2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Vision-based local multi-resolution mapping and path planning for Miniature Air Vehicles

Huili Yu Randal W. Beard Jeffrey Byrne

Abstract— **Miniature Air Vehicles (MAVs) are often used for low altitude flights where unknown obstacles might be encountered. Path planning and obstacle avoidance for MAVs involve planning a feasible path from an initial state to a goal state while avoiding obstacles in the environment. This paper presents a vision-based local multi-resolution mapping and path planning technique for MAVs using a forward-looking onboard camera. A depth map, which represents the time-to-collision (TTC) and bearing information of the obstacles, is obtained by computer vision algorithms. To account for measurement uncertainties introduced by the camera, a multi-resolution map in the body frame of the MAV is created in polar coordinates. Using the depth map, the locations of the obstacles are determined in the multi-resolution map. Dijkstra's algorithm is employed to find a collision-free path in the body frame. The simulation and flight test results show that the proposed technique is successful in solving path planning and multiple obstacles avoidance problems for MAVs.**

I. Introduction

Miniature Air Vehicles (MAVs) have the potential to perform tasks that are too difficult or dangerous for human pilots. For example, they can monitor critical infrastructure and real-time disasters, perform search and rescue, and perform in-storm weather measurements [1]. For many of these applications, MAVs are required to navigate in urban or unknown terrains where obstacles of various types and sizes may hinder the success of the mission. MAVs must have the capability to autonomously plan paths that do not collide with buildings, trees or other obstacles. Therefore, the path planning and obstacle avoidance problems for MAVs have received significant attention [1][2].

The general framework for the MAV path planning problem can be described as follows: *given a description of the environment, find a feasible path between two configurations in the environment that does not result in a collision with any obstacles*. The path planning problem can be grouped into global path planning and local path planning. Global path planning requires complete knowledge about the environment and a static terrain. In that setting a feasible path from the start to the destination configuration is generated before the agent starts its motion [3]. Therefore, global path planning is pre-mission planning and can be formulated as an optimization problem. The global path planning problem has been addressed by many researchers with the two most common solutions being roadmap methods and cell decomposition methods [4]. On the other hand, local path planning is executed in real-time during the flight. The basic idea is to first sense the obstacles in the environment and then determine a collision-free path [1]. This paper focuses on solving the local path planning problem for an unknown environment.

Local path planning and obstacle avoidance problems have been studied in the robotics and Unmanned Air Vehicle (UAV) communities. Reference [3] develops a genetic algorithm based path planning scheme for local obstacle avoidance of a mobile robot. However, this method was only studied and implemented for known environments. Reference [5] develops a ladar designed for path planning and obstacle avoidance. Reference [6] develops a forward-looking ladar to assist a helicopter pilot in obstacle detection and path planning. A laser range finder is used in [7] to detect unknown obstacles. Sonar is utilized primarily for underwater autonomous systems [8]. Many of these sensors are too heavy to be used on a small UAV.

Alternatively, video cameras are lightweight and inexpensive and fit the physical requirements of small UAVs [1]. Reference [9] presents an obstacle detection methodology using feature tracking in a forward-looking onboard camera and develops a path planning algorithm for a small UAV. However, [9] did not consider the measurement uncertainties produced by the camera, which increase as the distance from the UAV increases. Therefore, information about the region close to the UAV is more reliable than information about the areas that are far away, thus motivating the use of multi-resolution mapping and path planning.

There is some work on multi-resolution path planning schemes for robots and UAVs. To reduce the computational cost, Reference [10] uses a quadtree decomposition of the environment to give finer resolution to the areas close to the boundaries of the obstacles and coarser resolution to the areas far from the obstacles. Even though it is efficient in many cases, this quadtree-based decomposition still tends to waste computational resources because the finer resolution is used to map the boundaries of all obstacles, regardless of their distance from the robot. It also fails to take full advantage of the local information around the robot. Reference [11] uses high resolution to represent the configuration space that is in close proximity to the robot and low resolution for regions far from the robot. This approach concentrates the planning resource at the beginning part of the path, where the information about the obstacles is more reliable. Both [12] and [13] describe path planning algorithms for UAVs

