Australian Journal of Basic and Applied Sciences, 9(36) December 2015, Pages: 592-596

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Tangent Bug Algorithm Based Path Planning of Mobile Robot for Material Distribution Task in Manufacturing Environment through Image Processing

[1]M. Malathi, [2]K. Ramar and [3]C. Paramasivam

[1]*Asst. Professor, Department of Computer Science and Engineering, Vaigai College of Engineering, Madurai-625122, Tamil Nadu, India.*
[2]*Professor, EinsteinCollege of Engineering, Tirunelveli-627 012, Tamil Nadu, India.*
[3]*Asst. Professor, Department of Mechanical Engineering, Thiagarajar College of Engineering, Madurai-625 015, Tamilnadu, India.*

ARTICLE INFO

Article history:
Received 10 November 2015
Accepted 30 December 2015
Available online 18 January 2016

Keywords:
Obstacle avoidance, Path planning, Tangent bug algorithm and open field layout and Product mix.

ABSTRACT

Path planning is one of the important tasks in mobile robot to complete a specified task. This paper deals with mobile robot path planning in a static environment with obstacle avoidance for open field layout production environment with tangent bug algorithm through image processing. Here, a Mat lab code has been written to process the image of the working environment for identifying the different elements present within the working environment, which are to be considered as obstacles as for as mobile robot path planning is concerned. The tangent bug algorithm is proposed and implemented to get shortest path for material distribution task by the mobile robot. Through this algorithm a collision-free path is obtained with a scope to implement in real factory environment. Depending on the product mix sequence, a collision free path can be obtained and same may be used for mobile robot to complete the material distribution task.

To Cite This Article: M. Malathi, K. Ramar and C. Paramasivam., Tangent Bug Algorithm Based Path Planning of Mobile Robot for Material Distribution Task in Manufacturing Environment through Image Processing. ***Aust. J. Basic & Appl. Sci.,*** *9(36): 592-596, 2015*

INTRODUCTION

Many research works are going on to track the static or moving target in a specified environment by utilizing the sensors and image processing (Aji S.S and Tripty Singh, 2014; ShahedShojaeipour, 2014). However in the majority of surveillance and image tracking system, the sensors are stationary. The stationary system require the desired object to locate within the surveillance range of the system (Amol N. 2013). Integrating the image processing with robotic system, it contributes the improved adaptability of the system and it opens the door to other potential applications like material distribution, automated inspection, surveillance vehicle (Abhishek Chandak, 2013; Roland Siegwart and Illah R. Nourbakhsh, 2004). Path planning is the major issue in solving mazes for mobile robot with dynamic obstacle. In the field of space exploration and fire fighting autonomous robots are generally based on image navigation from a start to goal point in known or unknown environment (Hachour, O., 2008; HimanshuBorse, 2012). The efficiency of production system is depends on how well the various machines and other production facilities are located in a plant. In flexible manufacturing system the different layouts are utilized to improve the productivity and effectiveness of the resources. In particular, the open field layout, it is more difficult to use conveyors for the distribution of materials due to dynamic changes in machine routing. Hence, the mobile robots may be a viable alternative to distribute the materials or parts in a open field manufacturing environment.

Concepts Bug algorithm:

Generally any heuristic algorithm, assume global knowledge about the environment, but bug algorithms uses only local knowledge about the environment to reach the goal within the environment. In bug1 algorithm, the mobile robot move towards the goal directly, using sensors, the mobile robot takes the unknown obstacles during motion to the goal point. Once it encounters the obstacle it goes around the obstacle in clockwise direction and then determines the escape point from the periphery of the obstacle by calculating the distance between the current position of the mobile robot to goal point. The escape point is the closest point around the obstacle with reference to goal. This method is inefficient, but guarantees that the mobile robot is able to reach the goal. The bug2 algorithm is greedy search algorithm that the mobile robot follows a constant slope computed initially between start to goal. Generally, it provides the shortest

Corresponding Author: M. Malathi Asst. Professor, Department of Computer Science and Engineering, Vaigai College of Engineering, Madurai-625122, Tamil Nadu, India
E-mail: mmcse1979@gmail.com

distance to reach the goal compared to bug1 algorithm. The tangent bug algorithm is derivative of the bug2 algorithm, initially the slope from mobile robot to obstacle is created and mobile robot follows the slope, if there is any obstacle along the path then create a tangent line with respect to obstacle and follows the slope, finally reach the goal. The distance calculation in tangent bug algorithm is easier than the Bug2 algorithm.

