Two-stage Shortest Path Algorithm for Solving Optimal Obstacle Avoidance Problem

Article *in* IETE Journal of Research · January 2011

DOI: 10.4103/0377-2063.83650

Two-stage Shortest Path Algorithm for Solving Optimal Obstacle Avoidance Problem

Nizami Gasilov, Mustafa Doğan and Volkan Arici

Baskent University, Faculty of Engineering, Baglica-06810 Ankara, Turkey

ABSTRACT

In most of the path-planning applications, the controlled object (mobile robot) is expected to reach its predetermined target by following the shortest path and avoiding the obstacles. This navigation problem is also called optimal obstacle avoidance. In this work, obstacles are assumed to be motionless circles in different sizes. The object is supposed to be a point robot. The two-stage algorithm is proposed to find a numerical solution to the problem. At first stage, the method, which is optimal for one step, is applied iteratively. In every step of the method the first obstacle on the straight line between the current position and the target is assumed to be a single obstacle. The proposed method is realized using geometric representations. Some evaluations are made to prove that the method is convergent. The path obtained at the first stage might not be optimum. However, its length can be used to limit the feasible region through an ellipse, which contains the shortest path. Thus, the reduced search space makes the next stage more efficient and endurable for real-time applications. In the second stage of the algorithm, the elliptic region is meshed with squares, the side length of which is set in agreement with the minimum distance between obstacles. It is prohibited to pass through the squares that intersect obstacles. Thus, by discretization the problem becomes the shortest path problem in a graph, and is solved by applying the Dijkstra's algorithm. The proposed two-stage algorithm is verified with numerical simulations. Obstacles are chosen randomly. A target position is selected and fixed. For different starting points, the algorithm is tested repeatedly. The results show that the proposed algorithm can be applied to find an optimal solution for the obstacle avoidance problem.

Keywords:

Dijkstra's algorithm, Graph theory, Obstacle avoidance, Path planning.

1. INTRODUCTION

Various methods have been proposed for the solution of obstacle avoidance problem [1-9]. One of the real-time methods that has been developed for navigation of mobile robots is potential field approach [1,2]. The main advantage of this method is on-line efficiency as a result of the integration of the low-level robot control and path planning. However, its main disadvantage is that in some cases it could not escape from local minima that result in abnormal termination without reaching the target. Harmonic potential functions [1] and navigation functions [2] are proposed to overcome these difficulties and in this way obstacle avoidance is succeeded, but optimal path finding cannot be achieved. Besides, navigation functions are difficult to calculate and impossible to be implemented in real-time, especially for robots that have many degrees of freedom [4]. Furthermore, navigation functions should be differentiable by the definition and therefore, they can cause problems in piece-wise continuous or saturated robot control applications [2]. Nevertheless, potential field method is improved by the recent advances in both theoretical and application aspects, e.g., 3-D extension [10-13].

Probabilistic roadmap for path planning is just another alternative method [3,4]. This method, in comparison with the previous ones, can be more reliable and applicable in more general cases. On the other hand, theoretic analysis becomes more complex, which is an important disadvantage of this method.

Dynamic programming (Hamilton-Jacobi-Bellman) methods are used extensively as well [7-9]. For example, in [9], near-optimal solutions for the shortest path problem have been obtained by the geometric approach efficiently. The disadvantage of this method is that it lacks the minimal path in some cases.

Some efficient shortest-path algorithms for mobile robots are proposed based on graph theory approach [5,6]. Furthermore, there are other graph-based works especially for convex polygonal obstacles, e.g. [14-16]. The latter approaches are capable of search space reduction with some assumptions. The methodology used in these works can be summarized in three steps: (1) image processing for obstacle identification and approximation of obstacles with simple geometric shapes; (2) Road map (undirected graph) construction; (3) Finding the shortest path at composed graph. Previous researches on realization of these three steps, proposed methods and developed algorithms have been successfully aggregated in [14] by Huang *et al*. The structure of these researches can be described as below: Step (1) Obstacles are transformed into convex polygonal shapes with convex hull algorithm. Step (2) Visibility graph (V-graph) and vertices circle (V-circle) concepts are used to constitute the road map as follows: Given a pair of convex polygonal objects A and B, the reduced visibility graph consists of at most four distinct tangential line segments. For the start point S (or, final point F) and given a convex polygonal object C, at most two distinct segments will be added to graph. So, road map graph, which consists above tangential segments and polygonal boundaries of obstacles, will be obtained [14]. Thus, path planning problem becomes the shortest path problem in a graph (Step 3). Computational time will be excessive for the whole graph. This causes a new problem: Finding a reduced feasible region that contains the shortest path in order to achieve fast algorithm. In [14], this type of reduced search area is called as active region and is obtained under certain assumptions.

