

Path Planning in Complex 3D Environments Using a Probabilistic Roadmap Method

Fei Yan¹ Yi-Sha Liu² Ji-Zhong Xiao³

¹School of Control Science and Engineering, Dalian University of Technology, Dalian 116023, China

²School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China

³Department of Electrical and Engineering, The City College, City University of New York, New York 10031, USA

Abstract: This paper presents a 3D path planning algorithm for an unmanned aerial vehicle (UAV) in complex environments. In this algorithm, the environments are divided into voxels by octree algorithm. In order to satisfy the safety requirement of the UAV, free space is represented by free voxels, which have enough space margin for the UAV to pass through. A bounding box array is created in the whole 3D space to evaluate the free voxel connectivity. The probabilistic roadmap method (PRM) is improved by random sampling in the bounding box array to ensure a more efficient distribution of roadmap nodes in 3D space. According to the connectivity evaluation, the roadmap is used to plan a feasible path by using A* algorithm. Experimental results indicate that the proposed algorithm is valid in complex 3D environments.

Keywords: 3D path planning, complex environment, unmanned aerial vehicle (UAV), probabilistic roadmap method (PRM), octree.

1 Introduction

Recently, unmanned aerial vehicles (UAV) are being increasingly used in many fields to perform tasks which are too difficult or dangerous for human beings, such as fire fighting^[1], disaster relief^[2], search and rescue^[3], and target tracking^[4], as well as space-oriented applications^[5]. For these applications, the UAV should have the ability to plan a non-collision path in its work space. The central problem of path planning is to generate a collision-free path from an initial point to a target point in an environment with obstacles and optimize the path according to some criterions such as energy, time, and safety^[6].

Planning algorithms have been developed in last few decades. In [7], the wave front algorithm is used as the path planner on three-dimensional normal distributions transform (3D-NDT) maps. This algorithm is intuitive, easy to implement, but it will fail to find the optimal path in complex environments. On the other hand, A* (see [8–10]) are best-first heuristic search algorithms that find the least-cost path from a given start node to goal node. Unfortunately, as the work space becomes larger and larger, the time spent to find the optimal path increases exponentially^[5]. Roadmap is usually applied to assist these heuristic search algorithms, which is often represented on configuration space. Various roadmap approaches have been used such as visibility lines^[11], virtual force^[9], rapidly-exploring random

trees^[12, 13] and probabilistic roadmap method (PRM)^[14].

For UAV, the path planning will be carried out in 3D environments, which is a big challenge especially in complex environments. The method often used for high-dimensional configuration space is based on sampling-based approaches, including the PRM^[15, 16] and its variants. PRM has two phases: a learning phase and a query phase^[15]. During the learning phase, a data structure called the roadmap is constructed for a given scene. The roadmap is an undirected graph $R = (N, E)$. The N is a set of nodes, which are randomly sampled points from collision-free configuration space. Pairs of promising nodes are chosen in the graph and a local planner is used to try to connect such placements with an edge. If they can be connected, then the edge is added to E . A good graph R should cover the free space well. In the query phase, the path is usually found by a path query such as Dijkstra or A*.

The bottleneck of basic PRM is that it has trouble in finding paths through narrow passages. This is due to the random sampling nodes in the free space so that too few nodes are placed in narrow passages. One solution is to add configurations near obstacle boundaries^[17, 18]. Though this approach increases the probability of sampling nodes in narrow passages, many nodes outside the narrow passages do not help in improving the connectivity of roadmaps. Hsu et al.^[19, 20] proposed a hybrid strategy: The bridge test is applied in narrow passages to increase the sampling density and it also uses uniform sampling strategy in open free space. In [21], the author improved lazy probabilistic path planning by using a lazy significant edge algorithm to place

Manuscript received March 31, 2013; revised August 8, 2013

This work was supported by National Natural Science Foundation of China (No. 61305128), Fundamental Research Funds for the Central Universities, and U.S. Army Research Office (No. W911NF-09-1-0565).

new samples in areas that have not yet been navigated. In these methods, the obstacles of the environments are presented by geometrical models. But in practice, the practical experimental environment cannot be described by geometrical models. For complex 3D scene, the most popular representation is the 3D point cloud^[22–27] or voxels^[28, 29]. When using basic PRM or variants mentioned above in these 3D representations, the collision checking is a heavy computational burden because of the huge number of obstacles (points or voxels). To deal with this problem, cell decomposition is a good choice which can divide the free space into subspaces. This method is used by Berg and Overmars^[30] to separate large open regions for steering the sampling toward the most interesting regions of the scene. Also, Zhang et al.^[31] proposed a hybrid approach for complete motion planning that combines approximate cell decomposition with PRM. But in these methods, the collision checking for obtaining a collision-free local path for the roadmap construction is also a big problem in complex 3D environments.