Nomenclatures:

Symbol	-	Item description
θ_g	-	Angle between mobile robot to goal in degree
(x_g, y_g)	-	Coordinate points of the goal in cm
(x_{i+1}, y_{i+1})	-	Coordinate points of the next position of mobile robot in cm
d_{reach}	-	Distance between the initial to goal within the working environment in cm
θ_f	-	Angle between mobile robot to obstacle in degree
(x_u, y_u)	-	Coordinate point for updated position of mobile robot in cm
(x_i, y_i)	-	Coordinate point for current position of mobile robot in cm
(x_{i+1}, y_{i+1})		Coordinate point for next position of mobile robot in cm
d_{follow}		Distance between the robot to obstacle boundary in cm

Related Work:

Abishek Chandak *et al*. (2013) have proposed wave front algorithm to find the shortest path for mobile robot in indoor environment with obstacle avoidance. Taking the image of the environment, convert the image into binary form and set 1 as obstacle and 0 as surrounding then perform the morphological operation on the image to remove unwanted information. Maintain the robot move some distance from boundary around obstacles, so as to avoid the collision between robot and obstacles. Aji SS and Triptysingh (2014) have proposed a methodology to plan a path for the mobile robot using camera and LASER emitting diode. LASER beam emitter creates a spot on the obstacle, camera capture the image including the laser spot. From the captured image identify the obstacles and calculate how far it is from the mobile robot. Initially captured image is processed and findout the distance between the mobile robot to obstacle using right angle triangle technique. Amol N. Dumbare *et al*. (2013) have completed a workto track a moving object by utilizing a mobile robot with sensor and image processing. The robotic platform uses a visual camera to sense the movement of the desired object and range sensor to help the robot to detect and avoid obstacle in real time while continue to track and follow the desired object finally reach the goal.Chung-Haochen *et al*. (2006) have proposed the robotic platform represents visual camera to sense the movement of desired object and range sensor to help the robot detect and then avoid obstacles in real time. The robot can continuously track the moving object through the dynamic goal potential field algorithmand it can guide the robot to move a new position without colliding the obstacles.

Hachour O (2008) has proposedan algorithm for path planning to reach the target specified for mobile robot in unknown environment. Intelligent Autonomous System allows a mobile robot to navigate through static obstacles and finding the path in order to reach the target without collision. The proposed path finding strategy is designed in a grid map for an unknown environment by sensing and avoiding the obstacles coming across its way towards the target. HimansuBorse *et al*. (2012)have proposed a strategy to track the moving object using the image processing technique along with sensors. The camera senses the movement of the object and IR sensor to avoid the obstacles in real time. IshayKamon *et al*. (1998) have proposed the tangent bug algorithm to navigate mobile robot in unknown environment using sensors. The tangent bug algorithm is a local path planning methodology which detects the nearest obstacle and target with limited information about the environment. It uses obstacle border as guidance toward the target. Here, the robot calculates tangent of the obstacle and moves towards the target. Masaaki tomita *et al*. (2009) have proposed a procedure to plan a path for the mobile robot start to goal point with obstacle avoidance. The robot move toward the goal by using its sensor information in an unknown environment. Sensor based navigation algorithms such as tangent bug algorithm assure the robot to reach the goal, but it may fail because of dead-locks in the presence of moving obstacles, because it works only for fixed obstacles in the unknown environment. Shahedshojaeipour *et al*. (2014) have proposeda work to determine the shortest path for the mobile robot to transverse the target while avoiding the static obstacles using image processing through the voronoi diagram.

Summary:

The mobile robot path planning tasks are performed by different researchers using different types of algorithms based on the application and its complexity. It is necessary to plan an optimal path with obstacle avoidance for complex environment. Generally, the optimal path is referred by lower computation effort and time along with avoiding the local minima. The above papers provide a better understanding in terms of computation time, shortest and collision free path with safety, optimal solution, high success rate and improvements of statistical significance related with path planning tasks. Generally any heuristic algorithm needs more information about the environment to execute the path planning task, but for the tangent bug algorithm needs only relatively limited information. In open field environment, the sequence of operation is continuously changing due to the nature of the environment. The each machining sequence of the product mix is considered as target point. So, the target is continuously changing depending upon the availability of the specified machine. In such environment, the tangent bug algorithm may be suitable.

Implementation Of Tangent Bug Algorithm:

Tangent bug algorithm is the derivative of the Bug1 and Bug 2 algorithms. They required a global knowledge about the environment but the tangent bug algorithm needs only the limited information to plan a path for mobile robot to perform the specified task. The tangent bug algorithm consists of two behaviors, one is the motion planning and other one is the boundary following.