In this research, a new two-stage optimization algorithm is developed. Our motivation is the search space reduction. The proposed main idea to achieve this is described below. Assume we do not know optimal path, but have an upper bound L for its length: $L_{optimal} \le L$. We prove strictly that the feasible region, which contains optimal path, will be inside of an ellipse with focuses at start and finish points and with semi-major axis $a = L/2$ and semi-minor axis $b = \sqrt{L^2 - |SF|^2} / 2$. We find an upper bound L for optimal path in the first stage of the proposed method. For this purpose, we use geometric approach, and to explain how it works we use circular obstacles as convenient instrument for efficient visual representations. The second stage uses graph, obtained by discretization of the reduced elliptic region, and does not depend on form of the obstacles.

Consequently, at the first stage of the proposed method, near-optimal solution is provided by geometric incremental approach, and this solution is used to describe the elliptic region that contains the shortest path. Different from literature above, we have a firm proof of that the feasible region can be restricted to an ellipse, without any assumptions. Thus, the optimal solution can be readily searched by Dijkstra's algorithm after the completion of the first stage.

Different from the graph-based heuristic algorithms, e.g., A* [17-21], the proposed method does guarantee that the selected path is optimal. Furthermore, A* algorithm can result in the much longer path than Dijkstra's one, depending upon crucial choice of the heuristic function and world configuration.

The most important novelty of this work is that the initial search space is reduced a lot in order to find the shortest path efficiently. Therefore, two main disadvantages of Dijkstra's algorithm, namely large computational burden and difficulty with following the discrete paths [5], have been overcome by search space reduction and greedy path construction approach that explained in Section 4. These two properties are indispensable in real-time applications.

2. PROBLEM DEFINITION

Suppose, motionless circular obstacles located in rectangular domain (search space) are given in finite number. It is assumed that no obstacle cuts or touches any other obstacle. The motivating question behind this research is how point robot can navigate on the shortest path from a given starting point S to a given target position F with obstacle avoidance.

Note that the condition about point robot is not a restriction for the problem. Let the robot be circular with radius ρ. If we enlarge all obstacles in the amount of ρ radius-wise, then the robot itself can be considered as point robot.

Also, note that the proposed approach can be easily extended for the case when other types of obstacles such as ellipses, convex polygons are considered together with circles. Details can be found in [15,16].

Two-stage algorithm is proposed for numeric solution to the problem. The detailed explanations of these stages are given in the following sections.

3. THE FIRST STAGE: APPROXIMATE INCREMENTAL METHOD BASED ON GEOMETRY

The method applied at this stage is incremental since it is optimal just for one step. The method is realized by using geometric representations. The first obstacle on the straight line between the current position of the object and the target is assumed to be a single obstacle in each step of the method. In accordance with this, the tangential path is determined firstly from the initial point to this obstacle. Besides, extra obstacles are controlled whether they intersect the path or not. If not (refer to Subsection 3.1), this path is used to reach the obstacle. Then, the path is followed along the boundary of the obstacle until the point, where tangent from the target touches the obstacle. This point becomes the new starting point for the next step. If there are extra obstacles across the tangential path that connects the initial point S and the first obstacle (refer to Subsection 3.2), then the extra obstacle that is closest to S will be determined. This extra obstacle is reached along the tangential path closer to the baseline SF and avoided by following its boundary. Then, arrival point is determined as new starting point for next step. This process will be iterated until there is no obstacle on the way to target.

3.1 Single Obstacle Avoidance

Assume that on the path SF, there is only one circular obstacle with the radius r and centered at C as represented in Figure 1. Two pairs of tangent lines from points S and F can be drawn to the circle. We can choose the ones that have minimum angle with line SF, i.e., SA_1 and FB_1 in the figure. Therefore, according to geometrical rules the shortest path consists of line SA_1, arc A_1B_1 and line B_1F.

In order to calculate coordinates of points A_1 and A_2, the following equation can be used:

$$\mathbf{SA}=\frac{\sqrt{l^2-r^2}}{l}P_{\pm\theta}\mathbf{SC},$$

where SA and SC are vectors; θ=arcsin(r/l); P_θ is the rotation operator about point S through an angle θ. The sign of θ corresponds to choosing one of the points A_1 and A_2. One of them, which is the closest point to baseline SF, is selected, either A_1 or A_2.

