Path Planning through Maze Routing for a Mobile Robot with Nonholonomic Constraints

A.Makhal, M.Raj, K.Singh, P.Chakraborty and G.C.Nandi

***Abstract*— A comprehensive technique to plan path for a mobile robot with nonholonomic constraints through maze routing technique has been presented. Our robot uses a stereo vision based approach to detect the obstacles by creating dense 3D point clouds from the stereo images. ROS packages have been implemented on the robot for specific tasks of providing: i) Linear and angular velocity commands, ii) Calibration and rectification of the stereo images for generating point clouds, iii) Simulating the URDF (Unified Robot Description Format) module in real time, with respect to the real robot and iv) For visualizing the sensor data. For efficient path planning a hybrid technique using Lee's algorithm, modified by Hadlock and Soukup's algorithm has been implemented. Different path planning results have been shown using the maze routing algorithms. Preliminary results shows that Lee's algorithm is more time consuming in comparison with other algorithms. A hybrid of Lee's with Soukup's algorithm is more efficient but unpredictable for minimal path. A hybrid of Lee's with Hadlock's algorithm is the most efficient and least time consuming.**

I. Introduction

The most enticing capability of a mobile robot is to be able to explore its environment autonomously like any living creature on our planet which explores the environment. The most earnest local perception system needed is the visual perception system. Combining it with the other sensing system such as hearing and touch, improves the capability of achieving the goal of exploring the unstructured environment, decoupling the obstacle avoidance task. The theoretical underpinning for the task is to create a map of the environment and plan a safe path by avoiding obstacles. This is an essential requirement for a mobile robot.

For the task of robot localization, navigation and control, vision sensors have been extensively used. Navigation of robots by vision based control is very crucial and researchable task in field of robotics which has allured the attention of many researchers. The path-planning of a car-like robot with nonholonomic constraint for computationally optimized feasible trajectory is very consolidated and sophisticated task [1,3]. As the vision system of human being consists of two eyes, the stereovision technique for perceiving distance of the objects in its environment, is been highly employed for creating the perception system of the robot [9,10]. It is worth mentioning that the human being apprehends its environment in 3D. The same capability can be accomplished by storing the depth information of each pixel generated by the stereoscopic system in a point cloud format. PCL (Point Cloud Library) [11] presents an advanced and extensive approach to the subject of 3D perception, and it's meant to provide support for all the common 3D building blocks that robotics applications need. For path planning using Maze-routing, Lee algorithm [6] is an well-known method which is based on the Breadth-First Search (BFS) technique. It provides minimum path and guarantees to find the path if two nodes (start, destination) are present. Its drawback in the form of large memory consumption for dense layouts leads to the Hadlock algorithm [7] and Soukup algorithm [8] .

Fig. 1. The ANT (Autonomous Navigation Testbed) mobile robot used for the purpose of experiments of this paper.

In this paper we have proposed a comprehensive path planning technique for mobile robot by maze routing in indoor environment. Our focus is on modeling the environment in the robot's field of view by stereo images captured by the stereovision system equipped with the robot. The robot can recognize the obstacles by segmenting the 3D points generated from the stereo images. The considering environment is divided into small 2D grids which first have been labeled with "*unknown*" , "*free*" and "*occupied*" values. For generating the trajectory delimited by two points (initial, goal), Lee's algorithm has been used which labels the *free*

*Research supported by Indian Institute of Information technology, Allahabad (IIITA).

A. Makhal is a research scholar at the Robotics and AI Lab at IIITA. (e-mail: jontromanab.abhijit@gmail.com).

M. Raj and K. Singh are doing their M.Tech in Robotics from IIITA. (e-mail: rajmanish.03@gmail.com and karansingh4515@gmail.com).

P. Chakraborty is an Assistant professor at IIITA. (corresponding author phone:+91-532-2922121; fax: +91-532-2430006; e-mail: pavan@iiita.ac.in).

G. C. Nandi is a Professor and Dean (Academic) at IIITA (e-mail: gcnandi@iiita.ac.in).

and *occupied* grids with numbers depending on the distance from the initial position. By the Manhattan-distance of the final position from initial position, a path is planned. The linear and angular velocity parameters are generated for the trajectory considering the kinematic limitations of the mobile robot.

