

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,000

Open access books available

125,000

International authors and editors

140M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Hybrid Approach for Global Path Selection & Dynamic Obstacle Avoidance for Mobile Robot Navigation

D. Tamilselvi, S. Mercy Shalinie, M. Hariharasudan and G. Kiruba  
*Department of Computer Science and Engineering,  
Thiagarajar College of Engineering, Madurai, TamilNadu, South India,  
India*

## 1. Introduction

In Robotics, path planning has been an area gaining a major thrust and is being intensively researched nowadays. This planning depends on the environmental conditions they have to operate on. Unlike industrial robots, service robots have to operate in unpredictable and unstructured environments. Such robots are constantly faced with new situations for which there are no pre programmed motions. Thus, these robots have to plan their own motions. Path planning for service robots are much more difficult due to several reasons. First, the planning has to be sensor-based, implying incomplete and inaccurate world models. Second, the real time constraints, provides only limited resources for planning. Third, due to incomplete models of the environment, planning could involve secondary objectives, with the goal to reduce the uncertainty about the environment. Navigation for mobile robots is closely related to sensor-based path planning in 2D, and can be considered as a mature area of research.

Mobile robots navigation in dynamic environments represents still a challenge for real world applications. The robot should be able to reach its goal position, navigating safely amongst, people or vehicles in motion, facing the implicit uncertainty of the surrounding world. Because of the need for highly responsive algorithms, prior research on dynamic planning has focused on re-using information from previous queries across a series of planning iterations. The dynamic path-planning problem consists in finding a suitable plan for each new configuration of the environment by re computing a collision free path using the new information available at each time step.

The problem of collision detection or contact determination between two or more objects is fundamental to computer animation, physical based modeling, molecular modeling, computer simulated environments (e.g. virtual environments) and robot motion planning. In robotics, an essential component of robot motion planning and collision avoidance is a geometric reasoning system which can detect potential contacts and determine the exact collision points between the robot and the obstacles in the workspace.

## 2. Mobile robot indoor environment

Indoor Environment Navigation is the kind of navigation restricted to indoor arenas. Here the environment is generally well structured and map of the part from the robot to the

target is known. Mapping plays a vital role for Mobile Robot Navigation. Mapping the Mobile Robot environment representation is the active research in AI Field for the last two decades for machine intelligence device real time applications in various fields. Creating the spatial model with fine grid cells for physical indoor environment considering the geometric properties is the additional feature compared with the topological mapping. Grid mapping supports the conceptual motion planning for mobile robot navigation and extend the simulation into real time environments.

The robot and obstacle geometry is described in a 2D or 3D *workspace*, while the motion is represented as a path in (possibly higher-dimensional) configuration space. A configuration describes the pose of the robot, and the configuration space  $C$  is the set of all possible configurations. If the robot is a single point (zero-sized) translating in a 2-dimensional plane (the workspace),  $C$  is a plane, and a configuration can be represented using two parameters  $(x, y)$ . The indoor physical environment is represented as two dimensional arrays of cells. Each cell of the map contains two kinds of configuration space depending upon the status of obstacles known as free space and obstacle space.

The term 'path planning' usually refers to the creation of an overall motion plan from a start to a goal location. Most path planners create a path plan in  $C$ -space and then use obstacle avoidance to modify this plan as needed. The idea of using  $C$ -space for motion planning was first introduced by Lozano-Perez and Wesley [1979]. The idea for motion planning in  $C$ -space is to 'grow'  $C$ -space obstacles from physical obstacles while shrinking the mobile robot down to a single point. In the cell decomposition approach to motion planning, the free  $C$ -space in the robot environment is decomposed into disjoint cells which have interiors where planning is simple. The planning process then consists of locating the cells in which the start and goal configurations are and then finding a path between these cells using adjacency relationships between the cells.

There are two types of cell decomposition, exact and approximate. In exact cell decomposition, the union of all the cells corresponds exactly to the free  $C$ -space. Therefore, if a path exists this approach will find it. However, all cells must be computed analytically, which quickly becomes difficult and time consuming for robots with many DOF. Schwartz and Sharir [1983] use exact cell decomposition to study the 'piano mover's' problem. Approximate cell decomposition avoids the difficulty of analytically computing cells by decomposing the free  $C$ -space into many simple cells (often squares or cubes). In addition to easing the decomposition process, computing the adjacency relationships is simplified due to the sameness of all cells. Lozano-Perez used approximate cell decomposition in developing a resolution complete planner for arbitrary,  $n$ -DOF manipulators [1987].

### 3. DT Algorithm for global path planning

The DT algorithm was devised by Rosenfield and Pfaltz as a tool to study the shape of objects in 2D images by propagating distance in tessellated space from the boundaries of shapes into their centers. Various properties of shape can be extracted from the resultant transform and it can be shown that the skeletons of local maxima can be used to grow back the original shapes without information loss. Jarvis discovered that, by turning the algorithm 'inside out' to propagate distance from goals in the free space that includes the shapes interpreted as obstacles and by repeating a raster order and inverse raster order scan used in the algorithm until no further change takes place, a space filling transform with direct path planning application is resulted. Multiple starting points, multiple goals and multidimensional space

versions are easily devised. When the DT completes filling the free space with distance markers, the goal is achieved from any starting point through a steepest descent trajectory.

Using distance transforms for planning paths for mobile robot applications was first reported by R.A. Jarvis and J.C. Byrne. This approach considered the task of path planning to find paths from the goal location back to the start location. The path planner propagates a distance wave front through all free space grid cells in the environment from the goal cell. The distance wave front flows around obstacles and eventually through all free space in the environment. For any starting point within the environment representing the initial position of the mobile robot, the shortest path to the goal is traced by walking down hill via the steepest descent path. If there is no downhill path, and the start cell is on a plateau then it can be concluded that no path exists from the start cell to the goal cell i.e. the goal is unreachable. Global path planner finds the optimized path in the occupancy grid based environment.

