Proceedings of the 3rd
RSI International Conference on Robotics and Mechatronics
October 7-9, 2015, Tehran, Iran

Path Planning of Nonholonomic Flying Robots Using a New Virtual Obstacle Method

Nazanin Mohamadnejad
Faculty of Engineering
Tarbiat Modares University
Tehran, 14115-143, Iran
n.mohamadnejad@gmail.com

Ellips Masehian
Faculty of Engineering
Tarbiat Modares University
Tehran, 14115-143, Iran
masehian@modares.ac.ir

Abstract— **This paper presents a method for planning near-optimal collision-free path between two pre-specified points with given orientations for nonholonomic flying robots or UAVs navigating in a 3D space amid obstacles. The proposed method is based on the Fast Marching Method (FMM), which uses a first order numerical approximation of the Eikonal equation. The FMM is combined with average filter, B-spline functions and a new tool called Virtual Obstacle to guarantee smoothness of the produced path. The Virtual Obstacle is in the form of a torus to adjust the path with orientation of the flying robot at the start and goal points. Experimental results showed that the proposed method can find near-optimal paths with maximum curvatures less than the robot's minimum flying radius.**

Keywords— Flying robot; UAV; 3D path planning; Nonholonomic constraints; Fast Marching Method; Virtual obstacle; Torus.

I. Introduction

The last decade witnessed a surge in applying unmanned aerial vehicles (UAVs) and flying robots thanks to significant advancements in both their hardware and software. In particular, opticopters and quadcopters have been applied widely in various industries such as parcel delivery (by Amazon PrimeAir and DHL), food delivery (by Domino's DomiCopter), search and rescue (by TU Delft's Ambulance drone), hurricane hunting (by NASA's Global Hawk), protecting wildlife (by the Department of the Interior, the Bureau of Land Management, and the United States Geological Service), agriculture (by Yamaha RMAX in Japan), aerial photography and filming, and much more.

While copter-type drones proved to be extremely useful thanks to their agility, simplicity, lightness, and maneuverability, they are usually designed for relatively short tours, light payloads, and normal weather conditions. That is why the conventional wing-type UAVs still have their important role in special applications like long-range missions, reconnaissance, patrolling, surveying, etc. in hostile environments. However, the kinematic constraints on instant velocities of such drones pose a challenge in planning their motions, and nonholonomic motion planning in space has been an active research topic for more than two decades. A prevailing approach in solving this problem has been through simplifications by considering either a fixed predefined altitude for the UAV (which leads to a 2D planning problem), or a constant velocity for it. In fact, while an exact solution exists just for 1D and 2D problems solvable in exponential time and polynomial space, there is no exact solution for kinodynamic planning in 3D workspace. As a result, this problem is at least NP-hard in 3D [1].

UAVs are used for both military and civilian purposes, and in both applications, finding the 'best' path that is navigable by the UAV is critically important. A path may be best according to various criteria like total distance traveled, fuel consumption, total flight time, detection threat avoidance, and navigation performance. Some of these criteria are interrelated and finding the best path is often associated with finding the shortest path. The importance of path length is obvious as by minimizing the path length, total flight time and fuel consumption will be decreased as well. Another navigation criterion in planning paths for nonholonomic UAVs is the curvature of the path, which is very important for designing navigable paths.

The kinodynamic motion-planning problem deals with computing a collision-free, time-optimal trajectory for a robot whose acceleration and velocity are bounded. This problem is at least NP-hard in three and higher dimensions. In light of this lower bound, most studies have been focused on finding approximate solutions [1].

The algorithms developed for UAV path planning can be classified into some approaches. One is Grid-based state space search, which is resolution-complete and optimal for a polynomial approximation of the problem with differential constraints, which are constraints on velocity and acceleration. Another main approach is Sampling-based path planning methods [2] which avoid tedious construction of C-obstacles by sampling the C-space. The sampling scheme may be deterministic or probabilistic [3]. In this paper we implement the uniform Cartesian deterministic sampling scheme.

