

# Path Planning for a Fixed Wing Micro Air Vehicle in Fuzzy Quadtree Framework

Sayan Ghosh<sup>\*</sup> and Abhishek Halder<sup>†</sup> and Dr. Manoranjan Sinha<sup>‡</sup>  
Indian Institute of Technology, Kharagpur, West Bengal, 721302, India

In this paper, a path planner is proposed for a fixed wing micro air vehicle (MAV), in the framework of fuzzy quadtrees. It is shown that a fuzzy version of the quadtree planner is apt for micro air vehicle applications with severe space, power and budgetary constraints. The algorithm can significantly reduce both space and time complexity and still yield an optimal and safe path for the vehicle. Typical MAV path planning issues like presence of pop-up obstacles, vehicle dimensions and safety measures for congested environment have been accounted in the code. In addition, the turn rate kinematics of the MAV has been considered as a constraint during quadtree generation.

## Nomenclature

$\alpha$	= spread angle of a pop-up obstacle as measured from the MAV
$d$	= nearest distance of the pop-up obstacle as measured from the MAV
$D$	= effective diameter of the pop-up obstacle
$\lambda$	= minimum quad length threshold considering the dimension of the MAV
$R_{min}$	= minimum turn radius
$\delta$	= factor of safety for minimum quad length threshold considering the turn rate kinematics
$\varphi_{max}^S$	= stall limited bank angle
$\varphi_{max}^L$	= load limited bank angle
$\varphi_{max}$	= maximum bank angle
$g$	= acceleration due to gravity
$V$	= commanded velocity

## I. Introduction

Micro Air Vehicles (MAVs) have gained considerable interest in recent times. Superior maneuverability, ability to make close surveillance and less chance of being detected are some of the unique features which make an MAV suitable for dull, dirty and dangerous missions. In most cases, such missions demand paths to be planned in congested urban or industrial environment. Further, in-flight detection of static pop-up obstacles makes MAV path planning a challenging task.

Unlike bigger Unmanned Air Vehicles (UAVs), MAVs can not afford much space for on-board processors. In fact severe space restriction often necessitates the obstacle information to be transferred from MAV to the ground station; then the planner runs at ground computer to plan and/or re-plan the path and then command is uploaded to the MAV. Hence it is imperative that the MAV path planner must ensure time and space economy. Classical approaches like visibility graph,<sup>1</sup> Voronoi diagram,<sup>2</sup> freeway net,<sup>3</sup> potential field method<sup>4</sup> are computation intensive for MAV path planning problem, particularly in the context of urban area navigation. UAV path planning problem has seen the use of model predictive control,<sup>5</sup> vector calculus,<sup>6</sup> receding horizon control,<sup>7</sup> GPS feedback<sup>8</sup> and evolutionary algorithm<sup>9,10</sup>. The computational cost for these methods often outweighs their accuracy. On the other hand, MAVs have stringent budgetary constraint to the extent that they can be used as expendable vehicles. Application of heuristic methods can significantly reduce the computational

---

<sup>\*</sup> Student, Department of Aerospace Engineering, IIT Kharagpur, India, sayan.iitkgp@gmail.com.

<sup>†</sup> Student, Department of Aerospace Engineering, IIT Kharagpur, India, ahalder@iitkgp.ac.in.

<sup>‡</sup> Professor, Department of Aerospace Engineering, IIT Kharagpur, India, masinha@aero.iitkgp.ernet.in.

complexity, hence relaxing the requirement of high processing speed and storage capacity of the system, two major factors in MAV budget. Further, research<sup>11</sup> has shown that formulating one existing path planner in fuzzy logic framework yields better result than the original planner itself. This paper introduces a fuzzy quadtree based path planner for MAV navigation and shows its superiority compared to quadtree planner itself.

The organization of this paper is as follows. In the next section, conventional quadtree planner is illustrated in detail. Then the concept of fuzzy quadtree planner is explained, followed by a discussion of some critical aspects of the problem. Simulation result for a real map is presented next. The last section concludes the paper.

