The Pennsylvania State University
The Graduate School
College of Engineering

A QUASI-POLAR LOCAL OCCUPANCY GRID APPROACH FOR VISION-BASED OBSTACLE AVOIDANCE

A Thesis in
Aerospace Engineering
by
Junyi Geng

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2016

The thesis of Junyi Geng was reviewed and approved* by the following:

Jacob W. Langelaan
Associate Professor of Aerospace Engineering
Thesis Advisor

Joesph F. Horn
Professor of Aerospace Engineering

George A. Lesieutre
Professor of Aerospace Engineering
Head of the Department of Aerospace Engineering

*Signatures are on file in the Graduate School.

Abstract

The motivation behind the research described in this thesis is simplifying the process of small unmanned aerial vehicles (UAVs) navigation through an obstacle field: typical methods follow a two step process of map generation followed by trajectory computation; the method proposed here directly computes the likelihood that a given path is collision-free. Trajectory generation simply consists of selecting the path that is likely to be free over the longest distance. While the approach described here is independent of the choice of sensor, this thesis focuses on vision-based obstacle avoidance.

This thesis proposes a quasi-polar (turn rate-time) local occupancy grid approach for obstacle avoidance. It uses GPS and inertial navigation combined with a vision system to map sensor data directly onto dynamically feasible paths, so that path planning consists simply of selecting the path with lowest likelihood of collision.

This approach brings many new challenges besides the usual challenges posed by vision-based obstacle mapping (i.e. noisy vision measurements and highly non-linear governing equations of system lead to significant uncertainties). First, the mapping problem is not performed in the traditional Cartesian or polar coordinate frame. Second, the motion update is difficult since the cells in the quasi-polar map have different shapes and size. The boundary of grid cells can be straight lines or arcs, hence classical image processing methods cannot be used for the motion update.

These challenges are addressed by: (1) inverse sensor models that map directly onto the quasi-polar grid; (2) a numerical approach for motion updates.

Three exteroceptive sensor models (wide field monocular vision, pushbroom stereo, and pushbroom stereo combined with wide field monocular) are presented in this context. Two path selectors considering the occupied probability differently are created. Simulations of flight through a two dimensional environment consisting of both forest and urban terrain are used to demonstrate the utility of this approach. Preliminary work towards hardware mainly for vision system is presented to show the feasibility of sensor models used in simulation.

Contents

List of Figures

List of Tables

Acknowledgments

Though only my name appears on the cover of this thesis, a great many people have contributed to its production. I own my gratitude to all those people without whom the work would not be possible.

My deepest gratitude is to my advisor, Dr. Jack Langelaan. I have been amazingly fortunate to have an advisor who gave me the freedom to explore on my own as well as the guidance to proceed when my research faltered. Jack taught me how to smartly use interdisciplinary point of view to solve problems. I appreciate his rich ideas, vast knowledge and skills in many areas, and his assistance in writing papers. His patience and support helped me overcome many crisis situations and finish this thesis.

I am grateful to Dr. Joe Horn for his practical advice. Thank him for reviewing my thesis, commenting on my views and helping me understand and enrich my ideas. His teaching in 518 has a significant influence in my research style. He emphasized much on the fundamental and often presented the complex concepts and relate theoretical concepts in a clear way. He sets high standards and encourages and guides us to meet those standards. I am indebted to him for his guidance.

I feel lucky to join the AVIA lab. My labmates have exchanged ideas with me and offered advice. I deeply appreciate their friendship that helped me adjust to a new country. I'm especially grateful to Nathan Depenbusch, John Bird and William Holmes for helping me get started and develop my own research in the lab. Nate has been my labmate for one and a half years. He was like a lab captain who always ready to answer all kinds of my questions. He helped me improve my English, discussed with me the difference of culture, politics and custom between countries. He has helped create very relaxed atmosphere in the lab which makes everyday's work an enjoyable experience. Bird gave me great assistance in electrical, software and computer issues. He is like a magician who can always help me overcome wired errors. He also engaged in conversation and discussion that helped direct my thought way outside the stuck box. The surprise he gave me on my birthday made me touched to tears, which made me feel the warm of the lab family. Will

has helped me enormously in technical problems especially computer vision. He often shared with me the most effective way he found, without which I would have taken more detours.

I am also thankful to my friends in electrical engineering, without whom I cannot image how I get out of the depression when I was stuck on my research.

None of this would have been possible without the love and patience of my family. Throughout my academic studies, my parents have been there in every way possible. Without their support, I know that I would not have had the opportunity to pursue my ambition. I would like to express my heart-felt gratitude to my family.

Finally, the greatest thanks to my fiance Daning. He is the only relative I have in this country. Although we research in different specialty of aerospace engineering, he often came up with ideas that inspired me to develop effective solutions to my problems. He is my best partner in research and life. I am ever-blessed with him and our future.

Dedication

To my parents and Daning.

Chapter 1
Introduction

This thesis presents a method for obstacle avoidance that uses a vision system and an occupancy grid based on a novel turn rate-time frame. Since the shape of the new frame is similar to the polar frame, we will refer to it as quasi-polar frame in the following. The main motivation is simplifying the process of navigation through an obstacle field: typical methods follow a two step process of map generation followed by trajectory computation; the method proposed here directly computes the likelihood that a given path is collision-free. Trajectory generation simply consists of selecting the path that is likely to be free over the longest distance. While the method can be used for other sensor types, the focus of this thesis is vision systems.

Besides the usual challenges posed by vision-based obstacle mapping (i.e. noisy vision measurements reduce the certainty of obstacle information; highly non-linear governing equations of system and measurements lead to significant uncertainties), the novel method brings new challenges. First, the mapping problem cannot be performed in the traditional Cartesian or polar coordinate frame. A new frame based on vehicle kinematics and associated transformations need to be established. Second, motion update for the novel occupancy grid is difficult since the cells in the new map have different shapes and size. The boundary of grid cells can be straight line or arc curve, which means the classic image processing method cannot be used.

This thesis will:

- Develop a framework for obstacle avoidance based on the quasi-polar local occupancy grid using vision method and GPS/INS. This involves definition of the quasi-polar occupancy grid, mathematical models for vehicle dynamics

and different camera models, and a method to mapping detected obstacles onto the quasi-polar map.

- Propose two path selectors and solution to the navigation problem based on the local map.
- Present simulation results to show the effectiveness of the designed system in avoiding obstacles. Compare and discuss the performance of different camera sensor models.
- Introduce preliminary work towards hardware mainly for vision system. The vision system setup and implementation of image processing show the feasibility of the sensor models used in simulation.

1.1 Motivation

Small UAVs have significant potential for application in fields ranging from search and rescue to surveillance and exploration. Due to their small size and good mobility, they can fly in complex indoor and outdoor environments. However, their small size also limits payload capacity, leading to restrictions in the available sensor package, power availability, and computational capabilities that can be carried on board. Further, many missions require flight in close proximity to obstacles, so the problem of safe navigation remains to be an area of significant research.

Current mature technologies for obstacle detection using onboard sensors rely generally on LIDAR, sonar or radar, which require large power and weight cost. Although they provide high-quality information (range and bearings to obstacles with low uncertainty), these sensors generally have trouble meeting the size, weight and power (SWAP) restrictions of small UAVs. Light weight cameras are a good solution due to the low cost and low power charge, but they still require computationally efficient machine vision algorithm that can be run onboard. Monocular vision has longer accessible range and faster processing ability, but it cannot observe the depth information of obstacles. Scale and depth information are available on stereo vision. However, small baseline limits the stereo only accessible to obstacles in a short range. In the animal world, birds with eyes on the sides of their heads have a wide visual field, useful for detecting predators, while those with eyes on the

front of their heads, such as owls, have binocular vision and can estimate distances when hunting [1]. Fig. 1.1 and Fig. 1.2 show the field of view of a bird's visual geometry, which typically includes a region of stereo vision and a wide region of monocular vision.

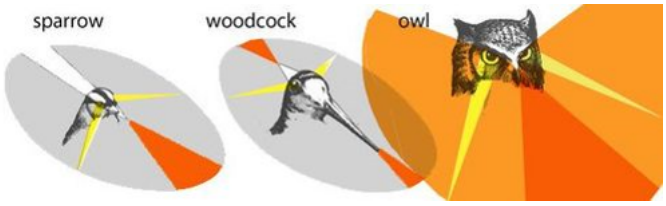


Figure 1.1. Bird vision [2]

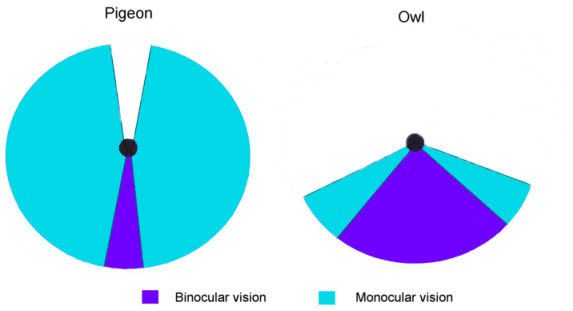


Figure 1.2. Region of bird vision [1]

A typical occupancy grid deals with obstacle mapping in a spatial lattice, usually under the Cartesian coordinate on a global dimension. Variation of occupancy grid have been explored by researchers [3–5]. In mobile robot navigation, grid-based representation of the environment are often used for planning a path from some initial robot location to a desired goal location. Existing methods for path planning are mainly based on graphic algorithms or potential field. However, all of the existing methods need two-step procedure of map generation followed by computation of feasible collision-free trajectory.

This thesis proposes a quasi-polar local occupancy grid that is based on a set of pre-calculated paths. Rather than following the two-step process of map generation followed by computation of a feasible collision-free trajectory, the method proposed here begins with a set of dynamically feasible paths and then uses exteroceptive

sensor data to compute the likely distance over which each path remains collision-free. A favorable path (such as the longest collision-free path) is then selected and followed. A procedure for generating a quasi-polar local map of a static environment using only GPS/INS and vision system is described. Different camera models are tested and compared: monocular vision with optical flow method, pushbroom stereo, a combination of monocular and stereo. New path selection methods based on the pre-calculated paths set are given.

1.2 System Overview

The problem considered here consists of a small UAV flying through an unsurveyed environment (such as a forest or an urban area) using an inertia measurement unit and vision system (one or two monocular cameras). The block diagram in Fig. 1.3 shows a system that uses the given sensors to perform obstacle avoidance.

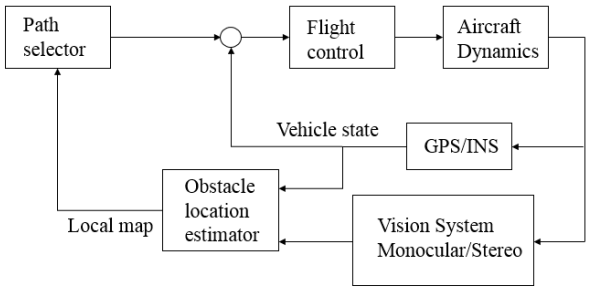
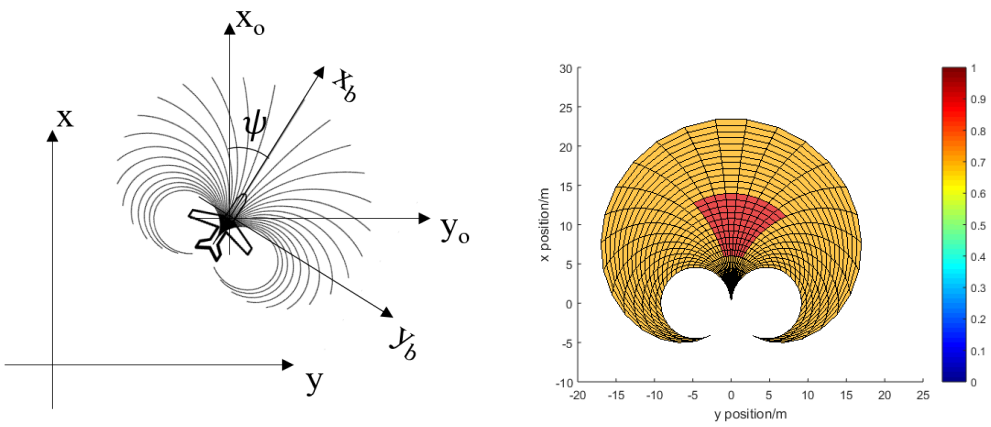


Figure 1.3. System block diagram

The system consists of three major parts: a stabilized aircraft, an obstacle location estimator, and a path selector. Here, the stabilized aircraft can maintain a desired flight condition. The GPS/INS sensors provides an estimate of vehicle state (position, orientation, velocity, and angular rates) as well as measurements of acceleration in the body frame. A vision system consisting of one or two monocular cameras obtains measurements of bearing (angle between the camera optical axis and an obstacle point), optical flow (rate of change of bearing) and depth information (for stereo) to features in the environment (which are assumed to be obstacles). The estimator uses the vision measurements combined with estimated vehicle states to compute a map of obstacle position estimates in local



(a) A set of curved feasible paths **(b) Local map**

Figure 1.4. Quasi-polar local map

quasi-polar frame. Note that this quasi-polar frame is based on flight paths that would result given constant speed, constant turn rate flight. The path selector uses the knowledge of vehicle states and the computed map to select a pre-calculated path(vehicle turn rate) to avoid obstacles.

1.3 Problem Description

The main goal is to develop a concise system framework with only one-step planning work which can successfully realize vision-based obstacle avoidance using only cameras and GPS/INS. This thesis presents solutions to the challenges faced in this problem: taking advantage of different kind of vision sensors and managing their noise, defining a quasi-polar local occupancy grid and developing the mapping, motion updating mechanism, estimating the obstacle location in the complex environment and select proper path based on the quasi-polar map. As shown in Fig. 1.4, a set of curved feasible paths are defined. A quasi-polar local map is generated from the paths.

1.3.1 Sensors usage

Monocular vision and stereo both have advantages and disadvantages. Measurements obtained by single cameras are very noisy. So a robust estimator is needed to get accurate obstacle location and to handle sensor dropout.

This thesis proposes a combination of the advantage of the monocular vision(longer range and faster processing) with that of stereo vision (availability of scale and depth information), while overcoming the disadvantages of monocular (unobservability and drift) and stereo (small baseline and shorter range). To combine data from those complementary sensors, this thesis proposes a version of the occupancy grid mapping algorithm. Rather than a traditional Cartesian grid, this thesis uses a grid defined by vehicle kinematics. There are two main advantages of map-based control/path planning over direct visual servoing: first, multiple measurements of the same obstacle reduce the effect of sensor noise; second, there is a retained memory of obstacle location that can be used in the event of sensor dropouts.

1.3.2 Estimation based on quasi-polar local map

Kalman filter based approaches work well in sparse environments where each feature can be assumed to represent an obstacle, but problems such as landmark initialization and data association must be addressed explicitly. An occupancy grid approach obviates the need for both landmark initialization and data association, but care must be taken in developing the appropriate sensor models. Typical occupancy grids can handle complex objects and portray their location accurately. However, the two-step path planning work is still required.

This thesis proposes a quasi-polar occupancy grid that operates directly on a set of dynamically feasible paths, so that trajectory planning consists simply of selecting the path that is free for a sufficient portion of its length. A new coordinate frame, time-turn rate frame is defined and the transformation between Cartesian frame or polar frame to the time-turn rate frame is presented. Obstacle information obtained from cameras is mapped on the quasi-polar local map. A numerical motion update method is developed to continuously update the map with information stored.

1.3.3 Path selection

The vehicle should navigate while avoiding obstacles along the path. The estimator and quasi-polar local map used in this thesis are designed to be convenient for planning trajectory. This thesis proposes two path selectors; one based on integrated

path free probability, the other based longest obstacle free time.

1.4 Related Work

Many researchers have explored the field of navigation and obstacle avoidance for autonomous vehicles. This section presents a detailed discussion of research in the fields of vision based navigation and obstacle avoidance, the history of occupancy grid and research on path planning.

1.4.1 Computer vision

Vision based navigation methods have been popular recently due to its low power and low weight. Monocular vision requires only a single, lightweight camera, which is readily available and easy to deploy so that this technology is attractive. The functional monocular vision based UAV PTAM (parallel tracking and mapping) proposed in [6] uses a downward facing camera to perform stable, drift-free and slow flight. It provides a powerful framework for real-time visual tracking and SLAM. However, this approach uses only a downward facing camera, which limits the application of obstacle detection for planning and control. Embedded optical flow techniques can extract depth information by computing the interframe changes between images. It has worked well on UAVs [7, 8]. Stereo vision systems can obtain depth information directly, but significant computational resources (as well as careful calibration) are required. Further, baseline limitations can lead to an inability to accurately compute range to distant features, which make them unsuitable for large spaces [9]. In [10, 11], Barry describes a pushbroom stereo algorithm for high-speed navigation in cluttered environments which performs a subset of standard block-matching stereo processing searching only for obstacles at a single depth. His system requires no external sensing or computation and is the first high-framerate stereo detection system running onboard a small UAV. Hrabar combined optical flow and stereo vision measurement on both a tractor and unmanned helicopter to fly in urban canyons in real time [12]. But this approach still requires two pair of cameras: one pair arranged for stereo vision, and the other a pair of fish-eye lens cameras arranged for optical flow measurements. He showed preliminary results that a UAV could potentially be controlled using optical flow

and stereo information from a single pair of forward-facing fisheye cameras.

1.4.2 Occupancy grids

Occupancy grids are a powerful mapping tool, first introduced by Elfes [13, 14]. In essence, the occupancy grid computes the likelihood that a particular cell in a lattice representation of the environment is occupied. It uses probabilistic models of exteroceptive sensors to capture the uncertainty in sensor data, and does not rely on geometric models of the environment. It thus does not depend on accuracy (or applicability) of a particular world model. Several variations of the occupancy grid have been explored by researchers. In [3], a novel method for the computation of free space based on dynamic programming on a polar occupancy grid is presented. Estimates of obstacle locations generated by stereo cameras with Kalman Filter are stored in a polar occupancy grid, which provides a constant depth resolution at the expense of some computation time. Danescu et al. [4] present a novel occupancy grid tracking solution based on particles for tracking the dynamic driving environment. The particles have a dual nature: one denotes hypotheses as in the particle filtering algorithm, the other is the building blocks of the modeled world. The tracking algorithm is centered on particles instead of cells. However, this algorithm is subject to a significant speed issue due to the amount of particles. And the usage of age information of the particles can be more flexible.

To overcome the problem of map growth (and the attendant problem of steadily increasing storage and processing), a local occupancy grid (one that moves with the vehicle) is described by Marlow [5].