While planning acceptable path curvature has been widely studied for 2D paths, few approaches exist for 3D cases. In 1957 Dubins defined Dubins' car, a 2D nonholonomic system and declared that there is an optimal trajectory between each two states which is composed of straight lines and circular arcs with minimum radius [4]. Reeds and Shepp in 1990 extended Dubins' model and allowed the backward motion to the vehicle which leads to generation of shorter paths [5]. Lavalle and Kuffner in 2002 extended nonholonomic path planning to 3D and higher-dimensional workspaces through a collection of trimmed trajectories and by applying Rapidly-exploring Random Trees to this collection [6]. Hwangbo in 2007 discretized the environment and used Dubins' trajectories to build a reachability tree for path planning of fixed-wing UAVs in a 3D environment [7]. In 2009, Koyuncu used RRT and straight lines to create waypoints and connect them by B-spline curves to generate a safe path from start to goal [8].

In this work, two criteria have been considered for planning the best path for nonholonomic UAVs: The first is *path length*, which is a commonly-addressed criterion, and the other is *path curvature*, which is a navigation performance criterion. The smaller the path curvature is, the smoother will the path be, and the flier can follow the path easier.

II. Framework of the Algorithm

A. Statement of the problem

Given a C-space Ω, the path planning problem is to find a curve:

$$\begin{aligned} C:\ & [0,1] \to \Omega \\ & s \mapsto C(s) \end{aligned}$$

In other words, this curve represents a sequence of configurations in C_{free}. An optimal path is a curve C that minimizes a set of internal and external constraints (e.g., time, fuel consumption, or danger). The set of constraints can be described in a cost function τ. In the isotropic case, τ depends only on the configuration x.

$$\begin{aligned} \tau:\ & \Omega \to R^{+}, \\ & q \mapsto \tau(q),\ \tau(q) > 0 \end{aligned}$$

Practically all sampling based methods require a metric function that defines the distance between any two configurations in the C-space, which becomes a metric space [2] . In this method the metric ρ we refer to is defined as:

$$\rho(q,q') = \int_0^1 \tau(C_{q,q'}(s))ds \qquad (1)$$

in which $C_{q,\,q'}$ is a path between two configurations q and q', and τ is the cost function which is strictly positive. This metric can be seen as the 'cost-to-go' for a specific robot to reach q' from q. At a configuration q, $\tau(q)$ can be interpreted as the cost of one step from x to its neighbors. If a C-obstacle in some region S is impenetrable, then $\tau(S)$ will be infinite. τ is supposed to be strictly positive for an obvious physical reason: $\tau(q) = 0$ would mean that free transportation from some configuration q is possible. The metric ρ is defined for a C-space assuming a continuous motion model. However, since the C-space is partitioned into a Cartesian grid, grid search algorithms commonly use a 4- or 8-connexity discrete motion model for the robot in 2D environment, and 6-connexity in 3D environment.

A discrete approximation ρ_d of metric ρ is defined as:

$$\rho_d(x,x') = \sum_{i=1}^{n} \tau(x_i)\,, \qquad (2)$$

where $x_1 = x$ and $x_n = x_0$, and transitions between x_i and x_{i+1} are governed by a discrete motion model. The mathematical model of the path panning problem for a flying robot can be expressed as a minimization problem:

$$C = \underset{C_{q,q'}}{\arg\min}\, \rho(q,q') = \underset{C_{q,q'}}{\arg\min} \int_0^1 \tau(C_{q,q'}(s))ds \qquad (3)$$

subject to:

$$\left| \frac{\partial^2 C}{\partial s^2} \right| \le \frac{1}{R_{\min}} \qquad (4)$$

$$G(q,\dot{q}) = 0 \ \ \forall\ q \in C(s)\ \ ,\ \ s \in [0,1] \qquad (5)$$

In this minimization problem the goal is to find the shortest path between the initial and final configuration with two constraints. In the objective function $C_{q,q'}(s)$ denotes all of the existing curves between the start and final points and τ denotes the cost to go from q to q'. Constraint (4) shows the limitation on path curvature in which the left hand side is the curvature magnitude of the curve $C_{q,q'}(s)$, and $R_{\min}$ is the minimum path radius that is flyable for the vehicle. Constraint (5) prevents the vehicle from moving sideways. However, moving sideways will be avoided all along the path by meeting constraint (4) except at the beginning and the end points.

The problem defined above is the model of path planning for an aerial vehicle with the curvature constraint between two prespecified configurations.