In this paper, we propose an algorithm based on PRM method for UAV path planning in 3D complex environments. The environment can be represented by 3D point cloud captured by laser scanner. However, the raw data from laser scanner is prohibitively huge for any path planning algorithms to be implemented in real time. So we use octree algorithm to divide the work space into voxels, which is a 3D grid representation of the environment. In order to extract free space for UAV navigation, we record the center point and size of free voxels during octree generation process. The size of the smallest free voxel for path planning is set to be larger than the physical size of UAV to ensure that it can pass through the path safely. The final path is a sequence of connected free voxels. In order to deal with the narrow-passage problem, the environment is equally divided into several sub-regions called bounding boxes. Different from our previous paper^[32], the nodes are sampled randomly from free voxels in each bounding box according to the volume ratio of the free and occupied voxels. The connectivity evaluation of free voxels is carried out in each bounding box. Then the local planner will search collision-free paths for the nodes in the bounding box. In order to make the roadmap cover the whole free space, an additional bounding box is added to overlap the neighboring bounding boxes for the connectivity evaluation of the local roadmaps in the adjacent region. Finally, we use A* method to search path from the start voxel to the goal voxel in the roadmap. We have implemented this algorithm in practical outdoor environments, which are represented by 3D point cloud.

The remainder of the paper is organized as follows: Section 2 introduces representation for complex 3D environment. Section 3 describes the proposed method for real-

time path planning. The results of experiments are presented in Section 4. The conclusions and future works are given in Section 5.

2 Environment representation

3D-grid representation provides the most information, which will be applied for 3D path planning in this paper. For UAV path planning, the free space of the environment is the essential information. So the focus of our method is to find free space in the environment.

For speeding up the process of building 3D-grid structure, octree algorithm is used. Octree is a hierarchical data structure for spatial subdivision in 3D^[33]. Each node in an octree represents the space contained in a cubic volume, usually called a voxel. This volume is recursively subdivided into eight sub-volumes until the sizes of voxels satisfy the expected resolution. In its most basic form, octree can be used to model a Boolean property^[29]: Occupied or free. Without loss of generality, the sub-volumes, which contain 3D data, are applied to represent the environment as occupied voxels. We focus on the free voxels, which are also built during the octree generation process. As shown in Fig. 1, the occupied voxels will be sub-divided in the next step. We record positions and sizes of the free voxels in each step. In Fig. 1, the light voxels are free ones and the dark voxels are occupied. The free ones represent the free space in the environment. In order to describe the environment accurately, the researchers usually divide the scenes into high resolution voxels. Ryde mapped an office environment in 3D at 0.02 m resolution^[28]. In [29], the voxel

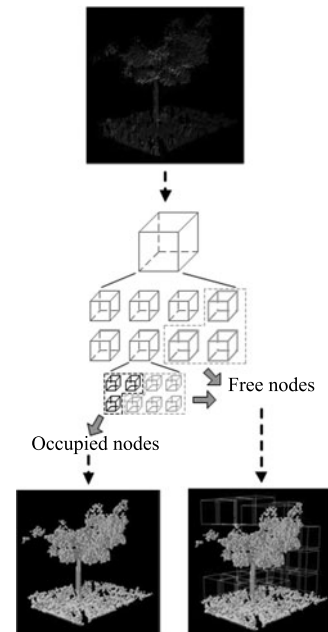


Fig. 1 The occupied voxels and free voxels are extracted from 3D data during octree building

size for maps is 0.04 m or 0.05 m. In this way, there are two problems: First, it suffers from significant computational burdens in detecting obstacles or planning path in complex environments. Second, the small free voxels cannot ensure the safety when used for UAV path planning (the size of our UAV is about 0.4 m). In our octree method, if the sub-volumes of one voxel are not large enough for UAV to pass through, this voxel will never be subdivided. So the resolution is dependent on the physical size of UAVs. Because the free voxels are extracted from different level of the octree algorithm, their sizes are not uniform.

3 3D PRM algorithm

Our path planning strategy is first to sample nodes effectively according to the volume ratio of free and occupied voxels in sub-regions called bounding boxes. After the first phase, local planner is carried out in each sub-region to build the roadmap. Finally, the search approach is used to obtain a path from the roadmap.

3.1 Node selection

The node selection is the basic but important process of PRM. In basic algorithm, the random sampling strategy is applied for choosing nodes in the entire free space, which causes low success rate of sampling nodes in narrow passage region.

In order to deal with the narrow-passage problem, we divide the environment space into equal-sized sub-regions called bounding boxes, whose sizes are m times the resolution of free voxels. So the maximum number of voxels in a bounding box is $n_{\text{voxel}} = m^3$ ($m \geq 2$). There are three kinds of bounding boxes: empty box, full box and mixed box. In full boxes, there are only occupied voxels so that no voxel can be selected for roadmap construction. On the contrary, the voxels belonged to empty boxes are all free voxels, which can ensure the safety of UAV path planning. We can randomly choose any voxels as nodes of the roadmap. Though more nodes can make the roadmap cover the free space better, the computational burden is also increased in this situation. How to select nodes in empty boxes will be discussed in next section. Without loss of generality, the mixed boxes are the boundaries of obstacles that mean the narrow passages are also included in this kind of bounding boxes. In these boxes, we will increase the density of random nodes according to the ratio of occupied and free voxels. Because the sizes of free voxels are not uniform in our algorithm, the number is proportional to the volume ratio of the voxels. Then the number of selected nodes in one box is

$$n_{\text{node}} = k \cdot \frac{\sum v_{\text{occupied}}}{\sum v_{\text{free}}^i} \quad (1)$$

where v_{occupied} is the volume of one occupied voxel and v_{free}^i is the volume of the i -th free voxel, and k is a preset

coefficient.