## II. Conventional Quadtree Planner

The path planner proposed in this paper is based on the assumption that the environment is structured, meaning *the map of the obstacles is known beforehand*. Furthermore, it is assumed that the *obstacles are static* and the path planning amid such obstacles is assumed to be a *two-dimensional problem*. The rest of this paper will adhere to these assumptions. The only exception to these assumptions, which has been addressed in this paper, is the case of pop-up obstacles which were not anticipated a priori. Thus, strictly speaking, the environment is assumed to be *quasi-structured*.

The steps followed by the planner are: *quadtree generation, distance transformation, optimal pavement generation and optimal path generation with safety taken into account*. First, an image of the environment is taken. Then a quadtree<sup>12</sup> is generated for that image by recursive subdivision of that image until all sub-images (nodes) are either free (white) or of obstacle (black) type. Generally, quadtree representation is done through pointers. Matrix and linear quadtree<sup>13,14</sup> representation is adopted here instead. This is due to the fact that matrix and linear representation lead to significant execution time economy and memory saving, which are of utmost consideration for MAV path planner. Fig. 1 shows a typical quadtree generated and Fig. 2 shows all the gray (mixed), white (free) and black (obstacle) nodes with the numbering convention in the order of north-west, north-east, south-west and south-east.

Once quadtree is generated for the structured environment, a distance transformation is performed for the generated quadtree to assign proper weights to the free nodes. This is accomplished by generating an artificial wave from the specified goal point and by giving a distance value or weight to each node encountered during the propagation of ripples. This weight is directly proportional to the distance propagated by the wave front and the size of the node encountered. Weight assigned to the obstacle nodes are assumed to be infinity (since wave can not propagate through it). Fig. 3 shows the quadtree of Fig. 1 and Fig. 2 after distance transformation is performed. The goal point is shown by boxed G. The minimum weight zero is assigned to the goal.

Distance transformation results in a two dimensional weighted terrain where the goal has the lowest value of weight. Now the problem remains to traverse downhill, starting from the specified start to the specified goal, in the weighted terrain in an optimal manner. The optimality condition considered here is the shortest path criteria. Distance transformation results the shortest pavement and then it was decided to lay the safest path in that shortest pavement by joining the mid points of the entry and exit boundary of each square. This ensures a shortest safe path for the MAV.

Since in the present algorithm, quadtree representation is not done through pointers, each black node was encoded with a quaternary integer whose digits represent successive quadrant subdivisions. It can be shown<sup>12</sup> that, for the linear and matrix quadtree representation adopted here, the space complexity, measured in terms of number of nodes, is equal to number of black nodes in a linear quadtree, which is less compared to total number of nodes in case of a regular quadtree. From this, one can prove that, linear representation of quadtrees lead to at least 66 % saving in storage space compared to regular quadtrees.

Further, it can be shown that, the worst case time complexity for encoding a region is of the order of the product of total number of black pixels and quadtree resolution parameter. Again spatial operations like union and intersection of two quadtrees become very simple due to the use of linear quadtrees. For union operation, two linear quadtrees are merged in a new sorted array. For performing union operation on more than two regions, multi-way merge can be used. For intersection operation, a similar approach will help. These spatial operations can be extremely useful for piecewise path planning, where the entire map is first sub-divided into some important sub-maps and separate path planning is done for each of them. Hence union and intersection operations are important from overall mission planning point of view.

Thus the quadtree generation in matrix and linear quadtree approach is suitable from both space and time complexity consideration. Moreover, linear quadtree representation aids to perform region based spatial union and intersection operations on quadtrees. Since such operations are crucial in practical mission critical applications (e.g. taking decisions at the ground control station by analyzing surveillance images transmitted by the MAV), ensuring their fast execution and space economy is mandatory for hardware implementation of the algorithm.

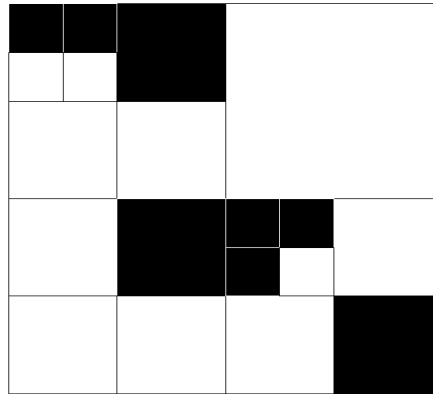


Figure 1. Quadtree of a two dimensional region with obstacle (*black*) and free (*white*) nodes.