We can make an important evaluation that will be used to prove convergence of approximate geometric method. Since circular obstacle centered at C crosses the line SF, we have: $d<r$ [Figure 2]. Hence,

$$|SF|=|SH|+|HF|\geq HF|=\sqrt{|CF|^2-d^2}>\sqrt{|CF|^2-r^2}$$
$$=|BF|\Rightarrow|BF|<|SF|.$$

Note that $|BF|$ and $|SF|$ are direct distances to the target before and after avoidance. According to last inequality, direct distance decreases by avoiding an obstacle. Avoided obstacle will not be considered again. If the obstacle is the last one to be avoided, the target will be reached.

3.2 Extra Obstacle Avoidance

There could be some extra obstacles across the tangential path SA that is mentioned in Subsection 3.1. It is represented in Figure 3, how the path can be constructed in this case. In Figure 3, the obstacle centered at C is ordinary one on the path SF, and the obstacle centered at E is the extra one.

The algorithm implemented for extra obstacle avoidance is explained briefly below.

Among extra obstacles crossing tangential path SA, the obstacle that is the closest one to the base point S is determined, i.e. the obstacle centered at E in Figure 3. Direction SF will be our reference to avoid this obstacle. Tangential path SP close to line SF is determined. Subsequently, QR, common cross tangent of obstacles E and C with end point Q close to P, is calculated. The obstacle E has been avoided by following tangent line SP first, and then arc PQ. Then the question is considered whether there is any other extra obstacle on path QR, or not. If not, then by following tangent line QR and arc RB the ordinary obstacle C will be avoided. If there is an extra obstacle, new iteration on avoidance of extra obstacle is started with taking Q as the new initial point.

Since number of the obstacles is finite, extra obstacles will be eliminated after finite number of steps and the ordinary obstacle will be avoided next. Refer to end of the Subsection 3.1, the evaluations prove that direct distance to the target decreases by avoiding ordinary obstacle. There is finite number of obstacles by assumption and the distance to the target diminishes at each step, then approximate method based on geometry is convergent.

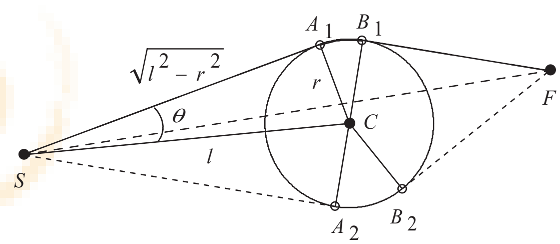


Figure 1: Optimal avoidance of a single obstacle.

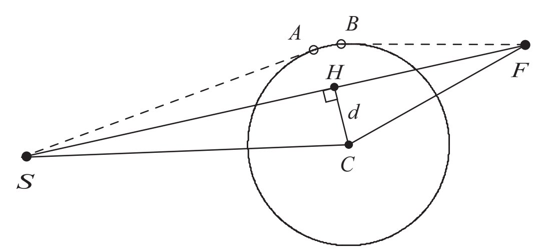


Figure 2: Schematic representation to prove that the direct distance to the target decreases with obstacle avoidance.

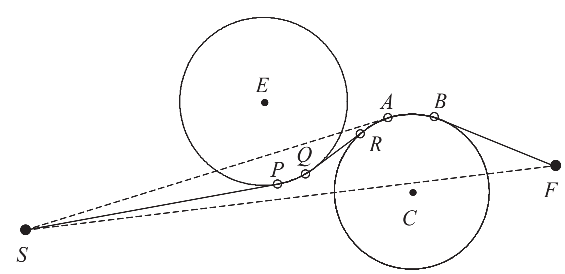


Figure 3: Avoidance of an extra obstacle.

Remark: The path obtained in this stage consists of lines and arcs sequence, e.g., line$S$, arc1, line12, arc2, ... , line$F$. Some lines between two obstacles can be replaced by their common tangent lines if this common tangent does not intersect any extra obstacle. Consequently, a shorter path can be obtained as further improvement.

Geometric method implemented at the first stage of the main algorithm results in near-optimal solutions. The path obtained through this method might not be optimal. Such an example is given in Figure 4.

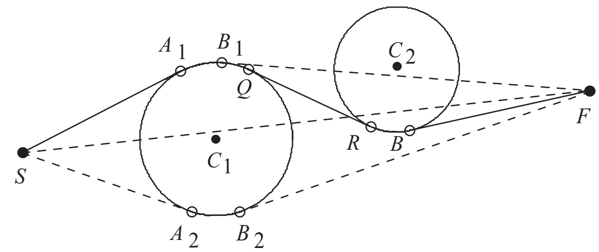


Figure 4: An example for which the path obtained by the geometric method is not optimal.

