



IMPACT OF DDA OPTIMIZATION ON MOBILE ROBOT PATH PLANNING FOR MIXED IMAGE IN IMAGE PROCESSING

Rama Kanta Choudhury
Department of Computer Science
Kalinga University
New Raipur, Chhattisgarh, India

Chandra Kanta Samal
Department of Computer Science
AND College, Delhi University
New Delhi, India

Abstract: Existing problem solving in the day to environment requires computational intelligence. Path planning is one of the most important technologies in the navigation of the mobile robot, which should meet the optimization and real-time requests. The objective of the paper is to present a noble approach to find the efficient and effective path planning for mobile robot. Here first the image is located on the graph and then a quadtree is formed, according to the working space image with respect to the obstacle image. Then the NFT algorithm is used to obtain the shortest path from the start point to the goal point in the graph. Finally the DDA optimization algorithm is adopted to get the optimal path. Aiming at the shortcoming of the DDA algorithm which is easily plunging into the local minimum, DDA algorithm with NFT is put forward. The results of the simulation demonstrate the effectiveness of the proposed method, which can meet the real-time requests of the mobile robot's navigation. Here we have taken two different types of image, one square shape and other is mixed image of different shapes like triangle and circle. The working space is tested and result is verified using NFT Algorithm with DDA optimization.

Keywords: DDA, Grid search, Quadtree, NFT (Neighbour finding technique)

I. INTRODUCTION

The area home land robotics has assumed a greater importance in the present age and robots are now used extensively to rescue survivors from dangerous environments when dealing with hazardous substances. Here the substance is taken as image and the image is presented as obstacle[2]. The goal of the path planning method is to determine a sequence of configurations for the robot to move around obstacles and avoid collisions while reaching a desired goal[11]. The Digital Differential Analyzer (DDA) method is widely used for planning the path of mobile robot. In a mobile robot path planning researchers used many algorithms for optimization and DDA is one of them and is considered to be subjected to other methods. This has two sections: first one is storing the location points in a vector array and second one is resolving the array step wise. In the DDA field method, we can imagine that all obstacles are represented by an image and we applied quadtree method on the image to make a tree[1]. once the tree is formed we applied the algorithm .The distance from the robot to obstacles will be judge on the basis of the tree structure. While the destination has followed with NFT (Neighbor finding technique) . Then we apply DDA on the path which is stored in the data base it has specific function and is finally the line of path is drawn. The function slopes down towards the target point, so that the robot can reach the target by following the path. We have applied this DDA algorithm on different shape or mixed image obstacles as shown in fig 9 and fig 10. we have applied DDA to 50 different locations and tested in c++ , here we have shown 10 different locations and as in the table III, and plotted the graph as shown in fig 13 and fig 14.

II. THE APPROACH

We have implemented DDA on A*, NFT algorithm to get the optimization .It is simulated and verified the result .The detail of the result and graph is shown in figure 3.

A. A*Algorithm

A* algorithm based implementation is easier and practically faster .to reach the destination, A*algorithm creates sub optimal paths using its neighbors. In A* representation, $f(n) = g(n)+h'(n)$, where $g(n)$ is the total distance from the initial position to the current position and $h'(n)$ is the estimated distance from the current position to the goal destination/state. To create this estimation a heuristic function is used. $f(n)$ is the sum of $g(n)$ and $h'(n)$ and is stated as the current estimated shortest path.