DT (Distance Transform) method proved to be one of the best global path selection and effective way of path planning in the static known environment. DT methodology is versatile and can be used for path planning alone or integration of path planning with obstacle avoidance. To predict the dynamic obstacles in the environment, avoid collision with the mobile robot, GJK (Gilbert-Johnson-Keerthi) distance algorithm supports for collision avoidance of obstacles in the dynamic environment combined with DT Algorithm.

#### **4. Collision Detection and Avoidance in mobile robots – an algorithmic approach**

Collision Detection and Avoidance plays a vital role in mobile robot applications. Algorithms are used to achieve this. This Collision Detection and Avoidance is also employed in various other applications like simulated computer games, unmanned vehicle guidance in military based applications, etc. In these applications the Collision detection strategy is achieved by checking whether the objects overlap in space or if their boundaries intersect each other during their movement.

The obstacle avoidance problem for robotics can be divided into three major areas. These are mapping the world, determining distances between manipulators and other objects in the world, and deciding how to move a given manipulator such that it best avoids contact with other objects in the world. Unless distances are determined directly from the physical world using range-finding hardware, they are calculated from the world model that is stored during the world mapping process. These calculations are largely based on the types of objects that are used to model the world. One of the most popular ways of modelling the world is to use polyhedral [Canny, 1988] [Gilbert, Johnson, and Keerthi, 1988] [Lin and Canny, 1991] [Mirtich, 1997]. Models built using polyhedral are capable of providing nearly exact models of the world. Unlike many other distance algorithms, it does not require the geometry data to be stored in any specific format, but instead relies solely on a support mapping function and using the result in its next iteration. Algorithm stability, speed and small storage makes GJK suitable for real time collision detection.(eg., Physics Engine for Video Games – sony play stations).

GJK supports mappings for reading the geometry of an object. A support mapping fully describes the geometry of a convex object and can be viewed as an implicit representation of the object. The class of objects recursively constructed are

- Convex primitives

- Polytopes (line segments, triangles, boxes and other convex polyhedra)
- Quadrics (spheres, cones and cylinders)
- Images of convex objects under affine transformation
- Minkowski sums of two convex objects
- Convex hulls of a collection of convex objects
- Convex Polyhedra and Collision Detection

Convex polyhedra have been studied extensively in the context of the minimum distance problem. The reason is that the minimum distance problem in this case can be cast as a linear programming problem, allowing us to use well-known results from convex optimization. The algorithms for convex polyhedra fall into two main categories: simplex-based and feature-based. The simplex-based algorithms treat the polyhedron as the convex hull of a point set and perform operations on simplexes defined by subsets of these points. Feature-based algorithms, on the other hand, treat the polyhedron as a set of features, where the different features are the vertices, edges and faces of the polyhedron.

#### 4.1 GJK (gilebert-Johnson-Keerthi) algorithm – simplex based

One of the most well-known simplex-based algorithms is the Gilbert- Johnson-Keerthi (GJK) algorithm. Conceptually, the GJK algorithm works with the Minkowski difference of the two convex polyhedra. The Minkowski difference is also a convex polyhedron and the minimum distance problem is reduced to find the point in that polyhedron that is closest to the origin; if the polyhedron includes the origin, then the two polyhedra intersect. However, forming the Minkowski difference explicitly would be a costly approach. Instead GJK work iteratively with small subsets, simplexes, of the Minkowski difference that quickly converge to a subset that contains the point closest to the origin. For problems involving continuous motion, the temporal coherence between two time instants can be exploited by initializing the algorithm with the final simplex from the previous distance computation.

Unlike many other distance algorithms, it does not require the geometry data to be stored in any specific format, but instead relies solely on a support mapping function and using the result in its next iteration. Relying on the support mapping function to read the geometry of the objects gives this algorithm great advantage, because every enhancement on the support mapping function leads to enhancement of the GJK algorithm.

## 5. Proposed methodology

Fig.1. shows the hybrid method proposed for the Global Path Selection and Dynamic Obstacle Avoidance. Path planning was based on the distance criteria for the simulation, and sensor values mapping for the FIRE BIRD V mobile robot, that was used for the real time study.

Distance Transform (DT) was generated through free space from the goal location using a cost function [17]. Path transform for the point  $c$  is calculated with the formula.1

$$\text{Cost function} = \min_{p \in P} \left[ \text{length}(p) + \sum_{c_i \in P} \alpha \text{ obstacle}(c_i) \right] \quad (1)$$

Where  $P$  is the set of all possible paths from the point  $c$  to the goal, and  $p \in P$  i.e. a single path to the goal. The function  $\text{length}(p)$  returns the Euclidean distance between the source and the

goal through the path  $p$ . The function  $obstacle(c)$  is a cost function generated using the values of the obstacle transform. It represents the degree of discomfort the nearest obstacle exerts on a point  $c$ . The weight  $\alpha$  is a constant  $\geq 0$  which determines how strongly the path transform will avoid obstacles. Local cost functions return the instantaneous cost for traversing a particular patch. Global Cost functions provide the estimated cost to reach a goal from a particular location.