Huili Yu is a Ph.D. student in Department of Electrical and Computer Engineering, BYU, Provo, Utah, huiliyu.yhl@gmail.com

Randy W. Beard is a professor in Department of Electrical and Computer Engineering, BYU, Provo, Utah, beard@byu.edu

Jeffrey Byrne is jointly appointed with Scientific Systems Company (jbyrne@ssci.com) and with GRASP Lab, Department of Computer and Information Sciences, University of Pennsylvania (jebyrne@cis.upenn.edu)

978-1-4244-4524-0/09/$25.00 ©2009 AACC

based on the multi-resolution cell decomposition of the environment using wavelets. These algorithms are applicable for solving the path planning problem for an agent operating in a partially known environment and the computational cost can be managed. These algorithms assume that the knowledge of the environment at the finest level of resolution is available. Based on that knowledge, the wavelet transform can be applied to decompose the environment at different levels of resolution. However, the UAV only has the knowledge obtained by its sensors and does not likely have knowledge of the environment at the desired levels of resolution. In addition, the algorithms in [11]-[13] were only implemented for the agents operating in known environments.

Based on the suitability of cameras for MAV platforms and the computational advantage of the multi-resolution path planning scheme, this paper explores a vision-based local multi-resolution mapping and path planning algorithm for MAVs. A single strap-down camera is employed to estimate the time-to-collision (TTC) and bearing information of the obstacles. Using computer vision algorithms, we obtain a depth map at each time step. Based on the depth map, the locations of the obstacles in the body frame are obtained and used to create a multi-resolution map in the body frame using polar coordinates. In the multi-resolution map, we use finer resolution to represent areas close to the MAV and coarser resolution to represent areas far from the MAV. We then apply Dijkstra's algorithm to the multi-resolution map and plan collision-free paths in the body frame.

The proposed algorithm can solve the path planning problem for a MAV operating in an unknown environment. Instead of building a map in the global frame, we build a map in the body frame without transforming the camera data from the body frame to the global frame. The sweet spot measurement model with Gaussian distribution is utilized to address the measurement uncertainties produced by a camera and the multi-resolution mapping scheme is introduced to reduce the computational cost. We create a multi-resolution map in polar coordinates that are more compatible with on-board camera information, thereby allowing the data to be processed more efficiently. Based on the multi-resolution map, a path is planned directly in the body frame. We also solve the data association problem in the body frame.

This paper is organized as follows. Section II describes the vision-based local multi-resolution mapping in polar coordinates. In Section III, Dijkstra's algorithm is applied to the multi-resolution map for finding a collision-free path. Section IV provides the simulation and flight test results for the proposed algorithm.

II. Vision-based local multi-resolution mapping for MAVs

In this section, we first describe the localization of the obstacles in the body frame using their TTC and bearing measurements. Since camera resolution decreases as range increases, we present a vision-based local multi-resolution mapping scheme to reduce the computational complexity. We will use polar coordinates in the body frame because this representation is ideally matched to camera information.

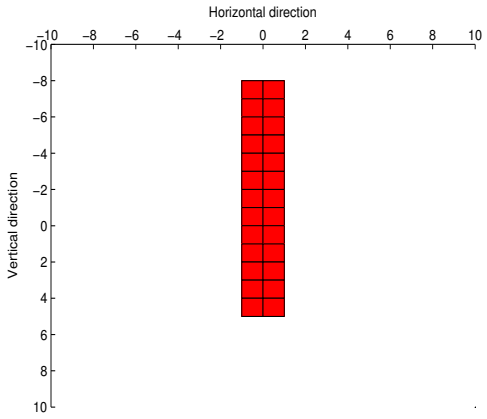


Fig. 1. A 20×20 (in units of pixels) depth map. The gray pixels give the time-to-collision (TTC) to the obstacle and their positions give the bearing to the obstacle.