We can see how the method works for this example below. Circle C_1 is the first ordinary obstacle across the path SF. According to Subsection 3.1, this obstacle will be avoided following tangent path SA_1 closer to baseline SF and then arc A_1B_1. Taking B_1 as the new starting point, the next step of the method will be initiated. Circle C_2 is determined as the ordinary obstacle across path B_1F. In order to avoid it, the tangent, which is closer to the baseline B_1F, is calculated. This tangent line crosses C_1. Thus, in this time the circle C_1 becomes extra obstacle when ordinary obstacle C_2 is avoided. The procedure described in this subsection is implemented to avoid the obstacle C_1. Since the starting point B_1 lies on C_1, the step to reach the extra obstacle will be passed. Only arc B_1Q is used to avoid C_1 (Here Q is the end point of QR, common cross tangent of circles). At the last iteration of the method, by following tangent line QR and arc RB, avoidance of the ordinary obstacle C_2 will be completed and by tangent path BF the target will be reached. Thus, the path calculated on proposed geometric method is SA_1QRBF. As it can be easily seen from Figure 4, this path is longer than the path SA_2B_2F, and consequently, is not optimal.

Thus, in general, solutions obtained through geometric method are only near-optimal. To find the optimal path the second stage of algorithm is applied, which is explained in the next section.

4. THE SECOND STAGE: OPTIMAL PATH BY DIJKSTRA'S ALGORITHM

We will prove later that the feasible region can be restricted to an ellipse. The proof is based on the geometric definition of ellipse: An ellipse is a curve that is the locus of all points the sum of whose distances d_1 and d_2 from two fixed points S and F (focuses) separated by a distance of $|SF|=2c$ is a given positive constant $L=2a$. Where a is the semi-major axis, $b = \sqrt{a^2 - c^2}$ is known as the semi-minor axis.

As it is mentioned in the previous section, the path obtained at the first stage might not be the optimal one. However, its length L_1^* gives an upper bound for optimal path length L^* such that $L^* \leq L_1^*$.

The feasible region that contains the optimal path can be reduced with this inequality on purpose.

Let X be an intermediate point on optimal path. Then it can be claimed that

$$L^* = L_{SF}^* = L_{SX}^* + L_{XF}^*.$$

Since the shortest path should be a line segment with no consideration for obstacles, the following inequalities can be written: $|SX| \leq L_{SX}^*$ and $|XF| \leq L_{XF}^*$.

Thus, we get $|SX| + |XF| \leq L_1^*$.

Regarding this inequality, sum of distances from S and F to a point X lying in the feasible region cannot exceed the value L^*_1. As a result, the point X will be inside the ellipse with focuses at S and F. Subsequently, the feasible region is inside the ellipse as well. Hence, based upon the value L^*_1 the feasible region can be diminished and restricted to an ellipse. Thus, the reduced search space makes the second stage much more efficient and endurable for real-time applications.

In this stage, coordinate transformation is applied such that new origin will be the midpoint M of the line segment SF, and the new horizontal axis will be in the direction of ray MF. In this new coordinate system, the feasible region can be described simply as follows:

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 \leq 1, \tag{1}$$

where $a = L_1^*/2$ and $b = \sqrt{(L_1^*)^2 - |SF|^2}/2$. In the mean time, changing the coordinate system is also beneficial such that the realizations of the following steps will be more efficient.

Discretization of the problem is the next step. For this purpose, a grid with equal squares is created over the region. The side length of a square, h, is complied with the minimum distance between obstacles, δ, such that

$h \le \delta/3$ (as a strong form of $h \le \delta/(2\sqrt{2})$) to avoid some possible errors. Intersection points of the grid or nodes, are assigned as graph vertices. Thus, the analyzed problem can be solved by graph theory approach. We can define two prohibited cases such that a) If the vertex N is out of feasible region, or b) If the square with side length h and centered at N intersects an obstacle. In both cases, the vertex N is marked as forbidden to pass. Graph edges can be constructed in two alternative ways such that

(1) 8-neighborhood vertices around any vertex V, which is not prohibited, are examined one by one. The edge is added between the vertex V and the one, which is permitted to pass.
(2) All pairs of vertices $(U; V)$ are to be examined one by one. If the vertices of a pair $(U; V)$ are not prohibited and line segment UV does not intersect any obstacle, the edge with the length $|UV|$ is constructed between U and V.

At the first alternative, discrete approach is also used to construct the edges. Therefore, the total number of edges is minimal and edge structure is easy to process. In the second alternative, which can also be characterized as greedy approach, edge structure is difficult to implement. However, it provides solution closer to the optimal solution than the first one does. In simulations, the results of which are represented in the next section, the second approach is applied.