B. NFT Algorithm

Quad Tree

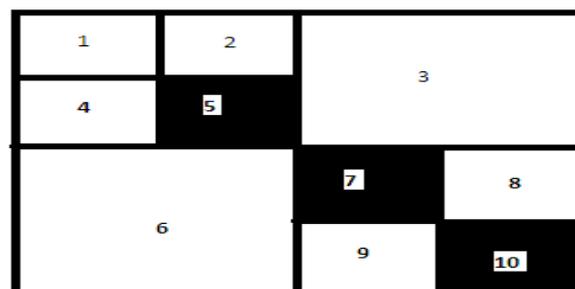


Fig 1 The image and its Notation

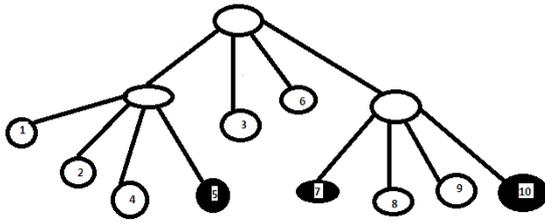


Fig 2 Representation of Quadtree

Image processing place an important role in the field of robotics path planning , here we have implemented the NFT (neighbor finding technique) and Applied DDA optimization over the NFT with quad tree approach[1,2,3].Here we have implemented,Equal_adj_neighbour,equal_corner_neighbour, Gte_adj_neighbour,Getqual_adj_neighbour,Get_corner_neighbour.

C. DDA Optimization

Let the path followed by the quad tree algorithm be stored in an array P1. Let P1 contains N points.

1. Check from 0 first point in P1 third element ,if a free path exist ,if yes store in a variable called Var.

Note ; Var initially contains 2nd element

2 .Check first point to 4th point in P1 if there is a path then overwrite with the element.

3 .Repeat the procedure until all the elements are scanned in such a way.

4. Let the Kth element be stored in Var stored in a new array []. Now repeat the above four steps from Kth Variable.

If the Kth variable is (n = i)th of the P1[], then add these points in P(New).

5. T last get the point and get the path. As

```

begin
For I = 0:n-2
begin
for j = 2 ; n
begin
if( path_bw(a [I ],a [j])
begin
var = a[ j]
end
end
b [++ counter] = var;
end
end
end
b[++ counter]= a[n-1]
end
    
```

III. OPTIMIZATION FOR OPTIMAL PATH FINDING

A. Application of DDA optimization

The DDA optimization network resolve the task to reach all the nodes on the map. The route from the starting node to the target node is resolved, but the solution is not optimal or close to the optimal different methods have been adopted to get the optimized path , DDA is one of them. The proposed DDA Algorithm for finding a closely optimal route between

the starting and the ending node is presented in the following:

- Elimination of the duplicate nodes. If the number of neurons is equal to the number of nodes on the map, the network does not find the solution, because a set of neurons will not be active during the network training process.
- Finding the address from the resulted array vector with the starting and ending nodes.
- The Array vector values which correspond to node coordinates on the map are compared with start and end node positions.
- Finding the nodes for which a neighborhood node with a lower cost to the target node exists.
- Calculating the cost from the start to the end node and from the neighborhood nodes to the end node for each node found in the previous step and storing the results in a table with the following fields: current node index, neighborhood node index, and the calculated cost.
- Ordering the table in ascending order according to the cost column.
- Deleting the higher cost overlaps the section with a Lower cost.
- Finally extracting the closely optimal path from the start to the end point.

IV. SIMULATED RESULT AND ANALYSIS

We have simulated A*, NFT algorithm with system configuration Intel® core(TM) i3-3220 GHz 3.30 and Ram-2GB (1.88 GB Usable).system type-32 – Bit Operating system. We applied DDA optimization on the oath finding and tested around 100 different locations i.e different start and goal points we fund that that the time taken and distance covered is very less ,as shown in fig 3,4,shows the unoptimized A*,f fig 4,5 shows optimized NFT , and fig 7,8 shows optimized A*. While comparing with the A* and NFT algorithm, the final output with time and distance covered presented in the graph. we have tested different shape images as shown in fig 11, 12 and tableIII and its graph which shows that the time and distance is reduced and get fully optimized. We have presented 10 outputs with its detail location and time and distance as shown in the fig 3and table1 and table 2. The comparison in graphical form as shown in fig 4.



