Noname manuscript No.
(will be inserted by the editor)

A Vision-based Collision Avoidance Technique for Micro Air Vehicles Using Local-level Frame Mapping and Path Planning

Huili Yu · **Randy Beard**

Received: date / Accepted: date

Abstract This paper presents a vision-based collision avoidance technique for small and Miniature Air Vehicles (MAVs) using local-level frame mapping and path planning. Using computer vision algorithms, a depth map that represents the range and bearing to obstacles is obtained. Based on the depth map, we estimate the range, azimuth to, and height of obstacles using an extended Kalman Filter (EKF) that takes into account the correlations between obstacles. We then construct maps in the local-level frame using cylindrical coordinates for three dimensional path planning and plan Dubins paths using the Rapidly-Exploring Random Tree (RRT) algorithm. The behavior of our approach is analyzed and the characteristics of the environments where the local path planning technique guarantees collision-free paths and maneuvers the MAV to a specific goal region are described. Numerical results show the proposed technique is successful in solving path planning and multiple obstacle avoidance problems for fixed wing MAVs.

Keywords Path planning · Collision avoidance · Micro Air Vehicle · Computer vision

1 Introduction

Small and 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 problem for MAVs has received significant attention [1–5].

The path planning problem can be grouped into global path planning and local path planning [6]. 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 vehicle starts its motion [6]. The global path planning problem has been addressed by many researchers with the three most common solutions being potential fields methods, probabilistic roadmap methods, and cell decomposition methods [7]. For environments where complete knowledge is not available, obstacles may pop up as the MAV flies along pre-defined trajectories generated by the global path planning algorithm and collisions may occur.

Consequently, in this paper a local path planning algorithm is designed to address the path planning problem for unknown or partially known environments. It is executed in real-time where the basic idea is to first sense the obstacles in the environment and then determine a collision-free path [1]. Local path planning algorithms require sensors to detect the obstacles. Among the suite of possible sensors, a video camera is cheap and lightweight and fits the physical requirements of small UAVs [1]. Since the camera measurements are obtained in the body frame of the MAV, it is most natural to directly build maps and to plan paths in the local-level frame using the camera measurements without transforming to the inertial frame. In addition, building maps and planning paths in the local-level frame does not require access to GPS data. Reference [8] designs a local path planning algorithm using the sensor data directly to construct a sector-like multi-resolution decomposition of the agent's immediate environment using wavelets. The cell decomposition is constructed in polar coordinates that are compatible with the on-board sensor data and a path is planned locally based on the cell decomposition. This algorithm assumes 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, for an agent operating in unknown environments, it only has the knowledge obtained by its sensors and does not likely have knowledge of the environment at the desired levels of resolution. Our previous work in [9–12] develops a vision-based local-level frame mapping and path planning technique for MAVs oper-

H. Yu
Department of Mechanical Engineering, Boston University, Boston, MA, 02215, USA
Tel.: +1-801-900-1271
E-mail: huiliyu.yhl@gmail.com

R. Beard
Department of Electrical and Computer Engineering, Brigham Young University, Provo, Utah, 84602, USA
E-mail: beard@byu.edu

ating in unknown environments. Based on the numerical and experimental results, the technique is successful in solving the two and a half dimensional path planning problem for MAVs. However, correlations between the locations of obstacles have not been considered for building maps, and the kinematic constraints of the MAV have not been considered for planning paths. In addition, the behavior of the local path planning algorithm was not analyzed in [9–12].

This paper explores a vision-based collision avoidance technique for MAVs using local-level frame mapping and planning. Using computer vision based time-to-collision estimation, we obtain a depth map at each time step. Based on the depth map, we use an extended Kalman Filter (EKF) to estimate the range, azimuth to, and height of obstacles. This information is used to construct a map of the environment that is suitable for path planning. A unique aspect of our work is that the map is referenced to the (time-varying) local-level frame. The map is built in cylindrical coordinates to facilitate solving the three dimensional path planning problem. Given the planning map, we use the Rapidly-Exploring Random Tree (RRT) algorithm to plan kinematically feasible collision-free Dubins paths in the local-level frame. The salient features of this paper are as follows.

- We build cylindrical maps in the local-level frame of the MAV for solving the three dimensional path planning problem.
- We design an EKF to jointly estimate the range, azimuth, and height for all the existing obstacles in the map, explicitly accounting for the correlations between obstacles.
- The RRT algorithm is used to plan collision-free Dubins paths in the local-level frame that satisfy the kinematic constraints of the MAV.
- We analyze the behavior of the local path planning algorithm and describe the characteristics of the environments in which the local path planning algorithm is guaranteed to generate collision-free paths and to maneuver the MAV to the goal region.

The paper is organized as follows. Section 2 formulates the path planning problem for MAVs. Section 3 describes the vision-based mapping process in the local-level frame. In Section 4, the RRT algorithm is applied to the maps for finding collision-free Dubins paths. The behavior of the proposed local planning algorithm is analyzed in Section 5. Section 6 presents numerical results for the algorithm.

2 Problem formulation

The kinematic guidance model of the MAV can be represented by $\dot{\mathbf{x}}_a = \mathbf{f}_a(\mathbf{x}_a, \mathbf{u})$, where $\mathbf{x}_a \in \mathscr{X}_a$ is the state, where $\mathscr{X}_a$ is the state space, and $\mathbf{u}$ is the control input, taking values from the set $\mathscr{U}$. Based on [3], the state space $\mathscr{X}_a$ can be decomposed into the product $\mathscr{C} \times \mathscr{Y}$, where $\mathscr{C}$ is the configuration space and $\mathscr{Y}$ encodes the vehicle's velocity and higher order derivatives of the state variables. Similar to [3], we design a planning algorithm that generates collision-free paths in the configuration space. The algorithm does not specify the actual control input, but rather relies on an inner loop controller to track the collision-free paths.

Let $\mathscr{C}_{obst}$ represent the set of configurations where the MAV is either in collision or cannot avoid a collision because of kinematic constraints, and let $\mathscr{C}_{free} = \mathscr{C} \backslash \mathscr{C}_{obst}$. A path between two configurations is said to be *feasible* if all configurations along the path satisfy the kinematic constraints and are contained in $\mathscr{C}_{free}$. The path planning problem for the MAV can be formulated as finding a feasible path from an initial configuration to a goal configuration. A path planning algorithm is *complete* if it returns a feasible solution when one exists and returns failure otherwise.

3 Vision-based local-level frame mapping

In this section, we describe the range and bearing measurements to obstacles, and we describe the measurement uncertainties produced by an on-board camera. Based on the camera measurements, we use an EKF to estimate the range, azimuth to, and height of obstacles. We use a joint compatibility branch and bound approach to address the data association problem. We then build a planning map using cylindrical coordinates in the local-level frame that is suitable for solving the three dimensional path planning problem in a computationally efficient manner.

3.1 Range and bearing measurements

There are numerous computer vision algorithms that estimate the time-to-collision to obstacles in the camera field of view. Reference [13] provides an overview of algorithms. By multiplying time-to-collision by the ground speed of the MAV, a depth map that represents the range and bearing to obstacles can be obtained. In this paper, we assume that the depth map is available. Figure 1 shows an image of a simulated environment that we call *Megacity*, and an associated 640×480 (in units of pixels) depth map, where each pixel provides the range information. The darker pixels represent the areas that are close to the MAV and the lighter pixels represent the areas that are far away from the MAV. The corresponding position of each pixel also provides the bearing information in the body frame. In this paper, we construct a map of the world in the (time-varying) local-level frame, where the local-level frame is the body frame with roll and pitch angles set to zero. The origin of the local-level frame is the MAV's center of mass. The x-axis points out the nose of the airframe when the airframe is not pitching, the y-axis

points out the right wing when the airframe is not rolling, and the z-axis points into the Earth. Let ϕ and θ represent the MAV roll and pitch angles. Figure 2 shows how the azimuth and elevation information expressed in the local-level frame for the pixel at the j^{th} row and the i^{th} column of the depth map can be obtained. Consider the center of the depth map as its origin and consider its width and height directions as its x and y axes. Let r_x and r_y represent the number of pixels in width and height of the depth map respectively. The coordinates of the pixel at the j^{th} row and the i^{th} column of the depth map are $[i-\frac{r_x+1}{2}, j-\frac{r_y+1}{2}]^\top$. The pixel coordinates after the pixel is rotated about the origin of the depth map by $-\phi$ are given by

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} i-\frac{r_x+1}{2} \\ j-\frac{r_y+1}{2} \end{bmatrix}. \tag{1}$$

The azimuth and elevation angles to the pixel with the roll angle removed are given by

$$\eta' = \tan^{-1}\left(\frac{i'}{f}\right), \tag{2}$$

$$\xi' = \tan^{-1}\left(\frac{j'}{\sqrt{f^2+(i')^2}}\right), \tag{3}$$

where f is the focal length (in units of pixels) of the camera. The azimuth and elevation angles to the pixel with both the roll and pitch angles removed are given by

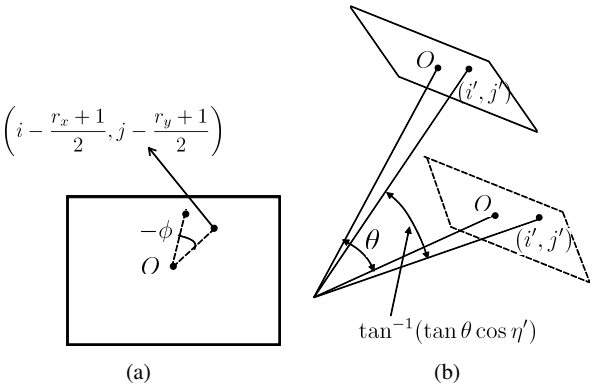
$$\eta = \eta', \tag{4}$$

$$\xi = \xi' + \tan^{-1}(\tan\theta\cos\eta'). \tag{5}$$

(a) Megacity (b) Depth map

Fig. 1 This figure shows the *Megacity* simulation environment and its 640×480 (in units of pixels) depth map. Each pixel in the depth map gives the range to obstacles and its position gives the bearing to obstacles.

