IJASRM

International Journal of Advanced Scientific Research and Management, Vol. 2 Issue 8,August 2017.

www.ijasrm.com

ISSN 2455-6378

Mobile Robot Path Planning Approach: A Review

Rama kanta Choudhury[1], Sanjay Kumar[2] And Chandra Kanta Samal[3]

[1] Computer Science, Kalinga University, New Raipur, Chhattisgarh, India,

[2] Computer Science, Kalinga University, New Raipur, Chhattisgarh, India,

3 Computer Science, Delhi University, AND College, New Delhi, India.

Abstract

The shortest path planning approach and it's optimization for mobile robot in static and dynamic environment is the major task in the field of Robotics. In the present day environment finding collision free shortest path is the fundamental issue in the path planning. We have reviewed 100 papers on path finding technique in static and dynamic environment. Path planning in static environment is easy as compared to dynamic environment as in dynamic environment the obstacles are moving. Whereas optimizing the path is difficult in the static environment as well as in the dynamic environment. As per the timing is concerned, the path time is to be reduced so that the complexity is reduced. During the review process it is found that few algorithms have its own advantages and disadvantages as presented. Also a comparative study of different path planning techniques for static and dynamic environment is provided in this paper. The main focus in this review is to find the shortest path efficiently and effectively.

Keywords: *Path Planning, Algorithms, Grid search, Quad Tree, NFT (Neighbour finding technique), D*Lite*

1. Introduction

The survey is distributed into different sections as per our requirement, the main aim of this survey is to find the shortest path from source to destination with known and unknown environment in the presence of obstacles or when the obstacles come on the way. Mobile robots are authorized to move automatically from one point to other to reach their destination. They choose their own path to reach their destination without any hindrance with their preplanned path [1]. This paper provides a review of such papers based on their way to reach the target with the specified algorithms. Here we have reviewed many approaches and implemented. Efficient path planning approach is very important for mobile robot to follow the path. Here the main approach of the review is to find the different algorithms used and approached to find the shortest path from source to destination. Here the configuration of the robot is described by number of obstacles based on their position and sense of direction of the robot in the 2D and 3D environment [2]. A clear idea of planning and computing of collision free path is presented with given points with different algorithms. While reviewing papers we found that in mobile robot navigation, researchers used many algorithms out of which are A*, D* and D*Lite algorithm with grid search and quad tree methods [2][3][4][10][12][13]**.** ***The task of moving robots in mapped environments is a step by step approach; planning of the paths, optimal by certain criteria and controlling the robot to execute the planned paths. In this path planning, the task of finding a collision free path is to lead a robot from the initial configuration to the goal configuration among a set of obstacles. Generally, obstacles are modeled as polygons. The initial and the goal configurations are described by the***

corresponding positions and orientations of the robot. It provides certain additional criteria such as (i) the total length is shorter. (ii) The path is much smoother (iii) the path is kept at a safer distance from obstacles. (iv)the complexity is much lower. The mobile robot cannot make a complete picture of the surroundings and hence it cannot distinctly identify spatial spread of the object. The conventional path tracing can be divided broadly into two categories: the regular grid search and vertex graph methods. The second method makes elaborate representational changes to convert to a representation, which is easier to analyze before planning the path. A potential and practical shortcoming of such method for mobile robot navigation is that the path tracing cost is still very high as it involved representation conversion process.Aquad-tree-based hierarchical representation method of path tracing for mobile robots has been elaborated in this paper. Further; a staged search based method for initializing the hierarchical method of representation has been described. The first part is a Quad tree approach based on A* search and the second part, a staged path tracing algorithm, which has computational advantage as compared to the pure A*search on Quad tree. Yellow blocks are introduced in the staged search method.

