Title	Exploration Priority Based Heuristic Approach to UAV Path Planning
Author(s)	Redwan Newaz, Abdullah Al; Pratama, Ferdian Adi; Chong, Nak Young
Citation	2013 IEEE RO-MAN: The 22nd IEEE International Symposium on Robot and Human Interactive Communication: 521-526
Issue Date	2013-08-26
Type	Conference Paper
Text version	author
URL	http://hdl.handle.net/10119/11615
Rights	This is the author's version of the work. Copyright (C) 2013 IEEE. 2013 IEEE RO-MAN: The 22nd IEEE International Symposium on Robot and Human Interactive Communication, 2013, 521-526. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Description	

Exploration Priority Based Heuristic Approach to UAV Path Planning

Abdullah Al Redwan Newaz, Ferdian Adi Pratama, and Nak Young Chong

***Abstract*— This paper presents a 3D online path planning algorithm for Unmanned Aerial Vehicles (UAVs) equipped with limited range sensors and computational resources in unknown cluttered environments. Even though quadrotor UAVs are considered to be a promising technology for surveillance purposes in indoor environments and for close observation in outdoor urban areas, it is very difficult to achieve autonomous aerial navigation toward a goal avoiding unpredicted collisions. Furthermore, greater attention and effort should be aimed at improving the computational efficiency and performance of path planning algorithms. The proposed heuristic algorithm offers on-the-fly path findings with a lesser computational complexity. We demonstrate the efficiency of our algorithm in a real world scenario implemented using the V-REP simulator.**

I. Introduction

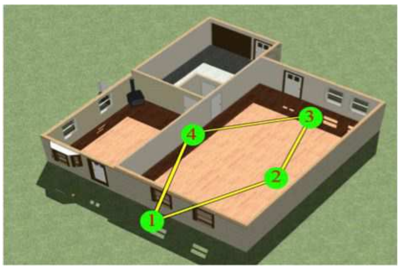
In this paper, we consider the surveillance and recovery mission after nuclear disasters or severe accidents in industrial areas which are inaccessible by humans. We divide the possible use of UAVs in the above-mentioned missions into three different categories: 1) indoor environment without having GPS or motion capture sensors [1], 2) indoor environment with motion capture sensors, 3) outdoor environment for close observation with the help of GPS.

In an indoor environment, one of the most important challenges for UAVs is to localize itself and search for the position to move, since the GPS does not work. Moreover, vision based localization is not always useful especially when after disaster situations are considered, since we cannot guarantee adequate lighting. Furthermore, scanning laser range finders or different types of vision sensors increase the computational burden, as a result it is quite difficult to implement in low-cost on-board processing units. Therefore, in the first scenario, without considering the computationally expensive sensors, the Exploration Priority Based Heuristic Approach (EPBHA) to online path planning is proposed.

Obviously, we must have an offline plan [2] before a surveillance mission that can be made by choosing several waypoints using the satellite maps as shown in Fig. 1. The waypoints are Cartesian coordinates representing spatial positions in the horizontal XY plane where a nominal height is assumed in the Z axis. Since every movement of an UAV creates its own coordinates, if we calibrate the UAV coordinate system with respect to outdoor GPS coordinates, the UAV can reach its goal along a set of waypoints. Moreover, we must have an approximate idea for maximum altitude for indoor environments. Specifically, if the previously unknown obstacles are detected, the UAV has to re-plan its nominal path in real time. If the path planner fails to generate a safe path within a bounded time, collisions with obstacles may result. Since the computational time of deterministic and complete algorithm grows exponentially with the dimension of the configuration space, those algorithms do not provide an adequate solution for online UAV path planning in indoor environments. However, as the UAV can not compare its coordinates with GPS or specific land coordinate systems, inertial navigation errors can be accumulated with its exploration. The accuracy of goal findings depends on a proper calibration system. Nevertheless, we assume that the requirements for the localization accuracy are not very strict for surveillance missions particularly within a small arena.