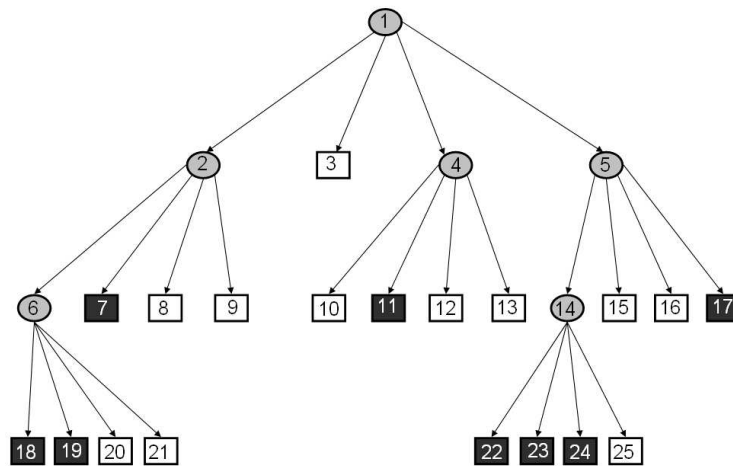


Figure 2. Tree representation of recursive decomposition process for quadtree generation.

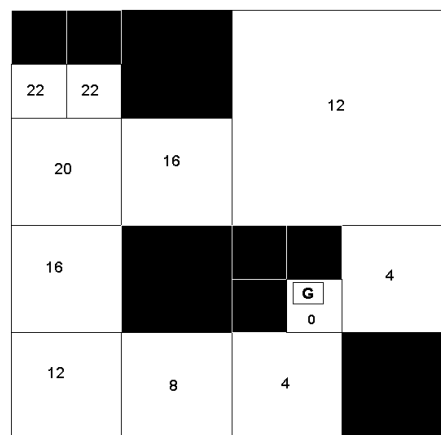


Figure 3. Quadtree with distance transformation values or weights.

To illustrate the optimum safest path planning method, a structured map of obstacles of size  $512 \times 512$  pixels is taken (Fig. 4) and then quadtree is generated for the same (Fig.5). Fig. 6 shows the optimal pavement (shaded free squares) and the safest path through it from specified start point (triangle in Fig. 6) to specified goal point (cross in Fig. 6).

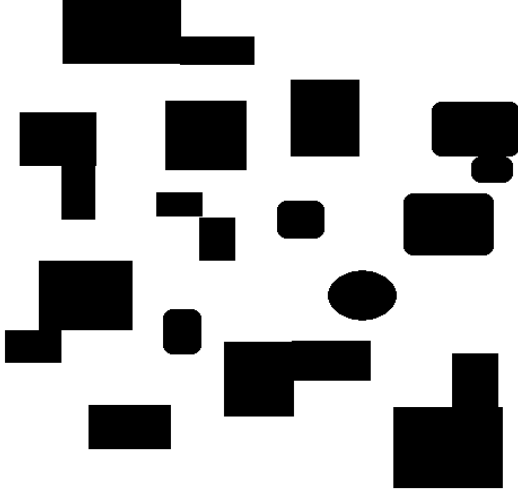


Figure 4. Structured obstacle map.

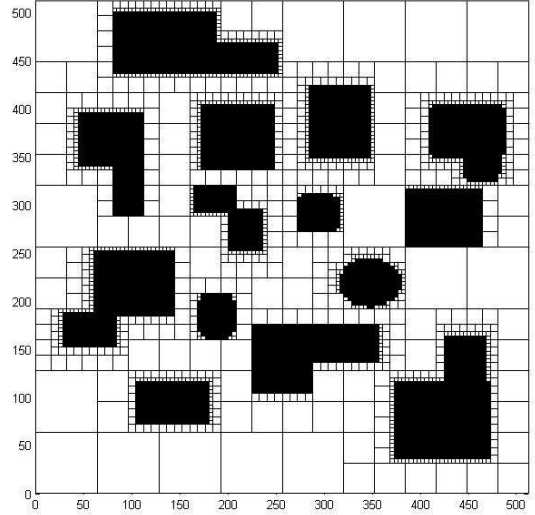


Figure 5. Quadtree for structured map.