A. Localization of the obstacles in the body frame

Using computer vision algorithms like time-to-collision methods, a depth map of the obstacles can be obtained [14]. Figure 1 shows a 20×20 (in units of pixels) depth map of a simulated obstacle. In the depth map, the gray pixels provide the TTC information, which gives the range information by multiplying it by the airspeed of the MAV. The positions of the pixels provide bearing information. For instance, the pixel at the jth row and the ith column provides the bearing to the obstacle

$$\begin{aligned} \eta &= tan^{-1}\left(\frac{i-\frac{r_x+1}{2}}{f}\right), \\ \theta &= tan^{-1}\left(\frac{j-\frac{r_y+1}{2}}{f}\right), \end{aligned} \tag{1}$$

where η and θ are the azimuth and elevation of the pixel, r_x and r_y represent the number of pixels along the horizontal and vertical directions in the image plane, and f is the focal length in units of pixels. Note that Eq. (1) is valid for strap-down cameras. If the camera is gimballed, the transformation between the gimbal and body frames must also be considered.

Equation (1) maps all of the gray pixels in the depth map to the points in the body frame, which are represented by the spherical coordinates (r, η, θ). These points form a region in the body frame. Therefore, the locations and sizes of the obstacles can be determined and a map in the body frame can be created in spherical coordinates. When the MAV rolls, the obstacles in the image plane rotate by the negative of the roll angle. Therefore, it is necessary to transform the data about the locations of the obstacles in the body frame to the unrolled body frame to create a more accurate map.

B. Measurement uncertainties

The measurement uncertainties produced by the camera increase as the distance from the MAV increases. This observation motivates the use of a cell decomposition of variable resolution. In the map, the resolution is high in regions adjacent to the MAV and becomes lower with increasing distance. The result is significant savings in computational cost for path planning. The resulting path has high accuracy in the near term, but loses accuracy toward the end of the path. We utilize the sweet spot measurement model with Gaussian distribution described in [15] to represent the measurement uncertainties. The observation made by the camera is given by the linear measurement model

$$z = x + v, \tag{2}$$

where the measurement noise v is a continuous-time Gaussian random variable with zero mean. We assume that the covariance matrix R of v has a diagonal structure

$$R = \begin{bmatrix} {\sigma_r}^2 & 0 \\ 0 & {\sigma_b}^2 \end{bmatrix}. \tag{3}$$

The variance of the range measurement ${\sigma_r}^2$ is a function $f_r(r)$ of the range r from the obstacle to the camera. The variance of the bearing measurement ${\sigma_b}^2$ is also a function $f_b(r)$ of the range. Accordingly, we use the *sweet spot* model

$$\begin{aligned} {\sigma_r}^2 &= f_r(r) = a_1(r-a_2)^2 + a_0 \\ {\sigma_b}^2 &= f_b(r) = \alpha f_r(r), \end{aligned} \tag{4}$$

where a_0, a_1, a_2 and α are model parameters. In this paper, we let $a_2 = 0$, which implies that the noise is at its minimum value at the origin of the body frame.

C. Local multi-resolution mapping in polar coordinates

For the vision-based mapping and path planning problem, the information about the environment is obtained by onboard cameras. Rectangle cell approximation is not the most efficient method of representing the information. We develop a local multi-resolution mapping scheme using polar coordinates, which are more compatible to the range and bearing information obtained by the camera and allow the data to be processed more efficiently. The scheme is based on the fact that a disk in the (x,y) plane can be mapped to a rectangle in the (η, r) plane using the polar transform $x = r\cos\eta$ and $y = r\sin\eta$. Using this scheme, we make the cell decomposition in the bearing-range plane and obtain the corresponding map in the body frame represented by Fig. 2.

In Fig. 2, three levels of resolution are utilized to describe the regions at the given distances from the current location of the MAV. The highest level of resolution is given to the region close to the origin and the lower levels of resolution are given to the region far away. For two adjacent resolution levels, the incremental angles and radii of the sectors at the higher level of resolution are twice as small as the incremental angles and radii of the sectors at the lower level of resolution. The circles represent the obstacles and the ellipses around them represent the range and bearing uncertainties, which decrease as the distances between the MAV and the obstacles decrease.