Fig. 2 shows the strategy to compute the number of selected nodes. There are 10 values for coefficient k , which are from $0.1n_{\text{voxel}}$ to n_{voxel} . If $k = n_{\text{voxel}}$, the curve of number of selected nodes is so steep that the computational burden increases fast. So in application, we set k to $0.1n_{\text{voxel}}$. Then as shown in Fig. 2, when the occupied voxels is less than $0.625n_{\text{voxel}}$, we will choose quarter of free voxels as nodes. If the number of occupied voxels is bigger than $0.73n_{\text{voxel}}$, all the free voxels are selected. In the other situations, node selection is carried out according to (1).

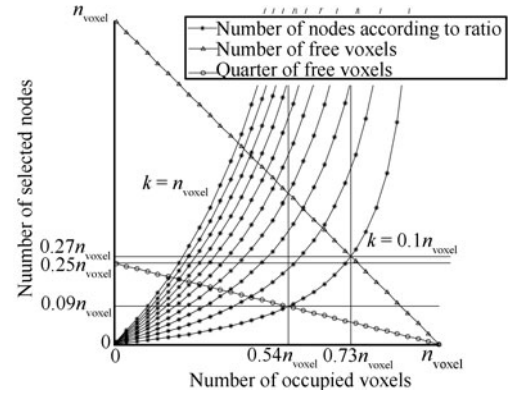


Fig. 2 The strategy of node selection

We compare our strategy with the random sampling method. Because the voxels usually block each other in 3D scene, we test the two methods in a 2D simulation environment in order to illustrate the difference clearly. The environment is composed of two big free regions and one narrow passage as shown in Fig. 3. We divided the free space into 2D grids by quadtree method. There are total 329 free grids, and 55 of them are in the passage. The size of bounding box is 4 times of the resolution of free grids. Then according to our strategy, 109 nodes are selected in this environment. For comparison purpose, we select 109, 155 and 220 nodes in the same environment in the random sampling method. Then 100 independent runs are carried out for each situation respectively. We take down the number of selected grids in the narrow passage and the results are shown in Fig. 4. Each box shows the number between the minimum and maximum. The square is the average value. The text above the box shows the success rate of constructing roadmap to connect the two big free spaces using our local planner (see Section 3.2). When 220 free grids in random sampling method are chosen, the result is similar to our strategy.

Two examples with 109 nodes are shown in Fig. 3. In Fig. 3 (b), there are no nodes in the middle of passage so that the roadmap cannot connect the two big free spaces. Furthermore, most of the random nodes are the smallest free grids, because the number of smallest free grids is much

bigger than the large ones. These smallest nodes near obstacle boundaries do not help in improving the connectivity of roadmap^[19]. Comparing with the random sampling method, our strategy is much better (Fig. 3 (a)).

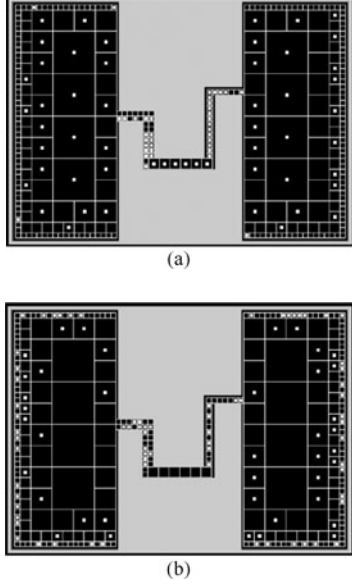


Fig. 3 Two examples for comparing our strategy for node selection with uniform sampling method. (a) 109 nodes are selected according to our strategy; (b) The same number of nodes is randomly sampled in the same environment

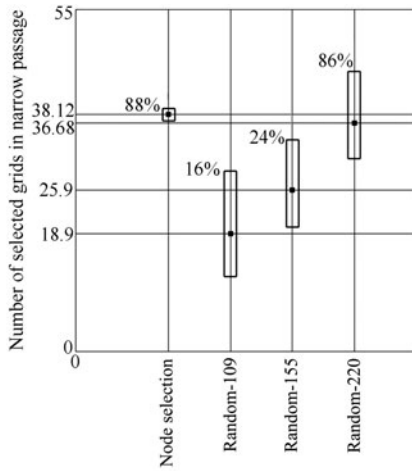


Fig. 4 The number of selected grids in narrow passage

3.2 Local planner

In some PRM methods, the local planners select a set of neighbors for each new node that is added to the roadmap. In this paper, bounding boxes have divided the work space into several sub-regions. So unlike other PRM methods, we do not need to search the k -nearest neighbors but consider all the selected nodes in the same bounding box as the neighbors. In the bounding box, if the number of nodes is

small, some nodes are not connected by using straight-line local planner. This will affect the roadmap construction. In Fig. 5, there are three nodes in the bounding box. When using a straight-line local planner, the edges between every two nodes collide with the obstacle. To deal with this problem, we propose a simple method as local planner, which can search a collision-free path for two nodes.