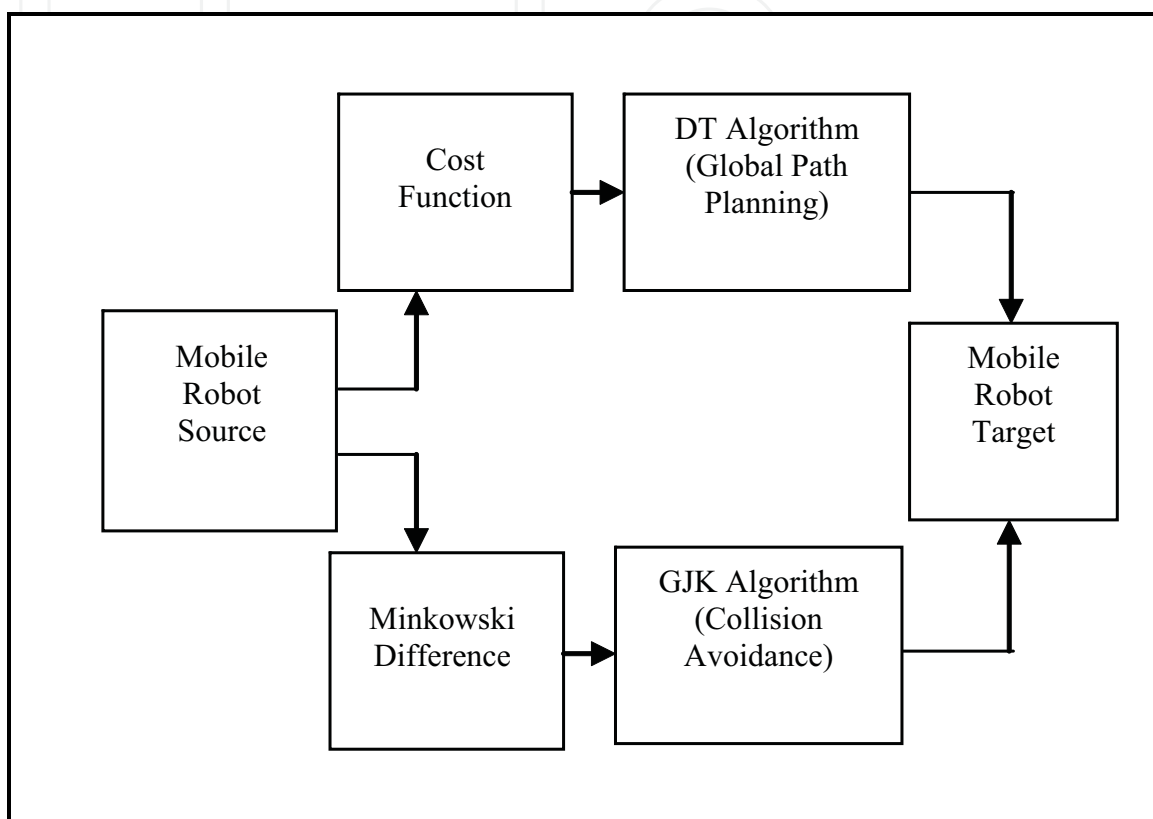


Fig. 1. Hybrid Method for Global Path Selection & Dynamic Obstacle Avoidance

GJK algorithm, involves the computationally intensive step of computing the supporting function of the set of vertices, is a set-theoretic approach in essence. Since in applications the distance between two objects needs to be updated from time to time, every possible enhancement of distance computation procedure can speed up the repetitive process as the time goes on[18].GJK key concept is instead of computing the minimum distance between A and B, the minimum distance between A-B and the origin is computed. This algorithm uses the fact that the shortest distance between two convex polygons is the shortest distance between their Minkowski Difference and the origin. If the Minkowski Difference encompasses the origin then the two objects intersect. Computing the collision translation of two convex bodies can be reduced to computing collision translations of pairs of planar sections and minimizing a bivariate convex function.

So this approach employs the concept of DT algorithm which returns the least Euclidean distant path  $p$  to the goal, then triggers the GJK algorithm to anticipate the possibility of a local collision through that path, for all the possible moves of the obstacle in the next instant. Accordingly the model employs the path which provides a least distant point which returns a 0 possibility of collision with the obstacle.