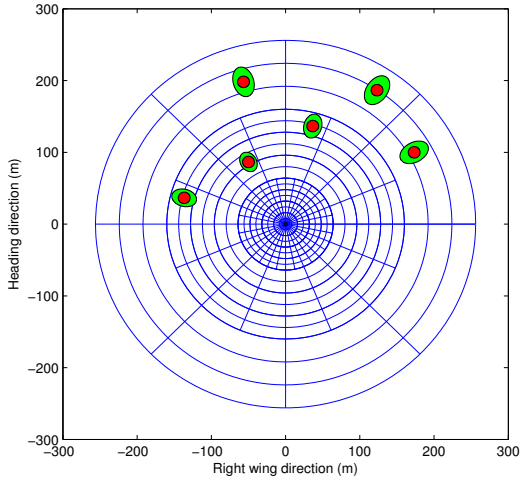


Fig. 2. The local multi-resolution map in the body frame of the MAV using polar coordinates. Three levels of resolution are used to describe the regions at the given ranges from the current location of the MAV. For two adjacent resolution levels, the incremental angles and radii of the sectors at the higher level of resolution are twice as small as the incremental angles and radii of the sectors at the lower level of resolution. The circles represent the obstacles and the ellipses around them represent the range and bearing uncertainties.

D. Data association

When the obstacles leave the field of view of the camera, their locations cannot be updated by the vision sensor. Therefore, the locations of these obstacles must be updated using a propagation model since the camera may observe them again during the flight. Instead of addressing the data association problem in the inertial frame, we directly solve the data association problem in the body frame, which saves the computational cost of transforming the data in the body frame to the inertial frame. Figure 3 shows the geometry of the locations of the MAV and the obstacles at two consecutive time steps in a two dimensional plane, where the vehicle and body frames at the two time steps are represented by $\mathscr{F}_1^v$ (specified by $(\hat{i}_1^v, \hat{j}_1^v, \hat{k}_1^v)$), $\mathscr{F}_1^b$ (specified by $(\hat{i}_1^b, \hat{j}_1^b, \hat{k}_1^b)$), $\mathscr{F}_2^v$ (specified by $(\hat{i}_2^v, \hat{j}_2^v, \hat{k}_2^v)$) and $\mathscr{F}_2^b$ (specified by $(\hat{i}_2^b, \hat{j}_2^b, \hat{k}_2^b)$) respectively, ψ_1 and ψ_2 are the heading angles, r_1 and r_2 are the ranges to the obstacle, η_1 and η_2 are the bearings to the obstacle, d_n and d_e are the distances traveled in the North and East directions between the two time steps, and $\mathbf{B}$ is the location of the obstacle.

Based on Fig. 3, the locations of the origin of the $\mathscr{F}_2^v$ frame and the obstacle with respect to the $\mathscr{F}_1^v$ frame are given by $(d_n, d_e, 0)$ and $(r_1\cos(\eta_1+\psi_1), r_1\sin(\eta_1+\psi_1), 0)$ respectively. Hence, the range and bearing to the obstacle at the current time step are given by

$$\begin{aligned} r_2 &= \sqrt{(r_1\cos(\eta_1+\psi_1) - d_n)^2 + (r_1\sin(\eta_1+\psi_1) - d_e)^2} \\ \eta_2 &= \tan^{-1}\left(\frac{r_1\sin(\eta_1+\psi_1) - d_e}{r_1\cos(\eta_1+\psi_1) - d_n}\right) - \psi_2. \end{aligned} \tag{5}$$