Thus, the solution of the problem is reduced only to find the shortest path from vertex S to vertex F in the obtained graph. This new problem is solved by applying Dijkstra's algorithm [10,11]. Furthermore, some improvements have been done based upon the properties of the problem in order to make the Dijkstra's algorithm application more efficient. For instance, forbidden vertices are not included to the set of graph vertices. Let v be the number of graph vertices. If the first alternative mentioned above is realized then instead of weight matrix of size $v \times v$ a zero-one (or binary) matrix of size $8 \times v$ is used. Hence, this approach is suitable for real-time applications. For the second alternative, as weight matrix is symmetric, then only lower triangle matrix can be stored at memory.

Note that in the first alternative the graph is sparse (number of edges $e \sim 8v$). In this case, the complexity of Dijkstra's algorithm, implemented with a binary heap, is $O(e \log v) \sim O(v \log v)$.

In order to evaluate the efficiency of the proposed algorithm, we need to analyze its complexity. It is known that Dijkstra's algorithm for a graph with v vertices has computational complexity of $O(v^2)$ regarding number of operations. If obstacles are assumed to be distributed uniformly, we can make an approximation such that $v \sim A_{region}$, where A_{region} is the area of the region. Thus, the proposed algorithm has $p=(A_{scene}/A_{ellipse})^2$ times as much efficiency as Dijkstra's algorithm when it is applied to the whole scene with the area of A_{scene}.

If the scene is considered to be a square then $A_{scene}=(a_{scene})^2$. On the other hand, the reduced search space can be represented as an ellipse with the area of $A_{ellipse} = \pi ab$.

The area of the ellipse is also shown (see, equation (1)) to be $A_{ellipse} = \pi \cdot \frac{L_1^*}{2} \cdot \frac{\sqrt{(L_1^*)^2 - |SF|^2}}{2}$ by above calculations. The direct distance $|SF|$ between S and F points is determined by input data, and can be considered as a given value. Besides, L_1^* is the length of the path that is obtained by geometric approach and can get different values depending on obstacle distribution. In the worst case, when L_1^* gets larger values the area of ellipse becomes bigger so that efficiency gain p as shown above will be diminished.

One of the main reasons that increases L_1^*, is using the common cross tangents between circular obstacles in the geometric path. Starting and finishing the path with circular arcs, and obstacles that come as close as possible each other, also cause to increase L_1^* as well. We can make an upper bound evaluation for L_1^* based on these conclusions.

The following assumptions are provided as a matter of convenience in order to make the evaluation. Obstacles are supposed to be circular with the same radius, r. The minimal distance between two obstacles would be $d=2r$. Finally, we assume that $|SF| \le a_{scene}/2$ and the number m of obstacles on the line segment SF would be at least 3: $m \ge 3$.

Note that upper bound evaluation for L_1^* can be obtained by the following representation [Figure 5] and the previous explanations.

Referring to Figure 5

$$L_1^* \le L_{SM_1} + \sum_{i=1}^{m-1} (L_{M_i B_i} + |B_i A_{i+1}| + L_{A_{i+1} M_{i+1}}) + L_{M_m F}$$

$$L_1^* \le \frac{\pi}{2} r + (m-1)(\frac{\pi}{6} r + 2\sqrt{3} r + \frac{\pi}{6} r) + \frac{\pi}{2} r \le [\pi + 4.52(m-1)]r$$

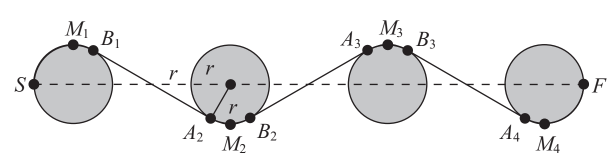


Figure 5: The worst-case diagram for geometric path.

$$|SF| = [2+4(m-1)]r$$

$$\frac{L_1^*}{|SF|} \le \frac{\pi+4.52(m-1)}{2+4(m+1)} = \frac{4.52}{4} + \frac{\pi-2.26}{2+4(m-1)}$$

Thus, we obtain the following evaluation for case $m \ge 3$:

$$\frac{L_1^*}{|SF|} \le 1.13 + \frac{0.882}{10} \le 1.22 \Rightarrow L_1^* \le 1.22\,|SF|$$

$$A_{ellipse} \le \pi \cdot \frac{1.22\,|SF|}{2} \cdot \frac{\sqrt{1.22^2-1}\,|SF|}{2} \le 0.67\,|SF|^2$$

If $|SF| \le a_{scene}/2$, then

$$\frac{A_{scene}}{A_{ellipse}} \ge \frac{(a_{scene})^2}{0.67(a_{scene}/2)^2} = \frac{4}{0.67} \ge 5.9$$

Consequently, we get $p=(A_{scene}/A_{ellipse})^2 \ge 34.8$ and observe that the proposed algorithm requires much less computational power at least 34.8 times. If the S and F points are closer ($|SF|$ would be smaller), then efficiency gain will increase, e.g. $p \ge 180.4$ is calculated for case of $|SF| \le a_{scene}/3$.

