Int. J. Mech. Eng. & Rob. Res. 2014

International Journal of Mechanical Engineering and Robotics Research

ISSN 2278 – 0149 www.ijmerr.com
Vol. 3, No. 1, January 2014

Research Paper

COLLISION AVOIDANCE RECURSIVE ALGORITHM

K Prasanna Lakshmi[1]*

*Corresponding Author: K Prasanna Lakshmi, ✉ prasannakaujala@gmail.com

This paper presents a technique for providing collision avoidance of arbitrary objects in a computer graphics system. The recursive algorithm implements the method of avoiding the collisions by taking an alternate path and therefore among the available alternate paths, the shortest one is chosen. Under the cluttered environment, maneuvering among the static and moving obstacles is the biggest hurdle which the robotic industry is facing. A small attempt is made to solve these issues.

Keywords: Collision avoidance, Arbitrary objects, Recursive algorithm

INTRODUCTION

An autonomous robot can learn or gain new capabilities like adjusting strategies for accomplishing its task(s) or adapting to changing surroundings. The theoretical and practical understanding of some of the issues has increased rapidly in the combined work of researchers in Artificial Intelligence, theoretical Computer Science, Mathematics and Mechanical Engineering. The methods definitely require the ability to plan motions to be developed automatically. Except in limited and carefully engineered environments, it is not realistic to anticipate and explicitly describe to the robot, all the possible motions that it may have to execute in order to accomplish requested tasks. Even in those cases where such a description is feasible, it would certainly be useful to incorporate automatic motion planning tools in off-line robot programming systems. Therefore an algorithm is proposed which is intelligent enough to make the robot take the decision if an extra robot is added or any change occur in the present system and act accordingly by considering the shortest path with less time.

PREVIOUS WORK

Many researchers have contributed, through their work, considering the environment involving the robots with different obstacles and suggested methods to overcome not only the hurdles but also constraints pertaining to the movement of the obstacles.

[1] Department of Mechanical Engineering, JNTUH College of Engineering, Manthani.

The dynamic problem is simulated for turning machine using the time to encode configurations for only two or three robots having six DOF, Reif and Sharir (1994), but did not consider the real time programming. The real time programming was found out using mathematical properties of distance function, leading to the formulation of path planning and optimal control, Gilbert and Johnson (1985), but the solution was limited to offline collision avoidance. The dynamic situation was handled by forming local trajectories for a mobile robot based on the limited knowledge of the environment having obstacles using separating hyper-planes, Tournassound (1986), but limited to six robots.

A world model method was used and updated continuously in real time with range data sampled by the ultra sonic range sensors, leading the mobile robot to travel in densely cluttered obstacles, Borenstein and Koren (1990), but could not detect the unknown obstacle. The method was extended to detect the unknown obstacles to avoid collision while simultaneously steering the mobile robot towards the target. VFH method employs two stages. In the first stage, histogram grid is reduced to 1-D polar histogram with construction around robot's momentary location. In the second stage, an algorithm selects the most suitable sector from among all polar histograms with a low polar obstacle density and steering of the robot to align in the suitable direction, Borenstein and Koren (1991), but the method could not give the solution if the target position is changed.

The planning of the motion of each robot is optimized using visibility graph based on the Genetic Algorithm (GA), where-in each robot acquires knowledge using Fuzzy Logic and the system behaves efficient evolutionary, Shibita and Fukuda (1993), but due to lack of global information, if the robot's position is changed, it may fail. Global information is provided to derive the most feasible velocity and stores the obstacle state for future reference, Tsubouchi and Hirose (1993), but considers only one robot with multiple obstacles.

A car like robot handles unexpected static obstacle while following an optimal path planned by global path planner and tries to minimize the Baye's risk, Huosheng *et al.* (1993), but is limited to only ten robots. Fifteen robots were taken with movable obstacles treated as mobile robot with many DOF, where-in the whole motion planning problem was decoupled into series of tractable problems and are solved using path planning algorithms Classical Path Planning Algorithm (CPPA), Chadzelek *et al.* (1996), and if a robot is added it becomes very tedious computational effort without considering the velocity. The robot chooses velocity commands that satisfy all the constraints and maximizes an objective function that trades off speed, safety and goal directedness using curvature velocity method, Simmons (1996) without taking real world information into account.