Fig 3 A* Grid Search Algorithm Result

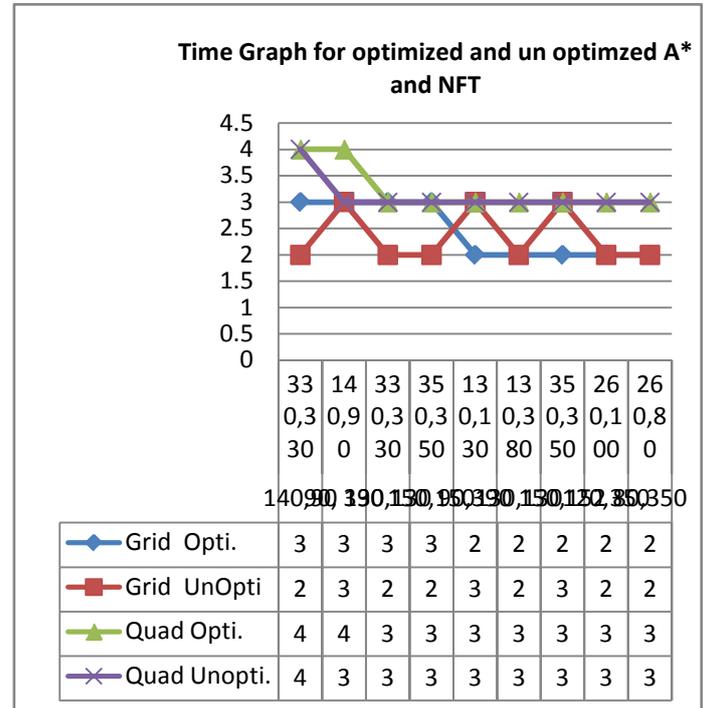
Table I Distance taken by the A*, NFT

Serial No	Start Point	Goal Point	Grid Opti.	Grid UnOpti	Quad Opti.	Quad Unopti.
1	140,90	330,330	3	2	4	4
2	90,390	140,90	3	3	4	3
3	130,130	330,330	3	2	3	3
4	150,150	350,350	3	2	3	3
5	90,390	130,130	2	3	3	3
6	130,130	130,380	2	2	3	3
7	150,152	350,350	2	3	3	3
8	120,350	260,100	2	2	3	3
9	80,350	260,80	2	2	3	3

Table II Distance taken by the A*, NFT

Serial No	Start point	Goal Point	Grid Un opti	Grid opti	Quad Un opti	Quad opti
1	140,90	330,330	800.42043	406.83041	606.66217	403.181763
2	90,390	140,90	299.404301	279.508484	307.263092	279.508484
3	130,130	330,330	559.010752	298.737732	567.652405	362.4644996
4	90,380	240,90	343.010752	336.4609	430.009308	326.156311
5	150,150	350,350	544.010752	295.853668	565.201	363.6339
6	90,390	130,130	250.803236	263.058929	264.728241	263.058929
7	130,130	130,380	240	125	251.209427	125
8	150,152	350,350	544.010752	295.853668	565.201294	363.633942
9	120,350	260,100	423.409677	334.588684	413.245544	312.856
10	80,350	260,80	338.61828	320.61417	428.5895	321.01
11	300,120	120,380	483.611828	507.457092	485.7	288.046