1.4.3 Path planning

Many algorithms to plan paths over graphs have been developed. A* uses a heuristic to focus the search from a particular start location towards the goal and thus very efficiently produces a path from a single location to the goal [15]. D* is an extension of A* that incrementally repairs solution paths when changes occur in the underlying graph [16]. These incremental algorithms have been used extensively in robotics for navigation in unknown environment. However, almost all of these approaches are limited by the small, discrete set of possible transitions they allow from each node in the graph. Ferguson and Stentz [17] presented Field D*

and Multi-resolution Field D*, two interpolation-based path planning algorithms that address two of the most significant shortcomings of grid-based planning: the quality of paths and the memory and computational requirements. RRT and RRT-Connect provided by LaValle and Kuffner in [18, 19] can perform motion planning in continuous configuration spaces. They can also solve path planning problems in high-dimensional configuration spaces. Besides the graph method algorithm, Borenstein and Koren came up with the potential field methods and the vector field Histogram (VFH) method to planning the path to avoid the obstacle and steer the mobile robot towards the target [20–22]. However, the sensitivity to local minima, that usually arises due to the symmetry of the environment and to concave obstacles, and robot oscillatory behavior when traversing narrow spaces become the drawback of the potential field approach. The VFH method may also lead to the robot being led far away from its target location.

1.5 Contribution

The main contributions of this thesis are describes as follow:

- **Quasi-polar local occupancy grid**

  A quasi-polar local occupancy grid is proposed in this thesis. This method calculates a set of feasible paths before obstacle mapping and path selection, which make the path selection work simplified.

- **Combination of different camera sensor models**

  Different kind of camera models are developed, including monocular vision based on optical flow, pushroom stereo based on single disparity. The method combines the advantage of the separate sensors while overcomes the disadvantage of them.

- **New path selection method**

  Two new path selection methods based on the pre-calculated paths set are given considering the integrated path free probability or longest obstacle free time.

- **Performance verification through simulation**

Two group of simulation are performed to test the performance of the designed system. Results of simulation show that the new quasi-polar local occupancy grid based implementation can be used for obstacle avoidance. The locations of obstacles are accurately estimated. The pre-calculated path based path selectors work well. Different sensor models are compared and discussed.

1.6 Reader's Guide

The remainder of this thesis is organized as follows:

- Chapter 2 gives a mathematical description of the obstacle avoidance problem. It then defines a new coordinate frame, vehicle kinematics and related system models. Quasi-polar local occupancy grid is defined.
- Chapter 3 defines three kind of sensor models. Mathematical expression of obstacles range and direction in camera frame are given.
- Chapter 4 presents the system design which describes the implementation and application of quasi-polar local occupancy grid and the inverse sensor model. Two path selectors are designed for control input.
- Chapter 5 begins with a description of the simulation platform setup designed to test the whole navigation system. The two dimensional simulations results are shown and discussed.
- Chapter 6 introduces the preliminary work towards hardware mainly for vision system. The setup of vision system and implementation of image processing are presented to show the feasibility of sensor models used in simulation.
- Chapter 7 summarizes the results of this research and provides recommendations for future work.

Chapter 2
The Obstacle Avoidance Problem

The following chapter defines the obstacle avoidance problem. The major topics discussed are:

1. Problem Statement: Setup an obstacle avoidance problem in detail. Describe a system which can navigate through an unsurvyed obstacle field. Propose a quasi-polar local occupancy grid to make the path selection work simple while can handle complex objects and noisy environment.

2. System Models: Propose a mathematical model to simulate the vehicle dynamics. Introduce other system modules.

3. Quasi-polar Local Occupancy Grid: Present a mathematical derivation for general occupancy grid. Give the definition of the quasi-polar local occupancy grid and the transformation between polar frame and time-turn rate frame.

2.1 Problem Statement

The situation considered here is a small UAV system that can navigate through an unsurveyed environment (such as a forest or an urban area), Fig. 2.1. An onboard vision system obtains measurements of bearing, optical flow or depth information (for stereo) of obstacles while GPS/INS provides estimates of velocity and heading of the flight vehicle.

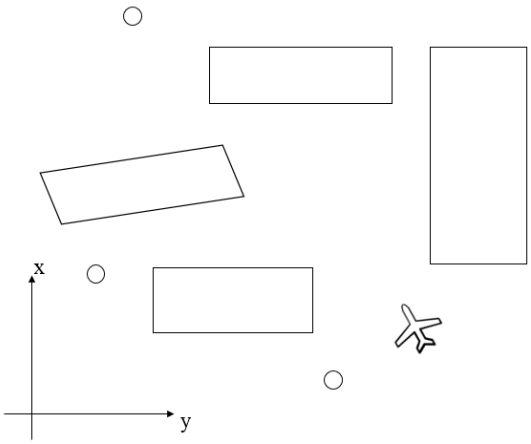


Figure 2.1. Obstacle avoidance scenario. The vehicle must fly through an unsurveyed environment while avoiding obstacles (e.g. small convex obstacles such as tree trunks or large, possibly concave obstacles such as buildings.)

As vehicle state is known through GPS/INS, the main problem to solve is the obstacle avoidance: while this thesis briefly addresses navigation to a particular destination, it is not a major focus. Obstacle avoidance requires measurements of obstacles from sensors and computing the relative obstacle position from the vehicle. A robust estimator is needed to get the relative position of obstacles from the noisy sensor measurements. The estimated results can then be used to generate a local map of obstacles. A path will be selected from the map based on the occupied probability. During sensor dropouts or in the areas which are currently not in the sensors field of view but previously surveyed, the vehicle can use the obstacles information stored in the map to make decision about which path to follow next.

Given the vehicle states ($\mathbf{x}$) and measurements of bearing and optical flow from two cameras, a quasi-polar local map of the environment ($\mathbf{m}$) can be computed. The vehicle state is

$$\mathbf{x} = \begin{bmatrix} x & y & \psi & u & v \end{bmatrix}^T \tag{2.1}$$

where x and y are the location of the vehicle in global frame, which is an inertial frame, ψ is the vehicle heading with respect to an inertial frame, and u, v are the velocity of the vehicle in body frame.

The map is represented as a likelihood given a state history $\mathbf{x}_{1,\cdots,t}$ and measure-

ment history $\mathbf{z}_{1,\cdots,t}$:

$$p(\mathbf{m}|\mathbf{x}_{1,\cdots,t}, \mathbf{z}_{1,\cdots,t}) \tag{2.2}$$

where the measurement model

$$\mathbf{z}_i = g(\mathbf{x}_i, \mathbf{m}) \tag{2.3}$$

defines the probability of obtaining a particular measurement given a vehicle state $\mathbf{x}_i$ and map $\mathbf{m}$.

Nonlinearities are introduced in two ways: first, through the vehicle's rotational degree of freedom; second, through the projection of the three-dimension world onto the two-dimensional image plane.

Note that the fields of view of the two cameras do not necessarily overlap. An overlapping region would provide a region of stereo data; one of the purposes of this research is to determine how much overlap is desirable for good navigation performance.

2.2 System Models

2.2.1 Coordinate frames

Referring to Fig. 2.2, the vehicle is located at position x, y in an inertial North-East-Down (NED) coordinate frame G. The body frame B stays fixed with the vehicle with its origin at the center of gravity of the vehicle. The orientation is defined by heading ψ with respect to the inertial frame. Sensors are fixed to the vehicle with known positions and angular offset relative to body frame. Velocity and angular rates are obtained in body frame B. Bearings to the obstacles and optical flow are obtained in camera frame C.

The quasi-polar occupancy grid coordinate frame P (defined by unit vectors $\dot{\psi}$, Δt, and shown in Fig. 2.3) translates with the vehicle sharing a common origin with frame B. Clearly, P is not a Cartesian coordinate system: $\dot{\psi}$ denotes the turn rate of a constant-speed path from the origin, and Δt is the time needed flying from origin along a specific path to a given point. The occupancy grid used to map obstacles will be defined by this quasi-polar frame.

For convenience, define a Cartesian coordinate occupancy grid O by unit vector $\mathbf{x}_o$ and $\mathbf{y}_o$ translates with the vehicle sharing a common origin with frame B, but the orientation of the axes remains fixed relative to the inertial frame.

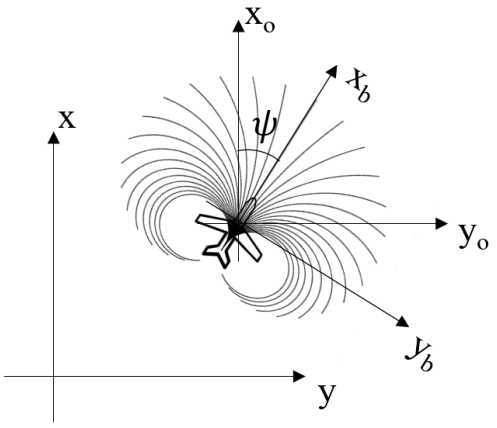


Figure 2.2. Coordinate frames

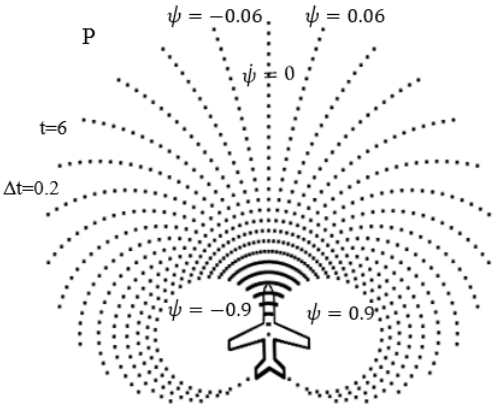


Figure 2.3. Quasi-polar coordinate

2.2.2 Kinematic model

For simplicity, the derivation described here is done for planar motion. Vehicle position x and y is expressed in the inertial frame, heading is defined by ψ. Velocities u and v are expressed in the body frame B. Vehicle kinematics can thus be defined as

$$\dot{x} = u\cos\psi - v\sin\psi \tag{2.4}$$

$$\dot{y} = u\sin\psi + v\cos\psi \tag{2.5}$$

$$\dot{\psi} = \omega + \mathcal{N}(0, \sigma_\omega^2) \tag{2.6}$$

$$\dot{u} = a_x + \mathcal{N}(0, \sigma_{a_x}^2) \tag{2.7}$$

$$\dot{v} = a_y + \mathcal{N}(0, \sigma_{a_y}^2) \tag{2.8}$$

Here a_x, a_y, and ω (acceleration in $\hat{x}_b$ and $\hat{y}_b$ and turn rate, respectively) are control inputs corrupted by process noise. $\mathcal{N}(0, \sigma^2)$ denotes a Gaussian random variable with mean 0 and standard deviation σ. Motion typical of a fixed-wing aircraft (i.e. coordinated flight) will be considered, hence $a_y = 0$.

2.2.3 Inertial/GPS measurement model

The GPS/INS measurement unit includes a GPS receiver, accelerometers, rate gyros, and a magnetic compass. Vehicle position, orientation, and velocity (corrupted by zero mean Gaussian noise) with respect to the Earth are provided by the GPS/INS unit. Note that even with the availability of GPS, exteroceptive sensors such as cameras or LIDAR are required to detect obstacles.

2.3 Quasi-polar Local Occupancy Grid

Given measurements of obstacle relative position from the vision system, a means of mapping obstacles is very useful: it provides a "retained memory" of obstacle positions in the event of sensor drop-outs or occlusions, and it provides a means of fusing multiple measurements of the same obstacle to improve localization accuracy.

The occupancy grid computes the likelihood that discrete environment cells are occupied by obstacles (Fig. 2.4). Since each cell in the environment is considered independently, computational cost is fairly low and complex obstacle shapes can be modeled.

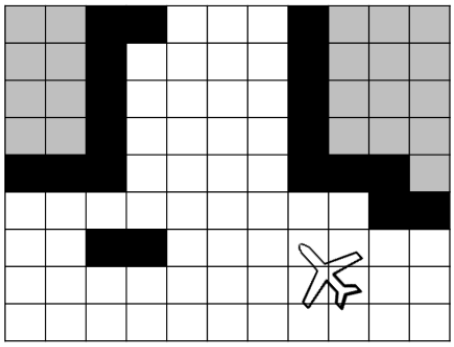


Figure 2.4. Schematic of occupancy grid. Occupied cells are shown in black, free cells in white, unknown cells in grey.

2.3.1 Derivation for general occupancy grid

Occupancy grids have been well documented (see for example Thrun [23]). For completeness we summarize the basic derivation here. The goal is to estimate a map of the environment, $p(\mathbf{m}|\mathbf{z}_{1,\cdots,t}, \mathbf{x}_{1,\cdots,t})$, where $\mathbf{z}_{1,\cdots,t}$ is the set of all measurements and $\mathbf{x}_{1,\cdots,t}$ is the set of all vehicle states up to time t. By discretizing the map into a finite number of cells, occupancy grids recursively computes the estimate of $p(\mathbf{m})$, which is a numerical implementation of a Bayes filter.

$$\mathbf{m} = \{m_{ij}\}$$

where m_{ij} is a cell in the map.

For each grid cell, the probability that a cell is occupied, $p(m_{ij}) = 1$, or free, $p(m_{ij}) = 0$, is computed. This makes the problem binary over all possible maps. Also, assuming that the probability of a particular cell's occupancy is independent of all other cells makes a conservative approximation, since the information about the world like the continuity between walls and buildings has been ignored. The estimation problem becomes

$$p(\mathbf{m}|z_{1,\cdots,t}, x_{1,\cdots,t}) = \prod_{i,j} p(m_{i,j}|z_{1,\cdots,t}, x_{1,\cdots,t}) \tag{2.9}$$

so the likelihood of each cell being occupied or not in occupancy grids can be computed independently.

For one cell, write the binary Bayes filter in discrete map of a static environment.

$$p(m_{i,j}|z_{1,\cdots,t}, x_{1,\cdots,t}) = \frac{p(z_t|m_{i,j}, x_t)p(m_{i,j}|z_{1,\cdots,t-1}, x_{1,\cdots,t-1})}{p(z_t|z_{1,\cdots,t-1}, x_{1,\cdots,t})} \tag{2.10}$$

Define odds and log-odds as

$$\begin{aligned} o(m_{i,j}|z_{1,\cdots,t}, x_{1,\cdots,t}) &= \tfrac{p(m_{i,j}|z_{1,\cdots,t}, x_{1,\cdots,t})}{1-p(m_{i,j}|z_{1,\cdots,t}, x_{1,\cdots,t})} \\ l_{t,ij} &= \log o(m_{i,j}|z_{1,\cdots,t}, x_{1,\cdots,t}) \end{aligned} \tag{2.11}$$

Using independence assumption and converting the map probability into log-

odds form, Eq. 2.10 becomes

$$l_{t,ij} = \ln \frac{p(m_{i,j}|z_t, x_t)}{1 - p(m_{i,j}|z_t, x_t)} + l_{t-1,ij} \tag{2.12}$$

In Eq. 2.12, the first term on the right-hand side is the inverse sensor model, which will be discussed in detail later. For multiple measurements, the inverse sensor model for each measurement will be added together:

$$l_{t,ij} = \sum_{n}^{N} \ln \frac{p(m_{i,j}|z_{t,n}, x_t)}{1 - p(m_{i,j}|z_{t,n}, x_t)} + l_{t-1,ij} \tag{2.13}$$

The second term on the right hand side of Eq. 2.13 is the accumulated log-odds of one occupancy grid cell over all previous time steps.

By solving Eq. 2.11, the probability of occupancy can be recovered from the log-odds:

$$p(m_{i,j}|z_{1,\cdots,t}, x_{1,\cdots,t}) = \frac{e^{l_{t,ij}}}{1 + e^{l_{t,ij}}} \tag{2.14}$$

The estimation problem now becomes a recursive equation to compute the occupancy of each grid cell, which can be implemented with great efficiency.

2.3.2 Quasi-polar local occupancy grid

Typical occupancy grid remains globally fixed as the robot moves through the grid. However, the map will quickly grow large and become computationally intractable as the vehicle explores the environment. To avoid obstacles near the vehicle, a local occupancy grid can be used instead of global occupancy grid, reducing computational requirements. Grid size can be selected based on sensor and computation ability. Marlow described a Cartesian local occupancy gird, whose origin is fixed to the vehicle, and the orientation remains fixed to an inertial frame [5]. Motion updates are done using standard image processing tools such as translation and convolution. This was also extended to a dynamically sized map, where the size of each cell is dependent on vehicle speed [24].

While Marlow's local occupancy grid reduced the computational cost, a trajectory planning step is still required. Here we describe a quasi-polar occupancy grid that operates directly on a set of dynamically feasible paths, so that trajectory planning consists simply of selecting the path that is free for a sufficient portion of

its length.

2.3.2.1 Definition and coordinate transformation

As defined in Sec. 2.2, the quasi-polar occupancy grid uses pre-calculated possible paths of the vehicle over a time horizon. For a constant speed vehicle, paths are differentiated by turn rate. The shape of the quasi-polar occupancy grid depends on vehicle speed, its maximum turn rate $|\psi|_{max}$ and time t_{max}.

Fig. 2.5 shows three sets of flight paths. Clearly one should choose a maximum turn rate $|\psi|_{max}$ that is within the vehicle's kinematic limits and a time t_{max} that results in a path length that is within vehicle sensor range but long enough that the vehicle can avoid obstacles.

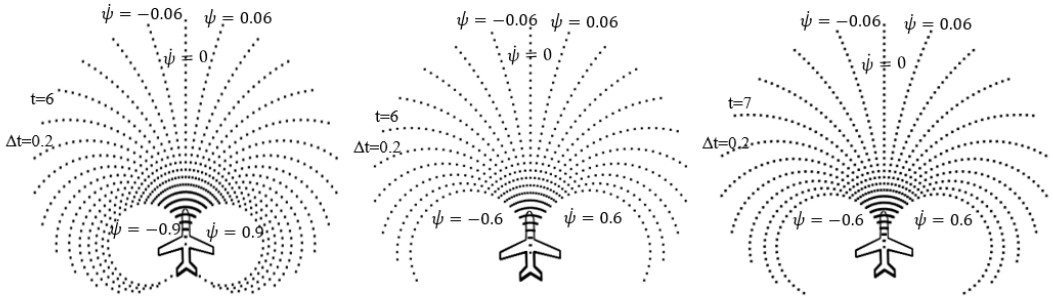


Figure 2.5. Comparison of quasi-polar occupancy grid. Left: Grid shape with $|\psi|_{max} = 0.9$rad/sec, $t_{max} = 6$sec; Middle: Grid shape with $|\psi|_{max} = 0.6$rad/sec, $t_{max} = 6$sec; Right: Grid shape with $|\psi|_{max} = 0.6$rad/sec, $t_{max} = 7$sec

The path time can be chosen as

$$t_{max} = \frac{s}{u} \tag{2.15}$$

where u is vehicle speed and s is a desired look-ahead distance.

Given the set of flight paths, a means of converting the quasi-polar coordinate to the range/bearing information in body frame is required.