B. Creating a Smooth Path by Fast Marching Method

The Fast Marching Method (FMM) was first proposed by J. Sethian in [9] and used in image processing. The significant point about FMM is that although its complexity is similar to the classical grid-base methods, it can attain an exact solution of the minimization problem expressed in (3) [10]. Before introducing the Fast Marching algorithm itself, we start from the observation that the functional minimization problem of (3) is equivalent to solving the Eikonal equation:

$$\|\nabla u\| = \tau\,. \qquad (6)$$

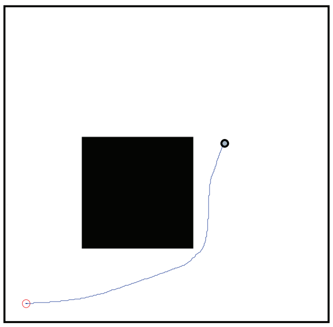
The Fast Marching algorithm uses a first order numerical approximation of the Eikonal equation (6) based on the following operators: suppose a function u is given with values $u_{i,\,j,\,k} = u(x_{i,\,j,\,k})$ on a Cartesian grid with grid spacing h. Then there is an upwind scheme used to estimate the gradient ∇u in three dimensions:

$$\|\nabla u_{i,j,k}\| = \begin{bmatrix} \max(D_{ijk}^{-x}u,0)^2 + \min(D_{ijk}^{+x}u,0)^2 + \\ \max(D_{ijk}^{-y}u,0)^2 + \min(D_{ijk}^{+y}u,0)^2 + \\ \max(D_{ijk}^{-z}u,0)^2 + \min(D_{ijk}^{+z}u,0)^2 \end{bmatrix} = \tau_{i,j,k}^2 \qquad (7)$$

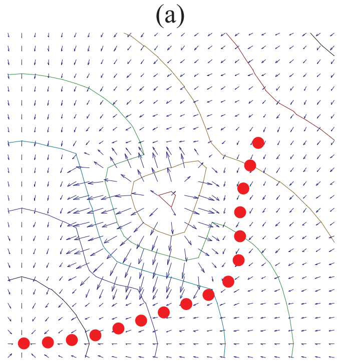
in which

$$\begin{aligned} D_{ijk}^{-x} &= \frac{u_{i,j,k} - u_{i-1,j,k}}{\Delta x}, \quad & D_{ijk}^{+x} &= \frac{u_{i+1,j,k} - u_{i,j,k}}{\Delta x} \\ D_{ijk}^{-y} &= \frac{u_{i,j,k} - u_{i,j-1,k}}{\Delta y}, \quad & D_{ijk}^{+y} &= \frac{u_{i,j+1,k} - u_{i,j,k}}{\Delta y} \\ D_{ijk}^{-z} &= \frac{u_{i,j,k} - u_{i,j,k-1}}{\Delta z}, \quad & D_{ijk}^{+z} &= \frac{u_{i,j,k+1} - u_{i,j,k}}{\Delta z} \end{aligned} \qquad (8)$$

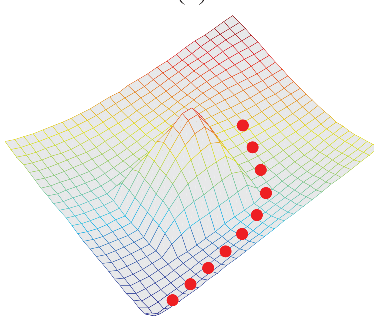
It is proved in [9] that this numerical scheme converges to a correct continuous solution. A simple algorithm based on (8) has been described in [10, 11]. The Fast Marching algorithm computes a first order estimate of the gradient of the distance function. The optimal path is naturally extracted from the goal to the start configuration by performing a gradient descent backtracking, as shown in Fig. 1.



(a)



(b)



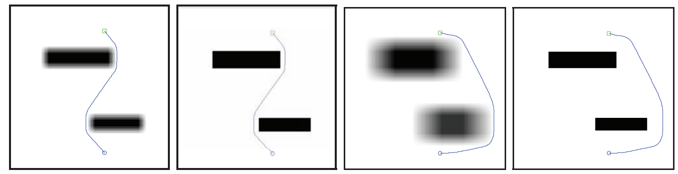
(c)

Fig. 1. Solving a path planning problem in a 2D environment by FMM (a) workspace of the problem (b) the gradient descent backtracking (c) u values obtained from FMM.