Let us focus on the basics of quad tree**.** A quad tree is a two dimensional picture uniformly colored on $2^i \times 2^i$ blocks. A block of a quad tree has been represented on $2^j \times 2^j$ square region of the picture. A free block of a Quad tree is a block representing a region of free space. An obstacle block is a block representing a region having a mixture of free space and obstacles. A leaf block of a Quad tree is the tip block of the tree. In an ordinary Quad tree, leaf blocks are free blocks, but in pruned Quad tree, they may also be Yellow blocks. For any block G, S(G) denotes the sub tree .and L (G) denotes the number of leaf blocks in S (G). The Yellow content of a Yellow block G is defined as the number of obstacle pixels in the region represented by G, and the Yellowness of G is the percentage of obstacles in that region. When some of the Yellow blocks of a Quad tree are made leaf blocks, the resulting structure is called pruned Quad tree. It represents same space as the original Quad tree but with reduced resolution [13].

D* Algorithm

D* is an interpolation based planning algorithm that alleviates the problem. The D* maintain the open list of states[41].The open list use to propagate information's about change to the arc cost and function and to calculate path costs to states in the space. Every state has an associated tag t(x), such that t(x) = NEW if X is never been on the open list, t(X) =OPEN if X is currently on the open list and t(x) is closed, if X is no longer on the open list for each X. D* maintains an estimate of the sum of the arc costs from X to G given by the past cost function h(G,X), given the proper condition that the estimation is equivalent to the optimal minimum costs from X to G .

D*Lite Algorithm

The optimized version of D* Lite, when a state s is proposed whose cost is increased so that rhs(s) > g(s) each effected neighbor of "s" recomputed its rhs(s) value. However there is a chance that some of these neighbors recomputed their values number of times. At their new successor states may be proposed increased costs and so on[42].We can avoid this multiple values and update the new values of the neighbor states to infinity, rather than recomposing its true value. Since the state is inserted in to the open list with a key value based on its g-value, not its rhs(s) value, there will not affect its priority. The one exception to this is when the state is already in the open list with its g value greater than or equal to its rhs value .To account to this possibility, we thus check if this is the case and if so we recomputed a new rhs value for .with such a grid we can extract a graph for path planning easily. The fig 1(a) shows the node and edge extraction process for one cell in a uniform 2D environment. A standard grid uses for path planning, in which nodes resides at the centre of each grid cell. The arcs emanating from the centre node represents all the possible actions that can be taken from this node. A modified representation used by fields*,in which nodes reside at the corners grid cell and he9nce optimal path from node s must intersect one of the edges.

2. The Quadtree Approach

The quad tree initially comes up with [4]. It is an approach where the complete tree is represented as a square and the square is divided in to four sub branches of equal size as shown in fig 1(a) as row1 column1(R1C1) and row1column2(R1C2) likewise, which has four different quadrants as shown in fig 1(b, c). While dividing the [6] image into four quadrants each node has four sons, each node is represented by(R1C1,R1C2,R2C1,R2C2).

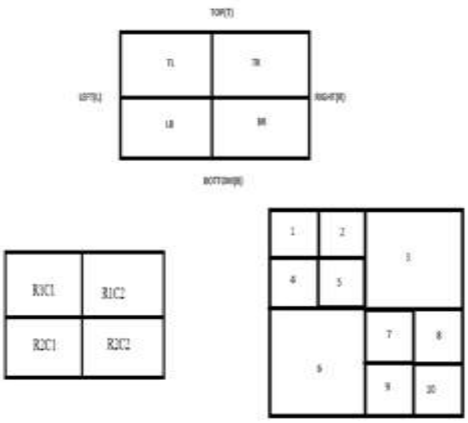


Fig1(a)Representation of image with notation Fig1(b) Location of the position of an image Fig 3(c) Numbering the cell