### 5.1 Flow chart representation for the proposed methodology

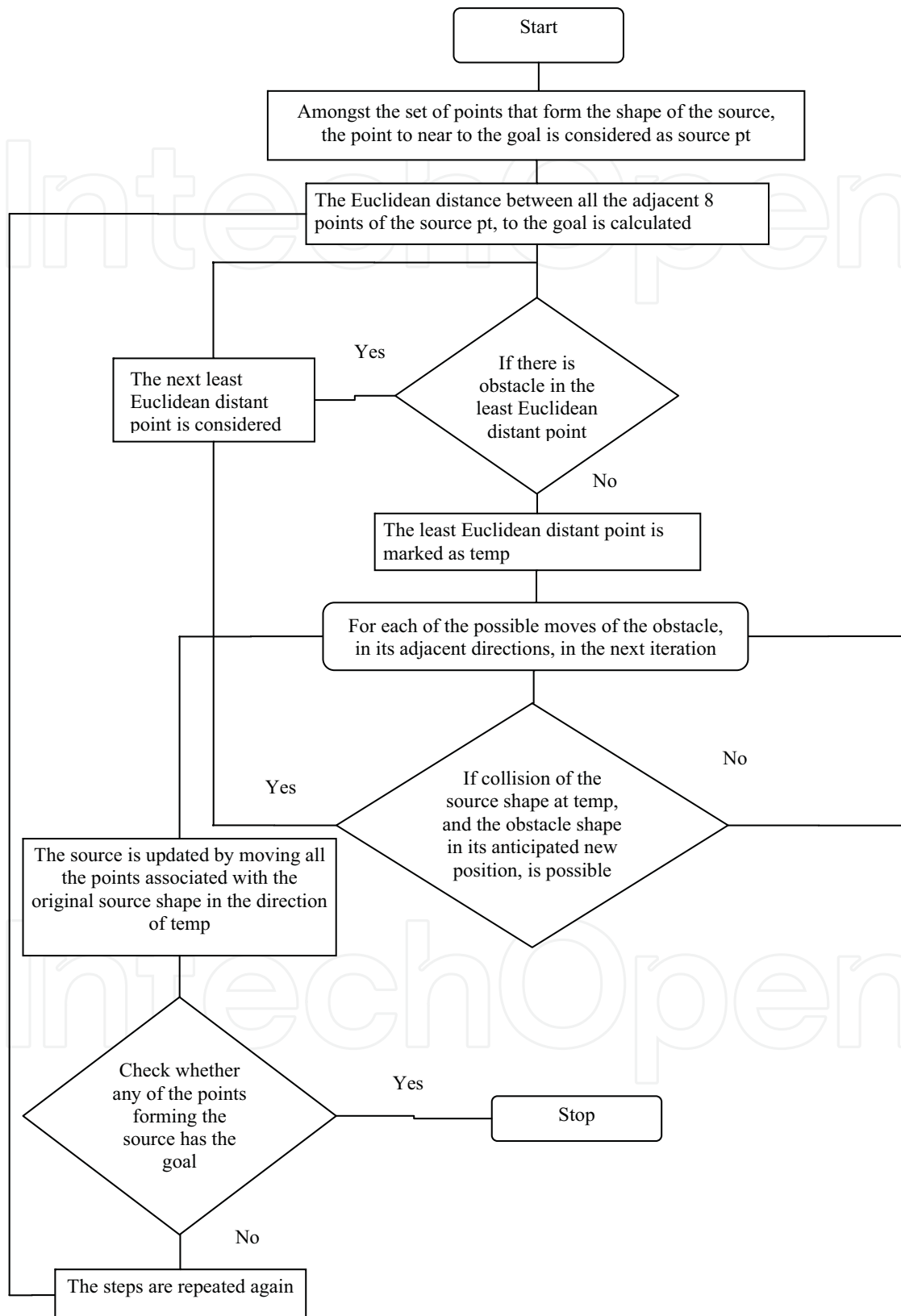


Fig. 2. Flowchart for the Proposed Method

The assumptions made in the simulation and real time environment are, the source and obstacle moves with the same speed of, one grid point per iteration. One dynamic obstacle is taken into consideration. Mobile Robot shape is assumed as triangle and square. The goal is fixed with green. Robot scans the 8 adjacent directions in the grid environment, to find the next adjacent position that promises the minimum Euclidean distance to the goal. From the calculated new position, it checks for all possible next iteration-movements of the obstacle in all its 8 adjacent directions and checks if collision would occur using GJK algorithm.

If not, the mobile robot moves to the calculated new position. If the mobile robot predicts the collision possibility, it repeats the above steps. GJK supports for the dynamic obstacle avoidance by calculating the Minkowski difference between the mobile robot and obstacle for every step movement in the grid environment. The simplex formula calculates the next step co ordinate values, and the zero value predicts the collision occurrences. Repeating the calculation of each Minkowski difference supports for prediction of collision and avoidance in the dynamic environment. The following example briefs about the calculation of distance values for dynamic obstacle prediction and avoidance.

Eg. Calculation for Minkowski difference for the Triangle Obstacle to predict and avoid the collision was given below.

Lets consider the source to be of a triangle shape at the coordinates (1,2), (0,3), (2,3) and the obstacle which is here taken as a quadrilateral, initially occupying the coordinates (3,2), (4,1), (3,4), (5,3). The goal is set at the point (8,3). Now the DT algorithm returns the coordinates (2,2), (3,3) and (1,3) as the new set of points for the source, returning a minimum Euclidean distance to the goal. Now the GJK algorithm detects that if the obstacle moves to the position (2,2), (3,1), (4,3), (2,4) then the new set of minkowski points would be

$$(0,0), (-1,1), (-2,-1), (0,-2), (1,1), (0,-2), (-1,0), (1,-1), (-1,1), (-2,-2), (-3,0), (-1,-1)$$

The shape that encloses these points would also enclose the origin. So a collision is possible with the obstacle if the source moves to that point. So it takes the next least Euclidean distant point and proceeds on till it finds a new position for the source which is devoid of any collision. If there is a possibility of collision in all the possible positions, then the source remains fixed to its current position for the next iteration also.

## 6. Simulation results

Grid environment (size 10x10) was created with the triangle shaped mobile robot and convex shaped obstacle. The dynamic obstacle movements were shown in Fig.3. Top rightmost green colour indicates the goal position. Mobile robot predicts movement using GJK and reaches the goal with DT.

The time taken for the mobile robot to reach the target was 0.32 seconds for Fig.3, which shows GJK predicts the obstacle by considering only the neighboring coordinates values and moves in the proposed environment. DT calculates the goal position in prior and checks for the optimal distance value to reach the goal.

Fig.4 shows the square shaped mobile robot with the same convex obstacle. Due to the dynamic nature of obstacle, mobile robot moving near the obstacle is not possible often. In this case, the time taken for the mobile robot to reach the goal is 0.29 seconds.



