An Overview of Autonomous Mobile Robot Path Planning Algorithms

N. Sariff [1] and N. Buniyamin [2]
[1,2] Faculty of Electrical Engineering
Universiti Teknologi MARA
40450 Shah Alam, Selangor, Malaysia
[1] nohaiddasariff@yahoo.com
[2] nbuniyamin@salam.uitm.edu.my

Abstract - **Determination of a collision free path for a robot between start and goal positions through obstacles cluttered in a workspace is central to the design of an autonomous robot path planning. This paper presents an overview of autonomous mobile robot path planning focusing on algorithms that produce an optimal path for a robot to navigate in an environment. To complete the navigation task, the algorithms will read the map of the environment or workspace and subsequently attempts to create free paths for the robot to traverse in the workspace without colliding with objects and obstacles. Appropriate or correct and suitable algorithms will fulfill its function fast enough, that is, to find an optimal path for the robot to traverse in, even if there are a large number of obstacles cluttered in a complex environment. To achieve this, various approaches in the design of algorithms used to develop an ideal path planning system for autonomous mobile robots have been proposed by many researchers.**

Simulation and experimental results from previous research shows that algorithms play an important role to produce an optimal path (short, smooth and robust) for autonomous robot navigation and simultaneously it prove that appropriate algorithms can run fast enough to be used practically without time-consuming problem. This paper presents an overview and discusses the strength and weakness of path planning algorithms developed and used by previous and current researchers.

I. INTRODUCTION

Path planning or find-path problem is well known in robotics and it plays an important role in the navigation of autonomous mobile robots (artificial intelligent mobile robots). Navigation which is a process or activity to plan and direct a route or path [21] is a task that an autonomous robot must do correctly in order to move safely from one location to another location without getting lost or colliding with other objects. The three general problems of navigation are localization, path planning and motion control. Between these three problems, it can be argued that path planning is one of the most important issues in the navigation process. Path planning enables the selection and identification of a suitable path for the robot to traverse in the workspace area.

Path planning research of autonomous mobile robot has attracted attention since the 1970s. Over the past several years and recently, research in this area has increased due to the reason that mobile robots are now applied in various applications. Thus these robots now have to operate in presence of uncertainty in various domains.

Accurate path planning enables autonomous mobile robots to follow or track an optimal collision free path from start position to the goal position without colliding with obstacles in the workspace area. An ideal path planner must be able to handle uncertainties in the sensed world model, to minimize the impact of objects to the robot and to find the optimum path in minimum time especially if the path is to be negotiated regularly.

Robot representation of the world in configuration space (C-space) and the algorithm implementation is the two main components for global or deliberative path planning and these two components are interrelated and greatly influence one another in the process to determine an optimal path for the robot to traverse in the workspace within an optimal time.

Path planning algorithms are usually based on configuration space representations such as the Voronoi diagram [1], regular grids/occupancy grid, generalized cones [2], quad-tree [3] and vertex graph, where the C-space is full with data structures that show the position and orientations of objects and robots in the workspace area including the free space regions and forbidden regions with obstacles or mazes.

In order to simplify the path planning problem and ensure that the robot run/move smoothly while avoiding obstacles in a cluttered environment, the configuration space must be matched with the algorithm being used. Hence the selection of algorithm for a given problem is an important issue. In computing, data structures play an important role and greatly influence the computational complexity and efficient implementations of algorithm [4].

Previous researchers for example, Donald, Brooks and Kambhampti [1,2,3] attempted to find an optimal path (shorter, smoother) with optimal time by combining a suitable approach on presenting or mapping the workspaces area with an efficient representation of free space and obstacles region and simultaneously implement an appropriate algorithm (robust, thoroughness) to search and manipulate those data structures.

There are numerous types of algorithms to search and manipulate the data structure used to store the maps of the environment/work space area. Examples of these algorithms are, a graph search algorithm that consists of best first search

algorithm which traverse a graph using a priority queue to find a shortest collision free path (A* search algorithm [5], modified A* search algorithm [6,7], genetic algorithm as an optimization path algorithm, potential field algorithm [8] and roadmap algorithm [9]. Previous research shows that different algorithm type may complete the same task (i.e. with a different set of instructions) in less or more time, space, effort etc. Each type of algorithm has its own strength and weakness which is compared and discussed in detail in Section III.

II. PATH PLANNING ALGORITHMS

Numerous algorithms have been developed over the last few years to create real time path planning system for autonomous robots. There are three things or activities that must be followed or carried out by an autonomous robotic system to enable the execution of the task of robot navigation. These activities are mapping and modeling the environment, path planning and driving systems. The selection of an appropriate algorithm in every stage of the path planning process is very important to ensure that the navigation process will run smoothly.