Fig1(c) Numbering the image as per the quad tree. Here the quad tree is a well organized composite configuration which helps to represent the image in a better way. Assumption of existence of link is not permissible here. The extra space required will be reduced and optimized path will be found. In diagonal direction the adjacent nodes is taken into consideration rather than horizontal and vertical direction. We validate the diagonally adjacent nodes sequentially related to previous node, once this approach is finished we get the source of shortest path. In this quad tree approach, the image is first imposed in the square and is checked whether the image has completely occupied the square or not, if not then the square is further sub divided into sub squares until and unless the image completely comes inside a square. Once it comes completely inside a square then it treated as "BLACK" otherwise it is treated as "GRAY". The black and white nodes are treated as nodes whereas the "GRAY" node is further subdivided into sub squares. An image with obstacles is shown in fig 2 (a) and fig2 (b) shows its quad tree and the image is imbedded in black having notations and with respect to the R1C1, R1C2, R2C1 and R2C2. The tree and its quad tree image related to the location of the image is marked in the tree. The black is marked as obstacles [3,5].

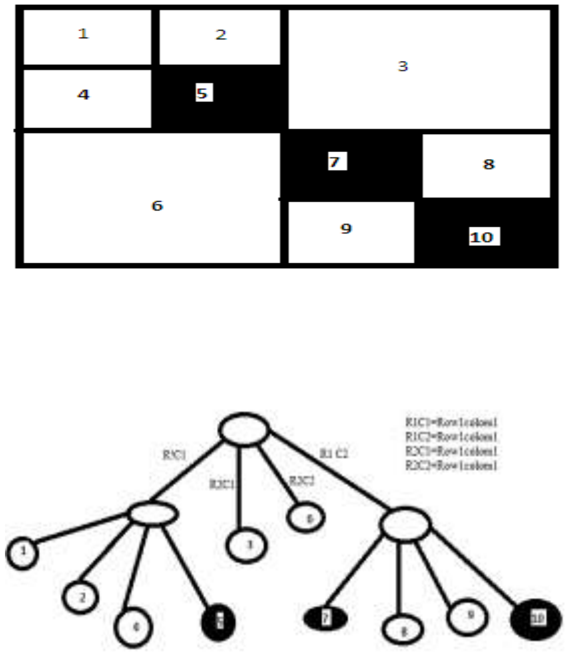


Fig2 (a) Complete region with obstacle Fig2(b) Quad tree representation.

3. Search Algorithm

Here we assume that each node is stored with six fields, out of which five fields are main field and having its four sub fields called as braches. If M will be the main node and Q is the branch, it will be treated as MAIN(M) and SUBFIELD(M,Q). We can treat relative branches to its main by using subfield as SUBFIELD(M).having value of Q if SUBFIELD (Main(M),Q)=M , the field are node type, specifies inside the branch of the image which represents "WHITE" if block contain no image or "BLACK" contains only unit number if having image ,and "GRAY" zero or one. BLACK and WHITE are represented as branch nodes where as GRAY are non terminal nodes .Let the four sides of a node is called as R1C1, R1C2, R2C1, R2C2 shown in fig 2(a,b). We use Some anticipation and functions involving certain quadrants and boundaries ADJACENCY(R,S) it is safe only if the branches S is adjacent to boundary R of the node block that is Adjacency (L,BL) is safe. REFLECTOR(R, S) represents as the SONTYPE having block of equal size that is the nearest side of R of the block having SONTYPE,
REFLECTOR (T, BL) =TL,
SAMEBOUNDARY (R1, R2) and representing block R1 and R2 are not adjacent blocks.

