Advances in Robotics & Automation
ISSN: 2168-9695

Kamkarian and Hexmoor, Adv Robot Autom 2015, 4:2
DOI: 10.4172/2168-9695.1000131

Research Article Open Access

A Robotic Path Planner Contender

Kamkarian P[1]* and Hexmoor H[2]

[1]Department of Electrical and Computer Engineering, Southern Illinois University, Carbondale, USA
[2]Department of Computer Science, Southern Illinois University, Carbondale, USA

Abstract

This article presents a novel offline path planner method to yield a collision-free trajectory among groups of obstacles in a static workspace. It enables a single holonomic point robot acting in a static environment including a fixed initial and goal configurations to achieve its goal toward a collision-free trajectory. In developing our novel path planner, we focused to elevate features that help the planner to route in a wide variety of different situations in regards to lowering the processing time needed for analyzing the workspace and determination of ultimate trajectory. Unlike to some other planners that are able to be applied on some certain obstacle shales such as polygonal, our planner is skillful enough to analyze any types of obstacles, such as circular, spiral, and curved edged obstacles successfully. To increase the performance of our proposed offline path planner, we defined and benefitted from introduction of parameters that help to achieve the best possible results among different scenarios in workspace components arrangements as well as reducing the planner needing to access system resources such as memory or processing unit to analyze the workspace while computing the shortest collision-free path from start point to the goal configuration. The novel planner analyses and transforms the two dimensional representation of the environment into a roadmap consisting a graph of nodes along with all possible routes from the moving robot's initial point into the goal configuration. In order to manage some of the popular problems such as being trapped inside a U-shaped obstacle or routing in narrow pathways among obstacles, that challenge other offline path planners such as Potential Field planner are facing, we used a multi-layer approach in form of different stages to help the planner considering all possible circumstances and hence, compute the best possible route.

Keywords: Path planning; Collision-free trajectory; Rapidly optimizing mapper; Trajectory builder

Introduction

Robotic science is recognized as a powerful tool that the mankind has developed in modern societies to facilitate performing tasks with higher quality and better accuracy rates. Robots are able to perform repetitive tasks tirelessly and without the loss their rate of accuracy because of factors that can cause inaccuracy symptoms contrasted with human performed tasks. Based on constructional and the type of operating tasks, robots are divided into many categories. As a popular category, we cite mobile robots. For a mobile robot, a major activity that has a vital role on operating assigned tasks successfully is the ability to dependably move from the initial to the goal configuration in the workspace. There are many factors involved with the rate of performing a successful movement among obstacles in workspace. For instance, a mobile robot has to have a proper planner to analyze the environment and to build a flawless trajectory in term of safety. The term safety generally refers to an obstacle collision-free path, which has enough distances from every obstacle as well as the workspace boundaries in regards to the accuracy of its sensors and physical actuators to be able to maneuver among obstacles toward goal appropriately. In order to construct a proper path planner, researchers have employed a variety of different methodologies adopted from a wide range of distinctive venues of science. In most cases, developing an ultimate path planner is a long process that is subject to be revised and advanced by different groups of researchers in different research articles. In other terms, when a purely novel path planner has been proposed, its advantages and disadvantages are addressed from the evaluation of applying it on different situations by further research. Our planner is termed rapidly optimizing mapper (ROM). The original contributions of ROM are its efficiency of generating trajectories [1] as well its parametric adjustability [2]. The generated paths are as short as any other planner's output. Our planner can be parametrically adjusted to deal with desired safety and navigation criteria. The addressed issues will be then covered by optimizing or updating the construction of the planner by further research. In the following, we demonstrate a few scenarios of proposed novel path planners along with the processes that further matured them by subsequent research.

Potential Field planner is categorized as a classic and solid path planning method suggested for offline robots. It is constructed based on considering virtual attractive and repulsive forces among components in workspace such as obstacles, start and goal locations along with the workspace boundaries. The fashion that Potential Field algorithm follows is to build trajectory based on deliberating on an attractive influence for the goal configuration as well as repulsive forces from obstacles. In other terms, the goal point has the strongest attraction among other components in workspace which interests robot the most while obstacles based on their shapes, sizes, and locations, have repulsive effects on the robot. Using this strategy helps planner to calculate locus of point in forms of a collision-free trajectory from initial to the goal configuration. The idea of adopting Potential Field for offline robot path planning has promoted by Khatib [3] for the first time. Although adopting the primitive Potential Field method to build trajectories for offline robots has its own advantages, but due to its constraints, it is unable to compute the ultimate trajectory in some certain scenarios. As instances of such conditions we can notice the local minima thatarises

***Corresponding author:** Kamkarian P, Department of Electrical and Computer Engineering, Southern Illinois University, Carbondale, USA, Tel: +1618-453-2121; E-mail: pejman@siu.edu

Received July 15, 2015; **Accepted** August 04, 2015; **Published** August 06, 2015

Citation: Kamkarian P, Hexmoor H (2015) A Robotic Path Planner Contender. Adv Robot Autom 4: 131. doi: 10.4172/2168-9695.1000131

when the robot is located inside a U-Shaped obstacle or is trapped in a closed obstacle. The mentioned cases are addresses by Borenstein and Koren, and Guldner et al. [4,5]. There are several method have proposed to fix the expressed problem by updating the potential field algorithm structure [6,7] as well as adopting other methods such as harmonic functions into the potential field planner [8-10].