5. SIMULATION RESULTS

The proposed two-stage algorithm is verified by many simulations. In simulations the obstacles are chosen randomly in a rectangular region. The target position is selected. Then the proposed algorithm is executed for different starting positions.

The results of one simulation are represented in Figure 6. Here, we take scene with size of $a \times b$=120×120 (unit length can be assigned arbitrarily). We randomly generate circles (x_c; y_c; r_c) with radius $r_c \in [4, 8]$. If next candidate circle does not intersect an existing one, we add this circle to the list of obstacles. Otherwise the candidate one is rejected. When predicted number of obstacles is achieved we stop the process. In calculations, we take number of attempts to be limited.

The paths that are obtained by the first stage have been represented with solid-line in Figure 6. For one of the starting points, (S_2), optimal path by the second stage has been shown as dashed-line. This optimal path has essential differences in comparison with the result of the first stage (solid-line starting from S_2). For other cases (S_1, S_3, S_4), the optimal paths, obtained at the second stage, have not been represented for the purpose of clarity of the figure, since they do not differ a lot from drawn ones.

For the case with starting point S_2, the boundary of feasible region, used at the second stage, is shown by an ellipse (thin dashed-line) in Figure 6. This ellipse envelops an area, which is about 1/5 of the whole search space (rectangle). Since the operation complexity of Dijkstra's algorithm is $O(v^2)$ and v is proportional to covered area, the benefit of proposed algorithm is expected to be about 25 times better than the algorithm applied to whole region. In Table 1, we represent the comparison of actual computational times for whole and restricted search regions. In calculations, we use a PC with the following properties: AMD Athlon(TM) XP 2000+, 1.67 GHz, 224 MB RAM. From Table 1 it can be seen that actual computational time is diminished considerably by reducing the search space.

In cluttered areas or in regions that have many close adjacent obstacles, the length of the path can increase or the structure of the path can become quite complex. However, number of forbidden vertices will increase and result in eligible vertices in graph will decrease. Thus, computation time would not change significantly.

It has been verified by simulations that the proposed algorithm is useful to solve the optimization problem for obstacle avoidance. According to obtained results, in some cases only the first stage of the algorithm can be sufficiently used, especially considering robotic applications that require essential time and memory resources.

Although the proposed algorithm works well for circular obstacles, more efficient approximations for obstacles can

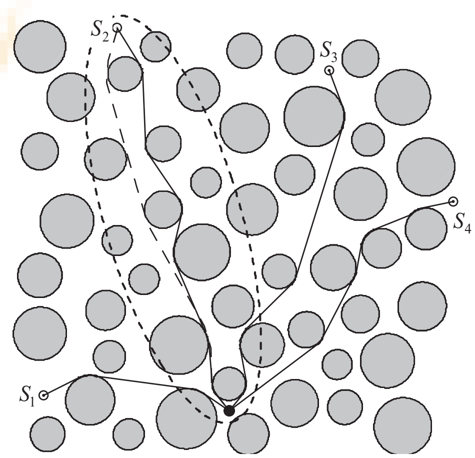


Figure 6: Near-optimal (solid lines) and optimal paths (dashed line) obtained from calculations in presence of 50 obstacles. Thin dashed line represents the boundary of elliptic feasible region used at the second stage of the algorithm.

Table 1: The comparison of actual computational times

	Whole region	Reduced region
Shape of the region	rectangle (120 x 120)	ellipse (radii: 54;17)
Area of the region	14400	2976
Number of obstacles, N	50	16 of 50
Grid step size, h	0.353	0.353
Time to compose the graph, T_G	1.06 sec	0.27 sec
Runtime of Dijkstra's algorithm, T_D	94.17 sec	1.61 sec

be obtained by implementing the other convex figures, e.g., rectangles and ellipses. The geometric method can be extended easily to cover these shapes.

6. EXPERIMENTAL RESULTS

Pioneer 3-DX mobile robot, which has embedded computer with C++ based ARIA (Advanced Robotics Interface for Applications) software and wireless communication capability, has been controlled by remote PC. Driving capabilities of the robot are 2-wheel drive, plus rear balancing caster with differential steering.