For example, in his path planner system, Alexopoulos [19] combined a visibility graph (VGraph) algorithm (to map and model the environment) with the Dijikstra's algorithm to select an optimal path for the autonomous mobile robot to traverse in the environment. Both algorithms have its own function to complete the robot navigation process. VGraph algorithm is an algorithm where the path is made up of straight line segments, which are connected by a subset of vertices (V) of obstacles. A constructed Vgraph normally have a starting position (s), the goal position (t), the vertex set (V), and the edge set (E) that contains an edge for each pair of visible vertices. A shortest path from s to t is then determined by Dijikstra's method.

In summary, a generic path planning algorithm needs to meet several requirement as listed below [20]:

1) "The resulting paths should have the lowest possible cost to prevent any indirection" [20].
2) "It should be fast and correct to not thwart the simulation process, for example it should be robust if there is no collisions occurred during the navigation process" [20].
3) "The algorithm should be generic with respect to different maps where its means that it should not be fully optimized for a specific map type" [20].

3.1 Traditional A search algorithm*

The standard search algorithm for the shortest path problem in a graph is A* algorithm. The A* algorithm as shown in Equation 1 below can be considered as the best first search (BFS) algorithm that combines the advantages of uniform-cost and greedy searches using a fitness function [7].

$$f(n) = g(n) + h(n) \quad (1)$$

The term g(n) denotes the accumulated cost from the start node to node n and h(n) is a heuristic estimation of the remaining cost to get from node n to the goal node.

During the search, the A* algorithm maintains two lists of nodes: The open list contains the nodes that have to be considered next and the closed list contains the nodes already visited. The algorithm itself consists of expanding the one node from the open list whose fitness function is minimal. Expanding a node means putting it into the closed list and inserts the neighbors into the open list and evaluating the fitness function. The algorithm stops when the goal of node is expand [7].

The choice of a good heuristic algorithm is necessary in order to achieve both quality and efficiency of a search. As long as the heuristic under estimates the real cost, the shortest path is guaranteed found. Nevertheless, under estimating can easily lead to an expansion of too many nodes. When the heuristic is allowed to over estimate the distance to the goal, the A* algorithm tends to expand nodes that lie on the direct path to the goal before trying other nodes. However, this can also lead to significantly slower searches if the final path contains directions that lead away from the goal.

Each algorithm have its own strength and weakness and for the A* algorithm, although it has been widely used because it finds the minimum cost path and it will determine the existence of free path, this conventional/ classical algorithm have a time consuming problem because it does not have a heuristic information to handle until it reach the goal. This will cause a lot of node being produced to find a short path and to avoid obstacles which will eventually make it run slower.

3.2 Modification of A algorithm*

Although A* algorithm is the most effective free space searching algorithms in term of path length optimization and thoroughness, it still has a tendency to be a slower planner due to amount of space that it searched. It means that it will spends much of the running time to complete the whole task if the number nodes to be search is large and simultaneously the time consuming problem will occurred in the robot systems. However, by increasing the speed of the algorithm by a previous research, its usefulness is enhanced.

Warren [7] found the way to increase the speed of this algorithm to make an ideal planner for real time path planning. The alternative proposed by Warren is to modify the conventional A* algorithm method to perform a loose search on a fine grid. It's done by using a trial vector that can cross several cells which help to produce a fast planning. Fan [6] also proposed another alternative to modify the A* algorithm to become a h' function that is incorporated in the Equation 1. Using this approach, the time taken to find the best path is far less than the time that the exhaustive depth first search needs to find the best path [6]. Moreover, the best path found by this searching algorithm is always as good as the best path found by the depth search. In other words, it's always finding a good path which is near optimal very quickly.

Hybrid planner has been developed by adding the beneficial features of A* method (produced a thorough and robust path planning) together with the vector based approach (increase the search algorithm speed) [7]. Solution from this modified A* method is that the path are slightly sub-optimal and the speed of the searching algorithm is increased dramatically compared to the traditional A* algorithm. Refer to Section four a detailed description.

III. PREVIOUS RESEARCH AND RELATED WORKS

Path planning has been studied extensively over the past twenty years and numerous path planning algorithms have been developed. This paper presents an overview of path planning algorithms that include algorithms to construct an approximation representation of free space regions and algorithms to search the data heuristically in a mapped environment.