IJASRM
International Journal of Advanced Scientific Research and Management, Vol. 2 Issue 8,August 2017.
www.ijasrm.com
ISSN 2455-6378

Table1 Adjacency, Table2 Reflector

Table 3 opposite block ,Table4 Same Boundary

Adjacency	TL	TR	BL	BR
T(Top)	T	T	F	F
B(Bottom)	F	T	F	T
L(Left)	F	F	T	T
R(Right)	T	F	T	F

Table 1 Adjacency

Mirror	TL	TR	BL	BR
T(TOP)	BL	BR	BR	BL
B(Bottom)	TR	TL	F	T
L(Left)	BL	BR	TL	TR
R(Right)	TR	TL	BR	BL

Table 2 Reflector

Opposite Block	Block
TL	BR
TR	BL
BL	NE
BR	TL

Table 3 Opposite block

Block1 to Block2	TL	TR	BL	BR
TL	NO	T	L	NO
TR	T	NO	NO	R
BL	W	NO	NO	B
BR	NO	E	S	NO

Table 4 Same boundary

4. Shortest path planning approach using quad tree

Here the algorithm is used for the path planning for mobile robot based on the quad tree representation for immediate use of the robot. If there is a big free space then the complete area is taken as the free space and is represented in a square shaped block as shown in Fig. 1, and the free space is represented as Top(T), Left(L), Right(R), Bottom(B). The square is represented as TL=Top Left, TR=Top Right, LB=Left Bottom, BR=Bottom Right. The static image is represented in black coloured squares which are numbered as 5,7,10. In quad tree approach we first divide the complete square to four quadrants and is named as shown in Fig fig2(a), after naming four quadrants and four quadrants are forming four branches of a tree and we divide the quadrant which is having the image in black colour as TL (Top Left) quadrant and BR(bottom Right) quadrant to equal shaped 4 quadrants .Dividing both TL, BR we start numbering the quadrants horizontally as shown in the table1,2,3,4.While dividing the quadrant it should checked weather the image is completely filled or not otherwise the quadrant is again divided in to four quadrants till the image is filled with image in black colour [5]. The complete tree is shown in fig2 (b).

The 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.

Algorithm as follows:

Establish a search graph G, where start node is $P_{0.}$ Put P_0 in the list SAFE. Generate a list which is initially empty. If SAFE is empty, then treat it as unsafe. Select the first node of SAFE and remove it from SAFE, AND put is on CLOSED, and name it as node P.
If P is the goal node terminates at goal, obtain a path from P to P_o, establish step 7.

- Expand node P, establish the set Z of its successors that are not already ancestors of n in G. Install these members of Z as successors of P in G.
- Establish a pointer to n from each of those members of Z that were not already in G (i.e., not already on either SAFE or CLOSED). Add these members of Z to SAFE. For each member, z, of Z that was already on SAFE or CLOSED, redirect its pointer to P if the best path to m found so far is through P. For each member of Z already on CLOSED, redirect the pointers of each of its descendants in G so that they point backward along the best paths found so far to these descendants.
- Reorder the list SAFE in order of increasing f values. (Ties among minimal f values are resolved in favor of the deepest node in the search tree.)
- Go to Step 3.

5. The Neighbour Finding Algorithm

In this algorithm we proposed four different functions followed by a function SD(M,I), it is same if the quadrant I is just beside of boundary M of the node block. JUSTOPPOSITE(M,I) approaches to SONTYPE of the same block that is adjacent to boundary B of the node block. COMMONSIDE (S_1 and S_2) indicates the same side boundaries. The opposite quadrant do not share block boundary with the quadrant that is OPPOSITEQUAD(S). If S_1 and S_2 are not adjacent brothers then they share same side which is undefined. This S_1 and S_2 represent adjacent brother quadrants.

The steps of neighbour finding algorithm can be listed as follows:

- A node is given to specific block of the image.
- Here each node has its own horizontal or vertical direction.
- We search for SD or JUSTOPPOSITE or COMMON_FOR_BOTH or OPPOSITE_QUAD.
- After this it is followed by certain functions which are
- Identical_ adj_ neighbor (K,L) that is it locates an equal sized neighbour of node “P” in the Y direction or X direction.(Horizontal X and Vertical Y).
- Identical_corner_neighbour which identifies same size neighbour of node K in the direction of O.
- Get identical_adj_neighbor (K, L) to identify a neighbour of node K in X or Y direction of L. If it is not available then return back. Get identical_adj_neighbor (K, L, E, F) which represent return of ‘S’ the neighbour of node K in X or Y direction in L and F represents the level of the tree at which node K is normally found and the tree at which node S is ultimately found. If such node does not exist then return to NULL.