Using the K-mean clustering method [14], the range, azimuth, and elevation data provided by all pixels in the depth map can be classified into a group of measurements $\{\mathbf{z}_1, \cdots, \mathbf{z}_m\}$, where $\mathbf{z}_j = [r_j, \eta_j, \xi_j]^\top$, $\forall j = 1, \cdots, m$. Suppose that there exist n obstacles in the local-level frame map. Let r_i, η_i, and h_i represent the range, azimuth to, and height of the i^{th} obstacle and let $\mathbf{x}_i = [r_i, \eta_i, h_i]^\top$ represent the state vector associated with the obstacle. Assuming that the measurement noise is normally distributed, the j^{th} measurement



(a) (b)

Fig. 2 This figure shows how the range and bearing expressed in the local-level frame for the pixel at the j^{th} row and the i^{th} column of the depth map can be obtained. The roll and pitch angles are represented by ϕ and θ. The origin of the depth map is O.

associated with the i^{th} obstacle in the local-level frame map at time step k is given by

$$\mathbf{z}_j[k] = \mathbf{h}_j(\mathbf{x}_i[k]) + \mathbf{v}_j[k] = \begin{bmatrix} r_i[k] \\ \eta_i[k] \\ \tan^{-1}\left(\frac{h_i[k]}{r_i[k]}\right) \end{bmatrix} + \mathbf{v}_j[k], \tag{6}$$

where the measurement noise $\mathbf{v}_j[k]$ is a Gaussian random vector with zero mean. We assume that the covariance matrix $\mathbf{R}_j[k]$ of $\mathbf{v}_j[k]$ has a diagonal structure

$$\mathbf{R}_j[k] = \begin{bmatrix} \sigma^2_{r_j}[k] & 0 & 0 \\ 0 & \sigma^2_{\eta_j}[k] & 0 \\ 0 & 0 & \sigma^2_{\xi_j}[k] \end{bmatrix}. \tag{7}$$

Since the measurement uncertainties produced by the camera increase as the distance from the MAV increases, we use the *sweet spot* measurement uncertainty model [15] to represent the diagonal terms of $\mathbf{R}_j[k]$ as

$$\sigma^2_{r_j}[k] \triangleq a_1(r_j[k]-a_2)^2 + a_0, \tag{8}$$

$$\sigma^2_{\eta_j}[k] \triangleq a_3\sigma^2_{r_j}[k], \tag{9}$$

$$\sigma^2_{\xi_j}[k] \triangleq a_4\sigma^2_{r_j}[k], \tag{10}$$

where a_0, a_1, a_2, a_3, and a_4 are model parameters. In this model, the range measurement noise variance $\sigma^2_{r_j}[k]$ is a function of the range $r_j[k]$ to the obstacle from the camera. The azimuth and elevation measurement noise variances are also related to the range. This measurement uncertainty model assumes there exists a "sweet spot" location $r_j[k] = a_2$ at which the noise is at its minimum value [15]. As the difference between the range $r_j[k]$ and the sweet spot a_2 increases, the noise increases.

3.2 Range, azimuth, and height estimation using the EKF

Based on the measurements, we use an EKF to estimate the range, azimuth, and height. Since the obstacle map is in the

local-level frame of the MAV, which is located at the origin, we need to derive the equation of motion of each obstacle relative to the MAV. In the paper, we assume zero wind conditions. Let V represent the ground speed of the MAV, which is assumed to be available. Let ψ represent the yaw angle and let γ represent the flight path angle. Figure 3 depicts the motion of the i^{th} obstacle relative to the MAV in the local-level frame, where $p(t)$ is the location of the MAV, O_i represents the obstacle, r_i is the range to the obstacle considered in the x-y plane of the local-level frame, η_i is the azimuth to the obstacle whose positive direction is defined as the right-handed rotation about the z-axis of the local-level frame, and h_i is the height of the obstacle in the local-level frame.

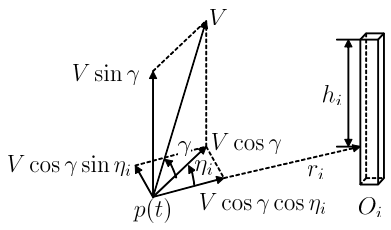


Fig. 3 This figure shows the motion of the i^{th} obstacle relative to the MAV. The current location of the MAV is $p(t)$. The obstacle is represented by O_i. The ground speed is represented by V. The flight path angle is represented by γ. The range and azimuth to the obstacle are represented by r_i and η_i. The height of the obstacle in the local-level frame is represented by h_i.

Based on Fig. 3, the equation of motion of the obstacle relative to the MAV in terms of range, azimuth, and height is given by

$$\dot{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{x}_i) + \mathbf{w}_i = \begin{bmatrix} -V\cos\gamma\cos\eta_i \\ \frac{V\cos\gamma\sin\eta_i}{r_i} - \dot{\psi} \\ -V\sin\gamma \end{bmatrix} + \mathbf{w}_i, \tag{11}$$

where we assume the coordinated turn conditions $\dot{\psi} = \frac{g}{V}\tan\phi$, where g is the gravity constant, and the process noise $\mathbf{w}_i$ is a Gaussian random vector with zero mean and covariance matrix $\mathbf{Q}_i$.

The motion of the MAV in the inertial frame determines the motion of obstacles relative to the MAV in the local-level frame. Accordingly, the locations of obstacles in the local-level frame are correlated to each other through the MAV's location. We design the EKF to take into account the correlations. Let

$$\mathbf{P}_{ij} = \begin{pmatrix} \sigma^2_{r_ir_j} & \sigma^2_{r_i\eta_j} & \sigma^2_{r_ih_j} \\ \sigma^2_{\eta_ir_j} & \sigma^2_{\eta_i\eta_j} & \sigma^2_{\eta_ih_j} \\ \sigma^2_{h_ir_j} & \sigma^2_{h_i\eta_j} & \sigma^2_{h_ih_j} \end{pmatrix}$$

represent the state error covariance matrix between the i^{th} and the j^{th} obstacles. Let $\mathbf{x} = \left[\mathbf{x}_1^\top, \mathbf{x}_2^\top, \cdots, \mathbf{x}_n^\top\right]^\top$ represent the augmented state vector of the n obstacles, where $\mathbf{x}_i = [r_i, \eta_i, h_i]^\top$, and let

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_{11} & \cdots & \mathbf{P}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{P}_{n1} & \cdots & \mathbf{P}_{nn} \end{pmatrix}$$

represent the augmented state error covariance matrix. Let $\mathbf{z}[k] = \left[\mathbf{z}_1[k]^\top, \mathbf{z}_2[k]^\top, \cdots, \mathbf{z}_m[k]^\top\right]^\top$ represent m measurements at time step k. The state transition and observation models for the augmented system are given by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{w}, \tag{12}$$

$$\mathbf{z}[k] = \mathbf{h}(\mathbf{x}[k]) + \mathbf{v}[k], \tag{13}$$

where $\mathbf{f} = \left[\mathbf{f}_1^\top, \mathbf{f}_2^\top, \cdots, \mathbf{f}_n^\top\right]^\top$ and $\mathbf{f}_i$ is given by Eq. (11), $\mathbf{h} = \left[\mathbf{h}_1^\top, \mathbf{h}_2^\top, \cdots, \mathbf{h}_m^\top\right]^\top$ and $\mathbf{h}_j$ is given by Eq. (6), $\mathbf{w} = \left[\mathbf{w}_1^\top, \mathbf{w}_2^\top, \cdots, \mathbf{w}_n^\top\right]^\top$ and its covariance matrix $\mathbf{Q} = diag(\mathbf{Q}_1, \mathbf{Q}_2, \cdots, \mathbf{Q}_n)$, $\mathbf{v}[k] = \left[\mathbf{v}_1[k]^\top, \mathbf{v}_2[k]^\top, \cdots, \mathbf{v}_m[k]^\top\right]^\top$ and its covariance matrix $\mathbf{R}[k] = diag(\mathbf{R}_1[k], \mathbf{R}_2[k], \cdots, \mathbf{R}_m[k])$. The proposed scheme for estimating $\mathbf{x}$ is a standard continuous-discrete time EKF algorithm [16].