The structure of the paper is as follows .The system architecture is described in Section II. In Section III, algorithms used for path planning is discussed followed by the experimental results shown in Section IV. Conclusions are described in section V.

II. SYSTEM DESCRIPTION

The goal of the system is to achieve the desired position for the robot from the initial position using the visual control technique by a grid searching algorithm based on LEE's algorithm and hybrid techniques by Lee-Hadlock and Lee-Soukup. The visual control system relies on the images captured from the two cameras, which have been calculated to construct dense 3D point cloud data for detecting obstacle in the Field of View (FOV) of the robot.

A. HARDWARE DESCRIPTION

The hardware system of the experimental platform that has been used in this approach is called ANT (Autonomous Navigation Testbed), constitutes of a base of a car-like toy for children with differential drive wheels which is a nonholonomic mechanical system. As the term *nonholonomic* suggests the robot is incapable of moving in the direction of the axis of rear wheels. Our perception system constitutes of a custom made stereovision system with two Logitech C600 720p HD glass lens cameras fixed on a metal sheet having baseline of 10 centimeters. The stereo vision system is advantageous over a single camera system as it capture the dense 3D point cloud data for obstacle avoidance and landmark detection. An on-board laptop (Intel(R) core(TM) i5 label processor, 2.4 GHz) with sufficient graphics compatibilities handles the image processing tasks. The robot is equipped with two rear wheels which have been driven independently. For steering the robot two supplementary front wheels is been extensively used. Figure 1 shows our mobile robot with the stereoscopic system mounted on it.

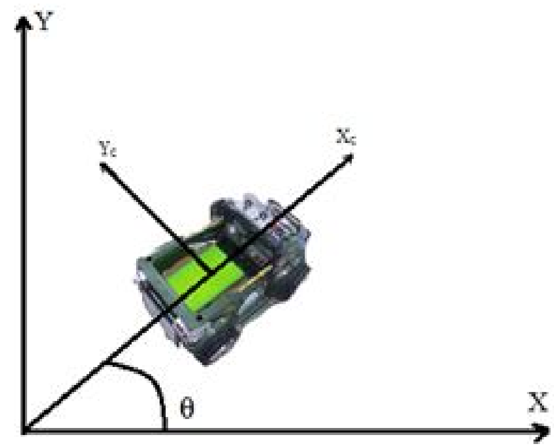
The main controller of the robot is the ATMega32 with Arduino platform which is been controlled by the processing unit by ROSSerial package of ROS. A LPY550AL integrated dual-axis angular rate sensor has been used as the gyro for gaining pose information of the robot.

As the hardware suggests our mobile robot is an example of a nonholonomic mechanical system. It consists of a vehicle with two driving wheels at the rear and two front wheels for steering the robot.

A mobile robot system with n-dimensional configuration space with generalized co-ordinates ($q_1,q_2,\ldots\ldots q_n$) with m constraints can be described as [1,2]

$$M(q)\ddot{q} + V_m(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - A^T(q)\lambda, \qquad (1)$$

Where M(q) is a symmetric ,positive definite inertia matrix, $V_m(q,\dot{q})$ is the centripetal and coriolis matrix, F($\dot{q}$) is the surface friction. G(q) is the gravitational vector , τ_d denotes bounded unknown disturbances including unstructured unmodeled dynamics, B(q) is the input transformation matrix, τ is the input vector ,A(q) is the matrix associated with the constraints and λ is the vector of constraint forces.