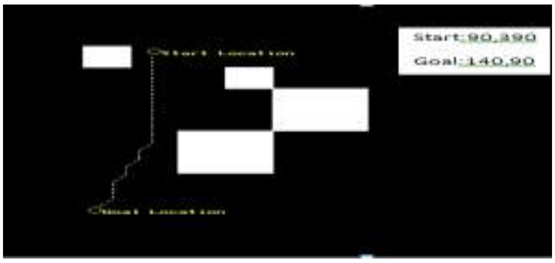
D* Lite Algorithm

In this algorithm the robot Main ( ) need to move the robot along the path determined by calculate path( ). Main( ) could calculate the priorities of the vertices in the priority queue every time the robot notice a change in the edges cost after it has moved.

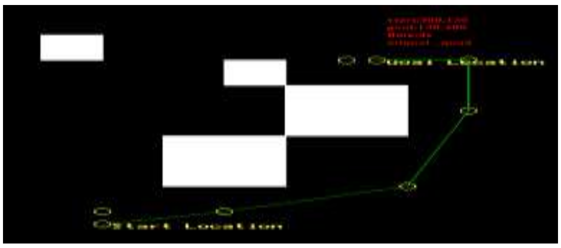
1. First it will set start and goal point.
2. From the starting point the robot will search for eight adjacent neighbors and continuously it will calculate the cost of the nodes, the small node cost will put in a list.
3. The neighbor may be ≥ 8.
4. It will continue as f = shortest to neighbor+ the distance to neighbor to goal.
5. It will continue till the current node = node ± goal.
6. Stops.
7. Finally it finds the path.
   Compute shortest path ( ) of D* Lite shares many properties like it seares each vertex at most twice until it returns.The result shows that ComputeShortestpath ( ) of D*Lite terminates and correct.

7. Simulation Results and Discussion

The comparison of A*, NFT, D*Lite using A optimization technique is shown in figure Fig 3(a,b,c) with difference in their time and distance graph is plotted in Fig 4 (a , b) .

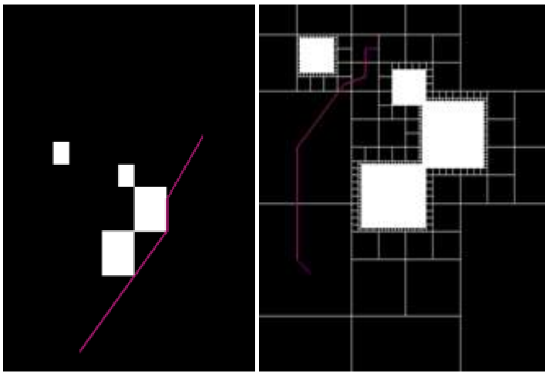


3(a) A* Grid search



3(b) NFT with quad tree

IJASRM
International Journal of Advanced Scientific Research and Management, Vol. 2 Issue 8,August 2017.
www.ijasrm.com
ISSN 2455-6378



3(c) D* 3(d) D* Lite

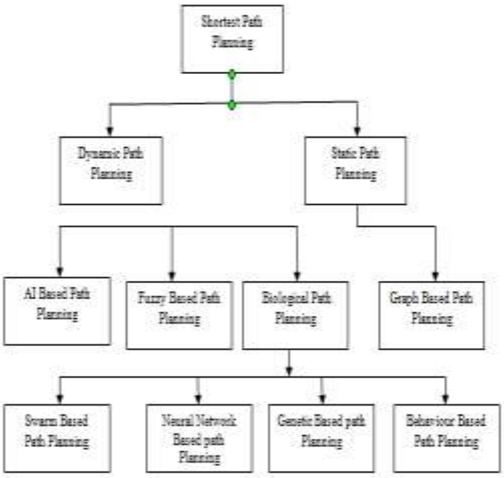


Fig4(a) Classification of path planning Techniques.