As another example, we cite Voronoi path planner. The Voronoi path planner is a method to partition a workspace with the purpose of building a collision-free trajectory and can be applied on environments with both local [11,12] and global knowledge [13] specifications. Trajectory safety is one of the salient key features of employing Voronoi path planner. Considering partitions as the robot pathways crossing from the middle distances of obstacles leads obtaining the maximum path to be used by robot and hence, causing a decent space for robots with poorly detectors or maneuvers skills to reach their goals successfully. Using Voronoi path planner method however, has its own constraints. For instance, in case of building the trajectory in form of a piecewise linear path, the robot, in order to follow the path, has to continuously proceed with the stop and start process and hence spending more time, energy, and effort to reach the goal. Several solutions have been proposed to overcome and improve the mentioned issue such as reconstructing the trajectory by replacing the sharp corners with different types of curves, such as using Bezier curves [11,14-16] or benefiting the technique of connecting vertices with splines [17-19]. As a more advanced strategy to treat the problem of generating sharp corners through the Voronoi method path planning, Grefenstette and Schultz [20] suggested a programming algorithm that dynamically smoothness the piecewise linear trajectories from Voronoi planner.

As the third category of path planners, we consider the randomized path planners. The randomized path planner approaches are generally built based on constructing a set of randomly generated graph of available configurations with the purpose of assessing the workspace in terms of examining different conditions of pathways toward the goal configuration. As the examples of earlier randomized approaches, we can mention probabilistic roadmap method [2,21] and randomized potential field solution [22]. In the probabilistic roadmap path planner method a graph forms from a set of free points randomly distributed in the workspace area at the beginning. The planner then attempts to build the trajectory by considering connecting pairs of nearby configurations from initial to the goal. The Probabilistic Roadmap approach is strong enough to route the trajectory in steerable non-holonomic [23], or holonomic systems, with no problems. It is however, not suitable to be used to build the trajectory in an environment using non-linier controller robots or sophisticated non-holonomic environments. Employing the randomized Potential Field solution is also a challenging task when facing with choosing a proper heuristic potential function in workspaces with several obstacles as well as dynamic and kinematic differential constraints.

Confronted with different variety in workspace objects arrangements such as obstacles and moving robots along with the specifications of each along with different constraints that may cause reduction of the regular path planner's performances, led researchers to consider developing more advanced path planners in form of hybrid path planners. Employing hybrid path planners [24-30] as promising approaches resulted in achievement of better and more reliable trajectories. A hybrid path planner typically benefits from a combination of more than one path planner inherits their advantages in a unique algorithm. Hybrid path planners have usually better performance than each individual path planner. This is because they are typically a collection of other path planners and hence possess more accuracy and possibility to generate the trajectory in a variety of different scenarios successfully.

This paper proposes a novel offline path planner; i.e., ROM, for single point robots. The planner is able to build a collision-free trajectory from initial to the goal configuration. The planner benefits from a multi-layer algorithm which consists of multiple methods with the sole purpose of computing the shortest and the safest possible path for the robot. Since the term safety will vary from a situation to another situation and depends on many factors such as the robot sensors accuracy and its ability to maneuver among obstacles, we considered using many variables that are adjustable based on both point robot physical specifications and the components of the workspace situations, such as the type and locations of obstacles. This paper aims to present our novel path planner in more details. In order to evaluate our planner performances, we will explore it by applying it on special scenarios that usually cause problems on the accuracy of building trajectories for other offline path planners. In addition, this research article has furnished results from applying our planner to more offline path planners. In the next section, we present the mentioned variables along with discussion of the usage and the range of each. The later sections are dedicated to the illustration of the path planner followed by the experimental results from applying it to variety of different workspaces. In order to validate the performance of our novel path planner, in the later section, we have compared it to a few other offline path planners. More detailed reports of these comparisons appear in recent reports [1,31]. Our offline planner can be used by an online planner by repeated applications of our algorithm. However, an online planner is not concerned with the entire route but rather it is seeking best decisions as it navigates. An online planner could be guided by an offline trajectory as generated by our method as a high level plan and can make minor local deviations as its navigation criteria dictates. The latter deviations are similar to local plans in contrast to overall trajectories that are global plans. Although our offline planner is inspired by online planners but it makes no claims about exceeding them.

Definitions and Key Parameters for ROM

In order to fabricate our planner, we considered and defined a few new concepts. These concepts are used in different parts of the planner algorithm construction. In this section we explain each concept in detail. The planner different phases and sections along with the usage of these concepts will be discussed in the next section. Our novel path planner consists of the following key concepts: standoff distance (SD), roadblock obstacle (RO), side edge node, degree of traverse (DT), degree of surface traversal (DST), node visibility, visible pathways, and isolated node. We introduce each of these concepts consequently:

Standoff distance (SD)

