Funnel lane created by a fixed landmark corner feature in 3D view (LEFT), X-Y view (RIGHT-TOP), and X-Z view (RIGHT-BOTTOM)

The purpose of using Funnel Lane theory is to define possible positions where the UAV is maneuvered to fly straight. By this way, in Fig. 4.1, a funnel lane is produced and presented in 3D space as a red pyramid with respect to a red landmark corner feature or pyramid with a flat top with respect to the UAV at the end of the segment, point E. The red landmark feature appears in the image plane of the UAV, which is indicated as a blue point. In Fig. 4.2 and Fig. 4.3, the UAV at P_s^E is presented in full-filled symbol while the UAV at P^C is presented in non-filled symbol. Fig. 4.2 presents the case of the UAV's same heading angle ($\psi_s^C - \psi_s^E = 0$) at the current position and the end of the segment s. Fig. 4.3 presents the case of different heading angles ($\psi_s^C - \psi_s^E \neq 0$). The funnel lane will rotate α angle as in *definition 2*.

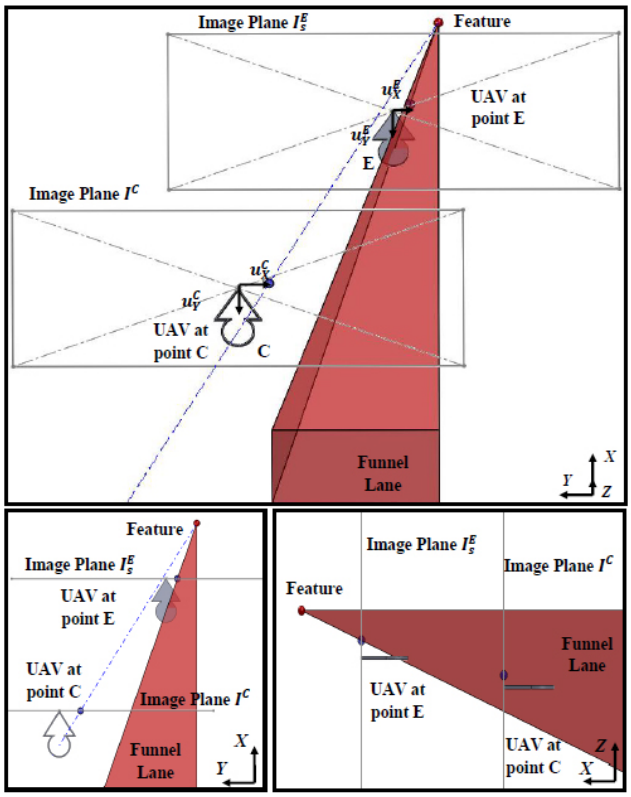

Figure 4.2: UAV from current position reaches the end of the segment s

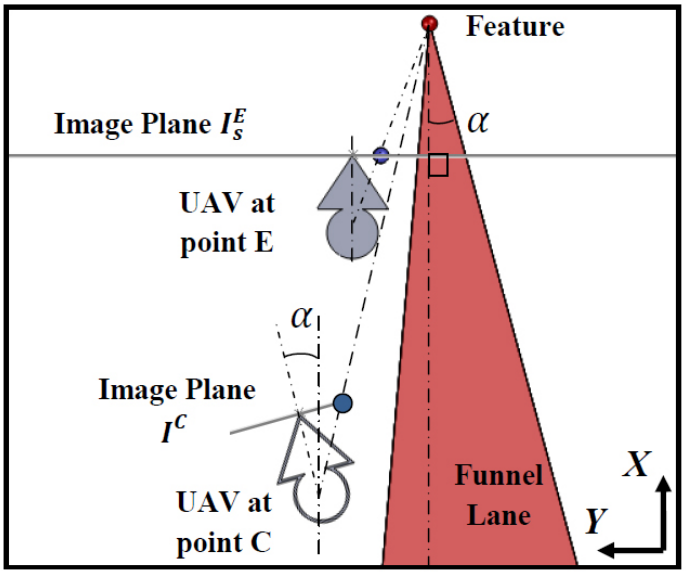

Figure 4.3: Funnel lane in case of different heading angles

Each landmark feature has a unique funnel lane. The intersection between these funnel lanes will satisfy both constraint conditions. If the UAV is in the intersection funnel lane, it will fly forward. Another advantage given from this point is the reliability of navigation, the UAV needs a minimum of only one landmark feature for navigation. When the UAV falls outside the intersection funnel lane, the navigation is computed in such a way that the UAV will fly back to the intersection funnel lane. Its performance depends on the current position with respect to the intersection funnel lane. Nine possible positions of the UAV are defined on the violation of four constraints and described in Fig. 4.4. The next section will depict a procedure to build and track a funnel-lane visual path.

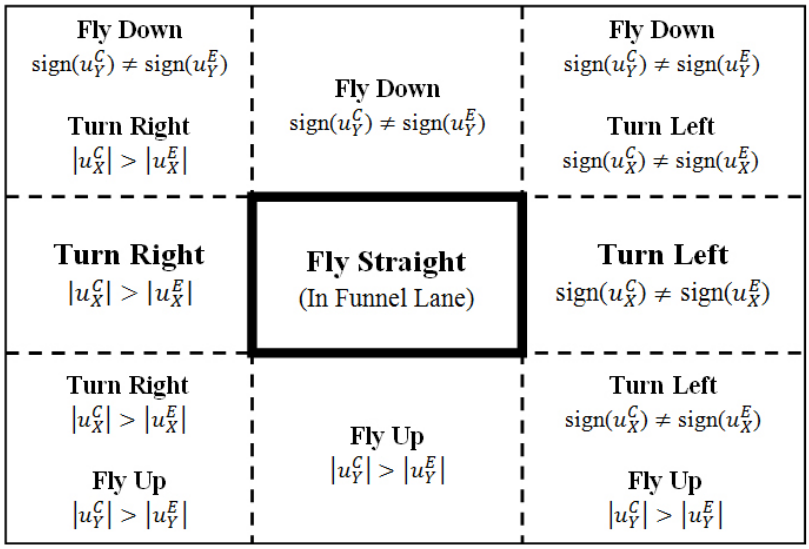

Figure 4.4: Nine possible positions of the UAV w. r. t. the intersection funnel lane

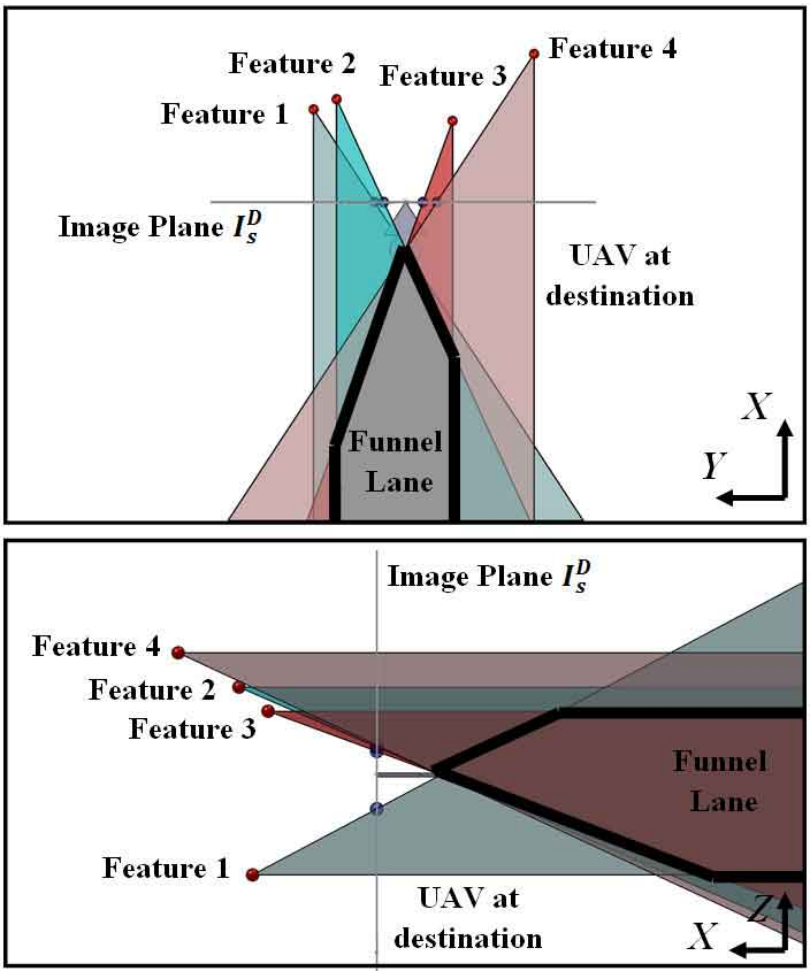

Figure 4.5: Multiple feature (four features) case with X-Y and Y-Z views

4.2 Building of The Desired Route

The desired route Ψ is broken into several continuous segments $\{Seg_s | s \in \{1, 2, ..., n\}\}$. We have a set of reference images: $\{I_s^{Ref} | s \in \{1, 2, ..., n\}\}$. If Funnel Lane theory is applied into each segment, I_{s+1}^{Ref} becomes I_s^E and P_{s+1}^{Ref} becomes P_s^E. The Funnel-Lane visual route is defined by the following:

- *Hypothesis 1:* The UAV's coordinate frame is in the segment s at current position P^C and the task is to reach the destination position P_{s+1}^{Ref}. Two key images I^C and I_{s+1}^{Ref} are respectively associated with P^C and P_{s+1}^{Ref}. There always exists an acceptable route χ from P^C to P_{s+1}^{Ref}.

- *Hypothesis 2:* In the segment s, a set of matched features $\{F_s^M\}$ between two key images I^C and I_{s+1}^{Ref} is observed along the path Ψ from P^C to P_{s+1}^{Ref} and allows the funnel-lane computation of the navigation. There exist nine possible locations of the UAV with respect to the intersection funnel lane formed by matched points $\{F_s^M\}$, $\{F_s^M | s \in \{1, 2, ..., n\}\}$.

- *Hypothesis 3:* In the segment s, the condition to apply the Funnel Lane theory is that the transformation between P^C frame and P_{s+1}^{Ref} frame does not include the case which contains only lateral transformation.

The calculation of motion control commands in order to control quadrotor to follow the desired route after self-localization is described in the next section: funnel-lane 3D motion control algorithm.

4.3 Funnel-Lane 3D Motion Control Algorithm

4.3.1 The Desired Heading of Quadrotor

For each feature j, a signed distance to the line $u_X^C = u_X^E$ is calculated:

$$f(u_X^C, u_X^E) = \frac{1}{\sqrt{2}}(u_X^C - u_X^E) \tag{4.1}$$

And a desired heading of feature j is computed by the feature horizontal coordinate:

$$\begin{aligned}
&\textbf{switch}(u_X^C, u_X^E) \\
&\textbf{case} : u_X^C > 0 \text{ and } u_X^C > u_X^E \\
&\qquad \psi_d^{(j)} = \gamma_1 \cdot \min\{u_X^C, f(u_X^C, u_X^E)\} \\
&\textbf{case} : u_X^C < 0 \text{ and } u_X^C < u_X^E \\
&\qquad \psi_d^{(j)} = \gamma_1 \cdot \max\{u_X^C, f(u_X^C, u_X^E)\} \\
&\textbf{case} : otherwise \\
&\qquad \psi_d^{(j)} = 0
\end{aligned} \tag{4.2}$$

The final desired heading for quadrotor is given by:

$$\psi_d = \eta_1 \frac{1}{N} \sum_{j=1}^{N} \psi_d^{(j)} + (1 - \eta_1)\psi_0 \tag{4.3}$$

Where: γ_1 is the approximate conversion from pixels to degrees. N is the total number of matched features used in the algorithm. ψ_0 is the desired heading obtained from magnetometery heading measurements at the start and the end of the segment s in the learning phase. η_1 $(0 \leq \eta_1 \leq 1)$ is the confidence between visual measurements versus magnetometery heading measurements. In experiment, η_1 is chosen as 0.5.

4.3.2 The Desired Height of Quadrotor

For each feature j, a signed distance to the line $u_Y^C = u_Y^E$ is calculated:

$$f(u_Y^C, u_Y^E) = \frac{1}{\sqrt{2}}(u_Y^C - u_Y^E) \tag{4.4}$$

And a desired height of feature j is computed by the vertical feature coordinate:

$$\begin{aligned}
&\textbf{switch}(u_Y^C, u_Y^E) \\
&\textbf{case} : u_Y^C > 0 \text{ and } u_Y^C > u_Y^E \\
&\qquad Z_d^{(j)} = \gamma_2 \cdot \min\{u_Y^C, f(u_Y^C, u_Y^E)\} \\
&\textbf{case} : u_Y^C < 0 \text{ and } u_Y^C < u_Y^E \\
&\qquad Z_d^{(j)} = \gamma_2 \cdot \max\{u_Y^C, f(u_Y^C, u_Y^E)\} \\
&\textbf{case} : \mathit{otherwise} \\
&\qquad Z_d^{(j)} = 0
\end{aligned} \tag{4.5}$$

The final desired height for quadrotor is given by:

$$Z_d = \eta_2 \frac{1}{N} \sum_{j=1}^{N} Z_d^{(j)} + (1 - \eta_2) Z_0 \tag{4.6}$$

Where: γ_2 is the approximate conversion from pixels to meters. N is the total number of matched features used in the algorithm. Z_0 is the desired height obtained from altimeter measurements at the start and the end of the segment in the learning phase. η_2 $(0 \leq \eta_2 \leq 1)$ is confidence between visual measurements versus altimeter measurements. In experiment, η_2 is chosen as 0.5.