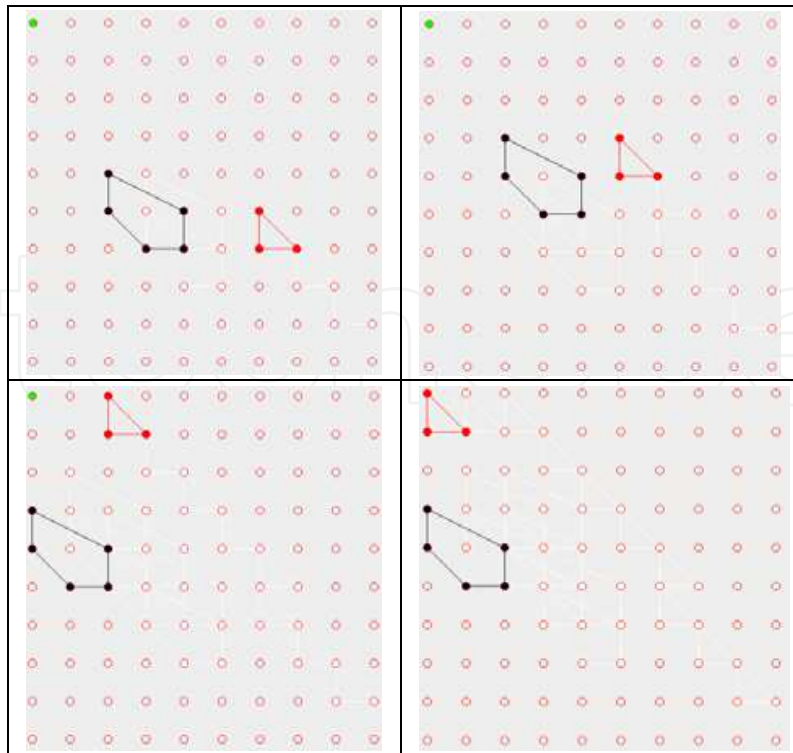


Fig. 3. Dynamic Obstacle Avoidance – Simulation Results

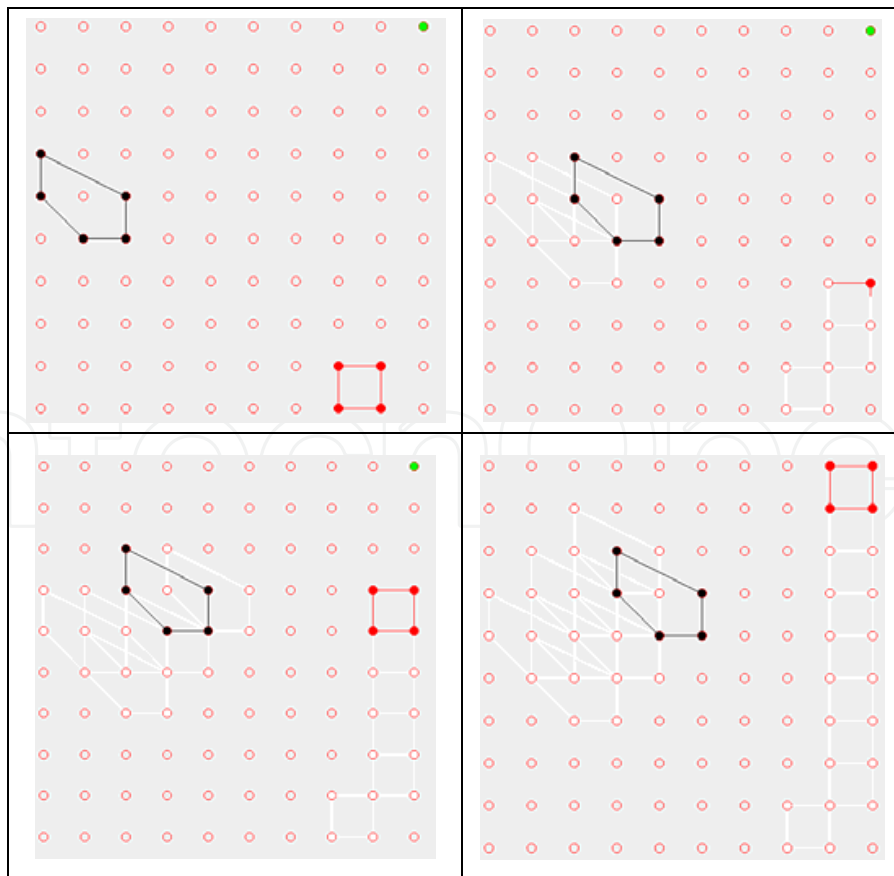


Fig. 4. Dynamic Obstacle Avoidance – Simulation Results

## 7. Real time results

The GJK was implemented in real time with Fire Bird V mobile robot. FIRE BIRD V robot has two DC geared motors for the locomotion. The robot has a top speed of 24cm/second. These motors are arranged in differential drive configuration. I.E. motors can move independently of each other. Front castor wheel provides support at the front side of the robot. Using this configuration, the robot can turn with zero turning radius by rotating one wheel in clockwise direction and other in counterclockwise direction. Position encoder discs are mounted on both the motor's axle to give a position feedback to the microcontroller. Fig.5 shows the Fire Bird V mobile robot with the IR Sensors to detect obstacles in all sides.

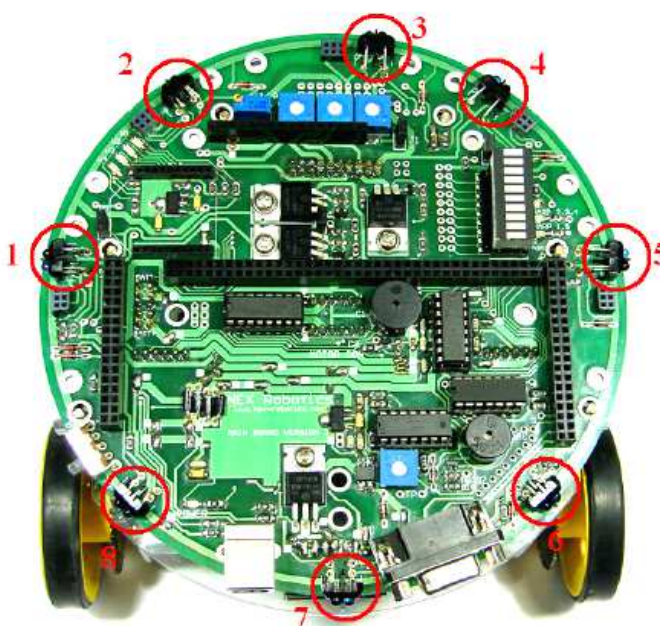


Fig. 5. Eight IR Sensors in Fire Bird V Mobile Robot

Instead of coordinates, mapping was created in real time with sensors. As GJK finds the distance between the objects using Minkowski Difference, we use the IR Range and IR Proximity Sensors in conjunction to detect the distance from the approaching object. The threshold values of each sensor are calculated already such that a collision does not occur. So if the threshold value is reached, it is an indication that a collision is about to happen in the near future. Sensor values are used for real time obstacle avoidance.