An investigation of path planning algorithms covering the period 1980 to 1989 indicates that the best-first search algorithms (A* search algorithm) will run on numerous type of configuration space to determine an optimal path. The configuration space approach originates in the work of Lozano-Perez [10] and underlies most path planning research. The most common cases are, planar polygonal robot with polygonal obstacles and a polyhedral robot with polyhedral obstacles. Brooks's planner [5] decomposes the configuration space into a graph of free, blocked, and mixed cuboids. Then it performs A* search for a piecewise linear path through the free nodes. If none is found, heuristics are used to select mixed nodes and to split them into smaller cuboids, the graph is updated, and the search is repeated.

Takahashi and Schilling [11] plan for a rectangular robot with polygonal obstacles via heuristic search of the generalized Voronoi diagram. The Voronoi diagram [1] is the free space data structure that is constructed by selecting the stop points of the spatial graph. This diagram is strong deformations retract of free space so that free space can be continuously deformed into the diagram. This means that the diagram is complete for path planning where searching the original space for paths can be reduced to a search on the diagram. Reducing the dimension of the set to be search usually reduced the time complexity of the search. One of the advantages of this diagram is that it lead to robust paths where paths are maximally clear of obstacles. Donald [12] performs A* search in the configuration space of a moving polyhedron with static polyhedral obstacles. He develops parametric expressions for contact patches and their intersections, but does not compute the free space topology. His planner moves through free space and from patch to patch via heuristic motions, such as sliding along an obstacle.

Table 1 below shows the strength and weaknesses based on simulation result of each of the approaches proposed by the previous researchers.

TABLE 1: STRENGTH AND WEAKNESS OF ALGORITHMS

Workspace	Type of Algorithm	Strength & Weakness
Polyhedral object moving to avoid polyhedral objects in a static environment [13]	Visibility graph algorithm + A* search algorithm	**Strength:** • Visibility graph will find collision free path while A* search algorithm will find optimum path and allow use of efficient heuristic information **Weakness:** • Not necessary for dynamic obstacles that always changes
Polygon moving to avoid polygon obstacles in the environment [5]	Generalized cones + A*search algorithm	**Strength:** • Run quite fast in uncluttered environment and find a good path. **Weakness:** • Does not find possible path in extremely cluttered situation but can be used to provide direction.
Static environment cluttered with obstacles [3]	Quadtree representation + A* search algorithm	**Strength:** • Dynamic path planning. **Weakness:** • Does not localize the effect of obstacles on the representation, Algorithms are time consuming because A strategy expands a lot of nodes.
Static environment cluttered with obstacles [14]	Feasible algorithm run it path graph on quad-tree	**Strength:** • Algorithm builds its path graph small enough to determine short

		collision free path. The smaller the path graph, the faster our algorithms is. • Proposed algorithm run in path graph faster compare to visibility graph algorithm. **Weakness:** • Does not localize the effect of obstacles on the representation
Complex environment cluttered with obstacles [8]	Artificial potential field algorithm	**Strength:** • Help in robot navigation **Weakness:** • Suffer in local minima problem especially in an environment cluttered with obstacles

Path planning algorithms proposed in the period between 1990 and 1998 consists of probabilistic algorithms such as artificial potential field algorithm and road map algorithms, modified A* search algorithm, and genetic algorithms. Extensions of artificial potential fields approach was originally proposed by O.Khatib [8]. A unique real time obstacle avoidance approach using this artificial potential fields concept have been used to allow real time robots operations in a complex environment. The algorithm uses repulsive potential fields around the obstacles to force away the robot that subject to this potential and use an attractive potential field around the goal to attract the robot to go through the goal.

Another potential field algorithm begins with a trial path through space and uses repulsive potential fields around obstacles to modify the path so that it moves towards the free space until a safe path is found. Most of the potential functions approach will help in the robot navigation however local minima problem remains. Multiple options have been taken by previous researchers to minimize and eliminates the local minima problem, for example, Hussein [15]. Road map algorithms [9] generate many random configurations, prune the blocked ones, and link adjacent free ones into a graph (normally with line segments). Path planning is performed by linking the initial and goal configurations to the graph then searching for a path between them. Table 2 below shows the strength of the modified A* search algorithm compared to the traditional A* algorithm and genetic algorithm in determining autonomous mobile robot path planning:

TABLE 2: ALGORITHMS STRENGTH

Workspace	Type of Algorithm	Strength
Convex polygon moving and rotate in free space to avoid obstacles [6]	Modified A* algorithm based on the revised depth-first search (Depth-first search: traverses a graph branch by branch) in setting up the h' value is proposed to find an optimal path.	**Strength**: • Time takes to find a best path is far less than depth first search. It finds a good path quickly.
Static environment with obstacles cluttered [7]	Modified A* search algorithm by using a trial vectors that span several cells to help in the planning	**Strength:** • Speed of searching algorithm is increased dramatically compare to traditional A* search algorithm
Time varying and unknown environment	Genetic algorithm	**Strength:** • Works for complex problem that traditional algorithms cannot find a satisfactory solution. • Can work in time varying and unknown environment.