Let $\mathscr{C}$ represent the configuration space and let $q(t) = [q_n(t), q_e(t), q_d(t), q_\psi(t)]^\top$ represent the MAV configuration at time t, where $q_n(t)$, $q_e(t)$, and $q_d(t)$ are North, East, and Down coordinates, and $q_\psi(t)$ is the heading angle. In this paper, for two configurations q_1 and q_2, we define the distance between q_1 and q_2 as

$$\|q_1 - q_2\| \triangleq \sqrt{(q_{1n}-q_{2n})^2 + (q_{1e}-q_{2e})^2 + (q_{1d}-q_{2d})^2}, \tag{14}$$

and define the distance between q_1 and q_2 projected on x-y plane of the local level frame as

$$\|q_1 - q_2\|_{2D} \triangleq \sqrt{(q_{1n}-q_{2n})^2 + (q_{1e}-q_{2e})^2}. \tag{15}$$

Let $M(q(t)) = \{q' \in \mathscr{C} : \|q' - q(t)\|_{2D} \leq R^l \text{ and } |q'_d - q_d(t)| \leq \frac{H}{2}\}$ represent the cylindrical region with radius R^l and with height H centered at the origin of the local-level frame map. When an obstacle disappears from $M(q(t))$ for a time t, it is removed from the map. We remove the states corresponding to the obstacle from the state vector and remove the associated row and column from the error covariance matrix.

3.3 Data association

The camera measurements must be associated with the existing obstacles in the map correctly. In addition, the MAV may revisit obstacles that already exist in the map after an extended period of time. The challenge is to associate the new camera measurements with those obstacles. These two problems motivate data association algorithms that relate sensor

measurements with the features included in the map. A data association algorithm is composed of two elements: a test to determine the compatibility between a sensor measurement and a map feature, and a selection criterion to choose the best match among the set of possible matches [17].

The nearest neighbor approach, which is a classic tracking technique, can be used to address the data association problem, where the normalized innovation square test is employed to determine compatibility, and the smallest Mahalanobis distance is used to select the best matchings [18]. However, the nearest neighbor approach does not take into account the correlations among obstacles and it causes the EKF to diverge especially for cluttered environments like urban terrains.

As an alternative, we use the joint compatibility branch and bound (JCBB) method [17]. This data association algorithm generates tentative sets of associations and searches the largest set that satisfies the joint compatibility test [17, 19]. For a given set of association pairs, the joint compatibility test is determined by computing a joint normalized innovation squared gate. The advantage of the test is that it preserves the correlations among the set of the observations and predicted observations [19].

3.4 Local-level frame mapping

Based on the range, azimuth to, and height of obstacles, we build a map directly in the local-level frame instead of the inertial frame. Accordingly, we save the computational resources of transforming the camera data from the local-level frame to the inertial frame, at the expense of updating the map from body motion. In addition, collision avoidance is inherently a local phenomenon and vision data is obtained in the body frame of the MAV. Therefore, transforming to the inertial frame is unneeded and introduces error. We construct maps in cylindrical coordinates for the three dimensional path planning problem that are more compatible with vision data than Cartesian coordinates and that allow the data to be processed more efficiently. We encode obstacles with general shape in the environment as the smallest cylinders that enclose obstacles. By doing so, all the obstacles in the local-level frame map are assumed to be cylinders. Figure 4 shows the local-level frame map in cylindrical coordinates, where the origin of the map is the current location of the MAV and the cylinders represent the obstacles.

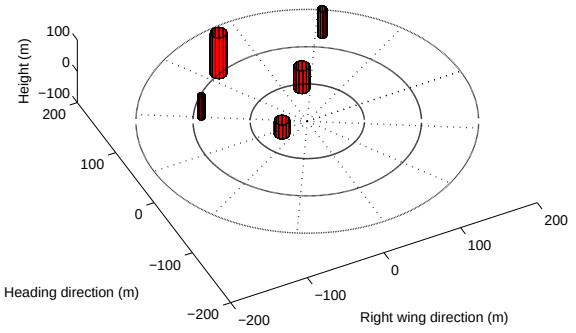


Fig. 4 This figure shows the local-level frame map in cylindrical coordinates. The origin of the map is the current location of the MAV. The cylinders represent the obstacles.

4 Path planning in the local-level frame

Given the local-level frame map, a standard planning method, like potential field methods, sample based planning methods, or cell decomposition methods, can be used as the local planning algorithm that generates collision-free paths that drive the MAV to a region of the goal configuration while avoiding the obstacles.

As a specific example, in this paper we propose to use the Rapidly-Exploring Random Tree (RRT) algorithm that generates collision-free Dubins paths, which are flyable for the MAV. The RRT algorithm was initially developed in [20] and has been widely used in robot path planning. The RRT algorithm can quickly search the space of possible solutions by extending a tree in random directions in the configuration space. When the RRT algorithm is employed to plan a path for the MAV, the nodes of the tree are potential MAV configurations (position and heading) and the branches are paths to the configurations. The tree initially consists of the MAV location as a single node. A random configuration is generated and the tree is extended toward that configuration, creating a new branch and node. When a path is found or a maximum number of iterations is reached, the RRT algorithm terminates.

The RRT algorithm can easily handle kinematic constraints since these constraints are considered when adding nodes to the tree. In the paper, we plan Dubins paths using the RRT algorithm to ensure that the paths are kinematically feasible. A Dubins path is defined as the path with the shortest path length [21, 22]. When the RRT algorithm is used to plan Dubins paths, branches between tree nodes are the Dubins paths and collision check relies on the Dubins paths instead of the straight lines.

Let q_f represent a goal configuration and define the goal region $G(q_f) \triangleq \{q \in \mathscr{C}_{free} : \|q - q_f\|_{2D} \leq R^l \text{ and } |q_d - q_{fd}| \leq \frac{H}{2}\}$. Let $\partial M(q(t)) = \{q' \in \mathscr{C} : \|q' - q(t)\|_{2D} = R^l \text{ and } |q'_d - q_d(t)| \leq \frac{H}{2}\}$ represent the boundary of $M(q(t))$. The idea of the local path planning algorithm is to plan a collision-free path to a configuration on $\partial M(q(t))$ such that the distance between the MAV and goal configurations decreases and that the MAV will eventually be ma-

neuvered to the goal region $G(q_f)$. Figure 5 shows the geometry of the relative positions of the local-level frame map centered at the current MAV configuration $q(t) = [q_n(t), q_e(t), q_d(t), q_\psi(t)]^\top$ and the goal configuration $q_f = [q_{fn}, q_{fe}, q_{fd}, q_{f\psi}]^\top$. The cylinder $C(q(t))$ is centered at $[q_{fn}, q_{fe}, (q_{fd} + q_d(t))/2]^\top$, and its radius and height are $\|q(t) - q_f\|_{2D}$ and $|q_d(t) - q_{fd}|$. The cylinders $M(q(t))$ and $C(q(t))$ intersect at line $\overline{q_I q_I''}$ and line $\overline{q_I' q_I'''}$.

Let $S(q(t))$ represent the set of configurations that are on $\partial M(q(t))$ and that are contained in $C(q(t))$, as shown by the red area in Fig. 5. If the MAV configuration $q(t)$ satisfies $\|q(t) - q_f\|_{2D} > R^l$ and satisfies $|q_d(t) - q_{fd}| > \frac{H}{2}$, i.e. the MAV is outside of the goal region, for a configuration $q' \in S(q(t))$, it must be that $\|q' - q_f\|_{2D} \leq \|q(t) - q_f\|_{2D}$ and $|q_d' - q_{fd}| \leq |q_d(t) - q_{fd}|$. Let $\varepsilon_r < R^l$ and $\varepsilon_d < \frac{H}{2}$ represent two positive numbers. Define $U(q(t)) \triangleq \{q' \in S(q(t)) : \|q(t) - q_f\|_{2D} - \|q' - q_f\|_{2D} \geq \varepsilon_r \text{ and } |q_d(t) - q_{fd}| - |q_d' - q_{fd}| \geq \varepsilon_d\}$ as the set of configurations in $S(q(t))$ such that the distance between each configuration in $U(q(t))$ and the goal is closer than the distance between $q(t)$ and the goal by a finite amount.

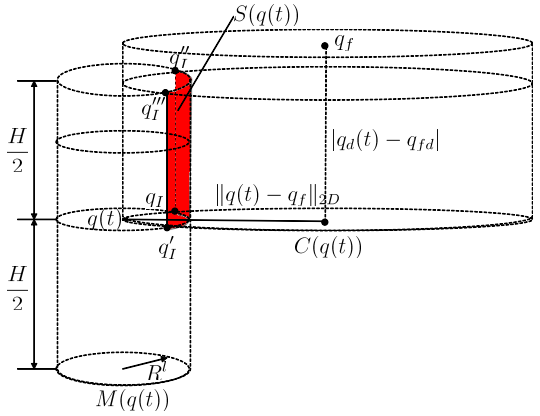


Fig. 5 This figure shows the geometry of relative positions of the local-level frame map centered at the MAV configuration $q(t)$ and the goal q_f. The cylinder $M(q(t))$ is the local-level frame map with the radius R^l and height H. The cylinder $C(q(t))$ is centered at $[q_{fn}, q_{fe}, \frac{q_{fd}+q_d(t)}{2}]^\top$ and its radius and height are $\|q(t) - q_f\|_{2D}$ and $|q_d(t) - q_{fd}|$. The cylinders $M(q(t))$ and $C(q(t))$ intersect at line $\overline{q_I q_I''}$ and line $\overline{q_I' q_I'''}$.