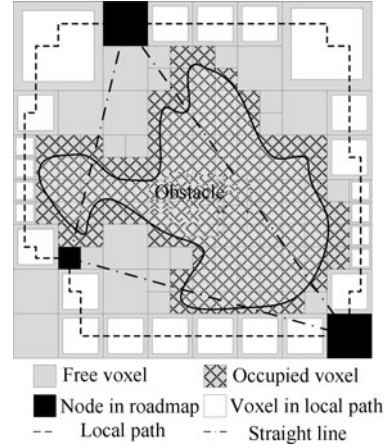


Fig. 5 Local planner in a bounding box

As mentioned in previous section, the size of free voxels extracted by the octree algorithm is larger than the physical size of the UAV to ensure that it can pass through safely. In order to create a non-collision path, the free voxels must be connected one by one. So the connectivity between voxels must be evaluated. In this paper, two voxels who share a same surface between them are considered as neighbor voxels. And if the positions of two voxels satisfy

$$\begin{cases} |x_a - x_b| \leq \frac{(S_a + S_b)}{2} \\ |y_a - y_b| \leq \frac{(S_a + S_b)}{2} \\ |z_a - z_b| \leq \frac{(S_a + S_b)}{2} \end{cases}$$

where S_a and S_b are sizes of the two voxels, the voxels are connected in x axis as shown in Fig. 6. In this way, the shared area of two voxels can ensure the UAVs to pass through safely.

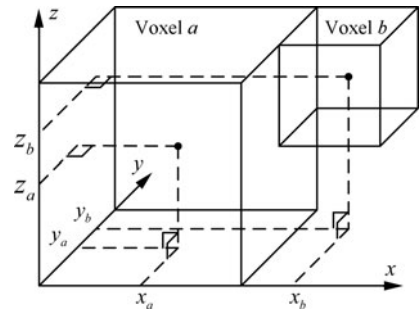


Fig. 6 The connectivity of two free voxels is evaluated by their position coordinates

Based on the connectivity evaluation, we search a local path between the selected nodes by using A* method as shown in Fig. 5. All the local paths are composed of connected free voxels, which can ensure the safety of UAV. But only having the local paths in the bounding box is not enough for roadmap construction. To plan a complete path from start position to goal position, the connection status of local path in different neighbor bounding boxes is necessary. This problem will be solved by inserting another bounding box between the neighbor ones. As shown in Fig. 7, A and B are two neighbor bounding boxes, a and b are two free voxels belonging to the different boxes, respectively. So after the process of evaluation mentioned above, there is no connectivity between voxel a and b , because they are evaluated in different bounding boxes. But when there is another bounding box C inserted between A and B , the connectivity of a and b can be evaluated in C .

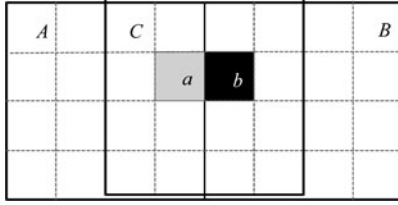


Fig. 7 Another bounding box C inserted between two neighbor ones (A and B) is used to evaluate connectivity of two free voxels (a and b) in different bounding boxes

The first condition for connecting the local paths in box A and B is that nodes a and b are belonged to box C . Because half part of A is in box C , when selecting a node randomly in box A , the probability of the node belonging to C is $\frac{1}{2}$. Suppose A is an empty box and $\omega \cdot n_{\text{voxel}}$ nodes are selected in box A . In order to ensure at least one node is placed in box C , the probability need to satisfy following condition

$$\frac{\omega \cdot n_{\text{voxel}}}{2} = \frac{\omega \cdot m^3}{2} \geq 1 \quad (2)$$

We can get $\omega \geq \frac{2}{m^3}$. Considering the computational burden, we select $\omega = \frac{2}{m^3}$ nodes in empty box. In this way, two nodes in one box can suffice for connecting the local path. But it is not reliable for roadmap construction. So in practice, we set ω to $\frac{2}{2^3}$, which means one in four free voxels are selected in empty bounding box. This strategy is also used for mixed boxes (Fig. 2).

3.3 Query phase

Suppose the roadmap is built and the start and goal free voxels are given. A* method is used to assist PRM for path planning during the query phase. A* is widely used for finding path. Here, we use A* to search the shortest path from the start node to the goal node in the roadmap.

In this process, we consider the straight-line distance and the number of free voxels connecting the two nodes in the roadmap as the searching cost. The specific steps are as follows:

1. $D[N_i] = \text{Arcs}[N_s, N_i]$, ($i = 1, \dots, n$)
// N_s is the start node.
 2. $P[N_i] = \begin{cases} N_i, & \text{Arcs}[N_s, N_i] = \infty \\ N_s, & \text{Arcs}[N_s, N_i] = \infty \end{cases}$
// P stores the previous node of N_i in path.
 3. $W[N_i] = D[N_i] + H(N_i, N_g)$
// N_g is the goal node.
// $H(N_i, N_g)$ is the heuristic function.
 4. Find N_k that satisfies $W[N_i] = \min(W)$
 5. $S.\text{push_back}(N_k)$
 6. While $N_k \neq N_g$
 7. For $j=1$ to n
 8. If $D[N_j] > D[N_k] + \text{Arcs}[N_k, N_j]$
 9. $D[N_j] = D[N_k] + \text{Arcs}[N_k, N_j]$;
 10. $W[N_j] = D[N_j] + H(N_j, N_g)$;
 11. $P[N_j] = N_k$;
 12. End
 13. End
 14. Find N_k that satisfies $W[N_i] = \min(W)$
 15. $S.\text{push_back}(N_k)$
// S stores the nodes in the path
 16. End
- $\text{Arcs}[N_i, N_j]$ describes the connection between node N_i and N_j . The value is

$$\text{Arcs}[N_i, N_j] = \begin{cases} \infty, & \overline{N_i N_j} \in E \\ D(N_i, N_j) + N(N_i, N_j), & \overline{N_i N_j} \notin E \end{cases} \quad (3)$$

where $D(N_i, N_j)$ describes the distance between N_i and N_j , and $N(N_i, N_j)$ is the number of free voxels connecting N_i and N_j based on voxel connectivity evaluation by local planner. By using A* searching algorithm, the path is stored in vector P by retrieval from goal node.