4.4 Experimental Setup

The ROS system for simulation in the previous chapter of self-localization is used to test the proposed algorithm. Initial realtime experiments are conducted in the Intelligent Systems Lab, Faculty of Engineering and Applied Science, Memorial University of Newfoundland. The model of quadrotor is Ar.Drone quadrotor ver 2.0. The front camera of the quadrotor is HD camera 1280x720 pixels (720p) with 92^0 wide-angle. The calculation of navigation is performed off-board on a laptop then sent to the Ar.Drone through wireless communication. The system architecture of realtime experiment is different from those of simulation. *adrone_autonomy* node is used to connect with the Ar.Drone quadrotor. Although the architecture can eliminate the independent operations of quadrotor, the purpose of experiments is to validate the performance of the whole system. The current development of hardware design can enable onboard processing for fully autonomous applications. The thesis do not consider the appearance of moving obstacles or obstacle avoidance in experiments. The video of initial realtime experiment can be located at *youtu.be/WVq9IttJx0g*.

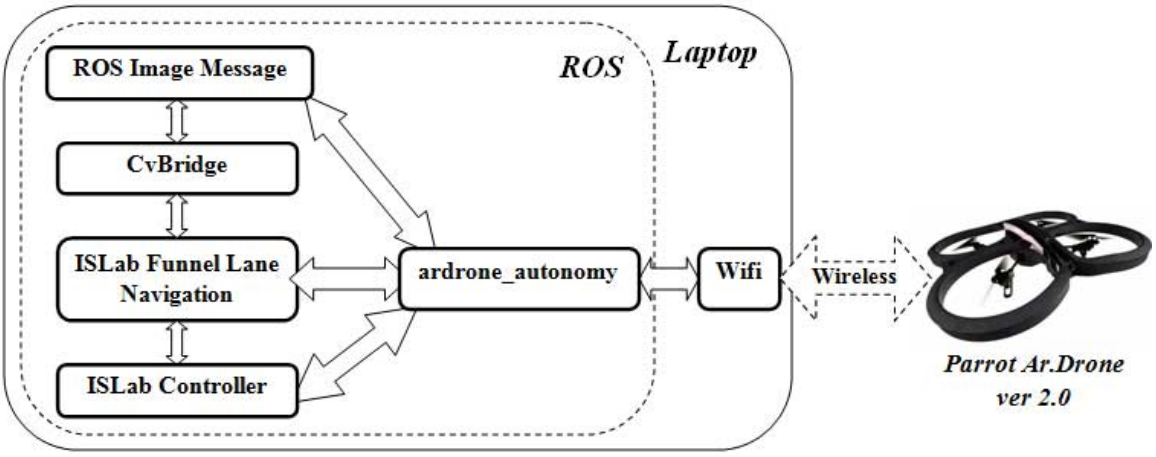

Figure 4.6: The system architecture of realtime experiment

4.5 Experimental Results

In the beginning, quadrotor is controlled along the desired route in order to collect reference images. After that, quadrotor will be placed closely to the same initial position as in the learning phase to perform the replaying phase. Quadrotor utilizes self-localization to define the current segment on the desired route. After the image in the database is loaded as reference image, quadrotor autonomously follows the current segment of the desired route. Quadrotor uses funnel-Lane 3D navigation algorithm to calculate desired yaw angle and desired height for the next movement. The front velocity command of quadrotor is kept to be $0.1m/s$. Numerous following experiments are conducted in simulation to evaluate the performance of the proposed system.

- Experiment I: tests following about 3m curve desired route with KLT feature when developing Funnel Lane theory for 3D navigation. Transition between two successive segments by setting a switching threshold for *Mean Square Error* MSE of matched feature coordinates in the images.
- Experiment II: tests like experiment I to follow about 6m open-loop route.
- Experiment III: tests following 3.5m curve route with different start positions with SURF features. Localization of quadrotor during following the desired route is method I of self-localization.
- Experiment IV: tests like experiment III with complicated 3D polyline route.
- Experiment V: is conducted to compare three methods of self-localization in combination with funnel-lane 3D motion control algorithm.
- Experiment VI: tests following a long route in mountain area with 55 reference images. The outdoor environment of mountain area is simulated on Gazebo simulator (for video: *youtu.be/0YVGK1-ObGM*).

Table 4.1: Configuration of Experiment I

Purpose of Experiment:	Test following route by Funnel-lane theory [49]
Feature:	KLT
Self-Localization:	Mean Square Error [47]
Type of Route:	(Simulation) 2D Curve

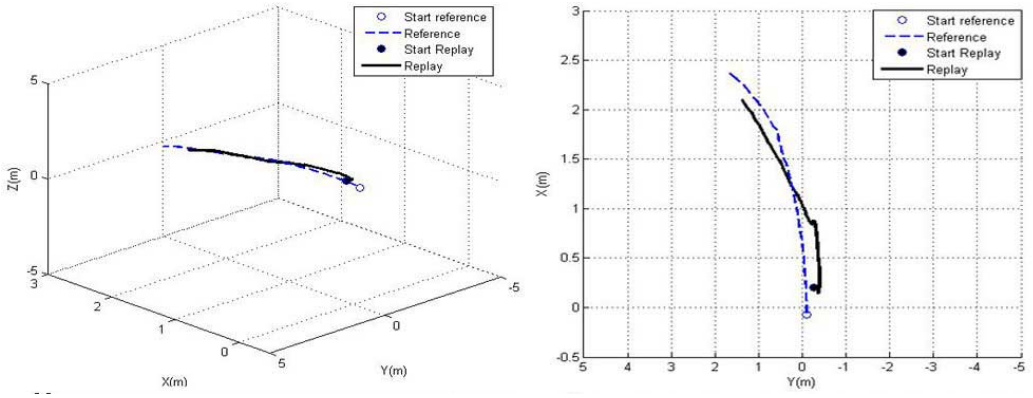

Figure 4.7: Result of Experiment I

Table 4.2: Configuration of Experiment II

Purpose of Experiment:	Test following route by Funnel-lane theory [49]
Feature:	KLT
Self-Localization:	Mean Square Error [47]
Type of Route:	(Simulation) Open-Loop

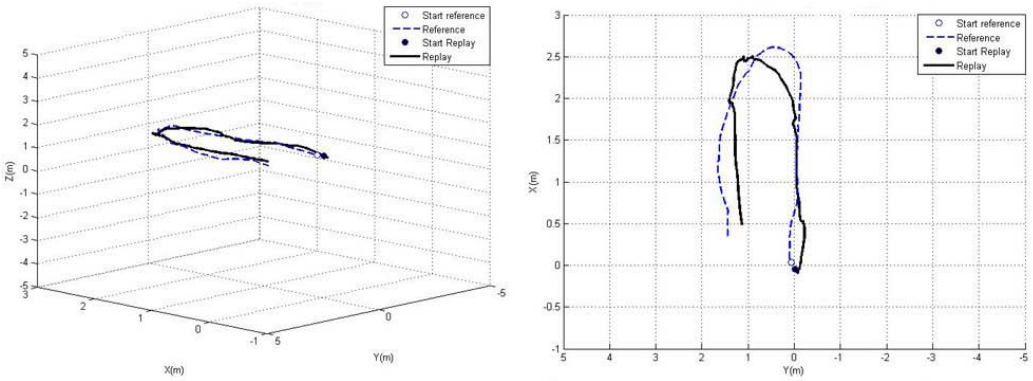

Figure 4.8: Result of Experiment II

Table 4.3: Configuration of Experiment III

Purpose of Experiment:	Test tracking 2D route with difference start positions [79]
Feature:	SURF
Self-Localization:	Method I
Type of Route:	(Simulation) 2D Curve

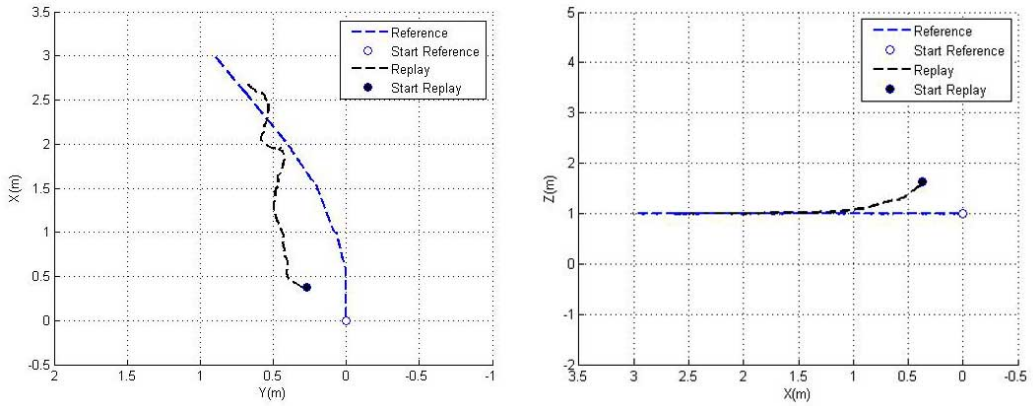

Figure 4.9: Result of Experiment III

Table 4.4: Configuration of Experiment IV

Purpose of Experiment:	Tracking complicated 3D route [79]
Feature:	SURF
Self-Localization:	Method I
Type of Route:	(Simulation) 3D Polyline

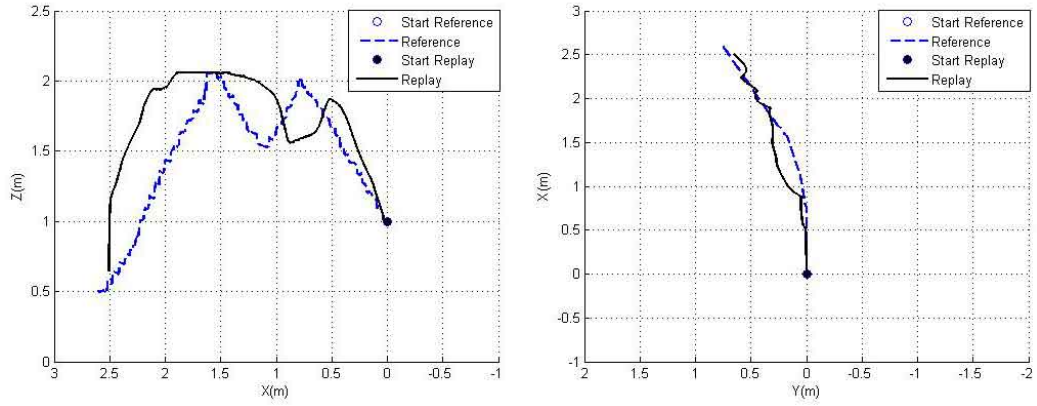

Figure 4.10: Result of Experiment IV

Table 4.5: Configuration of Experiment V

Purpose of Experiment:	Tracking indoor 2D route
Feature:	SURF
Self-Localization:	Method I, II and III
Type of Route:	(Simulation) 2D Polyline

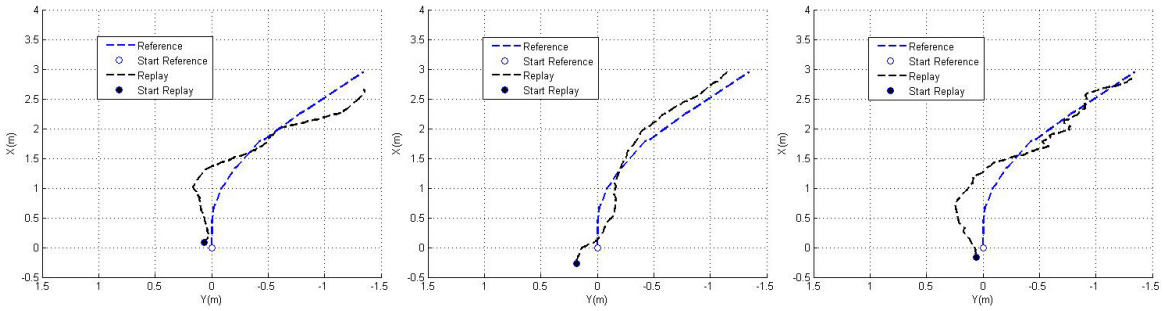

Figure 4.11: Result of Experiment V with three methods. LEFT: Method I; MIDDLE: Method II; RIGHT: Method III

Table 4.6: Configuration of Experiment VI

Purpose of Experiment:	Tracking a long route to survey mountain area
Feature:	SURF
Self-Localization:	Method III
Type of Route:	(Simulation) 3D Polyline Outdoor

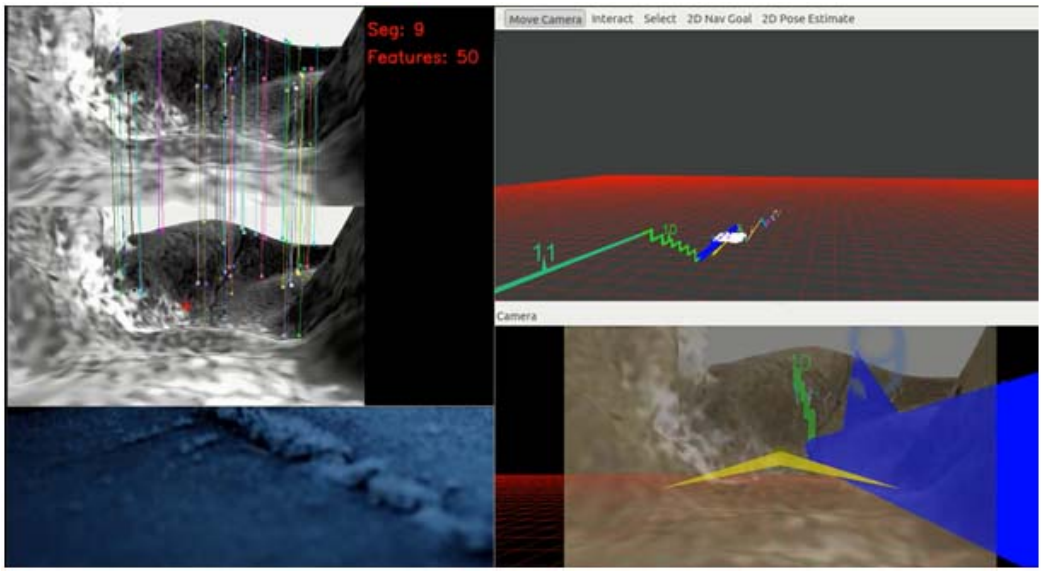

Figure 4.12: Experimental Setup. LEFT: Images showing matching SURF features. RIGHT-TOP: Rviz application to check ground truth error. RIGHT-BOTTOM: Camera-view of quadrotor in Rviz