The local path planner uses the RRT algorithm to generate a collision-free Dubins path to a configuration in $U(q(t))$ from the current MAV configuration $q(t)$, as shown in Algorithm 1, so that the distance between the MAV and goal configurations decreases. In line 1 a tree T^l is initialized to contain one node − the current MAV configuration $q(t)$. The while loop in Lines 2-10 adds nodes to the tree T^l until a configuration in $U(q(t))$ is included to the tree or the maximum number of iterations is reached. In Line 3 a random configuration q_{rand} is uniformly drawn from $M(q(t)) \setminus U(q(t))$ with probability P and from $U(q(t))$ with probability $1 - P$. Line 4 finds the node q_{near} in the tree T^l that is closest to q_{rand} and Line 5 finds the Dubins path between q_{near} and q_{rand}. If the Dubins path is feasible and satisfies the maximum flight path angle constraints as checked in Line 6, q_{rand} is added to the tree T^l as a tree node in Line 7 and the Dubins path is added to T^l as a tree edge in Line 8. Once the iteration loop has been executed, the algorithm checks whether a configuration in $U(q(t))$ is connected to the tree T^l in Line 11. If such a configuration is connected to T^l, the path from $q(t)$ to that configuration is extracted in Line 12. Otherwise, a tree node is randomly picked and a path from $q(t)$ to that tree node is extracted in Line 15.

Algorithm 1: *plan_RRT_Dubins*

```
1  Initialize the tree T^l so that it only contains q(t);
2  while No configuration in U(q(t)) is connected to the
   tree and the maximum number of iteration is not
   reached do
3      Uniformly draw a random configuration q_rand
       from M(q(t)) \ U(q(t)) with probability P and
       from U(q(t)) with probability 1 − P;
4      Find the node q_near in the tree T^l that is closest to
       q_rand;
5      Find the Dubins path between q_near and q_rand;
6      if the Dubins path is feasible and satisfies the
       maximum flight path angle constraints then
7          Add q_rand as a tree node to T^l;
8          Add the Dubins path as a tree edge to T^l;
9      end
10 end
11 if A configuration in U(q(t)) is connected to the tree
   T^l then
12     Extract the path from q(t) to that configuration;
13 end
14 else
15     Randomly pick a tree node and extract the path
       from q(t) to that tree node;
16 end
```

Note that for some environments there may not always exist a configuration in $U(q(t))$ that leads to a collision-free path from $q(t)$. In Section 5 we will determine the characteristics of the environments in which there exist configurations in $U(q(t))$ that result in collision-free paths from $q(t)$ for all time t given a suitable initial MAV configuration.

The path generated by the RRT algorithm may include numerous extraneous configurations that may cause unnecessary maneuvering, and should therefore be smoothed. We design a fast and effective configuration elimination algorithm to smooth the path. This algorithm walks through the path from the beginning to the end nodes, eliminating un-

necessary nodes along the way. It initially looks at the first node and tries to find the last node in the path to which the first node can be directly connected without collision. It is guaranteed to at least connect to one node in the graph. If the only node that can be connected is the next node in the path, the algorithm moves forward one step and tries to connect the second node to the last possible node. If the first node can be connected to any other node, all intermediate nodes are eliminated and the algorithm moves forward to the connecting node and repeats the process until a connection to the end node is found. Using this algorithm, all intermediate nodes that can be skipped without causing any collision are eliminated from the path.

5 Analysis

In this section, we analyze the behavior of the local planning algorithm. We first describe the characteristics of the environments in which the algorithm is guaranteed to generate collision-free paths for the MAV and we then illustrate under what environments the MAV is guaranteed to be maneuvered to the goal region using the local planning algorithm. We focus our analysis on the environments with cylindrical obstacles. The analysis is based on the following assumption.

Assumption 1 *The local path planning algorithm is complete and is guaranteed to find a collision-free path in finite time when one exists in the local map.*

By this assumption, our analysis applies to any planning algorithm that is guaranteed to generate a collision-free path in finite time when there exists one in the environment. As shown in [23], the RRT algorithm is probabilistically complete, which means the probability that the RRT algorithm is guaranteed to find a collision-free path when one exists goes to one as the number of the RRT nodes goes to infinity. References [23, 24] show the probability converges to one exponentially fast with the number of random samples used to construct the tree. Reference [3] develops a randomized planning algorithm that is a variant of the RRT algorithm, and shows the bound of convergence rate in terms of geometric complexity of the environment, which cannot be measured easily for nontrivial environments. Characterizing the convergence rate of the RRT algorithm in terms of simple parameters that can be checked remains an open problem. However, our experience is that the local planner using the RRT algorithm consistently finds collision-free paths for multiple obstacle avoidance scenarios in finite time.

5.1 Collision Avoidance

For collision avoidance, we design the local planning algorithm that searches a collision-free path in the local-level frame map until one is found. We analyze collision avoidance behavior of the local planning algorithm that satisfies Assumption 1. To guarantee collision avoidance with a cylindrical obstacle, it is necessary to establish a region around the obstacle outside which the MAV is guaranteed to avoid the obstacle using the local planning algorithm with feasible inputs. Let q_{O_i} represent the center of an obstacle O_i and let H_{O_i} represent the height of the obstacle. Define

$$F(q_{O_i}, R_c^i, H_{O_i}) \triangleq \{q' \in \mathscr{C} : \|q' - q_{O_i}\|_{2D} \leq R_c^i \text{ and } q'_d \geq -H_{O_i}\} \tag{16}$$

as a cylinder region centered at q_{O_i} with the radius R_c^i and with the height H_{O_i}. Let $\phi_{\max}$ and $\gamma_{\max}$ represent the maximum roll and flight path angles of the MAV. The minimum turning radius of the MAV is given by [21]

$$r_{mt} = \frac{V^2 \cos\gamma_{\max}}{g \tan\phi_{\max}}. \tag{17}$$

Since the distance between the MAV and an obstacle increases if the absolute value of the bearing to the obstacle is greater than $\frac{\pi}{2}$, we say that the MAV successfully avoids the obstacle if the MAV flies to a configuration with the bearing equal to $\frac{\pi}{2}$ or $-\frac{\pi}{2}$ from an initial configuration without causing collision with the obstacle.

Lemma 1 *Given Assumption 1, if the initial MAV configuration q_0 is not contained in the cylinder region $F(q_{O_i}, R^i_{c\min}, H_{O_i})$ given by Eq.* (16) *with the radius*

$$R^i_{c\min} = \sqrt{(R_i + r_{mt})^2 - r_{mt}^2} \tag{18}$$

around a cylindrical obstacle O_i with the radius R_i and with the height H_{O_i}, where r_{mt} is the minimum turning radius of the MAV given by Eq. (17)*, then the local planning algorithm guarantees a collision free path from q_0 to a configuration with the bearing angle equal to $\frac{\pi}{2}$ or $-\frac{\pi}{2}$ that is arbitrarily close to the boundary of the obstacle.*

Proof Consider the scenario where a MAV is flying at a cylindrical obstacle O_i with a negative bearing angle η_0, as shown in Fig. 6. The minimum distance to the obstacle by which the MAV has capability to avoid the obstacle can be determined when the maximum roll angle and the maximum flight path angle are applied and the generated circle with the minimum turning radius r_{mt} is tangent to the boundary of the cylindrical obstacle at q'. Based on the geometry, the minimum distance is given by $d = -r_{mt}\sin|\eta_0| + \sqrt{r_{mt}^2 \sin^2\eta_0 + R_i^2 + 2R_i r_{mt}} - R_i$. We can see that when the MAV with an initial configuration q_0 outside the cylinder region $F(q_{O_i}, R_i + d, H_{O_i})$ flies at the obstacle with the negative bearing angle η_0, there must exist a configuration with the bearing angle equal to $-\frac{\pi}{2}$, which leads to a collision-free path from q_0 and which is arbitrarily close to the boundary of the obstacle. Similarly, for $\eta_0 > 0$, there must exist such a configuration

with the bearing angle equal to $\frac{\pi}{2}$. When $\eta_0 = 0$, d reaches its maximum value $d_{\max} = \sqrt{(R_i + r_{mt})^2 - r_{mt}^2} - R_i$. Let $R^i_{c\min} = d_{\max} + R_i = \sqrt{(R_i + r_{mt})^2 - r_{mt}^2}$. Therefore, if the initial MAV configuration q_0 is not contained in the cylinder region $F(q_{O_i}, R^i_{c\min}, H_{O_i})$, there must exist a collision-free path from q_0 to a configuration with the bearing angle equal to $\frac{\pi}{2}$ or $-\frac{\pi}{2}$ that is arbitrarily close to the boundary of the obstacle. In addition, since the local planning algorithm satisfies Assumption 1, it is guaranteed to find the collision-free path. □

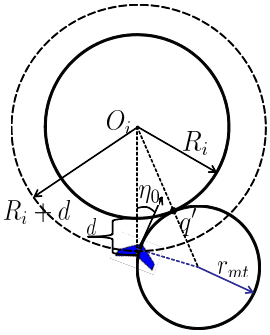


Fig. 6 This figure shows the geometry of the scenario projected onto x-y plane of the inertial frame where the MAV is flying at a cylindrical obstacle with the bearing angle η_0 and avoiding the obstacle by banking in right direction.