The desired motion and orientation can be achieved by providing two necessary torques in the driving and steering wheels. Figure 2 depicts the conceptual diagram of the differential type of the wheeled mobile robot working in an indoor environment. The position of the robot in the inertial Cartesian frame can be described by *{O,X,Y}* is specified by a vector $q=[x^c \quad y^c \quad \theta]^T$ where x^c and y^c denotes the co-ordinates of the center of mass at the plane and θ is the orientation of the robot with respect to the X axis. It is assumed that the center of mass is located at the middle of the axis of the rear wheels. v is the magnitude of the translational velocity and w is the angular velocity. We assume that the mobile base satisfies the conditions of *pure rolling and non slipping [3,4]* i.e., the velocity of the center of mass of the robot is orthogonal to the axis connecting the rear wheels. This assumption provides the nonholonomic constraint on the robot as:

$$\dot{y}_c \cos\theta - \dot{x}_c \sin\theta - d\,\dot{\theta} = 0 \qquad (2)$$

The nonholonomic constraint states that the robot can only move in the direction normal to the axis of the driving wheels .The kinematic model of the robot can be expressed as :

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_c \end{bmatrix} = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\sin\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix},$$

$$v = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \quad (3)$$

Where $[v1] \leq V_{max}$ and $[v2] \leq W_{max}$. V_{max} and W_{max} are the maximum linear and angular velocity of the robot respectively. Eq(3) describes the steering system of the robot.

The Lagrangian formulation can be used to determine the dynamics of the robot. As the trajectory of the mobile robot base is constrained on the horizontal plane G(q)=0.The potential energy, U of the robot is constant because the robot is incapable of moving in the vertical plane. The Kinetic energy [5] of the robot can be given as :

$$k_E^i = \frac{1}{2} m_i v_i^T v_i + \frac{1}{2} \omega_i^T I_i \omega_i, K_E = \sum_{i=1}^{n_i} k_E^i = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (4)$$

The dynamical model of the mobile base can be expressed as:

$$M(q) = \begin{bmatrix} m & 0 & md\cos\theta \\ 0 & m & -md\cos\theta \\ md\sin\theta & -md\cos\theta & I \end{bmatrix},$$

$$V(q,\dot{q}) = \begin{bmatrix} md\dot{\theta}^2\cos\theta \\ md\dot{\theta}^2\sin\theta \\ 0 \end{bmatrix},$$

$$G(q) = 0, \; B(q) = \frac{1}{r}\begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ R & -R \end{bmatrix},$$

$$\tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}, A^T(q) = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ -d \end{bmatrix},$$

$$\lambda = -m(\dot{x}_c\cos\theta + \dot{y}_c\sin\theta)\dot{\theta} \quad (5)$$

B. STEREO VISION SYSTEM

Stereo vision system is similar to the function of human vision system. The system's goal is to identify path and detect obstacle using the stereo images captured by the stereo vision system which consists of two Logitech C600 720p HD glass lens fixed with a baseline of 10 cm on a metal sheet.

Images from the two cameras get a shift between them which is proportional to the distance of the baseline. By this mechanism it is possible to determine depth of any point by its position in the two images with the following equation:

$$d = \frac{fT}{\|x_1 - x_2\|} \quad (6)$$

Where the focal distance of both the camera is *f*, T is the distance between the cameras, and x_1 and x_2 are the corresponding points in the two cameras.

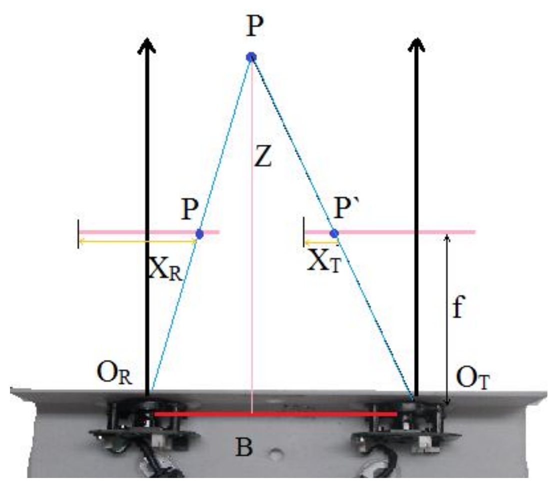


Fig. 3 .Canonical system of the stereoscopic system with two cameras

The Disparity is the distance on the X axis between a point P with its corresponding point P` in the other image. For making the images row aligned, it is necessary to rectify the images as for pixel matching the correlation routine searches along the x axis for matching the pixels in the two images. Stereo Calibration is used to establish geometrical relationship between the two stereo left and right cameras and calibrating the two images aligned with the epipolar lines using the OpenCV[13] libraries.