4 Experiment results

This section is composed of two parts. First, we implement our method and test its performance with different sizes of bounding box in outdoor environments. According to the results, we will determine the value of m . In the second part, a larger and more complex environment will be used to test the validity of the method. The raw data of all the environments is 3D point cloud. For convenience, the data is collected by a 3D laser range finder. The 3D laser range finder is composed of a 2D laser scanner (LMS 291) and a platform with tilting mechanism. Although, the data is not obtained by UAV, it is very similar to the data collected by UAV with laser range finder in [27]. The size of our UAV is 0.4 m. During the octree generation process,

the resolution of occupied voxel is set to be 0.2 m and the resolution of free voxel is 0.5 m. The computations are all performed on a PC with Intel(R) Core(TM) i5-2520M CPU and 4 GB memory, which is an off-board computer.

4.1 Experiments for m selection

The experiments are carried out in three outdoor environments as shown in Fig. 8. The first environment is simple, which has a path in the middle and two rows of bushes on each side of the path (Fig. 8(a)). In the second one, the sparse trees make the environment more challenging for UAV path planning (Fig. 8(b)). The last environment is more complex, which is composed of dense trees and bushes (Fig. 8(c)). Some parameters for the environments can be found in Table 1.

Table 1 Parameters for the experimental environments

Environment	Size ($x \times y \times z$)	Number of 3D points	Number of free voxels
1	15 m \times 16 m \times 7 m	127 794	584
2	15 m \times 27 m \times 9 m	85 557	1 223
3	15 m \times 30 m \times 5 m	129 238	584

The size of bounding box is decided by the value of m . If m is set to 4, the size of bounding box is $4 \times 0.5 = 2$ m. We run path planning with different values of m (from 2 to 10) in the three environments. For each value, the method is carried out 100 times. The success rates of finding a path from the start voxel to the goal voxel are depicted in Fig. 9. And the average time for each step is shown in Fig. 8(g)–(i). The numbers above the bars are the numbers of selected free voxels. The figures in the last row show the results of path planning (Fig. 8(j)–(l)).