For environments with multiple cylindrical obstacles, we specify the conditions under which the local planning algorithm guarantees collision-free paths for the MAV. We refer the cylinder region $F(q_{O_i}, R^i_{c\min}, H_{O_i})$ of the i^{th} obstacle to the *unsafe region* for that obstacle and define the shortest distance between the points on the boundaries of the i^{th} and the j^{th} obstacles as

$$d_{ij} \triangleq \min_{p_i \in \partial O_i, p_j \in \partial O_j} \sqrt{(p_{in} - p_{jn})^2 + (p_{ie} - p_{je})^2}, \tag{19}$$

where $p_i = [p_{in}, p_{ie}, p_{id}]^\top$ and $p_j = [p_{jn}, p_{je}, p_{jd}]^\top$. Let $\mathscr{I}$ represent the index set of all obstacles. We introduce the notion of a *passable* environment.

Definition 1 An environment is said to be *passable* if $d_{ij} > \max\{R^i_{c\min} - R_i, R^j_{c\min} - R_j\}$ for every i and j in $\mathscr{I}$, where d_{ij} is the distance between the i^{th} and j^{th} obstacles given by Eq. (19), R_i and R_j are the radii of the i^{th} and the j^{th} obstacles, and $R^i_{c\min}$ and $R^j_{c\min}$ are the radii of the unsafe region for the i^{th} and the j^{th} obstacles given by Eq. (18).

In other words, an environment is passable if every pair of obstacles is separated by a distance that is greater than the maximum of the differences between the radii of their unsafe region and their radii. This means that no points on the boundary of an obstacle are contained in the unsafe region of any other obstacle in the environment. Theorem 1 describes the collision avoidance behavior of the planning algorithm.

Theorem 1 *Given Assumption 1, if the environment is* passable *and if the initial MAV configuration q_0 is not contained in the unsafe region $F(q_{O_i}, R^i_{c\min}, H_{O_i})$, $\forall i \in \mathscr{I}$, then the MAV will remain in $\mathscr{C}_{free}$ for all time t using the local planning algorithm.*

Proof Suppose that the MAV is initially located at q_0, which is not contained in the unsafe region $F(q_{O_i}, R^i_{c\min}, H_{O_i})$, $\forall i \in \mathscr{I}$, and suppose that the MAV will collide with the obstacle O_i if it flies along its initial heading, as shown in Fig. 7. Since q_0 is not contained in the unsafe region $F(q_{O_i}, R^i_{c\min}, H_{O_i})$, $\forall i \in \mathscr{I}$, based on Lemma 1, there exists a collision-free path from q_0 to a configuration q_A with the bearing angle equal to $\frac{\pi}{2}$ or $-\frac{\pi}{2}$, where q_A is arbitrarily close to O_i. In addition, since the environment is passable, which implies $d_{ij} > \max\{R^i_{c\min} - R_i, R^j_{c\min} - R_j\}$ for every i and j in $\mathscr{I}$, the configuration q_A must be outside the unsafe region $F(q_{O_j}, R^j_{c\min}, H_{O_j})$, $\forall j \in \mathscr{I} \setminus \{i\}$. Accordingly, there exists a collision-free path from q_A to a configuration q_B with the bearing angle equal to $\frac{\pi}{2}$ or $-\frac{\pi}{2}$, where q_B is arbitrarily close to O_j and where q_B is also outside $F(q_{O_k}, R^k_{c\min}, H_{O_k})$, $\forall k \in \mathscr{I} \setminus \{j\}$. This process can be repeated so that there always exist collision-free paths. Since the local planning algorithm satisfies Assumption 1, it is guaranteed to find the collision-free paths. Therefore, the MAV will remain in $\mathscr{C}_{free}$ for all time t using the local planning algorithm. □

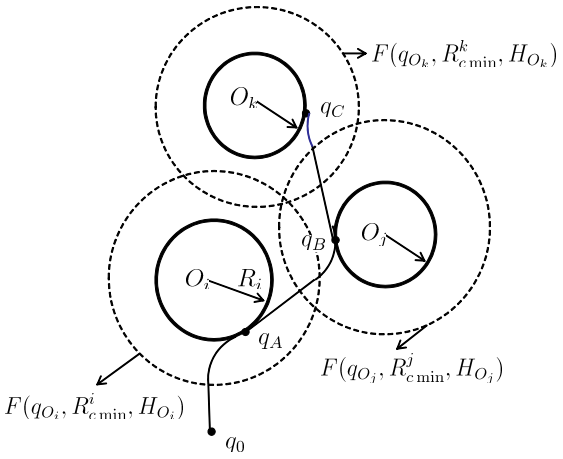


Fig. 7 This figure shows the local path planning algorithm generates a collision-free path for the MAV operating in the environment where the distance between each two obstacles is greater than the maximum of the differences between the radii of their unsafe region and their radii.

5.2 Goal Reaching

In the previous section, we derived the conditions under which the local planning algorithm guarantees collision-free paths for the MAV. Now suppose that the objective is not just to avoid obstacles, but to move to a region of a goal configuration, where the MAV knows its own configuration and the goal configuration in the inertial frame.

The proposed local planning algorithm is designed in the local-level frame to maneuver the MAV from an initial configuration $q_0 \in \mathscr{C}_{free}$ to a configuration in the goal region $G(q_f) = \{q \in \mathscr{C}_{free} : \|q - q_f\|_{2D} \leq R^l \text{ and } |q_d - q_{fd}| \leq \frac{H}{2}\}$ as defined in Section 4. We introduce the notion of *global convergence* to describe the performance of the local planning algorithm for goal reaching.

Definition 2 Given Assumption 1, for an initial configuration $q_0 \in \mathscr{C}_{free}$ and a goal configuration $q_f \in \mathscr{C}_{free}$, the local planning algorithm is *globally convergent* if it finds a feasible path from q_0 to the goal region $G(q_f)$ without collisions.

Our objective is to specify the conditions under which the local planning algorithm satisfying Assumption 1 achieves global convergence. Using a camera, we assume that the camera measurements are accurate enough so that the collision-free local path generated is safe until the MAV has finished following the local path.

To guarantee goal reaching performance, we require the local planning algorithm generates a path to a configuration in $U(q(t))$ from the MAV configuration $q(t)$ at all time t. To achieve this goal, we require (a) the separation between obstacles is large enough so that the local planning algorithm is guaranteed to find collision-free paths from $q(t)$ to collision-free configurations on the boundary of the local-level frame map; (b) the planning horizon, which is the radius of the local map, is large enough so that there exist such collision-free configurations in $U(q(t))$.

For achieving (a), if the MAV can avoid each obstacle by banking in either direction, it will fly between each two obstacles without collisions and will eventually reach the boundary of the local map. Accordingly, we establish a cylinder region around an obstacle outside which the MAV is guaranteed to avoid the obstacle by banking in either direction using the local planning algorithm.