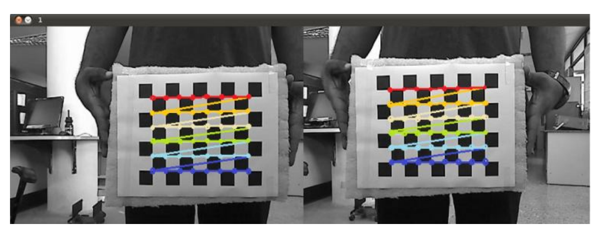


Fig. 4. Images captured by the two cameras for the calibration process

From the rectified image, disparity values for each pixel can be calculated by running the correlation routine on the image setup which leading to creating the disparity map. The depth of each pixel can be computed from the disparity map and further re-projected in to point cloud representation as the 3D point cloud.

A point cloud is generally a set of unrecognized, irregular points in 3D[11].A data clustering algorithm has been implemented on the point cloud data for segmenting each 2D point cloud as a separate set. The next step is the process of finding conclave-hull of the points for representing the each segmented set with a polygon[12]. In the end of the process, a smoothing algorithm is implemented through the vertices of each polygon for removing the very close straight line points. This reduces the complexity of the configuration space and the execution time of the path planner's execution time.

C. ROS Architecture

ROS defines as an open-source robot operating system which provides a structured communications layer above the host operating systems of a heterogeneous computer cluster [15]. In our system ROS has been extensively implemented for different purposes. We are not claiming that the system is fully ROS capable. But we have tried to implement some of the available ROS packages for solving different problems in our module.

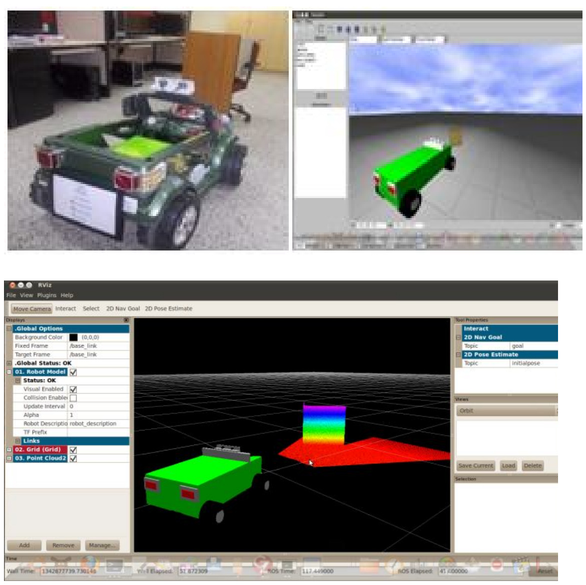


Fig 5. The model of our mobile robot ANT. From left to right and top to bottom: Robot model in real time, Robot model in Gazebo, Robot model and point cloud visualization in Rviz

ROSserial package has been used for interfacing the Arduino IDE for controlling the rear DC motors for driving and the front DC motor for steering the robot. ROSserial is a general protocol for sending the ROS messages over serial links using the rosserial_client library by creating a node on the host machine to bridge the connection from the serial protocol to the ROS network[14]. For navigating the robot, the ROS navigation stack has been implemented. The navigation package takes information from odometry, sensor streams, and a goal pose and outputs safe velocity commands that are sent to a mobile base. First the navigation stack has been configured with the shape and dynamics of the nonholonomic robot following a tf transform which maintains the relationship between coordinate frames in a tree structure buffered in time. Desired co-ordinate points in real time can be achieved using tf transform points, vectors, etc .

A Gazebo module of ANT has been created for testing our experiments in simulation. Gazebo which is the 3D simulated version of Player/Stage project, currently one of the most renowned 3D simulator for multiple robots. Gazebo requires ODE(Open Dynamic Engine),OPCODE collision Detection Library and GDAL(Geospatial Data Abstraction Library) for purposes like collision detections and terrain building. ANT has been modeled in gazebo by a URDF file consisting of the exact parameters of the robot.