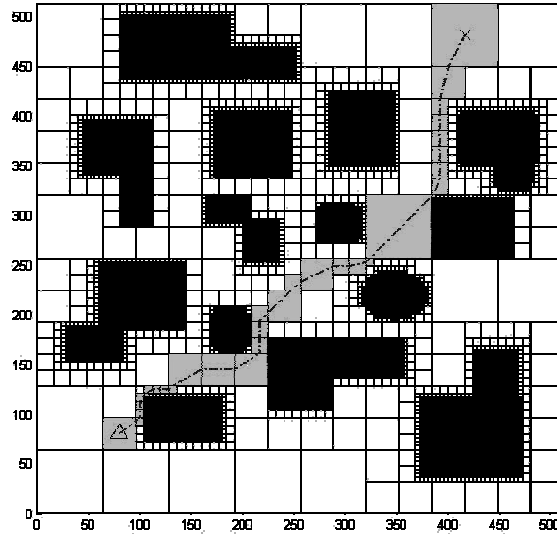


Figure 6. Optimal path for structured map.

### III. Fuzzy Quadtree Planner

The fuzzy quadtree planner is achieved by truncating a full fledged quadtree at a desired level and then assigning suitable membership function values to the resultant nodes at the truncated level. For that, one needs to count the total number of black and white daughter nodes under that level of gray nodes in which the user wants to stop. The user imposed fuzzy leaf node (which actually was a gray node) will then have a fuzzy membership function value equal to total number of black daughter nodes divided by total number of daughter nodes. Following this logic, if any white leaf node actually exists in the user defined level then its membership function value goes to zero (meaning it would have no black daughter nodes if it was not fuzzified). Similarly black leaf node, if exists, will assume the membership value 1. Clearly, this membership value reflects the

grayness of a node. The quadtree of Fig. 1 and Fig. 2 is fuzzified in Fig. 7 up to user defined level 2. The membership values of the nodes in level 2 are shown in Fig. 7.

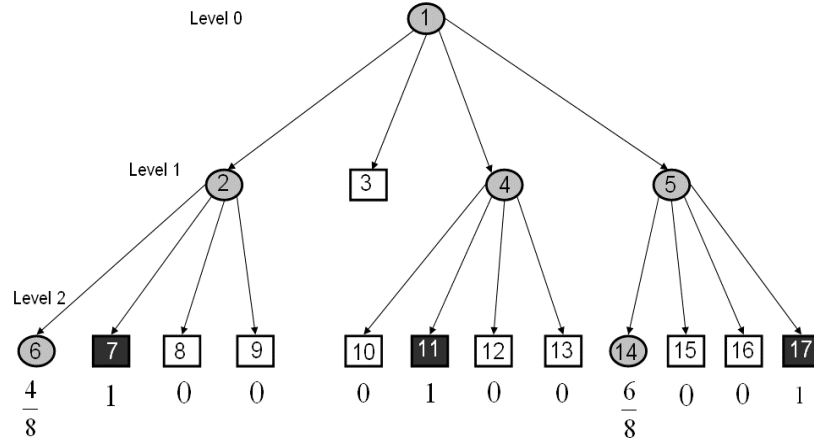


Figure 7. Fuzzy Quadtree up to level 2.

By assigning such fuzzy membership function values, the full-fledged quadtree is kept *folded*. The information of the actual quadtree can always be retrieved, if needed, from the membership values. Thus the seemingly truncation of the original quadtree does not lead to any loss of generality.