Fig 9 Time graph optimized Algorithm Result



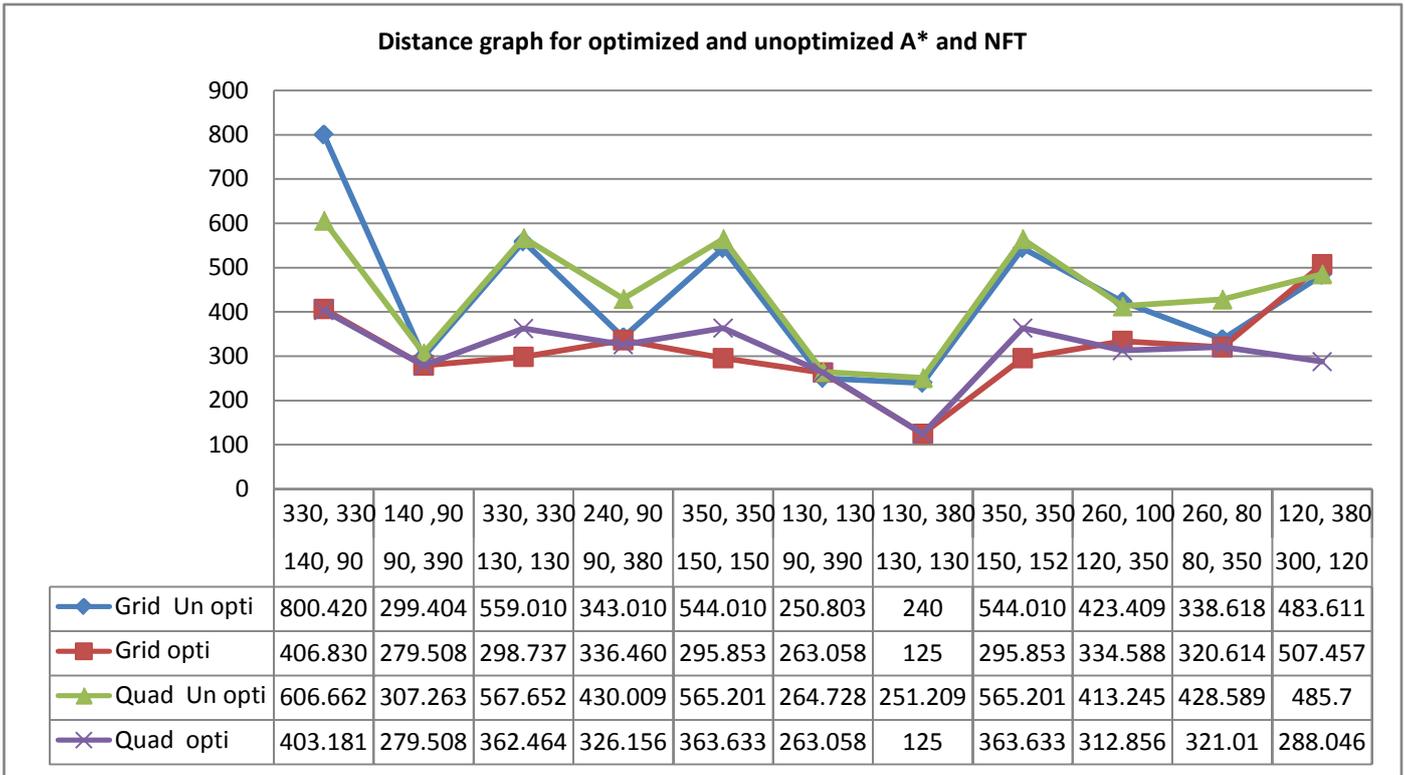


Fig 10 The Graph for Distance taken By A*, NFT

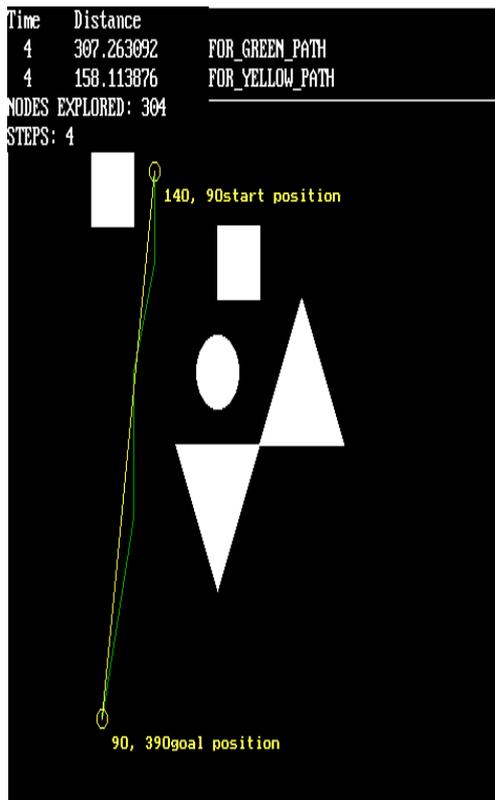


Fig 11 NFT optimized Algorithm Result

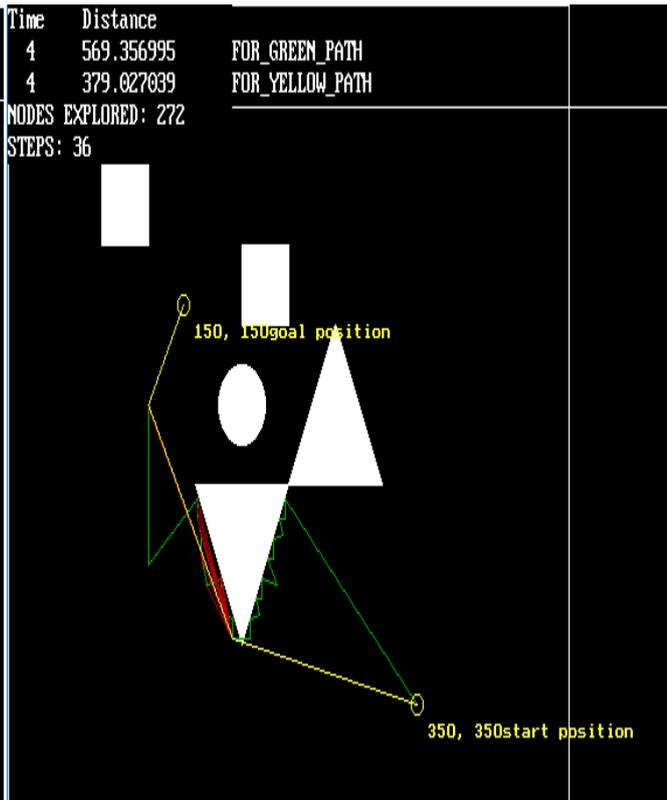


Fig 12 NFT optimized Algorithm Result

Table III Distance taken by the NFT different shape

Table IV Time taken by the NFT different shape

S.NO	Start Point	Goal Point	NFT without DDA (Distance)	NFT with DDA (Distance)
1	90,390	140,90	307.263092	158.113876
2	330,330	130,130	553.510986	364.208282
3	350,350	150,150	569.356995	379.027039
4	90,390	130,130	264.72821	117.04997
5	230,230	240,90	160.622574	65.764732
6	130,380	130,130	251.209427	250
7	220,380	160,160	360.796722	262.48114

S.NO	Start Point	Goal Point	NFT with out DDA (Time)	NFT with DDA(Time)
1	90,390	140,90	4	4
2	330,330	130,130	4	4
3	350,350	150,150	4	4
4	90,390	130,130	3	3
5	230,230	240,90	3	3
6	130,380	130,130	4	4
7	220,380	160,160	4	4