The distance mapping was created with sensor mapping values. Sharp IR range sensors consists of 2 parts - narrow IR beam for illumination and CCD array, which uses triangulation to measure the distance from any obstacle. A small linear CCD array is used for angle measurement. The IR beam generates light. The light hits the obstacle and reflects back to the linear CCD array. Depending on the distance from the obstacle, angle of the reflected light varies. This angle is measured using the CCD array to estimate distance from the obstacle. When away from obstacles, the sensor values are smaller. An obstacle can be a wall, any moving object. Near an obstacle or vice-versa, the sensor values increases. Table 1 shows the mapping of IR sensor values observed from the Fire Bird V Mobile Robot for obstacle prediction and avoidance.

Obstacle Position	IR1	IR2	IR3	IR4	IR5	IR6	IR7	IR8
No obstacle along any side	233	235	242	242	245	252	252	253
Obstacle at front	233	233	229	241	244	251	252	253
Obstacle at front and left	228	232	229	240	244	249	250	252
Obstacle at front and right	231	232	229	241	242	249	250	251
Obstacle at left only	229	235	240	241	245	251	252	253
Obstacle at right only	232	232	236	240	239	249	251	250
Obstacle at front, left and right	225	233	231	240	239	249	251	250
Obstacle at back	232	232	236	242	245	250	246	252

Table 1. Sensor Mapping Values for the Indoor Real Time Environment (80 cm x 75 cm)

The difference is distance values are mapped with IR sensor values and it predicts the behavior of mobile robot to avoid collision with obstacle. Rectangle and convex shapes obstacles were included in the dynamic environment. GJK predicts obstacle movement in the environment and plans the mobile robot for path selection. DT supports for the mobile robot to reach the target position. The sensor prediction for various obstacle position in real time and avoidance was described below.

- **Obstacle at Front**  
Initially when the robot is kept in open space with no obstacles in range, all the IR sensor values are high. It is then instructed to move forward, whereby it might sense obstacles at the front using sensor IR3. The threshold value for IR3 is 229 i.e. by GJK algorithm, the shortest possible distance that the robot can traverse forward is until IR3 value reaches 229. After this value has been reached, the robot has to take a turn to left or right to avoid the obstacle.
- **Obstacle at two sides**  
Consider the robot meets a corner i.e. obstacles at front and left (or) at front and right. Let's take the first case: obstacle at front and left. In this case, sensor values of IR3 and IR1 reduces on approaching the obstacle. These values are checked whether they meet the combinational threshold. If so, the robot is made to turn towards the free side, here it is the right side. The same is followed for the latter case where sensors IR3 and IR5 are taken into account. Here the robot makes a turn towards the left on meeting the threshold.  
The threshold values are as below:  
IR1: 228 IR3: 229  
IR5: 242 IR3: 229
- **Obstacle at back**  
This special case is taken into consideration only in a dynamic environment because in a static environment, any objects at the back do not account as obstacles. Here IR6, IR7, and IR8 help in detecting the obstacles. For activating these sensors, master-slave (microcontrollers) communication needs to be established. When an obstacle is sensed at the back, the robot needs to move forward with a greater velocity to avoid collision.  
The threshold value for IR7: 246.

White line sensors with the Fire Bird V mobile robot finds the target, localize itself and stop the navigation. White line sensors are used for detecting white line on the ground surface.

White line sensor consists of a highly directional phototransistor for line sensing and red LED for illumination.

Fig. 6 & Fig.7 shows Fire Bird V mobile robot path selection results with convex shape obstacles in the proposed real time environment.