#### IV. Some Critical Aspects of the Problem

A quadtree planner, in general, has less time complexity compared to other conventional planners like Voronoi diagram (quadtree planner requires  $O(N)$  time and Voronoi diagram needs  $O(N \log N)$  time). Further, fuzzy quadtree planner allows the user to specify the level of quadtree that the planner will retain. In the context of MAV navigation, the map of the static obstacles generally consists of some dedicated themes like building, street etc. for urban environment. So the flexibility that the user can choose the desired level of details saves much memory and execution time. A comparison of Fig. 2 with Fig. 7 depicts this fact.

In fact, if the user demand of detailing is not very high, the path planner can run in real time. This allows one to apply the proposed algorithm for on-line MAV navigation in which the ground computer can get the feedback from on-board sensor (like mini laser ranger) and the planner runs at ground and uploads the path in the MAV.

This paper presents a solution to a more realistic on-line MAV navigation problem, where the MAV encounters a static pop-up obstacle which was not present in the structured map of obstacles. It can be due to limited clarity of the satellite image, inefficient image processing or due to time lag between actual snapshot and MAV flight (which can result in typical pop-up obstacles like trees or temporary constructions). This necessitates an update of the pre-planned off-line calculated path in real time.

During flight, if the MAV detects (using mini laser ranger) a pop up obstacle (gray circle in Fig. 8) in the midst of Fig. 4, then it treats it as a two-dimensional object and sends its breadth information (spread angle  $\alpha$ ) and nearest distance ( $d$ ) back to ground computer and a modified quadtree in Fig. 9 and an updated path in Fig. 10 is generated and uploaded to the MAV in real time. The modified quadtree is generated assuming the effective diameter  $D$  of the obstacle as

$$D = \frac{2d \sin\left(\frac{\alpha}{2}\right)}{\left(1 - \sin\left(\frac{\alpha}{2}\right)\right)} \quad (1)$$

Interestingly, here the actual pop up obstacle need not be of circular shape. Constructing a circle around it simply ensures safety of the MAV. Also one must note that, for successful real time operation of the proposed method, the uplink and downlink of data transmission should be as quick as possible. Here fuzzy quadtree can offer significant encoding efficiency by reducing the bit length of data transmission. Same holds for the case

when surveillance images are sent to ground control station, thus aiding quick initiation of action based on the decision taken at ground.

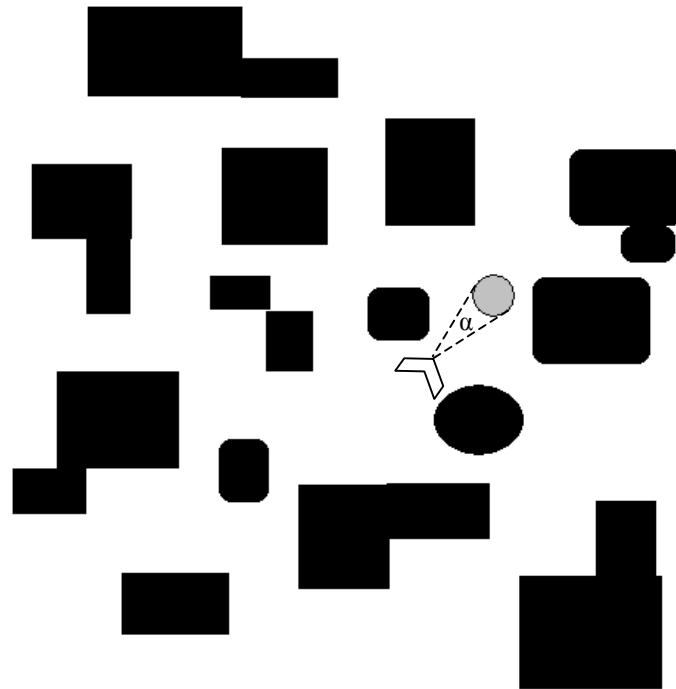


Figure 8. Pop-up obstacle in structured environment of Figure 4.