Algorithms proposed by previous researchers in the period between 2000 and 2005 include, a modified A* search algorithm known as Physical A* algorithm, Stentz' Focussed Dynamic A*algorithm and depth first search algorithm. Felner's Physical A* algorithm (PHA) [16] solves the problem of finding the shortest path between two points in unknown real physical environment by exploring unknown territories. The difference between Physical A* and Classical A* search algorithms is that the complexity of the algorithm is not based on the number of generated nodes but it based and measured by traveling moving agents.

Koenig's algorithm [17], termed Stentz' Focussed Dynamic A is a heuristic search method that repeatedly determines a shortest path from the current robot coordinates to the goal coordinates while the robot moves along the path. It is able to re-plan faster than planning from scratch since it modifies its previous search results

locally. Consequently, it has been extensively used in mobile robotics.

van den Berg [18] uses roadmap where an approximately time-optimal trajectory where a start to a goal configuration is computed, such that the robot does not collide with any moving obstacle. The trajectory is found by performing a two-level search for a shortest path. On the local level, trajectories on single edges of the roadmap are found using a depth-first search on an implicit grid in state-time space. On the global level, these local trajectories are coordinated using an A -search to find a global trajectory to the goal configuration.

In the attempt to highlight the difficulty in finding an optimal path for autonomous robots, a discussion on the basic concept of robot navigation focusing on path planning algorithm is discussed below. In the later sections, this paper also presents a few example of experiment and analysis result of algorithm implementations from previous research that attempted to determine an optimal path for autonomous robot to navigate in an environment.

IV. EXPERIMENTAL RESULTS FROM A PREVIOUS RESEARCH

4.1 Comparison of the proposed algorithm and the algorithm based on the VGRAPH in respect of calculating time

As most algorithms are time consuming, a group of researchers from Osaka University in Japan [14] search for alternatives and developed their own algorithms and subsequently compared the efficiency of operations (with respect to calculation time) when using these algorithms combined with 1) visibility graph algorithm or 2) path graph algorithm.

They proposed feasible algorithms that run in path graph on the quadtree representation. The specialty of this algorithm is that it can select the collision free shortest path even in the workspace in which the location of obstacle is dynamically altered. In addition, the proposed algorithm does not search directly in the quadtree but in a path graph. It selects the shortest path out of the path graph and synchronously modifies the path graph such that the selected path is to be collision free based on the geographical information in the quadtree. In comparison with the quadtree and the visibility graph, the number of nodes in the path graph is kept small by this synchronism and consequently the proposed algorithms runs fast enough to be used practically. Table 3 below shows that the proposed algorithm is superior in time calculation when it is run in path graph compared to when a conventional Algorithm A* is run in VGraph.

TABLE 3: COMPARISON OF CALCULATION TIME BETWEEN THE PROPOSED ALGORITHM USING PATH GRAPH AND A* ALGORITHM USING VGRAPH.

	Positions		**Time (sec)**	
	Start	Goal	Path graph	Vgraph (54
			(level 4)	nodes)
1	(32,250)	(450,250)	0.033	0.188
2	(270,32)	(450,200)	0.025	0.183

4.2 Resulting time produced by modified A technique for searching the free path without colliding with the obstacles cluttered in the workspaces*

The A* technique has long been used for planning purposes, but it can be extremely slow when there are a large number of cells to be searched. Because of this problem, Warren [7] proposed an alternative which is to modify the A* search algorithm by using a trial vector that span several cells to aid in the planning to increase the speed of the searching algorithm.

Figures 1, 2 and 3 from Warren [7] depicts examples that prove the time to calculate the optimal path using modified A*search algorithm is reduced dramatically compare to the A* search traditional algorithm even if the workspace environment changes extensively and the workspace is extremely cluttered with obstacles or mazes.