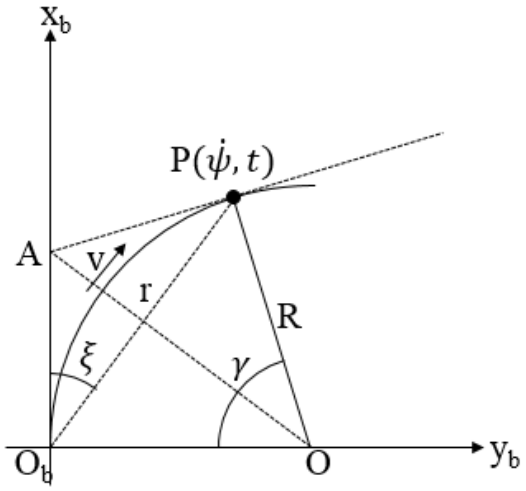


Figure 2.6. Pre-calculated path

As Fig. 2.6 shows, the arc $\widehat{O_bP}$ is a pre-calculated path with turn rate $\dot{\psi}$. The vehicle needs time t to move from O_b to point P with turn rate $\dot{\psi}$ and constant speed v along this path. The arc length of $\widehat{O_bP}$:

$$\widehat{O_bP} = vt = R\gamma \tag{2.16}$$

It's easy to see the geometric relation $\gamma = 2\xi$. In addition, using sine law,

$$2R = \frac{|O_bP|}{\sin\xi} \tag{2.17}$$

And from the physical definition of $\dot{\psi}$ and t,

$$\dot{\psi} = \frac{\gamma}{t} = \frac{v}{R} \tag{2.18}$$

$$t = \frac{R\gamma}{v} \tag{2.19}$$

Put R, γ into t and make simplification,

$$|O_bP| = \frac{2v\sin\xi}{\dot{\psi}} \tag{2.20}$$

$$t = \frac{\xi|O_bP|}{\sin\xi v} = \frac{2\xi}{\dot{\psi}} \tag{2.21}$$

Finally, convert the quasi-polar coordinate into body polar coordinate,

$$r = |O_bP| = \frac{2v\sin\xi}{\dot{\psi}} \tag{2.22}$$

$$\xi = \frac{t\dot{\psi}}{2} \tag{2.23}$$

Hence, if a point is known in quasi-polar occupancy grids with the coordinate $(\dot{\psi}, t)$, the coordinate of the point in body polar frame (r, ξ) can be obtained.

2.4 Summary

A description of the obstacle avoidance problem is given in Section 2.1. The problem is avoiding obstacles using noisy sensor measurement. An occupancy grid method is chosen to handle the uncertainty of obstacle estimate and the complex environment.

A quasi-polar frame and mathematical models of the vehicle kinematics is developed in Section 2.2. The quasi-polar coordinate is defined as a two dimensional turn rate-time frame based on a set of pre-calculated paths. Vehicle motion is modeled as a second order function with control inputs of acceleration and heading rate expressed in the body frame.

A mathematical derivation of the general occupancy grid is presented in Section 2.3. Then the formal definition of the quasi-polar local occupancy grid is given. The transformation between the new frame and Cartesian or polar frame is derived.

Chapter 3
Sensor Models

The following chapter details the vision model used to get the measurement introduced in Chapter 2. Three kinds of sensor models are given in this chapter.

1. Monocular Vision: Monocular vision model is given first as foundation. Mathematical expression of bearing and bearing rate are presented. The corresponding range model for detected obstacles and their uncertainties are also discussed.

2. General Stereo: A general stereo vision setup motivated by a bird's visual geometry is created. Two cameras share a region of overlapping field of view as well as maintaining a non-overlapping field of view. Mathematical derivation for bearing, bearing rate and range model are presented. Simple stereo as a specific case of general stereo is then discussed.

3. Combined Sensors: A combination that uses a "pushbroom stereo" combined with monocular vision is created to take advantages of monocular and stereo vision while overcome the disadvantages of them. Camera data flow then is listed for the whole sensor model.

The inverse sensor model, used to map sensor data onto the occupancy grid will be described in Chapter 4.

3.1 Monocular Vision

3.1.1 Vision model

For the monocular vision, one single camera is fixed to the vehicle pointed along the vehicle's body $\hat{x}_b$ axis. Bearing and bearing rate to objects within the field of view are the output measurements, see Fig. 3.1.

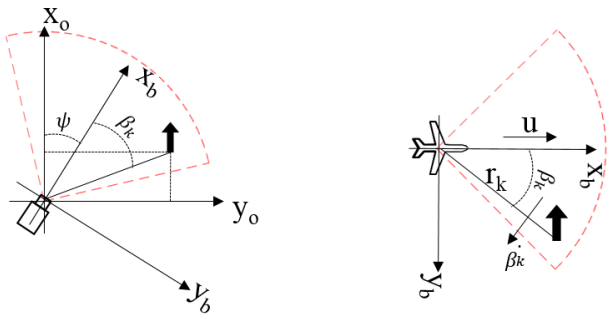


Figure 3.1. Monocular vision model

The bearing to the k^{th} feature located at $(x_{k/o},\ y_{k/o})$ in coordinate frame O is

$$\beta_k = \arctan\frac{y_{k/o}}{x_{k/o}} - \psi \tag{3.1}$$

The bearing rate is computed by taking the time derivative of Eq. 3.1:

$$\dot{\beta}_k = \frac{u \sin\beta_k}{r_k} - \frac{v\cos\beta_k}{r_k} - \dot{\psi} \tag{3.2}$$

where $r_k = \sqrt{x_{k/o}^2 + y_{k/o}^2}$ is the distance between the camera and the object.

The field of view of the camera is divided into a number of sectors to limit the number of measurements [5]. The resolution $\Delta\beta$ is the angular width of each sector, in which the largest value of optical flow is returned as the measured value (it is assumed to come from the nearest object in that sector). It is assumed that optical flow measurements is corrupted by Gaussian noise with constant standard deviation across the field of view. Also, there is no correlation between the inputs.

The measurement vector $\mathbf{z}$ is

$$\mathbf{z} = \begin{bmatrix} u & v & \dot{\psi} & \beta^T & \dot{\beta}^T \end{bmatrix}^T + \mathcal{N}(0, \Sigma_Z) \tag{3.3}$$

$$\Sigma_z = \text{diag}\begin{bmatrix} \sigma_u^2 & \sigma_v^2 & \sigma_{\dot{\psi}}^2 & \Sigma_\beta^2 & \Sigma_{\dot{\beta}}^2 \end{bmatrix} \tag{3.4}$$

where β and $\dot{\beta}$ are vectors containing the measurements of the maximum optical flow and associated bearing in each of the N sectors, respectively.

$$\beta = \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_N \end{bmatrix}^T \tag{3.5}$$

$$\dot{\beta} = \begin{bmatrix} \dot{\beta}_1 & \dot{\beta}_2 & \cdots & \dot{\beta}_N \end{bmatrix}^T \tag{3.6}$$

$$\Sigma_\beta = \sigma_\beta^2 \mathbf{I}_N \tag{3.7}$$

$$\Sigma_{\dot{\beta}} = \sigma_{\dot{\beta}}^2 \mathbf{I}_N \tag{3.8}$$

where $\mathbf{I}_N$ represents an identity matrix of size N.

3.1.2 Range model

Range r_k^* can be computed from Eq. 3.2:

$$r_k^* = \frac{1}{\dot{\beta}_k + \dot{\psi}}(u \sin \beta_k - v \cos \beta_k) \tag{3.9}$$

Since there is noise in the bearing and optical flow measurements and the estimate of velocity and heading also have uncertainty, the uncertainty in the estimate of $\sigma_{r_k^*}$ is needed. The Jacobian of Eq. 3.9 is computed for the uncertainty of range estimate:

$$\nabla_{r_k^*} = \begin{bmatrix} \frac{\partial r_k^*}{\partial u} & \frac{\partial r_k^*}{\partial v} & \frac{\partial r_k^*}{\partial \dot{\psi}} & \frac{\partial r_k^*}{\partial \beta_k} & \frac{\partial r_k^*}{\partial \dot{\beta}_k} \end{bmatrix} \tag{3.10}$$

Put $\nabla_{r_k^*}$ into $\sigma_{r_k^*}^2 = \nabla_{r_k^*}^T \Sigma_z \nabla_{r_k^*}$ and simplify:

$$\begin{aligned} \sigma_{r_k^*}^2 = {} & (\dot{\beta}_k + \dot{\psi})^{-2}[\sigma_u^2 \sin^2 \beta_k + \sigma_v^2 \cos^2 \beta_k + \sigma_\beta^2 (u \cos \beta_k + v \sin \beta_k)^2] \\ & + (\dot{\beta}_k + \dot{\psi})^{-4}[(\sigma_{\dot{\beta}}^2 + \sigma_{\dot{\psi}}^2)(u \sin \beta_k - v \cos \beta_k)^2] \end{aligned} \tag{3.11}$$

It's been shown that in the region where $|\beta|$ is a small value, the value of $\dot{\beta}$ is on the order of $\sigma_{\dot{\beta}}$, so that sensor measurements produce poor range estimate [5].

When an estimate is known to be inaccurate in advance, the uncertainty of the estimate is more important than the estimate itself. The linearization method, that is uses the measurements directly to compute r^* and linearizes f to compute σ_{r^*}, provides acceptable results for the areas of interest as unscented transform or particle transform, and is not as computationally intensive. Therefore, Eq. 3.9 and Eq. 3.11 are used to compute r^* and σ_{r^*} [5].

3.2 General Stereo Model

The general stereo vision consists of two cameras mounted as in Fig. 3.2. Note the angle θ defines the angle of each camera's optical axis with respect to the vehicle's body $\hat{x}_b$ axis. The outputs of the stereo vision system are also measurements of bearing and bearing rates to objects within the field of view of each separate single camera.

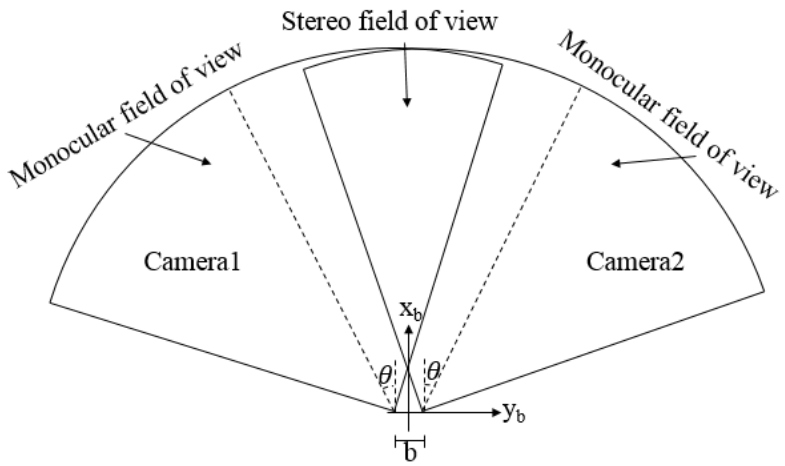


Figure 3.2. General stereo

Consider the right camera. If one feature in inertial frame is located at $\begin{bmatrix} x_k & y_k \end{bmatrix}$, the coordinate in the local vehicle carried frame is $\begin{bmatrix} x_{k/o} & y_{k/o} \end{bmatrix} = \begin{bmatrix} x_k - x & y_k - y \end{bmatrix}$, where x, y is the vehicle's location. First transform it into the body frame,

$$x_{b1} = x_{k/o}\cos\psi + y_{k/o}\sin\psi \tag{3.12}$$

$$y_{b1} = -x_{k/o}\sin\psi + y_{k/o}\cos\psi \tag{3.13}$$

Then translate it for half of the baseline into the baseline body frame.

$$x_{b2} = x_{k/o}\cos\psi + y_{k/o}\sin\psi \tag{3.14}$$

$$y_{b2} = -x_{k/o}\sin\psi + y_{k/o}\cos\psi - \frac{b}{2} \tag{3.15}$$

where b is length of baseline. Finally, transform it into camera frame

$$x_c = x_{b2}\cos\theta + y_{b2}\sin\theta \tag{3.16}$$

$$y_c = -x_{b2}\sin\theta + y_{b2}\cos\theta \tag{3.17}$$

where θ is the angle between the optical axis of the camera and the body axis. The bearing is now

$$\beta_k = \arctan\frac{y_c}{x_c} = \frac{-x_{b2}\sin\theta + y_{b2}\cos\theta}{x_{b2}\cos\theta + y_{b2}\sin\theta} \tag{3.18}$$

The bearing rate is computed by taking the time derivative of Eq. 3.18:

$$\dot{\beta}_k = \frac{\sin\beta_k}{r_{kc}}(u\cos\theta + v\sin\theta) - \frac{\cos\beta_k}{r_{kc}}(v\cos\theta - u\sin\theta) - \dot{\psi} \tag{3.19}$$

where $r_{kc} = \sqrt{x_c^2 + y_c^2}$ is the distance between the camera and the object.

As the monocular vision equations described earlier, Eq. 3.19 is solved for range r (since the baseline is small, $r_k \approx r_{kc}$):

$$r_k^* = \frac{(u\cos\theta + v\sin\theta)\sin\beta_k - (v\cos\theta - u\sin\theta)\cos\beta_k}{\dot{\beta}_k + \dot{\psi}} \tag{3.20}$$

The Jacobian of Eq. 3.20 is computed for the uncertainty in range estimate:

$$\nabla_{r_k^*} = \left[\begin{array}{ccccc} \frac{\partial r_k^*}{\partial u} & \frac{\partial r_k^*}{\partial v} & \frac{\partial r_k^*}{\partial \dot{\psi}} & \frac{\partial r_k^*}{\partial \beta_k} & \frac{\partial r_k^*}{\partial \dot{\beta}_k} \end{array}\right]$$

So,

$$\sigma_{r_k^*}^2 = \nabla_{r_k^*}^T \Sigma_z \nabla_{r_k^*}$$

Comparing Eqs. 3.1, 3.2 and 3.9 with Eqs. 3.18, 3.19 and 3.20, it can be seen the only difference for optical flow and range measurements between monocular and one separate camera in general stereo is that the latter projects the velocities onto camera optical flow axis, which leads to a different effective velocity.

3.2.1 Simple stereo

When $\theta = 0°$, it becomes the simple stereo case, see Fig. 3.3. In simple stereo case, if the corresponding bearing pair is known for a specific feature, its 3D position can be reconstructed.

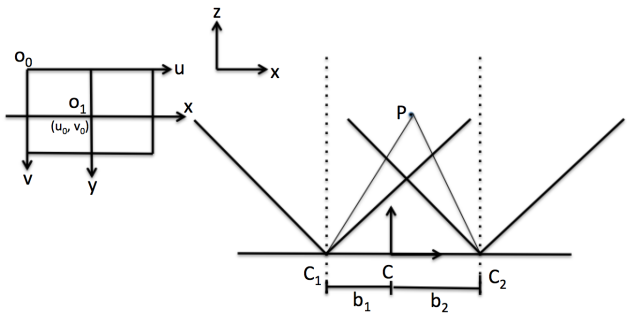


Figure 3.3. Simple stereo

Lemma 1 (Simple stereo reconstruction)**.** *As the setting in Fig. 3.3, if the bearing angles of one point in two cameras are known, β_1, β_2, which obtained from left eye and right eye. The coordinate of the point in the world coordinate can be reconstructed:*

$$x_w = \frac{b_1 \tan\beta_2 + b_1 \tan\beta_1}{\tan\beta_1 - \tan\beta_2}, z_w = \frac{b_1 + b_2}{\tan\beta_1 - \tan\beta_2} \tag{3.21}$$

Proof. As shown in Fig. 3.3, the relationship between the pixel coordinate of one point and the camera coordinate is

$$u = \frac{x}{dx} + u_0, v = \frac{y}{dy} + v_0 \tag{3.22}$$

where the dx and dy are the pixel grid space. Let $P(x_w, y_w, z_w)$ in the world frame. So in the left eye C_1, it will be $(x_w + b_1, y_w, z_w)$. And in the right eye C_2, it will be $(x_w - b_2, y_w, z_w)$. According to the projection geometry theorem,

$$\frac{z_w}{f} = \frac{x_w + b_1}{(u_1 - u_0)dx}, \frac{z_w}{f} = \frac{x_w - b_2}{(u_2 - u_0)dx} \tag{3.23}$$

where f is the focal length of the camera. $b_1 + b_2 = b$ is the baseline of the stereo.

Therefore,

$$x_w = \frac{b_1 x_2 + b_2 x_1}{x_1 - x_2} = \frac{b_1 \frac{x_2}{f} + b_2 \frac{x_1}{f}}{\frac{x_1}{f} - \frac{x_2}{f}} = \frac{b_1 \tan\beta_2 + b_1 \tan\beta_1}{\tan\beta_1 - \tan\beta_2} \tag{3.24}$$

$$z_w = \frac{(b_1 + b_2) f}{x_1 - x_2} = \frac{b_1 + b_2}{\frac{x_1}{f} - \frac{x_2}{f}} = \frac{b_1 + b_2}{\tan\beta_1 - \tan\beta_2} \tag{3.25}$$

□

Once the 3D location of a point is reconstructed, the bearing and range of the point relative to the vehicle can be known.

3.2.2 Single disparity method

A traditional stereo system uses block-matching method to produce depth estimates by finding pixel-block matches between two images. For example, if given a pixel block in the left eye's image, the standard stereo system will find the best match by searching through the epipolar line of the right eye's image, see Fig. 3.4. The disparity, which is the position of the match block relative to its coordinate in the left eye's image, can be used to compute the 3D position of the object in that pixel block, [25]. For the simple stereo case, Fig. 3.5

$$\frac{T + x_l - x_r}{Z - f} = \frac{T}{Z} \tag{3.26}$$

Solving Eq. 3.26 for Z we obtain

$$Z = f\frac{T}{D} \tag{3.27}$$

where Z is the depth of one point, T is the baseline of the stereo system, f is focal length, $D = x_r - x_l$, the disparity.