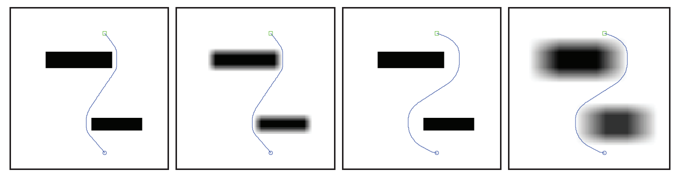
The main advantage of FMM compared to other algorithms for path planning of holonomic robots is its ability to produce smooth paths. Cohen has shown that the curvature of the path derived from FMM is related to the value of τ and its equation is as follows [12]:

$$R = \frac{1}{\kappa} \geq \frac{\inf_{\Omega} \tau}{\sup_{\Omega} \{\|\nabla \tau\|\}} \tag{9}$$

To increase the minimum radius of the curvature in (9), we can either increase the numerator or decrease its denominator. An Average Filter can be used to reduce the denominator, and also by adding τ the numerator is increased and thus we can increase the minimum radius of curvature [10]. By means of the average filter, the cost function τ will be smoothed as the value of τ in each cell is the average of its neighboring cells, as depicted in Fig. 2 for different sizes of Average Filters.



(a) (b)



(c) (d)

Fig. 2. Influence of average filter on cost function[13]: (a) applying a 5×5 average filter, R_{min} = 3.318; (b) applying a 19×19 average filter, R_{min} = 4.8165; (c) applying a 5×5 average filter and adding one unit to τ, R_{min} = 7.1248; (d) applying a 19×19 average filter and adding three units to τ, R_{min} = 7.8546.

C. Spline and Curvature Calculation

A piecewise polynomial function can have a locally very simple form, and at the same time be globally flexible and smooth. *Splines* are very useful for modeling arbitrary functions and are used extensively in computer graphics. Assume n+1 points (t_0, t_1, ..., t_n) and the condition of $t_0 < t_1 < \ldots < t_n$ is established. These points are named 'nodes'. Also, K is assumed as the power of the spline. A *spline function* in power of K and $t_0 < t_1 < \ldots < t_n$ nodes are defined as a function S such that in each interval $[t_{i-1}, t_i]$, it is a polynomial in power of less than or equal to K. The function S should be both continuous and continuously differentiable to order $K-1$ on the entire interval $[t_0, t_n]$. Therefore, S is a piecewise polynomial in maximum power of K that is continuously differentiable to order $K-1$. When the order of the spline is 3, the spline is called cubic, which guarantees smooth curves on a set of points.

Because of the kinodynamic constraints of winged UAVs, curvature of its path must not exceed an allowable value. In fact, the best path is a spline whose nodes are points yielded by the FMM. To obtain the maximum curvature value, the first-order differential of the curvature equation must be set to 0, and its second-order differential must be determined to check the maximum and minimum of the equation $\kappa'(t) = 0$.

The curvature of a parametric curve is shown as:

$$\kappa(t) = \frac{|r'(t) \times r''(t)|}{|r'(t)|^3}, \tag{10}$$

where $r(t)$ =($x(t)$, $y(t)$, $z(t)$) are defined based on cubic parametric spline equation on the interval $[t_i, t_{i+1}]$ (t_i and t_{i+1} are continuous spline nodes), as follows:

$$\begin{cases} x = a_1t^3 + b_1t^2 + c_1t + d_1 \\ y = a_2t^3 + b_2t^2 + c_2t + d_2 \\ z = a_3t^3 + b_3t^2 + c_3t + d_3 \end{cases} \quad (11)$$

Therefore, in each interval $[t_i, t_{i+1}]$ there is another equation for curvature. In fact, there are parametric curvature equations equal to the number of nodes. The maximum value of equations must be obtained to calculate the maximum value of the entire spline's curvatures, by assuming that output points in FMM are $P_i = (x_i, y_i, z_i)$, $i = 1,\ldots, n$. In order to obtain the maximum value of the path, n equations of $\kappa'(t) = 0$ must be solved.