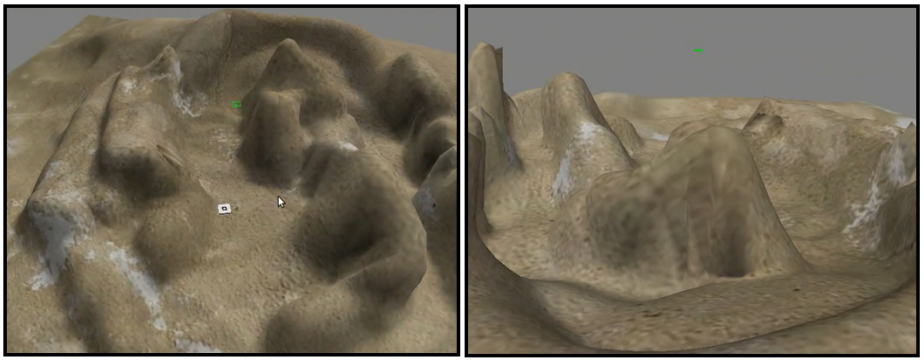

Figure 4.13: Gazebo simulation of mountain area

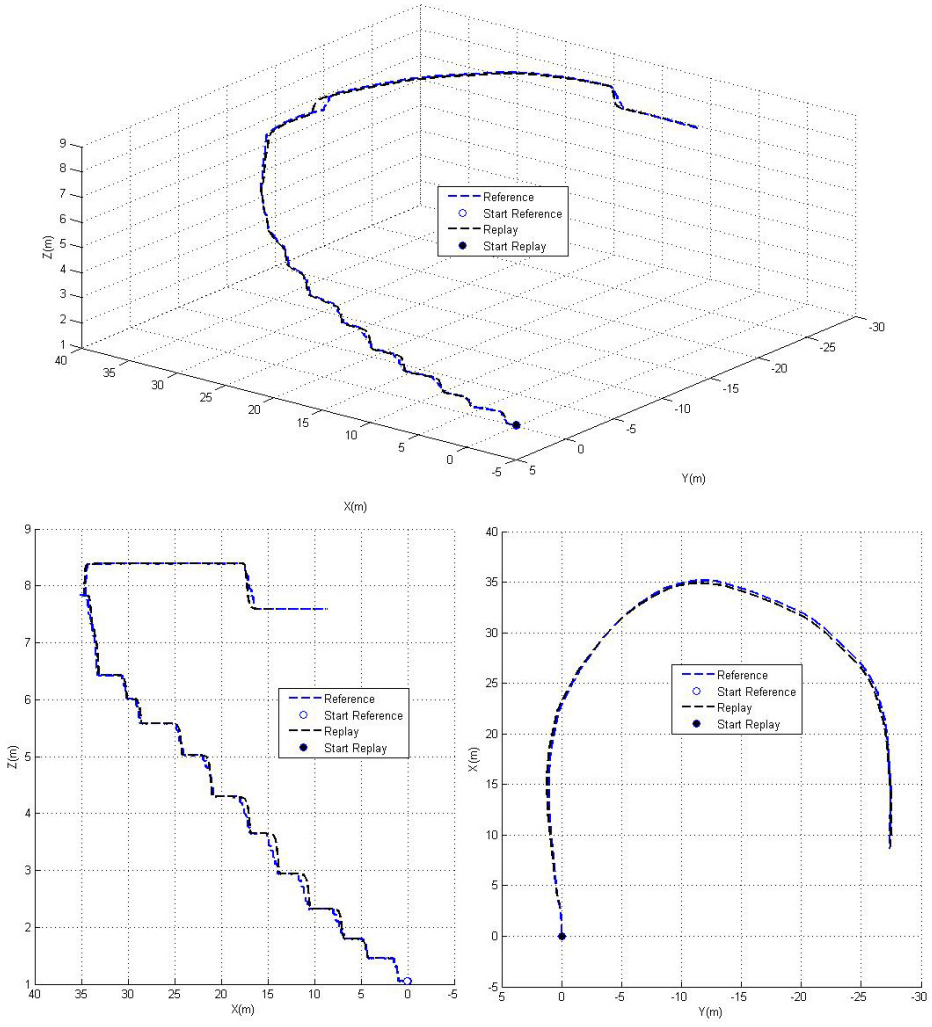

Figure 4.14: Simulation results of tracking the desired route in mountain area. TOP: 3D view. LEFT-BOTTOM: X-Z view. RIGHT-BOTTOM: X-Y view

4.6 Discussion

Experiments show that quadrotor with VT&R technique is able to navigate and follow the desired route. Some observations of the performance are described as follows:

- Firstly, the limitation of matching KLT features [2, 49] provides some incorrect matched features. The video of some experiments with KLT feature can be located at *youtu.be/WVq9IttJx0g*. The proposal to use SURF feature as well as the utilization of the relation between SURF feature size with spatial distance provides better self-localization.

- The transition between segments is overestimated when employing MSE method. Basically, MSE method considers the coordinate errors of matched features to make decision of transition. Hence, quadrotor can not possibly reach the end of the desired route although all reference images are used. The use of method I, II and III of self-localization greatly decreases the overshoot and produces better estimation of the current segment.

- In the experiments showed in Fig. 4.11, the thesis can not conclude which method of self-localization provides less position errors when following the desired route because the vehicle started at different poses. However, the thesis can observe how the performance of self-localization part, which influences to qualitative motion control command part. True to discussion in section 3.6, method III provides better estimation of accuracy and stability. It means that if quadrotor is at segment 5, the errors of method III can be fallen into segment 4 or segment 6, while those of method I and II can reach segment 1 or 7.

- Fig. 4.5 shows the experiment with the long desired route in mountain area, containing 55 reference images. A number of reference images can negatively

increase the computation cost if all reference images are processing at the same time. However, this challenge can be overcome if estimating the quadrotor location on all reference images in the beginning of the replaying phase. Then, estimating location is performed by 3 or 4 reference images logically related to the current segment. For example, if quadrotor is at segment 5 in the start position, self-localization will use reference images of segment 4, 5 and 6.

- As following the desired route by only the start and end images of the route's segment, the position errors are inevitable. It is a limitation of qualitative VT&R technique but considered as practical within GPS-denied environment. However, quadrotor shows the ability of route following with acceptable errors.

- Applying the SURF feature-size relation with spatial distance helps to eliminate unappropriate matched features. However, it also decreases a number of matched features for navigation. As a result, a number of matched features maybe not enough to perform navigation. This phenomenon also explains why working in outdoor environment receives more successful trials than working in indoor environment. In addition, quadrotor should move at least 1m to receive the obvious change of SURF feature-size. Basing on this property, the declaration of segment also needs to obey this rule.

- After accidentally colliding to other objects in environment, quadrotor can be rotated to unpredictable heading angle. Hence, all matched landmarks will fall out of the image plane of quadrotor camera, and fail the VT&R system. However, since the thesis utilizes magnetometery heading measurements in the calculation of motion control commands, quadrotor will rotate back as before collision and correct matched features will come back to the image plane of quadrotor camera.

- Compared to Courbon's strategy [27], our work has applied additional step to match the features in I^C from both I_s^{Ref} and I_{s+1}^{Ref}. This step provides more reliable results in self-localization than the method used in [27], which matches the features in I^C from only I_{s+1}^{Ref}. In addition, if the method of the work [27] will suffer more computational cost when trying to estimate the relative distances between the current image with the reference images as well as does not show ability to recover from accidental collision.

In the qualitative VT&R aerial system, motion control commands (desired yaw angle and desired height), generated by the funnel-lane 3D motion control algorithm, are sent to the controller part to compute the desired angular velocities for four rotors of the quadrotor. In experiment I and II, the controller uses PID control. In experiment III, IV, V and VI, the controller is developed to nonlinear geometric controller, which will be described in the next chapter.

Chapter 5

Nonlinear Geometric Controller

About this chapter: This chapter presents the nonlinear geometric controller of quadrotor. The controller will receive motion control commands (desired yaw angle and desired height) from the funnel-lane 3D motion control algorithm of the qualitative motion control command part to produce the desired angular velocities for four rotors. Applying bond-graph modeling method, the motions of quadrotor on the configuration space SE(3) are expressed in the graphical dynamic model. The nonlinear geometric controller is proposed to control the nonlinear and under-actuated dynamic model of quadrotor. Simulations in different scenarios are conducted on 20Sim program to evaluate the performance of the proposed controller before implementing into the ROS system.

5.1 The Design of Nonlinear Geometric Controller

5.1.1 The Role of Controller in VT&R System

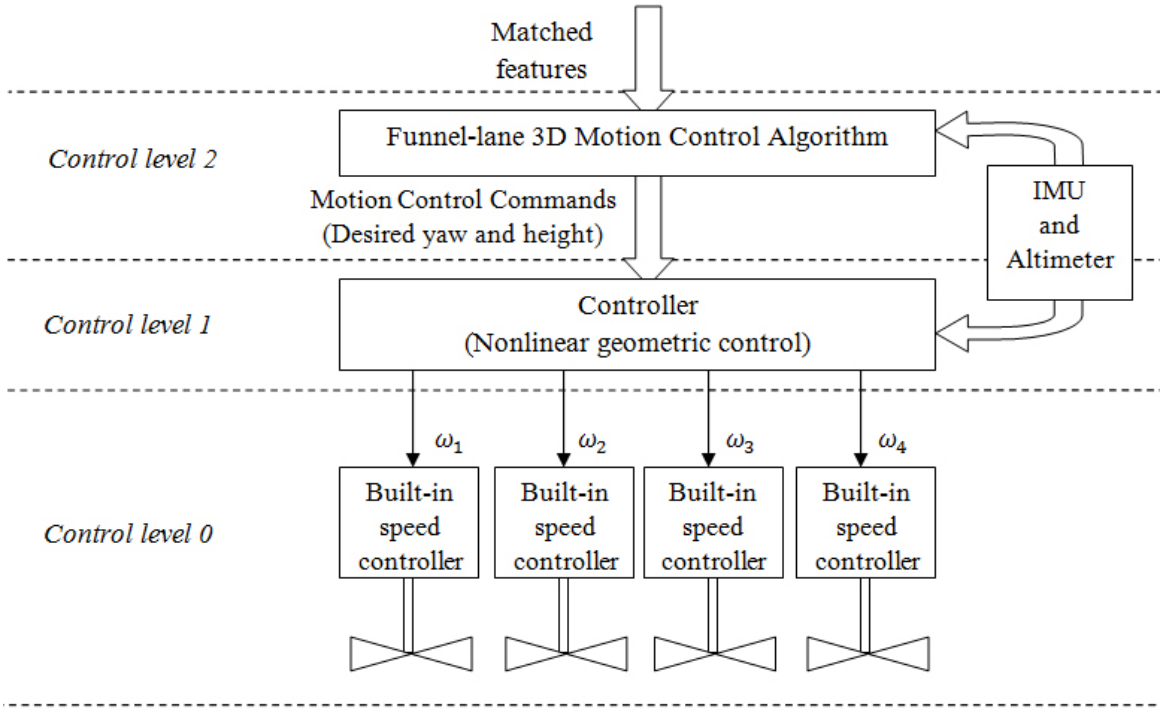

Figure 5.1: The levels of the control system

Fig. 5.1 describes the functions of the designed nonlinear geometric controller. The control system consists of three levels. At the control level 2, funnel-lane 3D motion control algorithm uses SURF matched features between the current image and the reference image to generate the motion control commands (desired yaw and height). At control level 1, the controller attempts to convert these motion control commands into the desired angular velocities for the four rotors of the quadrotor. At control level 0, four built-in PID speed controllers are used to control the angular velocities of four rotors basing on the desired angular velocities. Appropriate forces and torques of four rotors are produced to create the motions of the quadrotor.

5.1.2 Introduction to Geometric Control Theory

Traditional controller such as PID control is applied into controlling position and attitude of the quadrotor [5, 59]. Six PID controllers should be used to control traveling in X, Y, Z direction and Roll - Pitch - Yaw angles. The PID control is sufficient for simple applications [62] as well as possibly aggressive manoeuvres with a great support of a VICON^{TM} motion capture system [5]. However, the structure design of six PID controllers is complicated to improve the performance of the system for advanced applications [30]. LQR control is proposed as a design to control the full dynamic model of the quadrotor using state feedback [64]. MPC controller [54, 55] improves the performance of quadrotor controller by incorporating system constraints and accurate system prediction models. The output control efforts of MPC controller include the control actions according to the prediction of the future events, as well as optimizing steps basing on the quadrotor system constraints such as motor constraints. Generally, advanced calculations to improve the controller performance require more computational resources. On the other hand, the design of these controllers takes linear-approximation step to simplify the problem, which neglects the underlying geometric properties of quadrotor dynamic model [65]. Moreover, considering the integration within the system of VT&R technique, the positional and rotational errors of quadrotor controller should not considerably affect the quality of image feedback, which will be used to extract SURF features for navigation.

As a result, nonlinear geometric controller [65], applied geometric control theory [66], is developed for quadrotor controller. The use of geometric control theory, a coordinate-free control approach, attempts to convert the difficult problem of quadrotor control into a straight-forward linear problem [66]. The design of nonlinear geometric controller will concentrate on the errors of rotational matrix [65], instead of attempting to access the errors of each degree of freedom. Therefore, the geometric

controller design is simplified enough to implement into the embedded systems of quadrotor [80].