The authors are with the School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa 923-1292, Japan {redwan, ferdian, nakyoung}@jaist.ac.jp

(a) Outdoor GPS coordinates



(b) Indoor path planning

Fig. 1. Initial nominal path with respect to GPS coordinates

In the next scenario, stationary motion capture sensors are used to identify the coordinate of UAV. We can implement our algorithm in this scenario, since a number of infrared reflective markers are used to identify the object, therefore the environment is unknown unless the markers are attached to obstacles. Finally, our algorithm works for close observation in outdoor environments, where different size and shape of obstacles may appear in the path of a navigating UAV.

II. Related Works

Path planning has been one of the most important elements of mission definition and management of vehicles and it became crucial after birth and growth of UAVs. Quadrotors inaugurate the miniature form of UAV and furthermore their kinematics gives hovering capabilities that make it easy to create paths on the fly. Several algorithms were developed for robotic ground vehicles [3]–[5]. Likewise, physics for potential field algorithms [6], [7], mathematics for probabilistic approaches [8], or computer science for graph search algorithms [9] were applied to UAV path planning.

Binney *et al*. [10] presented a path planning method for autonomous underwater vehicles (AUVs) to maximize mutual information. Pereira and Sukhatme [11] presented minimum-risk path planning for AUVs operating in coastal regions with high ship traffic. Jung and Tsiotras [12] explained on-line path generation for UAVs using B-spline path templates where they investigated the problem of generating a smooth, planar reference path, given a family of discrete optimal paths. Jun and D'Andrea [13] used the probability map and Bellman-Ford shortest path algorithm in adversarial environments, maximizing the safety of the vehicles. Yang and Sukkarieh [14] discussed 3D path planning for an UAV operating in cluttered natural environments. Hrabar [15] proposed a synthesis of techniques for rotorcraft UAV navigation through obstacle-populated environments. Rohmer and Randall provided the target position programming solution [16], where a low level control of UAV was implemented with the target subdivided into *horizontal control and vertical control*.

Many techniques were developed to tackle the independent components for safe vehicle navigation in unknown environments. We handpick a selection of these that, when combined, offers what we believe is the best solution for the disaster surveillance with quadrotor UAVs. The need for off-line and real-time replanning substantially revises the path planning strategy. Moreover, the computational performances of the control station, where the mission management system is running, can influence the algorithm selection and design. The use of evolutionary algorithms for path optimization is an important solution permitting to apply kinematic constraints to the path. Using splines or random trees to model the trajectory, these algorithms can reallocate the waypoint sequence to generate optimum solutions in complex environments [17], [18]. Being interesting and flexible, the evolutionary algorithms are spreading on different planning problems, but their complexity is paid with a heavy computational effort [19]. The Dijkstra algorithm is one of the first greedy algorithms for graph search and permits to find the minimum path between two nodes of a graph with positive arc costs [20]. An evolution of the Dijkstra algorithm is the Bellman-Ford algorithm [21], [22] that finds the minimum path on oriented graphs with positive and negative costs. Another important method is the Floyd-Warshall algorithm [23], [24] that finds the shortest path on a weighted graph with positive and negative weights, but it reduces the number of evaluated nodes compared with the Dijkstra algorithm. The A* algorithm is one of the most important solvers explicitly oriented to robotics. A* improved the logic of graph search with heuristic evaluations inside the loop [10]. Dynamic re-planning with graph search algorithms was introduced. D* (Dynamic A*) represented the evolution of A* for re-planning [25]. Then, research on dynamic re-planning brought to the development of Lifelong Planning A* (LPA*) and D* Lite. They are based on the same principles of D* and D* focused, but they recall the heuristic aspect of A* to improve the speed of the search process [26], [27]. Different approaches were developed to cope with the suboptimal solutions problem, based on post-processing algorithms or on improvements of the graph-search algorithm itself. Very important examples are Field D* [28] and Theta* [29]. These algorithms refined the graph search obtaining generalized paths with any heading.