The standoff distance is a scalar value that will be adjusted based on the safe boundaries around obstacles that robot is able to maneuver without involving near misses. Moving robots, depending on their usage and the type of environment that they are built to react, are equipped with different types of sensors. The majority tasks for these sensors are to collect and analyze visual data from the robot surroundings. These data will be analyzed with the purpose of recognizing objects around the robot. Robot benefits the processed vision data to make proper decisions to adjust its path to avoid collisions. Moreover, robots, based on the moving equipment, have a variety maneuver skills. In other terms, when faced with an obstacle, robots need to maintain a proper

distance from the obstacle to adjust their path and hence, turn around the obstacle successfully. The standoff distance defines the safe width of the pathway around obstacles that robot adjusts its path when reaching it. Depending on the robot maneuvering skills, the SD value can be different. In the other terms, the predetermined standoff distance will be considered a lower value for a robot that has a quicker response and reaction to change its path. The standoff distance, however, needs to be considered with a higher value for robots that depending on their construction need wider safe channel around obstacles. Equation 1 is the mathematical relations for the mentioned concern.

$$\forall(r_1, r_2) \in R, if\ \Delta t_1 < \Delta t_2\ then\ SD_1 < SD_2 \quad (1)$$

In the equation 1, r_1 and r_2 are two arbitrary robots in the robot domain, R, and Δt_1, and Δt_2 indicate the reaction time needed for the first and the second robot respectively. SD_1 and SD_2 are also standoff distances for robot 1, and robot 2 respectively.

Roadblock obstacle

One of the main tasks for the planner to analyze the environment is to recognize roadblock obstacles. The planner benefits from the technique of considering virtual straight lines from the robot starting position to the goal configuration to build trajectory. Obstacles that have at least one collision point with the explained virtual straight line are added to the set of roadblock obstacles. The process of analyzing workspace and recognizing obstacles will be explained in details in the next section. Figure 1 shows a sample workspace along with the determined roadblock obstacles.

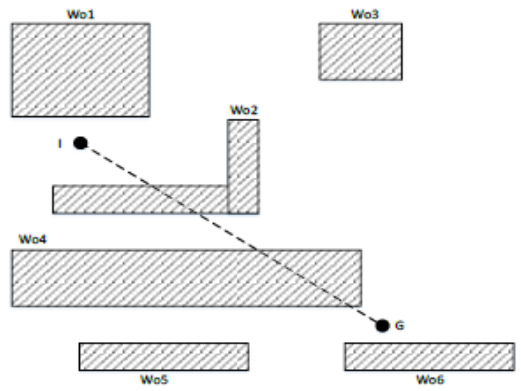

Figure 1: Showcase of roadblock obstacles in a sample workspace.

As it shown in the Figure 1, obstacles 2 and 4 are considered in the set of roadblock obstacles. This is because the virtual straight line from initial point, I, through the goal configuration, G, has intersected with the mentioned obstacles. Equation 2 illustrates the process of determining roadblock obstacles in math formula:

$$\forall w_{o_i} \in W_O, if\ (w_{o_i} \cap IG) \notin \phi\ then\ w_{o_i} \in RO \quad (2)$$

Where O_i represents the i^{th} obstacle and RO is the group of roadblock obstacles.

Side edge node

The side edge node is considered as either departure point for roadblock obstacles, or the location that the path builder is able to route a trajectory through the goal configuration. Depending on the workspace constraints as well as roadblock specifications, the location of considering side edge nodes will vary.

Figure 2 shows side edge nodes, n_1 through n_4 for obstacle, w_{o_1} are located in a sample workspace.

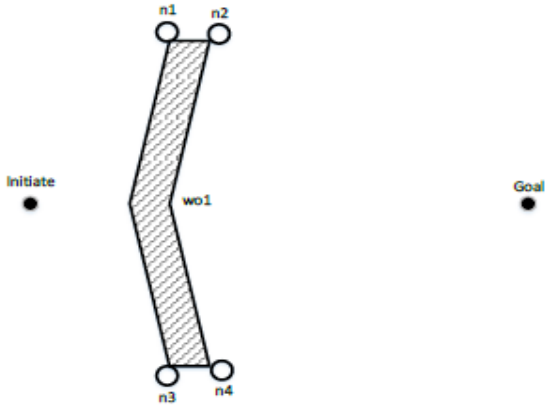

Figure 2: A sample showcase for side edge nodes.

Degree of Traverse (DT)

In order to build the proper collision avoidance trajectory, the path planner analyzes the surface of the roadblock obstacles. The Degree of Traverse is determined based on the sensitivity of the roadblock obstacle surface scanner. The DT value has a direct relation to the final trajectory smoothness. The smaller value for Degree of Traverse, the more smooth and accurate collision-free trajectory. Figure 3 demonstrates a sample DT value for a workspace.

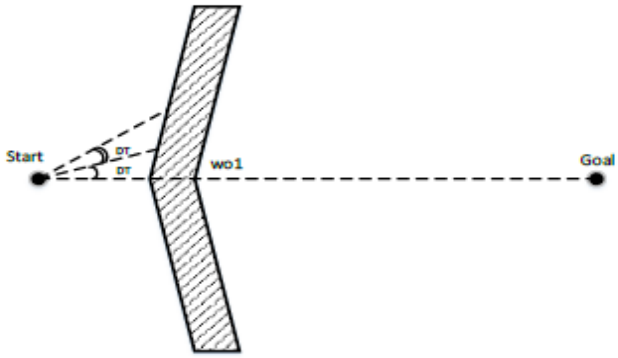

Figure 3: Illustrating DT value for a sample workspace.

Degree of surface traversal (DST)

In certain situations, to reach departure points, a robot needs to perform obstacle surface traverse. This is due to having limitations on either workspace constraints or obstacle specifications (Figure 4).