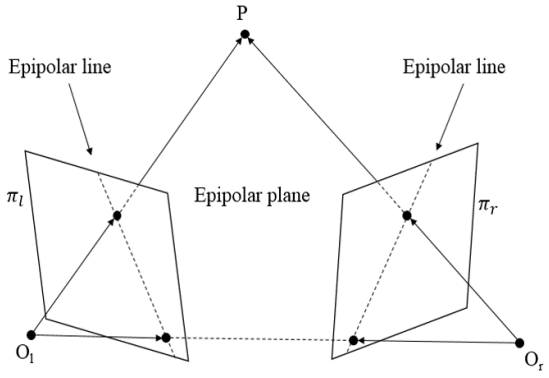


Figure 3.4. The epipolar geometry

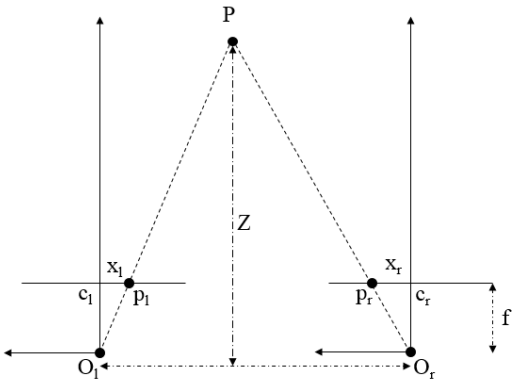


Figure 3.5. Depth estimate for simple stereo

Since for each block in the left image, the standard stereo needs to search along the whole epipolar line, or search through depth, Eq. 3.27, the real time system suffers from a speed issue. Consider the computational efficiency for real time hardware, a pushbroom stereo using single disparity searching algorithm provided by Barry is used [10]. Limiting the search through distance to a single value, d meters away, which also means a single disparity, the processing can be substantially speed up. Although the obstacles at the distance other than d being neglected seems limiting, the missing depth information can be quickly recovered by previous single disparity results [10] since the cameras are on a moving platform (aircraft or hexcopter). This is shown schematically in Fig. 3.6.

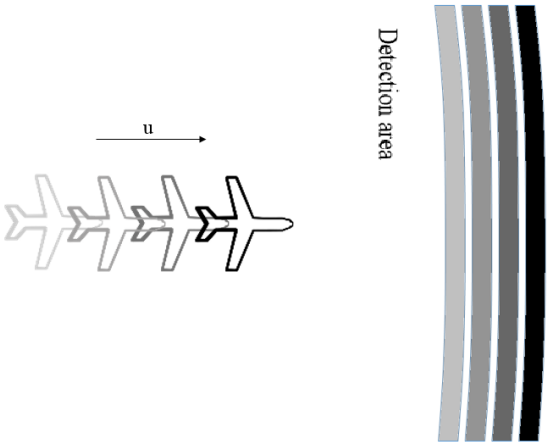


Figure 3.6. Pushbroom stereo using single disparity

Simply to say, at each sensor update, only the obstacles at the depth d from the camera can be seen. d should be set to the maximum resolvable distance for a specific stereo baseline.

3.3 Combined Sensors

Stereo vision can provide a more accurate depth estimate comparing to monocular vision since the optical flow measurements are noisier and knowledge of vehicle velocity is required to compute range using optical flow. However, simple stereo limits the obstacles detectable range when using single disparity algorithm. A natural idea emerges that combining the two kind of sensors to take advantage of each of them.

Again, consider the simple stereo case, setup the sensors as Fig. 3.7. The image pair can be regarded as two sensors. In the overlap region, the single disparity algorithm can be used to compute range using stereo vision. As state above, it can only find features at a fixed depth away from the camera, but computational cost is lower than dense stereo. The system can be designed so that the estimate is good at the specified depth. In region A of left eye's image, and region B of right eye's image, compute bearing and optical flow (bearing rate) for each feature as with a monocular vision system.

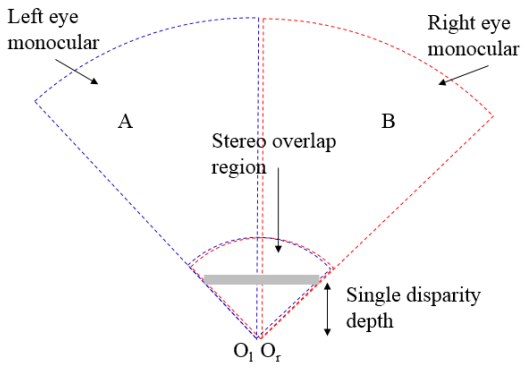


Figure 3.7. Combined sensor

In general stereo model of Section 3.2, assign a nonzero value to the angle θ. Similar to a bird's vision, the optical axes of the two cameras do not point forward. Instead, there is angle between the camera's optical axis and the direction

of airplane's body $\hat{x}_b$ axis. In this case, the overlap region of stereo vision is smaller than that in the simple stereo case. By choosing θ properly, the overall field of view of the two cameras is broader, and there is a region of stereo directly to the front of the vehicle (where there is poor range estimate information from optical flow).

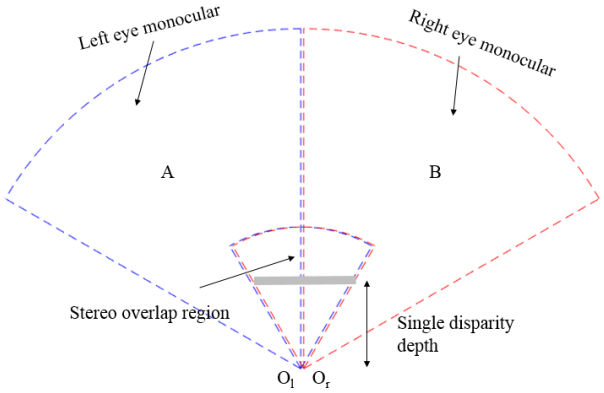


Figure 3.8. "Bird eyes" combined sensor

For example, suppose both two cameras have 90° field of view, and $\theta = 15°$, the bird eye model will be given like Fig. 3.8. The overlap region of stereo vision now is 60°. Since the single disparity algorithm is used, only the obstacle points on the fixed depth line can be seen at each sensor update. Region A belongs to the left eye, which will be used to find obstacle feature points and calculate optical flow like monocular vision. The same operation will be done for the right eye's region B.

3.3.1 Summary of data flow

1. Image pair obtained from two cameras, see Fig. 3.9.
2. In region S_1 and S_2, which is the stereo overlap region, using single disparity algorithm to find obstacle feature points and compute the range estimate.
3. In region A and B, detect obstacle feature points and calculate optical flow like monocular vision based on Harris corner.

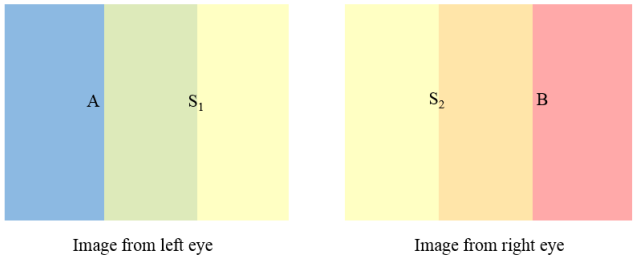


Figure 3.9. Cameras data flow

3.4 Summary

This chapter has presented three kind sensor models using vision method. Monocular vision is given as a foundation. Mathematical expression of the measurement, bearing and bearing rate are derivated. Motivated by a bird's visual geometry, a general stereo vision model is presented. As a specific case, simple stereo is then discussed in detail, which can be used to do 3D reconstruction. A pushbroom method using single disparity is introduced as a efficient way to process stereo data. Based on the idea that take advantage of different sensors, a combined sensor is created which can handle the separate region of images by using monocular method and the overlap region using pushbroom stereo algorithm.

Chapter 4
System Design

The following chapter details the techniques used to estimate the obstacle location on the quasi-polar local map once detected by vision system and the path selection method for vehicle control. The design is divided into four sections:

1. Obstacle Mapping: Map the detected obstacle by vision system onto the quasi-polar local map.

2. Inverse Sensor Model: The inverse sensor model is needed to update the occupancy grid with obtained measurements, as mentioned in Eq. 2.12. It includes two parts, range and direction. Functions are created to approximate the probability distribution around the estimated obstacle location considering time and computational efficiency.

3. Motion Update: The local map with occupied probability needs to be updated as the vehicle moving. Considering the construction and cell shape of the map, a numerical method is proposed for motion update.

4. Path Selection: Two kind of path selectors are presented based on integrated path free probability/longest obstacle free time.

4.1 Obstacle Mapping

From the several sensor models in Chapter 3, all the obstacle feature points seen by the cameras can be estimated in body frame with range r and bearing β (Here β and ξ describe the same angle but in different frames: β in camera frame, ξ in

polar coordinate frame). As discussed in Chapter 2, the polar representation of the grid cell locations for the quasi-polar local occupancy grid is Eq. 2.22:

$$r = |O_b P| = \frac{2v \sin \xi}{\dot{\psi}} \tag{4.1}$$

$$\xi = \frac{t\dot{\psi}}{2} \tag{4.2}$$

where $(\dot{\psi}, t)$ is the coordinates in quasi-polar frame, v is vehicle speed. The corresponding coordinate in quasi-polar local occupancy grid frame can be obtained once the coordinate in polar frame is known.

$$\dot{\psi} = \frac{2v}{r} \sin \xi \tag{4.3}$$

$$t = \frac{\xi r}{\sin \xi v} = \frac{2\xi}{\dot{\psi}} \tag{4.4}$$

So once the measurements of the obstacle feature points obtained from the cameras, their position relative to the vehicle and the location within the quasi-polar local occupancy grid can be obtained. The inverse sensor model, which will be discussed later, is used to assign the occupied probability of one cell with respect to one measurement.

4.2 Inverse Sensor Model

The inverse sensor model computes the probability of a cell being occupied for a single measurement. It is used to updated the occupancy grid when the measurements are obtained as shown in Chapter 2 Eq.2.12. From the three kinds of sensors given in Chapter 3, an estimated range to the obstacle is computed, thus the obstacle location in polar coordinates is known. Two pieces are considered in the inverse sensor model, a function of range to the obstacle, an angular position function that reflecting bearing of the measurements.

4.2.1 Range

In Chapter 3, the range r^* and uncertainty in the estimate σ_{r^*} are computed from the measurements of bearing β, optical flow $\dot{\beta}$, and depth information d from stereo

vision. In this thesis, the noise is assumed to be Gaussian.

Common sense tells us that the space between the camera and the detected obstacle is unoccupied and the space behind the obstacle remains unknown.

Here, assume the obstacle exists most likely two standard deviation in front of and three standard deviation behind the estimated range r^*. So the likelihood should be:

$$p = \begin{cases} > 0.5 & r^* - 2\sigma_{r^*} \leq r \leq r^* + 3\sigma_{r^*} \\ = 0.5 & r > r^* + 3\sigma_{r^*} \\ < 0.5 & 0 < r < r^* - 2\sigma_{r^*} \end{cases}$$

The superposition of Gaussian and Sigmoid curve can model this behavior.

$$p(r) = a_1(1 + \exp(\frac{2\pi(r - r^* + 2\sigma_{r^*})}{\sqrt{3}\sigma_{r^*}}))^{-1} + a_2 \exp(-\frac{(r - r^*)^2}{2\sigma_{r^*}^2}) \tag{4.5}$$

4.2.1.1 Far away obstacles

In monocular sensor model, when obstacles are far away or not present, $\dot{\beta} + \dot{\psi} \approx 0$, Eq. 3.2 will yield $r^* \approx \infty$, the uncertainty σ_{r*} will also be ∞, Eq. 3.11. The occupied region will extend from the estimated location all the way to the camera when the uncertainty is large, which doesn't make sense since there is no region to be marked as free space between the estimated location and the camera.

Here, a limit r_{max} is set for the r^* , when $r^* > r_{max}$, set $r^* = r_{max}$ and a small value to the uncertainty σ_{r^*}.

4.2.1.2 Obstacles in the direction of motion

For the obstacles located in the direction of motion, they generate almost no optical flow separate from the rotation of vehicle. Again, $\dot{\beta} + \dot{\psi} \approx 0$, Eq. 3.2. The limit can also be set for r^* here.

For these kind of obstacles, set the uncertainty $\sigma_r = \frac{r_{max}}{3}$, since we know that they are in the direction of motion and don't want the uncertainty to be too large. So give the uncertainty a number to make sure the uncertainty ellipse covers almost all the occupied probability.

4.2.1.3 Log-odds computation concern

The inverse sensor model must be in log odds form to be added to the previous log-odds belief, Eq. 2.13. However, performing the log-odds conversion over every cells for every measurement wastes too much computation cost. So it's necessary to directly make an approximation for the log-odds form of the inverse sensor model, Eq. 4.5.

In log-odds form, the range part of the inverse sensor model can be approximated by:

$$f_n(r) = -c_1(1 + \exp(\frac{2\pi(r - r_n^* + 2\sigma_{r_n^*})}{\sqrt{3}\sigma_{r_n^*}}))^{-1} + \frac{c_2}{\sigma_{r_n^*}\sqrt{2\pi}} \exp(-\frac{(r - r_n^*)^2}{2\sigma_{r_n^*}^2}) \quad (4.6)$$

Here, r is the range to a cell, r_n^* is the range estimate for the measurement in the nth obstacle feature point. Two parameters c_1 and c_2 can be used to scale contribution of the two functions for the probability.

4.2.2 Direction

Since there is noise in the bearing measurement, the direction part is necessary for the inverse sensor model.

A sigmoid weight function is used here to compute the scaling factor as a function of the difference between the direction of the grid cell and the measurement bearing.

$$g_n(\xi) = (1 + \exp \frac{c_3(|\xi - \beta_n| - 0.5 \triangle \beta - 1.25\sigma_\psi)}{\sigma_\psi})^{-1} \quad (4.7)$$

where ξ is the bearing to a cell, $\triangle \beta$ is the angular width of the sector, σ_ψ is the uncertainty of the current vehicle heading estimate, c_3 is a weight parameter.

For the grid cell located at $(\dot{\psi_{i,j}}, t_{i,j})$, the coordinate in body polar frame can be obtained from Eq. 2.22, the log-odds of one quasi-polar local occupancy grid cell induced by a measurement of nth obstacle feature points can be computed by:

$$\ln \frac{p(m_{i,j}|z_{t,n}, x_t)}{1 - p(m_{i,j}|z_{t,n}, x_t)} = f_n(r_{i,j})g_n(\xi_{i,j}) \quad (4.8)$$

An example is shown in Fig. 4.1. The example shows an obstacle located at $r^* = 15$ m, indicated by peak, with $\sigma_{r^*} = 1.2$ m. Between the sensor and the obstacle is

likely to be unoccupied, which has low occupied probability and the log-odds is negative, indicated by trench.

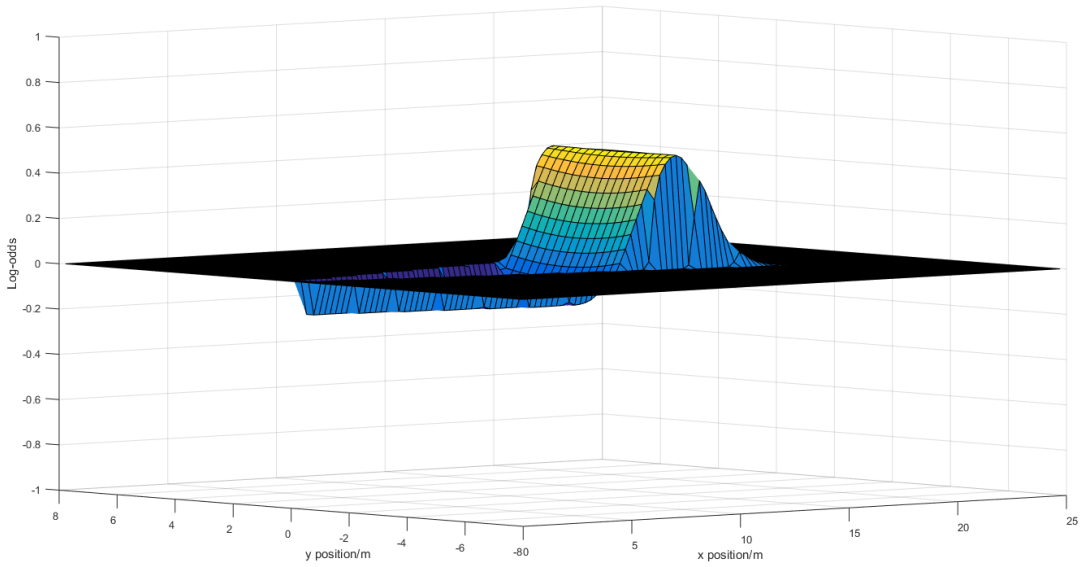


Figure 4.1. Example of the Inverse Sensor Model in log-odds form

This inverse sensor model will be used to update the quasi-polar local occupancy grid with the measurement from the sensors.

4.2.3 Sensor fusion

In Chapter 3, Section 3.2 and Section 3.3, at some time stamp, two images are obtained. And two kinds of algorithm are used for the same image pair. As Fig. 3.7, Fig. 3.8 and Fig. 3.9 show, there is some intersection between the stereo overlap region S_1, S_2 and each eye's monocular region, which means that there might be some detected obstacle feature points which belong to both stereo detection and detection of separate camera.

Since the inverse sensor model is used to assign probability for grid cells with respect to measurements, it is a conservative way to sum up all the obstacle feature points to the inverse sensor model. The repeated obstacle feature points can be regarded as the results of multi-sensors.

4.3 Motion Update

The map with assigned probability must be updated with the motion of vehicle. In an update, where one grid cell does not map directly to another, blurring will occur. For the Cartesian local occupancy grid, the grid cells only match if the rotation is a multiple 90° or the translation is the multiple grid sizes.

In the case of a Cartesian occupancy grid, methods derived from image processing techniques (such as rotation or shifting) can be used to quickly compute motion updates. For the quasi-polar occupancy grid, each cell is in a different size and shape, and cell boundaries are generally curved. Numerical methods are thus required to perform the motion update.

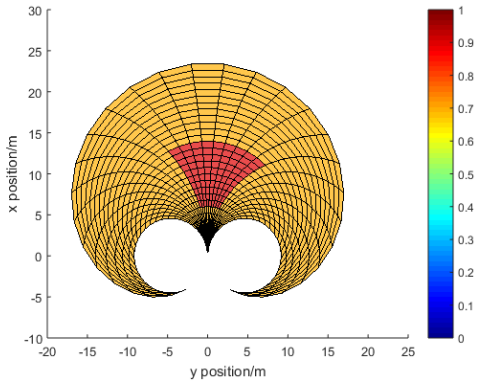


Figure 4.2. Initial grid

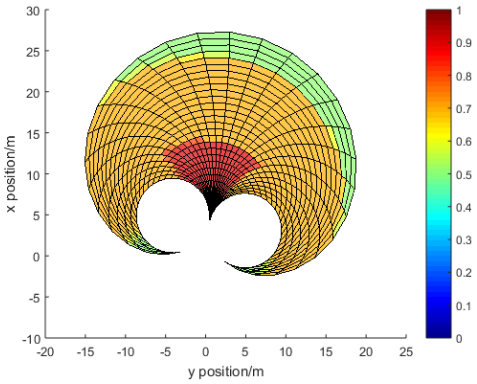


Figure 4.3. Grid after motion update

- Divide each cell into N sub small cells, $N = m \times n$.
- According to the vehicle motion, find the current small cells' location in the local occupancy grid of last time stamp.
- For one sub small cell, if the location corresponding to this cell in the local occupancy grid of last time stamp exists, assign this sub small cell $\frac{1}{N}$ the probability that in the last time stamp's occupancy grid.
- If it doesn't exist, set the probability 0.5, which means the sub small cell explores the new area so the occupied probability is 0.5.