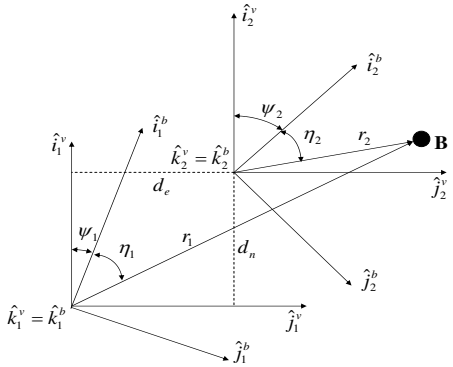


Fig. 3. The geometry of the locations of the MAV and the obstacle at two consecutive time steps in a two dimensional plane. The vehicle and body frames at the two time steps are represented by $\mathscr{F}_1^v$ (specified by $(\hat{i}_1^v, \hat{j}_1^v, \hat{k}_1^v)$), $\mathscr{F}_1^b$ (specified by $(\hat{i}_1^b, \hat{j}_1^b, \hat{k}_1^b)$), $\mathscr{F}_2^v$ (specified by $(\hat{i}_2^v, \hat{j}_2^v, \hat{k}_2^v)$) and $\mathscr{F}_2^b$ (specified by $(\hat{i}_2^b, \hat{j}_2^b, \hat{k}_2^b)$) respectively. The heading angles are given by ψ_1 and ψ_2. The ranges to the obstacle are given by r_1 and r_2 and the bearings to the obstacle are given by η_1 and η_2. The distances traveled in the North and East directions between the two time steps are given by d_n and d_e. The location of the obstacle is represented by **B**.

Using this method, the locations of the obstacles can be retained in memory and updated as more knowledge is obtained from the camera data. The data association algorithm in the body frame is described as Algorithm 1.

Algorithm 1: Data association algorithm

```
Localize the obstacles retained in memory in the body frame at current time step;
for i ← 1 to the number of the new obstacles do
    flag ← 0;
    for j ← 1 to the number of the obstacles retained in memory do
        if the distance between the ith new obstacle and the jth obstacle retained in memory is less than a threshold then
            Update the information of the jth obstacle retained in memory by combining the information of the ith new obstacle;
            flag ← 1;
            break;
        end
    end
    if flag = 0 then
        Store the ith new obstacle to the memory;
    end
end
```

III. Path planning for MAVs

Given the multi-resolution map, a path can be planned for the MAV to achieve its goal while avoiding the obstacles. In this section, we use Dijkstra's algorithm to plan a collision-free path.

Dijkstra's algorithm is a graph search algorithm that solves the shortest path problem for the graph. Based on the cell decomposition, we consider cells as nodes for the graph and create the node connectivity to form edges. The connectivity within the cells at the same level of the resolution is set to an 8-way neighborhood. Special care must be taken at the borders between different resolution levels to connect the neighboring cells.

Dijkstra's algorithm needs a cost for each edge in the graph to find an optimal path. We assign the cost function for each edge based on both the length of the edge and the risk measure of the ending node of that edge.

Let $\mathscr{F}$ and $\mathscr{O}$ be the free and occupied spaces in the environment and let $\mathscr{C} = \{\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_n}\}$, where n is the number of cells in the environment and $\mathbf{x_i} = (x_i, y_i)$, $i = 1, 2, \cdots, n$ is the center location of the ith cell. Suppose that $\mathbf{x} \in \mathscr{C}$ and $\mathbf{y} \in \mathscr{O}$. The risk measure of the ith cell can be computed as

$$\mathrm{rm}(\mathbf{x_i}) = \begin{cases} (d_{\max} - \min_{\mathbf{y}\in\mathscr{O}} \|\mathbf{x_i} - \mathbf{y}\|_2)/d_{\max} & \mathbf{x_i} \in \mathscr{F} \\ 1 & \mathbf{x_i} \in \mathscr{O} \end{cases}, \quad (6)$$

where $d_{\max} \triangleq \max_{\mathbf{x}\in\mathscr{C}} \min_{\mathbf{y}\in\mathscr{O}} \|\mathbf{x} - \mathbf{y}\|_2$.

The cost function $J(u,v) = \mathrm{rm}(v) + w\mathrm{dist}(u,v)$, which combines the risk measure and the Euclidean distance between the neighboring nodes, is assigned to the edge from node u to v , where the term $\mathrm{rm}(v)$ penalizes the risk measure of node v and the term $\mathrm{dist}(u,v)$ penalizes the distance between the nodes u and v. The parameter $w \geq 0$ is a positive weight. Using the cost function $J(u,v)$, Dijkstra's algorithm is employed to find a collision-free path. The path is optimal from the current location of the MAV to the goal based on the current map in the body frame.

The path generated by Dijkstra's algorithm includes a series of waypoints and may not be feasible due to the kinematic constraints of the MAV. Algorithm 2 generates a smoothed path for the MAV to track.