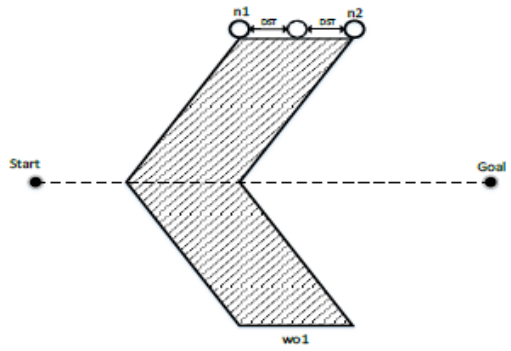

Figure 4: Surface traversing for a sample obstacle in a showcase workspace.

The Degree of Surface Traversal will be determined based on the accuracy of locus for the path crossing from the surface of obstacles. The DST value should be considered based on the robot skills on changing its direction and adjusting its path. A robot with a lower speed and reaction to change its trajectory will have a higher value of DST compared to the robot with a faster processing unit and path adjustment abilities.

Node visibility

Node visibility is a key concept that the planner uses to shorten the trajectory. This feature brings the planner the ability to achieve the best possible result in optimal collision avoidance path, in terms of the trajectory length from the initial to the goal configuration. For a group of nodes that are located on the same obstacle, pairs of nodes that share no intersection with any obstacles through straight rays are considered as visible nodes. In other terms, nodes that are located on the surface of obstacles and are able to see each other without any obstacles in between are categorized as visible nodes. Equation 3 commits the Node visibility concept to a mathematical formula.

$$\begin{aligned} &\forall(w_{o_i} \in W_O) \wedge (\mathrm{n}_j, \mathrm{n}_k \in w_{o_i}), \\ &if\left(\left(n_j \cap \cup_{v=1}^{l} w_{o_v} = \phi\right) \wedge \left(n_k \cap \cup_{v=1}^{l} w_{o_v} = \phi\right)\right) \\ &then(n_j, n_k \in VN) \end{aligned} \tag{3}$$

Where n_j and n_k are nodes located on the obstacle w_{o_i} and VN is the set of visible nodes.

Visible pathway

As one of the most important key features for the planner trajectory optimizer to determine the shortest possible paths toward goal, the Visibility pathways are defined based on the trajectories that connect both ends of a group of visible nodes. Figure 5 demonstrate the Visibility pathway concept.

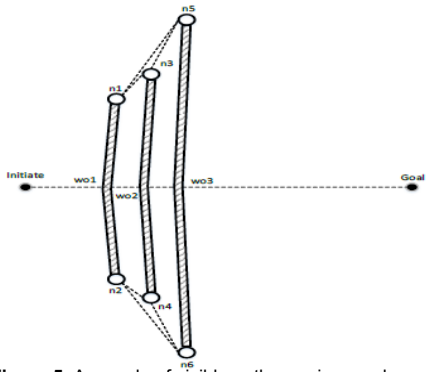

Figure 5: A sample of visible pathways in a workspace.

As it indicated in Figure 5, nodes 3 and 5 are visible from node 1. Nodes 4 and 6 are also visible from node 2. Therefore, connections from node 1 through node 5 and also node 2 through node 6 are considered as visible pathways.

In order to clarify the Visibility pathway concept, it will be illustrated through the following mathematical equation 4.

$$\begin{aligned} &\forall(n_i, n_j, n_k, \in N), if \\ &\left\{\begin{array}{l} \left(\left((\mathrm{n}_i\, n_j \cap \cup_{v=1}^{s} w_{o_v}\right) = \phi\right) \\ \wedge\left((n_j, n_k \cap \cup_{v=1}^{s} w_{o_v}) = \phi\right) \\ \wedge\left(((n_i, n_k \cap \cup_{v=1}^{s} w_{o_v}) = \phi\right) \end{array}\right\} \\ &then(n_i n_k \in VP) \end{aligned} \tag{4}$$

Where n_i, n_j *and* n_k are side edge nodes and VP is the set of visible pathways.

Isolated node

In order to find the optimal trajectory in terms of the shortest collision-free path, our planner considers all possible paths available around each roadblock obstacle. In order to achieve this goal, the planner analyzes each roadblock obstacle side edge nodes to assure there exists a pathway from both sides of roadblock obstacle which connects side edge nodes to one another. Any side edge nodes located on the same roadblock obstacle that is not connected to other nodes from either side of obstacle is considered as isolated node. As an important task for the side edge node analyzer unit of our path planner, it processes any isolated node with connecting them to one another, if possible.

Our algorithm enjoys a rapid computational running time. It examines obstacles once in finding relevant obstacles and then again in graph construction. Graph processing also uses the most computationally efficient algorithm. Therefore, this polynomial time algorithm runs faster than previous planners. Although it handles obstacles that are more complex than polygons, it suffer from computational drawbacks.

The Rom Path Planner Process

Through this section, we introduce our novel path planner construction in detail. The process of optimal trajectory calculation consists of four different and distinct phases as follows: Initial phase, Workspace analyzer, Graph (Roadmap) builder, and shortest path computation unit. The optimal trajectory refers to the shortest collision-free path from start to the goal configuration. Each phase, along with its related sections and steps will be discussed next.