Trajectories were rated in terms of collision probability and planning in dynamic environment yielding efficient paths based on the image data processing and compact statistical representation of motion patterns, Kruse and Wahl (1998), but could limit up to four robots. Real world application tasks were efficiently executed in potentially unknown environments by coupling a central mission planner for task definition with dynamic planning techniques, Brumitt and Stentz (1998) with fifteen robots in an unstructured environment without considering the sensors. The global potential field function is modified each time an obstacle is detected by infra red sensors with very low computational burden and algorithm is free from local minima, Lopez and Eliana Aude (2001), but could not consider the change in the position of the obstacle. Artificial Intelligence using Safety Channel Estimation Algorithm (SCEA), shows a smooth and stable path tracking under number of obstacles and intelligent decision can be taken for the obstacle avoidance even if the obstacle's configuration is changed, Bin *et al.* (2004). Since the obstacle density function is just a local information and therefore the effect of the Safety Channel Estimation Algorithm (SCEA) is not so good without the sensors.

A robot can be viewed as perceiving its environment through sensors and acting through actuators. The motion was planned considering coherent groups of robots using a two layered approach, where each coherent group is modeled using boundary element method, Yi Li (2006), but only four robots were used without considering the narrow passages. It was further studied using real time motion planning of multiple agents in a narrow passage avoiding deadlocks, by Yi Li and Kamal Gupta (2006). Narrow passages with moving obstacles and more number of robots in a dynamic environment is solved, Prasanna Lakshmi *et al.* (2007).

METHODOLOLOGY

Recursive Algorithm

This algorithm is introduced for the first time for travelling in the lowest distance alternate path. A Robot starts from its origin so as to reach its specified target position. During its travel if no obstacle is encountered, the robot reaches its target through its Shortest Path (SP) as shown Figure 1. But if any obstacle is faced across, it has to take alternate path. Therefore, a mathematical approach is suggested to identify the alternate paths by introducing Recursive Algorithm. Before that, the shortest path calculation is shown below with Slopes

RiSp = Slope of the path on which the i^{th} Robot is traveling

$RiSp_{mi}$ = Slope of the path on which the i^{th} Robot is traveling to the mth obstacle

$RiSp_{mi}$ = 0 Degrees if the path is parallel to X-axis or 1 when parallel to Y-axis

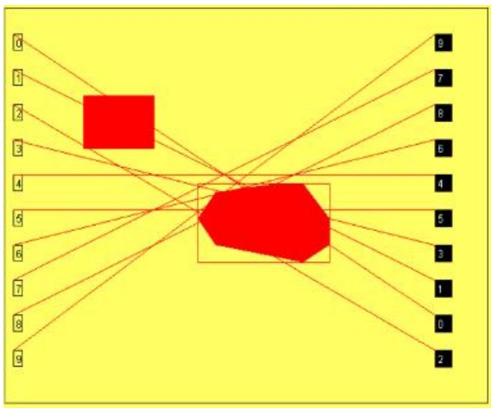
The slope of the path is calculated by using the Formula

$$Distance\ of\ the\ SP = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$= \sqrt{(1000 - 40)^2 + (200 - 800)^2} = 1154$$

$$Slope = \tan^{-1}\frac{y_2 - y_1}{x_2 - x_1} = 34^{\circ}$$

Figure 1: All the Robots with Their Initial and Final Positions



Definition of Recursive Algorithm

It is the path from one obstacle to the next obstacle and if there are m obstacles and the path (q) is an array of q locations containing 1 or 2 values. The total distance of any path is given by the equation:

Total Distance (D) of the of the AP =

$$\left[\overset{n}{\underset{i=1}{Rn_i}}\right] = m_i d0_1 + \sum_{p=1}^{m}\left\{dij_{p1} z\left[path\left(p-1\right)\right] + dij_{p2} z\left(p-2\right)\right\} \quad ...(1)$$

where p is the path and m is the number of obstacles.

Z-Function

$Z(x) = 1 \quad$ If $x = 0$

$= 0 \quad$ If $x \neq 0$

Path Array (3 x 2)

Path No.	1	2
Obstacle (m)		
1	1	2
2	1	2
3	1	2

For three obstacles, $m = 3$

$$D = md0_1 + dij_{1,1}\, z\{path\,[(1)–1] + dij_{1,2}\, z[path\,[(1)–2]\} + dij_{2,1}\, z\{path\,[(2)–1] + dij_{2,2}\, z[path\,[(2)–2]\} + dij_{3,1}\, z\{path\,[(3)–1] + dij_{3,2}\, z[path\,[(3)–2]\} \quad ...(2)$$

$Path[1] = 1$ or 2

$Path[2] = 1$ or 2

$Path[3] = 1$ or 2

$mid0_1$ = Distance from the origin to the first obstacle

$midij$ = Distance of the alternate path from an obstacle m to the next from the i^{th} position to the j^{th} position