- Weighted sum up the probability of all the sub small cells as the current cell's probability.

An example is shown in Fig. 4.2, Fig. 4.3. The vehicle is located at (0,0) with 0° heading angle, Fig. 4.2. Assume the red area has been occupied with the probability 0.9. The orange area is of 0.7 probability to be occupied. The obstacles are assumed to be static. If the vehicle moves with turn rate 0.2 rad/sec and speed 4 m/s along the body $\hat{x}_b$ axis for 1 second, the grid after motion update is shown as Fig. 4.3. One can see the relative position change between quasi-polar local occupancy grid and the obstacle area as the vehicle moves. The boundary of the occupied area also has little blur. The new explored area in green, is of 0.5 probability to be occupied.

4.4 Path Selection

The path selector selects a desired heading for vehicle from the pre-calculated path set. Then the controller uses the desired heading to compute to control inputs. Here assume the controller can be able to implement the desired control input within a range between maximum and minimum values.

With the quasi-polar local occupancy grid, the likelihood of occupancy for each path and for each discrete position along the path is known. One approach is to choose the path with the lowest occupied probability summation as the desired path. A second approach is to find a path with the longest obstacle free time t. According to the construction of the quasi-polar local occupancy grid, for two paths s_1, s_2, if the first obstacle along s_1 is located at t_1, while along s_2 the first obstacle is at t_2. $t_1 > t_2$ implies that s_1 has a longer obstacle-free distance.

4.4.1 Path selector I: Integrated path free probability

The occupied probability measures the situation whether the location of a cell in the map is occupied or not. Summation of occupied probability along each path (integrated path free probability) marks the occupied level or free situation of that path. Intuitively, the path with the lowest occupied probability summation has the low potential for vehicle to hit any obstacles. So we want to choose the path with lowest integrated path free probability.

At some time stamp, calculate the summation of occupied probability for each path. Since different path stands for different turn rate, the vehicle state of next time stamp along each path can be obtained. If only for escaping or obstacle avoidance, choose the path with the lowest probability summation as the path for next time step. The turn rate corresponding to this path will be used for the control input.

If considering the goal finding, besides the obstacles occupied probability, relative direction and distance between goal and vehicle should also be considered. For path i, create a cost function as:

$$W_i = w_1 P_i + w_2 \Delta\psi_i + w_3 \Delta r_i \tag{4.9}$$

where P_i is the summation of occupied probability for path i at current time stamp. $\Delta\psi_i$ is the relative bearing angle between the goal and the vehicle's direction of next time stamp along path i. Δr_i is the relative distance between the goal and the vehicle position of next time stamp along path i.

Choose the path with the lowest cost W_i. Use the turn rate as the control input for next time stamp.

4.4.2 Path selector II: Longest obstacle free time

There are cases where the integrated collision-free path fails to return useful paths. For example, when using the pushbroom stereo, the single disparity method can only detect obstacles in a same single depth instead of same distance, therefore the side paths, which have the higher turn rate, will cover more free space than the paths in the middle, which is unfair to use path selector I. So come up with a new path selector II:

- Set a probability threshold so that the grid cell with the likelihood greater than the threshold will be marked as occupied.
- Along each pre-calculated path from vehicle to the direction of motion, find the first cell with the occupied likelihood greater than the threshold. Record the coordinate t of that cell, which is the obstacle free time for that path.
- Consider the whole pre-calculated path set with all the obstacle free time t. Choose the path with the longest obstacle free time t^* as the path of next

time step. The turn rate $\dot{\psi}^*$ corresponding to this path will be used for the control input.

- If there are more than one path have the same obstacle free time t, choose the one that is closer to current direction.

This method can regard each path as equal, which is fair when choosing path. However, choosing threshold is another challenging issue, which will make the method less or more robust to small obstacles.

4.5 Data Flow

The flow of data through the estimation process is shown as follow.

Assume nothing known as prior. Initialize the log-odds form quasi-polar local occupancy grid to all zeros, 0.5 probability.

At some time stamp t_k:

- Prediction

  Prediction of system state and uncertainty are computed using previous estimates of state and uncertainty, system dynamics.

- Motion update

  Use the numerical method discussed in Chapter 4 to perform motion update for the quasi-polar local occupancy grid.

  Convert the grids from probability to log-odds form.

- Measurement

  Bearing and bearing rate are obtained from the separate cameras. Depth information are from stereo vision.

- Data processing

  If it's not empty in the measurement. Data processing is performed using range model Eq. 3.9, Eq. 3.20. Convert the raw vision measurement into body polar frame with range and bearing.

- Update the grid with measurements using inverse sensor model

  Use the measurement in body polar frame to update the quasi-polar local occupancy grid by inverse sensor model. Perform the recursive log-odds Eq. 2.13 in Chapter 4 by adding the current result summation.

  Convert the grids back to probability from log-odds.

- Path selection

  Select the desired path for next time stamp based on the summation of probability or cost function Eq. 4.9. The corresponding turn rate will be used for control input.

- Return to Prediction

4.6 Summary

This chapter has presented details about system design for the obstacle avoidance problem. The quasi-polar local occupancy grid serves as a map to store the location of free spaces and obstacles. How to map detected obstacles onto the local map, how to update the map as vehicle moving have been discussed. A numerical method is proposed to deal with the irregular shape of the local map cells for motion update. The inverse sensor model updates the grid with the measurements obtained from the vision system. Two pieces comprise the inverse sensor model: range and direction. It is designed as the log-odds form for time and computational efficiency. Two kind of path selector are created. One is based on the integrated path free probability. Tasks of escaping and goal finding have been discussed. A cost function is created which considers the affection of goal. The other is based on longest obstacle free time designed for the pushbroom stereo. Data flow of the whole process for the navigation problem is presented, which provides a clear overview of the whole obstacle avoidance framework.

Chapter 5
Simulation

The following chapter presents the results obtained from two set of simulation designed to test the quasi-polar frame approach and sensors proposed in this thesis. The simulation consists of two sets. The first set is for escaping from 3 different start locations with 80 initial headings of each to reach the edge of the map. The other set is for goal finding, which is navigation from 80 different starting locations to 3 different goal locations. The simulation is setup to represent a typical environment for navigation.

The description of the simulation environment and the definition of parameters and constants used in the model are described in Section 5.1. Results and analysis are presented in Section 5.2.

5.1 Setup

To demonstrate the utility of this approach, simulations of the flight through an environment modeled after the McKenna MOUT site at Ft. Benning, GA are performed.

5.1.1 Environment model

The environment is made up of several buildings and a surrounding forest, see Fig. 5.1. Each building is constructed by four walls, which represented by four lines. The urban town are defined by an ellipse 70 m by 55 m rotated 20°. For the forest, the tree trunks are represented by a circle of 8 equally spaced points. The diameters of the trees will be set varying between 0.5 m and 1.5 m. 100 trees are

located randomly outside the town, and placed no closer than 8 m to another tree.

Two groups of simulation will be performed: one for escaping, the other for goal finding. For the escape scenarios, three starting locations are selected near the middle of the urban environment. For each start location, eighty initial headings are defined (for a total of 240 starting states), and the vehicle attempts to reach the edge of the map, Fig. 5.1. In the goal-finding scenarios, eighty starting locations are defined in the forest area, and the vehicle attempts to reach each of the three locations within the urban area (for a total of 240 goal-finding scenarios, see Fig. 5.1). To prevent the vehicle starting too close to an obstacle, the start locations are defined to be at least 6 m from a tree.

Under this environment model, the measurements for the buildings are computed for each intersect points between light rays and walls by ray cast across a simulated surface. For the trees, the measurements are computed for each intersect points between light rays and the circles of trees. This method can satisfy the projection relationship so that one ray only has one intersection point with the closest surface because the surface behind will be blocked. Measurements of bearing and bearing rate are computed for each point by the sensor model discussed earlier. And those points which are within the field of view of one camera or within the stereo region are returned as valid measurements.

5.1.2 Vehicle

Assume the vehicle kinematics are non-holonomic, only motion along body $\hat{x}_b$ axis, v and $\dot{v}$ are set to be 0. It is represented by a second-order model and initialized with $u = 4$ m/s. The vehicle speed is assumed to be constant.

Vehicle kinematics are updated at a rate of 50 Hz with the second order dynamic model. Vision measurements are sampled at a rate of 10 Hz. Hence, the controller only generates control commands at a rate of 10 Hz.

5.1.3 Quasi-polar local occupancy grid

The quasi-polar local occupancy grid will be setup with $|\dot{\psi}|_{max} = 0.96$ rad/sec, with $\triangle\ \dot{\psi} = 0.06$ rad/sec, which is about 3.5°/sec. With this setup, the grid can cover an area up to 24 m away from the vehicle.

The initial belief of the environment is uncertain. Hence, the quasi-polar

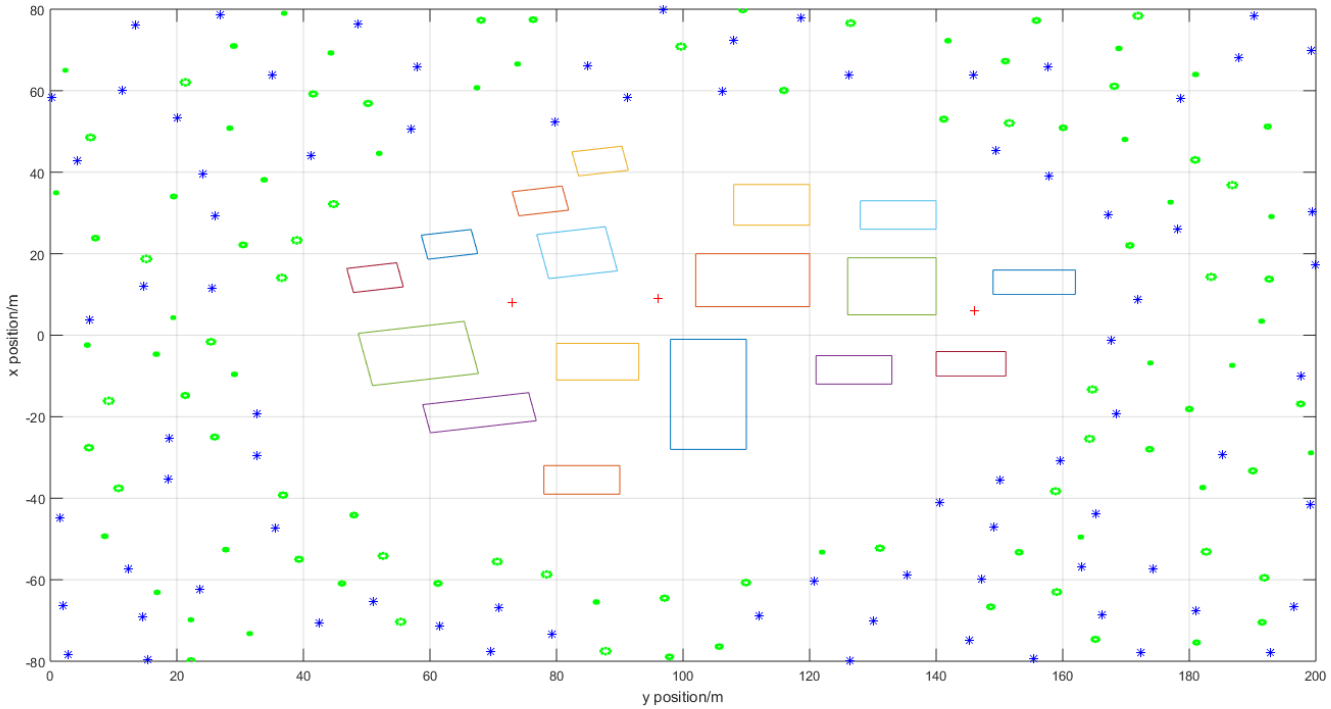


Figure 5.1. Simulation map. Polygons represent buildings, green dots circle represent trees. The blue stars are 80 different starting locations and the red crosses are the starting/goal locations.

local occupancy grid is initially set with the likelihood of 0.5 and the log-odds representation of the grid is initialized to all zeros.

5.1.4 Camera model

In these simulations, one single camera for stereo vision has a field of view of 90°. The maximum range is $r_{max} = 24$ m, where values of r^* beyond that return no useful information. With $t_{max} = 6$ s, and a constant speed 4 m/s, this means that most of the measurements can be represented in the local occupancy grid. For the stereo vision, the baseline is 0.2 m. The valid depth for single disparity method is set to 5 m. Set $\theta = 15°$, so there is 60° overlap field of view. For comparison, we also simulate a monocular vision camera with a 120° field of view. It can be seen from Fig. 5.2 and Fig. 5.3 that they have the same field of view for monocular region, while the combined stereo can obtain more depth information from pushbroom method. The purpose of the monocular system is to examine the benefit of the stereo region for obstacle avoidance.

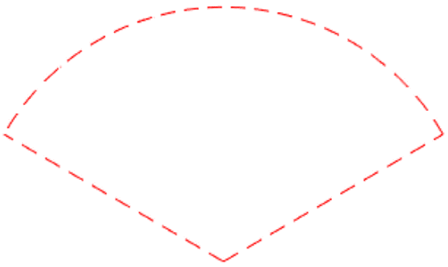


Figure 5.2. Monocular with wide field of view

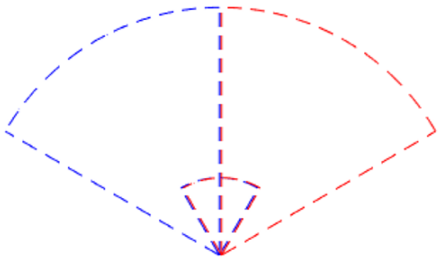


Figure 5.3. Combined stereo (Bird eye)

Hence each scenario will be tested with three camera models:

- Case I: Use only monocular vision with wide field of view (to test the proposed quasi-polar local occupancy grid).
- Case II: Use simple stereo with pushbroom method to test the stereo sensor.
- Case III: Use combined stereo model to do the navigation and obstacle avoidance.

Table 5.1. Standard Deviation for Noises

$\sigma_\beta = 0.625^\circ$	$\sigma_{\dot{\beta}} = 1.25^\circ/s$
$\sigma_u = 0.2m/s$	$\sigma_\omega = 2^\circ/s$
$\sigma_{a_x} = 0.05m/s^2$	$\sigma_{\dot{\psi}} = 0.25^\circ/s$

Table 5.2. Constants used for system

$c_1 = 0.15$	$c_2 = 1.5$	$c_3 = 15$
$w_1 = 0.9$	$w_2 = 40.0$	$w_3 = 0.2$

5.1.5 Success criteria

A single run consists of motion from a start position and flying towards the goal while keep moving to avoid obstacles. The simulation time allowed for a single run is 60 s. The run is called a failure if the vehicle comes within 1.0 m of any obstacle. One run will stop if the vehicle moves out side the boundary of the map. For escaping, one run is called a success if the vehicle moves out side the boundary of the map. For goal finding, if the vehicle reaches within 2 m of the goal, the simulation is called a success. The time for a single run before the vehicle stops or fails will be recorded for the following analysis.

5.1.6 Parameters in system

The parameters used for standard deviation are given in Table 5.1. The noises are assumed to be Gaussian.

The constant values $c_1 - c_3$ are for inverse sensor model, and $w_1 - w_3$ for path selector are given in Table 5.2.

5.2 Results

5.2.1 Escaping

For each camera model, one run is executed from each starting location in 80 starting directions for total 240 runs. The results for three cameras models are shown in Table 5.3, 5.4, 5.5.

Results for the wide field of view monocular vision system are summarized in Table 5.3 and shown in Fig. 5.4.

Table 5.3. Results for monocular with wide field of view

-	Starting location 1	Starting location 2	Starting location 3	Total
Success	72 (77)	69	52 (55)	193 (201)
Crash	3	10 (3)	22 (10)	35 (16)
DNF	5 (0)	1 (1)	6 (3)	12 (4)
Total	80	80 (73)	80 (72)	240 (225)

Table 5.3 shows that the monocular vision system combined with the quasi-polar occupancy grid is able to escape 80% of the cases, with 14.6% of cases leading to a crash and 5% did not finish. Closer analysis revealed that 19 of the 35 crashes occurred within the first 3 seconds, indicating that the vehicle simply did not have time to adequately resolve nearby obstacles to avoid collision (or was unable to avoid collision because of kinematic constraints). The results shown in parentheses remove cases where a crash occurred within 3 seconds, which gives a successful escape rate of nearly 90%.

Results for the pushbroom stereo system combined with the quasi-polar occupancy grid are summarized in Table 5.4 and shown in Fig. 5.5 and Fig. 5.6.

For the pushbroom stereo, the "integrated path free probability" (path selector I) is generally larger for the high-turn rate paths, leading to circular flight (and thus failures due to DNF). This can be explained that the single disparity method only detects obstacles in the same depth instead of same distance. The obstacles detected by this method on both the side paths have longer distance from the vehicle than that on the middle paths. Since the space between the vehicle and the obstacles is marked as free space, the summation of occupied probability of the side path will be lower than that of the path in the middle. Hence, the pre-calculated path with the maximum turn rate would often be selected. Changing the path selector to "longest obstacle free time" (path selector II) with an appropriate threshold allows the vehicle to select paths that are straighter, leading to a significantly higher success rate.

Table 5.4. Results for pushbroom stereo. Results are shown as x/y (z), where x is path selector I, y is path selector II, and z removes runs which crashed within 3 s of start for path selector II.

-	Starting location 1	Starting location 2	Starting location 3	Total
Success	0/5	0/10	0/13	0/28 (28)
Crash	8/75	14/70 (58)	12/67 (46)	34/212 (179)
DNF	72/0	66/0	68/0	206/0
Total	80	80 (68)	80 (59)	240 (207)

Note that in this environment the pushbroom stereo performed poorly, with a success rate of only 13.5%. More detailed analysis could provide more insight: a better path selector could improve results, but it is also likely that the parameters of the pushbroom stereo used here (baseline, distance) are not compatible with dense obstacle fields.

Results for the combined stereo system combined with the quasi-polar occupancy grid are summarized in Table 5.5 and shown in Fig. 5.7. Results are very similar to the monocular vision case, indicating that stereo vision is less useful for collision avoidance during forward flight at high speed than pure monocular vision.

Table 5.5. Results for combined stereo

-	Starting location 1	Starting location 2	Starting location 3	Total
Success	72 (73)	64	50 (56)	186 (193)
Crash	5	16 (9)	24 (7)	45 (21)
DNF	3 (2)	0	6 (0)	9 (2)
Total	80	80 (73)	80 (63)	240 (216)

We can conclude that for the high speed navigation using quasi-polar local occupancy grid, monocular vision based on optical flow can provide most of the useful information. Combining pushbroom stereo with the monocular doesn't make much improvement for navigation. However, pushbroom method, as a stereo vision method, has the characteristic of general stereo system. It can provide a more accurate depth information of obstacles. For safety flight and navigation, the pushbroom stereo can be used as final check to stop the vehicle (assuming that it is capable of sudden stop or hover).