Initial phase

At the beginning of the planning operation, the path planner uses parametric values that are adjusted based on the workspace and robot specifications. The planner uses the locations of the start point and goal configuration, along with the Standoff Distance, Degree of Travers and Degree of Surface Traversal values at the initial phase. These values guide the planner to route the best possible collision avoidance trajectory from start to the goal configuration.

Workspace analyser

This unit of the planner performs two major tasks. Those are analyzing the workspace to determine roadblock obstacles, and roadblock obstacle surface scanning with the purpose of calculating

side edge nodes. In order to determine roadblock obstacles, the planner considers a virtual ray in form of a straight line from the initial point into the goal configuration. The initial point will be considered as the start point at the beginning of the workspace analysis. The initial point will be however, considered as the either latest side edge nodes, a hit point, or a leave (i.e., departure) points. The process of determining roadblock obstacles along with determining side edge nodes continues until the planner reaches the goal configuration.

The second step of workspace analyzer unit consists of computing side edge nodes. In order to achieve this goal, the planner starts analyzing the surface of each roadblock obstacles, starting from both sides of each hit point resulting from intersecting the straight ray from the initial point toward goal configuration, for the size of DT. The process of scanningroadblock surface continues until either of the following criteria fulfilled.

a. The analyzer reaches to the leave point;

b. The virtual straight ray intersects any other obstacles.

Graph (Roadmap) builder: In order to find the optimal trajectory, the planner forms a lattice including side edge nodes along with hit and leave points obtained in the workspace analyzer unit. The following steps will be performed to build the graph:

a. All side edge nodes along with the start and goal configurations as well as any hit and leave points belonging to the same roadblock obstacles are joined together;

b. The graph (roadmap) builder removes isolated nodes by connecting them together, if possible;

c. Visible nodes belonging to the same roadblock obstacles will be recognized and visible pathways form based on them;

d. In order to achieve shorter pathways, all nodes located in between of each visible pathways will be removed;

e. Each edge participating to form the graph will be labeled based on Euclidean distances.

Figure 6 is the resultant of applying our path planner on the figure 5 along with the optimal path calculations in terms of the shortest length as well as determining a near miss avoidance trajectory, through the visibility pathways simplifications.

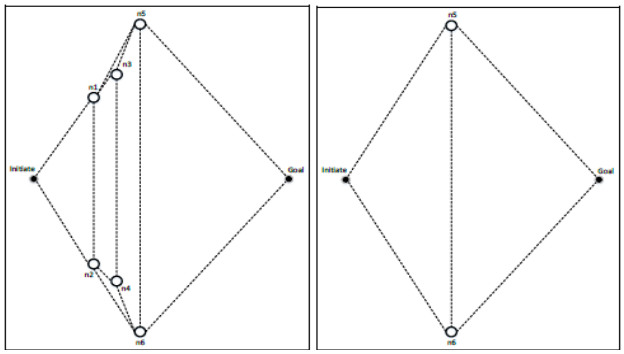

Figure 6: Left: the primitive graph consisting all possible pathways; Right: the optimal graph resulting from removing visible nodes from the graph.

As it shown in the figure 6, (left), the graph (roadmap) builder considers all available paths to form a complete lattice of all possible paths from each point. The graph (roadmap) builder unit will then simplify the graph by removing any visible nodes located among two sides of each visible pathway, as it illustrated in the figure 6, (right).

Shortest path computation unit: The main task of this phase of the planner is to analyze the graph constructed in the graph (roadmap) builder unit with the intention of refining the optimal trajectory from start to the goal configuration. In order to achieve the best possible result, we benefitted using the Dijkstra shortest path algorithm. Having start and goal nodes along with Euclidean distances for each edge of the graph enables Dijkstra algorithm to analyze and hence, determine the shortest trajectory as the optimal output.

Experimental Setup and Results

In order to validate the performance of our planner ROM, we have applied it on many different scenarios. For illustration, we considered five workspaces with having various constrains and limitations for each scenario. Each workspace has the size of 500 by 500 units and a start and goal configurations are considered as small circular points with the equally 10 points in diameters. We aim to evaluate the accuracy and skills of our planner to analyze the workspace and hence, compute the optimal trajectory. These scenarios along with the specification of each as well as the results of applying our path planner on them are categorized and illustrated in the following case studies. This paper focuses on demonstrating our algorithm on workspaces that are challenging for well-established exiting planners. We illustrate that our method overcomes those challenges. Early results show that our method produces the shortest paths [1]. More direct comparisons will be reported in subsequent papers.

Case study 1

For the first case study, we considered a crowded workspace with several obstacles in various shapes, sizes and locations. The aim of employing mentioned configuration is to put the path planner in an intense effort to build the ideal trajectory with the purpose of analyzing its performance along with its accuracy in finding the best possible collision-free route toward goal. We located the start point in the location of 50 by 50. The goal configuration is also considered at the position of 360 by 460 of the workspace Figure 7 shows the workspace arrangement that we considered for the first case study:

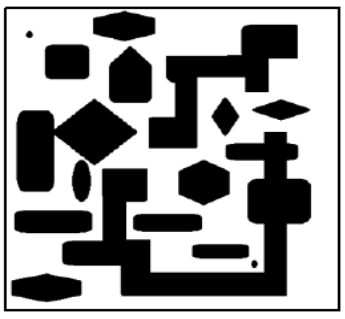

Figure 7: The workspace of the first scenario.

Figure 8 shows the result of building the optimal collision-free trajectory obtained from applying our path planner on the first workspace shown in the Figure 7.