In last three decades several papers were reviewed in the field of robot motion planning. In many of these papers, different comparisons among the approaches are provided. The researchers tried to bring together major applications of conventional and heuristic techniques and to come to conclusions about the nature and the course of research in motion planning discipline. Initially some simple methods of motion planning were applied but later on some Complex methods were developed. To increase the effectiveness and efficiency of the path planning the heuristic methods were developed. As illustrated in Table *2* (considering a total of 100 papers), the application of the heuristic methods was increased due to their success in coping with the problems of combinatorial explosion. As shown in Table 2 the 80th decade was the climax period for the conventional methods and, the appearance period for the heuristic methods. Meanwhile, the 90th decade was the descent time for the conventional approaches and improvement of the heuristic ones. The last 10 years (2001-2015) were the climax period for the heuristic methods. It seems that in the near future, the application of the heuristic approaches will decline, whereas the application of the compound and heuristic methods will improve in order to achieve some better solutions in shorter time and lesser cost, but with more effectiveness and efficiency. Tables 2 represent the quality. In total, about 25% of the papers are related to conventional approaches and 75% to heuristic. The severe difference between the portions of the conventional and heuristic methods indicates the rate of interest to the heuristic methods, according to their ability to decrease the time and error.

Table 2 Difference and comparative study of Different path planning techniques

No	Static/ Dynamic	Algorithm	Features	Efficiency	Effectiveness
1	Static	Genetic	Increases Efficiency & decreases computational Time	Better	Best
2	Static	A*	Better Efficiency Not good when shape changes	Good	Good
3	Static	NFT	Better Efficiency Not good when shape changes	Better	Good
4	Static/ Dynamic	D*	Increases Efficiency & decreases computational Time	Better	Good
5	Static/ Dynamic	D* Lite	High Efficiency , Best computational time	Best	Best
6	Static	MWF	Computational time increases	Good	Good
7	Static	FWF	Average Computational time	Better	Good
8	Static/ Dynamic	OFWF	Better Computational time	Better	Good

Conclusion

In this paper, after analyzing about 100 papers in the field of robot motion planning approaches, the amount of the existing works for each approach has been identified and classified. This paper divides the motion planning algorithms into two major groups, namely, the Conventional Approaches and Heuristic Approaches. The conventional approaches are Roadmap, Cell Decomposition, whereas the heuristic approaches include the Neural Network, Genetic Algorithms, Ant Colony and Fuzzy Logic. After a brief introduction of each approach, the important works in each field were presented. 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.

References

[1] Gargantini, I "An effective way to represent quadtrees communications of the ACM,25", pp.905-10,1982.

[2] KunioAizawa,Koyo Motomura,Shintaro Kimura , Ryosuke Kandowaki and jia Fan "Constant Time neighbor Finding In quad Trees: An Experimental Result ISCCSP @))*,Malta,12-14,March2008,978-1-4244-1688-2/08 2008,IEEE.

[3] Finkel.R.A and j. l. Bentley, Quadtree: A data structure for retrieval on composite keys, Acta Informatics, 4,pp.1-9,1974. [4] G.M. Hunter and K. Steiglitz, Operation of image using quad tree, IEEE Trans. Pattern Anal. Mach Intell.1, 1979, 145-153.

[4] Abraham Sanchez Lopez, Rene Zapata and Maria A.Osariolama"Sampling based motion planning: A survey" computation y systems Vol.12 No,1,2008, pp5-24,Issn 1405-5546.

[5] H.Samet, Region representation, quad trees from binary arrays, computer graphics and image processing 13, 1980, 88-93.

[6] H.Samet, computing perimeters of an image s represented by quad tree, IEEE Trans, pattern Anal. Mach, Intel press.