5.1.3 Quadrotor Configuration

$\{E\}$ is considered as an inertial frame with the unit vectors $\{\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z\}$. The body-fixed $\{B\}$ frame with the unit vectors $\{\mathbf{b}_x, \mathbf{b}_y, \mathbf{b}_z\}$ is attached to quadrotor's body. Quadrotor's attitude (ϕ-Roll, θ-Pitch, ψ-Yaw) is the orientation of the $\{B\}$ frame with respect to the $\{E\}$ frame. Rotational matrix $\mathbf{R}$ expresses the transformation from the $\{B\}$ frame to the $\{E\}$ frame, $\mathbf{R} \in \mathrm{SO}(3)$.

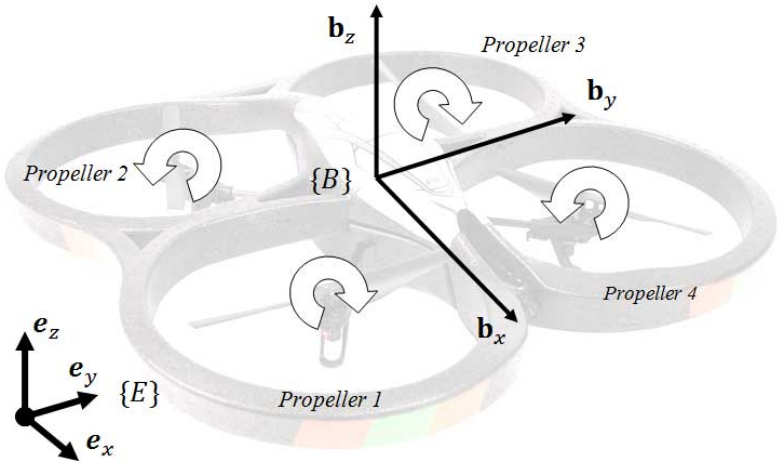

Figure 5.2: The coordinates of quadrotor

A rotation matrix (Z - X - Y Euler angles) $\mathbf{R}$ from the $\{B\}$ frame to the $\{E\}$ frame:

$$\mathbf{R} = \begin{bmatrix} \cos\psi\cos\theta - \sin\phi\sin\psi\sin\theta & -\cos\phi\sin\psi & \cos\psi\sin\theta + \cos\theta\sin\phi\sin\psi \\ \cos\theta\sin\psi + \cos\psi\sin\phi\sin\theta & \cos\phi\cos\psi & \sin\psi\sin\theta - \cos\psi\cos\theta\sin\phi \\ -\cos\phi\sin\theta & \sin\phi & \cos\phi\cos\theta \end{bmatrix} \tag{5.1}$$

In geometric viewpoint [66], the configuration manifold of rigid body as quadrotor is the special Euclidean group SE(3), a group of displacement in $\mathbb{R}^3$, and the special

orthogonal group SO(3), group of proper rotations.

$$\mathrm{SO}(3) = \{\mathbf{R} \in \mathbb{R}^{3\times 3} \mid \mathbf{R}^T\mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1\} \tag{5.2}$$

Table 5.1: Some specifications of quadrotor

$\mathbf{r}$	Quadrotor position vector in $\{E\}$ frame, $\mathbf{r} \in \mathbb{R}^3$ $\mathbf{r} = [x; y; z]$
$\mathbf{r}_{ref}$	Quadrotor reference position vector in $\{E\}$ frame, $\mathbf{r}_{ref} \in \mathbb{R}^3$ $\mathbf{r}_{ref} = [x_{ref}; y_{ref}; z_{ref}]$
$\mathbf{v}$	Quadrotor velocity vector in $\{E\}$ frame, $\mathbf{v} \in \mathbb{R}^3$ $\mathbf{v} = [v_x; v_y; v_z]$
$\mathbf{v}_{ref}$	Quadrotor reference velocity vector in $\{E\}$ frame, $\mathbf{v}_{ref} \in \mathbb{R}^3$ $\mathbf{v}_{ref} = [v_{x,ref}; v_{y,ref}; v_{z,ref}]$
$\mathbf{\Omega}$	The angular velocity in $\{B\}$ frame, $\mathbf{\Omega} \in \mathbb{R}^3$ $\mathbf{\Omega} = [\omega_x; \omega_y; \omega_z]$
f_i	The force vector produced by propeller i in $\{B\}$ frame
M_i	The moment vector produced by propeller i in $\{B\}$ frame
$\mathbf{F}^E$	The force acting on quadrotor in $\{E\}$ frame, $\mathbf{F}^E \in \mathbb{R}^3$ $\mathbf{F}^E = [F_x; F_y; F_z]$
F	The total thrust force of four propellers, $F = \sum_{i=1}^{4} f_i$ in $\{B\}$, $F \in \mathbb{R}$
$\mathbf{T}^B$	The external torque acting on quadrotor in $\{B\}$ frame, $\mathbf{T}^B \in \mathbb{R}^3$ $\mathbf{T}^B = [T_x; T_y; T_z]$
$\mathbf{F}_g^E$	The gravity vector in $\{E\}$, $\mathbf{F}_g^E \in \mathbb{R}^3$
$\mathbf{F}_P^B$	The propulsion force vector in $\{B\}$ frame, $\mathbf{F}_P^B \in \mathbb{R}^3$
$\mathbf{M}_P^B$	The propulsion torque vector in $\{B\}$ frame, $\mathbf{M}_P^B \in \mathbb{R}^3$
m	The total mass of quadrotor, $m \in \mathbb{R}$
$\mathbf{J}$	The inertial matrix of quadrotor w.r.t $\{B\}$ frame, $\mathbf{J} \in \mathbb{R}^{3\times 3}$
d	The distance from the geometric centers of motors to the geometric centers of quadrotor, $d \in \mathbb{R}$
$\mathbf{R}$	The rotation matrix from $\{B\}$ frame to $\{E\}$ frame, $\mathbf{R} \in \mathbb{R}^{3\times 3}$
$\mathbf{R}_{ref}$	The reference rotation matrix from $\{B\}$ frame to $\{E\}$ frame, $\mathbf{R}_{ref} \in \mathbb{R}^{3\times 3}$

The rotational kinematic equations of quadrotor's attitude [66] governing the trajectory $t \mapsto \mathbf{R}(t)$ is written as in Eq. 5.3. The Euler - Poincaré equation [66] describing the trajectory $t \mapsto \mathbf{\Omega}(t)$ for any torque acting on the body is described in Eq. 5.4.

$$\dot{\mathbf{R}} = \mathbf{R}\hat{\mathbf{\Omega}} \tag{5.3}$$

$$\mathbf{J}\dot{\mathbf{\Omega}}(t) + \mathbf{\Omega}(t) \times \mathbf{J}\mathbf{\Omega}(t) = \mathbf{T}^B(t) \tag{5.4}$$

Where:

- The hat map $\hat{\cdot}$ is $\mathbb{R}^3 \to \mathfrak{so}(3)$ defined by the condition $x\hat{y} = x \times y$ for all $x, y \in \mathbb{R}^3$.
- $\mathfrak{so}(3)$ is vector space of skew-symmetric 3×3 matrices and Lie algebra of SO(3) [66].

5.1.4 Nonlinear Geometric Controller

To solve the problem of underactuation, quadrotor is recognized as differential flatness [30]. Guiding vectors ($\mathbf{s}^{ref}$ and $\{\mathbf{b}_x^{ref}, \mathbf{b}_y^{ref}, \mathbf{b}_z^{ref}\}$) are created based on the trajectory error calculation. The reference rotation matrix, $\mathbf{R}_{ref}$, is generated from $\mathbf{b}_x^{ref}, \mathbf{b}_y^{ref}, \mathbf{b}_z^{ref}$. The attitude controller makes effort to control unit vectors of $\{B\}$ frame become guiding vectors through the errors between the rotational matrix $\mathbf{R}$ and the reference rotation matrix $\mathbf{R}_{ref}$.

$$\mathbf{s}^{ref} = [\cos\psi_{des}, \sin\psi_{des}, 0]^T \tag{5.5}$$

$$\mathbf{b}_z^{ref} = \frac{\mathbf{F}}{\|\mathbf{F}\|} \tag{5.6}$$

$$\mathbf{b}_y^{ref} = \frac{\mathbf{b}_z^{ref} \times \mathbf{s}^{ref}}{\|\mathbf{b}_z^{ref} \times \mathbf{s}^{ref}\|} \tag{5.7}$$

$$\mathbf{b}_x^{ref} = \mathbf{b}_y^{ref} \times \mathbf{b}_z^{ref} \tag{5.8}$$

The positional errors ($\mathbf{e_r}$) and the velocity errors ($\mathbf{e_v}$) are written as following equations:

$$\mathbf{e_r} = \mathbf{r} - \mathbf{r}_{ref} \ ; \ \mathbf{e_v} = \mathbf{v} - \mathbf{v}_{ref} \tag{5.9}$$

The geometric control theory is used to exactly define the error ($\mathbf{e_R}$) and error rate

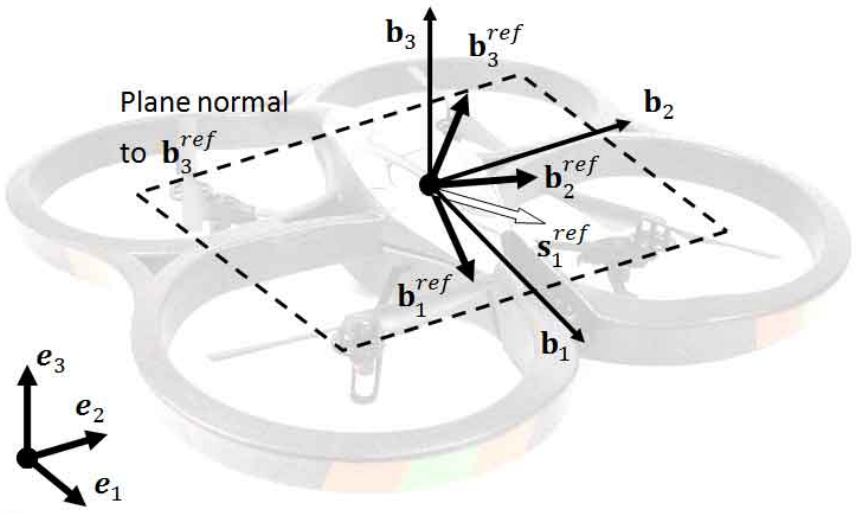

Figure 5.3: The reference coordinates of quadrotor

($\mathbf{e_\Omega}$) of the rotation matrix $\mathbf{R}$ [65], written as Eq. 5.10 and 5.11. Therefore, linear approximation is not taken to preserve the underlying geometric properties of dynamic model. In Eq. 5.10, $.^\vee$ is the vee map to transfer $\mathbb{R}^3 \leftarrow \mathfrak{so}(3)$.

$$\mathbf{e_R} = \frac{1}{2}(\mathbf{R}_{ref}^T\mathbf{R} - \mathbf{R}^T\mathbf{R}_{ref})^\vee \tag{5.10}$$

$$\mathbf{e_\Omega} = \mathbf{\Omega} - \mathbf{R}^T\mathbf{R}_{ref}\mathbf{\Omega}_{ref} \tag{5.11}$$

The desired thrust force vector and the desired moment vector is calculated as:

$$\mathbf{F} = -K_p\mathbf{e_r} - K_v\mathbf{e_v} + mg\mathbf{e_3} + m\ddot{\mathbf{r}}_{ref} \tag{5.12}$$

$$F = \mathbf{F}.\mathbf{b}_z \tag{5.13}$$

$$\mathbf{M} = \begin{bmatrix} M_x & M_y & M_z \end{bmatrix}^T = -K_\mathbf{R}\mathbf{e_R} - K_\mathbf{\Omega}\mathbf{e_\Omega} \tag{5.14}$$

K_p, K_v, $K_\mathbf{R}$ and $K_\mathbf{\Omega}$ are controller's parameters. Tuning controller's parameters is challenging because these parameters are positive real numbers (in $\mathbb{R}$), which is mul-

tiplied with error matrices (in $\mathbb{R}^3$). The simple method is try-and-error. However, it requires more studies to choose better values for these parameters. Fig. 5.4 describes the structure of the controller design. In the design, the position control is directly computed in the attitude control through guiding vectors or reference rotational matrix. Hence, the position errors of quadrotor converge to zero, when the rotational errors also converge to zero. This control strategy reflects the physical nature of quadrotor motions.