For visualizing the sensor data and in real time RVIZ which is one of the most useful package in ROS, has been utilized. We have also implemented interactive markers for navigating the robot in real time with the help of RVIZ.

III . Path planning using maze Routing

For navigation purpose, we extract the path of robot using Lee algorithm [6] which is one of the most widely known maze routing method for finding path in a maze. Fig.[6]. represents the structure of the algorithm which can be characterized into three phases.

LEE Algorithm

1) Initialisation
Select start point, mark with 0
i := 0
2) Wave expansion
REPEAT
Mark all unlabeled neighbors of points marked with i with i+1
- i := i+1
UNTIL ((target reached) or (no points can be marked))
3) Backtrace
- go to the target point
REPEAT
- go to next node that has a lower mark than the actual node
- add this node to path
UNTIL (start point reached)
4) Clearance
- Block the path for future wirings
- Delete all marks

Fig. 6.. The structure of the Lee's algorithm

The first phase is called the Filling or Wave Propagation phase where the main work focuses on labeling the grids. The labeling process is similar of creating waves ripple outwards by dropping a pebble in a still pond. Starting from the initial position *I* as 0, all vertices at Manhattan-distance *k* from *I* to goal position *G* are labeled with k. The cell adjacent to the cell labeled *i* is given the label *i+1*. Retrace is the second phase, in which the route has been backtracked starting from the goal position *G* to the initial position *I*. The third and final phase is called Level Clearance where the labeled grids not used in the path, are cleared.

In Lee's algorithm, processing time for filling is proportional to L^2 and processing time for retrace is proportional to L, where L is the length of the path. So the time-complexity of the algorithm is $O(L^2)$ for each path. As a Propagation List (FIFO) is used to keep track of the vertices to be considered next, for a N*N grid plane, the algorithm uses $O(N^2)$ memory which requires great number of storage for storing the positions of the cells.

To reduce the running time and storage improvement of Lee's algorithm Hadlock[7] and Soukup[8] has been also used in our proposed system for advancing the wavefront with a higher priority towards the target direction. Comparing to Lee's algorithm this improvement provides 40-50% reduced running time. In Hadlock a new labeling technique namely detour number has been introduced where the detour number $d(P)$ of a path P connecting two cells I and G is defined as the number of grid cells directed away from its target G. Manhattan distance between I and G is $MD(S, T)$, then the length $L(P)$ of a path P is given by

$$L(P) = MD(S, T) + 2 \times d(P) \quad (7)$$

Soukup is a combination of BFS(Breadth First Search) and DFS(Depth First Search),as the search is directed toward the goal position G until an obstacle or G is reached. If an obstacle is reached, Lee algorithm is used to roam around the obstacle to find the path to the goal.

For our system, 2d array on the ground are considered as grid where we initialize the first grid which is closest to the robot as the start point I. A grid is chosen randomly as destination point G. Lee algorithm surly can find path if there exists a path between the two nodes (starting, destination). Only in the one case the robot will not find path in our experiment if obstacle covers all the range of camera.

IV. Experimental Results

In order to validate our proposed algorithm, several experiments have been performed with ANT. OpenCV[13] library has been used to capture and calibrate the two images and create the disparity map. The images captured from the two cameras of stereo-vision system shown in fig.[7a] have been used to create the disparity map in fig [7b]. The point cloud has been created by re-projecting the depth of each pixel on 2D plane. The RGB image capture from the right camera has been superimposed on the point cloud to create the image in fig[7c]. The depth image is shown in fig [7d]. The Fig[7e] shows the detected object.