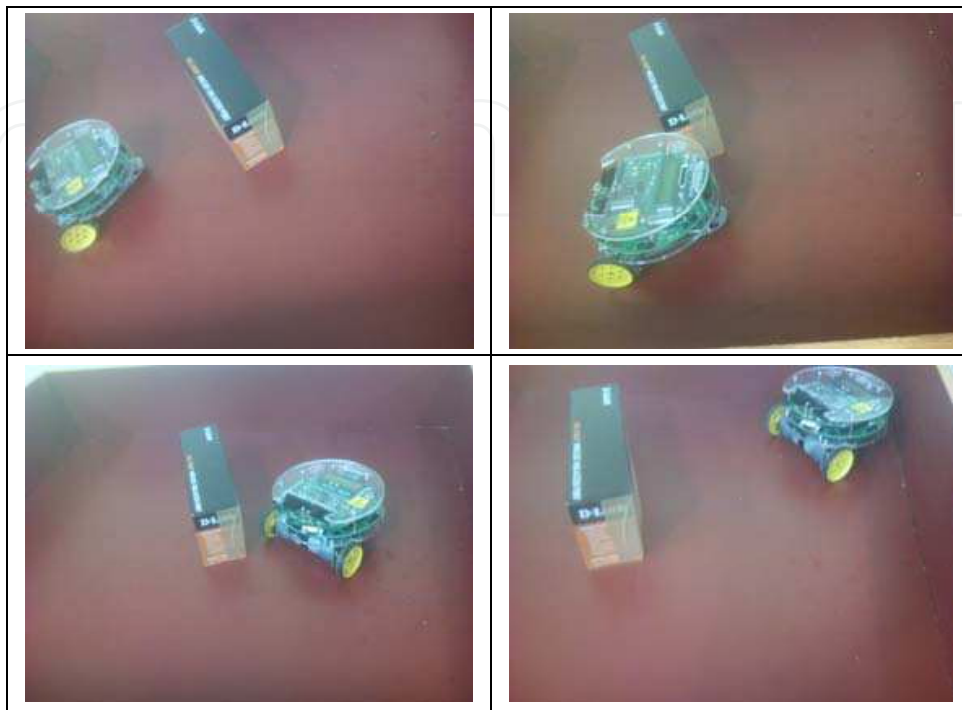


Fig. 6. Fire Bird V Mobile Robot Real Time Navigation Results

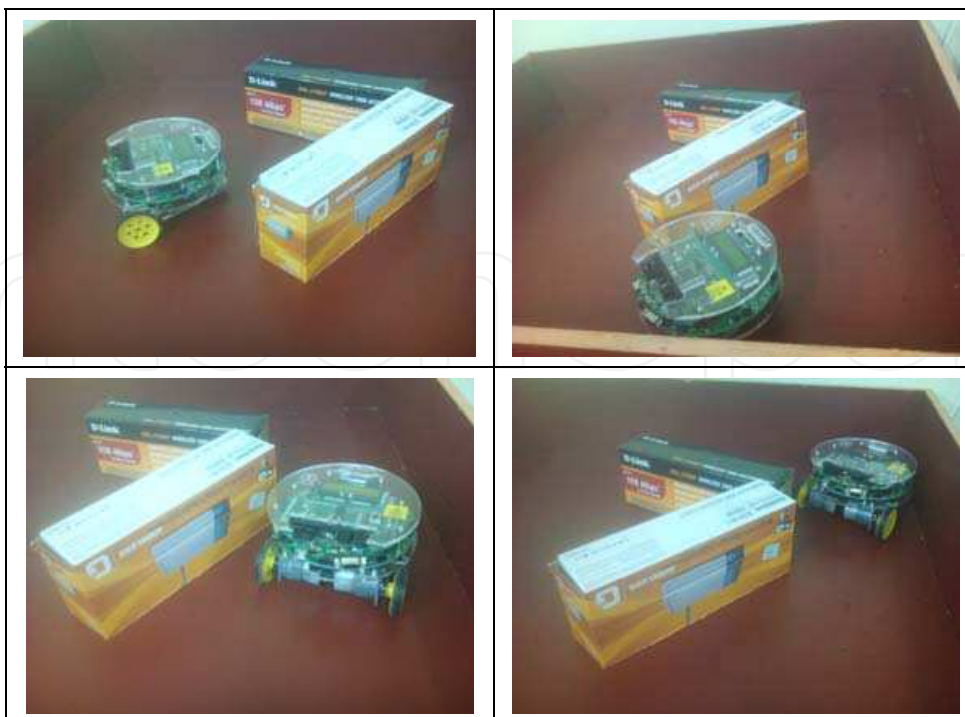


Fig. 7. Fire Bird V Mobile Robot Real Time Navigation Results

## 8. Conclusion

Proposed method builds the test bed with Fire Bird V Mobile Robot for future development of an intelligent wheelchair for elderly people assistance in the indoor environment. Real time results proves, DT computes the shortest path selection for the elderly people and movement of people/any object was predicted in prior by DT and collision avoidance with objects was effectively handled by GJK Algorithm. The proposed framework in real time with Fire Bird V Mobile Robot control program is easily scalable and portable for any indoor environment with various shapes of obstacles it encounters during navigation. Proposed method was tested successfully with various shapes of dynamic obstacles.

GJK Algorithm for computing the distance between objects shows the improved performance in simulation and real time results. The algorithm fits well especially for the collision detection of objects modelled with various types of geometric primitives such as convex obstacles. Unlike many other distance algorithms, it does not require the geometry data to be stored in any specific format, but instead relies solely on a support mapping function and using the result in its next iteration. Relying on the support mapping function to read the geometry of the objects gives this algorithm great advantage. The algorithm's stability, speed, and small storage footprint make it popular for real time collision detection.

## 9. Acknowledgement

This work is part of Research grant sponsored by Computer Society of India , Project titled" Intelligent Algorithmic Approach for Mobile Robot Navigation in an Indoor Environment" , File No. 1-14/2010-03 dated 29.03.2010.

## 10. References

- [1] F. Belkhouche, B. Belkhouche, Kinematics-Based Characterization of the Collision Course. *International Journal of Robotics and Automation*, Vol. 23, No. 2, pp 1-10, 2008
- [2] J.P. Laumond S. Sekhavat F. Lamiriaux, Guidelines in Nonholonomic Motion Planning for Mobile Robots, *Lectures Notes in Control and Information Sciences Springer*, pp 343-347, ISBN 3-540-76219-1, 1998.
- [3] Marcelo Becker, Carolina Meirelles Dantas, Weber Perdigão Macedo, Obstacle Avoidance Procedure for Mobile Robots, *ABCM Symposium on Mechatronics*, Vol.2,pp.250-257, 2006
- [4] Andrew Price, Artificially Intelligent Autonomous Aircraft Navigation System Using a DT on an FPGA. *NIOS II, Embedded Processor Design Contest, Star Award Paper*, pp 1-14, 2006
- [5] Christor Ericson, The GJK Algorithm,"The GJK Algorithm" *31<sup>st</sup> International Conference on Computer Graphics and Interactive Techniques*, pp1-12, 2004
- [6] Davor Jovanoski. The Gilbert Johnson Keerthi Algorithm, Department of Computer Science, Bachelor Seminar, University of Salzburg, February 2008
- [7] K. J. Kyriakopoulos ,G. N. Saridis, Distance estimation and collision prediction for on-line robotic motion planning: NASA Center of Intelligent Robotic Systems for Space Exploration (CIRSSE), Electrical, Computer and Systems Engineering, Dept., February 2003