5.2.2 Goal finding

The proposed approach can be applied to goal finding. Use the idea based on cost function to do the goal finding, which combines the factors of the occupied probability of each path, the relative bearing and distance between predicted vehicle location and goal. For each camera model, one run is executed from each randomly generated starting location to each goal location. Preliminary results shows that the idea has a great potential to navigate towards the target while avoiding the obstacles.

Fig. 5.8 and Fig. 5.9 show the vehicle flight paths from a sample of successful runs of simulation using the monocular system and the combined stereo/monocular system, respectively. Due to the complexity of the environment and the cost function based path selector, it is possible that the shortest path to the goal had a lower potential than a longer path considering the noisy measurements. In this case, the vehicle could have turned away from the optimal path to pursue one with a higher certainty of safety. The parameters in the cost function based path selector play an important role in the simulation results. How to balance the three factors in the cost function emerges to be a key issue. Larger w_1 makes the occupied probability of each path weighted more, which leads to a more conservative path selector so that most of the runs will come to DNF. Larger w_2 or w_3 makes the relative direction or distance between vehicle and goal more important so that the path selector works aggressively which results more crashes.

Fig. 5.10 and Fig. 5.11 display the results from a representative run of the simulation for the monocular system and the combined stereo/monocular system, respectively. Note the obstacles are accurately identified. The cost function approach based on pre-calculated path causes the vehicle to steer away from the detected obstacles and try to get close to the goal.

It can be seen that a particular obstacle is localized with greater accuracy if it remains within the field of view. Also, the confidence for the free space increases with time in the field of view as the space keep being unoccupied. The model handles both small point obstacles, trees and large complex obstacles, buildings equally well.

Like the traditional occupancy grid, the quasi-polar local occupancy grid also maintains its memory of space which has left the field of view. This is important

for the controller use the information of what is not currently in view but directly to the side of the vehicle to do the navigation, especially around corners and trees. The numerical method for motion update successfully translates the occupancy grid beliefs such that the grid memory is possible. Particularly, the novel construction of the quasi-polar local occupancy grid determines there is less work for path planning after the map obtained. Pre-calculated path only leaves path selection work at every sensor update.

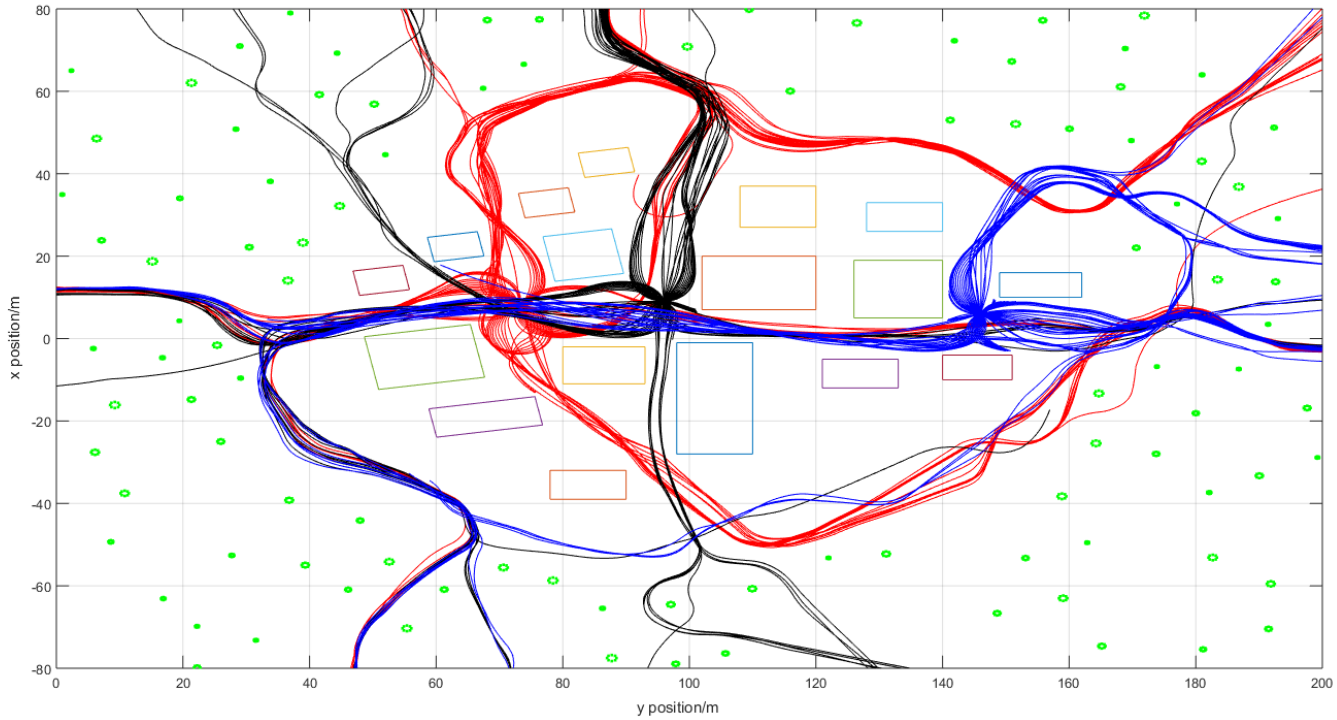


Figure 5.4. Results of escaping using sensor I. Red lines are the result paths from start location 1.Black lines are the result paths from start location 2. Blue lines are the result paths from start location 3.

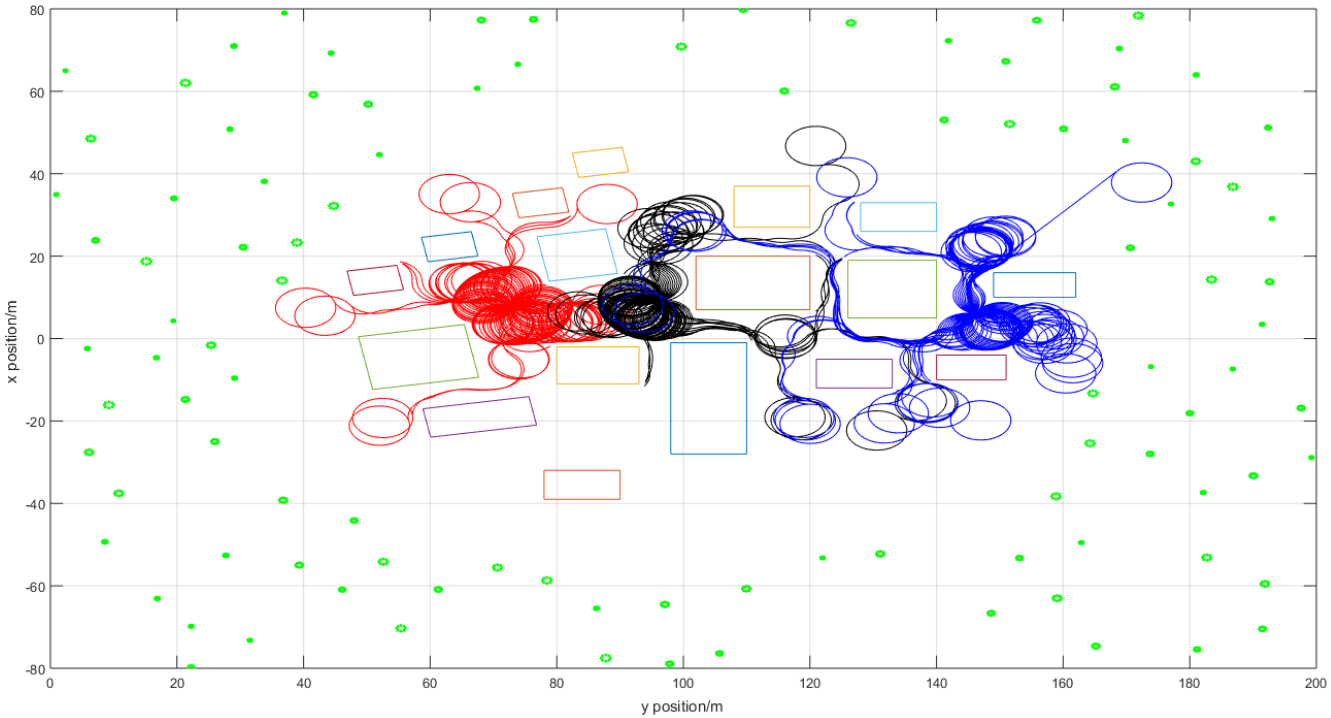


Figure 5.5. Results of escaping using sensor II with path selector I. Red lines are the result paths from start location 1.Black lines are the result paths from start location 2. Blue lines are the result paths from start location 3.

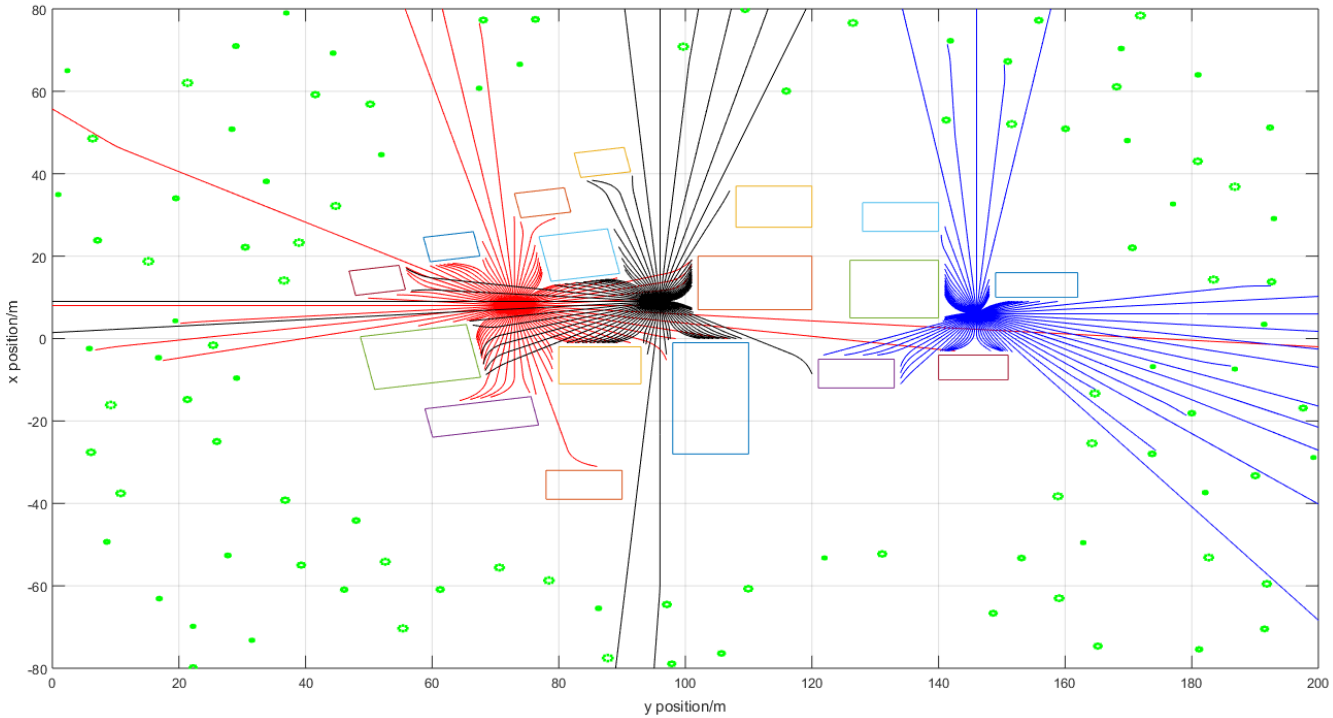


Figure 5.6. Results of escaping using sensor II with path selector II. Red lines are the result paths from start location 1.Black lines are the result paths from start location 2. Blue lines are the result paths from start location 3.

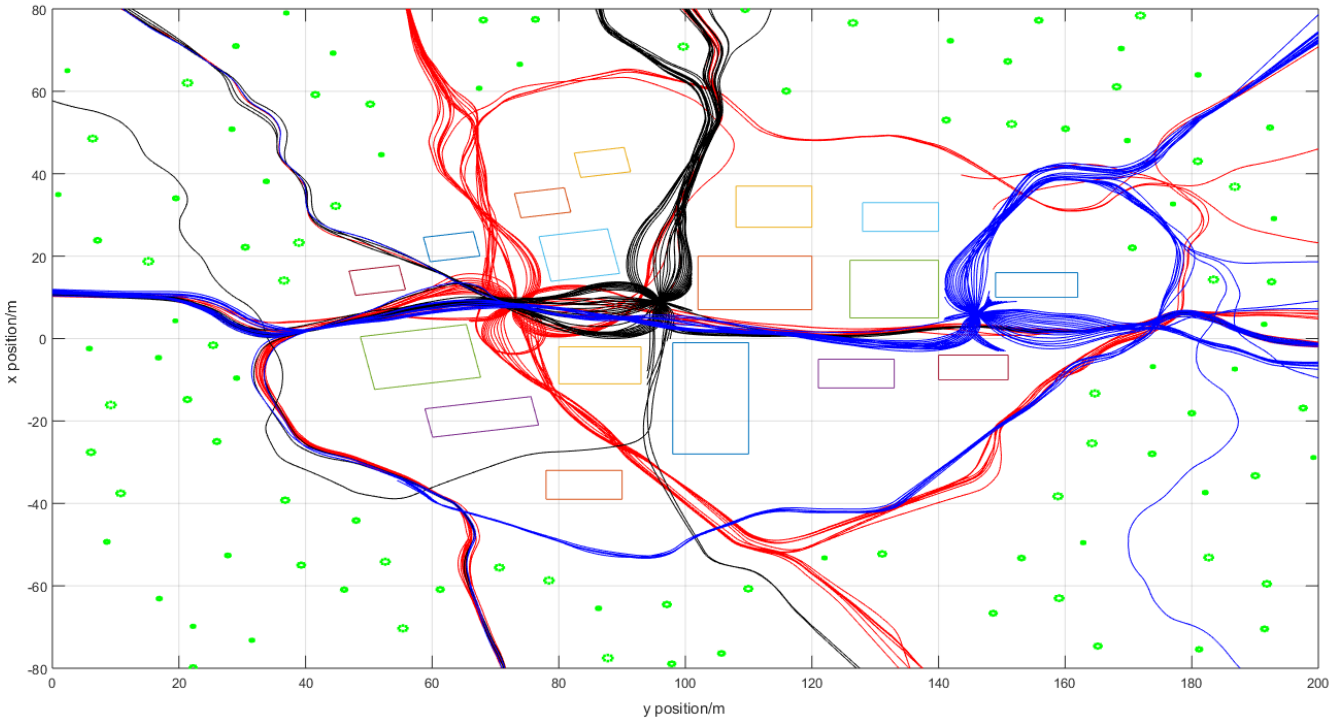


Figure 5.7. Results of escaping using sensor III. Red lines are the result paths from start location 1.Black lines are the result paths from start location 2. Blue lines are the result paths from start location 3.

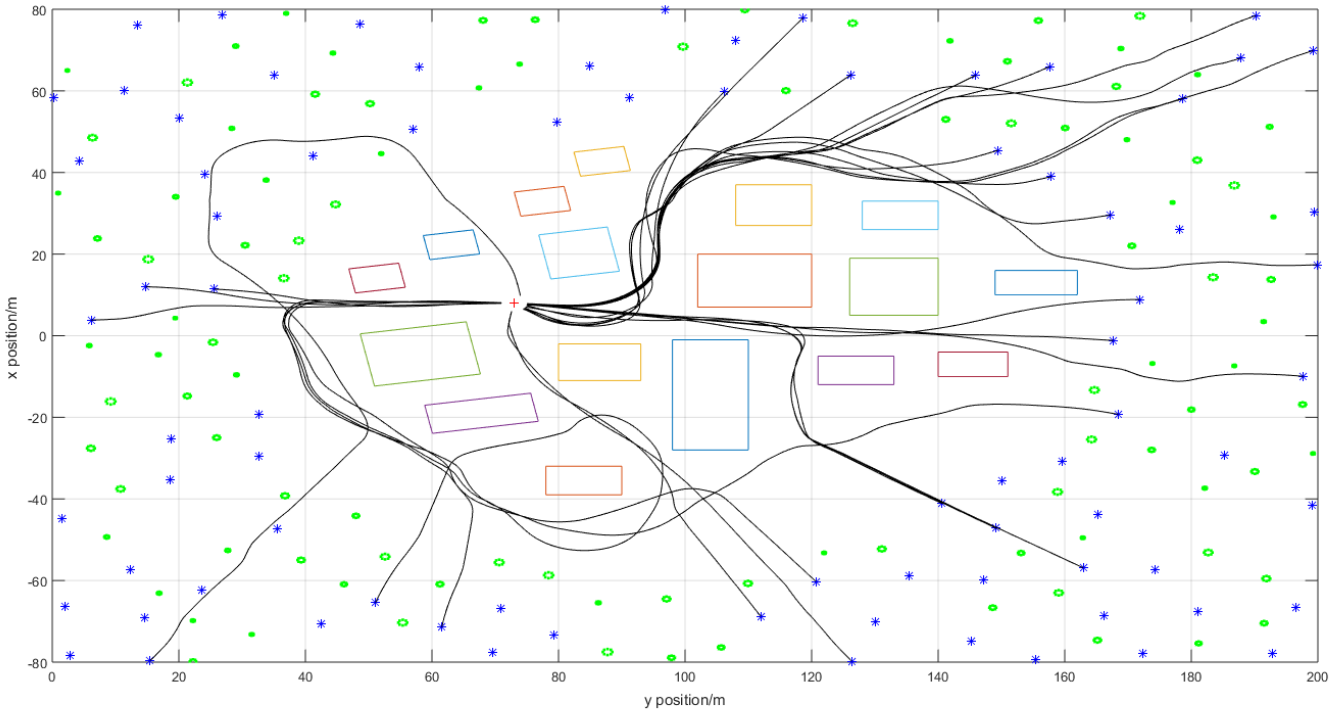


Figure 5.8. Flight path from select simulations using sensor I. The black lines here show the paths of simulations which successfully navigate to the goal.