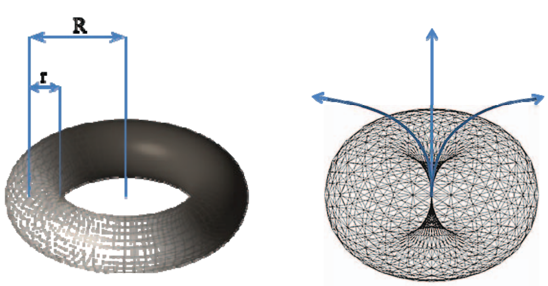
III. Employing Virtual Obstacles

In this section we present our proposed Virtual Obstacle concept, its implementation, and the method of creating a path using virtual obstacles. A Virtual Obstacle (VO) is a hypothetical obstacle placed around the flying robot in its first (start) and last (goal) configurations in order to meet its nonholonomic constraints. In fact, the VO prevents creation of sharp (non-smooth) points in the path and adjusts the path to be tangent to the robot's orientation in those configurations. The shape of the VO is a decisive factor in satisfying the dynamic constraints of the robot. For example, in our previous research [13] for Dubins' cars, two tangent circles were used as VOs to plan a navigable path with correct initial and final configurations of the robot.

In this paper, the VOs for 3D environments are proposed in the form of torus, which is a 3D manifold created by revolving a circle around an axis coplanar with it, and parametrically defined as follows:

$$\begin{aligned} x(u,v) &= (R + r\cos v)\cos u \\ y(u,v) &= (R + r\cos v)\sin u \\ z(u,v) &= r\sin v \end{aligned} \quad (12)$$

in which v, $u \in (0, 2\pi)$, R is the distance from center of the tube to axis of the torus, and r is radius of the tube. R and r are known as major and minor radii, as shown in Fig. 3:



(a) (b)

Fig. 3. (a) A torus; (b) The torus prevents the robot's motion to sideways in all directions and stipulates it to fly around with a radius more than radius of the torus.

According to [5], a nonholonomic robot must move on the perimeter of a circle with minimum allowed radius in order to move along the shortest path. In fact, the reason for proposing a torus as a VO is that it prevents the robot to move sideways in all directions. Therefore, if the minor radius (r) is considered equal to the minimum allowed radius of the curvature $R_{\min\,(allowed)}$ of the robot, then the minimum radius of curvature of the created path will be guaranteed to be larger than $R_{\min\,(allowed)}$ and so the path will be navigable by the robot.

Fig. 4 shows how two torus-shaped Virtual Obstacles are placed in the start and goal positions of the UAV, after which a path is planned by the Fast Marching and Average filter methods which has a maximum curvature larger than the turning radius of the UAV.

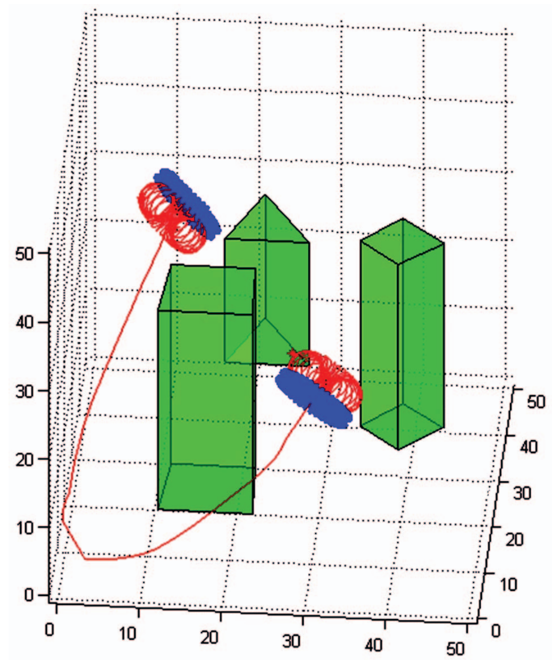


Fig. 4. A path planned by the FMM after placing two tori (toruses) (red obstacles) oriented along the UAV's initial and goal orientations. Backside surfaces of tori (blue virtual obstacles)) are used for preventing backward movement.

IV. Simulation and Results

In order to test the developed method four complex three-dimensional environments were constructed which were taken from [8], as follows: (A) narrow passage, (B) city-like environment, (C) mostly blocked environment, and (D) circuitous tunnel environment, as depicted in Fig. 5. In that work a combination of RRT and B-Spline methods were used for path planning of a UAV amidst the presented environments. We have solved the scenarios and compared our results with those presented in [8].

The simulation results of the proposed algorithm on the four sample environments are demonstrated in Fig. 6. The solution time of the VO with the FMM and average filter method are presented in Table I and are compared with the results of the algorithm proposed in [8], in which average results had been obtained on a 3.00 GHz Intel Pentium 4 processor over 20 runs. All of our experiments were conducted on a 2.00 GHz Intel core i3 processor. The results indicate that the presented method performed better than the RRT-based algorithm especially in narrow and maze-like environments. Also, in our method the UAV's start and goal orientations are adjustable by using VOs while it is not so in the compared method.