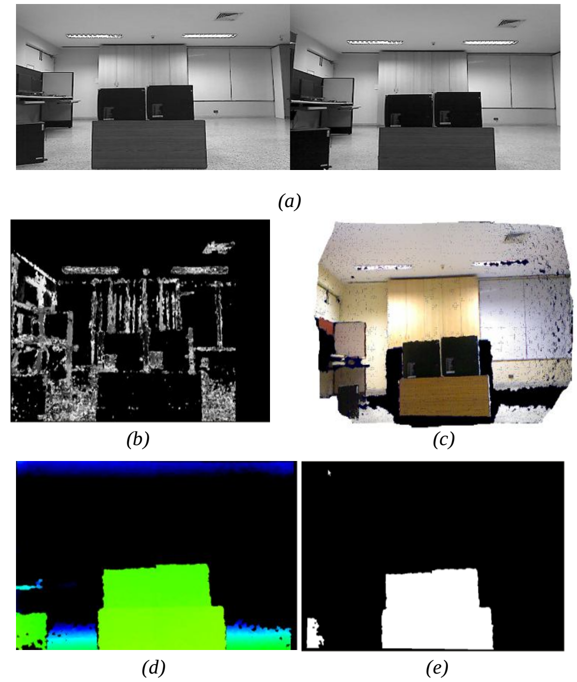


Fig. 7. The images by the robot in real time (From the left to right and top to bottom): a) Images taken from the left and right camera. b) Disparity map created by the two images. c) Point cloud image created by the disparity map with the superimposed RGB image from the right camera. d) The depth image perceived by the robot. e) Detected object in the FOV of the robot.

Table 1

TIME COMPLEXITY, DISTANCE TRAVELLED BY THE ROBOT AND ACCURACY FOR THE THREE DIFFERENT ALGORITHMS

Methods	Time Complexity	Distance Travelled by Robot (in m)	Average Accuracy (%)
Lee Algorithm	O(M,N)	12.70	80.5
Hadlock Algorithm	O(M,N)	9.87	88.0
Soukup Algorithm	O(M,N)	11.30	85.0

At the first stage Lee's Algorithm has been implemented for navigating the robot in the environment and avoiding the obstacle. Lee's algorithm is efficient but the time-space complexity and memory used is on the higher side. To improve the time-space complexity and economize the memory space, Soukup and Hadlock algorithm were implemented. Final results for the three algorithms considering time-complexity, distance travelled by the robot and average accuracy have been shown in the table[1]. The Soukup's algorithm is more efficient but unpredictable for minimal path.

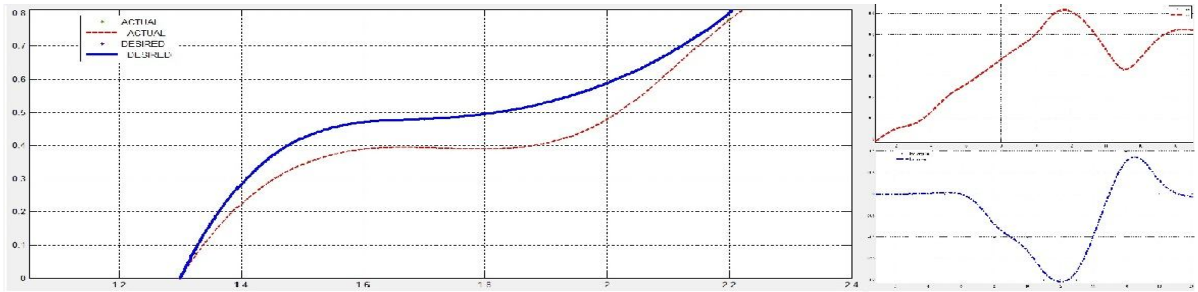


Fig. 8. The results obtained from the Lee's algorithm . Starting from the left to right and up to down: the desired and actual position of the robot in the Cartesian coordinates with respect to X and Y positions. Linear and angular velocity of the robot with respect to time

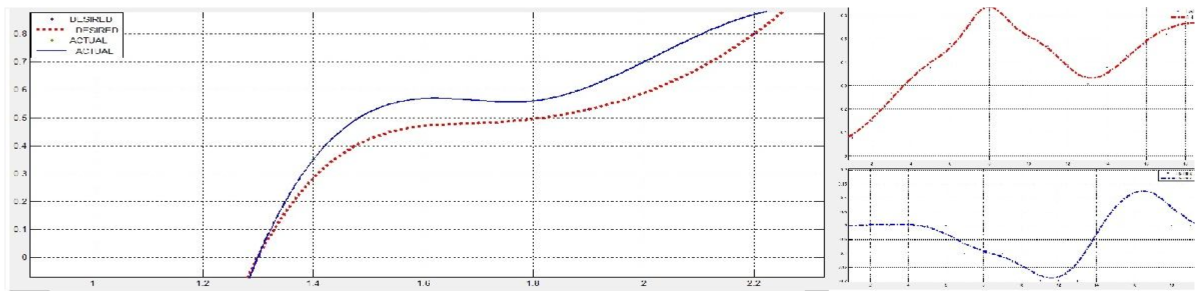


Fig. 9. . The results obtained from the Hadlock's algorithm. Starting from the left to right and up to down: the desired and actual position of the robot in the Cartesian coordinates with respect to X and Y .Linear and angular velocity of the robot with respect to time