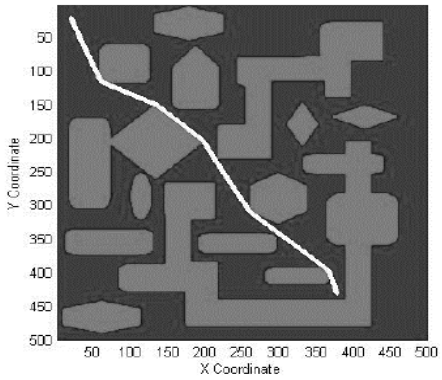

Figure 8: The trajectory resultant from applying our path planner on the first workspace.

As it illustrated in the Figure 8, our path planner was able to build a collision-free trajectory among various obstacles in the first workspace successfully. The distance trajectory from start to the goal configuration is calculated equaled 350 in Euclidean measurement system. We observed many minor and major roadblock obstacles surface scanning along with several trajectory adjustments which is the resultant of operation over a relatively large numbers of obstacles side edge nodes along with the roadblock surface nodes in form a complex lattice of nodes along with all possible joint paths among them.

Case study 2

As it indicated in the earlier sections, our path planner preforms a roadblock obstacle surface scanning when located to the hit point until reaching to a valid leave point. We used the following workspace to evaluate our path planner surface scanning skills. We located the start point at the 330 by 160 and the goal configuration at the 110 by 410 coordinates of the workspace. Generally, a surface scan will be enforced when either start or goal or both of them located inside a spiral or rounded obstacles (Figure 9).

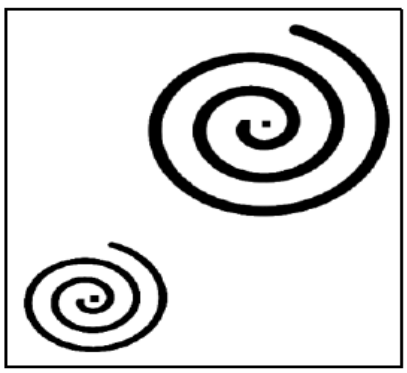

Figure 9: The workspace of the third case study.

As we expected to observe in this scenario, the majority portion of the optimal trajectory is located on the surface of the only roadblock obstacle in the workspace. This is because of the particular shape of the obstacle which prevents the planner to reach to a proper leave point toward the goal and hence, performs a long surface scanning throughout a continuously movement on the surface of the roadblock obstacle, as indicated in Figure 10.

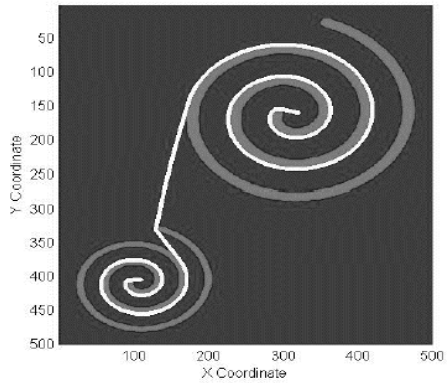

Figure 10: The resultant trajectory obtained from applying the path planner on the third workspace.

The distance of the collision-free trajectory computed by our path planner is equal 1569 units. The majority of the trajectory is located on the surface of the spiral roadblock obstacle except two zones of path from start point to the roadblock obstacle and the only leave point from roadblock obstacle toward goal configuration. This scenario revealed the strength of our path planner skills to compute and build the optimal trajectory in any types of workspaces including spiral obstacles without performance reductions.

Case study 3

In this scenario, we considered the start location inside a U-shaped obstacle while the goal configuration is located in the other side of the obstacle. This arrangement leads the planner evaluation skills in terms of building the shortest collision-free trajectory which is routed partially in opposite direction of the goal configuration. The start location is at the 190 by 230 and the goal configuration is in 450 by 210 coordinates of the workspace. Figure 11 shows the obstacle along with the start and goal points located in the fourth scenario:

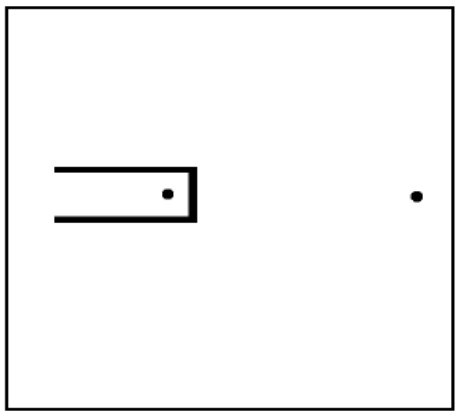

Figure 11: The workspace belongs to the fourth case study.

In this scenario, our path planner calculated the distance of 519 Euclidean for the optimal trajectory as illustrated in Figure 12.

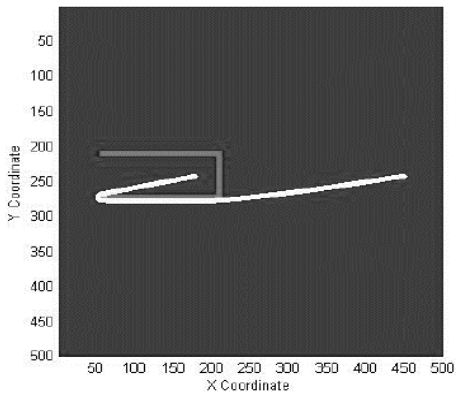

Figure 12: The optimal trajectory obtained from applying our path planner on the forth workspace configuration.