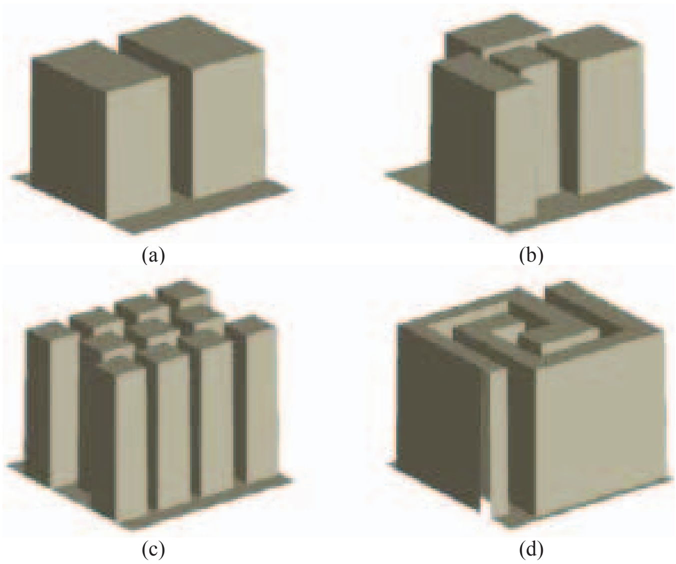


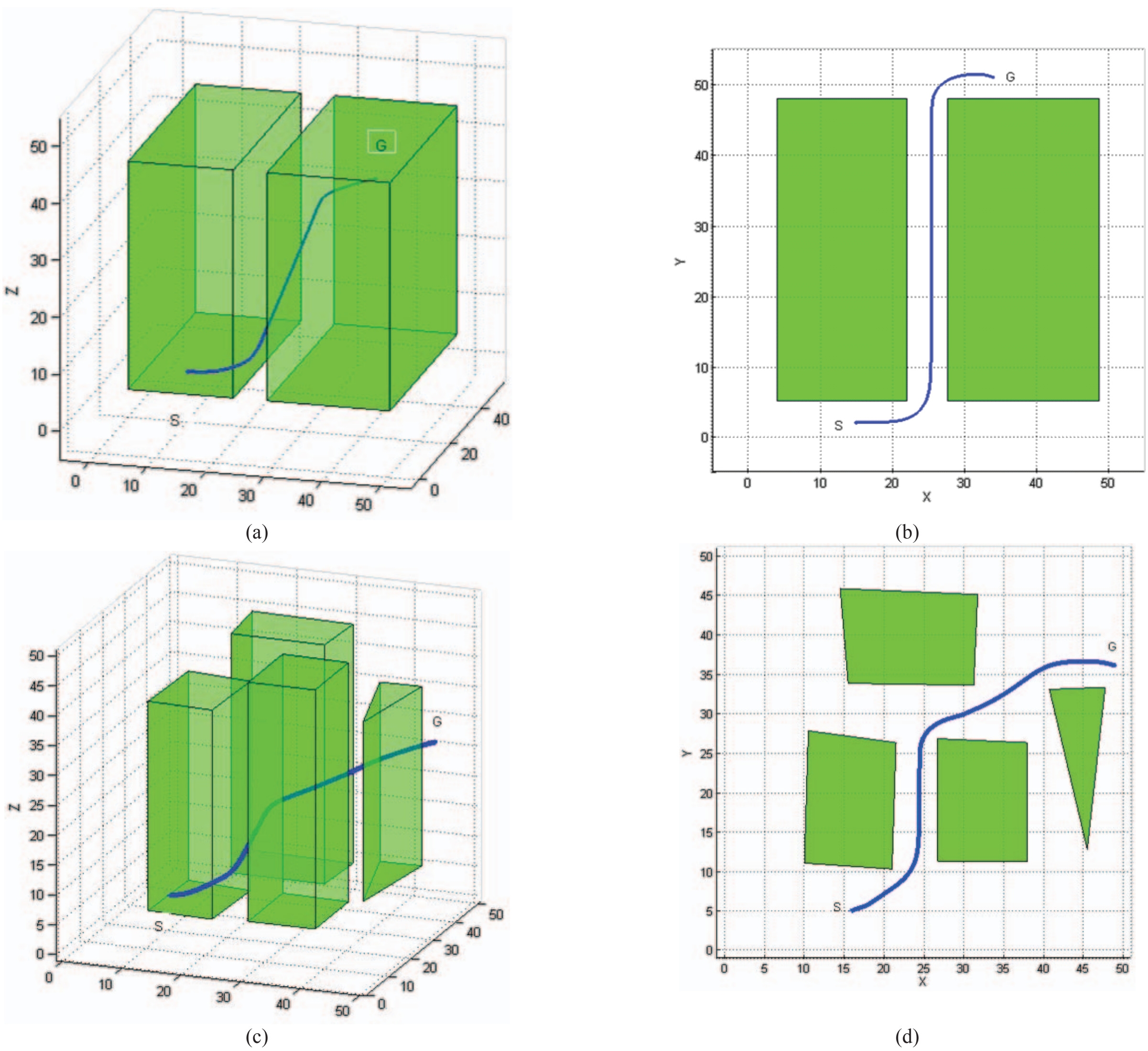
Fig. 5. Sample environments including: (a) narrow passage, (b) city-like, (c) mostly blocked, and (4) circuitous tunnel environments [8].