Algorithm 2: Path smoothing algorithm

```
Obtain the input waypoint list W_in = {w_1^in, w_2^in, ..., w_n^in} from Dijkstra's algorithm;
Allocate memory for the output waypoint list W_out to store the waypoints for the smoothed path;
Set w_1^in in W_in as the current waypoint;
Insert the current waypoint to W_out;
i ← n;
while i ≥ the index of current waypoint + 1 do
    if there do not exist collisions between the current waypoint and the ith waypoint then
        Set the ith waypoint as the current waypoint;
        Insert the current waypoint to W_out;
        i ← n;
    end
    else
        i ← i − 1;
    end
end
```

IV. Results

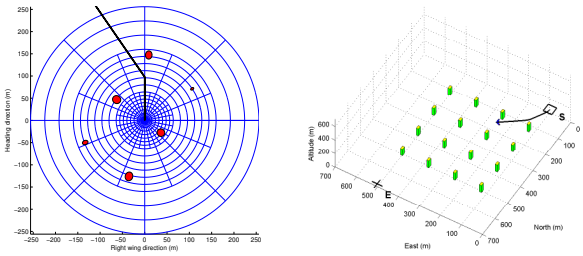
A. Simulation results

The feasibility of the method was tested using a simulation environment developed in MATLAB/SIMULINK. This simulator uses a six degree-of-freedom model for the aircraft. The coordinate system is represented by NED (North-East-Down) system. The MAV was commanded to continuously maneuver through sixteen $16m \times 10m \times 100m$ obstacles between the waypoint **S** (0,100,-40) and the waypoint **E** (700,500,-40), which are represented by the square and the plus sign in Fig. 5 respectively. A 20x20 pixel depth map was used with the weight constant $w = 0.005$. The parameters for the sweet spot measurement model were set at $a_0 = 0.1528$, $a_1 = 0.001$, $a_2 = 0$ and $\alpha = 0.00002$.

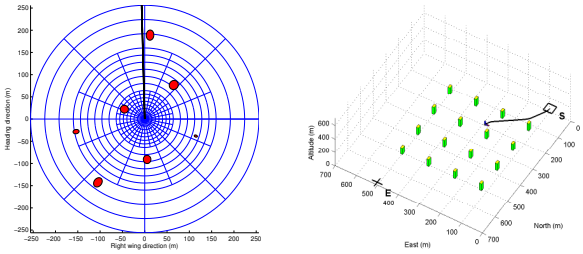
Using the proposed algorithm, the three levels of resolution map in the body frame using polar coordinates is shown in Fig. 4. At the three levels of resolution, the incremental angles of cells are 11.25°, 22.5°, and 45°, and the incremental radii are $8m$, $16m$ and $32m$ respectively. The ellipses represent the estimated locations and sizes of the obstacles. Figure 4 also shows the update of the map and the evolution of the corresponding optimal path as more and more obstacles are observed. Figure 4(a) shows the map and the corresponding path at time step $t = 18s$ and the location of the MAV in the inertial frame at that time. Figure 4(b) shows the map and the path at time step $t = 22s$. In Fig. 4(b), the predicted path is very close to an obstacle. This is due to the fact that the MAV does not have complete information about that obstacle at that time. As the MAV gets closer to the obstacle, new information is observed and the path is replanned, which is shown in Fig. 4(c). Figure 4(d) and (e) show the map and the path to the waypoint **E** in the body frame at time step $t = 42s$ and $t = 62s$, which is represented by the plus sign. The final path followed by the MAV between the two waypoints is shown in Fig. 5.

B. Flight test results

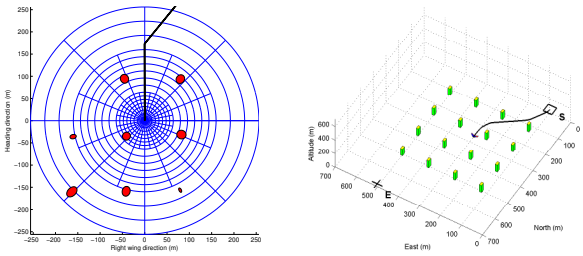
We conducted flight test using a MAV with a wing span of 48 inches. The Kestrel autopilot from Procerus Technologies navigated the MAV with control of roll, pitch and yaw [16][17]. The guidance law was processed in MATLAB on the ground station and roll commands transmitted to the autopilot. Three obstacles were simulated in MATLAB on the ground station and their locations were set as (-80,10), (-150,10) and (-100,80) respectively. The MAV was commanded to maneuver through the three obstacles from West to East at an altitude of 25 meters. The MAV had no a priori information about the locations of the obstacles. A GPS telemetry plot of the results is shown in Fig. 6. As the MAV approached the obstacles, the proposed algorithm generated a path around the obstacles and the MAV began to track the generated path. As the MAV passed the obstacles, it once again began to track the original waypoint path. The MAV successfully avoided the obstacles without human intervention.