$midij_t$ = Distance of the alternate path from an obstacle m to the next from the i^{th} position j^{th} position of the Target

Iteration – 1: [1, 1, 1]

$D1 = m1d0_1 + dij_{1,1}\, z(1–1) + dij_{1,2}\, z(1–2) + dij_{2,1}\, z(1–1) + dij_{2,2}\, z(1–2) + dij_{3,1}\, z(1–1) + dij_{3,2}\, z(1–2)$

$= m1d0_1 + dij_{1,1}\,(1) + dij_{1,2}\,(0) + dij_{2,1}\,(1) + dij_{2,2}\,(0) + dij_{3,1}\,(1) + dij_{3,2}\,(0)$

$= m1d0_1 + [dij_{1,1} + dij_{2,1} + dij_{3,1}]$ ...[3]

Iteration – 2: [1, 1, 2]

$D2 = m1d0_1 + dij_{1,1}\, z(1–1) + dij_{1,2}\, z(1–2) + dij_{2,1}\, z(1–1) + dij_{2,2}\, z(1–2) + dij_{3,1}\, z(2–1) + dij_{3,2}\, z(2–2)$

$= m1d0_1 + dij_{1,1}\,(1) + dij_{1,2}\,(0) + dij_{2,1}\,(1) + dij_{2,2}\,(0) + dij_{3,1}\,(0) + dij_{3,2}\,(1)$

$= m1d0_1 + [dij_{1,1} + dij_{2,1} + dij_{3,2}]$ ...[4]

Similarly the other equations are arrived for other iterations:

Iteration – 3: [1, 2, 1]

$D3 = m1d0_1 + [dij_{1,1} + dij_{2,2} + dij_{3,1}]$ ...[5]

Iteration – 4: [1, 2, 2]

$D4 = 1d0_1 + [dij_{1,1} + dij_{2,2} + dij_{3,2}]$ ...[6]

Iteration – 5: [2, 1, 1]

$D5 = 1d0_1 + [dij_{1,2} + dij_{2,1} + dij_{3,1}]$ ...[7]

Iteration – 6: [2, 1, 2]

$D6 = 1d0_1 + [dij_{1,2} + dij_{2,1} + dij_{3,2}]$ ...[8]

Iteration – 7: [2, 2, 1]

$D7 = 1d0_1 + [dij_{1,2} + dij_{2,2} + dij_{3,1}]$ ...[9]

Iteration – 8: [2, 2, 2]

$D8 = 1d0_1 + [dij_{1,2} + dij_{2,2} + dij_{3,2}]$ ...[10]

D_i = Total distance of each Alternate Path, where i = 1 to 8.

Rn: Robot Number

Coordinates of Intersection of the Robots with Obstacles

The coordinates of the intersection of the robots with the obstacles are given in the following.

mi_{xy} = Coordinates of the i^{th} obstacle with *xy* axes where i = 1 to m obstacles.

Rim_{xy} = Intersection of the i^{th} Robot with m^{th} obstacle with Coordinates *xiyi*.

where i = 1 to n robots m = obstacle and m = 1 to 3

$RiRcj_{xy}$ = Intersection of the collision of i^{th} Robot with Robot *Rj* with coordinates *xiyi*,

where j varies from 2 to n Robots.

$R(n–1)_{xy}$ = Intersection of the collision of (n–1) Robots with *Ri* with coordinates *xiyi*

RESULTS AND DISCUSSION

The suitable Recursive Algorithm is identified of all possible alternate paths to overtake the static obstacles. A heuristic Algorithm for collision detection and avoidance is investigated. Among all the possible paths, the shortest and non collidable path is chosen. Accordingly the path is changed dynamically and the other robots start identifying the collision avoidance with the newly designed path. All the paths and the instantaneous movements of the robots are shown from the Figures 2 to 7 and such paths also shows how the collision is avoided using the dynamic paths generated during the motion. The design and drawings are drawn with AutoCAD.