Lemma 2 *Given Assumption 1, if the initial MAV configuration q_0 is not contained in the cylinder region $F(q_{O_i}, R_i + 2r_{mt}, H_{O_i})$ around a cylindrical obstacle O_i with the radius R_i and with the height H_{O_i}, where r_{mt} is the minimum turning radius of the MAV given by Eq.* (17)*, then the local planning algorithm guarantees a collision-free path from q_0 to a configuration with the bearing angle equal to $\frac{\pi}{2}$ and a collision-free path from q_0 to a configuration with the bearing angle equal to $-\frac{\pi}{2}$, where the two configurations are arbitrarily close to the boundary of the obstacle.*

Proof Consider the scenario where the MAV is flying at a cylindrical obstacle O_i with a negative bearing angle η_0, as shown in Fig. 8. The minimum distance to the obstacle when the MAV has capability to avoid the obstacle by banking in left direction can be determined when the maximum roll and flight path angles are applied and the generated circle with the minimum turning radius r_{mt} is tangent to the obstacle boundary at q''. Based on the geometry, the minimum distance is given by $d' = r_{mt}\sin|\eta_0| + \sqrt{r_{mt}^2\sin^2\eta_0 + R_i^2 + 2R_i r_{mt}} - R_i$, which is greater than the minimum distance $d = -r_{mt}\sin|\eta_0| + \sqrt{r_{mt}^2\sin^2\eta_0 + R_i^2 + 2R_i r_{mt}} - R_i$ by which the MAV has capability to avoid the obstacle by banking in right direction. Accordingly, if the MAV with an initial configuration q_0 outside the cylinder region $F(q_{O_i}, R_i + d', H_{O_i})$ flies at the obstacle with the negative bearing angle η_0, there must exist a configuration with the bearing angle equal to $\frac{\pi}{2}$, which leads to a collision-free path from q_0 and which is arbitrarily close to the boundary of the obstacle. In addition, there must exist a configuration with the bearing angle equal to $-\frac{\pi}{2}$, which leads to a collision-free path from q_0 and which is arbitrarily close to the boundary of the obstacle. When $\eta_0 = \frac{\pi}{2}$, d' reaches its maximum value $2r_{mt}$. Therefore, if the initial MAV configuration q_0 is not contained in the cylinder region $F(q_{O_i}, R_i + 2r_{mt}, H_{O_i})$, there must exist a collision-free path from q_0 to a configuration with the bearing angle equal to $\frac{\pi}{2}$ and a collision-free path from q_0 to a configuration with the bearing angle equal to $-\frac{\pi}{2}$, where the two configurations are arbitrarily close to the boundary of the obstacle. Since the local planning algorithm satisfies Assumption 1, it is guaranteed to find the collision-free paths. □

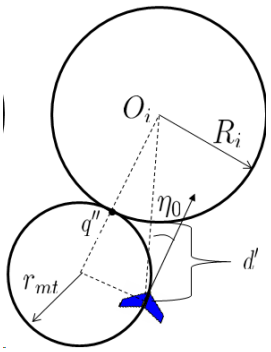


Fig. 8 This figure shows the geometry of the scenario projected onto *x-y* plane of the inertial frame where the MAV is flying at a cylindrical obstacle with the bearing angle η_0, and avoiding the obstacle by banking in left direction.

We say that a collision-free path goes through two obstacles if there exist a configuration with bearing angle to one obstacle equal to $\frac{\pi}{2}$ (-$\frac{\pi}{2}$) and a configuration with bearing angle to the other obstacle equal to $-\frac{\pi}{2}$ ($\frac{\pi}{2}$) along the

path, as shown in Fig. 9. Lemma 3 shows the environment

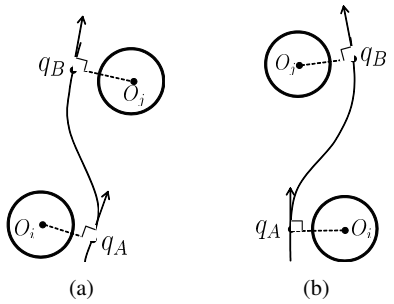


Fig. 9 This figure shows the examples of the collision-free paths that go through two obstacles.

where the local planning algorithm guarantees collision-free paths that go through each two obstacles and that reach the boundary of the local map.

Lemma 3 *Given Assumption 1, if* $d_{ij} > 2r_{mt},\ \forall i, j$, *and if the initial MAV configuration* q_0 *is not contained in the cylinder region* $F(q_{O_i}, R_i + 2r_{mt}, H_{O_i}),\ \forall i \in \mathscr{I}$, *where* r_{mt} *is the minimum turning radius given by Eq.* (17)*,* d_{ij} *is the distance between the* i^{th} *and* j^{th} *obstacles given by Eq.* (19)*, then the local planning algorithm guarantees collision-free paths that go through each two obstacles and that reach the boundary of the local-level frame map for all time* t.

Proof Since the initial MAV configuration q_0 is not contained in the cylinder region $F(q_{O_i}, R_i + 2r_{mt}, H_{O_i}),\ \forall i \in \mathscr{I}$, based on Lemma 2, there exists a collision-free path from q_0 to a configuration q_A with the bearing angle to an obstacle O_i equal to $\frac{\pi}{2}$, where q_A is arbitrarily close to the boundary of the obstacle O_i. In addition, since $d_{ij} > 2r_{mt},\ \forall i, j$, q_A must be outside the cylinder region $F(q_{O_j}, R_j + 2r_{mt}, H_{O_j}),\ \forall j \in \mathscr{I} \setminus \{i\}$. Accordingly, there exists a collision-free path from q_A to a configuration q_B with the bearing angle to another obstacle O_j equal to $-\frac{\pi}{2}$, where q_B is arbitrarily close to the boundary of O_j. This means there exists a collision-free path that goes through the obstacle O_i and the obstacle O_j. This process can be repeated so that there exists a collision-free path that goes through each two obstacles for all time t. Therefore, the local planning algorithm that satisfies Assumption 1 is guaranteed to find those collision-free paths and the algorithm will drive the MAV to the boundary of the local-level frame map for all time t. □

Besides the separation between obstacles, we also require the planning horizon (the radius of the local map) is large enough so that $U(q(t))$ contains configurations to which the local planning algorithm is guaranteed to drive the MAV. We define the notion of *local sparseness* of an environment as follows.

Definition 3 An environment is said *locally sparse* if it satisfies both: (a) $d_{ij} > 2r_{mt},\ \forall i, j$; (b) $\max_i R_i + 2r_{mt} + \varepsilon_l < R^l$, where r_{mt} is the minimum turning radius given by Eq. (17), d_{ij} is the distance between the i^{th} and j^{th} obstacles given by Eq. (19), R_i is the radius of the i^{th} obstacle, R^l is the size of the planning horizon, and ε_l is a positive number.

In Definition 3, condition (a) places a restriction on separation between obstacles and requires that the distance between each two obstacles is greater than $2r_{mt}$, and condition (b) places a restriction on the size of planning horizon.

Theorem 2 *Given Assumption 1, if an environment is locally sparse and if an initial MAV configuration* q_0 *that is not contained in the cylinder region* $F(q_{O_i}, R_i + 2r_{mt}, H_{O_i}),\ \forall i \in \mathscr{I}$, *then the local planning algorithm is globally convergent for the initial MAV configuration* q_0 *and a goal configuration* $q_f \in \mathscr{C}_{free}$.

Proof The local sparseness of the environment implies $d_{ij} > 2r_{mt},\ \forall i, j$. Based on Lemma 3 the local planning algorithm guarantees a collision-free path to a configuration $q(t)$, which is arbitrarily close to the boundary of an obstacle O_i and which is outside the cylinder region $F(q_{O_j}, R_j + 2r_{mt}, H_{O_j}),\ \forall j \in \mathscr{I} \setminus \{i\}$, at time t from an initial MAV configuration q_0 outside $F(q_{O_i}, R_i + 2r_{mt}, H_{O_i}),\ \forall i \in \mathscr{I}$. Accordingly, the proof begins with the case where the MAV is located at $q(t)$.

Figure 10 shows the geometry of the relative position of the local map centered at $q(t)$ and the goal q_f projected onto the x-y plane of the local-level frame, where q_s is the intersection point of $\overline{q(t)q_f}$ and the boundary of $M(q(t))$. It is obvious that q_s is the middle point on the arc from q_I to q_I'. Let η_f represent the inertial angle to q_f. Based on the geometry, the two dimensional distance between the intersection points q_s and q_I is given by $\|q_I - q_s\|_{2D} = \sqrt{2}R^l\sqrt{1 - \frac{R^l}{2\|q_f - q(t)\|_{2D}}}$ and decreases as $\|q_f - q(t)\|_{2D}$ decreases. When $\|q_f - q(t)\|_{2D} = R^l$, which means the MAV reaches the goal region, $\|q_I - q_s\|_{2D}$ reaches its minimum value R^l. That is, $\|q_I - q_s\|_{2D} \geq R^l$.

Consider the worst case scenario that an obstacle O_j with the maximum radius is located at q_s. Since $d_{ij} > 2r_{mt},\ \forall i, j$, based on Lemma 3 there exist collision-free paths from $q(t)$ that go between each two obstacles on the boundary of the local map. If the distance between the obstacle and q_I, which is $\|q_I - q_s\|_{2D}$, is greater by a finite amount than the total length of R_j and $2r_{mt}$, there must exist a collision-free path from $q(t)$ to the collision-free portion on the arc from q_I to q_s, which is contained in $U(q(t))$, by choosing a suitable finite amount. Since the environment is locally sparse, which implies $\max_j R_j + 2r_{mt} + \varepsilon_l < R^l$, and $\|q_I - q_s\|_{2D} \geq R^l$, there exists a collision-free path from $q(t)$ to $U(q(t))$ for all time t. For the case where the obstacle is

located at any configuration other than q_s, the longer distance between the obstacle and q_I or q'_I must be greater than $\|q_I - q_s\|_{2D}$ in the worst case scenario. Therefore, the longer distance is greater than $\max_j R_j + 2r_{mt} + \varepsilon_l$, which means there must exist a collision-free path from $q(t)$ to $U(q(t))$ for all time t. In addition, since the local planning algorithm satisfies Assumption 1, it is guaranteed to find such a path to $U(q(t))$. This process is repeated so that the distance between the MAV and the goal configurations decreases as time progresses. The MAV will eventually reach the goal region. □

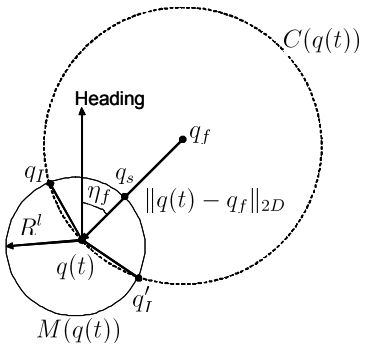


Fig. 10 This figure shows the geometry of relative positions of the local map centered at $q(t)$ and the goal configuration q_f projected onto the x-y plane of the local-level frame.