Another element which makes quadrotor unique is the simple mechanical structure with vertically fixed rotors. Each propeller generates the torque directly proportional to its thrust. k_F is a propeller force constant in $f_i = k_F\omega_i^2$, and k_M a propeller moment constant in $M_i = k_M\omega_i^2$ with ω_i an angular speed of the i-th propeller. The following equations are the unique property of Ar.Drone quadrotor when the X-direction of quadrotor is directed as Fig. 5.2. The motor model can be presented by bond-graph method. However, to simplify the system model, the electrical-mechanical structure of the motor is not considered.

$$\begin{bmatrix} F \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ -k_F d & -k_F d & k_F d & k_F d \\ -k_F d & k_F d & k_F d & -k_F d \\ -k_M & k_M & -k_M & k_M \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{5.15}$$

or

$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4k_F} & -\frac{1}{4k_F d} & -\frac{1}{4k_F d} & -\frac{1}{4k_M} \\ \frac{1}{4k_F} & -\frac{1}{4k_F d} & \frac{1}{4k_F d} & \frac{1}{4k_M} \\ \frac{1}{4k_F} & \frac{1}{4k_F d} & \frac{1}{4k_F d} & -\frac{1}{4k_M} \\ \frac{1}{4k_F} & \frac{1}{4k_F d} & -\frac{1}{4k_F d} & \frac{1}{4k_M} \end{bmatrix} \begin{bmatrix} F \\ M_x \\ M_y \\ M_z \end{bmatrix} \tag{5.16}$$

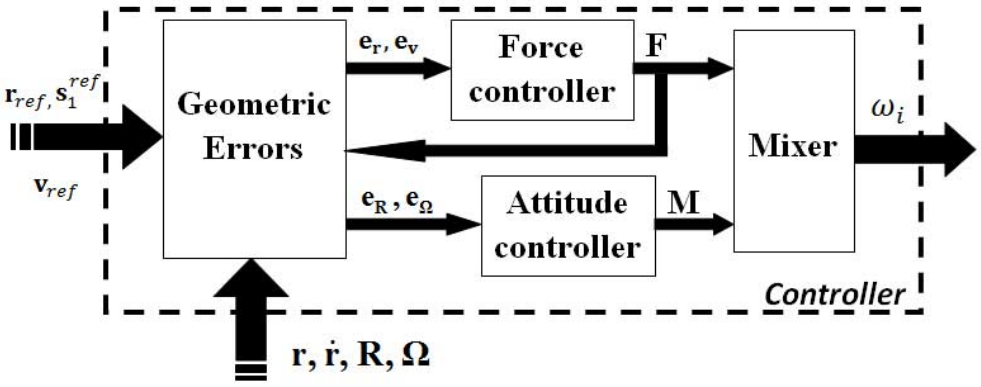

Figure 5.4: The structure of the nonlinear geometric controller

5.2 Quadrotor Dynamic Model

Before implementing into the VT&R system, the design of the nonlinear geometric controller is tested with the bond-graph dynamic model of quadrotor in 20Sim simulation software. Bond-graph modeling method is a graphically natural representation of a physical dynamic system.

5.2.1 Dynamic Equations

Considering quadrotor's motion as the movement of a rigid body described as the sum of all external forces and torques. The Euler's nonlinear translational equations are described [61]:

$$\begin{aligned} F_x &= m\dot{v}_x - mv_y\omega_z + mv_z\omega_y \\ F_y &= m\dot{v}_y - mv_z\omega_x + mv_x\omega_z \\ F_z &= m\dot{v}_z - mv_x\omega_y + mv_y\omega_x \end{aligned} \tag{5.17}$$

and the Euler's rotational equations [61]:

$$
\begin{aligned}
T_x &= J_x\dot{\omega}_x - (J_y - J_z)\omega_y\omega_z \\
T_y &= J_y\dot{\omega}_y - (J_z - J_x)\omega_z\omega_x \\
T_z &= J_z\dot{\omega}_z - (J_x - J_y)\omega_x\omega_y
\end{aligned} \tag{5.18}
$$

Considering the translational motions, the external forces include the gravity force $\mathbf{F}_g^E$, the propulsion thrusts of motors $\mathbf{F}_P^B$ and the drag force $\mathbf{F}_D^E$ with the coefficient of drag C_D:

$$
\begin{aligned}
\mathbf{F} &= \mathbf{F}_g^E + \mathbf{R}\mathbf{F}_M^B - \mathbf{F}_D^E \\
\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + \mathbf{R}\begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix} - \begin{bmatrix} C_{D_x}\dot{x}^2 \\ C_{D_y}\dot{y}^2 \\ C_{D_z}\dot{z}^2 \end{bmatrix}
\end{aligned} \tag{5.19}
$$

Considering the Euler's rotational equations, Eq. 5.18, the external torques include propulsion torque $\mathbf{T}_P^B$ and drag torques $\mathbf{T}_D^B$:

$$
\begin{aligned}
\mathbf{T}^B &= \mathbf{T}_P^B + \mathbf{T}_D^B \\
\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} &= \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} + \begin{bmatrix} -C_{Dx,T}\omega_x^2 \\ -C_{Dy,T}\omega_y^2 \\ -C_{Dz,T}\omega_z^2 \end{bmatrix}
\end{aligned} \tag{5.20}
$$

The angular velocity $\mathbf{\Omega}$ is a vector about the rotational axis, which is different from the derivatives of the Roll, Pitch, Yaw angles $(\dot{\phi}, \dot{\theta}, \dot{\psi})$. A transformation between

them is proposed as following:

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\cos\phi\sin\theta \\ 0 & 1 & \sin\phi \\ \sin\theta & 0 & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{5.21}$$

5.2.2 Quadrotor's bond graph model

Based on these Euler's nonlinear equations, bond-graph model is proposed to represent the quadrotor as the movement of a rigid body in SE(3) [81] in Fig.5.5. Fiq.5.6 presents the dynamic properties of quadrotor's movements. **MTF** element is used to express the transform from the derivatives of Roll-Pitch-Yaw to rotation velocities (Eq. 5.21) and calculate the rotation matrix **R**. **Se** element presents the effect of gravity. The inputs, force and torques, are set in matrix form. $Height\_Matrix$ block, created in **K** block, is needed for taking the force value along z-direction while $Torques\_Matrix$ block, created in **K** block is used for taking torques from $ForceTorques$ matrix input. The content of *EulerEquations* block is presented in Fig. 5.5, which is the bond-graph model of Eq. 5.17 and 5.18.

Fig. 5.7 presents the control system with some controller blocks and dynamic model. $AngularErrorCalculation$ block computes the exactly error of rotation based on the current rotation matrix and the desired rotation matrix. $Force\_Ref$ is used for calculating the values of guiding vectors. $AttitudeController$ calculates the desired-torques control signal based on the rotation error calculation. $Mixer$ block is used for converting control signals (force and torques) into matrix sending to $QuadrotorModel$ block.

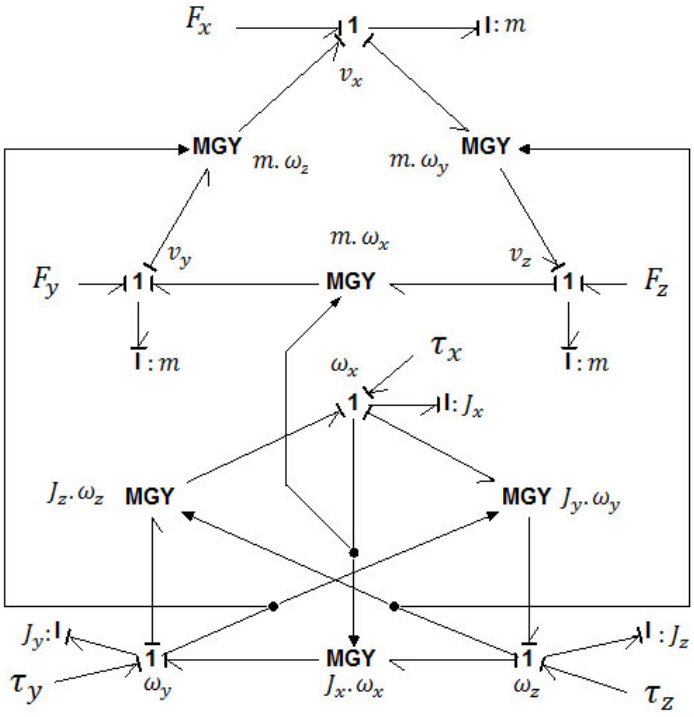

Figure 5.5: The bond graphs of rigid body both translating and rotating in 3D space

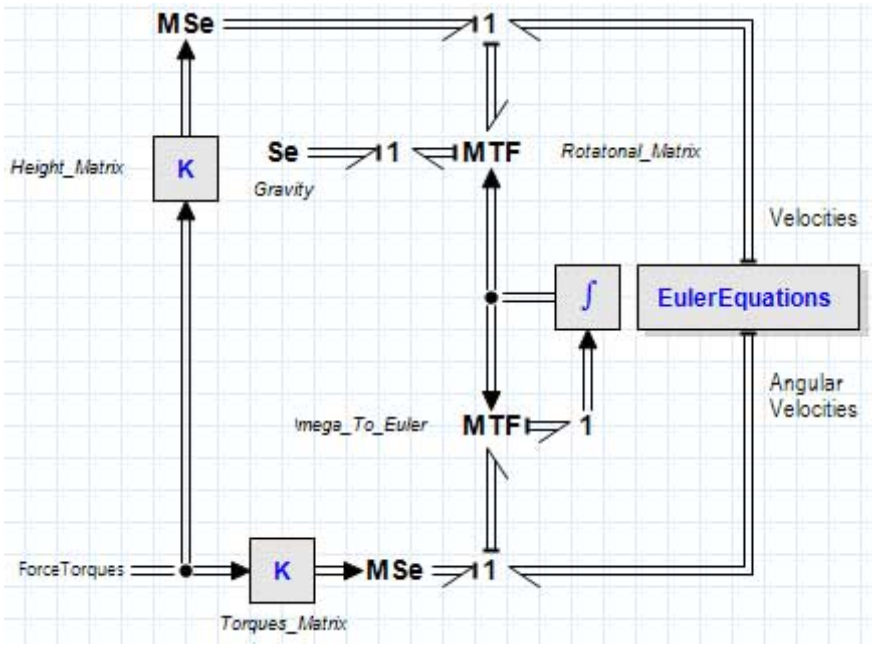

Figure 5.6: Quadrotor dynamic model in 20Sim

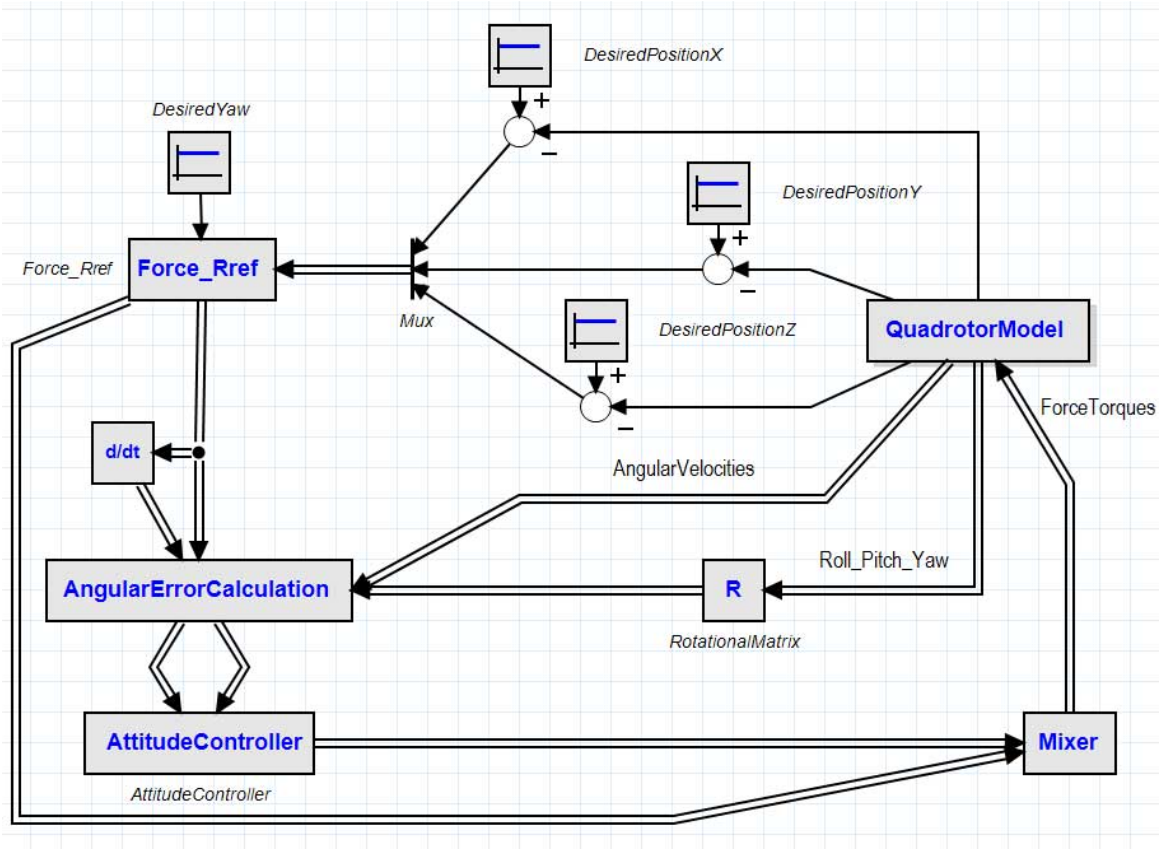

Figure 5.7: The control system of quadrotor in 20Sim

5.3 Simulation Results

5.3.1 Test in 20Sim

Following table shows the simulation parameters obtained at [82]. In addition, as the position tracking error can provide some negative values which makes the $\mathbf{b}_3^{ref}$ vector not direct on positive Z direction w.r.t $\{E\}$ frame. Quadrotor in this case will rotate 180 degree of Pitch. To prevent this case, we set the threshold value for $\mathbf{F}$'s elements. In simulation, the drag force and torques are negligible. The reference values for Roll - Pitch angles are always set to be zero.

Table 5.2: Some specifications of the simulation

Parameters	Unit	Value	Parameters	Unit	Value
m	kg	0.82	$K_{\mathbf{\Omega}}$		1
d	m	0.29	$K_{\mathbf{R}}$		2
I_{xx}	$kg.m^2$	0.0081	K_r		1.1
I_{yy}	$kg.m^2$	0.0081	K_v		1.1
I_{zz}	$kg.m^2$	0.0162	ω	rad / s	0.5
X_{Origin}	m	0.0	radius	m	1.0
Y_{Origin}	m	0.0	slope	1 / s	0.1
			$start\_time$	s	1

The desired values of position X, Y and Z according to t time is as in the following equations:

$$Circle = \begin{cases} X_{Circle} = X_{Origin} + radius \cdot \sin(\omega \cdot t) \\ Y_{Circle} = Y_{Origin} + radius \cdot \cos(\omega \cdot t) \\ Z_{Circle} = 0 \end{cases} \tag{5.22}$$

$$Spiral = \begin{cases} X_{Spiral} = X_{Origin} + radius \cdot \sin(\omega \cdot t) \\ Y_{Spiral} = Y_{Origin} + radius \cdot \cos(\omega \cdot t) \\ Z_{Spiral} = slope \cdot \text{ramp}(start\_time) \end{cases} \tag{5.23}$$

The proposed controller is tested with many cases:

- Simulation I (Fig. 5.8): tests with step input to reach a desired position $[1; 1; 1]$ in ideal environment.