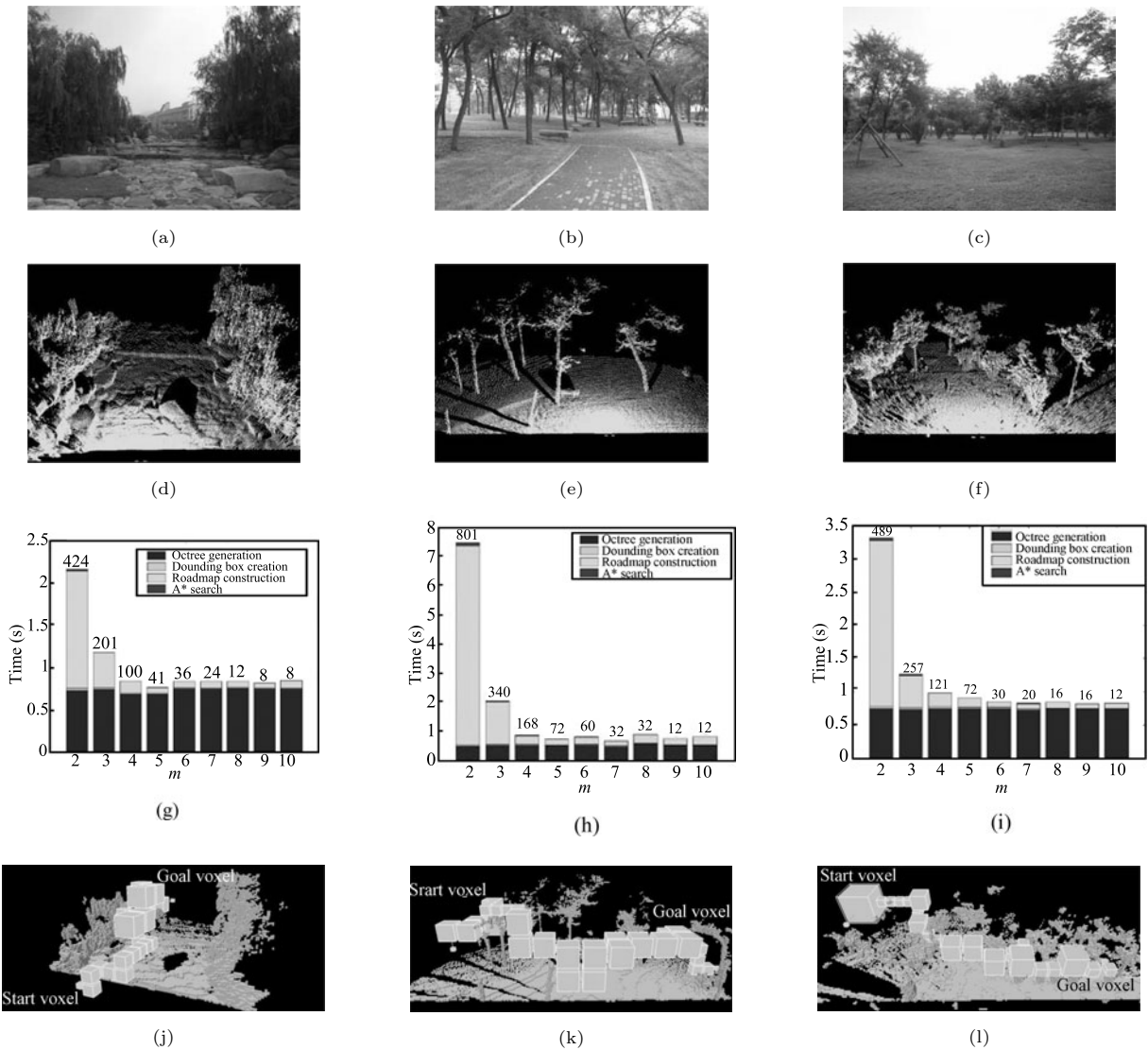


Fig. 8 Path planning in three outdoor environments. (a)–(c) Three outdoor environments; (d)–(f) The 3D point clouds for the three environments; (g)–(i) The time statistics of each step for different m ; (j)–(l) The results of path planning

When $m = 2$, the size of bounding box is so small that a few free voxels are selected in one bounding box. As mentioned in Section 3.2, if the number of nodes is small in bounding box, it is not reliable for roadmap construction. So the success rates are not 100% for all the three environments. On the other hand, when the value of m is bigger, the rates are much lower. This is because the nodes in the roadmap are too less (see the number above the bars in Fig. 8 (g)–(i)). In fact, the n_{node} obtained in Fig. 2 is a multiple of the volume of smallest free voxels. When using octree method to divide the free space, the volumes of the free voxels are not consistent. So the number of nodes is usually smaller than n_{node} . The bigger the value of m , the more possible that the bounding box includes big free voxels. So the number of nodes is less. In Fig. 9, when m is set to 3 and 4, the success rates are 100%. But the total time for path planning is different in Fig. 8 (g)–(i). In summary, the m should be set to 4.

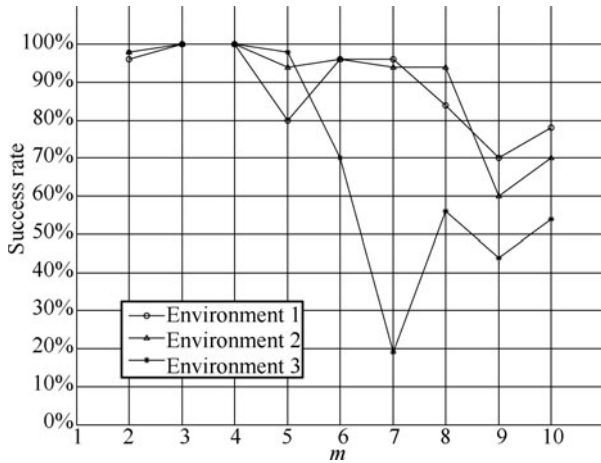


Fig. 9 The success rates for the different values of m in the three environments

4.2 Larger environment

Moreover, we also test the proposed method in a bigger outdoor environment. As shown in Fig. 10 (a), this is a wooded environment, which is composed of results of 10 laser scans. The size of the scene is $47\text{m} \times 32\text{m} \times 10\text{m}$, and the number of the raw laser points is 444 030. In this experiment, the free space is divided into 8169 free voxels. When m is set to 4, we choose 1194 free voxels as nodes to build the roadmap. The experiment is repeated 100 times. The final result of path planning is shown in Fig. 10 (b) and Fig. 10 (c). The experiment indicates that the proposed algorithm can find a safe path for UAV in complex environment. The average computation time in each step is shown in Table 2.

Table 2 The average computation time in each step

Step	Time (s)
Octree generation	1.71
Bounding box creation	0.322
Roadmap construction	14.135
A* search	0.224
Total	16.391

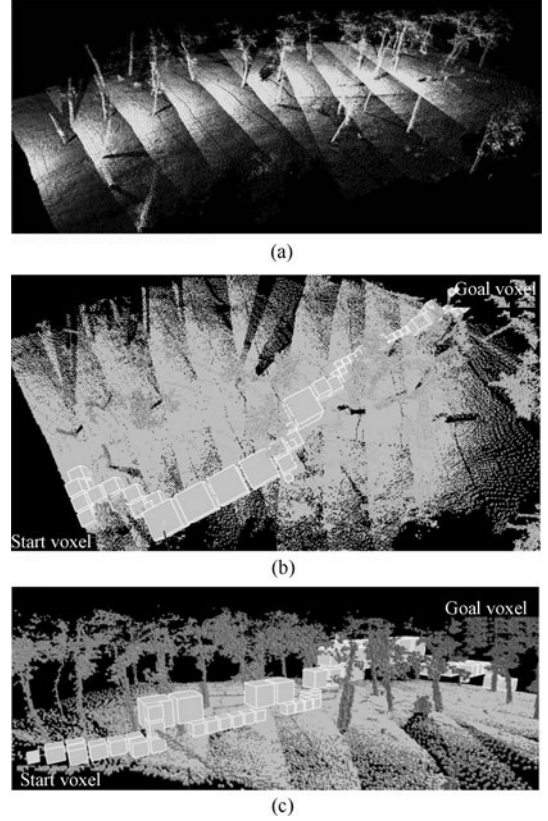


Fig. 10 Path planning in a large wooded environment. (a) A complex outdoor scene is composed of 3D points; (b) An overhead view of the path planning result; (c) A side view of the path