The results concluded from applying our planner on this workspace arrangement reveals that the planner is able to flawlessly handles and hence, draws a collision-free trajectory in situations that either start or goal configuration is located inside a partially closed obstacles along with the presents of various objects that are intensively located close to one another.

Case study 4

Figure 13 shows the workspace that we considered for this case study. This scenario is formed based on a maze shaped arrangement including several edges and walls in different directions. Placing the start point on top of the obstacle and goal configuration inside of the obstacle at the bottom side of it makes the planner to use extensive attempts to continuously analyzing and adjusting the optimal collision-free trajectory. The start and goal points for this scenario is considered to be located at the 150 by 50 and 370 by 470 coordinates of the workspace respectively.

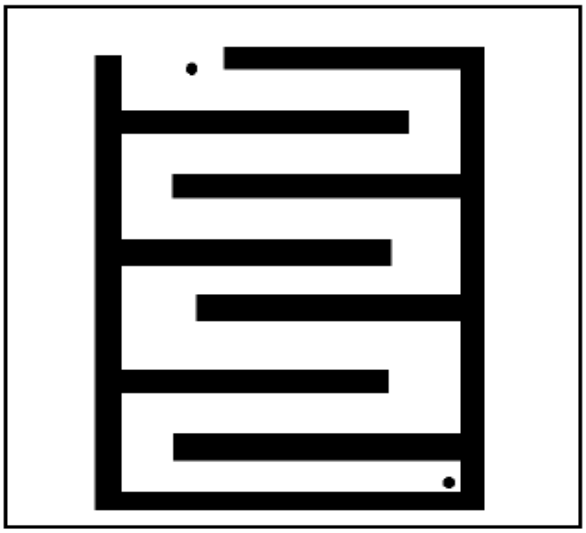

Figure 13: The workspace of the fifth case study.

The resultant of applying our path planner on the last considered scenario is illustrated in Figure 14.

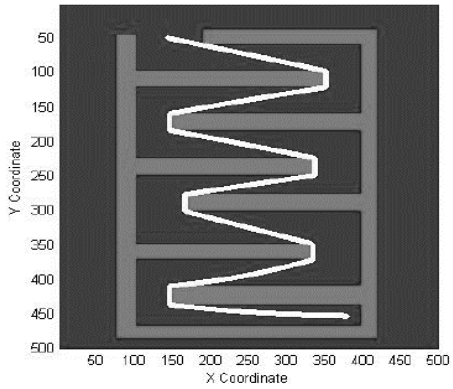

Figure 14: The trajectory resulting from applying our path planner on the fifth case study workspace.

The optimal collision-free trajectory distance that is built by our planner is equal 1568 units. Similar other scenario, our planner was able to successfully analyzing, computing and handling the situation and hence, concluded appropriate solution.

Conclusions

This research article introduced Rapidly Optimizing Mapper (ROM) as a novel approach for single point robots. In order to achieve the best possible result in terms of shortest safe trajectory, the planner algorithm benefits from a combination of many optimization strategies in different layers which enable it to be applied on a variety of different workspaces having different scenarios of start and goal configurations along with obstacle arrangements and shapes. The speed of handling workspaces for generating trajectories is one of our original contributions. The other major contribution is flexibility of handling complex environments using key parameters. These are reported in our most recent papers [1,31]. Despite many other offline path planners that are involved with problems in certain situations and obstacle shapes such as polygonal obstacles, we have observed that ROM is able to plan the trajectory on a more severely limited scenarios, successfully. For example, as it illustrated, it is able to be used to process any types of spiral edged obstacles flawlessly. It is able to detect any possible situations and hence, circumstances on the workspace and determines the best possible route appropriately. As instances of such a situations we can consider the local minima problems that the Potential Field planner is dealing, while ROM is able to successfully overcome to the mentioned situations. ROM analyzes the environment in order to find roadblock side edge nodes, it further transforms the workspace into a graph of roadblock side edge nodes along with recognizing and optimizing isolated nodes and eventually benefits of using Dijkstra's shortest path algorithm to refine the optimal path. The optimal path refers to the shortest path which allows the robot to perform a flawless transition in form of a near miss avoidance movement from initial point to the goal configuration.

References

1. Kamkarian P, Hexmoor H (2015) Efficiency Considerations of an Offline Mobile Robot Path Planner. The 17th International Conference on Artificial Intelligence, (ICAI'15): 35-40.

2. Amato NM, Wu Y (1996) A Randomized Roadmap Method for Path and Manipulation Planning. IEEE International Conference on Robotics and Automation 4: 113-120.

3. Khatib O (1985) Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. IEEE International Conference on Robotics and Automation, St Louis Missouri, pp. 500-505.

4. Borenstein J, Koren Y (1991) The Vector Field Histogram-fast Obstacle Avoidance for Mobile Robots. IEEE Transactions on Robotics and Automation 7: 278-288.

5. Grefenstette J, Schultz AC (1994) An Evolutionary Approach to Learning in Robots. Proceedings of the Machine Learning Workshop on Robot Learning International Conference on Robot Learning. New Brunswick, NJ, pp. 65-72.