Comparing all of these algorithm, De Filippis *et al*. [30] conclude that Theta* is the most promising solution for the path planning of fixed-wing UAVs. However, the shortcoming of Theta* is the computational time. Our proposed algorithm is thus aimed at reducing the computational time.

III. Problem Statement

We categorized indoor surveillance missions further into two cases: 1) navigate in the obstacle free space and 2) avoid obstacles and escape from a deadend passageway. The former case means that the UAV finds the minimum distance path toward a goal position, if there is free space to move. The latter, however, must conform to several crucial conditions: how to avoid unexpected obstacles that appear in the path (located in front, or to the left or right, or any possible arrangement of obstacles, except for the upward direction). A complex and unpredictably changing environment makes it difficult to accomplish safe path planning. Moreover, using of vision sensors increases the computational complexity that makes it difficult to accommodate on-board implementation requirements. Therefore, without having any *a priori* knowledge of the environment, this paper proposes a new heuristic approach to allow UAVs to navigate through complex terrains, ensuring near-constant computations.

Now we address the path planning of UAV in unknown environments as follows: *Assuming a surveillance UAV equipped with limited range sensors exploring an arena, where different types of unknown obstacles exist, how to make it go to a goal position avoiding the obstacles with comparatively little computational cost?*

The path planning problem above can be decomposed into two sub-problems:

- **Sub-problem 1** (*free space*) How does it travel a minimum possible distance in an obstacle free area?
- **Sub-problem 2** (*obstacle avoidance*) How does it re-plan its position, while avoiding obstacles in its path?

IV. Algorithm Description

The idea underlying the proposed algorithm is similar to A* algorithm [31]. However, in A* algorithm for 2D plane, 8 Cartesian coordinates are computed and the coordinate of minimum cost among the cost of all coordinates is required to determine the movement position. Since the UAV does not know *a priori* the location of the obstacle, the cost of each coordinate is calculated based on, for instance, 'Manhattan Distance'. Although the proposed algorithm is a 3D path planning algorithm, to reduce the computational complexity, only one plane is chosen at a time for maneuvering. In practice, 6 movement options (forward and backward, left and right, and up and down) are available for the proposed path planning, while up and down movements are considered the special cases of obstacle avoidance maneuver. Therefore, normally for maneuvering UAV (U), costs are calculated based on 4 coordinates, which are front (C12), left (C21), back (C32), right (C23), respectively, as shown in Fig. 2.

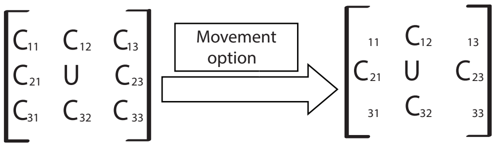


Fig. 2. Reduced cost assessment

Definition 1 (Input Description): The Cartesian coordinates of current position and goal position are given and the rest of coordinates are unknown. The UAV thus knows its own position and goal position but does not know *a priori* the obstacle position. The distance between one coordinate and the next coordinate is defined as step length d. The value of d is propositional to the velocity of UAV. For larger value of d, the UAV increases its velocity to cope with the distance that is required to travel within limited time boundary. The goal position is divided into two parts, *i.e.*, the goal in the XY plane and the YZ plane, respectively. After reaching the goal in one plane, the goal is automatically shifted to the other.

Definition 2 (Cost for coordinate): A coordinate cost is defined by the difference between the current position ($x1$, $y1$) and the next position ($x2$, $y2$) given by

$$Cost := A \times (x_1 - x_2) + B \times (y_1 - y_2),$$

where A and B are arbitrary even constants for emphasizing the straight forward (X-axis) or straight sideward (Y-axis) movements instead of the diagonal movements travel. If $A > B$, then the UAV moves forward or backward, while $A < B$ indicates left or right movements.

V. 3D Exploration Priority based Heuristic Approach for Obstacle Avoidance

In the proposed algorithm, the UAV searches two 2D planes separately to reduce the complexity of computations. After achieving the goal in the XY plane, it will shift its goal into the YZ plane that is the final goal. The searching algorithms is also divided into the obstacle free area and the obstacle cluttered area. The UAV tries to identify the shape of obstacle using its limited range of sensing which is analogous to the blind cane. It then chooses appropriate pre-defined maneuvering behaviors to avoid the particular type of obstacle. The proposed algorithm is basically divided into four parts: 1) grid making, 2) cost calculation, 3) obstacle avoidance, and 4) move to minimum cost point. Furthermore, four subfunctions are used which are the heading axis, sensor value, movement option, and next set position, respectively.

1. Grid making: The incremental distance between the parent coordinates and next coordinates is termed as d as defined in the previous section. For the next set position, one coordinate is chosen among four neighboring coordinates. The value of d could be determined by calibrating in the real world environment. If we compare our result to real world GPS values outdoors, then we have to calibrate it with respect to GPS values. Moreover, the smaller value of d ensures lesser probability of colliding with obstacles.

Algorithm 1 Pseudocode for grid making

for $i = 1; i < 5; i++$ **do**
    $grid[i][1] = i$ ▷ indexing
    $grid[i][2] = x \pm d$ ▷ next x-coordinate
    $grid[i][3] = y \pm d$ ▷ next y-coordinate
end for

2. Cost estimation: This part restricts the movement options of UAV: straight or perpendicular movements are more emphasized than diagonal movements. Therefore, costs of diagonal movements are higher than straight or perpendicular movements. This cost estimation (which is defined in Definition 2) is valid when there is no obstacle around the UAV.

3. Obstacle search: When the UAV finds an obstacle, it acquires two or more equal minimum cost coordinates at the same time. Therefore, according to A* or other existing algorithms, the UAV has to search every possible way to reach the goal, which we believe is quite impractical. In this work, the UAV has a preplanned idea about 'how to avoid the obstacles' and 'how to reduce the computational complexity'. Specifically, during the time of avoidance, it does not consider the cost for the goal. To acquire the knowledge of 'how to avoid the obstacles', we define several subfunctions detailed below.

Subfunction 1. (Direction of Heading) Comparing the current position ($x1$, $y1$) and the previous position ($x0$, $y0$), the UAV determines the X-axis or Y-axis along which it should move.

Algorithm 2 Pseudocode for heading direction in XY plane

if $(x0 - x1) > (y0 - y1)$ **then**
    `heading_is x`
else
    `heading_is y`
end if

Since the UAV's heading direction is likely to change, the variables of sensor value are also re-oriented accordingly. In Fig. 3, s1, s2, s3, and s4 represent the front, right, back, and left sensor value with respect to the UAV heading direction.

Subfunction 2. (Sensor Value) Obstacle detection is limited by the detection range and precision of sensors, where no detection range, the offset from the starting range, is

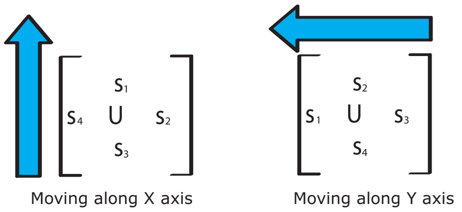


Fig. 3. Heading and sensor variables

introduced for sharp angle avoidance. Higher sensor range ensures safety, but decreases the accuracy to reach the goal.

Subfunction 3. (Movement Option) A sensor reports a certain range of numeric values, when it finds an obstacle. The available movement options are determined by counting the number of sensors that do not detect anything.

Algorithm 3 Pseudocode for movement option

for $i = 1; i < 7; i++$ **do**
  if value_of_sensor[i] $>$ sensor_range **then**
    $count += 1$ ▷ number of activated sensors
    `movement_option` $= 6 - count$
  end if
end for

Subfunction 4. (Next Set Position) The set position at the next moment ($x2$, $y2$) can be computed from the current heading and position ($x1$, $y1$) of the UAV.

Algorithm 4 Pseudocode for next set position in XY plane

if heading_axis $== X$ **then**
  if $y_1 > y_2$ **then**
    `next_set_position_is = left`
  else
    `next_set_position_is = right`
  end if
else if heading_axis $== Y$ **then**
  if $x_1 > x_2$ **then**
    `next_set_position_is = left`
  else
    `next_set_position_is = right`
  end if
end if

While the UAV moves along an axis and finds an obstacle in front of it, it calculates the set position at the next moment with respect to the current position, which gives priority to a certain direction (Subfunction 4). This change of heading is due to the UAV's myopia in orienteering. In most cases, the position of obstacle is close to the ground, hence the UAV may find an obstacle-free path at a certain height from the ground. As a result, in this algorithm, passing over is another priority after the heading changing movement for obstacle avoidance. The UAV will determine more than one obstacle from the sensor value and movement option subfunctions.

Moreover, the most interesting feature of the proposed algorithm is to avoid the cave type obstacle. In order to avoid such an obstacle, the UAV detects overhead obstacles and looks for its backward movements with respect to its heading direction (Subfunction 1). To reduce the penalty of backward movements, we have emphasized a special diagonal movement instead of straight backward movements.

4. Moving to minimum cost point: The UAV finds an optimal coordinate for its next set position and relocates its position to this coordinate. When the UAV changes its heading, the sensor indexes are also changed accordingly.

Below is a sketch of the proposed algorithm, incorporating the above-mentioned function modules:

Algorithm V.1: SEARCHING GOAL IN XY PLANE(x, y)

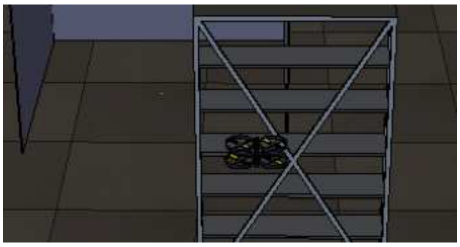
repeat
  GRIDMAKING()
  read sensor value
  if obstacle exist
    then EPBHA()
    else COSTESTIMATION()
  FINDMINIMUMINDEX()
  $compare(UAVPos(x, y), goalPos(x, y))$
  if $goalPos(x, y) - UAVPos(x, y) ==$ desired accuracy
    then xy search is finished
until xy search is not finished

Algorithm V.2: SEARCHING GOAL IN YZ PLANE(y, z)

repeat
  GRIDMAKING()
  read sensor value
  if obstacle exist
    then EPBHA()
    else COSTESTIMATION()
  FINDMINIMUMINDEX()
  $compare(UAVPos(y, z), goalPos(y, z))$
  if $goalPos(y, z) - UAVPos(y, z) ==$ desired accuracy
    then yz search is finished
until yz search is not finished

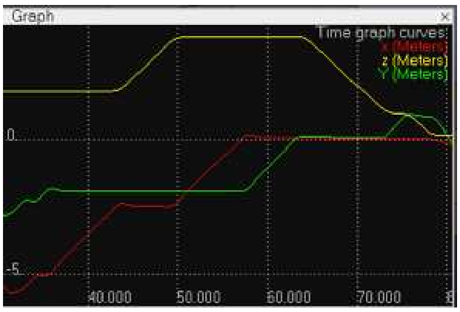
VI. Simulation Results and Discussion

Six infrared sensors are used as proximity sensors to detect obstacles which are mounted on top, front, right, left, back, and bottom, of the UAV, respectively. The proximity sensors have $0.5m$ range and $45°$ angle of detection. Moreover, it is imperative to place the sensors $15°$ to $30°$ inclined to the surface of body for proper detection and safe avoidance of obstacles. However, as we do not consider the measurement accuracy and signal processing of the sensors, sensor data is assumed to be accurate, noiseless, and achieved instantaneously. Although the proposed path planning is also valid for dynamically changing goals, the goal position is considered as static in this simulation. The initial status of the UAV is the standard hovering position, where we specify the goal position, seen in Fig. 4 (a) and (b), respectively, and the rest of the UAV kinematics are adjusted automatically using the dynamic simulation engine. Furthermore, as this paper does not deal with a low level control system, we therefore assume that we can accurately estimate the next movement of UAV without dead reckoning and/or other aerodynamics errors. Note that the flight path varies depending on the situation and environment as shown in Fig. 4. Moreover, it is assumed that the sensing range for UAV is limited (*i.e.*, $0.5m$) and there is no initial information such as map or pre-specified path. Therefore, the UAV can not plan a long distance path and does not require to retrieve previously given data, as a result the computation complexity is much lesser. We compare our algorithm with existing A* and D* search algorithms [32] which are commonly used for flight path planning to find the shortest path. The main difference starts while the UAV finds any obstacle along its path. When an obstacle appears in the path of UAV, it gets two or more minimum points for its next move. To find the shortest path, the UAV needs to explore every possible solution and decide which flight path it should choose.



(a) Starting position, in a stable, flying condition

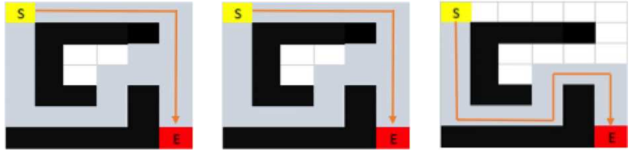
(b) Goal position, the box under the table



(c) Path traversing graph

Fig. 4. Dynamics simulation setup and result

In Fig. 5, the blue area indicates the searching area, where the yellow, black, red, and orange indicates the starting position, obstacle, goal position, and shortest path, respectively. From the figure, it is obvious that the proposed



(a) A* Search (b) D* Search (c) EPBHA

Fig. 5. Comparison between search algorithms and EPBHA

method offers less search, while other algorithms ensure the shortest path with higher search. We assumed that there is no initial information or map for the given place, therefore it is redundant for a single UAV to explore every possible way and choose the best one. Instead of searching for the minimum distance path, the 3D exploration capability of UAV allows it to easily avoid the obstacles. The most notable feature of the proposed method is that, obstacles reduce the searching time, while other existing searching algorithms always increase the computational parameters. Fig. 6 shows a significant decrease in coordinate cost estimation according to the obstacles position.

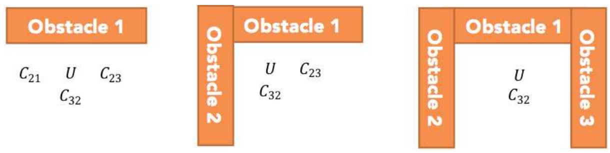


Fig. 6. Coordinate cost calculation during obstacle avoidance

Since this algorithm does not use the global information, it does not ensure the shortest path. It assures one of the feasible paths with lesser computations. For the surveillance mission considered, it is not essential to find the shortest path all the time. It should ensure a close-up view for that place. Likewise, obstacles do not always prevent UAVs from navigating a preplanned path, rather they could be also important items for surveillance purposes.

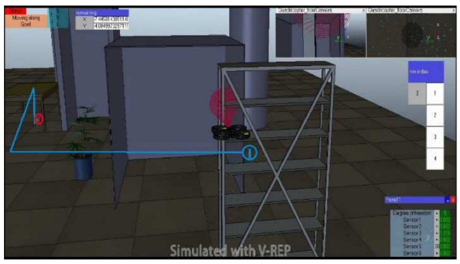


Fig. 7. Offline path planning for known environment

Fig. 7 represents the offline path planning, where all the environment information is initially available and the path obtained is the shortest path. Meanwhile, Fig. 8 shows a longer path compared to Fig. 7, but it ensures a close view for obstacles. This heuristic algorithm does not always guarantee to find the goal. For instance, it does not give any solution, while the UAV detects an obstacle in the backward direction. However, hovering is proposed for such a deadlock situation.

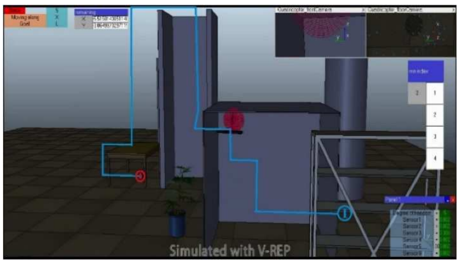


Fig. 8. EPBHA path planning

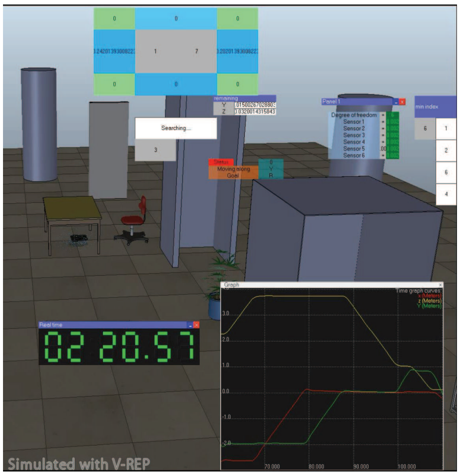


Fig. 9. Real time of exploration

Furthermore, the goal position is very close to the ground, therefore a small amount of error remains in the Y and the Z axis as shown in Fig. 4 (c). For better accuracy, the goal position should be located somewhere above the ground level and obstacle free environment. Fig. 9 shows that the real search time for the worst case setup is 2 minutes 20.57 seconds calculated by the real time function.

VII. Conclusion

Surveillance in unknown indoor environments is a challenging mission, since substantially more compact spaces and obstacles exist compared to spacious outdoor environments. The proposed algorithm offers one of the key technologies for low-cost surveillance UAVs in complex, cluttered areas ensuring low computational complexity. In addition, this algorithm envisions a new direction for online path planning, based on the fact that the obstacle does not always hinder us from reaching a goal position, rather sometimes it is helpful to reach a goal position easily. To recapitulate, we may conclude that this paper proposed a universal path planning algorithm of quadrotor UAVs equipped with limited range sensors and computational resources, particularly for small area surveillance purposes.

References

[1] R. L. Harmon, "Aerodynamic Modeling of a Flapping Membrane Wing Using Motion Tracking Experiments", *Master Thesis*, University of Maryland, College Park, 2008.

[2] `http://code.google.com/p/arducopter/`.

[3] T. Parsons, "Pursuit-Evasion in a Graph", *Theory and Applications of Graphs*, 426-441, 1978.

[4] G. Hollinger, S. Singh, J. Djugash, A. Kehagias, "Efficient Multi-robot Search for a Moving Target", *Int. J. Robotics Res.*, 28:201-219, 2009.

[5] N. Roy, G. J. Gordon, S. Thrun, "Finding Approximate POMDP Solutions through Belief Compression", *J. Artif. Intell. Res.*, 23:1-40, 2005.

[6] L. F. Bertuccelli, J. P. How, "Robust UAV Search for Environments with Imprecise Probability Maps", *IEEE Conf. Decision Control*, 2005.

[7] S. Waydo, R. M. Murray, "Vehicle Motion Planning Using Stream Functions", *IEEE Int. Conf. Robotics and Automation*, 2003.

[8] B. Pfeiffer, R. Batta, K. Klamroth, R. Nagi, "Path Planning for UAVs in the Presence of Threat Zones Using Probabilistic Modelling", *Handbook Military Industrial Engineering*, Taylor and Francis, 2008.

[9] P. Hart, N. Nilsson, B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", *IEEE Trans. Syst. Sci. Cybern*, 4(2):100-107, 1968.

[10] J. Binney, A. Krause, G. S. Sukhatme, "Informative Path Planning for an Autonomous Underwater Vehicle", *IEEE Int. Conf. Robotics and Automation*, 4791-4796, 2010.

[11] A. A. Pereira, G. S. Sukhatme, "Minimum-Risk Time-Expanded Planning for AUVs Using Ocean Current Predictions", 2012.

[12] D. Jung, P. Tsiotras, "On-line Path Generation for Small Unmanned Aerial Vehicles Using B-spline Path Templates", 2008.

[13] M. Jun, R. D'Andrea, "Path Planning for Unmanned Aerial Vehicles in Uncertain and Adversarial Environments", *Cooperative Control: Models, Applications and Algorithms*, 95-111, 2002.

[14] K. Yang, S. Sukkarieh, "3D Smooth Path Planning for a UAV in Cluttered Natural Environments", *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 794-800, 2008.

[15] S. Hrabar, "3D Path Planning and Stereo-based Obstacle Avoidance for Rotorcraft UAVs", *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 807-814, 2008.

[16] `http://coppeliarobotics.com/`.

[17] G. Guglieri, F. Quagliotti, G. Speciale, "Optimal Trajectory Tracking for an Autonomous UAV", *Automatic Control in Aerospace*, 1(1), 2008.

[18] I. K. Nikolos, N. C. Tsourveloudis, K. P. Valavanis, "Evolutionary Algorithm Based Offline/Online Path Planner for UAV Navigation", *IEEE Trans. Syst. Man Cybern., Part B, Cybern.*, 33(6):898-912, 2003.

[19] B. J. Capozzi, "Evolution-based Path Planning and Management for Autonomous UAVs", *Ph.D. Dissertation*, Univ. of Washington, 2001.

[20] E. W. Dijkstra, "A Note to Two Problems in Connexion with Graphs", *Numer. Math.*, 1:269-271, 1959.

[21] R. Bellman, "On a Routing Problem", *Q. Appl. Math.*, 16(1):87-90, 1958.

[22] L. R. Ford, D. R. Fulkerson, "Flows in Networks", Princeton University Press, 1962.

[23] R. W. Floyd, "Algorithm 97: Shortest Path", *Commun. ACM*, 5(6):345, 1962.

[24] S. Warshall, "A Theorem on Boolean Matrices", *J. ACM*, 9(1):11-12, 1962.

[25] A. Stentz, "Optimal and Efficient Path Planning for Unknown and Dynamic Environments", CMU Robotics Institute Tech. Report, 1993.

[26] S. Koenig, M. Likhachev, "D* Lite". *Proc. AAAI Conference on Artificial Intelligence*, 476-483, 2002.

[27] S. Koenig, M. Likhachev, "Incremental A*". *Proc. Natural Information Processing Systems*, 2001.

[28] D. Ferguson, A. Stentz, "Using Interpolation to Improve Path Planning: the field D* Algorithm", *J. Field Robot.*, 23(2):79-101, 2006.

[29] A. Nash, K. Daniel, S. Koenig, A. Felner, "Theta*: Any-angle Path Planning on Grids", *Proc. AAAI Conference on Artificial Intelligence*, 1177-1183, 2007.

[30] L. De Filippis, G. Guglieri, "Path Planning Strategies for UAVS in 3D Environments", *J. Intell. Robotics Syst.*, 247-264, 2012.

[31] K. Wang, G. L. Kovacs, M. Wozny, "Knowledge Enterprise: Intelligent Strategies in Product Design, Manufacturing, and Management", *Proc. Int. Conf. PROLAMAT, IFIP TC5*, 257, 2006.

[32] D. Ferguson, M. Likhachev, A. Stentz, "A Guide to Heuristic-based Path Planning", *Proc. WS Planning under Uncertainty for Autonomous Sys., Int. Conf. Automated Planning and Scheduling*, 2005.