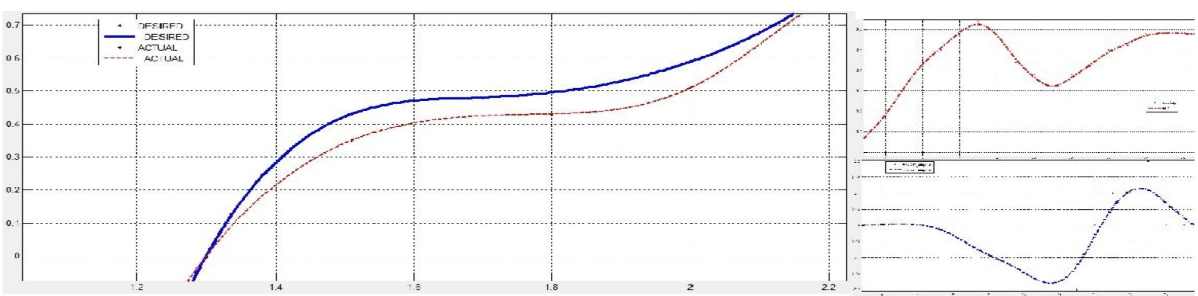
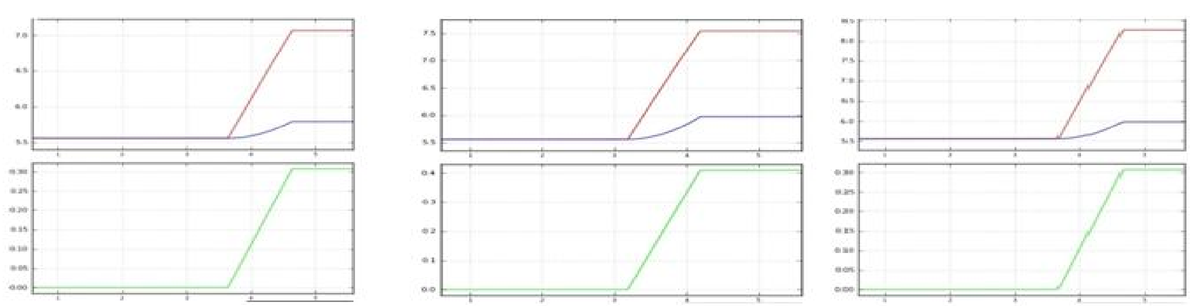


Fig. 10. The results obtained from the Soukup's algorithm. Starting from the left to right and up to down: the desired and actual position of the robot in the Cartesian coordinates with respect to X and Y .Linear and angular velocity of the robot with respect to time



(A) *(B)* *(C)*

Fig. 11. The results obtained from the rxplot in real-time *(a)*up: the robot pose with respect to time ,down: the robot orientation with respect to time using Lee's algorithm, *(b)*up:the robot pose with respect to time ,down: the robot orientation with respect to time using Hadlock's algorithm *(c)*up: the robot pose with respect to time ,down: the robot orientation with respect to time using Soukup's algorithm.

The Hadlock's algorithm was found to be the most efficient and least time consuming. Performance of the experiments has been extensively shown in the fig [8,9,10]. For the sake of completeness, we have also provided the graphs created by the rxplot, the package of ROS for showing the positions and orientation of the robot in real time. Fig.[11] depicts the graph of rxplot which has been captured in real time while navigating the robot with the algorithms, by which different results can be distinguishable.