Motion planning:

This behavior will be activated either, when there is no obstacle sensed on the direction from robot to the goal. Then mobile robot will find the distance to an obstacle and the distance from obstacle to goal point[8, 9] using Equations 1to 4. In Equation 4, 'atan2( )' is a mathematical function which is used to transform the cartesian coordinate values into polar coordinates. Here, this function is used to find out the angle of line generated between the start to goal with reference to horizontal line.

$$d_{reach} = \sqrt{\left(x_{i+1} - x_g\right)^2 + \left(y_{i+1} - y_g\right)^2} \quad (1)$$

$$x_g = x_{i+1} + \cos(\theta_g) \quad (2)$$

$$y_g = y_{i+1} + \sin(\theta_g) \quad (3)$$

$$\theta_g = \mathrm{atan2}\left(\left(y_{i+1} - y_g\right), \left(x_{i+1} - x_g\right)\right) \quad (4)$$

Boundary following:

When the boundary following behavior is initiated, the mobile robot start moving to the direction of the last discontinuity selected from heading towards the goal. With the direction selected, the robot moves on the tangent line. This tangent line is perpendicular to the line that connects the robot and the point closest to the obstacle [8, 9] using Equations 5 to 8.

$$d_{follow} = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \ldots \quad (5)$$

$$x_u = x_i + cos(\theta_f) \quad (6)$$

$$y_u = y_i + sin(\theta_f) \quad (7)$$

$$\theta_f = atan2((y_i - y_{i+1}), (x_i - x_{i+1})) \ldots \quad (8)$$

The value of d_{follow} is the shortest distance between mobile robot to obstacle boundary and d_{reach} is the shortest distance between mobile robot to the goal. Whenever the value of d_{reach} below the value of d_{follow}, the robot switch back to the motion planning behavior [8].In order to simulate the tangent bug algorithm, a mat lab code has been written with the following steps:

Step: 1:

Initially, a straight line is created between the start and the goal point using the input image, which is the representative image of the working environment may consists of different manufacturing machines like turning center, milling machine, drilling machine etc. The goal point is depends on the sequence of operation using the appropriate machine available within the working environment. The corresponding Mat lab code output is given in figure 1. In this output, a straight line has been draw from the initial to location of drilling machine, since here, the first operation is drilling operation.

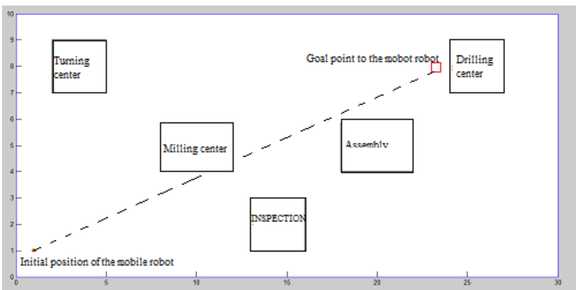

Fig. 1: Generation ofa straight line from start to drilling machine.

Step: 2:

Then, the mobile robot move toward the goal point using the coordinate values obtained using the Equations 2 and 3.

Step: 3:

If many obstacles are present along the initial path generated then mobile robot will select an obstacle which is closer to current position of mobile robot in the same direction towards the goal.

Step: 4:

Next, the boundary following will be activated to move to the mobile robot along the circumference of the obstacle using Equations 6 and 7 and the corresponding Mat lab output is given in figure 2.

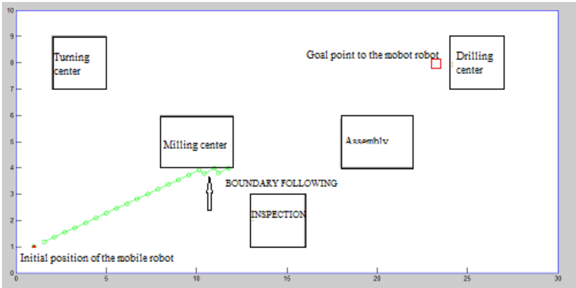

Fig. 2: Output image shown the boundary following activity.

Step: 5:

Next, the distance between the current position of the mobile robot to goal point will be calculated using the Equation 5 and then the mobile robot will reach the goal point. The corresponding Mat lab output is as shown in figure 3.

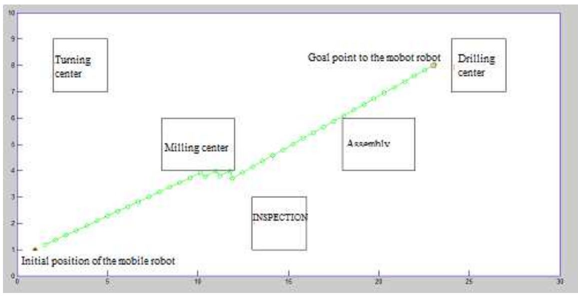

Fig. 3: Output image shown a path from start to goal.