5 Conclusions

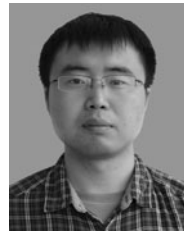
In this paper, we propose a 3D path planning method for UAV navigation in complex environments. In this study, we use octree algorithm to divide the work space into voxels that is a 3D grid representation of the environment. The free voxels, which have enough space tolerance to satisfy the safety needs of the UAV, are sorted to represent the free space of the environment. In order to obtain the connectivity of free voxels effectively, the environment is divided into several regions called bounding boxes, and the evaluation of free voxels connectivity is carried out in each bounding box instead of in the whole space. The basic PRM uses random sampling strategy to choose nodes in the entire free space, which causes the narrow-passage problem that may lead to failure of the path planning algorithm. We

improve the PRM by random sampling in bounding boxes to ensure a more reasonable distribution in the 3D space. Finally, based on the voxel connectivity, the selected nodes compose a roadmap, which is applied for path searching by A* algorithm for feasible path. The experiments of path planning are carried out in real outdoor environments. The result shows the algorithm can create a non-collision path for UAV.

References

- [1] J. Moore. UAV Fire-fighting System, U.S. Patent 20130134254 A1, May 2013.
- [2] M. B. Wang, A. Chu, L. A. Bush, B. C. Williams. Active detection of drivable surfaces in support of robotic disaster relief missions. In *Proceedings of Aerospace Conference*, IEEE, Big Sky, MT, USA, pp. 1–13, 2013.
- [3] D. Erdos, A. Erdos, S. E. Watkins. An experimental UAV system for search and rescue challenge. *IEEE Aerospace and Electronic Systems Magazine*, vol. 28, no. 5, pp. 32–37, 2013.
- [4] S. Q. Zhu, D. W. Wang, C. B. Low. Ground target tracking using UAV with input constraints. *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1–4, pp. 417–429, 2013.
- [5] Y. G. Fu, M. Y. Ding, C. P. Zhou. Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol. 42, no. 2, pp. 511–526, 2012.
- [6] O. Hachour. Path planning of autonomous mobile robot. *International Journal of Systems Applications, Engineering & Development*, vol. 2, no. 4, pp. 178–190, 2008.
- [7] T. Stoyanov, M. Magnusson, H. Andreasson, A. J. Lilienthal. Path planning in 3D environments using the normal distributions transform. In *Proceedings of 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Taipei, Taiwan, China, pp. 3263–3268, 2010.
- [8] Z. Qi, Z. H. Shao, Y. S. Ping, L. M. Hiot, Y. K. Leong. An improved heuristic algorithm for UAV path planning in 3D environment. In *Proceedings of the 2nd International Conference on Intelligent Human-machine System and Cybernetics*, IEEE, Nanjing, China, pp. 258–261, 2010.
- [9] Z. N. Dong, Z. J. Chen, R. Zhou, R. L. Zhang. A hybrid approach of virtual force and A* search algorithm for UAV path re-planning. In *Proceedings of the 6th IEEE International Conference on Industrial Electronics and Applications*, IEEE, Beijing, China, pp. 1140–1145, 2011.
- [10] S. Hrabar. 3D path planning and stereo-based obstacle avoidance for rotorcraft UAVs. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Nice, France, pp. 807–814, 2008.
- [11] R. Omar, D. W. Gu. Visibility line based methods for UAV path planning. In *Proceedings of ICROS-SICE International Joint Conference*, IEEE, Fukuoka, Japan, pp. 3176–3181, 2009.
- [12] H. L. Yu, R. Beard, J. Byrne. Vision-based navigation frame mapping and planning for collision avoidance for miniature air vehicles. *Control Engineering Practice*, vol. 18, no. 7, pp. 824–836, 2010.
- [13] K. Yang, S. Sukkarieh. 3D smooth path planning for a UAV in cluttered natural environments. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Nice, France, pp. 794–800, 2008.
- [14] K. Belghith, F. Kabanza, L. Hartman, R. Nkambou. Anytime dynamic path-planning with flexible probabilistic roadmaps. In *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, Orlando, USA, pp. 2372–2377, 2006.
- [15] L. E. Kavraki, P. Švestka, J. C. Latombe, M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [16] M. H. Overmars. A Random Approach to Motion Planning. Technical Report RUU-CS-92-32, Department Computer Science, Utrecht University, The Netherlands, 1992.
- [17] V. Boor, M. H. Overmars, A. Frank, A. F. van der Stappen. The Gaussian sampling strategy for probabilistic roadmap planners. In *Proceedings of the 1999 IEEE International Conference on Robotics & Automation*, IEEE, Detroit, USA, pp. 1018–1023, 1999.
- [18] N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In *Proceedings of the 3rd Workshop on the Algorithmic Foundations of Robotics on Robotics*, A. K. Peters, Ltd., Natick, MA, USA, pp. 155–168, 1998.
- [19] D. Hsu, T. T. Jiang, J. Reif, Z. Sun. The bridge test for sampling narrow passages with probabilistic roadmap planners. In *Proceedings of 2003 IEEE International Conference on Robotics & Automation*, IEEE, Taipei, Taiwan, China, pp. 4420–4426, 2003.
- [20] D. Hsu, G. Sanchez-Ante, Z. Sun. Hybrid PRM sampling with a cost-sensitive adaptive strategy. In *Proceedings of 2005 IEEE International Conference on Robotics and Automation*, IEEE, Barcelona, Spain, pp. 3874–3880, 2005.
- [21] J. Polden, Z. X. Pan, N. Larkin, S. Van Duin. Path planning with a Lazy Significant Edge Algorithm (LSEA). *International Journal of Advanced Robotic Systems*, vol. 10, pp. 1–8, 2013.
- [22] M. Bosse, R. Zlot. Place recognition using regional point descriptors for 3D mapping. In *Proceedings of the 7th International Conference on Field and Service Robotics*, Cambridge, Massachusetts, Springer, Berlin, Germany, pp. 195–204, 2009.
- [23] H. Men, B. Gebre, K. Pochiraju. Color point cloud registration with 4D ICP algorithm. In *Proceedings of 2011 IEEE International Conference on Robotics and Automation*, IEEE, Shanghai, China, pp. 1511–1516, 2011.
- [24] A. Gressin, C. Mallet, N. David. Improving 3D Lidar point cloud registration using optimal neighborhood knowledge. In *Proceedings of the ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, ISPRS, Melbourne, Australia, vol. I-3, pp. 111–116, 2012.