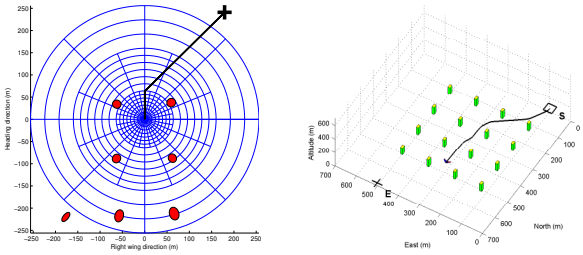
(a) $t = 18s$



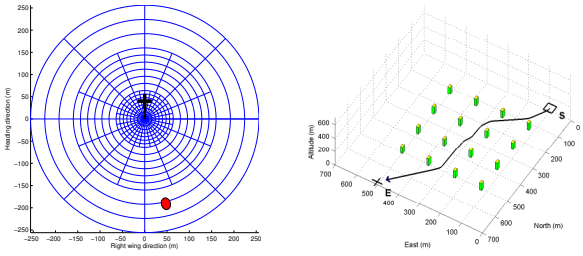
(b) $t = 22s$



(c) $t = 28s$



(d) $t = 42s$



(e) $t = 62s$

Fig. 4. The update of the multi-resolution map and the evolution of the path. Figures on the left show the predicted path based on the available information about the obstacles at different time steps. The ellipses represent the estimated locations and sizes of the obstacles and the plus signs in both subfigures (d) and (e) represent the waypoint **E** in the body frame. Figures on the right show the inertial path followed by the MAV.

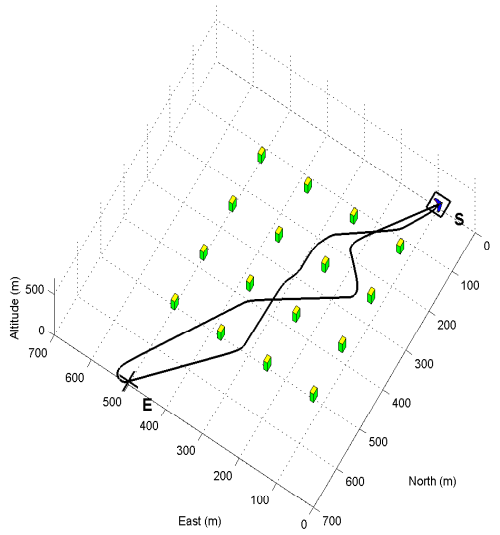


Fig. 5. The final path followed by the MAV between the waypoints **S** (square) and **E** (plus sign).

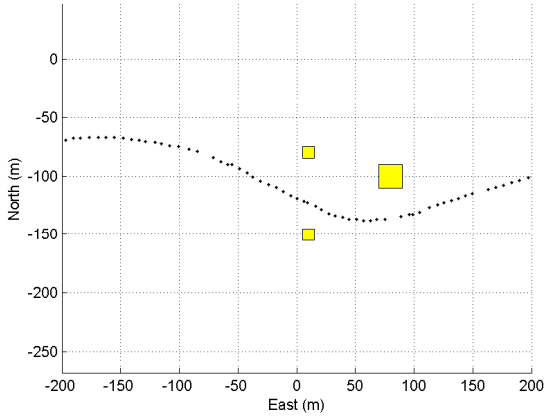


Fig. 6. Flight test results plot of a MAV avoiding three obstacles with a planned path through the obstacles. The three squares represent the three obstacles.

