DEVELOPMENT OF A PATH PLANNING ALGORITHM FOR UAVS IN DYNAMIC ENVIRONMENT USING DIFFERENTIAL GEOMETRY AND PROBABILITY FUNCTIONS

Sangwoo Moon*, David Hyunchul Shim**
***KAIST, ** KAIST**

Abstract

This paper deals with a real-time path planning algorithm for unmanned aerial vehicles in dynamic environment. To find an appropriate path, the differential geometry and probability functions are used to be robust to the uncertain information under the mission. In addition, the proposed algorithm is based on the line-searching algorithm which uses the axiom about Euclidean distance and the key ideas of visibility graph and A searching algorithm.*

Although the presented algorithm is simple, it can be used if the given environment is unknown or dynamically changed. Therefore, it is well fit to get the suitable path to avoid stationary or moving obstacles.

1 Introduction

From the past days, many of the techniques for unmanned aerial vehicles (UAVs) have been focused on the single UAV for many operations. However, those methods caused some disadvantages owing to the low efficiency, which means many hands are required for just one vehicle. Therefore, UAVs have been researched for risked missions such as the salvage works or the military operations owing to their high efficiency and prevent from damaging of the human life during those missions. In addition, multiple UAVs system is more resilient and adaptable to many kinds of missions than single UAV system, and it is also scalable and flexible to operate missions and to achieve goals.

Many reports have been announced the trend of UAV systems will be changed to the multi-agent based systems. However, there are some demands to accomplish. First of all, the advanced artificial intelligences should be developed for resilience. Biological approach is also needed to get some information from the nature. The motion planning and the task assignment are required to achieve missions efficiently under given conditions. Multi-based network and information integration are also essential requirements.

Among those stipulations, developing a path planning algorithm is a major issue which makes a system efficient and durable under the mission, and various techniques and ideas have been developed while the robotics field has been progressed. A* algorithm is well known algorithm to find the optimal path with respect to the minimum time using the heuristic approach [1]. There was an approach to find a path among polyhedral obstacles based on the geometry graph [2], and early works were focused on off-line planners for totally known surroundings [3-4]. Recently, probabilistic roadmap based path planning was introduced [5] and model predictive control was applied to path planning for vehicles [6-8].

However, real-time path planning is still a challenging work although this approach can be widely applied to the real world which has many uncertainties and unknown obstacles.

The proposed path planning algorithm on this paper has a key idea from an axiom about the Euclidean distance. Euclidean distance can be expressed as equation (1).

$$d(\mathbf{p},\mathbf{q}) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \qquad (1)$$

In the 2-dimensional Euclidean plane or the 3-dimensional Euclidean space, the Euclidean distance is the line which connects **p** and **q**, and this line is the shortest path from those two points. However, if any obstacles are located on the line, the path planning algorithm for collision avoidance is operated by using visibility graph and A* algorithm. However, visibility graph approach considers all obstacles so that the operation time and calculate sequence grows rapidly as the number of obstacles increases. In this algorithm, only one obstacle located on the line and close to the present position is concerned to calculate. A* algorithm is operated to supplement proposed algorithm.

To find the appropriate path on the unknown and dynamic environment, the fundamental equations of differential geometry are applied to this algorithm. Although obstacles and threats can be treated as polygons or circles composed by lines or curves, it may not be applied to the real situation because this assumption makes those kinds of things simple. In that situation, the differential geometry technique and using the probability functions for the scanned data can makes the information simple so that the proposed algorithm can be applied.

This paper introduces about how to use this algorithm and how this method is efficient and powerful compared to the other methods.

The remainder of the paper is as follows. In section 2, the formulated problem which contains conditions and requirements is introduced. Section 3 presents a path planning algorithm and the overall structure for the multi-agent system. Simulation results and analysis are discussed in section 4, while conclusions and future works are in section 5.

2 Problem Formulation

Path planning in the dynamic environment is NP-hard, which means it is hard to know the reasonable solution without considering all of the cases owing to some unknown factors. Many computer engineers have solved that problem as using the “approximation algorithms” or “heuristic algorithms”, which can give the admissible solutions, not the optimal solutions. One approach to get the path planning is to make the randomized candidates. Rapidly-exploring Random Trees is a method using a random procedure. However, the result using this approach may not give an admissible path because it can not consider any kinematics or dynamics for a vehicle [9].

Therefore, some principles to achieve a mission are needed before propose a fresh path planning algorithm. Since the researches about that kind of mission were progressed, the fundamental rules to carry out a mission have been also constructed. In this paper, the assumptions and criteria for the path generation are defined.

2.1 Criteria for the path generation

- UAV must move through an admissible area without forbidden regions.
- The procedure should be continuous and sequential operation is required.
- Obstacles or threats are defined as the things located on the field and disturb the operation.
- Environment is dynamically changed owing to the moving obstacles or pop-up threats.
- Each UAV has a laser scanner to detect obstacles which has certain range.

2.2 Assumptions for the path generation

- The shape of Obstacles or threats can be considered as the set of polygons and have concave or convex corners.
- Environment is on the 2 dimensional spaces.
- A vehicle has a maximum linear and angular velocity, so the rate of change of position has certain constraints.
- Position and velocity for each UAV are known.
- All UAVs move to dynamically assigned waypoint perfectly.
- The distribution for Multiple UAVs system is decentralized.

3 Algorithm Introduction

3.1 Fundamental Path Planning

In Euclidean space, it is an axiom that the shortest path between two points is the line which connects those points. If those points are $\mathbf{p}_s$ and $\mathbf{p}_g$, the cost function which minimizes the final time becomes to the equation (2) [10].

$$\min J = \int_0^{t_f} \|\mathbf{v}\| dt = dist(\mathbf{p}_s, \mathbf{p}_g) = \| \mathbf{p}_s - \mathbf{p}_g \| \qquad (2)$$

where 'dist' function is the distance defined in Euclidean space. However, if there are any obstacles or forbidden region on the line, equation (2) should be changed because that path is not admissible.

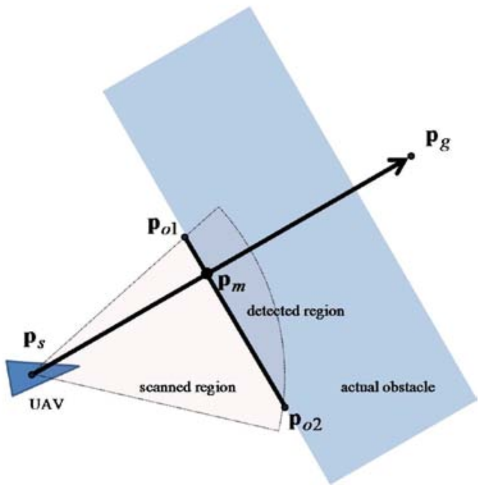

Fig. 1. Meeting Point between Shortest Path and Obstacle

To get an admissible path on this situation, the meeting point by the path from the equation (2) and the boundary of obstacles can be used. The first procedure of the proposed algorithm is to find this meeting point. As mentioned, an obstacle on 2-dementional space can be considered as a polygon, and the meeting point $\mathbf{p}_m$ in figure 1 can be solved by the geometrical approach. Equation (3) shows that how the meeting point $\mathbf{p}_m$ can be specified by using the determinant of a matrix.

$$\mathbf{p}_m = \left[\frac{\begin{vmatrix} \begin{vmatrix} \mathbf{p}_{sx} & \mathbf{p}_{sy} \\ \mathbf{p}_{gx} & \mathbf{p}_{gy} \end{vmatrix} & \begin{vmatrix} \mathbf{p}_{sx} & 1 \\ \mathbf{p}_{gx} & 1 \end{vmatrix} \\ \begin{vmatrix} \mathbf{p}_{o1x} & \mathbf{p}_{o1y} \\ \mathbf{p}_{o2x} & \mathbf{p}_{o2y} \end{vmatrix} & \begin{vmatrix} \mathbf{p}_{o1x} & 1 \\ \mathbf{p}_{o2x} & 1 \end{vmatrix} \end{vmatrix}}{\begin{vmatrix} \begin{vmatrix} \mathbf{p}_{sx} & 1 \\ \mathbf{p}_{gx} & 1 \end{vmatrix} & \begin{vmatrix} \mathbf{p}_{sy} & 1 \\ \mathbf{p}_{gy} & 1 \end{vmatrix} \\ \begin{vmatrix} \mathbf{p}_{o1x} & 1 \\ \mathbf{p}_{o2x} & 1 \end{vmatrix} & \begin{vmatrix} \mathbf{p}_{o1y} & 1 \\ \mathbf{p}_{o2y} & 1 \end{vmatrix} \end{vmatrix}} \quad \frac{\begin{vmatrix} \begin{vmatrix} \mathbf{p}_{sx} & \mathbf{p}_{sy} \\ \mathbf{p}_{gx} & \mathbf{p}_{gy} \end{vmatrix} & \begin{vmatrix} \mathbf{p}_{sy} & 1 \\ \mathbf{p}_{gy} & 1 \end{vmatrix} \\ \begin{vmatrix} \mathbf{p}_{o1x} & \mathbf{p}_{o1y} \\ \mathbf{p}_{o2x} & \mathbf{p}_{o2y} \end{vmatrix} & \begin{vmatrix} \mathbf{p}_{o1y} & 1 \\ \mathbf{p}_{o2y} & 1 \end{vmatrix} \end{vmatrix}}{\begin{vmatrix} \begin{vmatrix} \mathbf{p}_{sx} & 1 \\ \mathbf{p}_{gx} & 1 \end{vmatrix} & \begin{vmatrix} \mathbf{p}_{sy} & 1 \\ \mathbf{p}_{gy} & 1 \end{vmatrix} \\ \begin{vmatrix} \mathbf{p}_{o1x} & 1 \\ \mathbf{p}_{o2x} & 1 \end{vmatrix} & \begin{vmatrix} \mathbf{p}_{o1y} & 1 \\ \mathbf{p}_{o2y} & 1 \end{vmatrix} \end{vmatrix}} \right]^T \qquad (3)$$

In this situation, a new path should be defined which passes the boundary points of a detected obstacle, $\mathbf{p}_{o1}$ or $\mathbf{p}_{o2}$. Therefore, those two points are the candidates of waypoints for the path planning. The proposed algorithm uses A* approach expressed as equation (4).

$$\begin{aligned} f(\mathbf{p}) &= g(\mathbf{p}) + h(\mathbf{p}) \\ &= \| \mathbf{p}_s - \mathbf{p}_w \| + \| \mathbf{p}_g - \mathbf{p}_w \| \end{aligned} \qquad (4)$$

In equation (4), $\mathbf{p}_w$ is the boundary point of detected edge so that the first term of that equation means the distance from the starting point to the waypoint candidate and the other part is that from the waypoint candidate to the goal point. If the detected line is not the first since the operation started, this equation should be changed as equation (5) because the vehicle is already assigned a waypoint. If the detection is popped up at the N^{th} sequence and the time step for the algorithm is Δt, the first term of equation (4) is changed as the distance of the movement of a vehicle.

$$\begin{aligned} f(\mathbf{p}) &= g(\mathbf{p}) + h(\mathbf{p}) \\ &= \sum_{i=1}^{N} \| \mathbf{p}_{i\Delta t} - \mathbf{p}_{(i-1)\Delta t} \| + \| \mathbf{p}_g - \mathbf{p}_w \| \end{aligned} \qquad (5)$$

3.2 Probability model

However, the detected edge is not the whole edge of the actual obstacle so that only part of the obstacles is known. The proposed algorithm applies the probability model to the detected data described on the figure (2). The probability model is expressed as equation (6) by the scanned region.

$$p(\mathbf{p},\mathbf{p}_{o1},\mathbf{p}_{o2},\cdots,\mathbf{p}_{oN}) = \begin{cases} 0 & \text{if } \mathbf{p} \in \mathbf{R} \\ \max\left\{e^{-\|\mathbf{p}-\mathbf{p}_{o1}\|}, \ldots, e^{-\|\mathbf{p}-\mathbf{p}_{oN}\|}\right\} & \text{otherwise} \end{cases} \quad (6)$$

where $\boldsymbol{R}$ is the scanned region in Euclidean space.

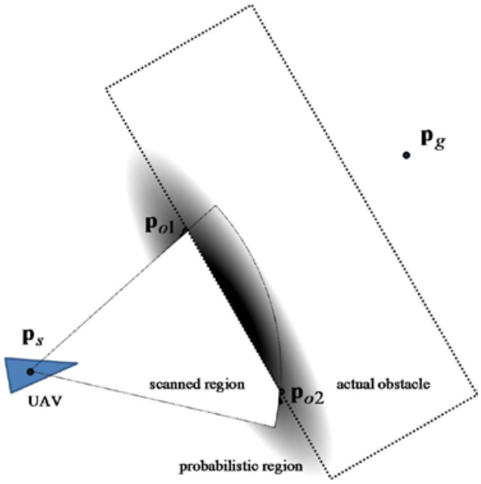

Fig. 2. Probability Region for Detected Data

In addition, the proposed algorithm simplifies to apply the probability which model is only applied to the extended line of the searched edge. "Probability line" is the result after this filtering. Figure 3 shows how the probability distribution is changed from the figure 2. As the equation (3) shows, the probability of the neighbor point which is on the extended edge is exponentially decreased as the distance from that point to the boundary point is increased. Furthermore, this proposed procedure of this algorithm has strength in the case of the data merging to make whole information of obstacles or threats.

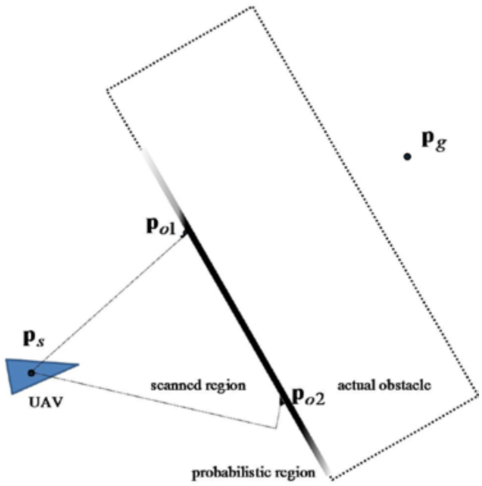

Fig. 3. Probability Line for Detected Data

3.3 Application of Differential Geometry

The proposed algorithm also uses the equations which are from the differential geometry field. As searching is going, the detected line is increased. Because this algorithm is based on the geometry, differential geometry can be an alternative approach to estimate the obstacles.

First of all, the boundary points of the detected edge are expressed as the Cartesian coordinate, and the each component can be also described as the function of the time. Furthermore, those functions are the 3rd order of the polynomial functions using the change of the location for the boundary points.

$$\mathbf{p}_o = \begin{bmatrix} x(t) & y(t) \end{bmatrix}^T \quad (7)$$

$$\mathbf{p}_o = \begin{bmatrix} C_{x1} & C_{x2} & C_{x3} & C_{x4} \\ C_{y1} & C_{y2} & C_{y3} & C_{y4} \end{bmatrix} \begin{bmatrix} 1 \\ t \\ t^2 \\ t^3 \end{bmatrix} \quad (8)$$

To solve those coefficients, at least 4 changed locations of the boundary points are required. However, the polynomial functions can be solved even the number of those locations are less than 4 because using the differential function in terms of the geometry. At this point, the probability form for the detected edges and the neighborhood points for the boundary of that line are applied to get the coefficient matrix. If there are more than 4 locations, the weighted matrix with respect to the time is applied which form is expressed as equation (11).

$$\dot{\mathbf{p}}_o = \begin{bmatrix} C_{x2} & C_{x3} & C_{x4} \\ C_{y2} & C_{y3} & C_{y4} \end{bmatrix} \begin{bmatrix} 1 \\ 2t \\ 3t^2 \end{bmatrix} \quad (9)$$

$$\ddot{\mathbf{p}}_o = \begin{bmatrix} C_{x3} & C_{x4} \\ C_{y3} & C_{y4} \end{bmatrix} \begin{bmatrix} 2 \\ 6t \end{bmatrix} \quad (10)$$

$$\mathbf{W} = \begin{bmatrix} e^{t_1-t} & 0 & \cdots & 0 \\ 0 & e^{t_2-t} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{t_i-t} \end{bmatrix} \quad (11)$$

Table 1. Pseudo Code for Proposed Algorithm

Step 1. Set the initial point and the goal point. If there are no paths, the starting point of this operation becomes the initial point.
Step 2. If there are any edges of obstacles or threats are detected, go to Step 3. Otherwise, this algorithm is stopped until any information of obstacles will be given.
Step 3. Draw the line which passes the initial point and the goal point set by Step 1. If there is a meeting point by an obstacle, go to Step 5, if there are some meeting point by an obstacle, go to Step 4. Otherwise, this algorithm is stopped until any information of obstacles will be given.
Step 4. The point which minimizes the distance between the initial point and each meeting point is considered the real meeting point.
Step 5. Select boundary points for a meeting line which is the line of an obstacle and the meeting point. Those points are candidates for a waypoint.
Step 6. Apply the probability function and the function for differential geometry to the boundary points and calculate the cost function from A* algorithm.
Step 7. Select a boundary points which minimize the cost from equation (4) or (5). This point is the waypoint and it will be used on the guidance for a vehicle and a mission planning.
Step 8. Move a vehicle to the waypoint by given position and velocity. If a vehicle reaches to the goal point, this algorithm is terminated. Otherwise, go to Step 1.

Table 1 shows the main procedure of this proposed algorithm. The selected waypoint is assigned in real-time so that this point will be changed as the information data for the obstacles are increased and changed. Each candidate point is considered when the probability of that point is bigger than the threshold.

$$\mathbf{p}_{o_{actual}} = \{\forall \mathbf{p} \mid p(\mathbf{p}, \mathbf{p}_{o1}, \ldots, \mathbf{p}_{oN}) \geq p_{threshold}\} \qquad (12)$$

4 Algorithm Application

If we do not know any information for obstacles and threats, real-time path planning approach is required to get a suitable path. In this algorithm, it is possible to solve a suitable path while a vehicle constructs a map which contains the location of obstacles.

As mentioned at section 2, each UAV has a laser scanner for mapping. The searching range of this equipment is 20 meter, and the searching angle of that is 60 degree. Figure 4 shows the information about the specification of a laser scanner.

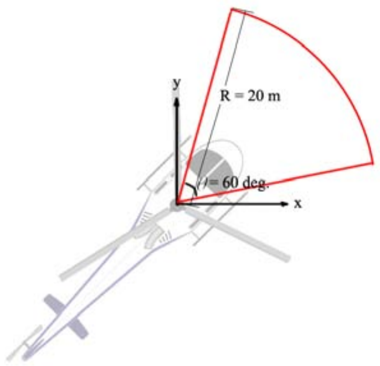

Fig. 4. Scanned Region of Laser Sensor

To get an admissible path, the simple kinematics for an UAV is used, and those equations are equation (13) and equation (14). In addition, the cruise velocity is 5 m/sec.

$$V_x = V\cos\theta \qquad V_y = V\sin\theta \qquad (13)$$

$$\dot{\theta} \in [-\delta, \delta] \qquad (14)$$

where δ is the maximum angular velocity. δ is assumed 0.8 rad/sec in this paper, which value is in the cruising phase of common UAVs. The time step for real-time planning is 0.2 sec.

Figure 5 shows the result from a given environment, which is totally unknown. The number of obstacles is 10, and their size and location is in random. Three UAVs are used to carry out the given scenario. This result shows that this algorithm cannot show the optimal solution because of the limited information for obstacles. However, because this algorithm is kind of the greedy algorithm, this proposed method generates the suboptimal solution. Figure 6 shows the cost time profile while this proposed algorithm is operated.

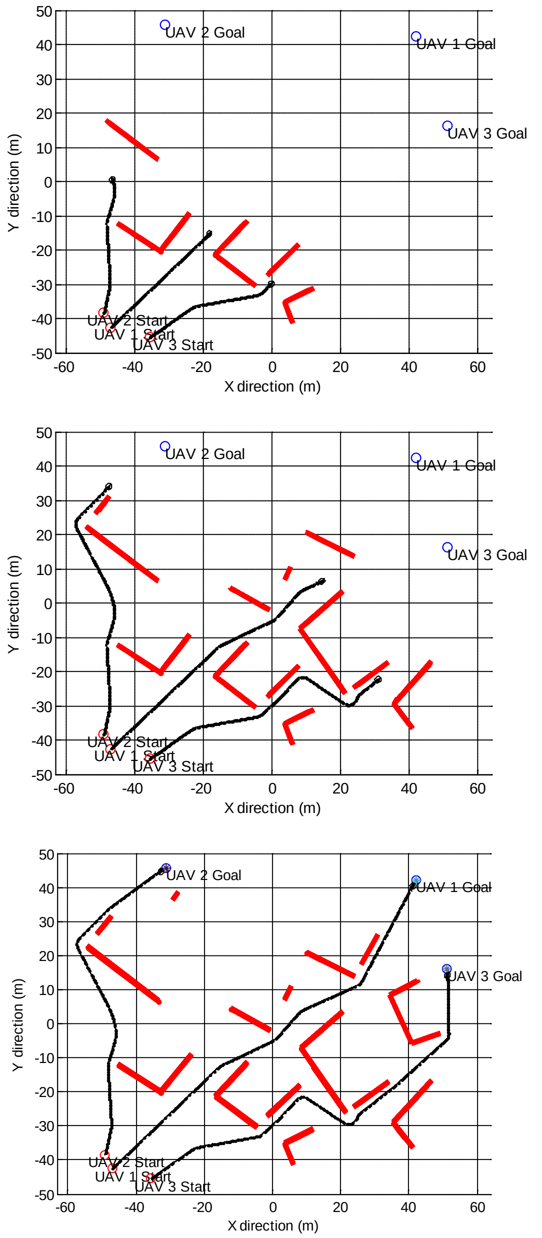

Fig. 5. Path Planning for Totally Unknown Environment

One of the methods, A* algorithm is based on the node. Therefore, a great deal of memory should be required to get the same result. In addition, because the path smoothing should be required if this algorithm is used, the cost time of that algorithm should be increased. On the other hand, though the proposed algorithm does not have to need this procedure, the result is almost same as the result from the other algorithms. This point shows how this algorithm is powerful contrast of the other algorithms.

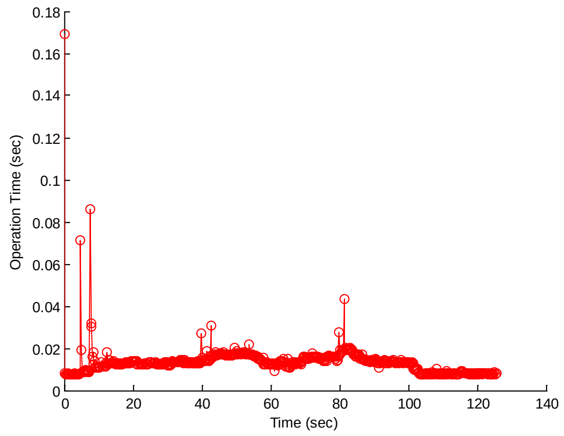

Fig. 6. Operation Time Profile

5 Conclusions

A real-time path planning algorithm to generate a suitable path in the dynamic environment is presented. The algorithms in former times have been separated to plan a certain path and to regenerate the suitable result by the dynamics for a certain generated path. Therefore, another procedure must have to be considered to apply the real vehicles. In addition, in the case of the dynamic environment, it is essential to use on the real-time path planning.

The proposed algorithm shows that those planning can be considered at the same time and it is powerful for the dynamic trajectory planning. In addition, this algorithm generates the useful path while it consumes lower cost time and memory contrast of the other algorithms. In addition, this algorithm is robust and powerful about disturbances because this proposed algorithm uses the probabilities and the differential geometry functions.

In addition, this algorithm can be applied to the multiple UAVs in a dynamic environment. Because this algorithm can be operated as a decentralized system, the proposed procedure is applied as an independent unit and the operation time is linear to the number of UAVs which is powerful rather than centralized system which cost time is rapidly increased.

Acknowledgements

The authors gratefully acknowledge for the financial support by Agency for Defense Development (09-01-03-04).

References

[1] P. Hart, N. Nilson and B. Raphael, A formal basis for the heuristic determination of minimum cost paths, *IEEE Transactions on System Science and Cybernetics SSC4*, vol. 4, no. 2, pp 100-107, 1968.

[2] T. Lozano-Perez, M. Wesley, An algorithm for planning collision-free paths among polyhedral obstacles, ACM 22, Commun, vol. 10, pp 560-570, 1979.

[3] D. Padmanabhan, Optimal 2-D path planning, AME 598C Project Report, 2000.

[4] Latombe, J. Robot motion planning, Kluwer Academic Publishers, 1991.

[5] P. Pettersson and P. Doherty, Probabilistic roadmap based path planning for an autonomous unmanned aerial vehicle, *In proceedings of the Workshop on Connecting Planning and Theory with Practice*, Whistler, 2004.

[6] T. Howard and A. Kelly, Optimal rough terrain trajectory generation for wheeled mobile robots, *International Journal of Robotics Research*, vol. 26, no. 2, pp 141-166, 2007.

[7] D. Shim and S. Sastry, A dynamic path generation method for a UAV swarm in the urban environment, *AIAA Guidance, Navigation, and Control Conference*, Honolulu, 2008.

[8] K. Yang, S. Gan, and S. Sukkarieh, An efficient path planning and control algorithm for an RUAV in an unknown and cluttered environment, *International Symposium on Unmanned Aerial Vehicle*, Reno, 2009.

[9] S. Lavalle, Rapidly-exploring random trees : A new tool for path planning. Computer Science Dept. Iowa State University, 1998.

[10] S. Moon and D. H. Shim, “Development of an Efficient Path Planning Algorithm using UAVs in Cluttered Environment,” *Institute of Control, Robotics and Systems Conference*, Chooncheon, 2010.