- [25] H. Andreasson, A. J. Lilienthal. 6D scan registration using depth-interpolated local image features. *Robotics and Autonomous Systems*, vol. 58, no. 2, pp. 157–165, 2010.
- [26] P. Henry, M. Krainin, E. Herbst, X. F. Ren, D. Fox. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.
- [27] I. Dryanovski, W. Morris, J. Z. Xiao. An open-source pose estimation system for micro-air vehicles. In *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, Shanghai, China, pp. 4449–4454, 2011.
- [28] J. Ryde, H. S. Hu. 3D mapping with multi-resolution occupied voxel lists. *Autonomous Robots*, vol. 28, no. 2, pp. 169–185, 2010.
- [29] J. Ryde, J. J. Corso. Fast voxel maps with counting bloom filters. In *Proceedings of 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Vilamoura, Algarve, Portugal, pp. 4413–4418, 2012.
- [30] J. P. van den Berg, M. H. Overmars. Using workspace information as a guide to non-uniform sampling in probabilistic roadmap planners. *International Journal of Robotics Research*, vol. 24, no. 12, pp. 1055–1071, 2005.
- [31] L. J. Zhang, Y. J. Kim, D. Manocha. A hybrid approach for complete motion planning. In *Proceedings of 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, San Diego, CA, USA, pp. 7–14, 2007.
- [32] F. Yan, Y. Zhuang, J. Z. Xiao. 3D PRM based real-time path planning for UAV in complex environment. In *Proceedings of 2012 IEEE International Conference on Robotics and Biomimetics*, IEEE, Guangzhou, China, pp. 1135–1140, 2012.
- [33] D. Meagher. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, vol. 19, no. 2, pp. 129–147, 1982.



Fei Yan received his B.Sc. and Ph.D. degrees in control theory and engineering from Dalian University of Technology, China in 2004 and 2011, respectively. Currently, he is a post-doctor in Mobile Robotics Laboratory, Dalian University of Technology.

His research interests include 3D environment modeling and path planning.

E-mail: fyan@dlut.edu.cn



Yi-Sha Liu received her B.Sc. and Ph.D. degrees in control theory and engineering from Dalian University of Technology, China in 2005 and 2011, respectively. Currently, she is a lecturer in School of Information Science and Technology, Dalian Maritime University.

Her research interests include mobile robot mapping and path planning.

E-mail: liuyisha@dlut.edu.cn



Ji-Zhong Xiao received his B.Sc. and M.Sc. degrees from the East China Institute of Technology, Nanjing, China in 1990 and 1993, respectively, M. Eng. degree from Nanyang Technological University, Singapore in 1999, and his Ph.D. degree from Michigan State University in 2002. He is currently a professor in electrical engineering at The City College, City University of

New York (CCNY/CUNY City College). He started the robotics research program at CCNY in 2002 as the founding director of CCNY Robotics Laboratory. He received the US National Science Foundation CAREER Award in 2007, the CCNY Outstanding Mentoring Award in 2011 and Humboldt Research Fellowship for Experienced Researchers by Alexander von Humboldt Foundation, Germany in 2013.

His current research interests include robotics and control, computer vision, sensor fusion, autonomous navigation, and multi-agent systems.

E-mail: jxiao@ccny.cuny.edu (Corresponding author)