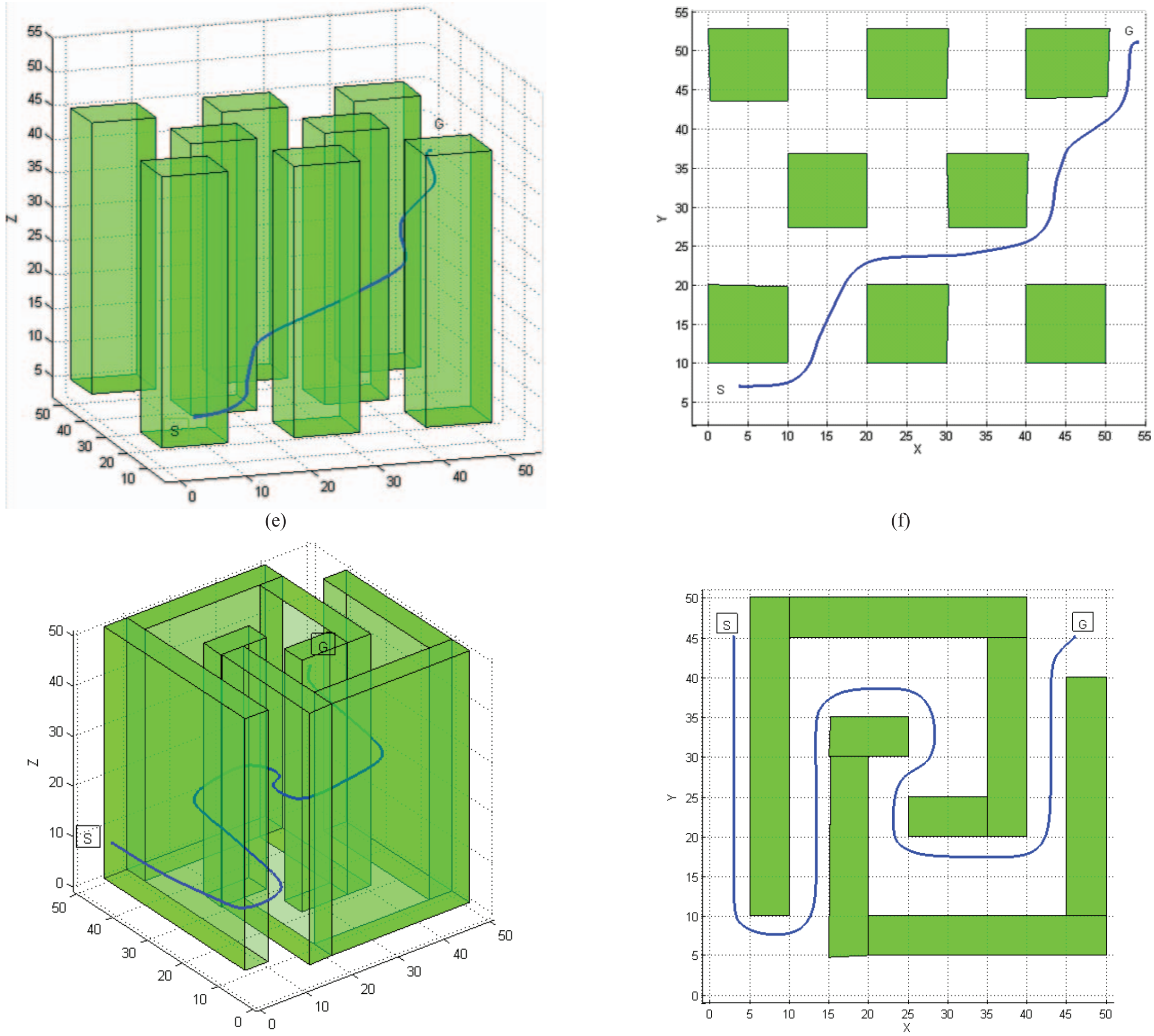
Table I. Comparing the solution times of the proposed algorithm with the RRT-based algorithm in [8].