V. Conclusions

In this paper we present a vision-based local multi-resolution mapping and path planning technique for MAVs operating in unknown environments. The multi-resolution representation of the environment is used. The higher resolution is given to areas close to the MAV and the lower resolution is given to areas which are far away. This technique is motivated by the observation that the measurement uncertainties produced by the camera increase as the distance from the MAV increases. The local multi-resolution map is created in polar coordinates, which are more compatible to on-board camera data. The measurement uncertainties are addressed using the sweet spot measurement model with Gaussian distribution. The data association problem is directly solved in the body frame without transforming the data between the body and inertial frames. Using Dijkstra's algorithm, the path is planned in the body frame.

In this work, we use polar coordinates to create the multi-resolution map for addressing the two dimensional path planning problem. In the future, we will use spherical coordinates to create multi-resolution maps that address the three dimensional path planning problem.

VI. Acknowledgement

This research was supported by the Air Force Research Laboratory, Munitions Directorate under SBFR contract No. FA 8651-07-c-0094 to Scientific Systems Company, Inc. and Brigham Young University.

References

[1] B. Call, "Obstacle avoidance for unmanned air vehicles using computer vision", Master Thesis, Brigham Young University, December, 2006.

[2] A. Curtis, "Path planning for unmanned air and ground vehicles in urban environments", Master Thesis, Brigham Young University, 2008.

[3] K.H. Sedighi, K. Ashenayi, R.L. Wainwright and H.M. Tai, "Autonomous local path planning for a mobile robot using a genetic algorithm", *Congress on Evolutionary Computation*, vol. 2, June 2004, pp. 1338-1345.

[4] J.C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, Boston,MA, 1991.

[5] R.E. Zelenka and L.D. Almsted, "Design and flight test of 35-Gigahertz radar for terrain and obstacle avoidance", *Journal of Aircraft*, vol. 34, March/April 1997, pp. 261-263.

[6] R.E. Zelenka, R.F. Clark and A. Zirkler, "Development and flight test of terrain-referenced guidance with ladar forward sensor", *Journal of Guidance Control and Dynamics*, vol. 19, July/August 1996, pp. 823-828.

[7] J. Saunders, B. Call, A. Curtis, R. Beard and T. McLain, "Static and dynamic obstacle avoidance in miniature air vehicles", *AIAA Infotech at Aerospace*, paper no. AIAA-2005-6950, 2005.

[8] Y. Petillot, I.T. Ruiz and D.M. Lane, "Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar", *IEEE Journal of Oceanic Engineering*, vol. 26, no. 2, April 2001, pp. 240-251.

[9] B. Call, R. Beard and C. Taylor, "Obstacle avoidance for unmanned air vehicles using image feature tracking", *AIAA Conference on Guidance, Navigation , and Control*, Keystone CO, paper no. AIAA-2006-6541, 2006.

[10] S. Kambhampati and L.S. Davis, "Multiresolution path planning for mobile robots", *IEEE Journal of Robotics And Automation*, vol. RA-2, no. 3, Septmber 1986.

[11] S. Behnke, "Local multiresolution path planning", vol. 3020 of *Lecture Notes in Computer Science*, Berlin: Springer, 2004, pp. 332-343.

[12] P. Tsiotras and E. Bakolas, "A hierarchical on-line path-planing scheme using wavlets", in *European Control Conference*, (Kos, Greece), July 2-5 2007.

[13] R.V. Cowlagi and P. Tsiotras, "Multiresolution path planning with wavelets: a local replanning approach", *IEEE American Control Conference*, June 2008.

[14] J. Byrne and C.J. Taylor, "Expansion segmentation for visual collision detection and estimation", Proceedings of the 2009 IEEE International Conference on Robotics and Automation, May 21-17, 2009.

[15] P. Yang, R. A. Freeman and K.M. Lynch, "Multi-agent coordination by decentralized estimation and control", *IEEE Transaction on Automatic Control*, 2008.

[16] R. Beard, D. Kingston and M. Quigley, etc, "Autonomous vehicle technologies for small fixed wing UAVS", *Jounral of Aerospace Computing, Information, And Communication*, vol. 2, January 2005, pp. 92-102.

[17] Procerus, "Procerus Technologies: Fly light with the world's smallest uav autopilot", http://www.procerusuav.com/, August 2006.