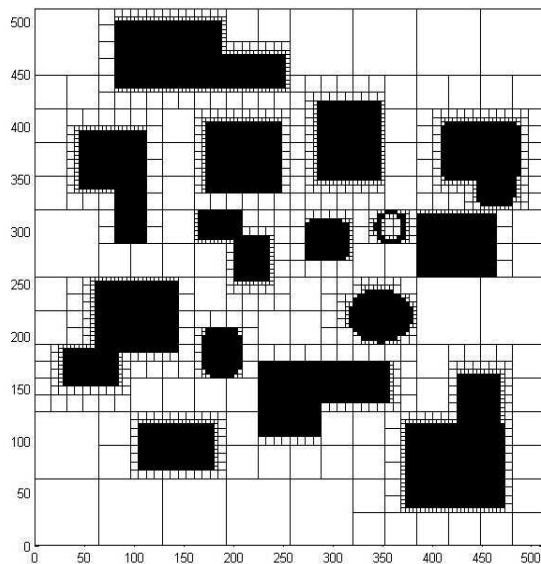


Figure 9. Modified quadtree.

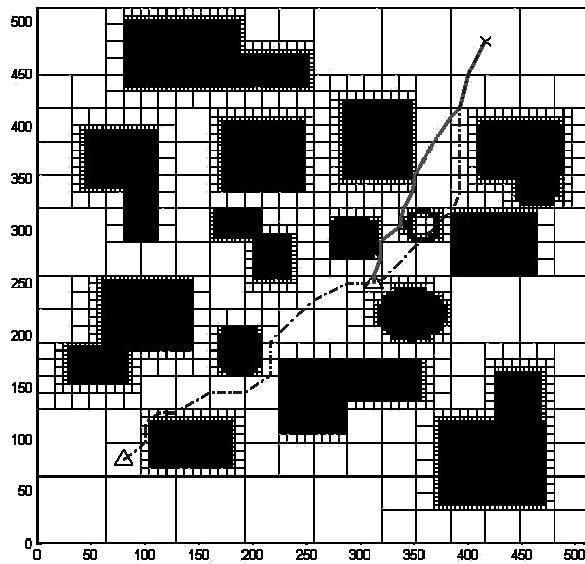


Figure 10. Comparison of the off-line planned (*dotted*) path and online updated (*solid*) path.

In conventional quadtree generation, the minimum quad length is one pixel. But this paper devises a strategy for specifying minimum quad length in the code for MAV planner. Two major factors in this context are vehicle's maximum dimension and its minimum turning diameter ( $2 R_{min}$ ). The minimum quad length will be the maximum of these two parameters, each one considered with a factor of safety i.e.  $\max(\lambda, 2 R_{min} + \delta)$ .

Again this paper deals with a fixed wing MAV with two elevons as its only control surfaces and having winglets for roll stability. From this point of view, turn rate kinematics of the MAV is considered to determine the minimum turning radius, since this is an important parameter for minimum quad length determination. The minimum turn radius ( $R_{min}$ ) corresponds to a maximum bank angle (since elevons are only control surfaces, the turning is accomplished through banking). This maximum bank angle is the minimum of stall limited bank angle  $\varphi_{max}^S$  (aerodynamic constraint) and load limited bank angle  $\varphi_{max}^L$  (structural constraint).

Thus to determine the minimum quad length, one must compute minimum turn radius  $R_{min}$  from the turn formula (assuming steady co-ordinated level turn):

$$R_{min} = \frac{V^2}{g \tan \varphi_{max}} \quad (2)$$

The  $\varphi_{max}$  in the right hand side of equation (2) corresponds to the maximum bank angle, calculate as explained in the above paragraph. It can be noted that the stall limited bank angle is governed by the lift coefficient versus angle of attack curve of the vehicle. The load limited bank angle is governed by the load-velocity diagram of the vehicle.