[7] H.Samet. An algorithm converting from roster to quad tree. IEEE Transaction, Pattern Anal, Mach. Intel press 3,1981,93-95.

[8] Sven Koenig; maxim Likhachev " D* Lite" American Association for Artificial Intelligence copyright 2002.www.aaai.org.

[9] Cheol-Taek Kim and Ju-Jang Lee "Mobile Robot Navigation using Multi-resolution Electrostatic Potential Field" 0-7803-9252-3/05/ ©2005 IEEE.

[10].Ronald C.Arkin, *Behavior-Based Robotics*, The MIT Press, 1998.

[11] Alberto Elfes,"Using Occupancy Grids for Mobile Robot Perception and Navigation," *IEEE. Computer*, vol. 6, pp. 46–57, June 1989.

[12] Nikos C.Tsourveloudis,Kimon P.Valavanis, and TimonthyHebert,"Autonomous Vehicle Navigation Utilizing Electrostatic Potential Fields and Fuzzy Logic," *IEEE.Trans.On. Robotics and Automation*, vol.17, pp. 490–497, August 2001.

[13] Kimon P.Valavanis, Timonthy Hebert, Ramesho Kolloru and Nikos C. Tsourveloudis, "Mobile Robot Navigation in 2-D Dynamic environments Using an Electrostatic Potential Field," *IEEE.* Trans. On. Systems, Man, Cybernetics. Part A, vol. 30, pp. 187–196, March 2000.

[14] J. Hutchinson, C. Koch, J. Lea, and C. Mead, "Computing motionusing analog and binary resistive networks,"IEEE. Comput. Mag., vol. 21, pp. 52–63, 1988.

[15]L.TarassenkoandA.Blake,"Analogue computation of collision free paths," in *Proc.* IEEE Int. Conf. Robotics and Automation, pp. 540–545, 1991.

[16] Sebastian Thrun, Robotic Mapping :A Survey, School of Computer Science Carnegie Mellon University, February 2002.

[17]Sebastian Thrun, Probabilistic Algorithms in Robotics, School of Computer Science Carnegie Mellon University, April 2000.

[18] .RaghvendraV.Cowlagi and Panagiotis Tsiotras "Multi-resolution Path Planning: Theoretical Analysis Efficient Implementation, and Extensions to Dynamic Environments" School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332.

[19] HuoshengHu, Michael Brady, Penelope Probert "Navigation and Control of a Mobile Robot among Moving Obstacles" CH3076-7/91/0000-0698$01 .OO 0 1991 IEEE.

[20] Leela Rani.P, Rajalakshmi.P "Clustering Gene Expression Data using Quad Tree based Expectation Maximization Approach" International Journal of Applied Information Systems (IJAIS) – ISSN: 2249-0868, Foundation of Computer Science FCS, New York, USA, Volume 2– No.2, June 2012 – www.ijais.org.

[21] N. Shobha Rani, "ArunGopi "A Quad Tree Based Binarization Approach to Improve quality of Degraded Document Images" International Journal of Computer Science Engineering (IJCSE), ISSN: 2319-7323, Vol. 3 No.01 Jan 2014 1.

[22] 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.

[23] M. Spann and R. Wilson "A quad tree approach to image segmentation which combines

statistically and spatial information" (Received 21 August 1984; in revised form 30 October 1984; received for publication 8 November 1984). Pattern Recailmtion Vol. 18, No 3/4, pp. 257 -269, 1985. 0031 3203/85 $3.00+ .U0, Pergamon Press Ltd.,c1985 Pattern Recognitions society.

[24].KunioAizawa, KoyoMotomura, Shintarokimura, RyosukekaDowaki and Jai Fan "Constant Time Neighbor finding in Quad tree" ISCCSP2008,978-1-4244-1688--2/08.c2008IEEE,ISCCSP 2008 .Malta 12-14 march2008.