As shown in Figure 7, after extensive image processing, necessary path planning commands are produced by the proposed algorithm that all running at PC, and are transmitted through wireless network to the robot. Obstacles are chosen as circular shaped disks. Besides, their positions are selected in accordance with robot dimensions (swing radius is 32 cm), and minimum inter-distance requirements, see Section 4.

The first preliminary experiments are done with two obstacles to evolve the implementation. Finally, the last experiment is done with five obstacles. After image processing with the initial conditions given in Table 2, the algorithm is implemented in an efficient way. At the end, the robot followed the prescribed path successfully as planned beforehand.

One of the important experimental results is removing the point robot assumption and also aware of problems, e.g., orientation of the robot and real-time applications requirements.

For future work, automatic identification and setting the robot orientation and pose will be an important achievement, since it took time to set the right orientation for the robot. Integrating both stages of the algorithm with image processing to work in real time while obeying the dynamic constraints will complete this research project.

7. CONCLUSION

Optimization problem for obstacle avoidance on the plane has been investigated. Two-stage algorithm has been proposed for solution of the problem and tested successfully with experiments. In the first stage, near-optimal solution is obtained through geometric approach. Using this solution, the feasible region is restricted to an ellipse. At the second stage using grid map, the problem is reformulated as the shortest path problem in graph, and optimal solution is found by applying Dijkstra's algorithm in the reduced search space. Consequently, two main contributions of this research come out clearly at the last stage. The first one,

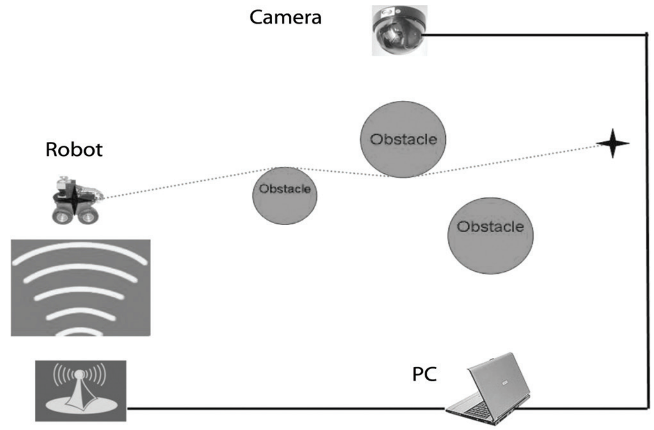


Figure 7: Experimental set-up.

Table 2: Initial conditions for the experiment

X axis	Y axis	
World Coordinates (cm)		
-7.45	40.00	Start
227.45	42.00	Finish
Positions and radii of obstacles (cm)		
155.00	50.00	16.33
63.00	50.00	11.51
110.00	35.00	16.13
25.00	35.00	15.35
192.00	37.00	9.42

the solution is optimal (with accuracy of grid step size), and the second one, it is obtained through an efficient way with a significant reduction of search space.

The proposed algorithm is computationally efficient and successful for finding the optimal path. Therefore, it is suitable for real-time robotic applications.

REFERENCES

1. C I Connoly, J B Burns, and R Weiss, "Path planning using Laplace's equation", Proc. IEEE Int. Conf. Robot. Automat., Vol. 1, pp. 2102-6, 1991.
2. E Rimon, and D E Koditschek, "Exact robot navigation using artificial potential functions", IEEE Transactions on Robotics and Automation, Vol. 8, pp. 501-18, 1992.
3. L E Kavraki, M L Kolountzakis, and J C Latombe, "Analysis of probabilistic roadmaps for path planning", IEEE Transactions on Robotics and Automation, Vol. 14, No. 3, pp. 166-71, 1998.
4. L E Kavraki, P Svestka, J C Latombe, and M H Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces", IEEE Transactions on Robotics and Automation, Vol. 12, No. 4, pp. 566-80, 1996.
5. R V Helgason, J L Kennington, and K R Lewis, "Cruise missile mission planning: A heuristic algorithm for automatic path generation", Journal of Heuristics, Kluwer Academic Publishers; Vol. 7, pp. 473-94, 2001.
6. Y H Liu, and S Arimoto, "Path planning using a tangent graph for mobile robots among polygonal and curved obstacles", Int. J. Robot. Res., Vol. 11, No. 4, pp. 376-82, 1992.
7. S Dreyfus, "Dynamic Programming and the Calculus of Variations", New York, London: Academic; 1965.