- [8] A.Zelinsky, Using Path Transforms to Guide the Search for Find path in 2D, *International Journal of Robotics Research*, Vol.13, No.4, pp 315-325, 1994
- [9] Sardjono Trihatmo, R.A. Jarvis, Short-Safe Compromise Path for Mobile Robot Navigation in a dynamic environment, *International Journal of Intelligent Systems*, Vol.20, pp 523-539, 2005
- [10] O.Hachour, Path planning of Autonomous Mobile robot, *International Journal of Systems Applications, Engineering & Development* , Issue 4, Volume 2, pp178 - 190, 2008
- [11] Morten Strandberg, Robot Path Planning: An Object-Oriented Approach, Automatic Control, Department of Signals, Sensors and Systems, Royal Institute of Technology (KTH), Ph.D thesis, 2004
- [12] Enrique J. Bernabeu, Josep Tornero, Masayoshi Tomizuka, Collision Prediction and Avoidance Amidst Moving Objects for Trajectory Planning Applications, *Proceedings of the 2001 IEEE International Conference on Robotics & Automation Seoul, Korea*, pp 3801-3806, 2001.
- [13] Davor Jovanoski, The Gilbert - Johnson - Keerthi (GJK) Algorithm, Department of Computer Science, University of Salzburg, 2008
- [14] Kaushik J, Vision Based Mobile Robot Navigation, Undergraduate Seminar Report, Department of Mechanical Engineering Indian Institute of Technology, Mumbai, 2007
- [15] Troy Anthony Harden, Minimum Distance Influence Coefficients for Obstacle Avoidance in Manipulator Motion Planning, University of Texas, Ph.D Thesis, 2002
- [16] Gino van den bergen, A Fast and Robust GJK Implementation for Collision Detection of Convex Objects, Department of Mathematics and Computing Science, Eindhoven University of Technology, Netherlands, 1999
- [17] A. Zelinsky, R.A. Jarvis, Byrne and S. Yuta, Planning Paths of Complete Coverage of an Unstructured Environment by a Mobile Robot, *Science Direct, Robotics and Autonomous Systems*, Vol.46, Issue 4, pp 195 - 204, 2004
- [18] Jing-Sin Liu, Yuh-ren Chien, Shen-Po Shiang and Wan-Chi Lee ,Geometric Interpretation and Comparisons of Enhancements of GJK Algorithm for Computing Euclidean Distance Between Convex Polyhedra, Institute of Information Science, Academia Sinica, Taiwan, 2001
- [19] Ray Jarvis, Distance Transform Based Visibility Measures for Covert Path Planning in known but Dynamic Environments, *International Conference on Autonomous Robots and Agents*, pp 396 - 400, 2004
- [20] Sardjono Trihatmo, R.A. Jarvis, Short-Safe Compromise Path for Mobile Robot Navigation in a dynamic environment, *International Journal of Intelligent Systems*, Vol.20, pp 523-539, 2005
- [21] Claudio Mirolo, Stefano Carpin, Enrico Pagello, Incremental Convex Minimization for Computing Collision Translations of Convex Polyhedra, *IEEE Transactions on Robotics*, Volume 23, Issue 3, pp 403 - 415, 2007
- [22] Madhur Ambastha, Dídac Busquets, Ramon López de Màntaras, Carles Sierra, Evolving a Multiagent System for Landmark-Based Robot Navigation, *International Journal of Intelligent Systems*, Vol. 20, pp 523-539, 2005

- [23] Stoytchev, A., Arkin, R., Incorporating Motivation in a Hybrid Robot Architecture, *Journal of Advanced Computational Intelligence and intelligent informatics* Vol.8,No.3, pp 100-105, 2004
- [24] Jonas Buchli, Mobile Robots Moving Intelligence, *Advanced Robotic Systems* International and pro Literature Verlag, Germany, ISBN 3-86611-284-X, 2006

IntechOpen

IntechOpen



## **Advances in Robot Navigation**

Edited by Prof. Alejandra Barrera

ISBN 978-953-307-346-0

Hard cover, 238 pages

**Publisher** InTech

**Published online** 15, June, 2011

**Published in print edition** June, 2011

Robot navigation includes different interrelated activities such as perception - obtaining and interpreting sensory information; exploration - the strategy that guides the robot to select the next direction to go; mapping - the construction of a spatial representation by using the sensory information perceived; localization - the strategy to estimate the robot position within the spatial map; path planning - the strategy to find a path towards a goal location being optimal or not; and path execution, where motor actions are determined and adapted to environmental changes. This book integrates results from the research work of authors all over the world, addressing the abovementioned activities and analyzing the critical implications of dealing with dynamic environments. Different solutions providing adaptive navigation are taken from nature inspiration, and diverse applications are described in the context of an important field of study: social robotics.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

D. Tamilselvi, S. Mercy Shalinie, M. Hariharasudan and G. Kiruba (2011). Hybrid Approach for Global Path Selection & Dynamic Obstacle Avoidance for Mobile Robot Navigation, *Advances in Robot Navigation*, Prof. Alejandra Barrera (Ed.), ISBN: 978-953-307-346-0, InTech, Available from:

<http://www.intechopen.com/books/advances-in-robot-navigation/hybrid-approach-for-global-path-selection-dynamic-obstacle-avoidance-for-mobile-robot-navigation>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821



© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen