AIAA SciTech Forum
9 - 13 January 2017, Grapevine, Texas
AIAA Guidance, Navigation, and Control Conference

A Quasi-polar Local Occupancy Grid Approach for Vision-based Obstacle Avoidance

Junyi Geng*, Jack W. Langelaan†
Department of Aerospace Engineering
The Pennsylvania State University, University Park, PA, 16802

This paper proposes a quasi-polar local (turn rate-time) 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. A numerical method for motion updates that can cope with the differing 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. Three exteroceptive sensor models (wide-field monocular vision, pushbroom stereo, and pushbroom stereo combined with wide field monocular) are presented in this context. Simulations of flight through a two dimensional environment consisting of both forest and urban terrain are used to demonstrate the utility of this approach.

I. Introduction

Small unmanned aerial vehicles (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 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, bird 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 shows 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.

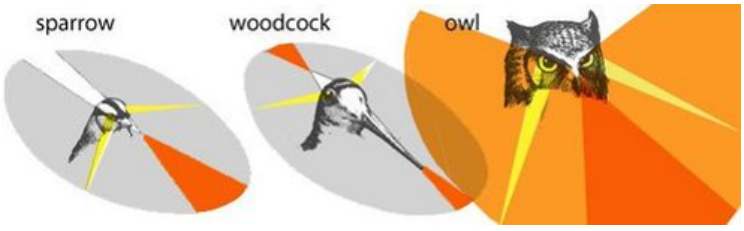


Figure 1. Bird vision [2]

Vision based navigation methods have been popular recently. The functional monocular vision based UAV PTAM (parallel tracking and mapping) proposed in [3] 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

*Research assistant of Aerospace Engineering, Student Member AIAA.
†Associate Professor of Aerospace Engineering, Associate Fellow AIAA.

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 [4, 5]. 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 [6]. In [7, 8], 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 [9]. But this approach still requires two pair of cameras. One pair of stereo cameras and another pair of fish eye cameras for optical flow. 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.

Occupancy grids are a powerful mapping tool, first introduced by Elfes [10, 11]. 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. A typical occupancy grid deals with obstacle mapping in a spatial lattice, usually under the Cartesian coordinate on a global dimension. Variations of occupancy grid have been explored by researchers. In [12], 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. [13] presents a new occupancy grid tracking solution based on particles for tracking the dynamic driving environment. The particles have a dual nature: one denote 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 [14].

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

This paper proposes a novel quasi-polar local occupancy grid 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. To demonstrate the effectiveness of this approach, results of two-dimensional Monte Carlo simulation are presented. The main contributions of this paper are the novel quasi-polar local occupancy grid based on pre-calculated paths, by which the path selection work is convenient; the combination of different camera sensor models, which takes advantage of the monocular and stereo; the performance verification through simulations.

The remainder of this paper is organized as follows. In section II, a description of the navigation problem is given.

Section III and IV define the models for vehicle kinematics and several camera sensors. An estimator design which describes the quasi-polar local occupancy grid definition and mathematical implementation are presented in Section V. Section VI designs the new path selection method based on the pre-calculated path set. All the ideas are brought together in the description of simulation platform and the simulation results are shown in Section VII.

II. System and Problem Review

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 main goal is obstacle avoidance: while this paper briefly addresses navigation to a particular destination, it is not a major focus. The block diagram in Fig. 2 shows a system that uses the given sensors to perform obstacle avoidance.

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 the local quasi-polar frame. Note that this quasi-polar frame is based on flight paths that would result from 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.

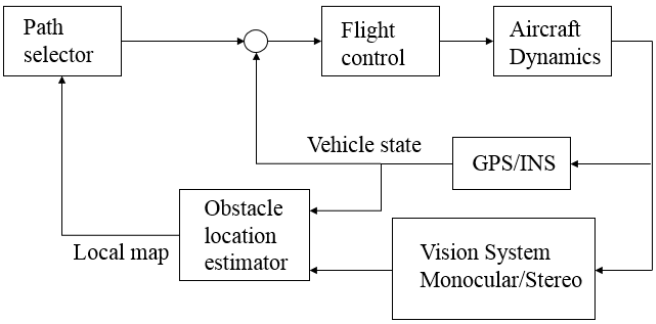


Figure 2. System block diagram

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

$$\mathbf{x} = \begin{bmatrix} x & y & \psi & u & v \end{bmatrix}^T \tag{1}$$

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

$$p(\mathbf{m}|\mathbf{x}_{1,\cdots,t}, \mathbf{z}_{1,\cdots,t}) \tag{2}$$

where the measurement model

$$\mathbf{z}_i = g(\mathbf{x}_i, \mathbf{m}) \tag{3}$$

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.

III. System Models

A. Coordinate frames

Referring to Fig. 3, 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. 4) 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.

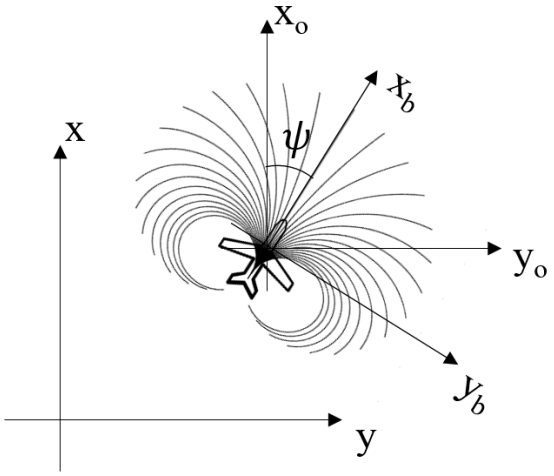


Figure 3. Coordinate frames

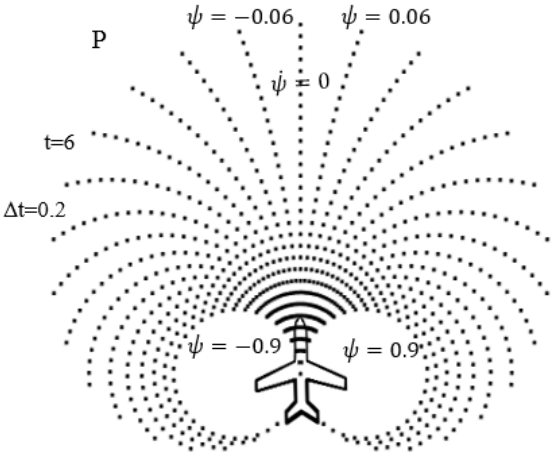


Figure 4. Quasi-polar coordinate

B. 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

$$\dot{x} = u\cos\psi - v\sin\psi \tag{4}$$

$$\dot{y} = u\sin\psi + v\cos\psi \tag{5}$$

$$\dot{\psi} = \omega + \mathcal{N}(0, \sigma_\omega^2) \tag{6}$$

$$\dot{u} = a_x + \mathcal{N}(0, \sigma_{a_x}^2) \tag{7}$$

$$\dot{v} = a_y + \mathcal{N}(0, \sigma_{a_y}^2) \tag{8}$$

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 $a_y = 0$.

C. 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.

IV. Sensor Models

This section details the vision model used to get the measurement. Three sensor models are described: monocular vision (which provides bearings and bearing rates); a general stereo model (where two cameras share a region of overlapping field of view as well as maintaining a non-overlapping field of view); and a combination that uses a "pushbroom stereo" combined with monocular vision.

A. Monocular Vision

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. 5.

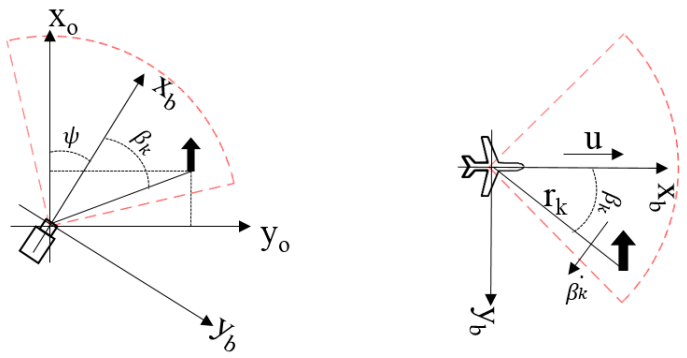


Figure 5. Monocular vision model

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

$$\beta_k = \arctan\frac{y_{k/o}}{x_{k/o}} - \psi \tag{9}$$

The bearing rate is computed by taking the time derivative of Eq. 9:

$$\dot{\beta}_k = \frac{u\sin\beta_k}{r_k} - \frac{v\cos\beta_k}{r_k} - \dot{\psi} \tag{10}$$

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 [14]. 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

$$\mathbf{z} = \begin{bmatrix} u & v & \dot{\psi} & \beta^T & \dot{\beta}^T \end{bmatrix}^T + \mathcal{N}(0, \Sigma_Z) \tag{11}$$

$$\Sigma_z = \text{diag}\begin{bmatrix} \sigma_u^2 & \sigma_v^2 & \sigma_{\dot{\psi}}^2 & \Sigma_\beta^2 & \Sigma_{\dot{\beta}}^2 \end{bmatrix} \tag{12}$$

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.

$$\beta = \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_N \end{bmatrix}^T \tag{13}$$

$$\dot{\beta} = \begin{bmatrix} \dot{\beta}_1 & \dot{\beta}_2 & \cdots & \dot{\beta}_N \end{bmatrix}^T \tag{14}$$

$$\Sigma_\beta = \sigma_\beta^2 \mathbf{I}_N \tag{15}$$

$$\Sigma_{\dot{\beta}} = \sigma_{\dot{\beta}}^2 \mathbf{I}_N \tag{16}$$

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

2. *Range model*

Range r_k^* can be computed from Eq. 10:

$$r_k^* = \frac{1}{\dot{\beta}_k + \dot{\psi}}(u \sin \beta_k - v \cos \beta_k) \tag{17}$$

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. 17 is computed for the uncertainty of range estimate:

$$\nabla_{r_k^*} = \begin{bmatrix} \frac{\partial r_k^*}{\partial u} & \frac{\partial r_k^*}{\partial v} & \frac{\partial r_k^*}{\partial \dot{\psi}} & \frac{\partial r_k^*}{\partial \beta_k} & \frac{\partial r_k^*}{\partial \dot{\beta}_k} \end{bmatrix} \tag{18}$$

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

$$\begin{aligned} \sigma_{r_k^*}^2 = {} & (\dot{\beta}_k + \dot{\psi})^{-2}[\sigma_u^2 \sin^2 \beta_k + \sigma_v^2 \cos^2 \beta_k + \sigma_\beta^2 (u \cos \beta_k + v \sin \beta_k)^2] \\ & + (\dot{\beta}_k + \dot{\psi})^{-4}[(\sigma_{\dot{\beta}}^2 + \sigma_{\dot{\psi}}^2)(u \sin \beta_k - v \cos \beta_k)^2] \end{aligned} \tag{19}$$

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 [14]. 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. 17 and Eq. 19 are used to compute r^* and σ_{r^*} [14].

B. General Stereo Model

The general stereo vision consists of two cameras mounted as in Fig. 6. 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.

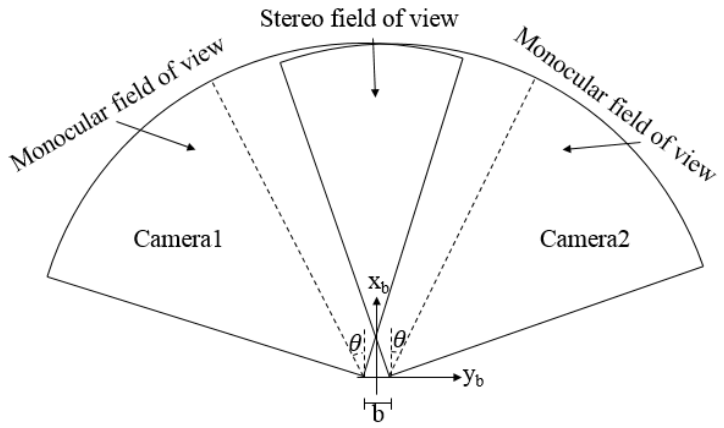


Figure 6. General stereo

Consider the right camera. If the k^{th} 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,

$$x_{b1} = x_{k/o}\cos\psi + y_{k/o}\sin\psi \tag{20}$$

$$y_{b1} = -x_{k/o}\sin\psi + y_{k/o}\cos\psi \tag{21}$$

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

$$x_{b2} = x_{k/o}\cos\psi + y_{k/o}\sin\psi \tag{22}$$

$$y_{b2} = -x_{k/o}\sin\psi + y_{k/o}\cos\psi - \frac{b}{2} \tag{23}$$

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

$$x_c = x_{b2}\cos\theta + y_{b2}\sin\theta \tag{24}$$

$$y_c = -x_{b2}\sin\theta + y_{b2}\cos\theta \tag{25}$$

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

$$\beta_k = \arctan\frac{y_c}{x_c} = \frac{-x_{b2}\sin\theta + y_{b2}\cos\theta}{x_{b2}\cos\theta + y_{b2}\sin\theta} \tag{26}$$

The bearing rate is computed by taking the time derivative of Eq. 26:

$$\dot{\beta}_k = \frac{\sin\beta_k}{r_{kc}}(u\cos\theta + v\sin\theta) - \frac{\cos\beta_k}{r_{kc}}(v\cos\theta - u\sin\theta) - \dot{\psi} \tag{27}$$

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. 27 is solved for range r (since the baseline is small, $r_k \approx r_{kc}$):

$$r_k^* = \frac{(u\cos\theta + v\sin\theta)\sin\beta_k - (v\cos\theta - u\sin\theta)\cos\beta_k}{\dot{\beta}_k + \dot{\psi}} \tag{28}$$

The Jacobian of Eq. 28 is computed for the uncertainty in range estimate:

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

So,

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

Comparing Eqs. 9, 10 and 17 with Eqs. 26, 27 and 28, 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.

Note that when $\theta = 0°$, general stereo becomes the simple stereo case. As shown in Fig. 7, two cameras face forward with the optical axes parallel.

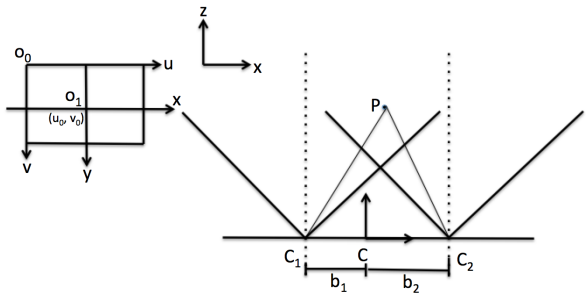


Figure 7. Simple stereo

1. *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. 8. 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, [23]. For the simple stereo case, Fig. 9

$$\frac{T + x_l - x_r}{Z - f} = \frac{T}{Z} \tag{29}$$

Solving Eq. 29 for Z we obtain

$$Z = f\frac{T}{D} \tag{30}$$

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.

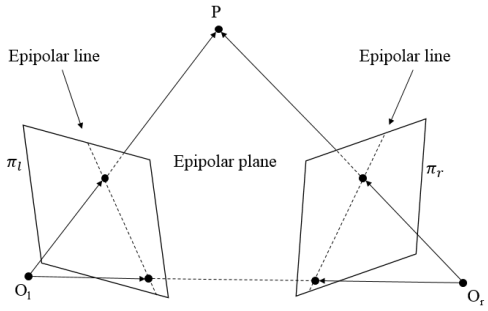


Figure 8. The epipolar geometry

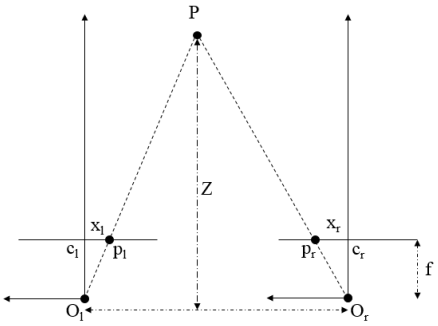


Figure 9. Depth estimate for simple stereo

Since for each block in the left image, the standard stereo needs to search along the whole epipolar line, or search through depth, Eq. 30, 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 [7]. Limiting the search through distance to a single value, d meters away, which also means a single disparity, processing time can be significantly reduced. 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 [7] since the cameras are on a moving platform (aircraft or hexcopter).

In summary, at each sensor update, only the obstacles at the depth d from the camera can be seen. One can choose the distance d up to the maximum resolvable distance for a given stereo baseline: choice of d may also depend on vehicle maneuverability constraints.

C. 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. 10. 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 mentioned 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.

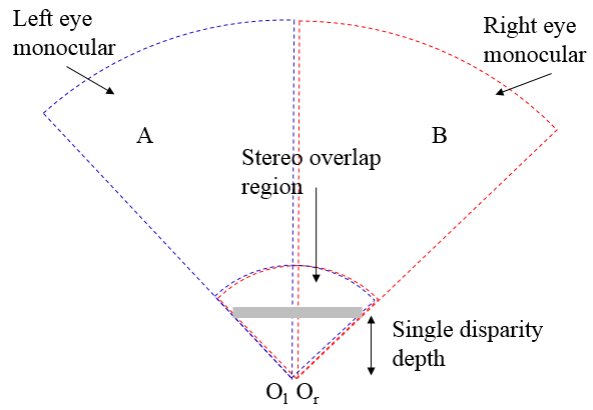


Figure 10. Combined sensor

In this case one does not attempt to correlate features seen by both cameras outside of the single disparity depth. Although one can obtain some additional information, the computational cost of performing the correlation outweighs the quality of the information that can be obtained.

In general stereo model of B, 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).

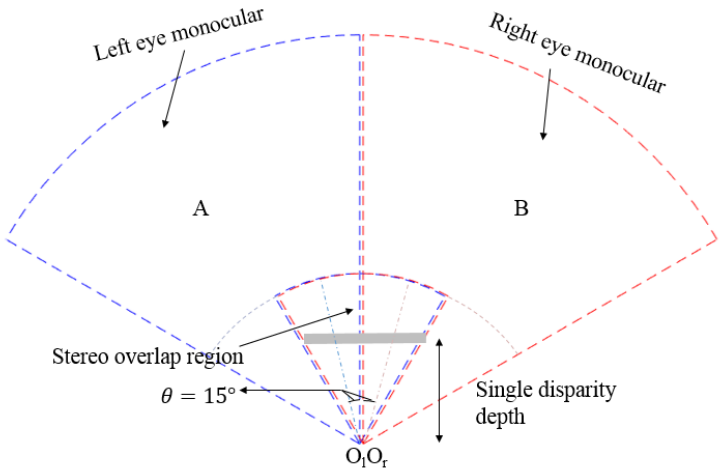


Figure 11. "Bird eyes" combined sensor

For example, suppose both two cameras have $90°$ field of view, and $\theta = 15°$, the bird eye model will be given like Fig. 11. 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.

V. Estimator Design

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. 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.

The occupancy grid computes the likelihood that discrete environment cells are occupied by obstacles (Fig. 12). Since each cell in the environment is considered independently, computational cost is fairly low and complex obstacle shapes can be modeled.

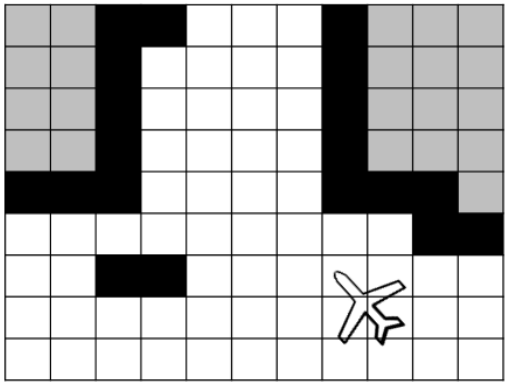


Figure 12. Schematic of occupancy grid. Occupied cells are shown in black, free cells in white, unknown cells in grey.