## V. Simulation Result for a Real Map

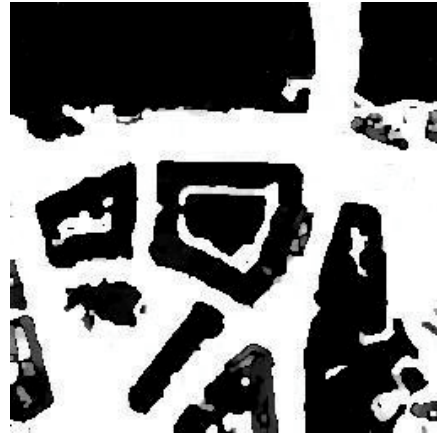
To demonstrate the efficacy of the algorithm, path planning has been performed for a real satellite image (of size  $256 \times 256$  pixels) of a part of Vatican City. The original image (Fig. 11) is processed digitally to remove noises from it. Various combinations of filters are applied for this purpose to yield an image (Fig. 12) which has clear obstacles and obstacle free regions. The path planning algorithm is applied next to generate a quadtree (Fig. 13) for this processed image. After that, distance transformation is carried out for the generated quadtree to result a weighted terrain where the goal has the lowest value of weight. The weights are depicted through colours in the contour map (Fig. 14). Then the optimal sequence of squares (pavement) is found (Fig. 15) by traversing downhill from the specified start to goal point. The sequence is constructed by successively following that obstacle free neighboring node which has the lowest value of weight. The safe and optimal path results (Fig. 16) by joining the mid points of the entry and exit boundary of each square.

## VI. Conclusion

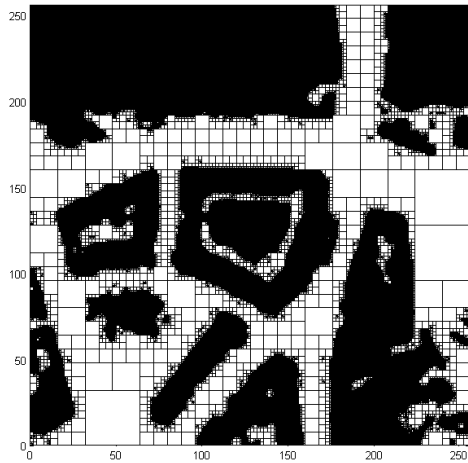
A fuzzy quadtree path planner is formulated in this paper for MAV navigation. The formulation addresses how to specify the minimum quad length depending upon vehicle's turn rate kinematics and MAV dimension. This, in fact, avoids pixel level computation and takes much less execution time and storage space compared to a conventional quadtree planner, which would otherwise be impossible to plan a path for MAV. Further, the heuristic formulation offers more flexibility by allowing the user to decide the level of planning complexity. Thus the proposed method is a faster, cheaper and better solution for MAV path planning problem.



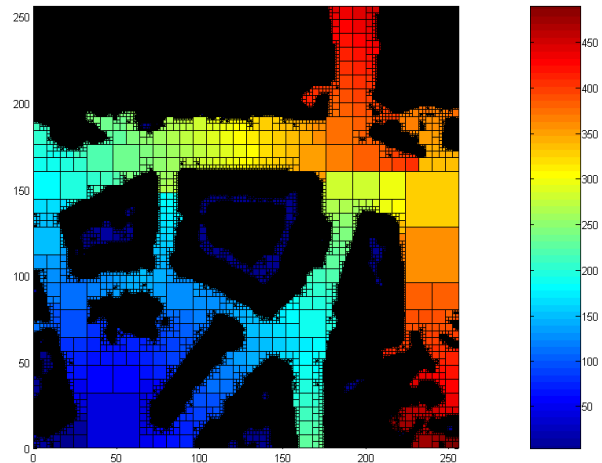
**Figure 11. Satellite image of a part of Vatican City.**



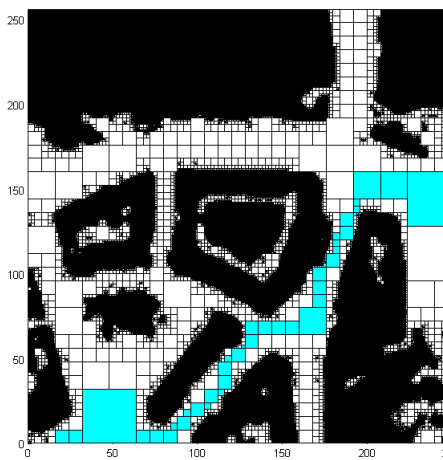
**Figure 12. Processed image.**



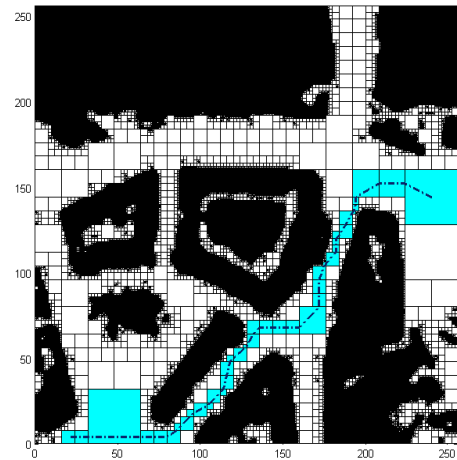
**Figure 13. Quadtree generated for the processed image.**