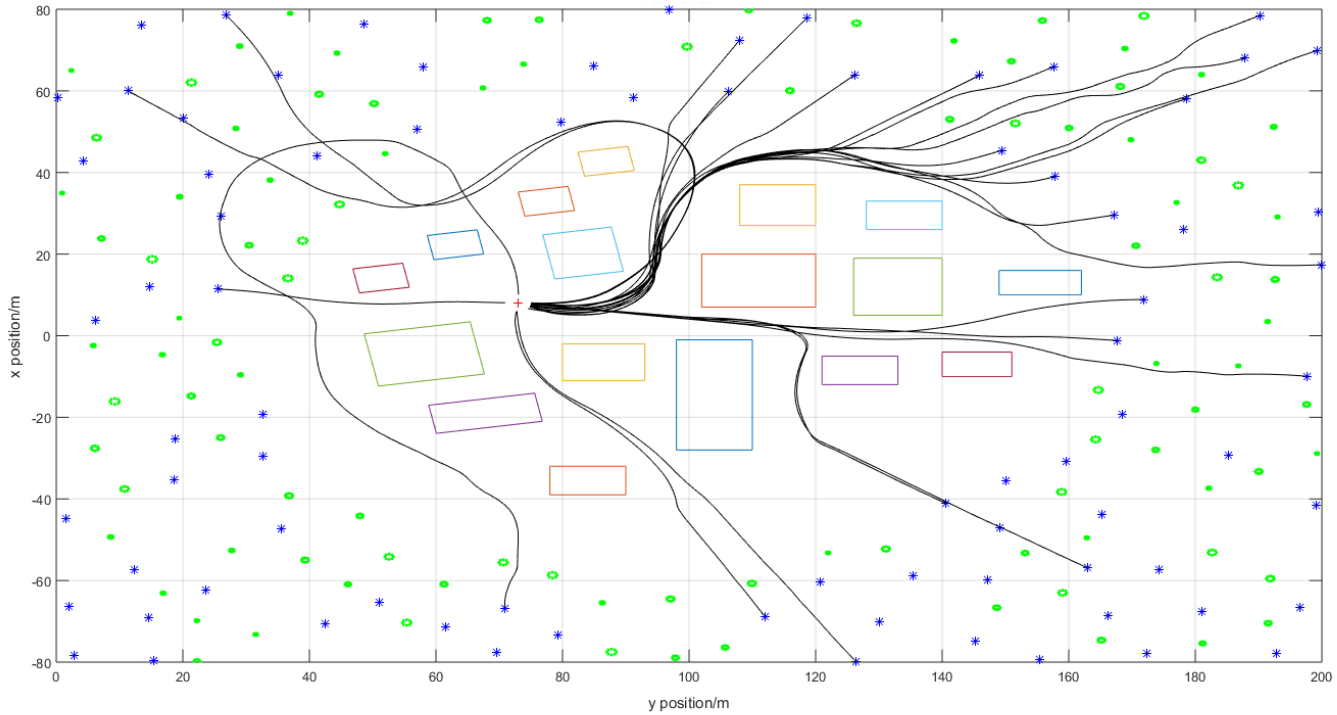
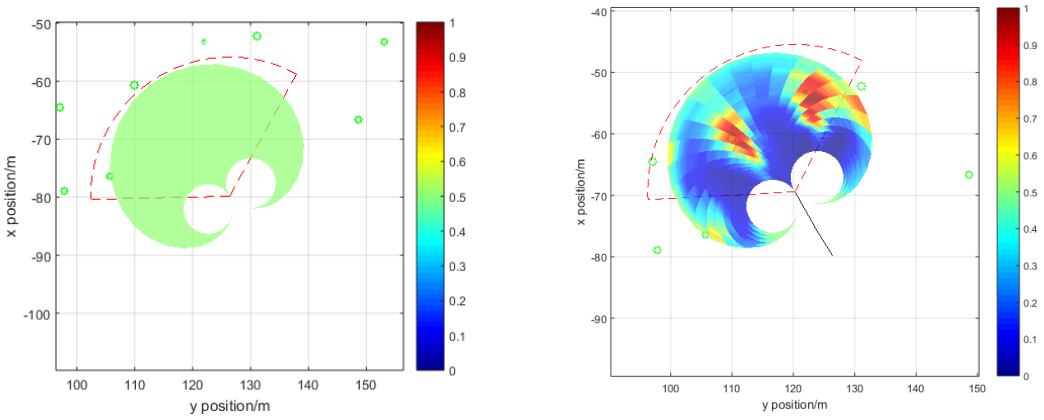
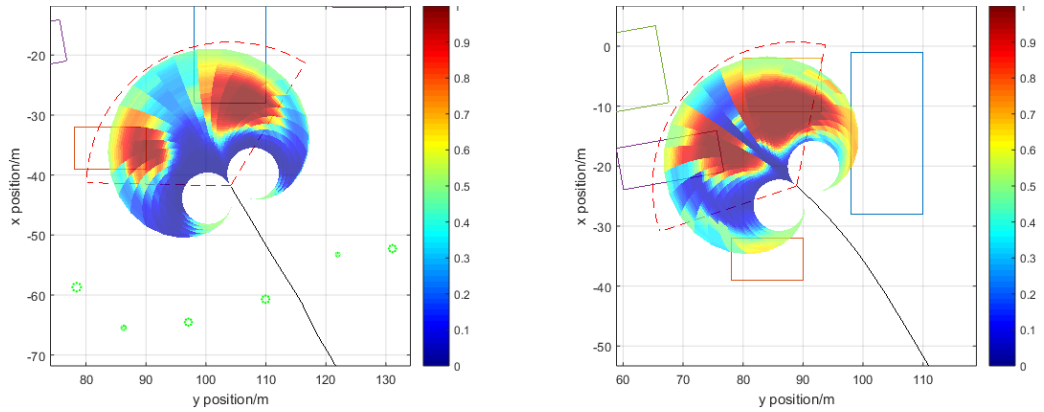


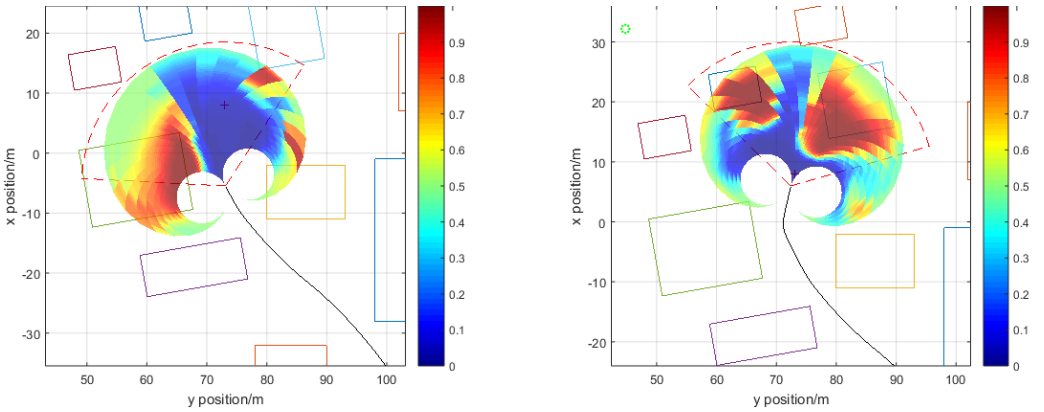
Figure 5.9. Flight path from select simulations using sensor III. Again, the black lines here show the paths of simulations which successfully navigate to the goal.



(a) 0 seconds **(b) 4 seconds**

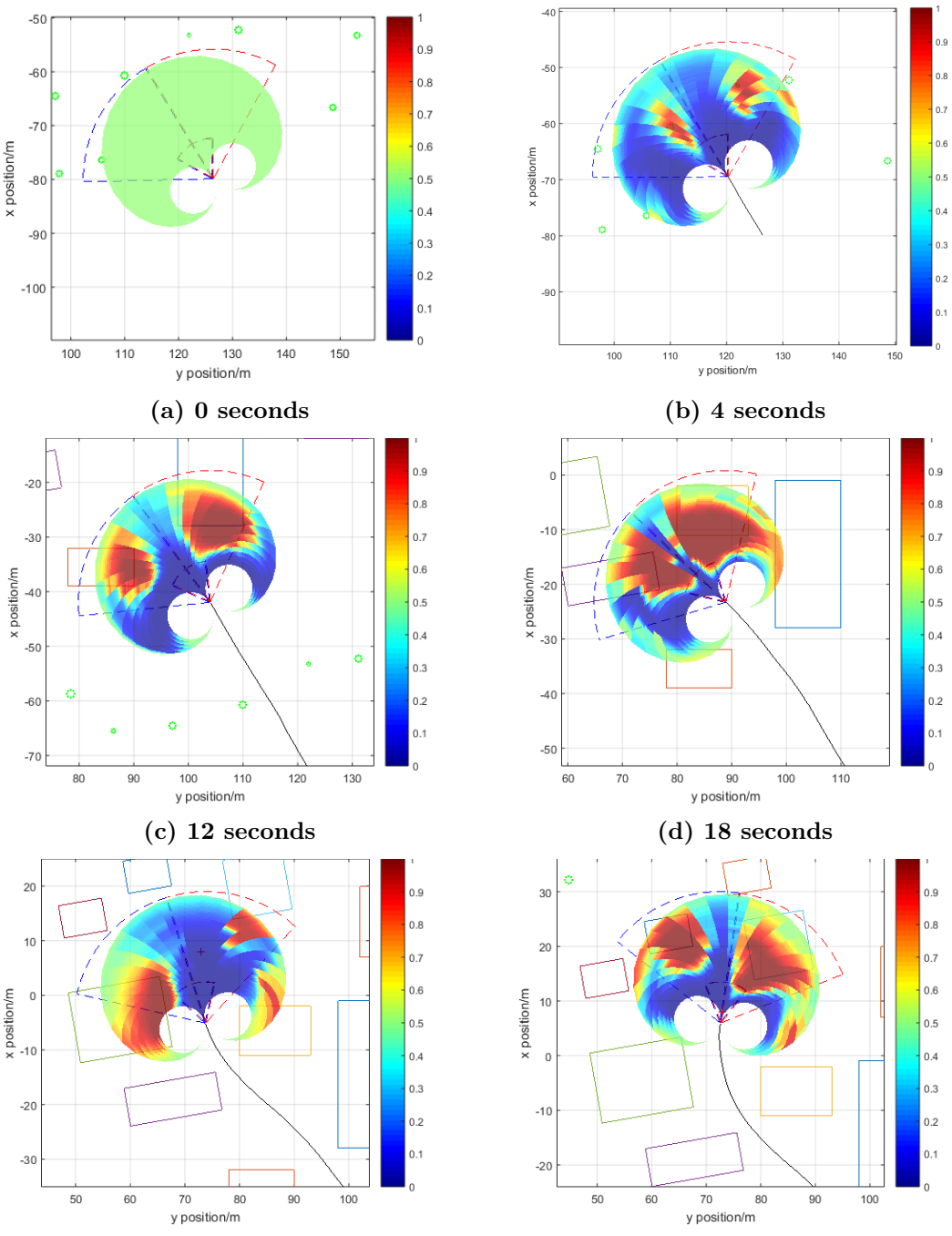


(c) 12 seconds **(d) 18 seconds**



(e) 24 seconds **(f) 26.04 seconds**

Figure 5.10. Sequence of images from representative run using sensor I. Camera field of view is shown as red dashed lines. Vehicle is regarded as a point. Quasi-polar local map is shown as the approximate sector. Cell color represents occupancy probability: red = high, blue = low, green = 50%(uncertainty). Obstacles are shown as lines of building or dot circle trees. The red cross is the goal, see subfigure (e). The vehicle trace is shown as black line.



(a) 0 seconds **(b) 4 seconds**

(c) 12 seconds **(d) 18 seconds**

(e) 24 seconds **(f) 25.82 seconds**

Figure 5.11. Sequence of images from representative run using sensor III. Camera field of view is shown as red and blue dashed lines. Vehicle is regarded as a point. Quasi-polar local map is shown as the approximate sector. Cell color represents occupancy probability: red = high, blue = low, green = 50%(uncertainty). Obstacles are shown as lines of building or dot circle trees. The red cross is the goal, see subfigure (e). The vehicle trace is shown as black line.

Chapter 6
Towards Hardware

This chapter introduces the preliminary work towards hardware mainly for the vision system. There are many challenges to make a good use of the sensors. For the stereo system consists of two separate cameras, camera calibration is challenging. Image processing, which relies on camera calibration, plays an key role in extracting useful information from images.

The flow of vision system is shown in Fig. 6.1

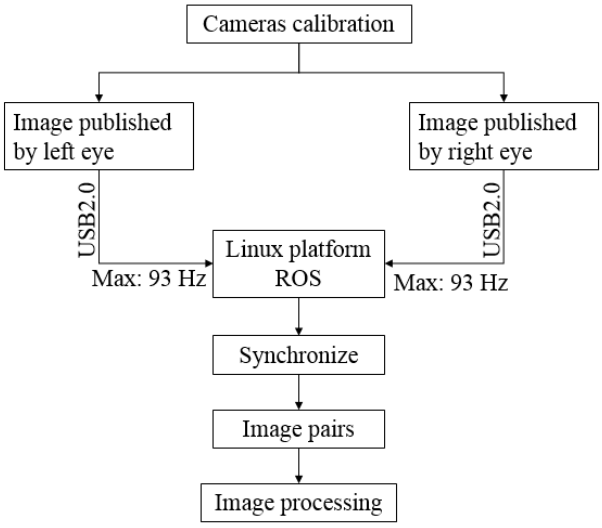


Figure 6.1. Schematic of vision system

6.1 Vision System Setup

The cameras are two Matrix Vision BlueFOX-IGC cameras equipped with two 90° field of view lens. Resolution of each camera is 752 × 480 pixel. The maximum frame rate is 93 Hz. The images are transmitted through USB 2.0 interface to ROS (Robot operation system) on a Linux platform computer.

The cameras are mounted on a flat plane of an aluminum bar. Several pair of holes are made to adjust the baseline of stereo vision.

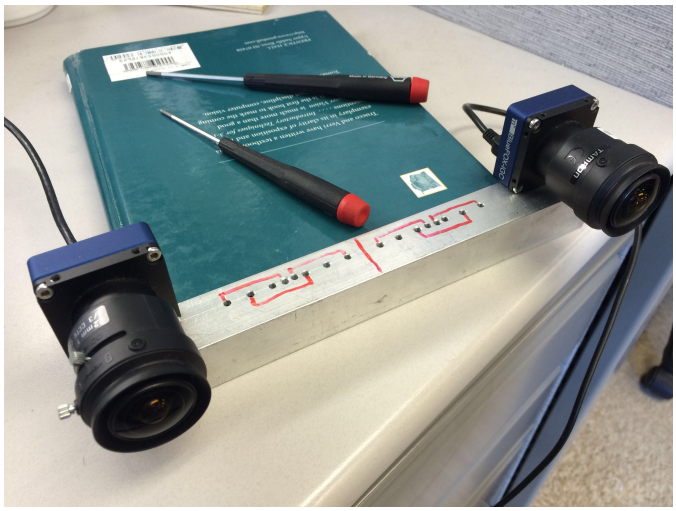


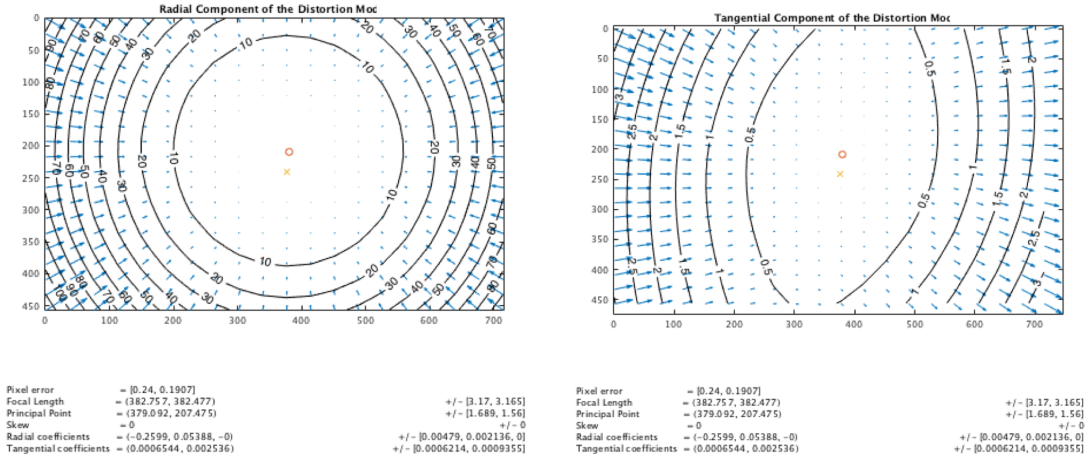
Figure 6.2. Mount cameras

6.2 Camera Calibration

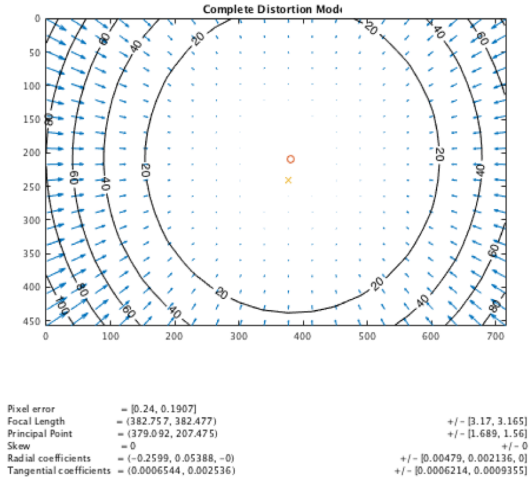
Camera calibration determines the value of the extrinsic and intrinsic parameters of the camera. The extrinsic parameters describes the relation of an object in camera coordinate system and world coordinate system. While the intrinsic parameters describes the coordinate transformation between film coordinates (projected image) and pixel array. A 3 × 4 projection matrix performs the coordinate transformation between world coordinate and pixel coordinate.

In reality, camera distortion exists since the optical device has non-linearities. Radial and tangential components consist the total distortion, see Fig. 6.3.

For stereo vision, rectification is needed before 3-D reconstruction. Given a pair of stereo images, rectification determines a transformation of each image such that



(a) Radial component **(b) Tangential component**



(c) Total distortion model

Figure 6.3. Camera distortion

pairs of conjugate epipolar lines become collinear and parallel to one of the image axes, usually the horizontal one, [25]. It is important that the correspondence problem, which involves 2-D search, is reduced to a 1-D search on a scanline identified trivially, see Fig. 6.4.