Occupancy grids have been well documented (see for example Thrun [24]). 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

$$p(\mathbf{m}|z_{1,\cdots,t}, x_{1,\cdots,t}) = \prod_{i,j} p(m_{i,j}|z_{1,\cdots,t}, x_{1,\cdots,t}) \tag{31}$$

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.

$$p(m_{i,j}|z_{1,\cdots,t}, x_{1,\cdots,t}) = \frac{p(z_t|m_{i,j}, x_t)p(m_{i,j}|z_{1,\cdots,t-1}, x_{1,\cdots,t-1})}{p(z_t|z_{1,\cdots,t-1}, x_{1,\cdots,t})} \tag{32}$$

Define odds and log-odds as

$$\begin{aligned} o(m_{i,j}|z_{1,\cdots,t}, x_{1,\cdots,t}) &= \tfrac{p(m_{i,j}|z_{1,\cdots,t}, x_{1,\cdots,t})}{1-p(m_{i,j}|z_{1,\cdots,t}, x_{1,\cdots,t})} \\ l_{t,ij} &= \log o(m_{i,j}|z_{1,\cdots,t}, x_{1,\cdots,t}) \end{aligned} \tag{33}$$

Using independence assumption and converting the map probability into log-odds form, Eq. 32 becomes

$$l_{t,ij} = \ln \frac{p(m_{i,j}|z_t, x_t)}{1 - p(m_{i,j}|z_t, x_t)} + l_{t-1,ij} \tag{34}$$

In Eq. 34, 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:

$$l_{t,ij} = \sum_{n}^{N} \ln \frac{p(m_{i,j}|z_{t,n}, x_t)}{1 - p(m_{i,j}|z_{t,n}, x_t)} + l_{t-1,ij} \tag{35}$$

The second term on the right hand side of Eq. 35 is the accumulated log-odds of one occupancy grid cell over all previous time steps.

By solving Eq. 33, the probability of occupancy can be recovered from the log-odds:

$$p(m_{i,j}|z_{1,\cdots,t}, x_{1,\cdots,t}) = \frac{e^{l_{t,ij}}}{1 + e^{l_{t,ij}}} \tag{36}$$

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

A. 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 [14]. 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 [25].

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.

1. Definition and coordinate transformation

As defined in Sec. III, 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. 13 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.

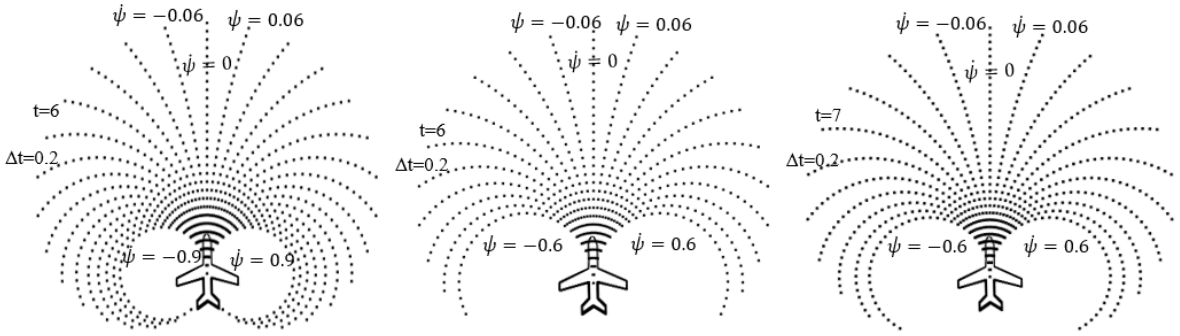


Figure 13. Comparison of quasi-polar occupancy grid. Left: Grid shape with $|\psi|_{max} = 0.9$rad/sec, $t_{max} = 6$sec; Middle: Grid shape with $|\psi|_{max} = 0.6$rad/sec, $t_{max} = 6$sec; Right: Grid shape with $|\psi|_{max} = 0.6$rad/sec, $t_{max} = 7$sec

The path time can be chosen as

$$t_{max} = \frac{s}{u} \tag{37}$$

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.

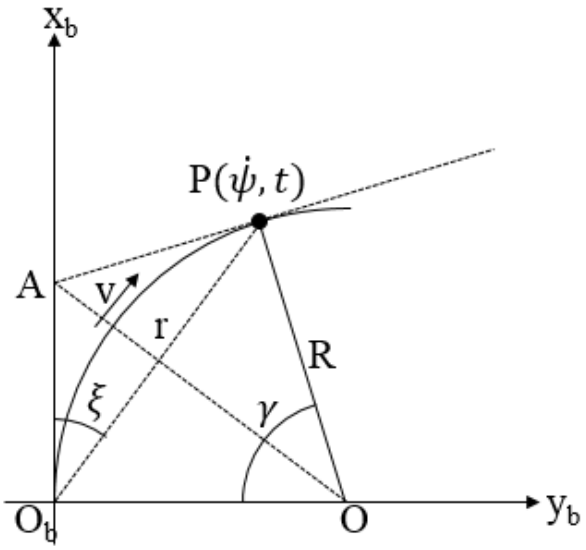


Figure 14. Pre-calculated path

As Fig. 14 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 $\widehat{O_bP}$:

$$\widehat{O_bP} = vt = R\gamma \tag{38}$$

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

$$2R = \frac{|O_bP|}{\sin\xi} \tag{39}$$

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

$$\dot{\psi} = \frac{\gamma}{t} = \frac{v}{R} \tag{40}$$

$$t = \frac{R\gamma}{v} \tag{41}$$

Put R, γ into t and make simplification,

$$|O_bP| = \frac{2v\sin\xi}{\dot{\psi}} \tag{42}$$

$$t = \frac{\xi|O_bP|}{\sin\xi v} = \frac{2\xi}{\dot{\psi}} \tag{43}$$

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

$$r = |O_bP| = \frac{2v\sin\xi}{\dot{\psi}} \tag{44}$$

$$\xi = \frac{t\dot{\psi}}{2} \tag{45}$$

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. *Obstacle mapping*

From the several sensor models in Sec. IV, 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). Using Eq. 44, the corresponding coordinate in quasi-polar local occupancy grid frame can be obtained once the coordinate in polar frame is known.

$$\dot{\psi} = \frac{2v}{r} \sin \beta \tag{46}$$

$$t = \frac{\beta r}{\sin \beta v} = \frac{2\beta}{\dot{\psi}} \tag{47}$$

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.

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.

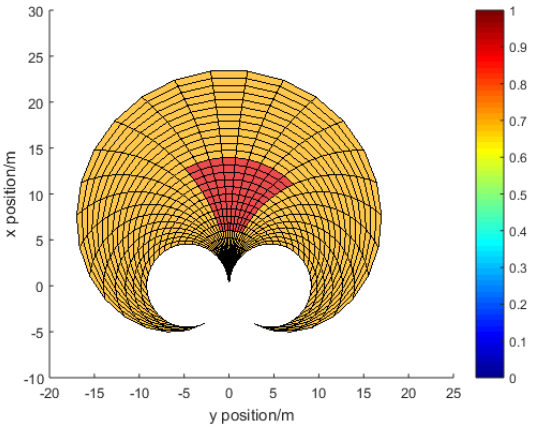


Figure 15. Initial grid

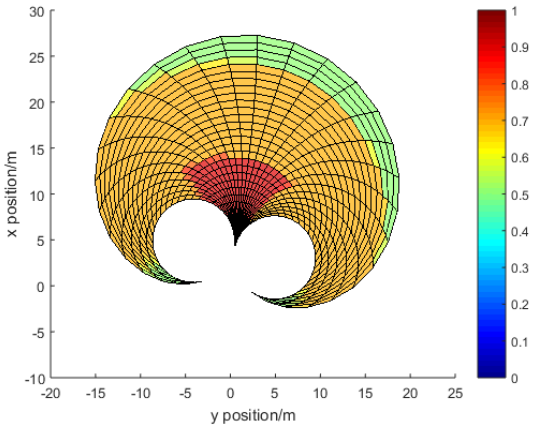


Figure 16. Grid after motion update

- 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. 15, Fig. 16. The vehicle is located at (0,0) with 0° heading angle, Fig. 15. 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. 16. 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.