**Figure 14. Weight contour after distance transformation.**



**Figure 15. Generated optimal pavement for the processed image.**



**Figure 16. Generated optimal path for the processed image.**



## VII. References

- <sup>1</sup> Lozano-Pérez, T., and Wesley, M. A., "An Algorithm for Planning Collision-free Paths Among Polyhedral Obstacles," *ACM*, Vol. 22, No. 10, 1969, pp. 560-570.
- <sup>2</sup> Leven, D., and Sharir, M., "Planning a Purely Translational Motion for a Convex Object in Two-dimensional Space using Generalized Voronoi Diagrams," *Discrete Computational Geometry*, Vol. 2, 1987, pp. 9-31.
- <sup>3</sup> Brooks, R. A., "Solving the Find-path Problem by Good Representation of Free Space," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-13, No. 3, 1983, pp. 190-197.
- <sup>4</sup> Khatib, O., "Real Time Obstacle Avoidance for Manipulators and Mobile Robots," *The International Journal of Robotics Research*, Vol. 5, No. 1, 1986, pp. 90-98.
- <sup>5</sup> Singh, L., and Fuller, J., "Trajectory Generation for a UAV in Urban Terrain using Non-linear MPC," *Proceedings of the American Control Conference*, Arlington, VA, 2001, pp. 2301-2308.
- <sup>6</sup> Yang, G., and Kapila, V., "Optimal Path Planning in Unmanned Aerial Vehicles with Kinematic and Tactical Constraints," *Proceedings of the 41<sup>st</sup> IEEE Conference on Decision and Control*, Las Vegas, Nevada, 2002, pp. 1301-1306.
- <sup>7</sup> Kuwata, Y., "Real Time Trajectory Design for Unmanned Aerial Vehicles using Receding Horizon Control," M.S. Thesis, Aeronautics and Astronautics Dept., MIT, Cambridge, MA, 2003.
- <sup>8</sup> Bhat, S., and Kumar, P., "A Feedback Guidance Strategy for an Autonomous Mini-Air-Vehicle," *Proceedings of the National Conference on Control and Dynamical Systems*, Indian Institute of Technology, Bombay, 2005.
- <sup>9</sup> Nikolos, I. K., Valavanis, K. P., Tsourveloudis, N. C., and Kostaras, A. N., "Algorithm Based Offline/ Online Path Planner for UAV Navigation," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 33, No. 6, 2003.
- <sup>10</sup> Capozzi, B. J., "Evolution Based Path-Planning and Management for Autonomous Vehicles," Ph.D. Dissertation, Aeronautics and Astronautics Dept., Univ. of Washington, 2001.
- <sup>11</sup> Tu, K. Y., and Baltes, J., "Fuzzy Potential Energy for a Map Approach to Robot Navigation," *Navigation, Robotics and Autonomous Systems*, Vol. 54, 2006, pp. 574-589.
- <sup>12</sup> Gargantini, I., "An Effective Way to Represent Quadrees," *Communications of the ACM*, Vol. 25, No. 12, 1982, pp. 905-910.
- <sup>13</sup> Vörös, J., "A Strategy for Repetitive Neighbor Finding in Images Represented by Quadrees," *Pattern Recognition Letters*, Vol. 18, 1997, pp. 955-962.
- <sup>14</sup> Vörös, J., "Low-cost Implementation of Distance Maps for Path Planning Using Matrix Quadrees and Octrees," *Robotics and Computer Integrated Manufacturing*, Vol. 17, 2001, pp. 447-459.