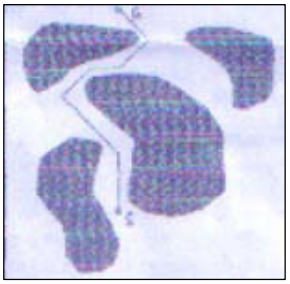

Fig. 1: Example of first workspace

Result: The algorithm solved the planning problem less than 0.055 seconds

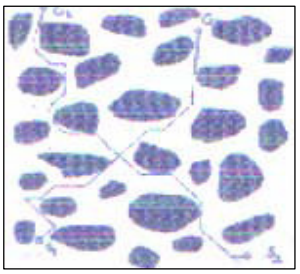

Fig. 2: Example of second workspace

Result: The algorithm solved the planning problem in 30 miliseconds

Fig. 3: Example of third workspace

Result: The algorithm solved the planning problem in 1/3 seconds

V. CONCLUSIONS

An overview of path planning algorithms for autonomous robot, the strengths and weakness of these algorithms were presented and discussed. The basic concept of robot navigation, relationship between mapping, path planning and driving system was also discussed in this paper which highlighted the problems of path planning and the numerous approaches taken by past researchers to solve problems posed by currently available algorithms.

VI. REFERENCES

[1] B.R Donald, "Motion Planning With Six Degrees of Freedom,"Massachusetts Institute Technology Artificial Intelligence Laboratory, Technical report AIM-791, 1984.

[2] Rodney A.Brooks, "Solving the find path problem by representing free space as generalized cones" A.I Memo No 674, Massachusetts Institute Technology, May 1982.

[3]Subbarao Kambhampati and Larry S. Davis, "Multiresolution path planning for mobile robots," in *International Journal of Robotics Research*, 5(10):90-98, Spring 1986.

[4] Http//:www.en.**wikipedia**.org/wiki/Dictionary

[5] R.A Brooks and T.Lorenzo-Perez. "A subdivision algorithm inconfiguration space for findpath with rotation," in *IEEE Transactions on Systems,Man and Cybernetics*, Mar./Apr.1985, SMC-15(2), pp 225-233.

[6] Kuo-Chin Fan and Po-Chang Lui, "Solving the find-path problem in mapped environments using modified A* search algorithm," in *IEEE Transactions on Systems, Man, and Cybernetics,* vol.24, no.9, September 1994, pp 1390-1396.

[7] Charles W.Warren, "Fast path planning method using modified A* method," in . *IEEE International Conference on Robotics and Automation, pp* 662-667.

[8] O.Khatib, "Real time obstacle avoidance for manipulators and mobile robots," in *International Journal of Robotics Research*, 5(10):90-98, Spring 1986.

[9] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, 1996.

[10] T. Lozano-Perez, "Spatial planning: A configuration space approach," in *IEEE Transactions on Computers*. 1983, vol. C-32, pp. 108–120.

[11] O. Takahashi and J. Schilling, "Motion planning in a plane using generalized Voronoi diagrams," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 2, pp. 143–150, 1989.

[12] Bruce R. Donald, "A search algorithm for motion planning with six degrees of freedom," *Artificial Intelligence*, vol. 31, no. 3, pp. 295–353, 1987.

[13] T.Lozano-Perez and M.A Wesly. "An algorithm for planning collision-free paths among polyhedral obstacles," *in Communication of the ACM,22*(10), Oct 1979, pp 560-570.

[14] Hiroshi Noborio, Tomohide Naniwa and Suguru Arimoto, "A fast path planning algorithm by synchronizing modification and search of its path graph," in *Internationalworkshop on Artificial intelligence for Industrial Applications 1988*, pp 351-357.

[15] A.M Hussein and A.Elnagar, "A fast path planning algorithm for robot navigation with limited visibility" in *IEEE Proceedings2003,*pp 373-376.

[16]Ariel Felner, Roni Stern, Sarit Kraus," PHA*: Performing A* in Unknown Physical Environments," in *AAMAS* , July 2002, pp 15-19.

[17] Sven Koenig and Maxim Likhachev, "Fast Replanning for Navigation in Unknown Terrain," in IEEE, *Transactions on Systems, Man, and Cybernetics* Vol. 17, No. 4, June 2005, pp 354-363.

[18] Jur P. van den Berg and Mark H. Overmars," Roadmap-Based Motion Planning in Dynamic Environments," in *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 5, October 2005, 885-897.

[19]Christos Alexopoulos and Paul M.Griffin, "Path planning for mobile robot," in *IEEE Transactions on Systems, Man, and Cybernetics,* Vol 22,No 2,Mar/April.1992, pp 318-322.

[20]Christoph Niederberger,ejan Radovie, and Markus Gross, "Generic path planning for real time application," in *Proceedings of the Computer Graphics International (CGI 04),* 2004, pp 1-8.

[21] Pearsall, J., ed. *Concise Oxford Dictionary*. Tenth Edition, Revised ed. 2001, Oxford University Press.