Figure 2: All the Robots Started Moving on Their Instantaneous Paths (Red Lines)

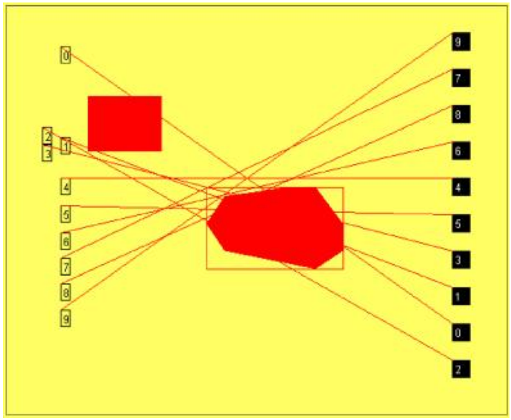


Figure 3: All the Robots have Crossed the 1st Obstacle Without Collision

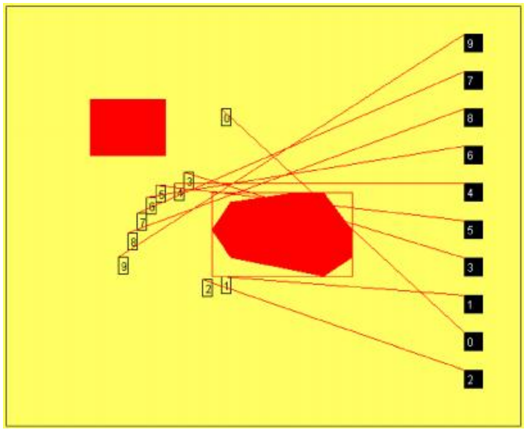


Figure 4: All the Robots have Crossed 2nd Obstacle Without Collision

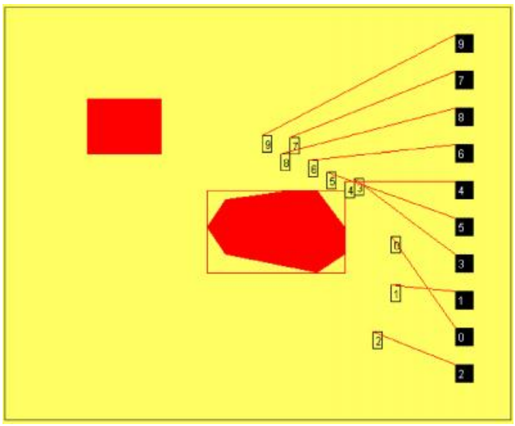


Figure 5: All the Approaching Their Final Positions

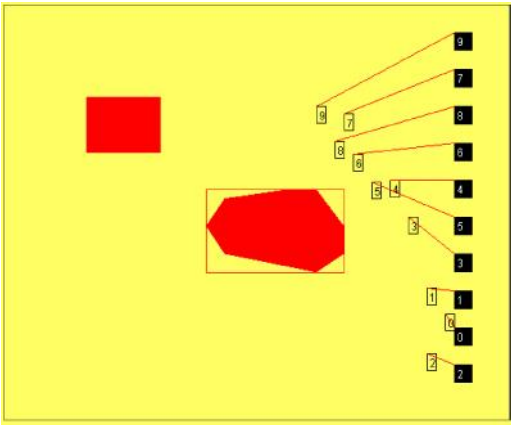


Figure 6: The Robots 0, 1, 2, 3 have Reached Their Targets

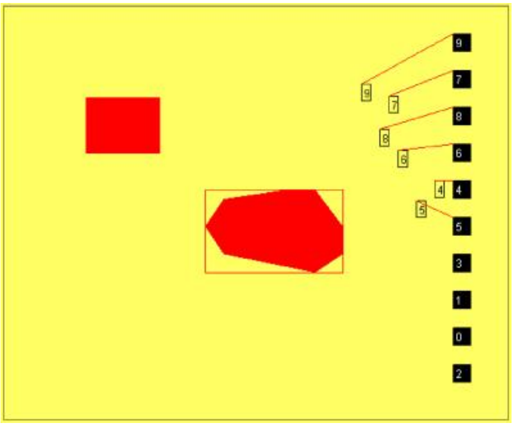
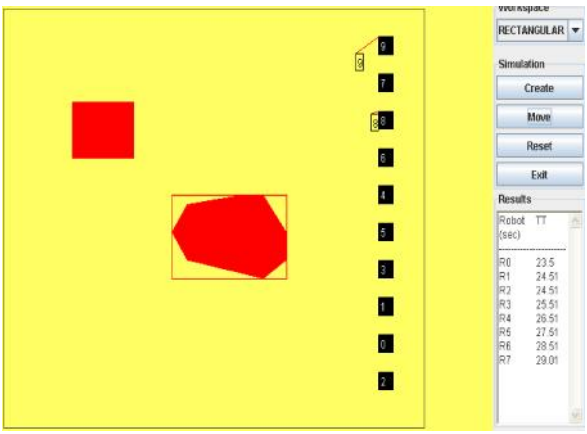


Figure 7: All the Robots Except 8 and 9 have Reached Their Target



CONCLUSION

The 10 robots taken in the workspace are placed initially with its initial and target position. The slope changes to their instant position changing continuously. Since according to Equation (3), the shortest distance is found out using recursive algorithm which is formed due to obstruction created by the obstacles. As a result, only trajectory is changed which is forms the shortest path and thereby the robot reaches its target in the smallest possible time. Back tracking is not considered in the present work, so there is a future scope with two way problem.