[25] ShahedShojaeipour, Sallehuddin Mohamed Haris, and Muhammad IhsanKhairir "Vision-Based Mobile Robot Navigation Using Image Processing and Cell Decomposition"H. BadiozeZaman et al. (Eds.): IVIC2009, LNCS 5857, pp. 90–96, 2009. © Springer-Verlag Berlin Heidelberg 2009.

[26] Campbell, J.Sukthankar,R,Nourbakhsh,I.Pahwa, A: A robust visual odometry and precipice detection system using consumer-grade monocular vision. In: Proc. ICRA 2005, Barcelona, Spain (2005).

[27] Bbatsaikhan Dugarjav Hhyungkim , Soon Geul Lee "Online cell decomposition with a laser range finder for coverage path in an unknown work space" Proceedings of Fourteenth The IIER International Conference, Paris, France, 8th March 2015, ISBN: 978-93-82702-72-6.

[28] Abraham Sanchez ,Rane Zapata and Maaria A. Osorio Lama Sampling based Motion planning : A survey ,Computation y systems Vol 12 No.1 ,@\2008,pp5-24 ISSN 1405-5546.

[29] D. Ferguson and A.Stentz, "Field D*: An interpolation-based path planner and replanner," in Proceedings of the International Symposium on Robotics Research (ISRR), October 2005.

[30] S. Choi, J. Y. Lee, and W. Yu, "Fast any-angle path planning on grid maps with non-collision pruning," in IEEE International Conference on Robotics and Biomimetics, Tianjin, China, December 2010, pp. 1051–1056.

[31] P. Yap, "Grid-based path-finding," in Advances in Artificial Intelligence,ser. Lecture Notes in Computer Science, vol. 2338. Springer Berlin /Heidelberg, 2002, pp. 44–55.

[32]. I.Millington and J.Funge, Artificial Intelligence for Games, 2nd Morgan Kaufmann Publishers, 2009.

[33] G.Ayorkor, A.Stentz, and M. B. Dias, "Continuous-field path planning with constrained path-dependent state variables," in ICRA 2008 Workshop on Path Planning on Costmaps, May 2008.

[34] C.E.Thorpe and L. H.Matthies, "Path relaxation: Path planning for a mobile robot," in OCEANS Conference, September 1984, pp. 576–581.

[35] M.Kanehara,S.Kagami,J.Kuffner, S. Thompson, and H. Mizoguhi,"Path shortening and smoothing of grid-based path planning with consideration of obstacles," in IEEE International Conference on Systems,Man and Cybernetics, ISIC., October 2007, pp. 991–996.

[36] A. Botea, M. Muller, and J. Schaeffer, "Near optimal hierarchical pathfinding,"Journal of Game Development, vol. 1, pp. 1–22, 2004.

[37] A. Nash, K. Daniel, S. Koenig, and A. Felner, "Theta*: Any-angle path planning on grids,"In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2007, pp. 1177–1183.

[38] 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.

[39] Kunio Aizawa,KoyoMotomura,Shintro Krimura ,Ryosuke kadowaki and jia Fan"Constant Time Neighbor Finding In Quad Tree"978-1-4244-1688-2/08/ c @ 2008 ,IEEE.

[40] Subbarao kambhampati and Larry S.Davis "Multiresolution path planning for mobile robot.Journal of robotics and automation vol,RA-2,No-3,Septmber.1996.

[41] Dave Ferguson and Anthny Stentz." The field D*Algoritum for improved path planning and replaning in uniform and non uniform cost environment".Technical report CMu -tr-ri-05-19,chrnegiemellonUniversity, [dif,tony}@cmu.edu.

[42] Khalid Al-Mutib, Mansour AlSulaiman, Muhammad Emaduddin, Hedjar Ramdane Ebrahim Mattar, ''D* Lite Based Real-Time Multi-Agent Path Planning in Dynamic Environments'' International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 2,Mar-Apr 2012, pp.1414-1419.