RESULT AND DISCUSSION

The robot path planning for the test environment is simulated using Matlab coding. Here, an environment is considered with a sample product mix-1 with different operations as given in Table 1. Here, the distribution of materials or semi-finished component with respect to sample product mix-1 is implemented and simulated. The corresponding complete optimal path through simulation for the product mix-1 is obtained as shown in figure 4. The distance to be travelled by the mobile robot is also calculated.

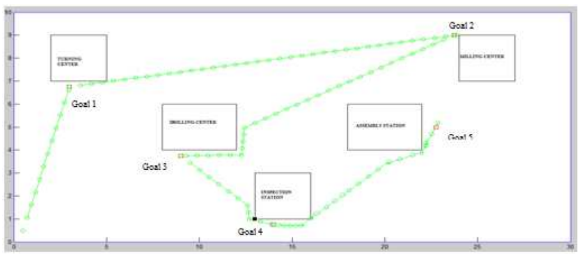

Fig. 4: Optimal path obtained for the sample product mix-1.

Table 1: Sequence of operations for the sample product mix-1.

S. No.	Sequence of operation	Coordinates of the optimal path of mobile robot		Euclidian distance in cm
		Start	End	
1.	Turning Operation	(00.50, 00.50)	(02.50, 06.75)	06.56
2.	Milling Operation	(02.50, 06.75)	(23.50, 09.00)	26.06
3.	Drilling Operation	(23.50, 09.00)	(08.00, 03.75)	16.36
4.	Inspection	(08.00, 03.75)	(13.00, 00.75)	05.83
5.	Assembly	(13.00, 00.75)	(26.00, 05.00)	13.68
Total distance travelled				68.49

Conclusion:

Simulation of mobile robot path planning based on tangent bug algorithm is implemented with a scope to apply in an open field manufacturing environment. A Mat lab code has been written to simulate the model environment. Further, the simulation may be extended with environment having moving obstacles.

REFERENCES

Abhishek Chandak, KetkiGosavi, ShalakaGiri, Sumeet Agrawal and Pooja Kulkarni, 2013. Path planning for Mobile Robot Navigation using Image processing. International Journal of Scientific and Engineering Research, 4: 1490-1496.

Aji, S.S. and Tripty Singh, 2014. Obstacle Detection for Navigation of Robot using Computer Vision and Laser Range finder. International Journal of Electrical, Electronics and Computer Systems, 2: 64-70.

Amol, N., Dumbare, Kiran P. Somase and Sachin, B. Bhosale, 2013. Mobile robot for Object Detection using Image processing. International journals of advance research in computer studies, 1: 81-84.

Chung-Hao Chen, Chang Cheng, David Page, Andreas Koschan and MongiAbidi, 2006. A Moving object Tracked by Mobile robot with Real-time Obstacles avoidance capacity. International Journal of Computer Science and Information Technologies, 5: 1091-1094.

Hachour, O., 2008. Path planning of Autonomous Mobile robot. International Journal of system application, Engineering and Development, 2: 178-190.

HimanshuBorse, Amol Dumbare, Rohit Gaikwad and Nikhil Lende, 2012. Mobile robot for Object detection using Image processing. Global Journal of Computer Science and Technology Neural and Artificial Intelligence, 12: 13-14.

Howie Choset, Kevin Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard and Sebastian Thrun, 2005. Principles of robot motion: Theory, Algorithm and Implementation. MIT Press.

IshayKamon, Elon Rimont and Ehud Rivlin, 1998. TangentBug: A Range-Sensor-Based Navigation Algorithm. International Journal of Robotics Research, 17: 934-953.

Masaaki Tomita and Motoji Yamamoto, 2009. A Sensor Based Navigation Algorithm for Moving Obstacles Assuring Convergence Property. IAPR Conference on Machine Vision Applications, 1: 295-299.

Mikell, P., Groover, 2007. Automation, Production system and Computer Aided Manufacturing. Prentice Hall of India Press.

Roland Siegwart and Illah R. Nourbakhsh, 2004. Introduction to Autonomous Mobile Robots. A Bradford Books, MIT press.

ShahedShojaeipour, Sallehuddin Mohamed Haris, ElhamGholami and Ali Shojaeipour, 2014. Webcambased Mobile Robot Path planning using Voronoi Diagram and Image processing. Proceedings of the 9thInternational Conference on applications of Electrical Engineering, 4: 151-156.