B. Inverse sensor model

The inverse sensor model compute the probability of a cell being occupied for a single measurement. It's used to updated the occupancy grid when the measurements are obtained as shown in Sec. V Eq. 34. From the four kinds of sensors given in Sec. IV, an estimated range to the obstacle is computed, thus the obstacle location in polar coordinates is known. So 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.

1. Range

In Sec. IV, 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 paper, 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.

$$p(r) = a_1(1 + \exp(\frac{2\pi(r - r^* + 2\sigma_{r^*})}{\sqrt{3}\sigma_{r^*}}))^{-1} + a_2 \exp(-\frac{(r - r^*)^2}{2\sigma_{r^*}^2}) \tag{48}$$

FAR AWAY OBSTACLES In monocular sensor model, when obstacles are far away or not present, $\dot{\beta} + \dot{\psi} \approx 0$, Eq. 10 will yield $r^* \approx \infty$, the uncertainty σ_{r*} will also be ∞, Eq. 19. 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^*}.

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. 10. 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.

LOG-ODDS COMPUTATION CONCERN The inverse sensor model must be in log odds form to be added to the previous log-odds belief, Eq. 35. 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. 48.

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

$$f_n(r) = -c_1(1 + \exp(\frac{2\pi(r - r_n^* + 2\sigma_{r_n^*})}{\sqrt{3}\sigma_{r_n^*}}))^{-1} + \frac{c_2}{\sigma_{r_n^*}\sqrt{2\pi}} \exp(-\frac{(r - r_n^*)^2}{2\sigma_{r_n^*}^2}) \tag{49}$$

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.

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.

$$g_n(\xi) = (1 + \exp \frac{c_3(|\xi - \beta_n| - 0.5 \triangle \beta - 1.25\sigma_\psi)}{\sigma_\psi})^{-1} \tag{50}$$

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 $(\psi_{i,j}, t_{i,j})$, the coordinate in body polar frame can be obtained from Eq. 44, the log-odds of one quasi-polar local occupancy grid cell induced by a measurement of nth obstacle feature points can be computed by:

$$\ln \frac{p(m_{i,j}|z_{t,n}, x_t)}{1 - p(m_{i,j}|z_{t,n}, x_t)} = f_n(r_{i,j}) g_n(\xi_{i,j}) \tag{51}$$

An example is shown in Fig. 17. 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.

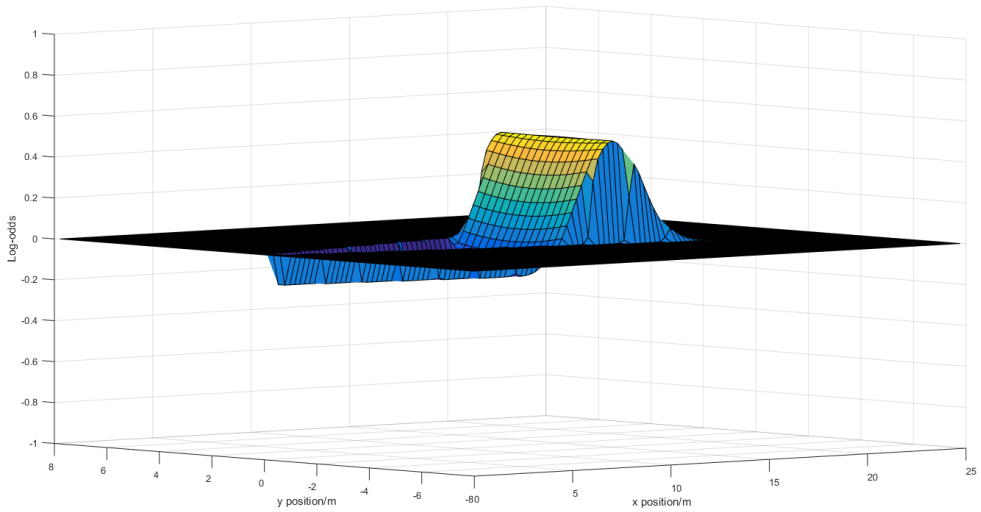


Figure 17. Example of the Inverse Sensor Model in log-odds form

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

VI. Path Selection

The path selector selects a desired turn rate for the vehicle from the pre-calculated path set. Then the controller uses the desired heading to compute to control inputs. Here assume the controller is 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.

A. 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 the purpose is only escape 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:

$$W_i = w_1 P_i + w_2 \Delta\psi_i + w_3 \Delta r_i \tag{52}$$

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.

B. 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 push-broom 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 selecting path. However, choosing threshold is another challenging issue. A too large threshold will make the path selector lose the ability to avoid small obstacles while a too small threshold will result an over sensitive path selector, which causes the vehicle fail to escape.

VII. Simulation

A. 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.

1. Environment model

The environment is made up of several buildings and a surrounding forest, see Fig. 18. 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. 18. 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. 18). To prevent the vehicle starting too close to an obstacle, the start locations are defined to be at least 6 m from a tree.

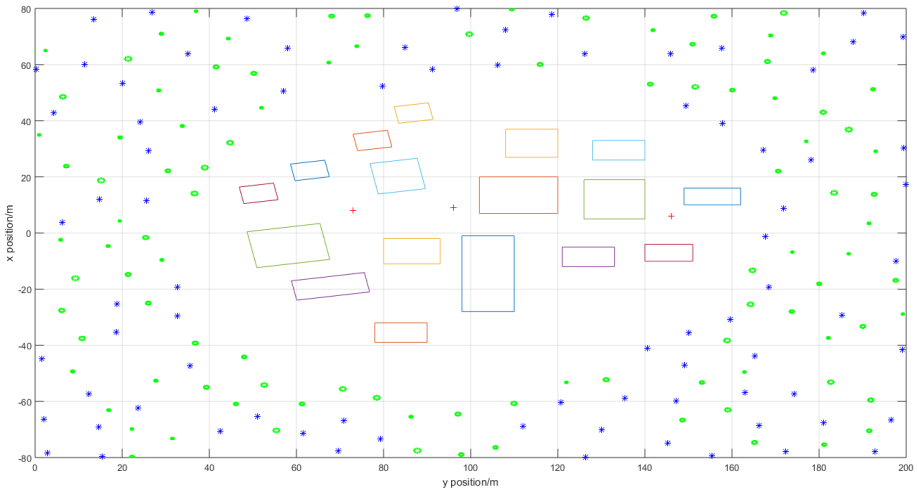


Figure 18. Simulation map. Polygons represent buildings, green dots circle represent trees. The blue stars are 80 different starting locations and the red crosses are the starting/goal locations.

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.

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.

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 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.

4. *Camera models*

Each scenario will be tested with three camera models:

- Case I: Monocular vision with $120°$ 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, valid depth set to 5 m and $90°$ field of view.
- Case III: Combined model ('bird eye') with $120°$ monocular field of view, $60°$ pushbroom stereo field of view.

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.

6. *Parameters in system*

The parameters used for standard deviation are given in Table 1. The noises are assumed to be Gaussian.

Table 1. Standard Deviation for Noises

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

The constant values $c_1 - c_3$ are for inverse sensor model, and $w_1 - w_3$ for path selector are given in Table 2.

Table 2. Constants used for system

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

B. Results

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 3, 4, 5.

Results for the wide field of view monocular vision system are summarized in Table 3 and shown in Fig. 19.

Table 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)

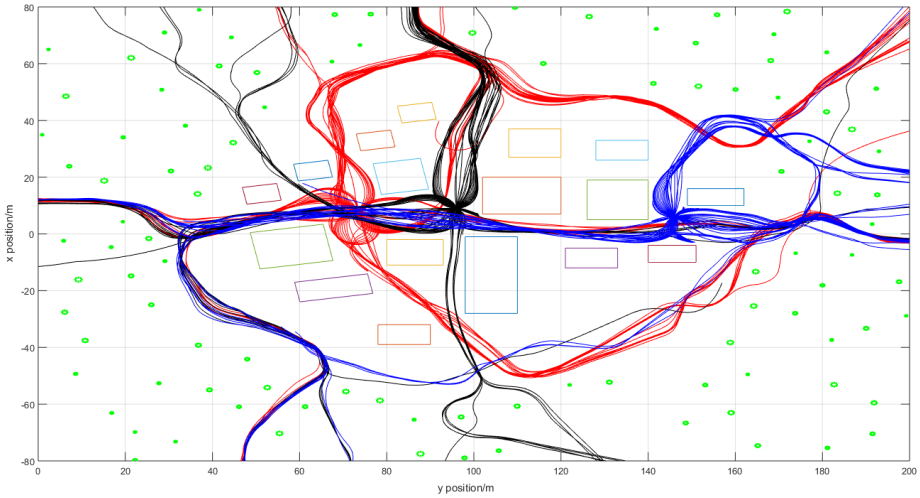


Figure 19. Results of escaping using sensor I. Red lines are the result paths from start location 1.Black lines are the result paths from start location 2. Blue lines are the result paths from start location 3.

Table 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 4 and shown in Fig. 20a and Fig. 20b.

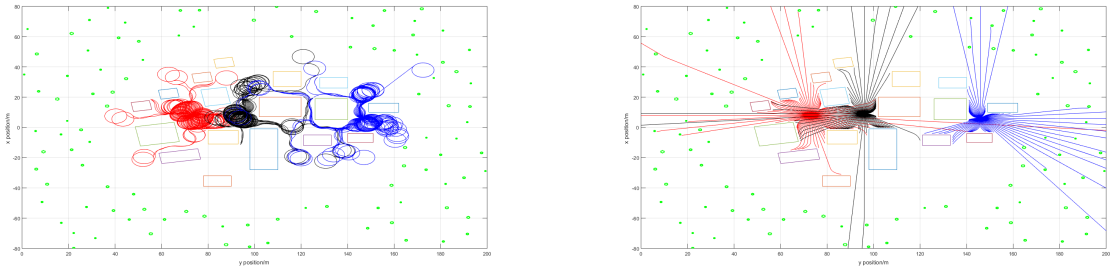
For the pushbroom stereo, the "integrated path free probability" (path selector I) is generally larger for the high-turn rate paths, leading to circular flight (and thus failures due to DNF). This can be explained that the single disparity method only detects obstacles in the same depth instead of same 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 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 and shown in Fig. 21. 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.



a) Path selector I

b) Path selector II

Figure 20. Results of escaping using sensor 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.

Table 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)

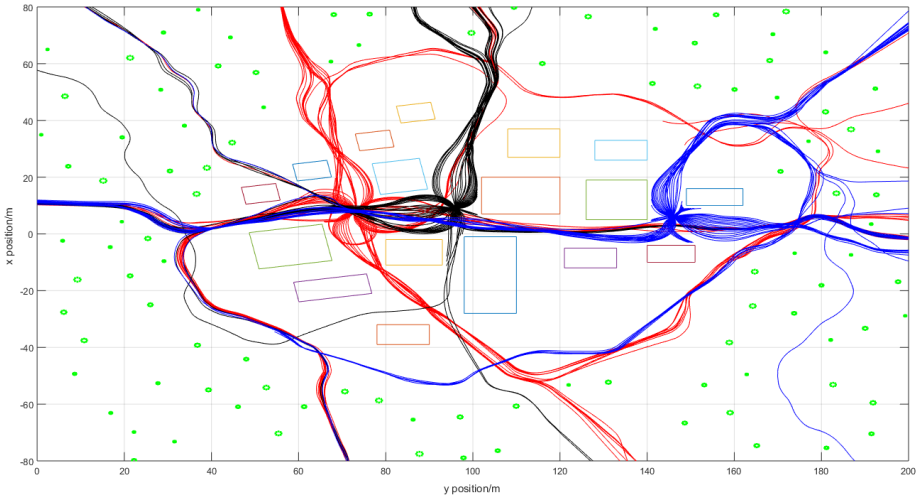


Figure 21. Results of escaping using sensor III. Red lines are the result paths from start location 1.Black lines are the result paths from start location 2. Blue lines are the result paths from start location 3.

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).

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. 22 and Fig. 23 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.

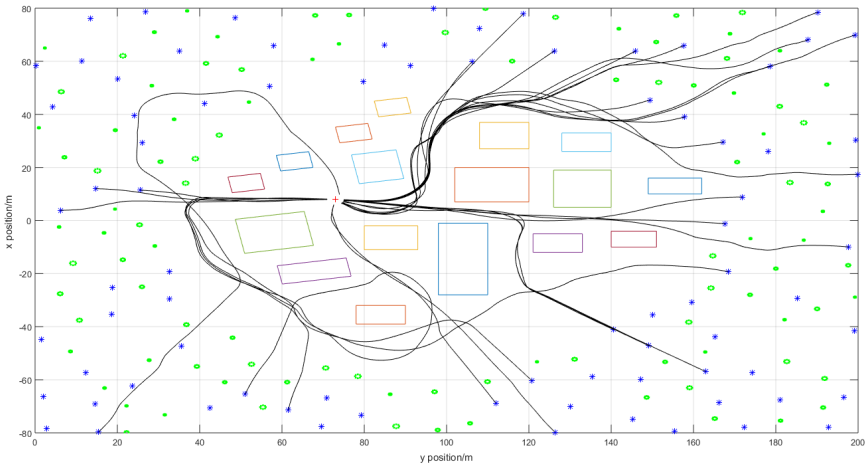


Figure 22. Flight path from select simulations using sensor I. The black lines here show the paths of simulations which successfully navigate to the goal.

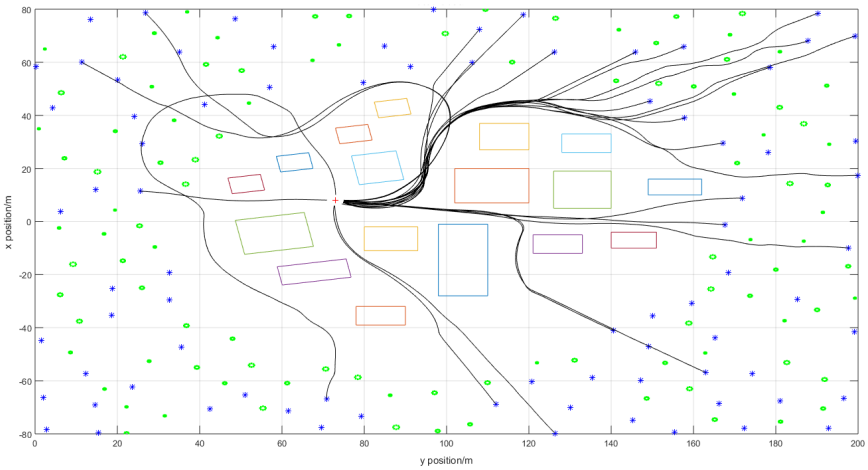


Figure 23. Flight path from select simulations using sensor III. Again, the black lines here show the paths of simulations which successfully navigate to the goal.

Fig. 24 and Fig. 25 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.

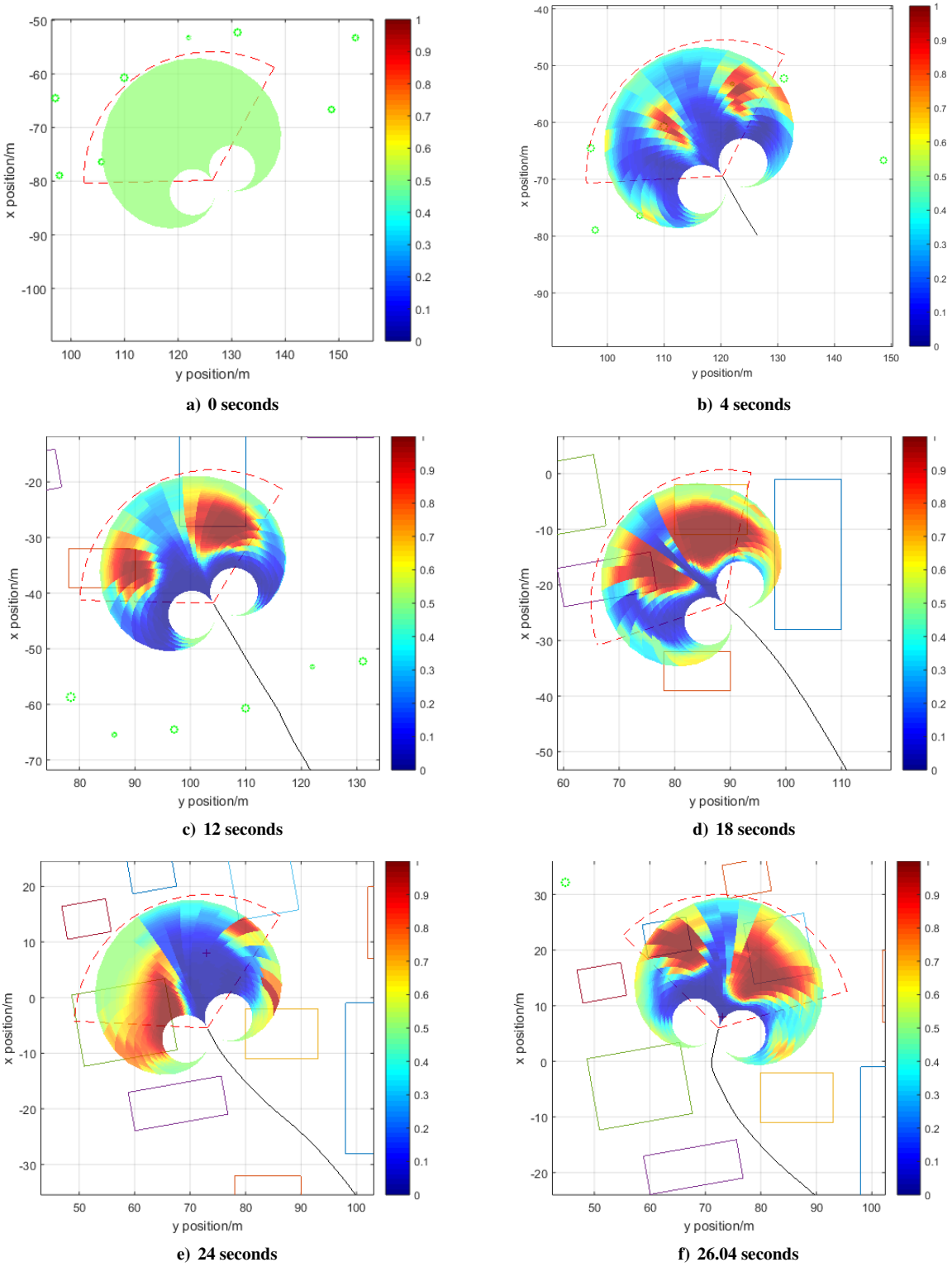
VIII. Conclusion

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.