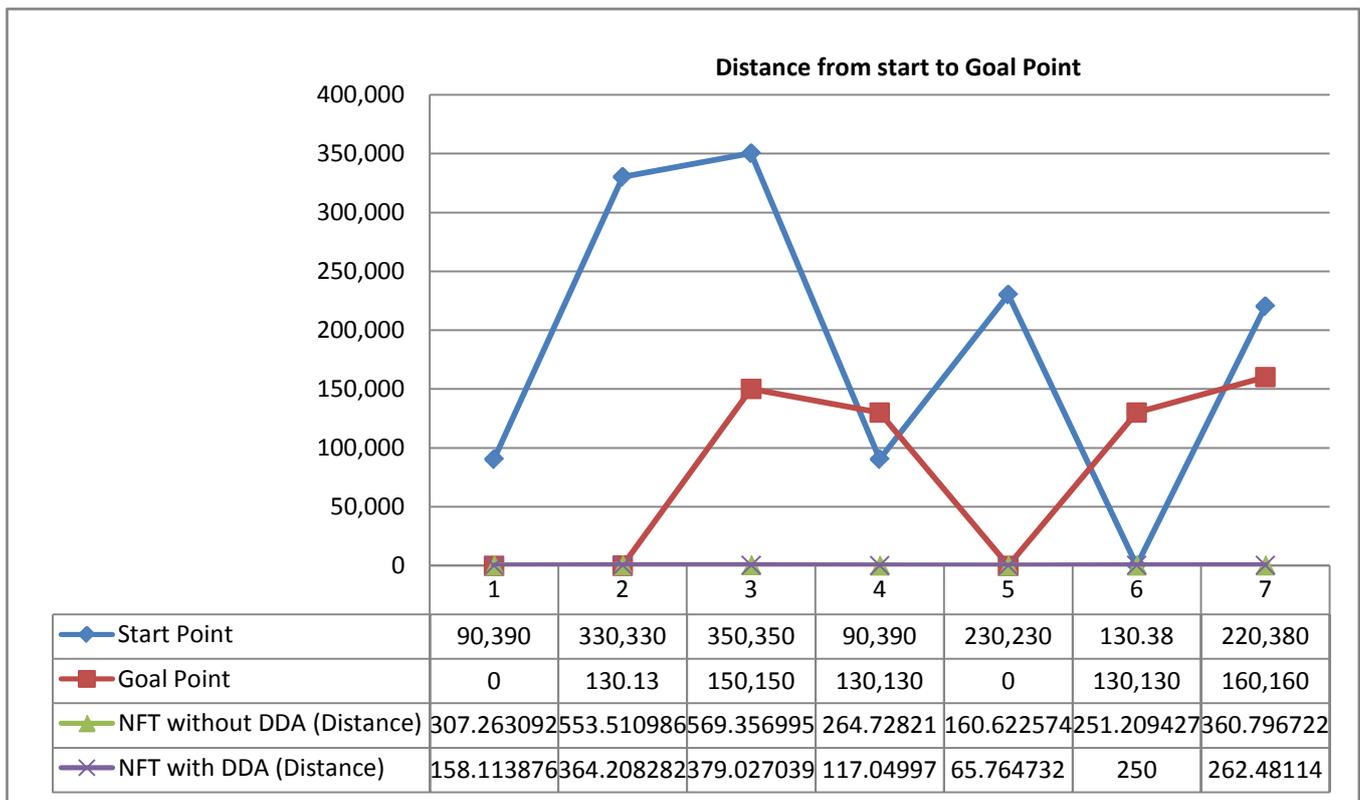


Fig 13 The Graph for Distance taken By NFT

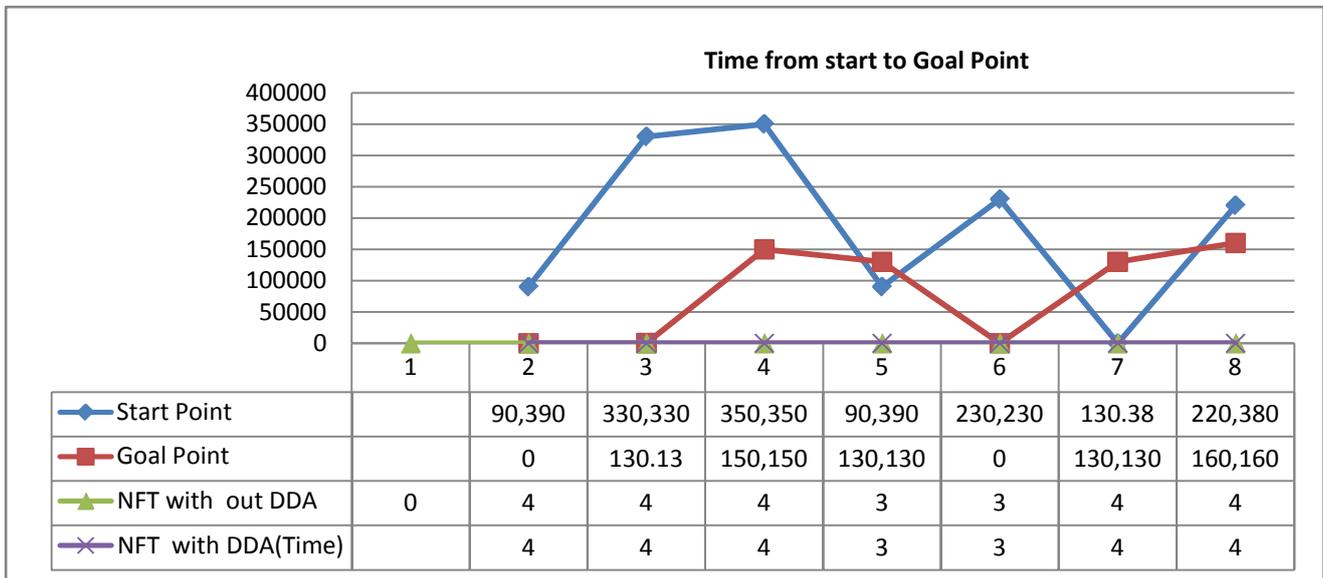


Fig 14 The Graph for Distance taken By NFT

IV. CONCLUSION

In this paper we have implemented DDA optimization technique on A*, NFT algorithm and tested the result taking different start and goal points. We found that the time and distance is reduced around 60 %. The amount of the existing works for each approach has been identified and classified and tested with C++ language. This paper divides the motion planning algorithms into two major groups, namely, the Conventional Approaches and Heuristic Approaches. The conventional approaches are Roadmap, grid search or Quadtree approach, here we tested in heuristic method also and the result as shown. A complete discussion of the portion of each approach in the field of robot motion planning is also presented, including different comparative figures and charts.

V. REFERENCES

[1] H.Samet, Region representation, quad trees from binary arrays, computer graphics and image processing 13, 1980, 88-93.
 [2] H.Samet, computing perimeters of an image s represented by quad tree, IEEE Trans, pattern Anal. Mach, Intel press.
 [3] H.Samet. An algorithm converting from roster to quad tree. IEEE Transaction, Pattern Anal, Mach. Intel press 3,1981,93-95.