The passability and local sparseness assumptions are used to provide a theoretical guarantee in terms of environment properties, which can be checked prior to the flight, for collision avoidance and goal reaching behaviors of the local planning algorithm. However, we note that the assumptions are only sufficient conditions for collision avoidance and goal reaching, which implies there may exist environments that do not satisfy the assumptions but where the planning algorithm can still maneuver the MAV to the goal without causing collisions.

6 Numerical results

The feasibility of the vision-based mapping and planning algorithm was tested using a simulation environment developed in MATLAB/SIMULINK. The simulator uses a six degree-of-freedom model for the aircraft, where a North-East-Down (NED) coordinate system is used. We tested the algorithm in two scenarios as described below. We also conducted Monte Carlo simulations for testing the collision avoidance and goal reaching behaviors of the algorithm in environments with varying minimum distance between obstacles to demonstrate that the behaviors degrade gracefully as we relax the passability and local sparseness conditions. In the simulations, the maximum roll angle and the maximum flight path angle for the MAV were 30° and 15°.

6.1 Scenario I

In the first simulation scenario, the MAV was maneuvered through twenty-five buildings between waypoint **S** (0,100,-20) and waypoint **E** (600,700,-80), as shown in Fig. 11, where the square and cross signs represent the waypoints **S** and **E**. The heights of the buildings were randomly generated. A 20×20 pixel depth map was used. The parameters for the sweet spot measurement model were set at $a_0 = 0.1528$, $a_1 = 0.002$, $a_2 = 0$, $a_3 = 0.000076$, and $a_4 = 0.000076$. The covariance matrix of the process noise for each obstacle was $\mathbf{Q}_i = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 0.0076 & 0 \\ 0 & 0 & 0.0076 \end{bmatrix}$ and the ground speed was $V = 10$ m/s.

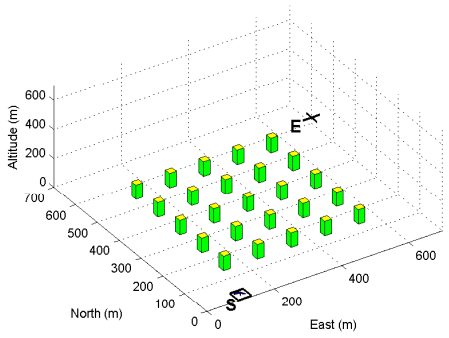
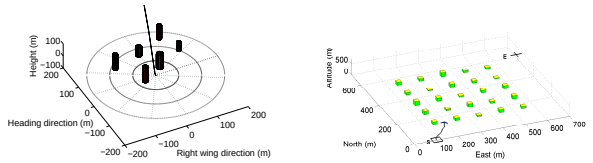


Fig. 11 This figure shows the terrain for the first simulation scenario. The MAV is maneuvered through 25 obstacles between waypoints **S** and **E**, which are represented by square and cross signs.

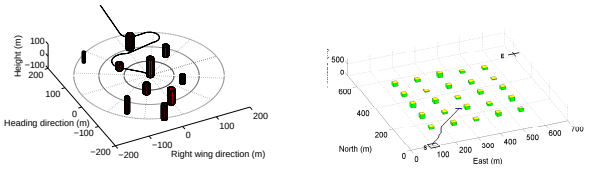
Figure 12 shows the update of the local-level frame map in cylindrical coordinates and the evolution of the path. Subfigures on the left show the local-level frame maps and Dubins paths based on the available information about the obstacles at different time. Subfigures on the right show the inertial paths followed by the MAV. Figure 13 shows the altitude of the MAV during the entire flight. Figure 14 shows the tracking error for the range, azimuth to, and height of the obstacle located at $(150, 250)$ using the EKF.

6.2 Scenario II

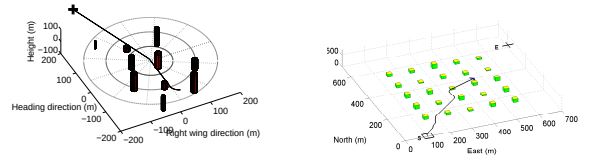
In the second simulation scenario, the MAV was commanded to maneuver through a simulated city called *Megacity* as shown in Fig. 1. The path followed waypoint $(-200, -300, -20)$ to waypoint $(250, 150, -80)$. A 640×480 pixel depth map was used. The parameters for the



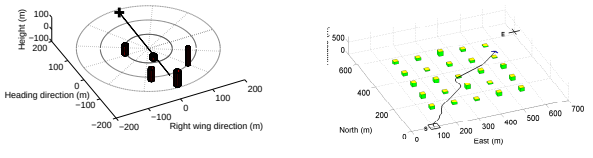
(a) $t = 10s$



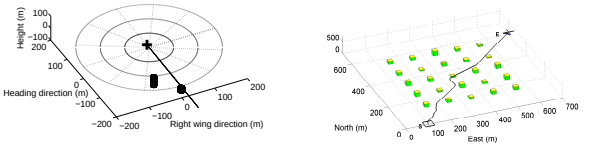
(b) $t = 28s$



(c) $t = 45s$



(d) $t = 57s$



(e) $t = 70s$

Fig. 12 This figure shows the update of the local-level frame map in cylindrical coordinates and the evolution of the path for the first simulation scenario. Subfigures on the left show the local-level frame maps and paths based on the available information about the obstacles at different time. Subfigures on the right show the actual paths followed by the MAV.

sweet spot measurement model were set at $a_0 = 0.1528$, $a_1 = 0.001$, $a_2 = 0$, $a_3 = 0.00002$, and $a_4 = 0.00002$. The covariance matrix of the process noise was $\mathbf{Q}_i = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 0.000076 & 0 \\ 0 & 0 & 0.000076 \end{bmatrix}$ and the ground speed was $V = 10$ m/s.

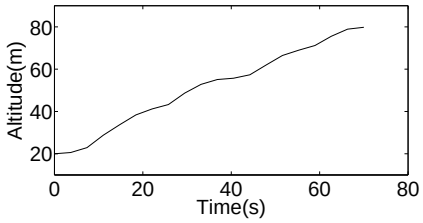
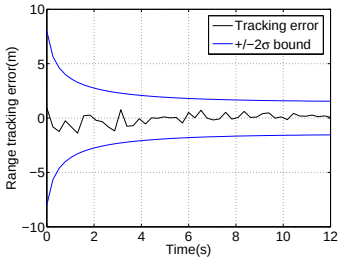
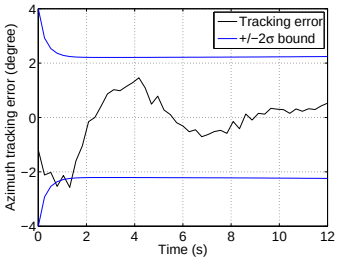


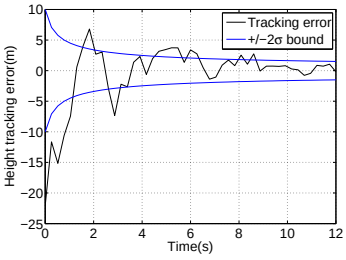
Fig. 13 This figure shows the altitude of the MAV during the entire flight.



(a) Range tracking error



(b) Azimuth tracking error



(c) Height tracking error

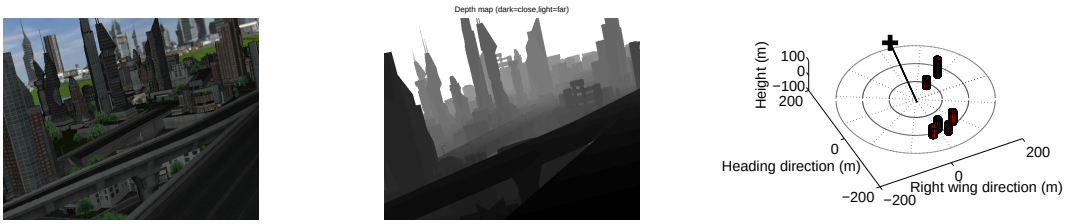
Fig. 14 This figure shows the tracking error for the range, azimuth to, and height of the obstacle located at $(150, 250)$ using the EKF.