- Simulation II (Fig. 5.9): tests tracking 2D circular trajectory in ideal environment. The desired heading angle is set to be zero.

- Simulation III (Fig. 5.10): tests tracking 3D spiral trajectory in ideal environment. The desired heading angle is set to be zero.

- Simulation IV (Fig. 5.11): tests tracking 3D spiral trajectory with gaussian noise. The gaussian noise is produced by $SignalGenerator - GaussianNoise$ block with following parameters: $amplitude = 0.01$; $seed = 0.0$; $frequency = 10.0\{Hz\}$. The desired heading angle is set to be zero.

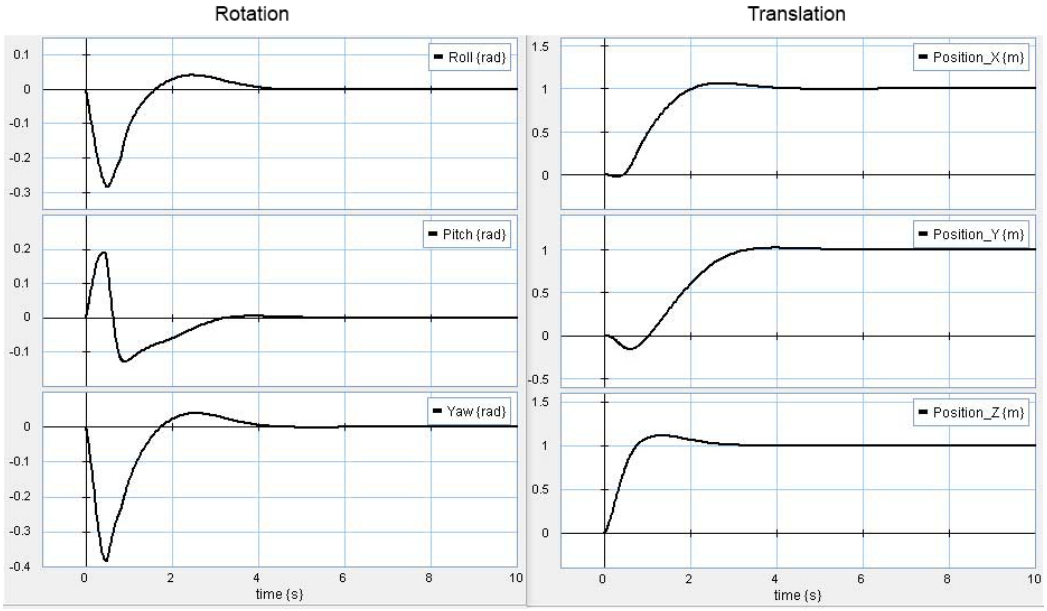

Figure 5.8: Simulation I reaching the desired position [1;1;1]

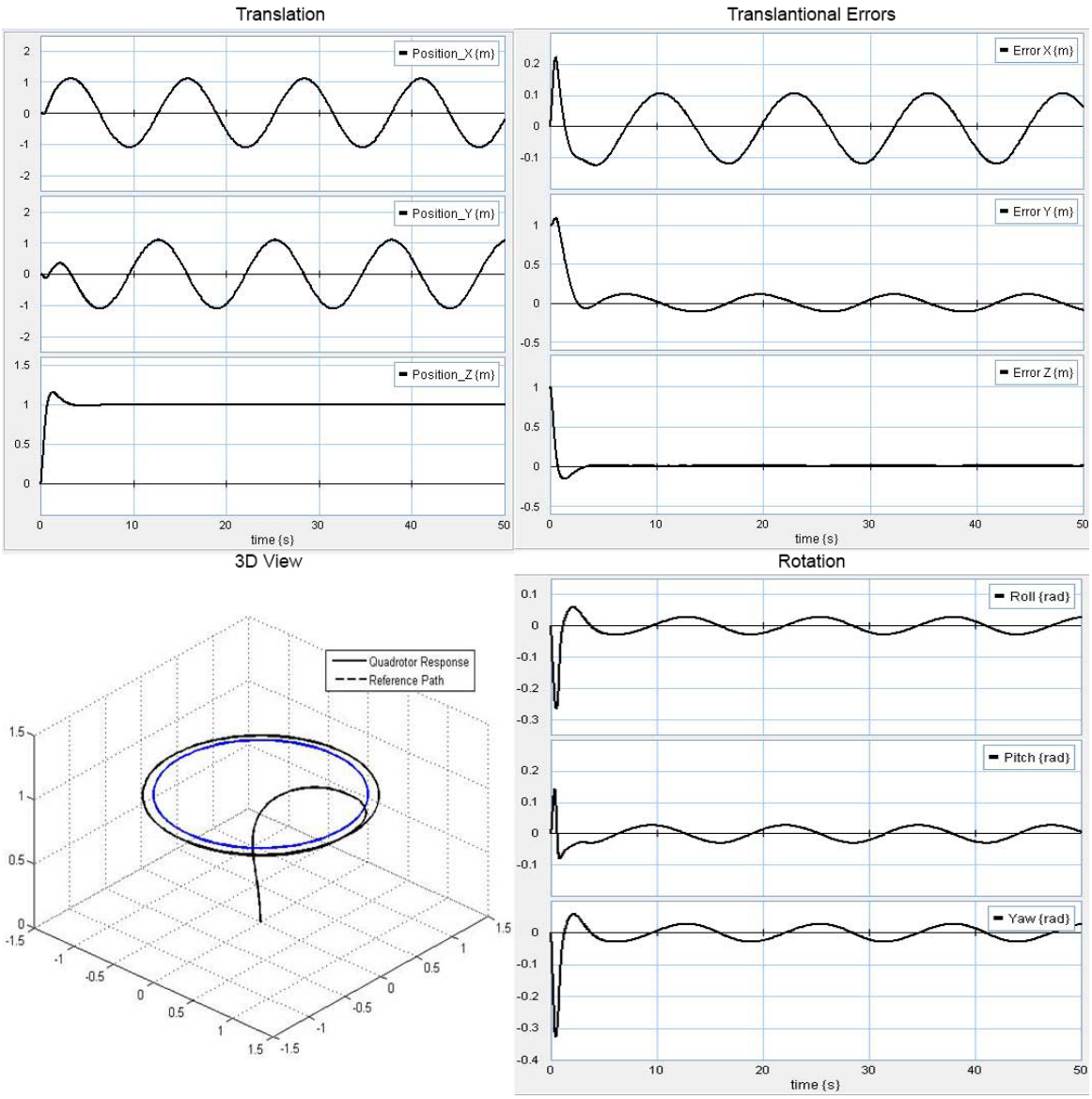

Figure 5.9: Results of simulation II

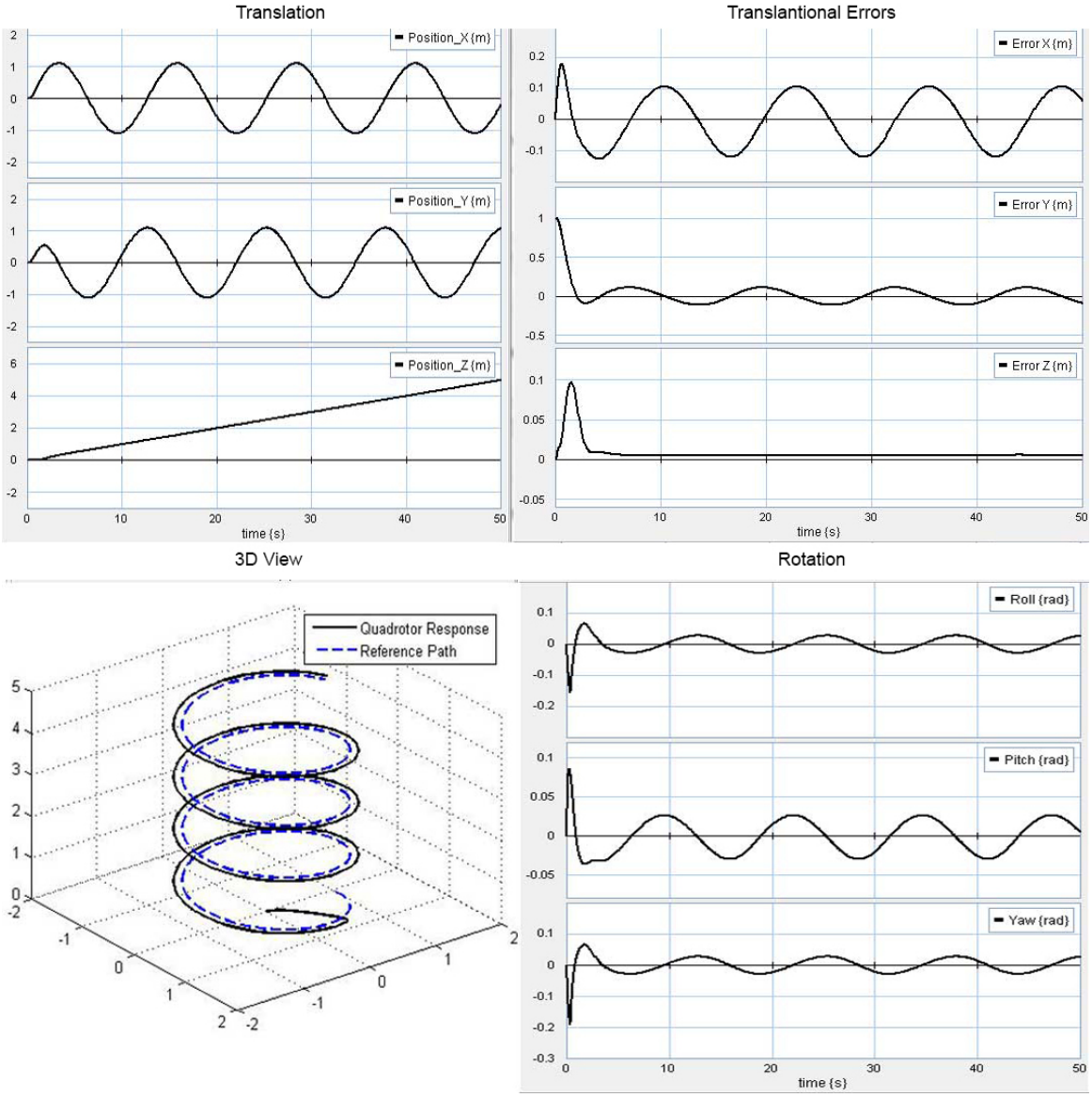

Figure 5.10: Results of simulation III

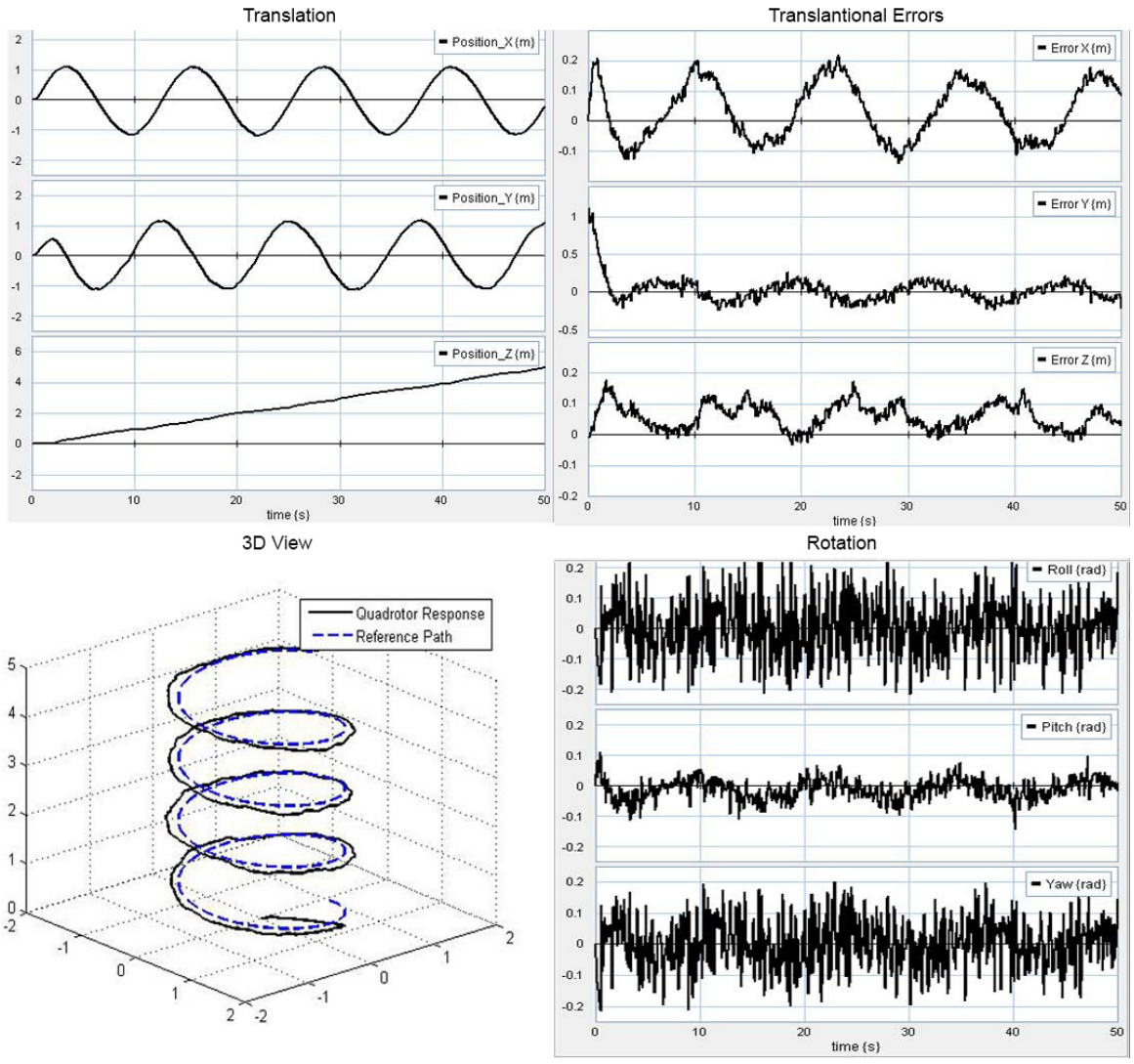

Figure 5.11: Results of simulation IV

These simulations I, II and III focus on the desired position without changing the heading (Yaw angle is kept zero). In simulation I, quadrotor can reach and hover about a position $[1; 1; 1]$ after 3 seconds. In simulation II and simulation III, quadrotor can track the desired trajectory. Quadrotor is begun at the initial position which is far from the desired trajectory. In 5 seconds, the controller produces efforts making quadrotor fly closed to the desired trajectory. After that, quadrotor keeps track of the desired trajectory. The performance shows that the proposed controller can enable quadrotor for aggressive trajectory tracking. Simulation IV shows the trajectory tracking with additional Gaussian noise. The controller is tested in noisy environment. The noise is considered as the effects of aerodynamic forces and winds. Working in noisy condition, the controller can keep quadrotor tracking the desired spiral trajectory. For discussion, the thesis has applied bond-graph modeling method and geometric control theory into controlling quadrotor. Applying geometric control, the thesis can overcome some linear approximation steps for controller design and preserve geometric dynamic properties of quadrotor.