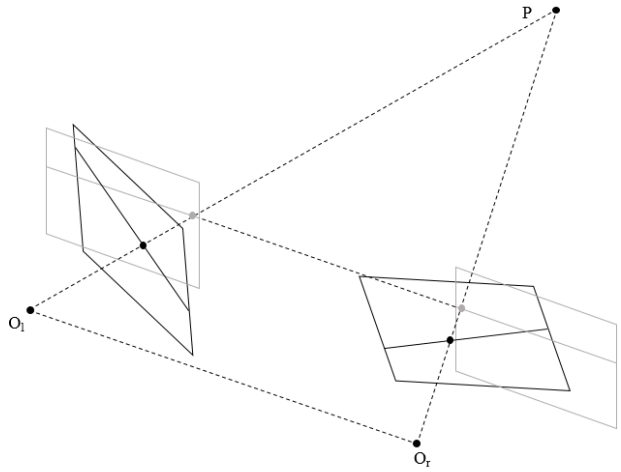


Figure 6.4. Rectification of a stereo pair. The epipolar lines associated to a 3-D point P in the original cameras (black lines) become collinear in the rectified cameras (light grey) [25]

Use the ROS package "camera_calibration" with a 14×9 chessboard to do the camera calibration. The important matrices $K_{3\times3}$: camera matrix, $D_{1\times5}$: distortion coefficients, $R_{3\times3}$: rectification matrix, $P_{3\times4}$: projection matrix can be obtained.

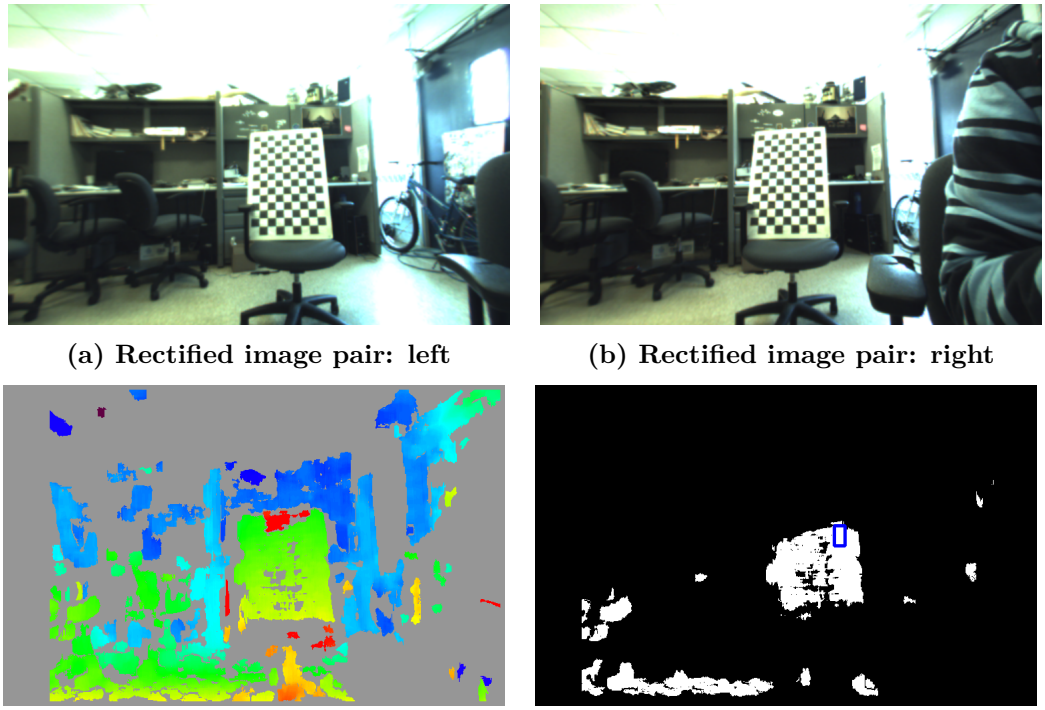
6.3 Communication and Synchronization

The two cameras are connected to the Linux computer with ROS through USB 2.0. Images can be published by the cameras at the maximum rate of 93 Hz. 10 Hz and 20 Hz have been tried. The two cameras are synchronized by hardware synchronization. Also, as for the receiver side, ROS package "message_filters" is used to synchronize the topics of cameras to grab image pairs. The synchronized image pairs are used for the image processing.

6.4 Image Processing

6.4.1 Disparity map

The first idea is to compute the disparity map of every image pair. Disparity map is the apparent pixel difference or motion between a pair of stereo image. With a



(a) Rectified image pair: left

(b) Rectified image pair: right

(c) Raw disparity map.

(d) Processed disparity map by block-mean method

Figure 6.5. Processing on a disparity map (Baseline is 20 cm, simple stereo model). The raw disparity map is noisy. A chessboard is 1.7 meters away from the cameras. By applying the block-mean method, the blue rectangle is used to locate the chessboard.

disparity map, the real location of one object can be reconstructed: direction is obtained from the corresponding pixel location in image, depth can be obtained from the disparity value Eq. 3.27.

The disparity map can be obtained by using ROS package "stereo_image_proc". The algorithm used in this package is the standard block-matching stereo processing, searching for obstacles at every depth. The raw disparity map contains much noise and will be noisier when the cameras are in motion. It is very sensitive to the light changes. A block-mean method is used to find interested feature point. First, a binary disparity map is obtained by setting a disparity threshold (depth threshold) so that the objects in the interested depth region can be filtered out. Divide the binary disparity map into blocks with each block size 30×16. Compute the block mean and use the center location of maximum mean block to stand for the real object location, see Fig. 6.5.

There are many shortcomings of this method. The accuracy of disparity map

relies heavily on camera calibration. In real flight, small vibration of the UAV will cause poor disparity map. Also, the disparity map is sensitive to the light changes, which brings much challenges for indoor/outdoor flight. As for the computation concern, high frame rate stereo vision processing for disparity map requires high computational memory and effort.

6.4.2 Pushbroom stereo

Barry speeds up the stereo processing by limiting the search through distance to a single value. Edge map and sum of absolute differences (SAD) are used for block-matching to detect pixel block similarity [10].

We implemented this algorithm based on Barry's online open source repository. As a frontier work for multi-region processing like bird vision, the image pairs are divided into three regions. In the middle region, the cut images are re-calibrated and sent to pushbroom stereo algorithm to reconstruct the 3-D location of detected feature points. In the two side region, feature points are detected based on Harris corner. Track these feature points and calculate the optical flow by applying the OpenCV functions: "goodFeatureToTrack", "calcOpticalFlowPyrLK", see Fig. 6.6.

This method is robust to camera calibration. The single disparity detection makes the processing fast, which is suitable for flight experiment of small UAV. The combination of monocular and stereo method also shows the sensor model used in simulation is feasible.

6.5 Summary

The chapter presents the preliminary work towards hardware. The setup of vision system is proposed. Camera calibration, communication and synchronization are then introduced. Two methods of image processing are presented to make use of stereo information: disparity map based processing and pushbroom stereo based processing. Preliminary results show the feasibility of the sensor model used in simulation.

(a) Left eye

(b) Middle region of the stereo pair

(c) Right eye

(d) Feature tracking of right eye's right region

Figure 6.6. Multi-region processing (Baseline is 20 cm, simple stereo model). (a) and (b) show the application of pushbroom stereo. Apply pushbroom algorithm to the middle region of the stereo pair. 3-D location of the features are shown on the left eye's image as grey small rectangles. The green dots in left eye's left region are the feature points obtained by Harris corner extraction. (c) and (d) show the feature tracking of the right eye's right region. Also, the green dots in right eye's right region are the Harris corner feature points. Optical flow is calculated in the monocular region.

Chapter 7
Conclusion

The research in this thesis is motivated by simplifying the process of navigation through an obstacle field: typical methods follow a two step process of map generation followed by trajectory computation; the method proposed here directly computes the likelihood that a given path is collision-free. Trajectory generation simply consists of selecting the path that is likely to be free over the longest distance. Small UAVs have significant potential for application in fields ranging from search and rescue to surveillance and exploration. Navigating quickly through indoor, outdoor or 3D unstructured environment due to their small size, superior mobility requires a system of navigation and obstacle avoidance using onboard sensors. This thesis focused on the problem of estimating obstacle locations on a quasi-polar map and vision-based avoiding obstacles using cameras and GPS/INS.

Current mature technologies for obstacle detection using onboard sensors rely generally on LIDAR, sonar or radar, which require large power and weight cost. Although they provide high-quality information, these sensors generally have trouble meeting the size, weight and power restrictions of small UAVs. Light weight cameras are a good solution due to the low cost and low power charge. Monocular vision has longer accessible range and faster processing ability, but it cannot observe the depth information of obstacles. Scale and depth information are available on stereo vision. However, small baseline limits the stereo only accessible to obstacles in a short range.

Typical occupancy grid deals with obstacle mapping in a spatial lattice, usually under the Cartesian coordinate on a global dimension. In mobile robot navigation, grid-based representation of the environment are often used for planning a path from some initial robot location to a desired goal location. Existing methods for

path planning are mainly based on graphic algorithms or potential field. However, all of the existing methods need two-step procedure of map generation followed by computation of feasible collision-free trajectory.

This thesis proposed a quasi-polar local occupancy grid based algorithm for obstacle avoidance so that the two-step procedure for planning work can be simplified as one step. Also, different kind of sensors are presented. Especially, a combined sensor which combines the general stereo with the pushbroom stereo motivated by a bird's vision system is come up with to take advantage of separate sensors.

The approach brings many new challenges besides the challenges posed by the estimation problem: noisy vision measurements reduce the certainty of obstacle information; highly non-linear governing equations of system and measurements lead to significant uncertainties in the problem. First, the mapping problem cannot be performed in the traditional Cartesian or polar coordinate. Second, motion update for the novel occupancy grid is difficult since the cells in the new map have different shapes and size. The boundary of grid cells can be straight line or arc curve, which means the classic image processing method cannot be used.

A quasi-polar local occupancy grid method for UAV obstacle avoidance using only GPS/INS and vision framework is developed in this thesis. This occupancy grid maps sensor data directly onto a set of feasible paths, so that path planning consist simply of selecting the path with the lowest likelihood of collision. The map also stores estimates of obstacle locations and take account the measurement noise and sensor dropout. A numerical method for motion updates that can cope with the different sizes and shapes of each cell in the occupancy grid is proposed, and a probability-based inverse sensor model that maps range and bearing-based sensor data to this path-based occupancy grid is developed.

The new framework is tested on a 2D simulation platform by using three kind of sensor models: monocular vision based on optical flow with a wide field of view, pushbroom stereo only and combined monocular and pushbroom stereo. This last sensor model is inspired by a bird's vision system, which uses eyes that have only a partially-overlapping field of view (so that there is wide-field monocular vision available to the side, and narrow field of view stereo vision available forwards).

Two sets of Monte Carlo simulations are performed for each sensor model in an environment modeled after the McKenna Military Operations in Urban Terran site as Ft. Benning, GA: escaping and goal finding. The results of escaping shows

that pushbroom stereo alone is not suitable for high speed obstacle avoidance and navigation under the quasi-polar local map, since the stereo can provide too little information to satisfy the navigation requirement. With the same width field of view, monocular vision works almost as well as the combined sensor model, which has a success rate of 89.5% to escape, that means the monocular provides most of the valid information of the environment for the vehicle navigation.

The path-based occupancy grid approach works well for mapping the obstacles, storing and memorizing the occupied information of the obstacles not in field of view. Particularly, most of the path planning work is essentially done before the map obtained: only path selection is required, which significantly reduces the complexity of path planning.

Preliminary work towards hardware is presented mainly for vision system. The setup of the vision system and implementation of image processing using two algorithm show the feasibility of the sensors models used in simulation.

7.1 Summary of Contributions

7.1.1 Quasi-polar local occupancy grid

A quasi-polar local occupancy grid is proposed in this thesis. This method calculates a set of feasible paths before obstacle mapping and path selection, which makes the path selection work simplified.

7.1.2 Combination of different camera sensor models

Different kind of camera models are developed, including monocular vision based on optical flow, pushroom stereo based on single disparity and a combined sensor which combines the monocular vision with the pushbroom stereo. The method takes advantage of the separate sensors while overcomes the disadvantage of them.

7.1.3 New path selection method

Two new path selection methods based on the pre-calculated paths set are given considering the integrated path free probability or longest obstacle free time.

7.1.4 Performance verification through simulation

Two group of simulation are performed to test the performance of the designed system. Results of simulation show that the quasi-polar local occupancy grid based implementation can be used for obstacle avoidance. The locations of obstacles are accurately estimated. The pre-calculated path based path selectors work well. Different sensor models are compared and discussed.

7.2 Recommendations for Future Work

7.2.1 Path Selector for Goal Finding

The parameters in path selector and inverse sensor model play a key role in goal finding. How to balance the three factors in the cost function of the path selector emerges to be a challenging issue. Larger w_1 makes the occupied probability of each path weighted more, which leads to a more conservative path selector so that most of the runs will come to DNF. Larger w_2 or w_3 makes the relative direction or distance between vehicle and goal more important so that the path selector works aggressively which results more crashes. The parameters in inverse sensor model determines the raw occupied probability level in the local map. A method (such as machine learning) to train these parameters could provide an effective way selecting path so that lead to the goal. A good path selector model to solve the optimization problem could also help.

7.2.2 Vehicle Speed

The situation considered in this thesis is the small UAV moving in a constant speed. More general situation is that the vehicle speed varies. Some random motion in v should also be considered. Since the shape of quasi-polar local map depends on the vehicle speed, various speed in different time stamp will cause the shape of local map different so that a new method (such as computer graphics) for motion update is needed to be applied.

In addition, various speed requires regenerating local map in different time stamp. The required computational effort will grow. Further research is needed for the computational performance.

7.2.3 Two dimensions To Three Dimensions

The work presented in this thesis accounts for only planar motion, though the world is three-dimensional. In practice, these small UAVs will have to navigation through obstacles with vertical components such as branches, doorways.

The quasi-polar local occupancy grid used here can work in a third dimension, which becomes a quasi-conic occupancy grid. Different from the planar motion, there will be roll and pitch attitude and vertical translation as the third dimension added. Since pitch and roll variations could add significant noise to the vision based solution, a new method to processing the vision data and estimate the obstacle location is needed. Also, new techniques need to be researched to meet the computer memory and computational effort.

7.2.4 Complete Hardware Implementation

The simulation results show that the approach works in theory. Preliminary work towards hardware for vision system shows the feasibility of the sensors model used in simulation. An actual complete hardware implementation will prove whether the system can function in real time. The vehicle will operate in a three dimensional environment and thus have six degrees of freedom (DOF) (in contrast to the three DOF discussed in this thesis).

Bibliography

[1] "Bird vision," `http://en.wikipedia.org/wiki/Bird_vision`, accessed: 2016-05-05.

[2] "Binocular Vision in Pterosaurs," `http://pterosaurheresies.wordpress.com/2012/04/04/binocular-vision-in-pterosaurs`, accessed: 2012-04-04.

[3] Badino, H., U. Franke, and R. Mester (2007) "Free space computation using stochastic occupancy grids and dynamic programming," in *Workshop on Dynamical Vision, ICCV*, vol. 20, Rio de Janeiro, Brazil.

[4] Danescu, R., F. Oniga, and S. Nedevschi (2011) "Modeling and tracking the driving environment with a particle-based occupancy grid," *Intelligent Transportation Systems, IEEE Transactions on*, **12**(4), pp. 1331–1342.

[5] Marlow, S. Q. and J. W. Langelaan (2011) "Local terrain mapping for obstacle avoidance using monocular vision," *Journal of the American Helicopter Society*, **56**(2), pp. 22007–22007.

[6] Weiss, S., M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart (2012) "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, IEEE, St Paul, MN, pp. 957–964.

[7] Beyeler, A., J.-C. Zufferey, and D. Floreano (2009) "Vision-based control of near-obstacle flight," *Autonomous robots*, **27**(3), pp. 201–219.

[8] Zufferey, J.-C., A. Beyeler, and D. Floreano (2008) "Near-obstacle flight with small UAVs," in *UAV″ 2008*, LIS-CONF-2008-020, Springer Verlag, Orlando, FL.

[9] Fraundorfer, F., L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys (2012) "Vision-based autonomous mapping and exploration using a quadrotor MAV," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, IEEE, Vilamoura, Portugal, pp. 4557–4564.

[10] Barry, A. J. and R. Tedrake (2015) "Pushbroom stereo for high-speed navigation in cluttered environments," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, IEEE, Seattle, WA, pp. 3046–3052.

[11] Barry, A. J., H. Oleynikova, D. Honegger, M. Pollefeys, and R. Tedrake (2015) "FPGA vs. Pushbroom Stereo Vision for MAVs," in *Vision-based Control and Navigation of Small Lightweight UAVs, IROS Workshop*, Hamburg, Germany.

[12] Hrabar, S., G. S. Sukhatme, P. Corke, K. Usher, and J. Roberts (2005) "Combined optic-flow and stereo-based navigation of urban canyons for a UAV," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, IEEE, Edmonton, AB, Canada, pp. 3309–3316.

[13] Elfes, A. (1987) "Sonar-based real-world mapping and navigation," *Robotics and Automation, IEEE Journal of*, **3**(3), pp. 249–265.

[14] ——— (2013) "Occupancy grids: A stochastic spatial representation for active robot perception," *arXiv preprint arXiv:1304.1098*.

[15] Hart, P. E., N. J. Nilsson, and B. Raphael (1968) "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, **4**(2), pp. 100–107.

[16] Stentz, A. et al. (1995) "The Focussed D* Algorithm for Real-Time Replanning." in *IJCAI*, vol. 95, pp. 1652–1659.

[17] Ferguson, D. and A. Stentz (2006) "Using interpolation to improve path planning: The Field D* algorithm," *Journal of Field Robotics*, **23**(2), pp. 79–101.

[18] Lavalle, S. M. (1998) *Rapidly-Exploring Random Trees: A New Tool for Path Planning*, *Tech. rep.*

[19] Kuffner, J. J. and S. M. LaValle (2000) "RRT-connect: An efficient approach to single-query path planning," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 2, IEEE, San Francisco, CA, pp. 995–1001.

[20] Borenstein, J. and Y. Koren (1989) "Real-time obstacle avoidance for fast mobile robots," *Systems, Man and Cybernetics, IEEE Transactions on*, **19**(5), pp. 1179–1187.

[21] Koren, Y. and J. Borenstein (1991) "Potential field methods and their inherent limitations for mobile robot navigation," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, IEEE, Sacramento, CA, pp. 1398–1404.

[22] Borenstein, J. and Y. Koren (1991) "The vector field histogram-fast obstacle avoidance for mobile robots," *Robotics and Automation, IEEE Transactions on*, **7**(3), pp. 278–288.

[23] Thrun, S., W. Burgard, and D. Fox (2005) *Probabilistic robotics*, MIT press, pp. 284–304.

[24] Marlow, S. Q. and J. W. Langelaan (2010) "Dynamically sized occupancy grids for obstacle avoidance," in *AIAA Guidance, Navigation and Control Conference*, AIAA Paper 2010-7848, Toronto, Canada.

[25] Trucco, E. and A. Verri (1998) *Introductory techniques for 3-D computer vision*, vol. 201, Prentice Hall Englewood Cliffs, pp. 139–175.