REFERENCES

1. Bin H, Hui J, Ke-Zhong H and Gui-Sheng C (2004), "Research on the Strategy of Obstacles Avoidance of Outdoor Mobile Robot", Proceedings on American Control Conference, Vol. 3, June 30-July 2, pp. 2362-2367, Boston, Massachusetts, USA.
2. Borenstein J and Koren Y (1990), "Real Time Obstacle Avoidance for Mobile Robots in Cluttered Environments", *IEEE International Conference on Robotics and Automation*, Vol. 1, May 13-18, pp. 572-577, Cincinnati, OH, USA.
3. Borenstein J and Koren Y (1991), "The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots", *IEEE Trans on Robotics and Automation*, Vol. 7, No. 3, pp. 278-288.
4. Brumitt B L and Stentz A (1998), "GRAMMPS: Generalized Mission Planner for Multiple Mobile Robots in Unstructured Environments", Proceedings

on IEEE International Conference on Robotics and Automation, Vol. 2, May 16-20, pp. 1564-1571, Leuven, Belgium.

5. Chadzelek Thomas, Jens Eckstein and Elmar Schomer (1996), "Heuristic Motion Planning with Movable Obstacles", Proceedings of the 8th Canadian Conference on Computational Geometry, August 12-15, pp. 131-136, Calgary, Canada.
6. Gilbert Elmer G and Daniel Johnson (1985), "Distance Functions and their Applications to Robot Path Planning in the Presence of Obstacles", *IEEE Transactions on Robotics and Automation*, Vol. 1, No. 1, pp. 21-30.
7. Huoseng Hu, Brady M and Probert P (1993), "A Decision Theoretic Approach to Real Time Obstacle for a Mobile Robot", Proceedings on International Conference on Intelligent Robots and Systems, Vol. 2, July 26-30, pp. 1457-1464, Yokohama, Japan.
8. Kruse E and Wahl F M (1998), "Camera Based Observation of Obstacle Motion to Derive Statistical Data for Mobile Robot Motion Planning", Proceedings on International Conference on Robotics and Automation, Vol. 1, May 16-20, pp. 662-667, Leuven, Belgium.
9. Lopez E P and Eliana Aude P L (2001), "Application of a Blind Person Strategy for Obstacle Avoidance with the Use of Potential Fields", Proceedings on International Conference on Robotics and Automation, Vol. 3, May 21-26, pp. 2911-2916, Seoul, Korea.
10. Prasanna Lakshmi K, Eswara Reddy C and Rajagopal K (2007), "Robots with Obstacles", *International Journal on Research and Applied Sciences*, December, pp. 1218-1219.
11. Reif J H and Sharir M (1994), "Motion Planning in the Presence of Moving Obstacles", *Journal of ACM*, Vol. 41, No. 4, pp. 144-154.
12. Shibita T and Fukuda T (1993), "Coordinative Behavior in Evolutionary Multi Agent Robot System", Proceedings on International Conference on Intelligent Robots and Systems, Vol. 1, pp. 448-453, Yokohama, Japan.
13. Simmons Reid (1996), "The Curvature Velocity Method for Local Obstacle Avoidance", Proceedings on International Conference on Robotics and Automation, Vol. 4, April 22-28, pp. 3375-3382, Minneapolis, MN, USA.
14. Tournassoud Pierre (1996), "A Strategy for Obstacle Avoidance and its Application to Multi Robot Systems", Proceedings on International Conference on Robotics and Automation, Vol. 3, April, pp. 1224-1229, Seoul, South Korea.
15. Tsubouchi T and Hirose A (1993), "A Navigation Scheme with Learning for a Mobile Robot Among Multiple Moving Obstacles", Proceedings on International Conference on Intelligent Robots and Systems, Vol. 3, July 26-30, pp. 2234-2240, Yokohama, Japan.
16. Yi Li (2006), "Ph.D. Thesis Proposal – Motion Planning of Multiple Agents in Virtual Environments", August 12.

17. Yi Li and Kamal Gupta (2006), "A Hybrid Two Layered Approach to Real Time Motion Planning of Multiple Agents in Virtual Environment", Proceedings on International Conference on Intelligent Robots and Systems, October, pp. 4362-4367, Beijing, China.