Generally, geometric control theory is applied in order to overcome the nonlinearity of quadrotor's dynamic model. Geometric control handles nonlinearity by processing the input signals of the controller and defining the exact errors of rotational matrix. Compared to previous PID controller design [81, 59], the thesis decreases the number of chosen parameters (four parameters of the controller: K_r, K_v, $K_{\mathbf{R}}$ and $K_{\mathbf{\Omega}}$).

5.3.2 Test in Robot Operating System (ROS)

The VT&R system is developed in ROS with Gazebo simulator. After verifying the performance of the nonlinear geometric control in 20Sim, the controller is implemented into ROS system in order to integrate with other parts (Fig. 1.2) in route-following applications. Fig. 5.12 presents testing trajectory control of quadrotor in ROS before

applying into VT&R system. The video can be located at *youtu.be/s5JLUR1HaXw*. In the first 20s, quadrotor needs to generate enough forces to take off, which produces overshoot response in tracking errors. Then, quadrotor can perform trajectory control to follow the desired trajectory. The integration with other parts in VT&R system has been presented in chapter 4 with experiment V and VI.

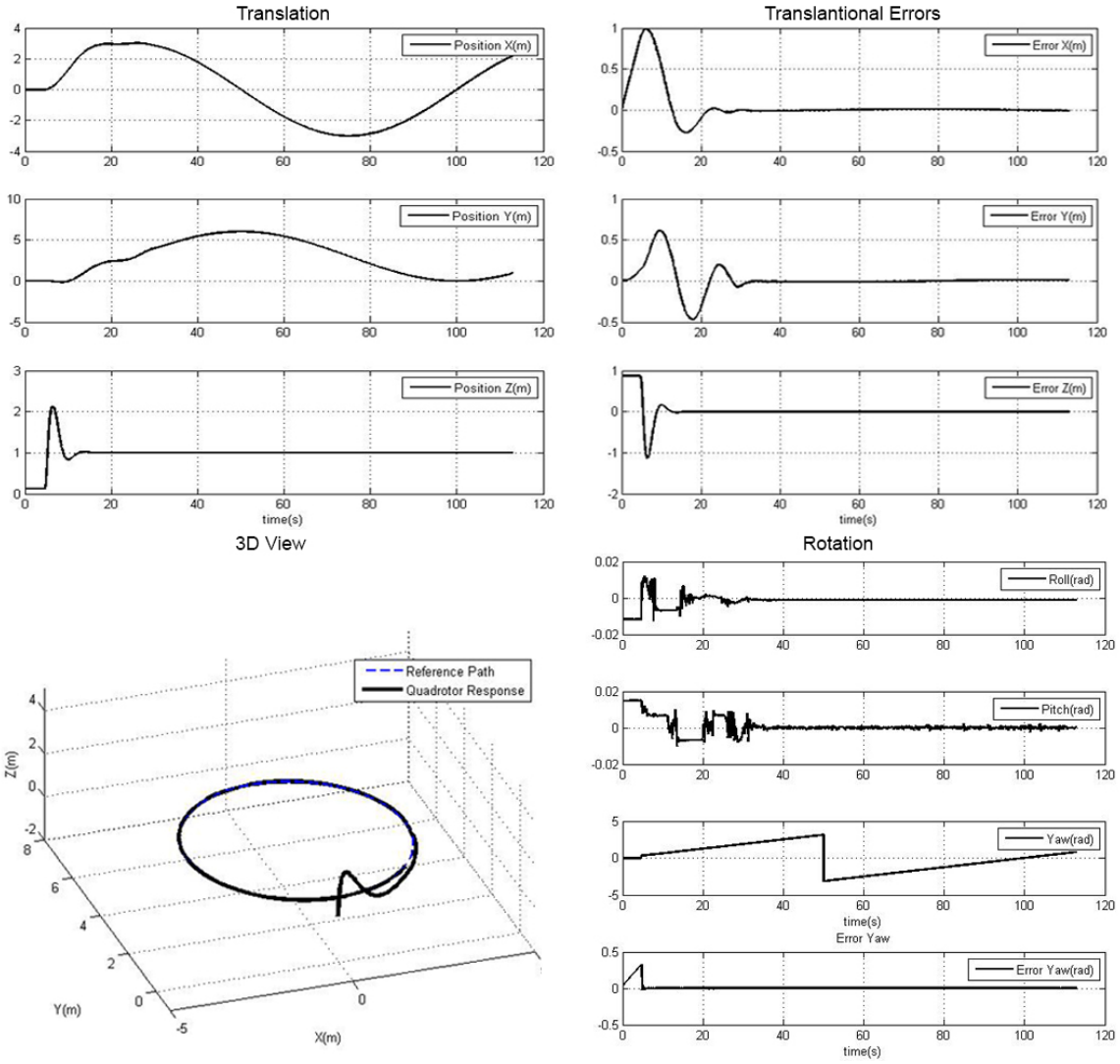

Figure 5.12: Test the nonlinear geometric controller in ROS

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The thesis has developed qualitative VT&R technique on quadrotor aerial vehicle in order to navigate in GPS-denied environment. The proposed design of VT&R aerial system is constructed using three different components: self-localization, qualitative motion control command developed on Funnel Lane theory, and controller using nonlinear geometric control. Firstly, three methods of self-localization technique are proposed to improve the performance of estimating the current segment of quadrotor. Utilizing the relation between spatial distance and the size of SURF feature, method II and method III show better self-localization than method I. Method II produces insignificantly better self-localization than method III but shows some disadvantages at the uniformity of the probability distribution. Secondly, qualitative motion control command part is developed on Funnel Lane theory to decrease the computational cost and to reach the simpler form of motion control command calculation. Qualitative motion control commands are calculated by funnel-lane 3D motion control algorithm. Lastly, nonlinear geometric control is applied into the controller of the quadrotor.

Attitude control of quadrotor is performed on directly computing errors of rotation matrix, which helps to overcome some assumptions in traditional quadrotor controller. The proposed design of VT&R aerial system is implemented on ROS for simulation and realtime experiments with Ar.Drone quadrotor aerial vehicle. When combining three parts together and testing in different experimental scenarios, quadrotor can show the ability to follow the desired route. The applications of the proposed technique are visual-homing when quadrotor is working with other ground vehicles, as well as visual-guiding in GPS-denied environment. When developing and implementing VT&R technique to navigate Ar.Drone quadrotor follow the desired route, the experiments reveal some disadvantages and challenges, which are defined as follows:

- *Vulnerable to ambient lighting*: In the replaying phase, navigation is performed by the similarity between the current image and the reference images such as landmark features. The changes of ambient lighting can adversely affect the calculation of the matched features, that can make the VT&R system fail. The works, presented in [49, 2], use KLT features in indoor environment with the assumption of trivial changes of lighting conditions between the learning phase and the replaying phase. However, KLT features are sensitive to the changes in ambient lighting. When the results of feature matching technique are not accurate and stable enough for localization, the navigation can not control the quadrotor along the desired route. To address this issue, the thesis proposes to use SURF features in order to have more reliable matched features for navigation. If the lighting condition significantly changes such as night and day, the improvement to use more effective system of sensors is necessary. The use of LIDAR system [41] on ground vehicle has been proved to overcome some of the problems. Nevertheless, using LIDAR system is limited for quadrotor, which has limitation of payload.

- *Dynamic environment*: The experiments are performed with the assumption of static environment. In reality, the work site of the vehicle can be a dynamic environment. The appearance, disappearance and movement of temporarily physical parts surrounding or on the desired route (humans, other vehicles, furniture, and doors) can lead to the failure of the VT&R system. However, the use of SURF features for navigation somewhat overcomes the issue. Features representing the landmarks of the temporarily physical parts will not be matched with SURF features on the reference images. These unwanted features can be used for obstacle avoidance in the appearance-based controller. Utilizing the method, proposed in the work [68], the quadrotor can have ability to detect and avoid unexpected obstacles in the relaying phase. Combining the SURF features of the obstacle with the technique of template matching, the method can estimate the position of obstacle. However, the avoiding step is still limited in 2D performance, which needs to be extended for 3D performance.

- *Memory Consumption*: The structure of VT&R system requires considerable memory to store the reference images of the desired route. The quality and quantity of the reference images decide the size of required memories. Hence, an effective method to store and update the database of the desired route is also needed. Due to the limitation of Ar.Drone quadrotor hardware, the thesis does not attempt to store the database of the desired route onboard, which removes the independent operations of VT&R technique. However, this problem can be easily overcome by the additional flight recorder onto Ar.Drone quadrotor hardware [3] or using other models of quadrotor. On the other hand, for practical applications, the VT&R does not need to store the whole set of reference images. Storing the SURF features and its descriptors is enough to perform feature matching technique.

- *How are many reference images are enough to represent the desired route?*: A number of reference images decide the requirement of memory, as well as partly influence the long distance of traveling. The curve route requires more reference images than the straight route. When processing with the four databases, the thesis chooses the reference images so that the length of the segment is at-least 1m and the qualitative difference between I_s^{Ref} image and the I_{s+1}^{Ref} image is sufficient to localize. In the initial works of the thesis [79, 49], the segments of the desired route are divided basing on a number of matched features between the I_s^{Ref} image and the I_{s+1}^{Ref} image. This method is appropriate to apply for practical applications because the quadrotor does not need to remember all images feedback. However, the way to chose the threshold value for declaring a new segment or for remembering the reference image is challenging because of the rotational motions and drifts of quadrotor.

- *The influence of the errors of controller response*: Initial experiments [49] with PID controller reveal the problems of the controller. The overshoot of performance and the vulnerability in noisy environments motivates the study of advanced control strategies for the quadrotor. The nonlinear geometric controller can satisfy the requirement of VT&R system with simpler calculation and inexpensive computational cost. However, the performance of the controller still has overshoot because of the chosen parameters of control gains.

- *The position errors in route following*: In VT&R technique, the navigation and control are basing on the reference images. Therefore, the achieved accuracy of the system is not more significant than the other strategy with VICON motion capture system [5, 31, 32, 33]. However, these errors are acceptable in exploring and searching-and-rescuing missions.

- *The optimal combination between feature detection and feature descriptor for matching*: In the thesis, SURF features are detected in the current image and the reference images. Feature matching between images is performed by BRIEF descriptors. Basing on the experiments, this combination can satisfy the requirements of the system, but maybe not optimal combination. The optimal combination becomes interesting topic to study in the future work.

6.2 Future Work

In order to increase the stability and reliability of the VT&R system for practical applications, additional parts such as obstacle avoidance, memory management will be developed. Obstacle avoidance and path planning will handle the unexpected appearance of obstacles on the desired route in the replaying phase. Memory management is to effectively manage and adaptively update the database of the desired route in dynamic environment. Path planning shows potential to develop in case of multiple databases, which allows many ways to reach destination. In addition, the ROS system should be redesigned to enable onboard processing and fully autonomous applications. One of the reason for the failure of the VT&R system is the changes of the working environment between the learning phase and the replaying phase. It requires another effective fusion of numerous sensors and advanced navigating calculation to help quadrotor working in dynamic environment but still satisfies the limitation of quadrotor's payload. The navigation on Funnel Lane theory can be extended for the case of bottom camera.

The study of nonlinear geometric control theory opens potential research projects. The error calculation on rotation matrix simplifies the structure of the quadrotor controller as well as enables computational resources for other computations, which

can help to improve the performance of the quadrotor controller. The current geometric controller can be improved by better methods for tuning controller gains. Additional estimations or observers have potentials to study in order to improve the accuracy of position control.

Bibliography

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[2] Z. Chen and S. Birchfield, "Qualitative vision-based path following," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 749–754, 2009.

[3] Parrot AR.Drone ver 2.0 quadrotor, http://ardrone2.parrot.com.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[5] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The GRASP multiple micro-UAV testbed," *IEEE Robotics Automation Magazine*, vol. 17, pp. 56–65, September 2010.