V. CONCLUSION

In this paper we have integrated a comprehensive path-planning technique for mobile robot using the maze routing algorithms. In the first stage our robot uses the Lee's algorithm to find the path to the goal position from the initial position. As the result suggests the robot is getting average 80.5% accuracy in finding the path to the desired position avoiding the obstacle. The distance travelled by the robot using Lee's algorithm is 12.7 m .The accuracy found in Hadlock algorithm is 88%, which is computationally much higher than the Lee's Algorithm. On the other hand using Soukup's algorithm the robot gets 85% accuracy in achieving the task which is also higher than the Lee's algorithm. But the most undesirable drawback in Soukup's algorithm is that it does not guarantee to get the shortest path possible. It is also been experimented that in some few attempts the robot could not find the path to reach the goal using Soukup's algorithm.

As the point cloud is generated from the stereo images captured by the system, the stereo depth data is not as precise as data accused by laser systems. The depth data captured by stereo images are highly sensitive to the corresponding points found in two images depending on the light conditions and texture information, which results in high mismatch in corresponding frame in the process of clustering in real time.

In future work, we plan to continue our research in path planning of mobile robot through maze routing using visual sensors in more cluttered environment, as well as to integrate our approach with other path-planning algorithms.

REFERENCES

[1] N. Sarkar, X. Yun, and V. Kumar, ''Control of mechanical systems with rolling constraints: Application to dy-namic control of mobile robots,'' Int. J. Rob. Res., 13(1),55–69, 1994.

[2] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, Control of Robot Manipulators, MacMillan, New York, 1993

[3] Y. Yamamoto and X. Yun, ''Coordinating locomotion and manipulation of a mobile manipulator,'' in Recent Trends in Mobile Robots, Y. F. Zheng, Ed., World Scien-tific, 1993, pp. 157–181

[4] J. Barraquand and J.-C. Latombe, ''Nonholonomic multibody mobile robots: Controllability and motion

[5] R. Fierro* and F. L. Lewis, "Control of a Nonholonomic mobile robot: Backstepping kinematics into Dynamics" Journal of Robotic Systems 14(3), 149–163 (1997)

[6] C. Y. Lee, "An algorithm for path connections and its applications," IRE Transactions on Electronic Computers, vol. EC-10, pp. 346–365, 1961.

[7] F. O. Hadlock, "A shortest path algorithm for grid graphs", *Networks*, Vol. 7, No. 4, Winter 1977, pp. 323-334.

[8] J..Soukup, "Fast Maze Router," Proc. 15th Design Automation Conf., June 1978, pp. 100-102.

[9] S.S. Chen, W.H. Zuo and Z.L. Feng, "Depth Estimation via Stereo Vision Using Birchfield's Algorithm", IEEE 3rd International Conference on Communication Software and Networks (ICCSN), 2011, pp. 403-407

[10] L. Copertari, "Stereoscopic Vision for depth perception", Revista Digital de la Universidad Autonoma de Zacatecas, Nueva epoca, Publicacion cautrimestral, Mayo – Agosto 2007, volumen 3, numero 2.

[11] Rusu, R.B.; Cousins, S.; , "3D is here: Point Cloud Library (PCL)," *Robotics and Automation (ICRA), 2011 IEEE International Conference on* , vol.,no.,pp.1-4,9-13May,2011doi: 10.1109/ICRA.2011.5980567

[12] Rusu, R.B.; Marton, Z.C.; Blodow, N.; Beetz, M.; , "Learning informative point classes for the acquisition of object model maps," *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on* , vol., no., pp.643-650, 17-20 Dec. 2008 doi: 10.1109/ICARCV.2008.4795593

[13] G. Bradski and A. Kaehler, "Learning OpenCV", in O'REILLY Media Inc., ISBN: 978-0-596-51613-0Y

[14] http://www.ros.org

[15] Quigley, Morgan., Conley, Ken., Gerkey, Brian P.., Faust, Josh., Foote, Tully., Leibs, Jeremy., Wheeler, Rob., and Ng, Andrew Y.;" ROS: an open-source Robot Operating System" ICRA Workshop on Open Source Software,2009

Fig. 12. Scene example accuired during the experiments performed using ANT with maze routing algorithms