[4] Rahul Kala, Dr.Anupam Shukla, Dr. RituTiwari, SourabhRungta, RJanghe “Mobile Robot Navigation Control in Moving Obstacle Environment using Genetic Algorithm, Artificial Neural Networks and A* Algorithm]”2009 World Congress on Computer Science and Information Engineering,978-0-7695-3507-4/08©2008IEEE,DOI10.1109/CSIE.2009.854.
 [5] Leela Rani.P, Rajalakshmi.P “Clustering Gene Expression Data using Quad Tree based Expectation Maximization Approach” International Journal of Applied Information Systems (IJ AIS) – ISSN: 2249-0868, Foundation of Computer Science FCS, New York, USA, Volume 2– No.2, June 2012 – www.ijais.org.
 [6] K. Daniel, A. Nash, S. Koenig, and A. Felner, “Theta*: Any-angle path planning on grids,” Journal of Artificial Intelligence Research, vol. 39,pp. 533–579, 2010.
 [7] K. Aizawa and S. Tanaka, “A Constant-Time Algorithm for Finding Neighbors in Quadtrees,” Pattern Anal. Mach. Intell. IEEE Trans., vol. 31, no. 7, pp. 1178–1183, Jul. 2009.
 Gargantini, I “An effective way to represent quadtrees communications of the ACM,25”, pp.905-10,1982.
 [8] G.M. Hunter and K. Steiglitz, Operation of image using quad tree, IEEE Trans. Pattern Anal. Mach Intell.1, 1979, 145-153.
 [9] Sebastian Thrun, Robotic Mapping :A Survey, School of Computer Science Carnegie Mellon University, February 2002.
 [10] Shahab Sheikh-Bahaei, "Discrete Event Modeling Simulation and Control with Application to Sensor Based Intelligent Mobile Robotics", M.Sc. Thesis, Electrical Engineering, The University of New Mexico, December, 2003.