a) 0 seconds

b) 4 seconds

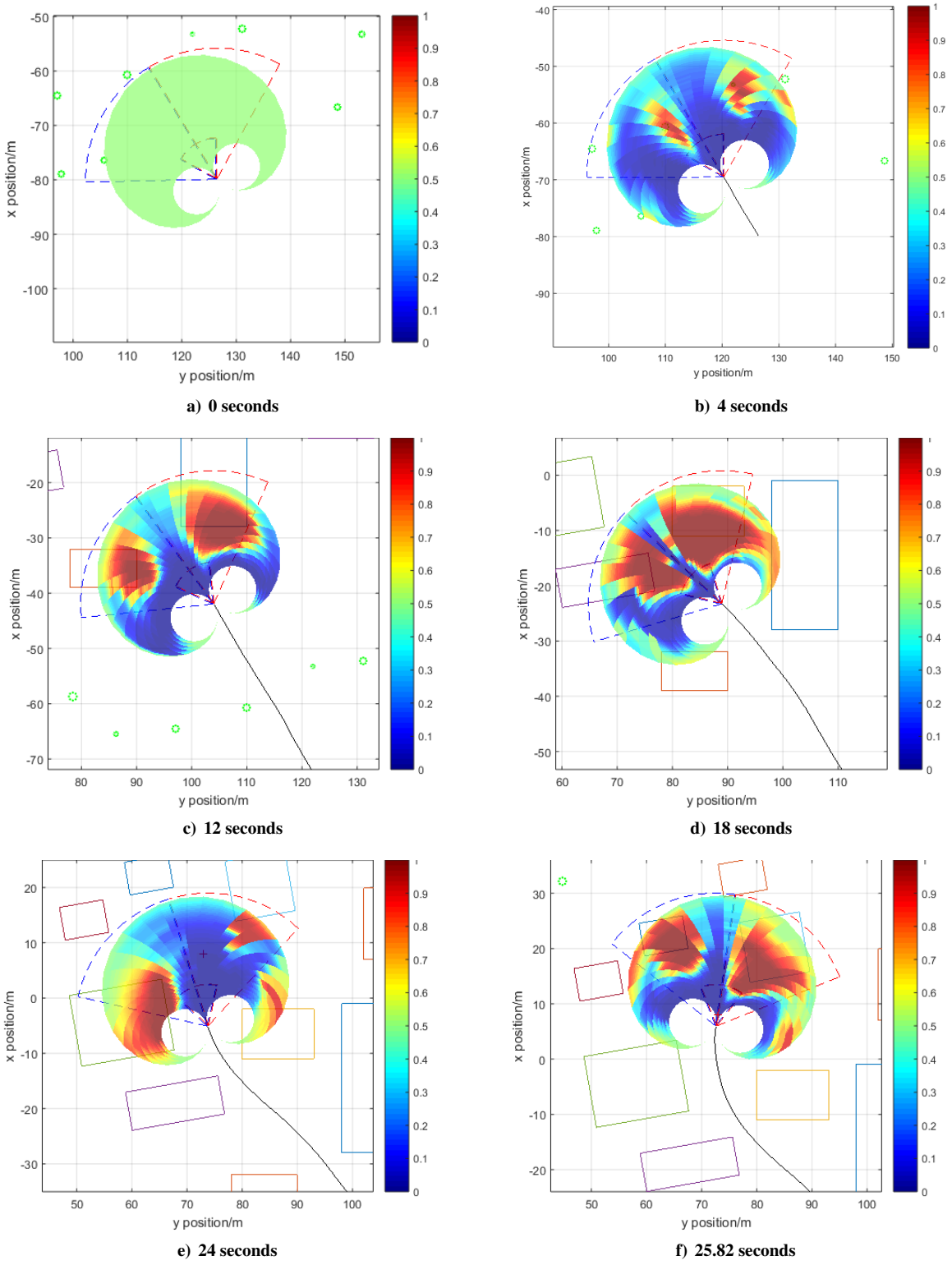
c) 12 seconds

d) 18 seconds

e) 24 seconds

f) 26.04 seconds

Figure 24. 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 25. 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.

References

[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] Weiss, S., Achtelik, M. W., Lynen, S., Chli, M., and Siegwart, R., "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, IEEE, St Paul, MN, May 14-18 2012, pp. 957–964.

[4] Beyeler, A., Zufferey, J.-C., and Floreano, D., "Vision-based control of near-obstacle flight," *Autonomous robots*, Vol. 27, No. 3, 2009, pp. 201–219.

[5] Zufferey, J.-C., Beyeler, A., and Floreano, D., "Near-obstacle flight with small UAVs," *UAV" 2008*, No. LIS-CONF-2008-020, Springer Verlag, Orlando, FL, 2008.

[6] Fraundorfer, F., Heng, L., Honegger, D., Lee, G. H., Meier, L., Tanskanen, P., and Pollefeys, M., "Vision-based autonomous mapping and exploration using a quadrotor MAV," *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, IEEE, Vilamoura, Portugal, Oct 7-12 2012, pp. 4557–4564.

[7] Barry, A. J. and Tedrake, R., "Pushbroom stereo for high-speed navigation in cluttered environments," *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, IEEE, Seattle, WA, May 26-30 2015, pp. 3046–3052.

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

[9] Hrabar, S., Sukhatme, G. S., Corke, P., Usher, K., and Roberts, J., "Combined optic-flow and stereo-based navigation of urban canyons for a UAV," *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, IEEE, Edmonton, AB, Canada, Aug 2-6 2005, pp. 3309–3316.

[10] Elfes, A., "Sonar-based real-world mapping and navigation," *Robotics and Automation, IEEE Journal of*, Vol. 3, No. 3, 1987, pp. 249–265.

[11] Elfes, A., "Occupancy grids: A stochastic spatial representation for active robot perception," *arXiv preprint arXiv:1304.1098*, 2013.

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

[13] Danescu, R., Oniga, F., and Nedevschi, S., "Modeling and tracking the driving environment with a particle-based occupancy grid," *Intelligent Transportation Systems, IEEE Transactions on*, Vol. 12, No. 4, 2011, pp. 1331–1342.

[14] Marlow, S. Q. and Langelaan, J. W., "Local terrain mapping for obstacle avoidance using monocular vision," *Journal of the American Helicopter Society*, Vol. 56, No. 2, 2011, pp. 22007–22007.

[15] Hart, P. E., Nilsson, N. J., and Raphael, B., "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, Vol. 4, No. 2, 1968, pp. 100–107.

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

[17] Ferguson, D. and Stentz, A., "Using interpolation to improve path planning: The Field D* algorithm," *Journal of Field Robotics*, Vol. 23, No. 2, 2006, pp. 79–101.

[18] Lavalle, S. M., "Rapidly-Exploring Random Trees: A New Tool for Path Planning," , 1998.

[19] Kuffner, J. J. and LaValle, S. M., "RRT-connect: An efficient approach to single-query path planning," *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, Vol. 2, IEEE, San Francisco, CA, Apr 24-28 2000, pp. 995–1001.

[20] Borenstein, J. and Koren, Y., "Real-time obstacle avoidance for fast mobile robots," *Systems, Man and Cybernetics, IEEE Transactions on*, Vol. 19, No. 5, 1989, pp. 1179–1187.

[21] Koren, Y. and Borenstein, J., "Potential field methods and their inherent limitations for mobile robot navigation," *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, IEEE, Sacramento, CA, Apr 9-11 1991, pp. 1398–1404.

[22] Borenstein, J. and Koren, Y., "The vector field histogram-fast obstacle avoidance for mobile robots," *Robotics and Automation, IEEE Transactions on*, Vol. 7, No. 3, 1991, pp. 278–288.

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

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

[25] Marlow, S. Q. and Langelaan, J. W., "Dynamically sized occupancy grids for obstacle avoidance," *AIAA Guidance, Navigation and Control Conference*, No. AIAA Paper 2010-7848, Toronto, Canada, August 2-5 2010.