Figure 15 shows the local-level frame maps and the corresponding collision-free paths as the MAV maneuvered through the Megacity terrain. Subfigures in the first and second columns show the camera views and the depth maps at different time. Subfigures in the third column show the update of the map and the evolution of the path in the local-

(a) $t = 10s$

(b) $t = 17s$

(c) $t = 25s$



(d) $t = 45s$

(e) $t = 51s$

Fig. 15 This figure shows the update of the local-level frame map in cylindrical coordinates and the evolution of the path for the second simulation scenario. Subfigures in the first, second, and third columns show the camera view, depth maps, and the maps and paths in the local-level frame.

level frame. Figure 16 shows the actual path followed by the MAV for the Megacity terrain.

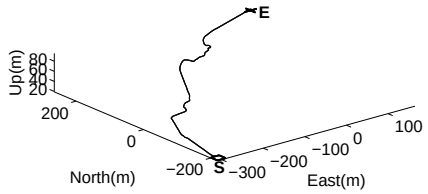


Fig. 16 This figure shows the actual path followed by the MAV using the local path planning algorithm for the second simulation scenario.

6.3 Monte Carlo simulations

The passability assumption of Theorem 1 and the local sparseness assumption of Theorem 2 are only sufficient conditions for collision avoidance and goal reaching of the local planning algorithm. When the environment does not satisfy the assumptions, the algorithm may still generate collision-free paths and maneuver the MAV to the goal region. Accordingly, we conducted Monte Carlo simulations to demonstrate this is true and to quantify the expected behavior of the algorithm.

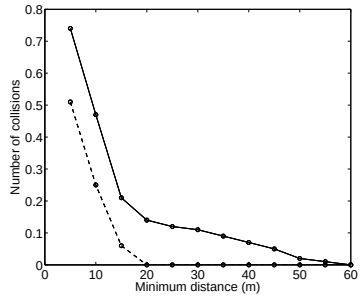
For each environment with a fixed minimum distance between obstacles, 100 simulation runs were executed. In each simulation run, the MAV was maneuvered from the initial position (0,100,-40) to the end position (580,580,-40) through an environment. The environment was constructed with each obstacle being added to the environment based on a uniform distribution over the rectangular area with the South-West corner (100,100) and the North-East corner (600,600) until no more obstacles could be added. The radius and height of all obstacles were 20 meters and 100 meters respectively. We evaluated two criteria: the number of collisions and the percentage of runs where the MAV reached the goal. We conclude that the MAV reaches the goal if it is maneuvered to the goal in $t < 100$ seconds without causing collisions.

Figure 17 plots the average number of collisions over 100 simulation runs and the percentage of runs where the MAV reached the goal versus the minimum distance between obstacles for the case where the measurement uncertainties are given by the sweet spot model with the parameters $a_0 = 0.1528$, $a_1 = 0.001$, $a_2 = 0$, $a_3 = 0.000076$, and $a_4 = 0.000076$, and for the case where the measurement uncertainties do not exist. The solid and dashed lines show the results for the cases with and without measurement uncertainties respectively. For the case with the measurement uncertainties, the number of collisions decreases dramatically as the minimum distance between obstacles increases from 5 to 20 meters. After the minimum distance is greater than 20 meters, the number of collision decreases slowly. Similar results happen to the percentage of runs where the MAV reached the goal. This is because collisions become less frequent as the environments become more sparse. Given $V = 13$ m/s, $\phi_{\max} = 30^\circ$, $\theta_{\max} = 15^\circ$, and $R_i = 20$ meters, the minimum distance that satisfies the passability condition is 19.42 meters. Accordingly, when the minimum distance is greater than or equal to 20 meters, the environments are passable and the local planning algorithm guarantees collision-free paths if the obstacle locations are perfectly known, which corresponds to the dashed line in Fig. 17 (a). Since the estimation uncertainties exist, the MAV still encounters a small number of collisions, which corresponds to the solid line in Fig. 17 (a). When the minimum distance is 60 meters, which is greater than the distance $2r_{mt} = 57.70$ meters for the local sparseness condition, the environment is locally sparse and the percentage of runs where the MAV reached the goal is 100%, as shown in Fig. 17 (b).

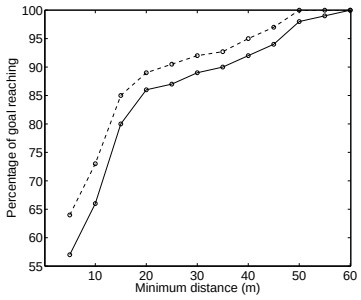
In addition, when the minimum distance between obstacles is less than 19.42 meters, the average number of collisions is less than one. This implies that the local planning algorithm generates collision-free paths for the environments that are not passable, and that the passability is only a sufficient condition for collision avoidance of the planning algorithm. Similarly, for the environments with the minimum distance less than 57.70 meters, the percentage of runs where the MAV reached the goal is nonzero. When the minimum distance is 50 meters and 55 meters, the percentage is 100%. Accordingly, the local planning algorithm can drive the MAV to the goal for the environments that are not locally sparse and the local sparseness is only a sufficient condition for goal reaching of the planning algorithm.

7 Conclusions

This paper presents a vision-based local-level frame mapping and path planning technique for collision avoidance for MAVs operating in unknown environments. We create the local-level frame maps in cylindrical coordinates for addressing the three dimensional path planning problem without transforming the camera data to the inertial frame. An EKF that takes into account the correlations between obstacles is used to jointly estimate the range, azimuth, and height for all obstacles. The data association problem is solved in the local-level frame using the joint compatibility branch and bound approach. Dubins paths are planned in the local-level frame using the RRT algorithm. We analyze the behavior of the local path planning technique and describe the



(a)



(b)

Fig. 17 This figure shows the statistical performance of the local planning algorithm implemented in the environments with varying minimum distance between obstacles for the cases with and without measurement uncertainties. Subfigure (a) plots the average number of collisions over 100 simulation runs versus the minimum distance. Subfigure (b) plots the percentage of runs where the MAV reached the goal versus the minimum distance.

characteristics of the environments in which the technique guarantees collision-free paths and drives the MAV to the goal region. Numerical results show that the proposed technique is successful in solving path planning and multiple obstacle avoidance problems for MAVs.

Acknowledgements This research was supported in part by the Air Force Research Laboratory, Munition Directorate under SBIR 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 vehicle using computer vision. Master's thesis, Brigham Young University, December, 2006.
2. A. Curtis. Path planning for unmanned air and ground vehicles in urban environments. Master's thesis, Brigham Young University, 2008.
3. E. Frazzoli, M. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control and Dynamics*, 25:116–129, Jan.-Feb. 2002.
4. Y. Watanabe, E. Johnson, and A. Calise. Vision-based approach to obstacle avoidance. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 2005.
5. A. Pongpunwattana and R. Rysdyk. Real-time planning for multiple autonomous vehicles in dynamics uncertain environments. *AIAA Journal of Aerospace Computing, Information, and Communication*, 1:580–604, December 2004.
6. K. Sedighi, K. Ashenayi, R. Wainwright, and H. Tai. Autonomous local path planning for a mobile robot using a genetic algorithm. *Congress on Evolutionary Computation*, 2:1338–1345, June 2004.
7. J. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.
8. E. Bakolas and P. Tsiotras. Multiresolution path planning via sector decompositions compatible to on-board sensor data. In *Proceedings of AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, August 2008.
9. H. Yu, R. Beard, and J. Byrne. Vision-based local multi-resolution mapping and path planning for miniature air vehicles. In *Proceedings of American Control Conference*, June 10-12 2009.
10. H. Yu, R. Beard, and J. Byrne. Vision-based local multi-resolution path planning and obstacle avoidance for micro air vehicles. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, August 10 2009.
11. H. Yu, R. Beard, and J. Byrne. Vision-based navigation frame mapping and planning for collision avoidance for miniature air vehicles. *Special Issue on Aerial Robotics, Control Engineering Practice*, 18(7):824–836, July 2010.
12. H Yu and R. Beard. Vision-based three dimensional navigation frame mapping and planning for collision avoidance for Micro Air Vehicles. In *Proceedings of AIAA Guidance, Navigation, and Control Conference*, August 2-5 2010.
13. J. Byrne and C. Taylor. Expansion segmentation for visual collision detection and estimation. In *IEEE International Conference on Robotics and Automation (ICRA'09)*, 2009.
14. R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 1st ed. edition, November 24 2010.
15. P. Yang, R. Freeman, and K. Lynch. Multi-agent coordination by decentralized estimation and control. *IEEE Transaction on Automatic Control*, 2008.
16. F. Lewis. *Optimal Estimation: With An Introduction To Stochastic*. New York: Wiley, 1986.
17. J. Neira and J. Tardos. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17, no.6:890–897, December 2001.
18. Y. Bar-Shalom and T. Fortmann. *Tracking and Data Association*. Boston, MA: Academic, 1988.
19. T. Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. PhD thesis, University of Sydney, 2002.
20. S. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical report, Computer Science Dept, Iowa State University, Tech. Rep. TR 98-11, August 1998.
21. R. W. Beard and T. W. McLain. *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012 (to appear).
22. L. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Amercian Journal of Mathematics*, 79:497–516, July 1957.
23. S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. *Algorithmic and Computational Robotics: New Directions*, pages 293–308, 2000.
24. S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. *International Journal of Robotics Research*, 20(5):378–400, May 2001.