Environment Type	Proposed method		RRT method [8]	
	Time (s)	Path length	Time (s)	Path length
(A) narrow passage	0.227	55.24	0.629	N/A
(B) city-like environment	0.286	87.68	2.121	N/A
(C) mostly blocked environment	0.959	96.87	5.561	N/A
(D) circuitous tunnel environment	0.503	172.37	24.67	N/A

V. Conclusions

This paper presents a new algorithm for 3D path planning of winged UAVs, based on a combination of Fast Marching Method, average filter, and Virtual Obstacles. In this algorithm the environment is uniformly discretized into voxels, and the speed of the fast marching method plays an important role in the algorithm's efficiency. Computational results yielded smooth and short paths for the UAV. As a future work, to improve the speed of the algorithm, the environment can be discretized non-uniformly with fewer cells in free space and more cells in obstacle-occupied space.





(e) (f)

(g) (h)

Fig. 6. Simulation results of the proposed method: (a)-(b) Path planning in narrow passage; (c)-(d) Path planning in city-like environment; (e)-(f) Path planning in mostly blocked environment; (g)-(h) Path planning in circuitous tunnel environment.

REFERENCES

[1] J. H. Reif and H. Wang, "Nonuniform discretization for kinodynamic motion planning and its applications," *SIAM J. Comput,* vol. 30, pp. 161–190, 2000.

[2] S. LaValle, *Planning Algorithms*. Cambridge: Cambridge University Press, 2006.

[3] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E.Kavraki, *et al.*, *Principles of Robot Motion :Theory, Algorithms, and Implementations.*: The MIT Press, 2005.

[4] L. E. Dubins, " On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents," *Amer. J. Math.,* vol. 79, 1957.

[5] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics,* vol. 145, 1990.

[6] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *In: Proceedings of IEEE International Conference on Robotics and Automation,* vol. 20, p. 378, 2002.

[7] M. Hwangbo, J. Kuffner, and T. Kanade, "Efficient Two-phase 3D Motion Planning for Small Fixed-wing UAVs," *IEEE International Conference on Robotics and Automation,* 2007.

[8] E. Koyuncu and G. Inalhan, "A probabilistic B-spline motion planning algorithm for unmanned helicopters flying in dense 3D environments," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2008, pp. 815-821.

[9] J. A. Sethian, *Level Set Methods and Fast Marching Methods.*: Cambridge, Massachusetts: Cambridge University Press, 1999.

[10] C. Petres, Y. Pailhas, P. Patron, Petillot, Y.,, J. Evans, and D. Lane, "Path planning for autonomous underwater vehicles," *IEEE Trans. Robotics,* vol. 23, pp. 331-341, 2007.

[11] C. Petres, Pailhas, Y., Patron, P., Petillot, Y., Evans, J., Lane, D., "Path Planning for Autonomous Underwater Vehicles," *IEEE Trans. Robotics,* vol. 23, pp. 331-341, 2007.

[12] L. D. Cohen and R. Kimmel, "Global minimum for active contour models: a minimal path approach," *Int. Journal of Computer Vision* vol. 24, pp. 57-78, 1997.

[13] D. Jannat and E. Masehian, "Path Planning of Tractor-Trailer Robot by Fast Marching Method (FMM)," *Journal of Modeling in Engineering,* vol. 11, pp. 31-47, 2013.