6. ArámbulaCosío F, Padilla Castañeda MA (2004) Autonomous Robot Navigation using Adaptive Potential Fields. Mathematical and Computer Modelling 40: 1141-1156.

7. Ge SS, Cui YJ (2000) New Potential Functions for Mobile Robot Path Planning. IEEE Transactions on Robotics and Automation 16: 615-620.

8. Connolly CI, Burns JB, Weiss R (1990) Path Planning using Laplace's Equation. Proceedings of the IEEE Conference On Robotics and Automation, pp. 2102-2106.

9. Utkin VI, Drakunov S, Hashimoto H, Harashima F (1991) Robot Path Obstacle Avoidance Control via Sliding Mode Approach. Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems, Osaka, Japan, pp. 1287-1290.

10. Guldner J, Utkin V, Hashimoto H (1997) Robot Obstacle Avoidance in N-Dimensional Space using Planar Harmonic Artificial Potential Fields. Journal of Dynamic Systems Measurement and Control 119: 160-166.

11. Guechi E, Lauber J, Dambrine M (2008) On-line Moving-Obstacle Avoidance using Piecewise Bezier Curves with Unknown Obstacle Trajectory. 16th Mediterranean Conference on Control and Automation, pp. 505-510.

12. Mohammadi S, Hazar N (2009) A Voronoi-Based Reactive Approach for Mobile Robot Navigation. Advances in Computer Science and Engineering Springer Berlin Heidelberg 6: 901-904.

13. Bhattacharya P, Grvrilova ML (2007) Voronoi Diagram in Optimal Path Planning. 4th IEEE International Symposium on Voronoi Diagrams in Science and Engineering.

14. Choi J-W, Curry RE, Elkaim GH (2009) Obstacle Avoiding Real-Time Trajectory Generation and Control of Omnidirectional Vehicles. American Control Conference.

15. Hwang J-H, Arkin RC, Kwon D-S (2003) Mobile Robots at Your Fingertip: Bezier Curve On-line Trajectory Generation for Supervisory Control. IEEE International Conference on Intelligent Robots and Systems 2: 1444-1449.

16. Skrjanc I, Klan˘car G (2007) Cooperative Collision Avoidance between Multiple Robots Based on B´ezier Curves. 29th International Conference on Information Technology Interfaces.

17. Costa TAA, Ferreira AM, Dutra MS (2007) Parametric Trajectory Generation for Mobile Robots. 19th International Congress of Mechanical Engineering 3: 300-307.

18. Eren H, Fung CC, Evans J (1999) Implementation of the Spline Method for Mobile Robot Path Control. 16th IEEE In-strumentation and Measurement Technology Conference 2: 739-744.

19. Magid E, Keren D, Rivlin E, Yavneh I (2006) Spline-Based Robot Navigation. International Conference on Intelilgent Robots and System.

20. Ho Y-J, Liu J-S (2009) Collision-free Curvature-bounded Smooth Path Planning using Composite Bezier Curve Based on Voronoi Diagram. IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA).

21. Kavaraki LE, Svestka P, Latombe J-C, Overmars MH (1996) Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces. IEEE Transactions on Robotics and Automation 12: 566-580.

22. Barraquand J, Latombe J-C (1991) Robot Motion Planning: A Distributed Representation Approach. International Journal of Robotics Research 10: 628-649.

23. Laumond JP, Sekhavat S, Lamiraux F (1998) Guid-lines in Nonholonomic Motion Planning for Mobile Robots. In: Laumond (ed.) Robot Motion Planning and Control. Springer-Verlag, Berlin, pp. 1-53.

24. Abinaya S, Hemanth Kumar V, SrinivasaKarthic P, Tamilselvi D, Mercy Shalinie S (2014) Hybrid Genetic Algorithm Approach for Mobile Robot Path Planning. Journal of Advances in Natural and Applied Sciences 8: 41-47.

25. Bashra K, Oleiwi R, Hubert R (2014) Modified Genetic Algorithm Based on A* Algorithm of Multi Objective Optimization for Path Planning. Journal of Automation and Control Engineering 2: 357-362.

26. Chaari I, Koubaa A, Trigui S, Bennaceur H, Ammar A, et al. (2014) SmartPATH: An Efficient Hybrid ACO-GA Algorithm for Solving the Global Path Planning Problem of Mobile Robots. International Jounal of Advanced Robotics Systems: 1-15.

27. Ju M, Wang S, Guo J (2014) Path Planning using a Hybrid Evolutionary Algorithm Based on Tree Structure Encoding. The Scientific World Journal 2014: 1-8.

28. Loo CK, Rajeswari M, Wong EK, Rao MVC (2004) Mobile Robot Path Planning using Hybrid Genetic Algorithm and Traversability Vectors Method. Intelligent Automation and Soft Computing 10: 51-64.

29. McFetridge L, Ibrahim MY (1998) New Technique of Mobile Robot Navigation using a Hybrid Adaptive Fuzzy-Potential Field Approach. Computers and Industrial Engineering 35: 471-474.

30. Yao Z, Ren Z (2014) Path Planning for Coalmine Rescue Robot based on Hybrid Adaptive Artificial Fish Swarm Algorithm. International Journal of Control and Automation 7: 1-12.

31. Kamkarian P, Hexmoor H (2015) A Novel Offline Path Planning Method. The 17th International Conference on Artificial Intelligence (ICAI'15): 10-15.