[6] C. Dillow. (2013, May) Norwegian geologists begin drone-guided quest for oil. [Online]. Available: www.popsci.com/technology/article/2013-05/norwegian-geologists-are-searching-oil-drones

[7] R. Koronowski. (2013, Aug.) FAA approves use of drones by ConocoPhillips to monitor oil drilling activities in Alaska. [Online]. Available: thinkprogress.org/climate/2013/08/26/2524731/drones-conocophillips-alaska/

[8] K. Krishnamurthy. (2013, Jun.) In Alaska's oilfields, drones count down to takeoff. [Online]. Available: www.reuters.com/article/2013/06/07/drones-oil-idUSL3N0E828H20130607

[9] E. Mitsui. (2013, Jul.) Commercial drones could have 'endless' uses under Canada 's laws. [Online]. Available: www.cbc.ca/news/technology/commercial-drones-could-have-endless-uses-under-canada-s-laws-1.1386300

[10] Q. Zhou and Y. Zhang, "Dead reckoning and Kalman filter design for trajectory tracking of a quadrotor UAV," in *Proc. of IEEE/ASME International Conference on Mechatronics and Embedded Systems*, 2010, pp. 119–124.

[11] L. E. Munoz, P. Castillo, and P. Garcia, "Observer-control scheme for autonomous navigation: Flight tests validation in a quadrotor vehicle," *Proc. of International Conference on Unmanned Aircraft Systems*, pp. 795–804, May 2013.

[12] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained MAV," in *Proc. of IEEE International Conference on Robotics and Automation*, May 2011, pp. 20–25.

[13] A. Bachrach and S. Prentice, "RANGE: Robust autonomous navigation in GPS-denied environments," *Journal of Field Robotics*, vol. 28, no. 5, pp. 644–666, 2011.

[14] S. Hochdorfer and C. Schlegel, "6 DoF SLAM using a ToF camera: The challenge of a continuously growing number of landmarks," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 3981–3986.

[15] F. Fraundorfer and L. Heng, "Vision-based autonomous mapping and exploration using a quadrotor MAV," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 4557–4564.

[16] C. Fu, M. a. Olivares-Mendez, R. Suarez-Fernandez, and P. Campoy, "Monocular visual-inertial SLAM-based collision avoidance strategy for fail-safe UAV using fuzzy logic controllers," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1-4, pp. 513–533, Oct. 2013.

[17] C. Forster, M. Pizzoli, and D. Scaramuzza, "Air-ground localization and map augmentation using monocular dense reconstruction," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, November 2013, pp. 3971–3978.

[18] J. Eckert, R. German, and F. Dressler, "An indoor localization framework for four-rotor flying robots using low-power sensor nodes," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 2, pp. 336–344, 2011.

[19] ——, "Self-Organized localization in indoor environments using the ALF framework," *International Journal of Communication Networks and Distributed Systems*, vol. 10, no. 2, pp. 102–122, 2013.

[20] O. De Silva, G. K. I. Mann, and R. G. Gosine, "Development of a relative localization scheme for ground-aerial multi-robot systems," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2012, pp. 870–875.

[21] C. Bills, J. Chen, and A. Saxena, "Autonomous MAV flight in indoor environments using single image perspective cues," in *Proc. of IEEE International Conference on Robotics and Automation*, 2011.

[22] D. Lee, T. Ryan, and H. Kim, "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing," *Proc. of IEEE International Conference on Robotics and Automation*, pp. 971–976, 2012.

[23] W. Li, T. Zhang, and K. Kuhnlenz, "A vision-guided autonomous quadrotor in an air-ground multi-robot system," in *Proc. of IEEE International Conference on Robotics and Automation*, 2011.

[24] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long range rover autonomy," *Journal of Field Robotics*, vol. 27, no. 2006, pp. 534–560, 2010.

[25] T. D. Barfoot, B. Stenning, P. Furgale, and C. McManus, "Exploiting reusable paths in mobile robotics: benefits and challenges for long-term autonomy," in *Proc. of Conference on Computer and Robot Vision*, May 2012, pp. 388–395.

[26] P. Vandrish, A. Vardy, and P. King, "Towards AUV route following using qualitative navigation," in *Proc. of IEEE Canadian Conference on Computer and Robot Vision*, May 2012, pp. 425–432.

[27] J. Courbon, Y. Mezouar, and P. Martinet, "Autonomous navigation of vehicles from a visual memory using a generic camera model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 392–402, 2009.

[28] A. Pfrunder, A. P. Schoellig, and T. D. Barfoot, "A proof-of-concept demonstration of visual teach and repeat on a quadrocopter using an altitude sensor and a monocular camera," in *Proc. of Canadian Conference on Computer and Robot Vision*, 2014.

[29] Robot Operating System (ROS). [Online]. Available: http://www.ros.org

[30] V. Kumar and N. Michael, "Opportunities and challenges with autonomous micro aerial vehicles," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1279–1291, Aug. 2012.

[31] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 2520–2525.

[32] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, "Towards a swarm of agile micro quadrotors," *Autonomous Robots*, vol. 35, no. 4, pp. 287–300, Jul. 2013.

[33] S. Lupashin, M. Hehn, M. W. Mueller, A. P. Schoellig, M. Sherback, and R. D'Andrea, "A platform for aerial robotics research and demonstration: The Flying Machine Arena," *Mechatronics*, vol. 24, no. 1, pp. 41–54, Feb. 2014.

[34] N. Michael, S. Shen, K. Mohta, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, E. Takeuchi, and S. Tadokoro, "Collaborative mapping of an earthquake damaged building via ground and aerial robots," in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, K. Yoshida and S. Tadokoro, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, vol. 92, pp. 33–47.

[35] Y. Pang, Y. Sun, Y. Gan, and L. Wan, "An integrated GPS/DR navigation system for AUV," *Journal of Marine Science and Application*, vol. 5, no. 4, pp. 8–13, Dec. 2006.

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

[37] S. A. Scherer and A. Zell, "Efficient onbard RGBD-SLAM for autonomous MAVs," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1062–1068, Nov. 2013.

[38] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a navigation system for autonomous indoor flying," in *Proc. of IEEE International Conference on Robotics and Automation*, 2009, pp. 2878–2883.

[39] G. H. Lee, M. Achtelik, F. Fraundorfer, M. Pollefeys, and R. Siegwart, "A benchmarking tool for MAV visual pose estimation," in *Proc. of 11th International Conference on Control Automation Robotics & Vision*, vol. 1, Dec. 2010, pp. 1541–1546.

[40] P. Furgale and T. Barfoot, "Stereo mapping and localization for long-range path following on rough terrain," in *Proc. of IEEE International Conference on Robotics and Automation*, May 2010, pp. 4410–4416.

[41] C. McManus, P. Furgale, B. Stenning, and T. D. Barfoot, "Visual teach and repeat using appearance-based lidar," in *Proc. of IEEE International Conference on Robotics and Automation*, May 2012, pp. 389–396.

[42] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, "Visual teach and repeat, repeat, repeat: Iterative Learning Control to improve mobile robot path tracking in challenging outdoor environments," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 176–181.

[43] J. Courbon and Y. Mezouar, "Visual navigation of a quadrotor aerial vehicle," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2009, pp. 5315–5320.

[44] S. Erhard, K. E. Wenzel, and A. Zell, "Flyphone: Visual self-localisation using a mobile phone as onboard image processor on a quadrocopter," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, pp. 451–465, Sep. 2009.

[45] A. L. Majdik, Y. Albers-Schoenberg, and D. Scaramuzza, "MAV urban localization from Google street view data," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 3979–3986.

[46] A. Vardy, "Using feature scale change for robot localization along a route," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 4830–4835.

[47] Z. Chen and S. T. Birchfield, "Qualitative vision-based mobile robot navigation," in *Proc. of IEEE International Conference on Robotics and Automation*, no. May, 2006, pp. 2686–2692.

[48] H. M. Becerral, J. Courbon, Y. Mezouar, and C. Sagues, "Wheeled mobile robots navigation from a visual memory using wide field of view cameras," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 5693–5699.

[49] T. Nguyen, G. K. Mann, and R. G. Gosine, "Vision-based qualitative path-following control of quadrotor aerial vehicle," in *Proc. of International Conference on Unmanned Aircraft Systems*, 2014.

[50] G. Vaca-Castano, a. R. Zamir, and M. Shah, "City scale geo-spatial trajectory estimation of a moving camera," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2012, pp. 1186–1193.

[51] A. M. Zhang and L. Kleeman, "Robust appearance based visual route following for navigation in large-scale outdoor environments," *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 331–356, Mar. 2009.

[52] W. Liu, N. Zheng, J. Xue, X. Zhang, and Z. Yu, "Visual appearance-based unmanned vehicle sequential localization," *International Journal of Advanced Robotic Systems*, vol. 10, 2013.

[53] A. Dame and E. Marchand, "A new information theoretic approach for appearance-based navigation of non-holonomic vehicle," *Proc. of IEEE International Conference on Robotics and Automation*, pp. 2459–2464, May 2011.

[54] K. Alexis, C. Papachristos, R. Siegwart, and A. Tzes, "Robust explicit model predictive flight control of unmanned rotorcrafts : design and experimental evaluation," in *Proc. of European Control Conference*, 2014.

[55] M. Abdolhosseini, Y. M. Zhang, and C. A. Rabbath, "An efficient model predictive control scheme for an unmanned quadrotor helicopter," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1-4, pp. 27–38, Aug. 2012.

[56] A. Mcfadyen, L. Mejias, P. Corke, and C. Pradalier, "Aircraft collision avoidance using spherical visual predictive control and single point features," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 50–56.

[57] J. Pestana, "Computer vision based general object following for GPS-denied multirotor unmanned vehicles," in *Proc. of American Control Conference*, 2014, pp. 1886–1891.

[58] R. L. Pedro Castillo, Alejandro Dzul, "Real-Time stabilization and tracking of a four-rotor mini rotorcraft," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 4, pp. 510–516, 2004.

[59] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. Stryk, "Comprehensive simulation of quadrotor uavs using ros and gazebo," in *Simulation, Modeling,*

and Programming for Autonomous Robots. Springer Berlin Heidelberg, 2012, vol. 7628, pp. 400–411.

[60] M. Orsag, C. Korpela, M. Pekala, and P. Oh, "Stability control in aerial manipulation," in *Proc. of American Control Conference*, 2013, pp. 5581–5586.

[61] D. C. Karnopp, D. L. Margolis, and R. C. Rosenberg, *System dynamics: modeling and simulation of mechatronic systems*, 4th ed. Wiley-Interscience, 2005.

[62] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision," *Autonomous Robots*, vol. 33, no. 1-2, pp. 21–39, Feb. 2012.

[63] M. Amoozgar, A. Chamseddine, and Y. Zhang, "Fault-tolerant fuzzy gain scheduled PID for a quadrotor helicopter testbed in the presence of actuator faults," in *Proc. of IFAC Conference on Advances in PID Control*, no. 2002, 2012.

[64] R. S. Samir Bouabdallah, Andre Noth, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, 2004, pp. 2451–2456.

[65] T. Lee, M. Leoky, and N. McClamroch, "Geometric tracking control of a quadrotor uav on se(3)," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, 2010, pp. 5420–5425.

[66] F. Bullo and A. D. Lewis, *Geometric control of mechanical systems*, ser. Texts in Applied Mathematics. New York-Heidelberg-Berlin: Springer Verlag, 2004, vol. 49.

[67] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. I–511–I–518.

[68] T. Mori and S. Scherer, "First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles," in *Proc. of IEEE International Conference on Robotics and Automation*, May 2013, pp. 1750–1757.

[69] A. Murillo, "Surf features for efficient robot localization with omnidirectional images," in *Proc. of IEEE International Conference on Robotics and Automation*, no. April, 2007, pp. 10–14.

[70] C. Valgren and A. J. Lilienthal, "SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments," *Robotics and Autonomous Systems*, vol. 58, no. 2, pp. 149–156, Feb. 2010.

[71] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: Computing a local binary descriptor very fast," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Nov. 2011.

[72] T. Krajnik, P. De Cristóforis, J. Faigl, H. Szücsová, M. Nitsche, L. Preucil, and M. Mejail, "Image features for long-term mobile robot autonomy," *IEEE Conference on Robotics and Automation, Workshop on Long-Term Autonomy*, 2013.

[73] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Proc. of European Conference on Computer Vision*, 2010.

[74] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the openCV library*. O'Reilly Media, September 2008.

[75] A. Cartography and M. Park, "Random Sample Consensus: A paradigm for model fitting with," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[76] D. F. Sebastian Thrun, Wolfram Burgard, *Probabilistic Robotics*. MIT Press, 2005.

[77] A. Pronobis and B. Caputo, "COLD: The CoSy Localization Database," *The International Journal of Robotics Research*, vol. 28, no. 5, pp. 588–594, May 2009.

[78] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, "The New College vision and laser data set," *The International Journal of Robotics Research*, vol. 28, no. 5, pp. 595–599, May 2009.

[79] T. Nguyen, G. K. Mann, and R. G. Gosine, "Vision-Based qualitative path-following control of quadrotor aerial vehicle with speeded-up robust features," in *Proc. of Canadian Conference on Computer and Robot Vision*, May 2014, pp. 321–327.

[80] T. Tomic, "Evaluation of acceleration-based disturbance observation for multi-copter control," in *Proc. of European Control Conference*, Jun. 2014, pp. 2937–2944.

[81] D. G. R. M. Raju Hossain and D. N. Krouglicof, "Bond graph dynamic modeling and stabilization of a quad-rotor helicopter," in *11th International Conference on Bond Graph Modeling and Simulation (ICBGM 2010)*, 2010.

[82] J. M. B. Domingues, "Quadrotor prototype," Master's thesis, Instituto Superior Tecnico, UT Lisbon, 2009.