8. A I Moskalenko, "Bellman equations for optimal processes with constraints on the phase coordinates", Autom. Remote Cont. (Translation of Avtomatika i Telemekhanika), Vol. 4, pp. 1853-64, 1967.
9. S Sundar, and Z Shiller, "Optimal obstacle avoidance based on the Hamilton-Jacobi-Bellman equation", IEEE Transactions on Robotics and Automation, Vol. 13, No. 2, pp. 305-10, 1997.
10. J H Chuang, "Potential-based modeling of three-dimensional workspace for obstacle avoidance", IEEE Transactions on Robotics and Automation, Vol. 14, No. 5, pp. 778-85, 1998.
11. J H Chuang, and N Ahuja, "An analytically tractable potential field model of free space and its application in obstacle avoidance", IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, Vol. 28, No. 5, pp. 729-36, 1998.
12. J Ren, K A McIsaac, and R V Patel, "Modified Newton's method applied to potential field-based navigation for mobile robots", IEEE Transactions on Robotics, Vol. 22, No. 2, pp. 384-91, 2006.
13. N J Cowan, "Navigation functions on cross product spaces", IEEE Transactions on Automatic Control, Vol. 52, No. 7, pp. 1297-302, 2007.
14. H P Huang, and S Y Chung, "Dynamic Visibility Graph for Path Planning", Proceedings of IEEE/RSJ, Intl. Conf. on Intelligent Robots and Systems, (IROS 2004) Sendai-Japan, pp. 2813-8, 2004.
15. T Lozano-Perez, and M A Wesley, "An algorithm for planning collisionfree paths among polyhedral obstacles", Commun. ACM, Vol. 22, pp. 560-70, 1979.
16. H Rohnert, "Shortest paths in the plane with convex polygonal obstacles", Information Processing Letters, Vol. 23, pp. 71-6, 1986.
17. R Dechter, and J Pearl, "Generalized best-first search strategies and the optimality of A*", Journal of the ACM, Vol. 32, pp. 505-36, 1985.
18. P E Hart, N J Nilsson, and B Raphael, "A formal basis for the heuristic determination of minimum cost paths", IEEE Transactions on Systems Science and Cybernetics, SSC-4, No. 2, pp. 100-7, 1968.
19. P E Hart, N J Nilsson, and B Raphael, "Correction to a formal basis for the heuristic determination of minimum cost paths", SIGART Newsletter, Vol. 37, pp. 28-9, 1972.
20. N J Nilsson, "Principles of artificial intelligence", Palo Alto: Tioga; 1980.
21. J R Bruce, and M M Veloso, "Safe multi-robot navigation withindynamic constraints", Proceedings of the IEEE, Vol. 94, No. 7, pp. 1398-411, 2006.
22. J A Anderson, "Discrete Mathematics with Combinatorics", 2nd ed, Prentice Hall; 2004.
23. E Dijkstra, "Two problems in connexion with graphs", Numerische Mathematik, Vol. 1, pp. 269-71, 1959.

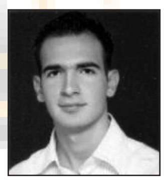
AUTHORS

Nizami Gasilov is an Associate Professor at Baskent University, Ankara, Turkey. He received B.Sc. and Ph.D. degrees in applied mathematics in 1984 and 1989, respectively, from Lomonosov Moscow State University, Russia. From 1987 to 2001 he worked at the Institute of Terrestrial Magnetism, Ionosphere and Radio Wave Propagation, Russian Academy of Sciences. His research interests include the areas of mathematical modeling, numerical methods, fuzzy sets and systems, graph theory, modeling of equilibrium and stability in tokamak, models of ionosphere.

E-mail: gasilov@baskent.edu.tr

Mustafa Doğan received the B.Sc. degree in electronics and communications engineering from Istanbul Technical University, the M.Sc. degree in electrical and electronics engineering in 1999 from Old Dominion University, Norfolk, VA, and the Ph.D. degree in optimal and adaptive nonlinear control in 2006 from Bogazici University, Istanbul, Turkey. He is an Assistant Professor at Baskent University, Ankara, Turkey. His research interests include Markovian jump-linear systems, flexible robot arms, optimal nonlinear control, adaptive internal model application, power electronics and tissue engineering.

E-mail: mustafa.dogan@gmail.com



Volkan Arıcı was born in Turkey in 1982. He received the B.Sc. degree in antenna engineering and robotics in 2005 from Bilkent University, Turkey. He received the M.Sc. degree in electrical and electronics engineering in 2008 from Baskent University, Ankara, Turkey. His research interests include the areas of image processing, robotics, vision and control, nonlinear system theory.

E-mail: varici@gmail.com

DOI: 10.4103/0377-2063.83650; Paper No JR 588_09; Copyright © 2011 by the IETE